Hitachi Virtual Storage Software

1.12

Block REST API Reference

This document provides information and instructions on how to use the REST API Hitachi Virtual Storage Software block (VSS block). The REST API provides the Web API complying with the principles of REST (representational state transfer) to obtain VSS block information and change the configuration of the storage system.

MK-91VSSB003-03
February 2023

Hitachi reserves the right to make changes to this Material at any time without notice and assumes no responsibility for its use. The Materials contain the most current information available at the time of publication.

Some of the features described in the Materials might not be currently available. Refer to the most recent product announcement for information about feature and product availability, or contact Hitachi Vantara LLC at https://support.hitachivantara.com/en_us/contact-us.html.

Notice: Hitachi products and services can be ordered only under the terms and conditions of the applicable Hitachi agreements. The use of Hitachi products is governed by the terms of your agreements with Hitachi Vantara LLC.

By using this software, you agree that you are responsible for:

1. Acquiring the relevant consents as may be required under local privacy laws or otherwise from authorized employees and other individuals; and
2. Verifying that your data continues to be held, retrieved, deleted, or otherwise processed in accordance with relevant laws.

Notice on Export Controls. The technical data and technology inherent in this Document may be subject to U.S. export control laws, including the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations in other countries. Reader agrees to comply strictly with all such regulations and acknowledges that Reader has the responsibility to obtain licenses to export, re-export, or import the Document and any Compliant Products.

Contents

Preface

This document provides information and instructions on how to use the REST API for Hitachi Virtual Storage Software block (VSS block). The REST API provides the Web API complying with the principles of REST (representational state transfer) to obtain VSS block information and change the configuration of the storage system.

This manual applies to both the virtual machine and bare metal models of VSS block.

- Sections in this manual marked with (Virtual machine) apply to the virtual machine model.
- Sections in this manual marked with (Bare metal) apply to the bare metal model.

Please read this document carefully to understand how to use these products and maintain a copy for reference purposes.

Intended audience

This document is intended for system administrators, VPS(Virtual Private Storage) administrators, Hitachi Vantara representatives, and authorized service providers who are involved in installing, configuring, and operating Virtual Storage Software block.

Readers of this document should have at least the following knowledge and experience:

- Knowledge of networks
- Knowledge of Windows and Linux
- Programming skills using REST API
- Knowledge of hypervisor type virtualization environment

Product version

This document revision applies to Virtual Storage Software block version 1.12 (01.12.*xx*.*xx*).

The version in this document is described only by [aa.bb], and [aa.bb.cc.dd] is used only when required.

Release notes

Read the release notes before installing and using this product. They may contain requirements or restrictions that are not fully described in this document or updates or corrections to this document. Release notes are available on the Hitachi Vantara Support Website: https://knowledge.hitachivantara.com/Documents.

Changes made in this revision

- Added information about multi-tenancy support.
- Added information about spare node support.
- Added information about Q0S support.

Document conventions

This document uses the following typographic conventions:

Convention	Description
Bold	▪ Indicates text in a window, including window titles, menus, menu options, buttons, fields, and labels. Example: Click **OK**. ▪ Indicates emphasized words in list items.
Italic	▪ Indicates a document title or emphasized words in text. ▪ Indicates a variable, which is a placeholder for actual text provided by the user or for output by the system. Example: `pairdisplay -g` *`group`* (For exceptions to this convention for variables, see the entry for angle brackets.)
`Monospace`	Indicates text that is displayed on screen or entered by the user. Example: `pairdisplay -g oradb`
< > angle brackets	Indicates variables in the following scenarios: ▪ Variables are not clearly separated from the surrounding text or from other variables. Example: `Status-<`*`report-name`*`><`*`file-version`*`>.csv` ▪ Variables in headings.
[ ] square brackets	Indicates optional values. Example: [a | b] indicates that you can choose a, b, or nothing.
{ } braces	Indicates required or expected values. Example: { a | b } indicates that you must choose either a or b.
| vertical bar	Indicates that you have a choice between two or more options or arguments. Examples:

Convention	Description
	[a | b] indicates that you can choose a, b, or nothing. { a | b } indicates that you must choose either a or b.

This document uses the following icons to draw attention to information:

Icon	Label	Description
📄	Note	Calls attention to additional information.
💡	Tip	Provides helpful information, guidelines, or suggestions for performing tasks more effectively.
❗	Important	Highlights information that is essential to the completion of a task.
⚠	Caution	Warns the user of adverse conditions and/or consequences (for example, disruptive operations, data loss, or a system crash).
⚠	CAUTION	Warns the user of a hazardous situation that, if not avoided, could result in major or minor injury.
⚠	WARNING	Warns the user of a hazardous situation which, if not avoided, could result in death or serious injury.

Accessing product documentation

Product user documentation is available on the Hitachi Vantara Support Website: https://knowledge.hitachivantara.com/Documents. Check this site for the most current documentation, including important updates that may have been made after the release of the product.

Getting help

The Hitachi Vantara Support Website is the destination for technical support of products and solutions sold by Hitachi Vantara. To contact technical support, log on to the Hitachi Vantara Support Website for contact information: https://support.hitachivantara.com/en_us/contact-us.html.

Hitachi Vantara Community is a global online community for Hitachi Vantara customers, partners, independent software vendors, employees, and prospects. It is the destination to get answers, discover insights, and make connections. **Join the conversation today!** Go to community.hitachivantara.com, register, and complete your profile.

Comments

Please send comments to doc.comments@hitachivantara.com. Include the document title and number, including the revision level (for example, -07), and refer to specific sections and paragraphs whenever possible. All comments become the property of Hitachi Vantara LLC.

Thank you!

Chapter 1: Overview

This chapter describes the basic system configuration for using the REST API, how to specify requests, and information common to each API (for example, information required before executing API operation or referenced after the operation).

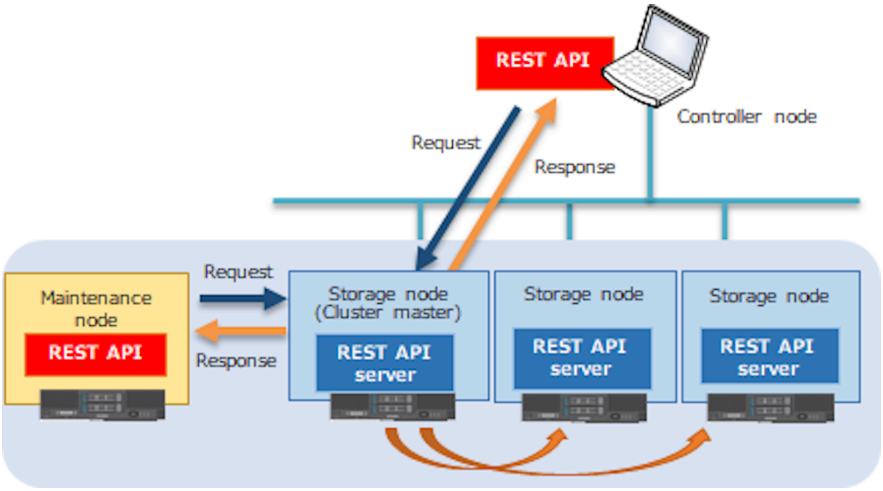
System configuration

This section describes the system configuration for using the REST API and the requirements for SSL/TLS communications.

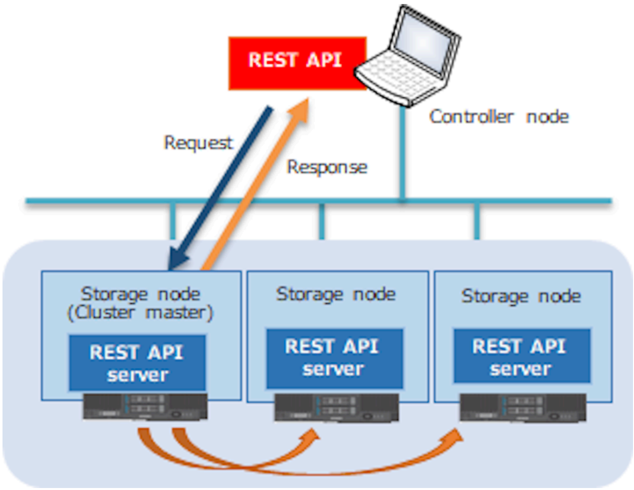
Basic system configuration

The following shows the basic system configuration of the REST API.

Virtual machine



Bare metal



REST API server

This is a component that acts as a server to receive and execute REST API requests from REST API clients, and then returns REST API responses to the REST API clients.

Each storage node has one REST API server in the system. The REST API server of the cluster master node generally accepts processing and distributes it throughout the system.

REST API server accepts a request for communications through the control port on the storage node.

REST API client

The REST API client issues a request from the controller node or maintenance node to the REST API server. The REST API client is a script or software that uses the REST API.

Supported HTTP method

The REST API provides the operations that can be performed on resources as the HTTP methods.

The REST API supports the following HTTP methods.

HTTP method	Description
GET	Obtains object information or an object list (for example, a list of volumes).
POST	Creates new objects (for example, volumes).
PATCH	Changes the attributes or statuses of objects (for example, nicknames of volumes).
DELETE	Deletes objects (for example, volumes).

Roles

The roles for the Virtual Storage Software block are as follows:

Security

Role for security management (browsing and setting). With this role, you can perform the following operations:

- Account management
- Log management (SNMP, Syslog, Email)
- IP network management and time setting
- Certificate management
- External authentication, login message, session management, CHAP authentication
- Encryption and key management

Storage

Role for storage cluster management (browsing and setting). With this role, you can perform the following operations:

- License management
- Storage pool, volume
- Compute port, LUN
- Snapshot
- Monitoring data management

Monitor

Role for storage cluster management (browsing only). Among the operations enabled for the storage role, only browsing is possible.

Service

Role for storage node maintenance. With this role, you can perform the following operations:

- Installation and uninstallation
- Configuration backup, restore
- Management of the configuration definition file
- Power supply management

- Updating storage software
- Fault analysis
- Storage node management (extension, maintenance, removal)

Audit

Role for audit logs (browsing and setting).

Resource

Role for managing (viewing and configuring) a virtual private storage (VPS).

None

A user can execute operations using any desired role.

APIs which do not need authentication itself are also included in this category.

To check whether authentication is required, see the *Authentication schemes* section at the bottom of each API description. If no authentication scheme is written, execute the API request without specifying an authentication scheme.

VpsSecurity

Role for VPS administrators to manage (viewing and configuring) security.

When you perform a configuration operation with this role, you can update only the resources on the VPS that you belong to. When you perform a view operation with this role, you can display only the resources on the VPS that you belong to.

With this role, you can perform the following operations:

- Account management

VpsStorage

Role for VPS administrators to manage (viewing and configuring) a storage cluster.

When you perform a configuration operation with this role, you can update only the resources on the VPS that you belong to. When you perform a view operation with this role, you can display only the resources on the VPS that you belong to.

With this role, you can perform the following operations:

- Volumes
- Compute ports
- Snapshots

VpsMonitor

Role for VPS administrators to manage (viewing) a storage cluster.

When you perform a view operation with this role, you can display only the resources on the VPS that you belong to.

Of the operations that can be performed with the VpsStorage role, only the view operation can be performed.

Authentication scheme

basic authentication

Basic HTTP authentication. For the Authorization header, specify a Base-64-encoded string of a user ID and password delimited with a colon (:). The following is an example of the Authentication header when the user ID is sample-user, and the password is sample-password.

Example: Authorization: Basic c2FtcGxlLXVzZXI6c2FtcGxlLXBhc3N3b3Jk

session authentication

Session authentication. For the Authorization header, specify a pre-generated token.

Example: Authorization: Session <*token*>

ticket authentication

Ticket authentication. For the Authorization header, specify a pre-generated ticket.

When you use ticket authentication, also specify the X-Ticket-Parameter header. For the X-Ticket-Parameter header, specify a Base-64-encoded string of a user ID and password delimited with a colon (:).

The following shows a ticket authentication example where the user ID is sample-user and the password is sample-password.

Example: Authorization: Ticket <*ticket*>

X-Ticket-Parameter: c2FtcGxlLXVzZXI6c2FtcGxlLXBhc3N3b3Jk

How to execute API requests

In the REST API, for resources to be managed, specify an operation in request URL format. The following shows the request URL format for the REST API.

```
<protocol>://<host-name>:<tcp-port-number>/ConfigurationManager/simple/<path>
```

- Protocol

  Specify https.

- Host name

  Specify the host name or the IP address (IPv4) of the REST API server on Virtual Storage Software block.

  <IP address>: The representative IP address of the storage cluster (or the corresponding host name).

  If ClusterIpv4Address in the VM configuration file (SystemConfigurationFile.csv) is set, specify the IP address (or the corresponding host name). If ClusterIpv4Address in the VM configuration file (SystemConfigurationFile.csv) is not set, specify the IP address (ControlNWIPv4 in the VM configuration file (SystemConfigurationFile.csv) or the corresponding host name) for the Control network of any storage node.

- TCP Port number

  Specify the port number used for network connection.

  To use SSL/TLS communications, specify 443.

- Path

  Specify each API-specific path.

Note:

- In this document, the following URL is described as a base URL:

  <*protocol*>://<*host-name*>:<*tcp-port-number*>/ConfigurationManager/simple

- Percent-encode a reserved string defined in RFC 3986, and then enter it as a URL.

  Example: When entering a symbol as userID to be specified in an API (GET /v1/objects/users/<*userId*>) that references the user

Curl-based REST API execution example

This section shows how a REST API is executed in version 7.55.1 of curl as a client software example.

```
curl -u <user-name>:<password> -i -X <HTTP-method> -H "<header-name-1>: <value-1>" [-
H "<header-name-2>: <value-2>" ...] [[-d "request body"]|[-F '<parameter-name-
1>=@"<path-1-to-the-file-to-be-uploaded>"' [-F '<parameter-name-2>=@"<path-2-to-the-
file-to-be-uploaded>."'..]]] https://<IP-address-of-Virtual-Storage-Software-Block>/
ConfigurationManager/simple<paths-to-APIs>[?<query-name-1>=<value-1>[&<query-name-
2>=<value-2>...]]
```

The following example shows how a REST API for volume creation is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
'{ "capacity":1024, "nicknameParam": { "baseName": "test-volume" }, "poolId":
"12345678-9abc-4def-8123-456789abcdef" }' https://192.168.1.100/ConfigurationManager/
simple/v1/objects/volumes
```

The following is an output example.

```
HTTP/1.1 202 Accepted
Content-Type:application/json

{"jobId":"58460547-10b5-4c6a-99e5-9b4d692ee056","self":"/ConfigurationManager/
simple/v1/objects/jobs/58460547-10b5-4c6a-99e5-9b4d692ee056","userId":"admin",
"status":"Initializing","state":"Queued","createdTime":"2019-07-24T00:13:02Z",
"updatedTime":"2019-07-24T00:13:02Z","completedTime":null,"request":{"requestUrl":"/
ConfigurationManager/simple/v1/objects/volumes","requestMethod":"POST",
"requestBody":"{"capacity":1024,"number":1,"nicknameParam":{"baseName":"test-volume",
"startNumber":null,"numberOfDigits":1},"poolId":"12345678-9abc-4def-8123-
456789abcdef"}"},"affectedResources":[],"error":null}
```

Tip:

To show how a REST API is executed, a curl execution example is included. It is assumed that a REST API is called from a program written in Java or Python.

Use the CLI to operate Virtual Storage Software block from the command line.

Caution:

- Depending on the specifications of the client software including curl, "Expect: 100-continue" might be added to the request header automatically when a large request body is specified.

  Since Virtual Storage Software block does not accept this "Expect" header and returns HTTP status code 417 (Expectation Failed), the API is not executed. For this reason, to execute an API with a request body specified, set the client software not to add an "Expect" header automatically, and then execute the API.

  In version 7.55.1 of curl, you can execute an API with no "Expect" header added automatically by specifying -H "Expect:".

- To execute a REST API whose HTTP method is POST without specifying a request body, specify "Content-Length: 0" for the request header.

  If you do not specify it, HTTP status code 411 (Length Required) is returned and the API is not executed.

- To execute an API by specifying a request body, specify "Content-Type" for the request header.

  If you do not specify it, HTTP status code 415 (Unsupported Media Type) is returned and the API is not executed. For the value to be specified in "Content-Type", see "REQUEST BODY" described in the description of each API.

HTTP status codes

After completion of request processing, the REST API returns an HTTP status code.

By checking the HTTP status code, you can verify whether the result of API processing is correct.

For a list of all the HTTP status codes used in the REST API and detailed information, see Response definitions (on page 575).

For details about the type of HTTP status code for each API, see each API description.

Note:

On rare occasions, it is possible and not abnormal that HTTP status code 503 (Service Unavailable) is returned, when all the following conditions are met:

- A file is specified as a parameter.

- Either of the following conditions are met:
  - Credentials (information of the Authorization header) is invalid.
  - A nonexistent URL or HTTP method is specified as an API.

Query parameters

When obtaining an object by the GET method, you can specify query parameters to filter the execution results based on specific conditions.

Query parameters can be specified at the end of the URL in the following format:

```
?<parameter>=<value>
```

To specify multiple query parameters, combine them with ampersands (&). The following example shows how to specify multiple query parameters:

```
?<parameter>=<value>&<parameter>=<value>&…
```

The query parameters you can specify differ with each API operation. For details, see each API description.

REST API input and output format

Use the JSON format for REST API requests and responses.

The supported character code is UTF-8.

The following shows the data types supported by the REST API and the corresponding data types in the json format.

Data type	JSON format data type	Description
boolean	boolean	Data type that represents true or false. Only lowercase letters can be specified. Example : true
int32	number	Data type that represents a 32-bit signed integer. Example: 100
int64	number	Data type that represents a 64-bit signed integer. Example: 1048576
string	string	Data type that represents a character string a user can specify arbitrarily. Example: "volume_1"
date-time	string	Data type that represents a time in ISO 8601 extended format (YYYY-MM-DDThh:mm:ssZ). UTC is the only time zone you can specify.

Data type	JSON format data type	Description
		Example : "2015-03-20T09:27:35Z"
link	string	Data type that represents a URL path. The link type indicates a URL for a resource. For example, when a request for asynchronous processing is issued, a URL for the job object is returned by using the link type. The link type data is a character string created by excluding the protocol name, host name, and port number from a URL. When creating a URL based on the link type data, add the protocol name, host name, and port number to it. If an ID is specified, a URL containing the ID is returned. Likewise, if a nickname is specified, a URL containing the nickname is returned. Example : "/ConfigurationManager/v1/objects/volumes/100"
uuid	string	Data type that represents UUID. Example : "9852ee63-7b57-47d6-9e05-7c2c052632d9"
double	number	Data type that represents a signed number including its decimal part. A minus ("-") sign is prefixed for a negative number. No sign is prefixed for a positive number. A number is output after a sign in the following format: ▪ 0.0 if the absolute value of a number is zero. ▪ Integral part, period ("."), and a one or more digit decimal number representing the decimal part in this order if the absolute value of the number is 10^{-3} or more to less than 10^{7}. ▪ Floating-point representation is used if the absolute value of the number is less than 10^{-3} or more than 10^{7}. The effective number of digits might be 16 or more.

In addition to the preceding data, the following JSON data is used.

- The object type

  A character string created by combining an attribute and a value with a colon (:), and then bracketing it with curly brackets {}. If more than one attribute-value pair exists, the pairs are separated by commas.

- The array type

  A character string consisting of multiple values separated by commas. The character string is bracketed with square brackets [].

Caution:

If the following three conditions are met when the data type of the response body is string and the string type of the attribute is determined (for example, volumeType in an API command that references a volume), "_NotSupported" might be output.

1. A string type does not exist in a version older than the running version.
2. The string type in 1 above is now output as a response body.
3. Downgrade to the old version in 1 above is attempted.

To restore "_NotSupported" string type to the original string type, upgrade to a newer version that supports the string type.

Note:

- If one of the following values is specified without being enclosed in double quotes for a string type parameter in the request body, it is treated as a character string.
  - true
  - false
  - A numeric value
- Enter values (without initial values padded with zeros) into the int32- and int64-type parameters. Zero padding of the initial values will cause an error.

  Example: When specifying 123
  - Valid entry: 123
  - Invalid entry: 00000123

Request header

The request header is a request message sent from the REST API clients to the REST API servers. The following table describes the request headers supported by the REST API.

Header	Category	Description	Specifiable value
Accept	Optional	Specifies the media type of a response.	*/* Default value: */*
Accept-Language	Optional	A language tag for messages that are received by clients.	en
Content-Type	Optional	Specifies the media type of a request body. When specifying a request body, a Content-Type header must be specified. If a request body is not specified, this header is ignored even if specified.	application/json multipart/form-data Default value: None
Content-Length	Optional	Specifies the size of a request body. You can specify a request body with a Content-Length header. "Content-Length: 0" is required when there is no request body in POST API. This header is automatically assigned depending on the specifications of the client software.	Specify the header in bytes. Default value: None
Authorization	Optional	Specifies the authentication method. The following three methods are available: ▪ basic authentication ▪ session authentication ▪ ticket authentication	For details about how to specify this header, see Authentication scheme (on page 24). Default value: None

Response header

The response header is a response message sent from the REST API servers to the REST API clients. The following table describes the response headers returned by the REST API servers.

Header	Description	Default
Content-Type	Shows the media type of the response data.	application/json;charset=UTF-8

Header	Description	Default
WWW-Authenticate	Returns the authentication information. ▪ When the HTTP status code 401 is returned: Indicates that authentication is required.	▪ When the HTTP status code 401 is returned: • When basic authentication did not succeed (including session creation failures) Basic realm="Block storage" • When session authentication did not succeed Session realm="Block storage"

Object

For data processing, the REST API sends and receives various objects as information, according to each API type.

For the object unique to each API operation, see each API description.

For a list of the objects used in the REST API, see Schema definitions (on page 580).

Uploading and downloading files

To upload files using the REST API, send the request body in the multipart/form-data format based on the multipart/form-data specifications in RFC7578 (https://tools.ietf.org/html/rfc7578).

Specify multipart/form-data as Content-Type of the request header.

The following is an example of an HTTP request.

```
POST /ConfigurationManager/simple/v1/objects/server-certificate/actions/import/invoke
HTTP/1.1
Authorization: Basic c2FtcGxlLXVzZXI6c2FtcGxlLXBhc3N3b3Jk
Content-Type: multipart/form-data
Content-Length: 6482
Content-Type: multipart/form-data; boundary=------------------------0f0789ece387a8bb

--------------------------0f0789ece387a8bb
Content-Disposition: form-data; name="serverCertificate"; filename="servercert.crt"
Content-Type: application/octet-stream

Certificate:
    Data:
(Contents of the file "servercert.crt". Omitted.)
-----END CERTIFICATE-----

--------------------------0f0789ece387a8bb
Content-Disposition: form-data; name="secretKey"; filename="serverkey.pem"
Content-Type: application/octet-stream

-----BEGIN RSA PRIVATE KEY-----
(Contents of the file "serverkey.pem". Omitted.)
-----END RSA PRIVATE KEY-----

--------------------------0f0789ece387a8bb--
```

When downloading a file, the data of the file is returned as the body of the response. For the format (for example, CSV) of the returned file, see each API specification.

Chapter 2: Audit log management

GET /v1/objects/audit-log-setting

Required Role: Security

DESCRIPTION

Obtains the audit log settings.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/audit-log-setting
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 231
Date: Fri, 30 Jul 2021 07:56:28 GMT

{"syslogForwardingSetting":{"locationName":"","syslogServers":[{"index":1,
"isEnabled":false,"serverName":"","port":514,"transportProtocol":"UDP"},{"index":2,
"isEnabled":false,"serverName":"","port":514,"transportProtocol":"UDP"}]}}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

auditLogSetting: object (on page 583)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

PATCH /v1/objects/audit-log-setting

Required Role: Security

DESCRIPTION

Edits the audit log settings.

REQUEST BODY

Uses default content-types: application/json

patchAuditLogSettingParam: object (on page 641)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d '{"syslogForwardingSetting":{"locationName":"HSP3F-Rack0320","syslogServers":
[{"index":1,"isEnabled":true,"serverName":"syslog_server1.hitachi.com","port":514,
"transportProtocol":"UDP"},{"index":2,"isEnabled":true,
"serverName":"syslog_server2.hitachi.com","port":514,"transportProtocol":"UDP"}]}}'
https://192.168.1.100/ConfigurationManager/simple/v1/objects/audit-log-setting
```

The following shows an example using a json body.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d @example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/audit-
log-setting
```

example.json is defined as follows:

```
{
    "syslogForwardingSetting":
        {
            "locationName":"HSP3F-Rack0320",
            "syslogServers":
                [
                    {
                        "index":1,
                        "isEnabled":true,
                        "serverName":"syslog_server1.hitachi.com",
                        "port":514,
                        "transportProtocol":"UDP"
                    },
                    {
                        "index":2,
                        "isEnabled":true,
                        "serverName":"syslog_server2.hitachi.com",
                        "port":514,
                        "transportProtocol":"UDP"
                    }
                ]
        }
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/audit-logs/actions/create-file/invoke

Required Role: Audit or Security

DESCRIPTION

Creates an audit log file. This API can be executed only for a cluster master node (primary). If this API is executed for any node other than a cluster master node (primary), HTTP status code 412 (Precondition Failed) is returned.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Length: 0" https://192.168.1.100/
ConfigurationManager/simple/v1/objects/audit-logs/actions/create-file/invoke
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)
Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/audit-logs/download

Required Role: Audit or Security

DESCRIPTION

Downloads an audit log file. This API can be executed only for a cluster master node (primary). If this API is executed for any node other than a cluster master node (primary), HTTP status code 412 (Precondition Failed) is returned.

If you have not created an audit log file, you may see message ID: KARS15553-E. In that case, create an audit log file.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -J -O -X GET -u admin:password https://192.168.1.100/ConfigurationManager/
simple/v1/objects/audit-logs/download
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
```

```
100  1326  100  1326    0     0   1461      0 --:--:-- --:--:-- --:--:--  1460
curl: Saved to filename 'download-auditlog.zip'
```

RESPONSES

Uses default content-types: application/zip, application/json

200 OK
Audit log file

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 412 (on page 577)
Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

Chapter 3: Common information management

GET /configuration/version

Required Role: None

DESCRIPTION

Obtains the API version information.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/
configuration/version
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 92
Date: Fri, 30 Jul 2021 07:56:13 GMT

{"apiVersion":"01.10.02.00","productName":"Hitachi Virtual Storage Software Block
REST API"}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

version: object (on page 727)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

GET /v1/objects/jobs

Required Role: None

DESCRIPTION

Obtains a list of information items on jobs submitted from REST API by the user.

Logs are listed in descending order of the date and time when they were created.

Only information about the jobs for APIs executable for the role the user has can be obtained.

If you have the VPS administrator role, you can obtain the information about only your own jobs.

Jobs to be obtained can be filtered by using query parameter "userId".

- all: Information about all jobs for APIs executable for the local user is obtained.
- self: A list of jobs submitted by the local user is obtained (default).
- User ID: A list of jobs submitted by using the specified user ID is obtained.

The following describes the relationship between the specification and output of the startCreatedTime and endCreatedTime query parameters.

When both startCreatedTime and endCreatedTime are omitted

- Assuming that the following are specified, outputs job information in reverse chronological order of the date and time when jobs were created:
  - The maximum value (9999-12-31T23:59:59Z) was specified as endCreatedTime.
  - 1970-01-01T00:00:01Z was specified as startCreatedTime.

When both startCreatedTime and endCreatedTime are specified

- Logs within the specified period are listed in reverse chronological order of the date and time when jobs were created.

When only startCreatedTime is specified

- Assuming that the maximum value (9999-12-31T23:59:59Z) was specified as endCreatedTime, outputs job information within the specified period in reverse chronological order of the date and time when jobs were created.

When only endCreatedTime is specified

- Assuming that 1970-01-01T00:00:01Z was specified as startCreatedTime, outputs job information within the specified period in reverse chronological order of the date and time when jobs were created.

When the rage specified using startCreatedTime and endCreatedTime is invalid

- HTTP status code 400 (Bad Request) is returned if either of the following applies:
  - A date and time after endCreatedTime is specified as startCreatedTime, including when endCreatedTime is not specified.
  - The same date and time is specified for startCreatedTime and for endCreatedTime, including cases when startCreatedTime is not specified and 1970-01-01T00:00:01Z is specified for endCreatedTime.
  - A date and time earlier than 1970-01-01T00:00:01Z is specified for startCreatedTime or endCreatedTime.

REQUEST PARAMETERS

Name	Description	Type	Data type
startCreatedTime	The start date and time, including the specified time, of job submission.	query	*string* (date-time)
endCreatedTime	The end date and time, excluding the specified time, of job submission.	query	*string* (date-time)
count	The maximum number of obtained job information items.	query	*integer* , { x ∈ ℤ | 1 ≤ x ≤ 100 } (default: "100")

Name	Description	Type	Data type
status	The progress of the job.	query	*string* , x ∈ { "Initializing" , "Running" , "Completed" }
state	Job state.	query	*string* , x ∈ { "Queued" , "Started" , "Stopping" , "Succeeded" , "Failed" , "Stopped" , "Unknown" }
userId	User ID. By specifying all, you can obtain information about all jobs for APIs executable for the local user. By specifying self, you can obtain a list of jobs submitted by the local user.	query	*string*(3 to 255 chars), must match /^all$\|^self$|^[\-A-Za-z0-9!#\$%&'\.@\^_`\{\}~]{5,255}$/
ids	Job IDs. You can specify plural IDs up to 32 with comma (,) delimiter. Example: ids=<*id1*>,<*id2*>,<*id3*>… HTTP status code 400 (Bad Request) is returned when the following conditions are met: ▪ 33 or more IDs are specified. ▪ A character string other than UUID is specified (including when a space character exists before or after a comma, or when the string is an empty string).	query	*string*

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/jobs
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
```

```
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 634
Date: Mon, 08 Nov 2021 08:00:50 GMT

{"data":[{"jobId":"c7efc074-8c8d-4cc6-a31e-4197cc48c953","self":"/
ConfigurationManager/simple/v1/objects/jobs/c7efc074-8c8d-4cc6-a31e-4197cc48c953",
"userId":"admin","status":"Completed","state":"Succeeded","createdTime":"2021-11-
08T08:00:37Z","updatedTime":"2021-11-08T08:00:37Z","completedTime":"2021-11-
08T08:00:37Z","request":{"requestUrl":"/ConfigurationManager/simple/v1/objects/chap-
users","requestMethod":"POST","requestBody":"{\"targetChapUserName\":\"sample-user\",
\"targetChapSecret\":********}"},"affectedResources":["/ConfigurationManager/
simple/v1/objects/chap-users/661908f2-1bdb-4606-b6ee-d52d2b754a01"],"error":null}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of job information.

PROPERTIES

data: ***object[]***

ITEMS

job: object (on page 627)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/jobs/<*jobId*>

Required Role: None

DESCRIPTION

Obtains a job.

A job can be obtained only when the user has a role that can run the API that issued the job.

If you have the VPS administrator role, you can obtain the information about only your own jobs.

REQUEST PARAMETERS

Name	Description	Type	Data type
jobId	Job ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/jobs/76f627f7-d2a3-4867-ac1e-55d1953cf555
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 628
Date: Fri, 30 Jul 2021 07:56:53 GMT

{"jobId":"76f627f7-d2a3-4867-ac1e-55d1953cf555","self":"/ConfigurationManager/
simple/v1/objects/jobs/76f627f7-d2a3-4867-ac1e-55d1953cf555","userId":"admin",
"status":"Completed","state":"Succeeded","createdTime":"2021-07-30T07:28:03Z",
"updatedTime":"2021-07-30T07:28:03Z","completedTime":"2021-07-30T07:28:03Z","request":
{"requestUrl":"/ConfigurationManager/simple/v1/objects/chap-users",
"requestMethod":"POST","requestBody":"{\"targetChapUserName\":\"target_chap_user\",
\"targetChapSecret\":********}"},"affectedResources":["/ConfigurationManager/
simple/v1/objects/chap-users/bdb358f6-0a31-49a2-b5e9-831b9f4ad38f"],"error":null}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

job: object (on page 627)

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

Chapter 4: Compute node connection management

GET /v1/objects/chap-users

Required Role: Security

DESCRIPTION

Obtains a list of CHAP user information.

REQUEST PARAMETERS

Name	Description	Type	Data type
targetChapUserName	CHAP user name used for CHAP authentication on the compute port (i.e., target side) (exact match).	path	*string*

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/chap-users
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 123
Date: Fri, 30 Jul 2021 07:55:55 GMT
```

```
{"data":[{"id":"bdb358f6-0a31-49a2-b5e9-831b9f4ad38f",
"targetChapUserName":"target_chap_user","initiatorChapUserName":""}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of CHAP user summary information.

PROPERTIES

data:*object[]*

ITEMS

chapUserSummary: object (on page 586)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/chap-users

Required Role: Security

DESCRIPTION

Creates a CHAP user.

REQUEST BODY

Uses default content-types: application/json

createChapUserParam: object (on page 591)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
'{ "targetChapUserName": "CUser", "targetChapSecret": "123456789012"}' https://
192.168.1.100/ConfigurationManager/simple/v1/objects/chap-users
```

The following shows an example using a json body.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
@example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/chap-users
```

example.json is defined as follows:

```
{
    "targetChapUserName": "CUser",
    "targetChapSecret": "123456789012"
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)

Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 409 (on page 577)

Indicates that an inconsistent or impossible request was issued for the resource specified by URL (for example, creating a resource whose ID is same as the resource already created).

HTTP status code 411 (on page 577)

Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)

Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

DELETE /v1/objects/chap-users/<*chapUserId*>

Required Role: Security

DESCRIPTION

Deletes a CHAP user.

REQUEST PARAMETERS

Name	Description	Type	Data type
chapUserId	The ID of the CHAP user.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -X DELETE -u admin:password https://192.168.1.100/ConfigurationManager/simple/v1/
objects/chap-users/bdb358f6-0a31-49a2-b5e9-831b9f4ad38f
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/chap-users/<*chapUserId*>

Required Role: Security

DESCRIPTION

Obtains the information about a CHAP user.

REQUEST PARAMETERS

Name	Description	Type	Data type
chapUserId	The ID of the CHAP user.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/chap-users/bdb358f6-0a31-49a2-b5e9-831b9f4ad38f
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
```

```
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 163
Date: Fri, 30 Jul 2021 07:56:35 GMT

{"portIds":["343db11a-21d0-45eb-a66c-c31ab1ac6863"],"id":"bdb358f6-0a31-49a2-b5e9-
831b9f4ad38f","targetChapUserName":"target_chap_user","initiatorChapUserName":""}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

chapUser: object (on page 586)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

PATCH /v1/objects/chap-users/<*chapUserId*>

Required Role: Security

DESCRIPTION

Edits the information about a CHAP user.

REQUEST BODY

Uses default content-types: application/json

patchChapUserParam: object (on page 641)

REQUEST PARAMETERS

Name	Description	Type	Data type
chapUserId	The ID of the CHAP user.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d '{ "targetChapUserName": "CUserEdit", "targetChapSecret": "012345678901"}' https://
192.168.1.100/ConfigurationManager/simple/v1/objects/chap-users/5d7848e6-88d8-4252-
965c-b098954607a0
```

The following shows an example using a json body.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d @example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/chap-
users/5d7848e6-88d8-4252-965c-b098954607a0
```

example.json is defined as follows:

```
{
    "targetChapUserName": "CUserEdit",
    "targetChapSecret": "012345678901"
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)

Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/port-auth-settings/<*id*>

Required Role: Security

DESCRIPTION

Obtains the authentication settings for the compute port for the target operation.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the compute port.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/port-auth-settings/343db11a-21d0-45eb-a66c-c31ab1ac6863
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 116
Date: Fri, 30 Jul 2021 07:56:35 GMT

{"id":"343db11a-21d0-45eb-a66c-c31ab1ac6863","authMode":"None",
"isDiscoveryChapAuth":false,"isMutualChapAuth":false}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

portAuthSetting: object (on page 662)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

PATCH /v1/objects/port-auth-settings/<*id*>

Required Role: Security

DESCRIPTION

Edits the authentication settings for the compute port for the target operation.

REQUEST BODY

Uses default content-types: application/json

patchPortAuthSettingParam: object (on page 643)

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the compute port.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d '{ "authMode": "CHAP", "isMutualChapAuth": "true"}' https://192.168.1.100/
ConfigurationManager/simple/v1/objects/port-auth-settings/eaef1f2a-f101-4ced-9d56-
ba43d33700dc
```

The following shows an example using a json body.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d @example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/port-
auth-settings/eaef1f2a-f101-4ced-9d56-ba43d33700dc
```

example.json is defined as follows:

```
{
    "authMode": "CHAP",
    "isMutualChapAuth": "true"
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)

Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/port-auth-settings/<*id*>/chap-users

Required Role: Security

DESCRIPTION

Obtains a list of information about a CHAP user who is allowed to access the compute port.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the compute port.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/port-auth-settings/343db11a-21d0-45eb-a66c-c31ab1ac6863/chap-users
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 123
Date: Fri, 30 Jul 2021 07:56:36 GMT

{"data":[{"id":"bdb358f6-0a31-49a2-b5e9-831b9f4ad38f",
"targetChapUserName":"target_chap_user","initiatorChapUserName":""}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of information about a CHAP user who is allowed to access the compute port.

PROPERTIES

data:*object[]*

ITEMS

approvedChapUser: object (on page 582)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/port-auth-settings/<*id*>/chap-users

Required Role: Security

DESCRIPTION

Allows a CHAP user to access the compute port.

REQUEST BODY

Uses default content-types: application/json

approveChapUserParam: object (on page 582)

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the compute port.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
'{ "chapUserId": "5d7848e6-88d8-4252-965c-b098954607a0"}' https://192.168.1.100/
```

```
ConfigurationManager/simple/v1/objects/port-auth-settings/eaef1f2a-f101-4ced-9d56-
ba43d33700dc/chap-users
```

The following shows an example using a json body.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
@example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/port-auth-
settings/eaef1f2a-f101-4ced-9d56-ba43d33700dc/chap-users
```

example.json is defined as follows:

```
{
    "chapUserId": "5d7848e6-88d8-4252-965c-b098954607a0"
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)

Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 409 (on page 577)

Indicates that an inconsistent or impossible request was issued for the resource specified by URL (for example, creating a resource whose ID is same as the resource already created).

HTTP status code 411 (on page 577)

Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

DELETE /v1/objects/port-auth-settings/<*id*>/chap-users/<*chapUserId*>

Required Role: Security

DESCRIPTION

Cancels compute port access permission for a CHAP user.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the compute port.	path	*string* (uuid) required
chapUserId	The ID of the CHAP user.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -X DELETE -u admin:password https://192.168.1.100/ConfigurationManager/simple/v1/
objects/port-auth-settings/343db11a-21d0-45eb-a66c-c31ab1ac6863/chap-users/bdb358f6-
0a31-49a2-b5e9-831b9f4ad38f
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/port-auth-settings/<*id*>/chap-users/<*chapUserId*>

Required Role: Security

DESCRIPTION

Obtains the information about a CHAP user who is allowed to access the compute port.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the compute port.	path	*string* (uuid) required
chapUserId	The ID of the CHAP user.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/port-auth-settings/343db11a-21d0-45eb-a66c-c31ab1ac6863/chap-users/bdb358f6-
0a31-49a2-b5e9-831b9f4ad38f
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 112
Date: Fri, 30 Jul 2021 07:56:36 GMT

{"id":"bdb358f6-0a31-49a2-b5e9-831b9f4ad38f","targetChapUserName":"target_chap_user",
"initiatorChapUserName":""}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

approvedChapUser: object (on page 582)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/ports

Required Role: Security, Storage, Monitor, Service, Resource, VpsStorage, or VpsMonitor

DESCRIPTION

Obtains a list of compute port information.

REQUEST PARAMETERS

Name	Description	Type	Data type
protocol	The protocol for connecting compute ports.	query	*string* , x ∈ { "FC" , "iSCSI" }
storageNodeId	Storage node ID.	query	*string* (uuid)
name	The WWN for FC connection or iSCSI name for iSCSI connection of the compute port. For this parameter, specify a WWN or iSCSI name as a case-sensitive exact match.	query	*string*
names	The WWN for FC connection or iSCSI name for iSCSI connection of the compute port. You can specify plural names up to 32 with comma (,) delimiter. Example: names=<*name1*>,<*name2*>,<*name3*>… If you enter a space character after a comma, the space character is recognized as a part of the name specified after the comma. Also, if you specify 33 or more names, HTTP status code 400 (Bad Request) is returned. For this parameter, specify a WWN or iSCSI name as a case-sensitive exact match.	query	*string* , Each item must match /^[a-f0-9]{16}$\|^[A-F0-9]{16}$|^((iqn\.[0-9]{4}\-[0-9]{2}\.[a-zA-Z0-9\-:\.]{0,211})|(eui\.[0-9a-fA-F]{16}))$/

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/ports
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 5290
Date: Fri, 30 Jul 2021 07:55:55 GMT

{"data":[{"id":"9f066e39-fd8e-4545-b1d2-120d44a94b45","protocol":"iSCSI",
"type":"Universal","nickname":"002-iSCSI-002","name":"iqn.1994-
04.jp.co.hitachi:rsd.sph.t.0d4c2.002","configuredPortSpeed":"Auto",
"portSpeed":"_NotSupported","portSpeedDuplex":"_NotSupported",
"protectionDomainId":"1b34a094-9bd3-4bdf-9b81-7247ad8b0e39","storageNodeId":"3123caa9-
f13a-4905-a6ee-815ffb7f98a2","interfaceName":"eth2","statusSummary":"Normal",
"status":"Normal","fcInformation":null,"iscsiInformation":{"ipMode":"ipv4",
"ipv4Information":{"address":"192.168.142.243","subnetMask":"255.255.192.0",
"defaultGateway":""},"ipv6Information":{"linklocalAddressMode":"Auto",
"linklocalAddress":"","globalAddressMode":"Manual","globalAddress1":"",
"subnetPrefixLength1":0,"defaultGateway":""},"delayedAck":true,"mtuSize":1500,
"macAddress":"80:30:e0:1f:d3:10","isIsnsClientEnabled":false,"isnsServers":
[{"index":1,"serverName":"","port":3205}]}},{"id":"a928b416-cf8d-4443-aa08-
eeea2000f694","protocol":"iSCSI","type":"Universal","nickname":"001-iSCSI-000",
"name":"iqn.1994-04.jp.co.hitachi:rsd.sph.t.0d4c2.000","configuredPortSpeed":"Auto",
"portSpeed":"_NotSupported","portSpeedDuplex":"_NotSupported",
"protectionDomainId":"1b34a094-9bd3-4bdf-9b81-7247ad8b0e39","storageNodeId":"ad36eab7-
20c3-41cd-acac-d334ff5347d7","interfaceName":"eth2","statusSummary":"Normal",
"status":"Normal","fcInformation":null,"iscsiInformation":{"ipMode":"ipv4",
"ipv4Information":{"address":"192.168.142.242","subnetMask":"255.255.192.0",
"defaultGateway":""},"ipv6Information":{"linklocalAddressMode":"Auto",
"linklocalAddress":"","globalAddressMode":"Manual","globalAddress1":"",
"subnetPrefixLength1":0,"defaultGateway":""},"delayedAck":true,"mtuSize":1500,
"macAddress":"80:30:e0:1e:9d:c8","isIsnsClientEnabled":false,"isnsServers":
[{"index":1,"serverName":"","port":3205}]}},{"id":"d3e31ad2-c763-4af3-9455-
7ce1c829531e","protocol":"iSCSI","type":"Universal","nickname":"000-iSCSI-001",
"name":"iqn.1994-04.jp.co.hitachi:rsd.sph.t.0d4c2.001","configuredPortSpeed":"Auto",
"portSpeed":"_NotSupported","portSpeedDuplex":"_NotSupported",
"protectionDomainId":"1b34a094-9bd3-4bdf-9b81-7247ad8b0e39","storageNodeId":"8e01e189-
38bc-4002-a1fe-7a73633bce8c","interfaceName":"eth2","statusSummary":"Normal",
"status":"Normal","fcInformation":null,"iscsiInformation":{"ipMode":"ipv4",
"ipv4Information":{"address":"192.168.142.241","subnetMask":"255.255.192.0",
"defaultGateway":""},"ipv6Information":{"linklocalAddressMode":"Auto",
"linklocalAddress":"","globalAddressMode":"Manual","globalAddress1":"",
"subnetPrefixLength1":0,"defaultGateway":""},"delayedAck":true,"mtuSize":1500,
"macAddress":"80:30:e0:1e:9d:60","isIsnsClientEnabled":false,"isnsServers":
[{"index":1,"serverName":"","port":3205}]}}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of summary information about the compute port.

PROPERTIES

data: ***object[]***

ITEMS

portSummary: object (on page 664)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/ports/<*id*>

Required Role: Security, Storage, Monitor, Service, Resource, VpsStorage, or VpsMonitor

DESCRIPTION

Obtains the compute port information.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the compute port.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/ports/343db11a-21d0-45eb-a66c-c31ab1ac6863
HTTP/1.1 200 OK
```

```
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 879
Date: Fri, 30 Jul 2021 07:56:37 GMT

{"id":"343db11a-21d0-45eb-a66c-c31ab1ac6863","protocol":"iSCSI","type":"Universal",
"nickname":"005-iSCSI-001","name":"iqn.1994-04.jp.co.hitachi:rsd.sph.t.04b7a.001",
"configuredPortSpeed":"Auto","portSpeed":"DependsOnHypervisor",
"protectionDomainId":"0778a123-42e5-43ff-8fbc-c8580b79f2cf","storageNodeId":"8fdbc008-
ec6c-42b3-b930-146c94c5cef9","interfaceName":"eth2","statusSummary":"Normal",
"status":"Normal","fcInformation":null,"iscsiInformation":{"ipMode":"ipv4",
"ipv4Information":{"address":"192.168.137.216","subnetMask":"255.255.192.0",
"defaultGateway":""},"ipv6Information":{"linklocalAddressMode":"Auto",
"linklocalAddress":"","globalAddressMode":"Manual","globalAddress1":"",
"subnetPrefixLength1":0,"defaultGateway":""},"delayedAck":true,"mtuSize":1500,
"macAddress":"00:50:56:bd:6f:15","isIsnsClientEnabled":false,"isnsServers":
[{"index":1,"serverName":"","port":3205}]}}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

port: object (on page 660)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

PATCH /v1/objects/ports/<*id*>

Required Role: Storage

DESCRIPTION

Edits the compute port settings.

REQUEST BODY

Uses default content-types: application/json

patchPortParam: object (on page 644)

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the compute port.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d '{ "nickname": "002-iSCSI-100", "name": "iqn.1994-
04.jp.co.hitachi:rsd.sph.t.0b06a.000"}' https://192.168.1.100/ConfigurationManager/
simple/v1/objects/ports/eaef1f2a-f101-4ced-9d56-ba43d33700dc
```

The following shows an example using a json body.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d @example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/ports/
eaef1f2a-f101-4ced-9d56-ba43d33700dc
```

example.json is defined as follows:

```
{
    "nickname": "002-iSCSI-100",
    "name": "iqn.1994-04.jp.co.hitachi:rsd.sph.t.0b06a.000"
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

Chapter 5: Compute node management

GET /v1/objects/servers

Required Role: Security, Storage, Monitor, Service, Resource, VpsSecurity, VpsStorage, or VpsMonitor

DESCRIPTION

Obtains a list of compute node information.

REQUEST PARAMETERS

Name	Description	Type	Data type
nickname	The compute node nickname (exact match).	query	*string*
nicknames	A list of the nicknames of the compute nodes (exact match). You can specify plural nicknames up to 32 with comma (,) delimiter. Example: nicknames=<*name1*>,<*name2*>,<*name3*>... If the nickname includes a comma, escape it using '\' (backslash). Example: if the nickname is "name,1", specify "nicknames=name\,1". If you enter a space character after a comma, the space character is recognized as a part of the nickname specified after the comma.	query	*string* , Each item must match /^[a-zA-Z0-9,\.:@_]([a-zA-Z0-9,\.:@_\-]{0,228})$/

Name	Description	Type	Data type
	Also, if you specify 33 or more names, HTTP status code 400 (Bad Request) is returned.		
hbaName	The initiator WWN for FC connection or iSCSI name for iSCSI connection of the compute node. For this parameter, specify a WWN or iSCSI name as a case-sensitive exact match.	query	*string*
hbaId	The initiator ID of the compute node.	query	*string* (uuid)
vpsId	The ID of the virtual private storage (VPS) that the acquisition-target resource belongs to. To filter out the resources that do not belong to the VPS, specify "system". To filter the resources by the VPS that the resources belong to, specify it in UUID format.	query	*string* , must match /^system$\|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/servers

HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
```

```
Content-Type: application/json
Content-Length: 392
Date: Thu, 20 Oct 2022 05:35:49 GMT

{"data":[{"id":"13ff2e16-fede-4a32-88b1-e891a5c1857e","nickname":"vps_comp",
"osType":"Linux","totalCapacity":0,"usedCapacity":0,"numberOfPaths":0,
"vpsId":"54665c29-784a-4b71-9b61-2894d55697e9","vpsName":"VPS_01"},{"id":"b0883058-
fc49-4064-bbb6-601e732619a9","nickname":"comp_all","osType":"Linux","totalCapacity":0,
"usedCapacity":0,"numberOfPaths":0,"vpsId":"(system)","vpsName":"(system)"}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of summary information of the compute node.

PROPERTIES

data: ***object[]***

ITEMS

serverSummary: object (on page 684)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/servers

Required Role: Storage or VpsStorage

DESCRIPTION

Registers the information of the compute node.

REQUEST BODY

Uses default content-types: application/json

createServerParam: object (on page 593)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
'{ "serverNickname": "ComputeNode2", "osType": "VMware"}' https://192.168.1.100/
ConfigurationManager/simple/v1/objects/servers
```

The following shows an example using a json body.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
@example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/servers
```

example.json is defined as follows:

```
{
    "serverNickname": "ComputeNode2",
    "osType": "VMware"
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 409 (on page 577)
Indicates that an inconsistent or impossible request was issued for the resource specified by URL (for example, creating a resource whose ID is same as the resource already created).

HTTP status code 411 (on page 577)
Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

DELETE /v1/objects/servers/<*id*>

Required Role: Storage or VpsStorage

DESCRIPTION

Deletes the information of the compute node.

REQUEST BODY

Uses default content-types: application/json

deleteServerParam: object (on page 603)

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the compute node.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -X DELETE -u admin:password https://192.168.1.100/ConfigurationManager/simple/v1/
objects/servers/53c96ca6-fdd4-4e03-ab1e-4725e8e4e6d9
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)

Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/servers/<*id*>

Required Role: Security, Storage, Monitor, Service, Resource, VpsSecurity, VpsStorage, or VpsMonitor

DESCRIPTION

Obtains the information of the compute node.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the compute node.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/servers/13ff2e16-fede-4a32-88b1-e891a5c1857e

HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 234
Date: Thu, 20 Oct 2022 05:30:02 GMT

{"numberOfVolumes":0,"paths":[],"id":"13ff2e16-fede-4a32-88b1-e891a5c1857e",
"nickname":"vps_comp","osType":"Linux","totalCapacity":0,"usedCapacity":0,
"numberOfPaths":0,"vpsId":"54665c29-784a-4b71-9b61-2894d55697e9","vpsName":"VPS_01"}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

server: object (on page 683)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

PATCH /v1/objects/servers/<*id*>

Required Role: Storage or VpsStorage

DESCRIPTION

Edits the information of the compute node.

REQUEST BODY

Uses default content-types: application/json

patchServerParam: object (on page 645)

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the compute node.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d '{ "nickname": "ComputeNode2", "osType": "VMware"}' https://192.168.1.100/
ConfigurationManager/simple/v1/objects/servers/84bf2812-4642-4aea-b6ac-27b713df00a4
```

The following shows an example using a json body.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d @example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/servers/
84bf2812-4642-4aea-b6ac-27b713df00a4
```

example.json is defined as follows:

```
{
    "nickname": "ComputeNode2",
    "osType": "VMware"
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/servers/<*id*>/hbas

Required Role: Security, Storage, Monitor, Service, Resource, VpsSecurity, VpsStorage, or VpsMonitor

DESCRIPTION

Obtains a list of initiator information of the compute node.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the compute node.	path	*string* (uuid) required
name	The initiator WWN for FC connection or iSCSI name for iSCSI connection of the compute node. For this parameter, specify a WWN or iSCSI name as a case-sensitive exact match.	query	*string*
vpsId	The ID of the virtual private storage (VPS) that the acquisition-target resource belongs to. To filter out the resources that do not belong to the VPS, specify "system". To filter the resources by the VPS that the resources belong to, specify it in UUID format.	query	*string* , must match /^system $\|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/servers/b0883058-fc49-4064-bbb6-601e732619a9/hbas

HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 233
Date: Thu, 20 Oct 2022 06:45:07 GMT

{"data":[{"id":"644d0858-8092-49da-b18b-6df4bbfcf503","serverId":"b0883058-fc49-4064-
bbb6-601e732619a9","name":"iqn.1994-04.jp.co.hitachi:rsd.sph.t.056e6.001",
"protocol":"iSCSI","portIds":[],"vpsId":"(system)","vpsName":"(system)"}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of initiator information of the compute node.

PROPERTIES

data: *object[]*

ITEMS

hba: object (on page 617)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/servers/<*id*>/hbas

Required Role: Storage or VpsStorage

DESCRIPTION

Registers the initiator information of a compute node.

REQUEST BODY

Uses default content-types: application/json

addHbasServerParam: object (on page 580)

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the compute node.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
'{ "protocol": "iSCSI", "iscsiName": "iqn.1991-05.com.microsoft:win-
g7mqgirmfls12016"}' https://192.168.1.100/ConfigurationManager/simple/v1/objects/
servers/84bf2812-4642-4aea-b6ac-27b713df00a4/hbas
```

The following shows an example using a json body.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
@example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/servers/
84bf2812-4642-4aea-b6ac-27b713df00a4/hbas
```

example.json is defined as follows:

```
{
    "protocol": "iSCSI",
    "iscsiName": "iqn.1991-05.com.microsoft:win-g7mqgirmfls12016"
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 409 (on page 577)
Indicates that an inconsistent or impossible request was issued for the resource specified by URL (for example, creating a resource whose ID is same as the resource already created).

HTTP status code 411 (on page 577)
Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

DELETE /v1/objects/servers/<*id*>/hbas/<*hbaId*>

Required Role: Storage or VpsStorage

DESCRIPTION

Deletes the initiator information of a compute node.

REQUEST BODY

Uses default content-types: application/json

deleteHbasServerParam: object (on page 603)

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the compute node.	path	*string* (uuid) required
hbaId	The initiator ID of the compute node.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -X DELETE -u admin:password https://192.168.1.100/ConfigurationManager/simple/v1/
objects/servers/53c96ca6-fdd4-4e03-ab1e-4725e8e4e6d9/hbas/667fb6e2-1cf7-448e-8e5d-
4309756f7863
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/servers/<*id*>/hbas/<*hbaId*>

Required Role: Security, Storage, Monitor, Service, Resource, VpsSecurity, VpsStorage, or VpsMonitor

DESCRIPTION

Obtains the initiator information of a compute node.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the compute node.	path	*string* (uuid) required
hbaId	The initiator ID of the compute node.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/servers/b0883058-fc49-4064-bbb6-601e732619a9/hbas/644d0858-8092-49da-b18b-
```

```
6df4bbfcf503
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 222
Date: Thu, 20 Oct 2022 06:48:05 GMT

{"id":"644d0858-8092-49da-b18b-6df4bbfcf503","serverId":"b0883058-fc49-4064-bbb6-
601e732619a9","name":"iqn.1994-04.jp.co.hitachi:rsd.sph.t.056e6.001",
"protocol":"iSCSI","portIds":[],"vpsId":"(system)","vpsName":"(system)"}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

hba: object (on page 617)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/servers/<*id*>/paths

Required Role: Security, Storage, Monitor, Service, Resource, VpsSecurity, VpsStorage, or VpsMonitor

DESCRIPTION

Obtains the path information list of the compute node. Path information to be obtained can be filtered using the query parameter.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the compute node.	path	*string* (uuid) required
hbaName	The initiator WWN for FC connection or iSCSI name for iSCSI connection of the compute node. For this parameter, specify a WWN or iSCSI name as a case-sensitive exact match.	query	*string*
hbaId	The initiator ID of the compute node.	query	*string* (uuid)
portId	The ID of the compute port.	query	*string* (uuid)
portName	The WWN for FC connection or iSCSI name for iSCSI connection of the compute port. For this parameter, specify a WWN or iSCSI name as a case-sensitive exact match.	query	*string*
vpsId	The ID of the virtual private storage (VPS) that the acquisition-target resource belongs to. To filter out the resources that do not belong to the VPS, specify "system". To filter the resources by the VPS that the resources belong to, specify it in UUID format.	query	*string* , must match /^system $\|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/servers/b0883058-fc49-4064-bbb6-601e732619a9/paths

HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 1258
Date: Thu, 20 Oct 2022 06:51:04 GMT

{"data":[{"id":"644d0858-8092-49da-b18b-6df4bbfcf503,29eb1271-aaad-4589-b561-
960bc0dba4a4","serverId":"b0883058-fc49-4064-bbb6-601e732619a9","hbaName":"iqn.1994-
04.jp.co.hitachi:rsd.sph.t.056e6.001","hbaId":"644d0858-8092-49da-b18b-6df4bbfcf503",
"portId":"29eb1271-aaad-4589-b561-960bc0dba4a4","portName":"iqn.1994-
04.jp.co.hitachi:rsd.sph.t.056e6.002","portNickname":"002-iSCSI-002",
"vpsId":"(system)","vpsName":"(system)"},{"id":"644d0858-8092-49da-b18b-6df4bbfcf503,
6cb1a798-fcc4-4ff1-833c-e175061adc57","serverId":"b0883058-fc49-4064-bbb6-
601e732619a9","hbaName":"iqn.1994-04.jp.co.hitachi:rsd.sph.t.056e6.001",
"hbaId":"644d0858-8092-49da-b18b-6df4bbfcf503","portId":"6cb1a798-fcc4-4ff1-833c-
e175061adc57","portName":"iqn.1994-04.jp.co.hitachi:rsd.sph.t.056e6.001",
"portNickname":"001-iSCSI-001","vpsId":"(system)","vpsName":"(system)"},
{"id":"644d0858-8092-49da-b18b-6df4bbfcf503,1b9434ea-7e34-4b44-a12b-4ccbf8924b9f",
"serverId":"b0883058-fc49-4064-bbb6-601e732619a9","hbaName":"iqn.1994-
04.jp.co.hitachi:rsd.sph.t.056e6.001","hbaId":"644d0858-8092-49da-b18b-6df4bbfcf503",
"portId":"1b9434ea-7e34-4b44-a12b-4ccbf8924b9f","portName":"iqn.1994-
04.jp.co.hitachi:rsd.sph.t.056e6.000","portNickname":"000-iSCSI-000",
"vpsId":"(system)","vpsName":"(system)"}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of path information.

PROPERTIES

data: ***object[]***

ITEMS

path: object (on page 653)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/servers/<*id*>/paths

Required Role: Storage, VpsStorage

DESCRIPTION

Adds path information to a compute node.

REQUEST BODY

Uses default content-types: application/json

addPathServerParam: object (on page 580)

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the compute node.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
'{ "hbaId": "b4298375-cb1d-4176-b6c4-c1d43b7085a3", "portId": "113987dd-3108-4327-
9e8e-c1f4ff3a766c"}' https://192.168.1.100/ConfigurationManager/simple/v1/objects/
servers/c0554e3d-615a-48fa-bdbd-7d4fc3573ebe/paths
```

The following shows an example using a json body.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
@example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/servers/
c0554e3d-615a-48fa-bdbd-7d4fc3573ebe/paths
```

example.json is defined as follows:

```
{
    "hbaId": "b4298375-cb1d-4176-b6c4-c1d43b7085a3",
    "portId": "113987dd-3108-4327-9e8e-c1f4ff3a766c"
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)

Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 409 (on page 577)

Indicates that an inconsistent or impossible request was issued for the resource specified by URL (for example, creating a resource whose ID is same as the resource already created).

HTTP status code 411 (on page 577)

Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

DELETE /v1/objects/servers/<*id*>/paths/<*hbaId*>,<*portId*>

Required Role: Storage or VpsStorage

DESCRIPTION

Deletes the compute node path information.

REQUEST BODY

Uses default content-types: application/json

deletePathServerParam: object (on page 603)

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the compute node.	path	*string* (uuid) required
hbaId	The initiator ID of the compute node.	path	*string* (uuid) required
portId	The compute port ID of the target operation.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -X DELETE -u admin:password https://192.168.1.100/ConfigurationManager/simple/v1/
objects/servers/53c96ca6-fdd4-4e03-ab1e-4725e8e4e6d9/paths/667fb6e2-1cf7-448e-8e5d-
4309756f7863,343db11a-21d0-45eb-a66c-c31ab1ac6863
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)

Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/servers/<*id*>/paths/<*hbaId*>,<*portId*>

Required Role: Security, Storage, Monitor, Service, Resource, VpsSecurity, VpsStorage, or VpsMonitor

DESCRIPTION

Obtains the compute node path information. This API is used to verify that the path is deleted correctly after the deletion. The response data is the same as the one specified by the request parameter.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the compute node.	path	*string* (uuid) required
hbaId	The initiator ID of the compute node.	path	*string* (uuid) required
portId	The compute port ID of the target operation.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/servers/b0883058-fc49-4064-bbb6-601e732619a9/paths/644d0858-8092-49da-b18b-
6df4bbfcf503,6cb1a798-fcc4-4ff1-833c-e175061adc57

HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 415
Date: Thu, 20 Oct 2022 06:55:59 GMT

{"id":"644d0858-8092-49da-b18b-6df4bbfcf503,6cb1a798-fcc4-4ff1-833c-e175061adc57",
"serverId":"b0883058-fc49-4064-bbb6-601e732619a9","hbaName":"iqn.1994-
04.jp.co.hitachi:rsd.sph.t.056e6.001","hbaId":"644d0858-8092-49da-b18b-6df4bbfcf503",
"portId":"6cb1a798-fcc4-4ff1-833c-e175061adc57","portName":"iqn.1994-
04.jp.co.hitachi:rsd.sph.t.056e6.001","portNickname":"001-iSCSI-001",
"vpsId":"(system)","vpsName":"(system)"}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

path: object (on page 653)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/volume-server-connections

Required Role: Security, Storage, Monitor, Service, Resource, VpsSecurity, VpsStorage, or VpsMonitor

DESCRIPTION

Obtains a list of connection information between the volume and the compute node.

REQUEST PARAMETERS

Name	Description	Type	Data type
volumeId	Volume ID. serverId or volumeId must be specified.	query	*string* (uuid)
serverId	The ID of the compute node. serverId or volumeId must be specified.	query	*string* (uuid)
vpsId	The ID of the virtual private storage (VPS) that the acquisition-target resource belongs to. To filter out the resources that do not belong to the VPS, specify "system". To filter the resources by the VPS that the resources belong to, specify it in UUID format.	query	*string* , must match /^system $\|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/volume-server-connections?volumeId=35d16c28-b033-4c00-9956-67e8221493d3
```

```
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 241
Date: Thu, 20 Oct 2022 06:32:38 GMT

{"data":[{"id":"35d16c28-b033-4c00-9956-67e8221493d3,b0883058-fc49-4064-bbb6-
601e732619a9","serverId":"b0883058-fc49-4064-bbb6-601e732619a9","volumeId":"35d16c28-
b033-4c00-9956-67e8221493d3","lun":1,"vpsId":"(system)","vpsName":"(system)"}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of connection information between the volume and the compute node.

PROPERTIES

data: ***object[]***

ITEMS

volumePath: object (on page 740)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/volume-server-connections

Required Role: Storage or VpsStorage

DESCRIPTION

Sets a path between the specified volume and compute node.

REQUEST BODY

Uses default content-types: application/json

assignToServerVolumeParam: object (on page 582)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
'{ "volumeId": "006c2fa9-26e2-402b-bd28-9193e3a6a3d6", "serverId": "002c1b34-82e8-
4e52-9221-a206c831a367"}' https://192.168.1.100/ConfigurationManager/simple/v1/
objects/volume-server-connections
```

The following shows an example using a json body.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
@example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/volume-
server-connections
```

example.json is defined as follows:

```
{
    "volumeId": "006c2fa9-26e2-402b-bd28-9193e3a6a3d6",
    "serverId": "002c1b34-82e8-4e52-9221-a206c831a367"
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 409 (on page 577)

Indicates that an inconsistent or impossible request was issued for the resource specified by URL (for example, creating a resource whose ID is same as the resource already created).

HTTP status code 411 (on page 577)

Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)

Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/volume-server-connections/actions/release/invoke

Required Role: Storage or VpsStorage

DESCRIPTION

Releases multiple connections between the volumes and compute nodes. Make sure that there are no I/Os from a compute node before running this API.

REQUEST BODY

Uses default content-types: application/json

releaseFromServerVolumeParam: object (on page 671)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
'{ "volumeIds": ["006c2fa9-26e2-402b-bd28-9193e3a6a3d6","006c2fa9-26e2-402b-bd28-
9193e3a6a3d7"], "serverIds":["002c1b34-82e8-4e52-9221-a206c831a367","002c1b34-82e8-
4e52-9221-a206c831a368"]}' https://192.168.1.100/ConfigurationManager/simple/v1/
objects/volume-server-connections/actions/release/invoke
```

The following shows an example using a json body.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
@example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/volume-
server-connections/actions/release/invoke
```

example.json is defined as follows:

```
{
    "volumeIds":
        [
            "006c2fa9-26e2-402b-bd28-9193e3a6a3d6",
            "006c2fa9-26e2-402b-bd28-9193e3a6a3d7"
        ],
    "serverIds":
        [
            "002c1b34-82e8-4e52-9221-a206c831a367",
            "002c1b34-82e8-4e52-9221-a206c831a368"
        ]
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 409 (on page 577)
Indicates that an inconsistent or impossible request was issued for the resource specified by URL (for example, creating a resource whose ID is same as the resource already created).

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

DELETE /v1/objects/volume-server-connections/ <*volumeId*>,<*serverId*>

Required Role: Storage or VpsStorage

DESCRIPTION

Releases the connection between the volume and the compute node. Make sure that there are no I/Os from the compute node, and then execute the API.

REQUEST BODY

Uses default content-types: application/json

releaseFromSingleServerSingleVolumeParam: object (on page 672)

REQUEST PARAMETERS

Name	Description	Type	Data type
volumeId	Volume ID.	path	*string* (uuid) required
serverId	The ID of the compute node.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -X DELETE -u admin:password https://192.168.1.100/ConfigurationManager/simple/v1/
objects/volume-server-connections/fa476fba-7633-4b51-94f2-e50973782a5c,53c96ca6-fdd4-
4e03-ab1e-4725e8e4e6d9
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/volume-server-connections/<*volumeId*>,<*serverId*>

Required Role: Security, Storage, Monitor, Service, Resource, VpsSecurity, VpsStorage, or VpsMonitor

DESCRIPTION

Obtains the connection information between the volume and the compute node.

REQUEST PARAMETERS

Name	Description	Type	Data type
volumeId	Volume ID.	path	*string* (uuid) required
serverId	The ID of the compute node.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/volume-server-connections/35d16c28-b033-4c00-9956-67e8221493d3,b0883058-fc49-
4064-bbb6-601e732619a9

HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 230
Date: Thu, 20 Oct 2022 06:36:26 GMT

{"id":"35d16c28-b033-4c00-9956-67e8221493d3,b0883058-fc49-4064-bbb6-601e732619a9",
"serverId":"b0883058-fc49-4064-bbb6-601e732619a9","volumeId":"35d16c28-b033-4c00-9956-
67e8221493d3","lun":1,"vpsId":"(system)","vpsName":"(system)"}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

volumePath: object (on page 740)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

Chapter 6: Drive management

GET /v1/objects/drives

Required Role: Security, Storage, Monitor, Service, or Resource

DESCRIPTION

Obtains a list of drive information.

REQUEST PARAMETERS

Name	Description	Type	Data type
statusSummary	Drive status summary.	query	*string* , x ∈ { "Normal" , "Warning" , "Error" }
status	The drive status.	query	*string* , x ∈ { "Offline" , "Normal" , "TemporaryBlockage" , "Blockage" }
storageNodeId	Storage node ID.	query	*string* (uuid)
locatorLedStatus	On/off state of the locator LED. ▪ Off: The LED is off. ▪ On: The LED is on.	query	*string* , x ∈ { "Off" , "On" }

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.19.210/ConfigurationManager/
simple/v1/objects/drives
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
```

```
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 2278
Date: Fri, 30 Jul 2021 07:56:01 GMT

{"data":[{"id":"126f360e-c79e-4e75-8f7c-7d91bfd2f0b8","wwid":"naa.50000f0b00c060c0",
"statusSummary":"Normal","status":"Normal","typeCode":"VO001600JWZJQ   ",
"serialNumber":"      S5KWNE0NC01548","storageNodeId":"9d36c162-e379-4c85-bcc2-
ccf98fe774a6","deviceFileName":"sdb","vendorName":"HP      ",
"firmwareRevision":"HPD2","locatorLedStatus":"Off","driveType":"SSD",
"driveCapacity":195},{"id":"31d5de2c-4c24-48e6-acb2-afe73c7ae950",
"wwid":"naa.50000f0b00c05990","statusSummary":"Normal","status":"Normal",
"typeCode":"VO001600JWZJQ   ","serialNumber":"      S5KWNE0NC01433",
"storageNodeId":"44f1d113-405e-448f-ad77-fd5554971c36","deviceFileName":"sdb",
"vendorName":"HP      ","firmwareRevision":"HPD2","locatorLedStatus":"Off",
"driveType":"SSD","driveCapacity":195},{"id":"e426d5dc-b86d-4e61-9643-891e002ed1dd",
"wwid":"naa.50000f0b00c05ea0","statusSummary":"Normal","status":"Normal",
"typeCode":"VO001600JWZJQ   ","serialNumber":"      S5KWNE0NC01514",
"storageNodeId":"9f29d742-9a75-4cef-9539-35b3e76c8a20","deviceFileName":"sdb",
"vendorName":"HP      ","firmwareRevision":"HPD2","locatorLedStatus":"Off",
"driveType":"SSD","driveCapacity":195},{"id":"e63956f7-237d-43d7-a59b-2ba54d24094d",
"wwid":"naa.50000f0b00c050d0","statusSummary":"Normal","status":"Normal",
"typeCode":"VO001600JWZJQ   ","serialNumber":"      S5KWNE0NC01293",
"storageNodeId":"a5b19d19-ca5d-4d19-988d-d625850c85de","deviceFileName":"sdb",
"vendorName":"HP      ","firmwareRevision":"HPD2","locatorLedStatus":"Off",
"driveType":"SSD","driveCapacity":195},{"id":"f2d6ea88-70b6-4905-8dea-f85255c2237c",
"wwid":"naa.50000f0b00c04ff0","statusSummary":"Normal","status":"Normal",
"typeCode":"VO001600JWZJQ   ","serialNumber":"      S5KWNE0NC01279",
"storageNodeId":"8fdbc008-ec6c-42b3-b930-146c94c5cef9","deviceFileName":"sdb",
"vendorName":"HP      ","firmwareRevision":"HPD2","locatorLedStatus":"Off",
"driveType":"SSD","driveCapacity":195},{"id":"fa480d4c-c906-4a38-b534-b8286010c6e9",
"wwid":"naa.50000f0b00c054d0","statusSummary":"Normal","status":"Normal",
"typeCode":"VO001600JWZJQ   ","serialNumber":"      S5KWNE0NC01357",
"storageNodeId":"dd5ed12b-cf39-47da-9a30-5ba53404050e","deviceFileName":"sdb",
"vendorName":"HP      ","firmwareRevision":"HPD2","locatorLedStatus":"Off",
"driveType":"SSD","driveCapacity":195}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of drive information.

PROPERTIES

data: ***object[]***
ITEMS

drive: object (on page 605)

Note:

The output results of typeCode, serialNumber, and vendorName might include spaces depending on the drive type.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/drives/<*id*>

Required Role: Security, Storage, Monitor, Service, or Resource

DESCRIPTION

Obtains the drive information.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Drive ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.19.210/ConfigurationManager/
simple/v1/objects/drives/fa480d4c-c906-4a38-b534-b8286010c6e9
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 377
Date: Fri, 30 Jul 2021 07:56:41 GMT
```

```
{"id":"fa480d4c-c906-4a38-b534-b8286010c6e9","wwid":"naa.50000f0b00c054d0",
"statusSummary":"Normal","status":"Normal","typeCode":"VO001600JWZJQ   ",
"serialNumber":"      S5KWNE0NC01357","storageNodeId":"dd5ed12b-cf39-47da-9a30-
5ba53404050e","deviceFileName":"sdb","vendorName":"HP      ",
"firmwareRevision":"HPD2","locatorLedStatus":"Off","driveType":"SSD",
"driveCapacity":195}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

drive: object (on page 605)

Note:

The output results of typeCode, serialNumber, and vendorName might include spaces depending on the drive type.

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/drives/<*id*>/actions/control-locator-led/invoke

Required Role: Storage, Service

DESCRIPTION

Controls on and off of the locator LED of a drive.

REQUEST BODY

Uses default content-types: application/json

driveLocatorLedParam: object (on page 607)

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Drive ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl  -H "content-type: application/json" -d '{ "operationType": "TurnOn" }' -u
admin:password -i -X POST https://192.168.1.100/ConfigurationManager/simple/v1/
objects/drives/9e17e3ed-806e-4aa0-a53b-c7c00667596d/actions/control-locator-led/invoke
```

The following shows an example using a json body.

```
curl  -H "content-type: application/json" -d @example.json -u admin:password -i -X
POST https://192.168.1.100/ConfigurationManager/simple/v1/objects/drives/9e17e3ed-
806e-4aa0-a53b-c7c00667596d/actions/control-locator-led/invoke
```

example.json is defined as follows:

```
{
    "operationType": "TurnOn"
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)
Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/drives/<*id*>/actions/remove/invoke

Required Role: Storage

DESCRIPTION

Removes a drive. When processing of this API completes, the status of the drive becomes Offline.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Drive ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Length: 0" https://192.168.1.100/
ConfigurationManager/simple/v1/objects/drives/12345678-1234-5678-90ab-1234567890ab/
actions/remove/invoke
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)
Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

Chapter 7: Event log management

GET /v1/objects/event-log-setting

Required Role: Security

DESCRIPTION

Obtains the event log settings.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/event-log-setting
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 481
Date: Fri, 30 Jul 2021 07:56:14 GMT

{"syslogForwardingSetting":{"locationName":"","syslogServers":[{"index":1,
"isEnabled":false,"serverName":"","port":514,"transportProtocol":"UDP"},{"index":2,
"isEnabled":false,"serverName":"","port":514,"transportProtocol":"UDP"}]},
"emailReportSetting":{"smtpSettings":[{"index":1,"isEnabled":false,
"smtpServerName":"","port":587,"connectionEncryptionType":"STARTTLS",
"isSmtpAuthEnabled":true,"smtpAuthAccount":"","fromAddress":"","toAddress1":"",
"toAddress2":"","toAddress3":""}]}}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

eventLogSetting: object (on page 613)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

PATCH /v1/objects/event-log-setting

Required Role: Security

DESCRIPTION

Edits the event log settings.

REQUEST BODY

Uses default content-types: application/json

patchEventLogSettingParam: object (on page 642)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

For syslog transfer:

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d '{"syslogForwardingSetting":{"locationName":"HSP3F-Rack0320","syslogServers":
[{"index":1,"isEnabled":true,"serverName":"syslog_server1.hitachi.com","port":514,
"transportProtocol":"UDP"},{"index":2,"isEnabled":true,
"serverName":"syslog_server2.hitachi.com","port":514,"transportProtocol":"UDP"}]}}'
https://192.168.1.100/ConfigurationManager/simple/v1/objects/event-log-setting
```

The following shows an example using a json body.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d @example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/event-
log-setting
```

example.json is defined as follows:

```
{
    "syslogForwardingSetting":
        {
            "locationName":"HSP3F-Rack0320",
            "syslogServers":
                [
                    {
                        "index":1,
                        "isEnabled":true,
                        "serverName":"syslog_server1.hitachi.com",
                        "port":514,
                        "transportProtocol":"UDP"
                    },
                    {
                        "index":2,
                        "isEnabled":true,
                        "serverName":"syslog_server2.hitachi.com",
                        "port":514,
                        "transportProtocol":"UDP"
                    }
            ]
        }
}
```

For email notification:

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d '{"emailReportSetting":{"smtpSettings":[{"index":1,"isEnabled":true,
"smtpServerName":"smtp_server.hitachi.com","smtpAuthAccount":"admin",
"smtpAuthPassword":"password","fromAddress":"from_mail@hitachi.com",
"toAddress1":"to_mail1@hitachi.com","toAddress2":"to_mail2@hitachi.com",
"toAddress3":"to_mail3@hitachi.com"}]}}' https://192.168.1.100/ConfigurationManager/
simple/v1/objects/event-log-setting
```

The following shows an example using a json body.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d @example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/event-
log-setting
```

example.json is defined as follows:

```
{
    "emailReportSetting":
        {
            "smtpSettings":
                [
                    {
                        "index":1,
                        "isEnabled":true,
                        "smtpServerName":"smtp_server.hitachi.com",
                        "smtpAuthAccount":"admin",
                        "smtpAuthPassword":"password",
                        "fromAddress":"from_mail@hitachi.com",
                        "toAddress1":"to_mail1@hitachi.com",
                        "toAddress2":"to_mail2@hitachi.com",
                        "toAddress3":"to_mail3@hitachi.com"
                    }
                ]
    }
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/event-logs

Required Role: Security, Storage, Monitor, Service, or Resource

DESCRIPTION

Obtains a list of event logs. Logs are listed in descending order of the date and time when events were detected.

The following describes data to be output according to the specification of the startTime and endTime query parameters:

When both startTime and endTime are omitted

- Assuming that the following are specified, outputs event logs in reverse chronological order of the date and time when the event was detected:
  - The time when the REST API server received the request was specified as endTime.
  - 1970-01-01T00:00:01Z was specified as startTime.

When both startTime and endTime are specified

- Logs within the specified period are listed in reverse chronological order of the date and time when the event was detected.

When only startTime is specified

- Assuming that the time when the REST API server received the request is specified as endTime, logs within the specified period are listed in reverse chronological order of the data and time when the event was detected.

When only endTime is specified

- Assuming that 1970-01-01T00:00:01Z was specified as startTime, outputs event logs within the specified period in reverse chronological order of the date and time when the event was detected.

When the range specified by startTime and endTime is invalid

- HTTP status code 400 (Bad Request) is returned when the following date and time are specified:
  - A date and time later than endTime is specified as startTime, including when endTime is not specified.
  - A date and time which is earlier than 1970-01-01T00:00:01Z is specified for startTime or endTime.

REQUEST PARAMETERS

Name	Description	Type	Data type
severity	The severity of the event. Only events with the specified severity are extracted. If you specify both parameters at the same time, HTTP status code 400 (Bad Request) is returned.	query	*string* , x ∈ { "Info" , "Warning" , "Error" , "Critical" }

Name	Description	Type	Data type
severityGe	The severity of the event. Events with the specified severity or higher are extracted. Info < Warning < Error < Critical < Unknown Unknown events are output regardless of the selected severity level. Either severity or severityGe can be specified. If you specify both parameters at the same time, HTTP status code 400 (Bad Request) is returned.	query	*string* ,x ∈ { "Info" , "Warning" , "Error" , "Critical" }
startTime	The start date and time of the event log you obtain. The data of the specified time is included in the obtained event log.	query	*string* (date-time)
endTime	The end data and time of the event log you obtain. The data of the specified time is included in the obtained event log.	query	*string* (date-time)
maxEvents	Maximum number of event log records that can be obtained.	query	*integer* , { x ∈ ℤ | 1 ≤ x ≤ 1000 } (default: "1000")

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/event-logs
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
```

```
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 382
Date: Mon, 08 Nov 2021 07:56:25 GMT

{"data":[{"id":"29c35cd4-41ab-4a26-a4b6-c13fc8f86d11","time":"2021-11-08T06:43:08Z",
"timeInMicroseconds":1636353788052656,"category":"Service","eventName":"Storage
cluster start","messageId":"KARS08100-I","severity":"Info","message":"The storage
cluster started.","solution":"","nodeLocation":"449dcdeb-755f-4254-b27e-ed6f3bd31770",
"eventType":"Informational","severityLevel":"1"}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

Event log list.

PROPERTIES

data: ***object[]***

ITEMS

eventLog: object (on page 612)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/event-logs/<*id*>

Required Role: Security, Storage, Monitor, Service, or Resource

DESCRIPTION

Obtains the event log.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Event log ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/event-logs/356cfd72-f9e0-4a1e-b57a-81ba66a15a43
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 371
Date: Fri, 30 Jul 2021 07:56:53 GMT

{"id":"356cfd72-f9e0-4a1e-b57a-81ba66a15a43","time":"2021-07-30T04:56:15Z",
"timeInMicroseconds":1627620975039655,"category":"Service","eventName":"Storage
cluster start","messageId":"KARS08100-I","severity":"Info","message":"The storage
cluster started.","solution":"","nodeLocation":"a5b19d19-ca5d-4d19-988d-d625850c85de",
"eventType":"Informational","severityLevel":"1"}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

eventLog: object (on page 612)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/smtp-server-root-certificates/<*targetServer*>/actions/import/invoke

Required Role: Security

DESCRIPTION

Imports a root certificate (to be used in communication with the SMTP server) to a storage cluster.

PEM- and DER-format certificate files are supported. If TLS communication with the SMTP server is enabled in the storage cluster setting for the SMTP server specified in "targetServer", a root certificate is applied immediately.

You can run this API only for the cluster master node (primary). If you call this API for a node other than the cluster master node (primary), HTTP status code 412 (Precondition Failed) is returned.

REQUEST BODY

multipart/form-data

REQUEST PARAMETERS

Name	Description	Type	Data type
rootCertificate	Root certificate file used in communication with the SMTP server.	formData	*file* required
targetServer	Identification number of the target SMTP server.	path	*integer* (int32) , { x ∈ ℤ | 1 ≤ x ≤ 1 } required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: multipart/form-data" -H "Expect:
" -F rootCertificate=@/home/user/maintenance/users/masaaki_sato/ca-crt_1MB.pem
https://192.168.1.100/ConfigurationManager/simple/v1/objects/smtp-server-root-
certificates/1/actions/import/invoke
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)
Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 412 (on page 577)
Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/smtp-server-root-certificates/<*targetServer*>/ download

Required Role: Security

DESCRIPTION

Obtains a root certificate (to be used in communication with the SMTP server) which is imported to a storage cluster.

A root certificate is output as a DER file.

If you call this API when no root certificate is imported, HTTP status code 404 (Not Found) is returned.

You can run this API only for the cluster master node (primary). If you call this API for a node other than the cluster master node (primary), HTTP status code 412 (Precondition Failed) is returned.

If you have not imported the root certificate for the SMTP server, you may see message ID: KARS15553-E. In that case, import the root certificate of the SMTP server.

REQUEST PARAMETERS

Name	Description	Type	Data type
targetServer	Identification number of the target SMTP server.	path	*integer* (int32) , { x ∈ ℤ | 1 ≤ x ≤ 1 } required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -J -O -X GET -u admin:password https://192.168.1.100/ConfigurationManager/
simple/v1/objects/smtp-server-root-certificates/1/download
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   832  100   832    0     0   1139      0 --:--:-- --:--:-- --:--:--  1139
curl: Saved to filename 'download-smtp-server-root-certificate.crt'
```

RESPONSES

Uses default content-types: application/x-x509-ca-cert, application/json

200 OK

Root certificate file corresponding to the server specified as "targetServer".

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 412 (on page 577)
Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

Chapter 8: Failure analysis

POST /v1/objects/dump-file/actions/create-file/invoke

Required Role: Service

DESCRIPTION

Requests creation of a dump log file of the storage node that issued the API.

Specify the IP address or the host name of the control port for the storage node for which you want to create a dump log file as the connection destination.

To create dump log files for all the storage nodes in the storage cluster, issue the REST API for all the storage nodes in the storage cluster, and then create dump log files.

It is recommended that you use ticket authentication for this API. Another authentication scheme can be used if there is no ticket, but authentication might not succeed depending on the failure condition of the storage node.

REQUEST BODY

Uses default content-types: application/json

createDumpFileParam: object (on page 593)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for the storage node at the IP address of the control port 192.168.1.100 with the user name as "admin" and the password as "password".

```
curl -X POST -H 'Authorization: Ticket <ticket>' -H "X-Ticket-Parameter: `echo -n
admin:password | base64`"  -H 'Expect:' -H 'Content-Type: application/json' -d
'{ "label": "test", "mode": "Base" }'  https://192.168.1.100/ConfigurationManager/
simple/v1/objects/dump-file/actions/create-file/invoke
```

The following shows an example using a json body.

```
curl -X POST -H 'Authorization: Ticket <ticket>' -H "X-Ticket-Parameter: `echo -n
admin:password | base64`"  -H 'Expect:' -H 'Content-Type: application/json' -d
@example.json  https://192.168.1.100/ConfigurationManager/simple/v1/objects/dump-file/
actions/create-file/invoke
```

example.json is defined as follows:

```
{
    "label": "test",
    "mode": "Base"
}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

dumpStatus: object (on page 608)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)

Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication
- ticket authentication

GET /v1/objects/dump-file/download

Required Role: Service

DESCRIPTION

Downloads the latest dump log file of the storage node to which the API was issued.

The dump log file with the latest creation completion time is targeted.

Specify the IP address or the host name of the control port for the storage node for which you want to download a dump log file as the connection destination.

If you specified indexOfSplitFiles, the target dump log files are downloaded separately.

To download dump log files for all the storage nodes in the storage cluster, issue the REST API for all the storage nodes in the storage cluster, and then download dump log files.

It is recommended that you use ticket authentication for this API. Another authentication scheme can be used if there is no ticket, but authentication might not succeed depending on the failure condition of the storage node.

If download does not complete within an hour, verify the size of the file whose name starts with "hsds_log_" or "Error_hsds_log_", which is created in the current directory. If the file size has not changed, verify that no error has occurred in the network connection to the destination storage node.

Verify that the free disk space on the controller node is not less than the dump log file size by running GET /v1/objects/dump-status, and then execute the API.

Note:

(Virtual machine) The API cannot be executed on a maintenance node.

For how to verify whether an error has occurred in a network connection, contact customer support.

If you have not created a dump log file for the storage node, you may see message ID: KARS15553-E. In that case, perform a request to create a dump log file for the storage node.

REQUEST PARAMETERS

Name	Description	Type	Data type
indexOfSplitFiles	Index of the split dump log files. Specify this parameter in the range from 1 to the number that the dump log files are split. You can view the number that the dump log files are split in numberOfSplitFiles by using the following API: GET/v1/objects/dump-statuses	query	*integer* , { x ∈ ℤ | 1 ≤ x ≤ 369 }

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for the storage node at the IP address of the control port 192.168.1.100 with the user name as "admin" and the password as "password".

```
curl -J -O -X GET -H 'Authorization: Ticket <ticket>' -H "X-Ticket-Parameter: `echo -
n admin:password | base64`" https://192.168.1.100/ConfigurationManager/simple/v1/
objects/dump-file/download
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 85.5M    0 85.5M    0     0  4728k      0 --:--:--  0:00:18 --:--:-- 1233k
curl: Saved to filename 'hsds_log_20211110_164657_test_maint-test-SN03_Base.tar.gz'
```

RESPONSES

Uses default content-types: application/octet-stream, application/json

200 OK

Dump log file.

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 412 (on page 577)

Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication
- ticket authentication

GET /v1/objects/dump-files/<*fileName*>/download

Required Role: Service

DESCRIPTION

Downloads a dump log file of the storage node to which the API was issued by specifying the name of the dump log file.

The dump log file specified for fileName is targeted.

Specify the IP address or the host name of the control port for the storage node for which you want to download a dump log file as the connection destination.

If you specified indexOfSplitFiles, the target dump log files are downloaded separately.

To download dump log files for all the storage nodes in the storage cluster, issue the REST API for all the storage nodes in the storage cluster, and then download dump log files.

It is recommended that you use ticket authentication for this API. Another authentication scheme can be used if there is no ticket, but authentication might not succeed depending on the failure condition of the storage node.

If download does not complete within an hour, verify the size of the file whose name starts with "hsds_log_" or "Error_hsds_log_", which is created in the current directory. If the file size has not changed, verify that no error has occurred in the network connection to the destination storage node.

Verify that the free disk space on the controller node is not less than the dump log file size by running GET /v1/objects/dump-statuses, and then execute the API. The API cannot be executed on a maintenance node.

Note:

(Virtual machine) The API cannot be executed on a maintenance node.

For how to verify whether an error has occurred in a network connection, contact customer support.

If you have not created a dump log file for the storage node, you may see message ID: KARS15553-E. In that case, perform a request to create a dump log file for the storage node.

REQUEST PARAMETERS

Name	Description	Type	Data type
fileName	Name of the target dump log file.	path	*string* (1 to 255 chars) , must match /^[a-zA-Z0-9!#$%&\'\-\.@\^_`\{\}~]{1,255}$/ required
indexOfSplitFiles	Index of the split dump log files. Specify this parameter in the range from 1 to the number that the dump log files are split. You can view the number that the dump log files are split in numberOfSplitFiles by using the following API: GET/v1/objects/dump-statuses	query	*integer* , { x ∈ ℤ | 1 ≤ x ≤ 369 }

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -J -O -X GET -H 'Authorization: Ticket <ticket>' -H "X-Ticket-Parameter: `echo -
n admin:password | base64`" https://192.168.1.100/ConfigurationManager/simple/v1/
objects/dump-files/hsds_log_20211110_164657_test_maint-test-SN03_Base.tar.gz/download
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 85.5M    0 85.5M    0     0  5836k      0 --:--:--  0:00:15 --:--:-- 7948k
curl: Saved to filename 'hsds_log_20211110_164657_test_maint-test-SN03_Base.tar.gz'
```

RESPONSES

Uses default content-types: application/octet-stream, application/json

200 OK

Dump log file specified for fileName.

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 412 (on page 577)

Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication
- ticket authentication

DELETE /v1/objects/dump-files/<*fileName*>

Required Role: Service

DESCRIPTION

Deletes a dump log file of the storage node to which the API was issued.

The dump log file specified for fileName is targeted.

For the connection destination, specify the IP address or the host name of the control port for the storage node for which you want to delete a dump log file.

It is recommended that you use ticket authentication for this API. Another authentication scheme can be used if there is no ticket, but authentication might not succeed depending on the failure condition of the storage node.

REQUEST PARAMETERS

Name	Description	Type	Data type
fileName	Name of the target dump log file.	path	*string* (1 to 255 chars) , must match /^[a-zA-Z0-9!#$%&\'\-\.@\^_`\{\}~]{1,255}$/ required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
 curl -i -X DELETE -H 'Authorization: Ticket <ticket>' -H "X-Ticket-Parameter: `echo -
n admin:password | base64`" https://192.168.1.100/ConfigurationManager/simple/v1/
objects/dump-files/hsds_log_20211110_164657_test_maint-test-SN03_Base.tar.gz
```

RESPONSES

Uses default content-types: application/json

HTTP status code 204 (on page 575)
Indicates that the request was processed correctly, but no response data to be returned exists.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication
- ticket authentication

GET /v1/objects/dump-status

Required Role: Service

DESCRIPTION

Verifies the latest creation status of the dump log file of the storage node to which the API was issued.

The status of the dump log file with the latest creation completion time is returned.

For the connection destination, specify the IP address or the host name of the control port for the storage node for which you want to confirm the creation status of the dump log file.

It is recommended that you use ticket authentication for this API. Another authentication scheme can be used if there is no ticket, but authentication might not succeed depending on the failure condition of the storage node.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for the storage node at the IP address of the control port 192.168.1.100 with the user name as "admin" and the password as "password".

```
curl -i -X GET -H 'Authorization: Ticket <ticket>' -H "X-Ticket-Parameter: `echo -n
admin:password | base64`" https://192.168.1.100/ConfigurationManager/simple/v1/
objects/dump-status
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
```

```
Content-Type: application/json
Content-Length: 265
Date: Wed, 10 Nov 2021 07:54:05 GMT

{"startedTime":"2021-11-10T07:46:56Z","completedTime":"2021-11-10T07:53:02Z",
"label":"test","status":"Created","size":86,"triggerType":"Manual","mode":"Base",
"fileName":"hsds_log_20211110_164657_test_maint-test-SN03_Base.tar.gz",
"numberOfSplitFiles":1,"error":null}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

dumpStatus: object (on page 608)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication
- ticket authentication

GET /v1/objects/dump-statuses

Required Role: Service

DESCRIPTION

Verifies the list of dump log file creation status of the storage node to which the API was issued.

For the connection destination, specify the IP address or the host name of the control port for the storage node for which you want to confirm the creation status of the dump log file.

It is recommended that you use ticket authentication for this API. Another authentication scheme can be used if there is no ticket, but authentication might not succeed depending on the failure condition of the storage node.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for the storage node at the IP address of the control port 192.168.1.100 with the user name as "admin" and the password as "password".

```
curl -i -X GET -H 'Authorization: Ticket <ticket>' -H "X-Ticket-Parameter: `echo -n
admin:password | base64`" https://192.168.1.100/ConfigurationManager/simple/v1/
objects/dump-statuses
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
```

```
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 552
Date: Wed, 10 Nov 2021 07:55:04 GMT

{"data":[{"startedTime":"2021-11-10T07:46:56Z","completedTime":"2021-11-10T07:53:02Z",
"label":"test","status":"Created","size":86,"triggerType":"Manual","mode":"Base",
"fileName":"hsds_log_20211110_164657_test_maint-test-SN03_Base.tar.gz",
"numberOfSplitFiles":1,"error":null},{"startedTime":"2021-11-10T07:09:18Z",
"completedTime":"2021-11-10T07:10:36Z","label":"","status":"Created","size":78,
"triggerType":"Auto","mode":"AutoCollection",
"fileName":"hsds_log_20211110_160919__maint-test-SN03_AutoCollection.tar.gz",
"numberOfSplitFiles":1,"error":null}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

dumpStatusList: object (on page 610)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication
- ticket authentication

POST /v1/objects/tickets

Required Role: Service or Security

DESCRIPTION

Issues an authentication ticket. This authentication ticket is used for authentication, such as a dump file creation API or CLI for which ticket authentication is allowed, when basic authentication and session authentication cannot be used due to a failure.

REQUEST BODY

Uses default content-types: application/json

createTicketParam: object (on page 596)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -d
'{ "maxAgeDays": 365 }' https://192.168.1.100/ConfigurationManager/simple/v1/objects/
tickets
```

The following shows an example using a json body.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -d
@example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/tickets
```

example.json is defined as follows:

```
{
    "maxAgeDays": 365
}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

ticket: object (on page 721)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)

Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)

Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication

POST /v1/objects/tickets/actions/revoke-all/invoke

Required Role: Security

DESCRIPTION

Discards all the authentication tickets issued before this API is run. If the discarded authentication tickets are not reflected in some of the storage nodes due to a failure or other reason, such storage nodes are notified via an event log.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Length: 0" https://192.168.1.100/
ConfigurationManager/simple/v1/objects/tickets/actions/revoke-all/invoke
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)

Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)

Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)

Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

Chapter 9: Login message management

GET /configuration/login-message

Required Role: None

DESCRIPTION

Obtains the message to be displayed in the GUI login window and CLI warning banner.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/
configuration/login-message
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 27
Date: Fri, 30 Jul 2021 07:56:13 GMT

{"message":"login message"}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

loginMessage: object (on page 637)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

PATCH /configuration/login-message

Required Role: Security

DESCRIPTION

Edits the message to be displayed in the GUI login window and CLI warning banner.

REQUEST BODY

Uses default content-types: application/json

patchLoginMessageParam: object (on page 643)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d '{ "message": "Login Message" }' https://192.168.1.100/ConfigurationManager/simple/
configuration/login-message
```

The following shows an example using a json body.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d @example.json https://192.168.1.100/ConfigurationManager/simple/configuration/login-
message
```

example.json is defined as follows:

```
{
   "message": "Login Message"
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

Chapter 10: High resolution monitor

Note: Performance information described in this chapter includes internal I/Os such as volume migration.

GET /v1/objects/detail-performances/control-ports

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains the performance information (monitor information) of the control port specified in query parameter ids.

The performance information about high resolution monitors is internally collected at five second intervals. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned. To verify whether the information is the latest one, compare the timestamp contained in the response data with that obtained last.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.
- Immediately after Virtual Storage Software block is started or a control port is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no performance information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

The high resolution monitor can obtain information only for five seconds until the REST API server receives a request. Basically, only the monitor information of the latest cycle is returned. However, depending on the timing for synchronizing NTP servers, monitor information of the latest cycle and the one immediately prior to the latest cycle might be returned.

REQUEST PARAMETERS

Name	Description	Type	Data type
ids	The ID of the control port. You can specify plural IDs up to 32 with comma (,) delimiter. Example: ids=<*id1*>,<*id2*>,<*id3*>… HTTP status code 400 (Bad Request) is returned when the following conditions are met: ▪ 33 or more IDs are specified. ▪ A character string other than UUID is specified (including when a space character exists before or after a comma, or when the string is an empty string).	query	*string* required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/detail-performances/control-ports?ids=f93b583a-478d-413c-abe9-028d8ae4f67b
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
```

```
Content-Type: application/json
Content-Length: 182
Date: Fri, 30 Jul 2021 08:04:47 GMT

{"data":[{"timestamp":"2021-07-30T08:04:44Z","performanceObjects":[{"id":"f93b583a-
478d-413c-abe9-028d8ae4f67b","receiveTransferRate":0.004067039489746094,
"sendTransferRate":0.0}]}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

controlPortPerformanceListResponse: object (on page 590)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/detail-performances/control-ports/<*id*>

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains the performance information (monitor information) about the specified control port.

The performance information about high resolution monitors is internally collected at five second intervals. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned. To verify whether the information is the latest one, compare the timestamp contained in the response data with that obtained last.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.
- Immediately after Virtual Storage Software block is started or a control port is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no performance information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

The high resolution monitor can obtain information only for five seconds until the REST API server receives a request. Basically, only the monitor information of the latest cycle is returned. However, depending on the timing for synchronizing NTP servers, monitor information of the latest cycle and the one immediately prior to the latest cycle might be returned.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the control port.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/detail-performances/control-ports/f93b583a-478d-413c-abe9-028d8ae4f67b
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 199
Date: Fri, 30 Jul 2021 07:56:48 GMT

{"data":[{"timestamp":"2021-07-30T07:56:46Z","performanceObjects":[{"id":"f93b583a-
478d-413c-abe9-028d8ae4f67b","receiveTransferRate":0.0058917999267578125,
"sendTransferRate":1.33514404296875E-5}]}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

controlPortPerformanceListResponse: object (on page 590)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/detail-performances/drives

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains the performance information (monitor information) of the drive specified in query parameter ids.

The performance information about high resolution monitors is internally collected at five second intervals. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.
- Immediately after Virtual Storage Software block is started or a drive is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no performance information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

The high resolution monitor can obtain information only for five seconds until the REST API server receives a request. Basically, only the monitor information of the latest cycle is returned. However, depending on the timing for synchronizing NTP servers, monitor information of the latest cycle and the one immediately prior to the latest cycle might be returned.

REQUEST PARAMETERS

Name	Description	Type	Data type
ids	Drive ID. You can specify plural IDs up to 32 with comma (,) delimiter. Example: ids=<*id1*>,<*id2*>,<*id3*>… HTTP status code 400 (Bad Request) is returned when the following conditions are met: ▪ 33 or more IDs are specified. ▪ A character string other than UUID is specified (including when a space character exists before or after a comma, or when the string is an empty string).	query	*string* required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/detail-performances/drives?ids=fa480d4c-c906-4a38-b534-b8286010c6e9
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 235
Date: Fri, 30 Jul 2021 08:04:48 GMT

{"data":[{"timestamp":"2021-07-30T08:04:43Z","performanceObjects":[{"id":"fa480d4c-
c906-4a38-b534-b8286010c6e9","readIOPS":7,"writeIOPS":30,"readTransferRate":1.9296875,
"writeTransferRate":5.8224609375,"responseTime":0.0,"usage":0}]}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

drivePerformanceListResponse: object (on page 608)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/detail-performances/drives/<*id*>

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains the performance information (monitor information) about the specified drive.

The performance information about high resolution monitors is internally collected at five second intervals. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.
- Immediately after Virtual Storage Software block is started or a drive is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no performance information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

The high resolution monitor can obtain information only for five seconds until the REST API server receives a request. Basically, only the monitor information of the latest cycle is returned. However, depending on the timing for synchronizing NTP servers, monitor information of the latest cycle and the one immediately prior to the latest cycle might be returned.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Drive ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/detail-performances/drives/fa480d4c-c906-4a38-b534-b8286010c6e9
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 239
Date: Fri, 30 Jul 2021 07:56:48 GMT

{"data":[{"timestamp":"2021-07-30T07:56:45Z","performanceObjects":[{"id":"fa480d4c-
c906-4a38-b534-b8286010c6e9","readIOPS":86,"writeIOPS":477,
"readTransferRate":9.05546875,"writeTransferRate":78.7892578125,"responseTime":0.0,
"usage":0}]}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

drivePerformanceListResponse: object (on page 608)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/detail-performances/internode-ports

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains the performance information (monitor information) of the internode port specified in query parameter ids.

The performance information about high resolution monitors is internally collected at five second intervals. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned. To verify whether the information is the latest one, compare the timestamp contained in the response data with that obtained last.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.
- Immediately after Virtual Storage Software block is started or a internode port is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no performance information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

The high resolution monitor can obtain information only for five seconds until the REST API server receives a request. Basically, only the monitor information of the latest cycle is returned. However, depending on the timing for synchronizing NTP servers, monitor information of the latest cycle and the one immediately prior to the latest cycle might be returned.

REQUEST PARAMETERS

Name	Description	Type	Data type
ids	The ID of the internode port.	query	*string* required

Name	Description	Type	Data type
	You can specify plural IDs up to 32 with comma (,) delimiter. Example: ids=<*id1*>,<*id2*>,<*id3*>… HTTP status code 400 (Bad Request) is returned when the following conditions are met: ▪ 33 or more IDs are specified. ▪ A character string other than UUID is specified (including when a space character exists before or after a comma, or when the string is an empty string).		

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/detail-performances/internode-ports?ids=f9713eb8-5c93-4150-a969-84f7f1ad9582
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 195
Date: Fri, 30 Jul 2021 08:04:48 GMT

{"data":[{"timestamp":"2021-07-30T08:04:48Z","performanceObjects":[{"id":"f9713eb8-
5c93-4150-a969-84f7f1ad9582","receiveTransferRate":2.552302360534668,
"sendTransferRate":0.21334800720214844}]}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

internodePortPerformanceListResponse: object (on page 622)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/detail-performances/internode-ports/<*id*>

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains the performance information (monitor information) of the specified internode port.

The performance information about high resolution monitors is internally collected at five second intervals. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned. To verify whether the information is the latest one, compare the timestamp contained in the response data with that obtained last.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.
- Immediately after Virtual Storage Software block is started or a internode port is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no performance information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

The high resolution monitor can obtain information only for five seconds until the REST API server receives a request. Basically, only the monitor information of the latest cycle is returned. However, depending on the timing for synchronizing NTP servers, monitor information of the latest cycle and the one immediately prior to the latest cycle might be returned.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the internode port.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/detail-performances/internode-ports/f9713eb8-5c93-4150-a969-84f7f1ad9582
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 193
Date: Fri, 30 Jul 2021 07:56:49 GMT

{"data":[{"timestamp":"2021-07-30T07:56:45Z","performanceObjects":[{"id":"f9713eb8-
5c93-4150-a969-84f7f1ad9582","receiveTransferRate":32.31705303192139,
"sendTransferRate":34.74436283111572}]}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

internodePortPerformanceListResponse: object (on page 622)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/detail-performances/ports

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains the performance information (monitor information) of the compute port specified in query parameter ids.

The performance information about high resolution monitors is internally collected at five second intervals. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned. To verify whether the information is the latest one, compare the timestamp contained in the response data with that obtained last.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.
- Immediately after Virtual Storage Software block is started or a compute port is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no performance information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

The high resolution monitor can obtain information only for five seconds until the REST API server receives a request. Basically, only the monitor information of the latest cycle is returned. However, depending on the timing for synchronizing NTP servers, monitor information of the latest cycle and the one immediately prior to the latest cycle might be returned.

REQUEST PARAMETERS

Name	Description	Type	Data type
ids	The ID of the compute port. You can specify plural IDs up to 32 with comma (,) delimiter.	query	*string* required

Name	Description	Type	Data type
	Example: ids=<*id1*>,<*id2*>,<*id3*>… HTTP status code 400 (Bad Request) is returned when the following conditions are met: ▪ 33 or more IDs are specified. ▪ A character string other than UUID is specified (including when a space character exists before or after a comma, or when the string is an empty string).		

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.19.210/ConfigurationManager/
simple/v1/objects/detail-performances/ports/343db11a-21d0-45eb-a66c-c31ab1ac6863
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 240
Date: Fri, 30 Jul 2021 07:56:50 GMT

{"data":[{"timestamp":"2021-07-30T07:56:47Z","performanceObjects":[{"id":"343db11a-
21d0-45eb-a66c-c31ab1ac6863","fc":null,"iscsi":{"target":{"readIOPS":0,"writeIOPS":0,
"readTransferRate":0.0,"writeTransferRate":0.0,"responseTime":0.0}}}]}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

portPerformanceListResponse: object (on page 663)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/detail-performances/ports/<*id*>

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains the performance information (monitor information) about the specified compute port.

The performance information about high resolution monitors is internally collected at five second intervals. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned. To verify whether the information is the latest one, compare the timestamp contained in the response data with that obtained last.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.
- Immediately after Virtual Storage Software block is started or a compute port is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no performance information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

The high resolution monitor can obtain information only for five seconds until the REST API server receives a request. Basically, only the monitor information of the latest cycle is returned. However, depending on the timing for synchronizing NTP servers, monitor information of the latest cycle and the one immediately prior to the latest cycle might be returned.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the compute port.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/detail-performances/ports/343db11a-21d0-45eb-a66c-c31ab1ac6863
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 240
Date: Fri, 30 Jul 2021 07:56:50 GMT

{"data":[{"timestamp":"2021-07-30T07:56:47Z","performanceObjects":[{"id":"343db11a-
21d0-45eb-a66c-c31ab1ac6863","fc":null,"iscsi":{"target":{"readIOPS":0,"writeIOPS":0,
"readTransferRate":0.0,"writeTransferRate":0.0,"responseTime":0.0}}}]}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

portPerformanceListResponse: object (on page 663)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/detail-performances/storage-nodes

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains the performance information (monitor information) of the storage node specified in query parameter ids.

The performance information about high resolution monitors is internally collected at five second intervals. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.
- Immediately after Virtual Storage Software block is started or a storage node is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no performance information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

The high resolution monitor can obtain information only for five seconds until the REST API server receives a request. Basically, only the monitor information of the latest cycle is returned. However, depending on the timing for synchronizing NTP servers, monitor information of the latest cycle and the one immediately prior to the latest cycle might be returned.

REQUEST PARAMETERS

Name	Description	Type	Data type
ids	Storage node ID. You can specify plural IDs up to 32 with comma (,) delimiter. Example: ids=<*id1*>,<*id2*>,<*id3*>…	query	*string* required

Name	Description	Type	Data type
	HTTP status code 400 (Bad Request) is returned when the following conditions are met: ▪ 33 or more IDs are specified. ▪ A character string other than UUID is specified (including when a space character exists before or after a comma, or when the string is an empty string).		

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/detail-performances/storage-nodes?ids=dd5ed12b-cf39-47da-9a30-5ba53404050e
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 1208
Date: Fri, 30 Jul 2021 08:04:50 GMT

{"data":[{"timestamp":"2021-07-30T08:04:48Z","performanceObjects":[{"id":"dd5ed12b-
cf39-47da-9a30-5ba53404050e","volumeReadIOPS":0,"volumeWriteIOPS":0,
"volumeReadTransferRate":0.0,"volumeWriteTransferRate":0.0,"driveReadIOPS":9,
"driveWriteIOPS":38,"driveReadTransferRate":2.4375,"driveWriteTransferRate":7.3546875,
"cpu":[{"index":3,"process":"StorageController","usage":18.2},{"index":4,
"process":"StorageController","usage":17.8},{"index":5,"process":"StorageController",
"usage":18.4},{"index":0,"process":"SystemController","usage":2.80561122244489},
{"index":6,"process":"SystemController","usage":3.225806451612903},{"index":7,
"process":"SystemController","usage":100.0},{"index":8,"process":"SystemController",
"usage":7.551020408163265},{"index":9,"process":"SystemController",
"usage":7.392197125256674},{"index":10,"process":"SystemController",
"usage":6.036217303822937},{"index":11,"process":"SystemController","usage":100.0},
{"index":12,"process":"SystemController","usage":7.377049180327869},{"index":13,
"process":"SystemController","usage":6.813627254509018},{"index":1,
"process":"StorageController","usage":0.0},{"index":2,"process":"StorageController",
"usage":0.0}],"memory":{"usage":16.9169}}]}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

storageNodePerformanceListResponse: object (on page 709)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/detail-performances/storage-nodes/<*id*>

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains the performance information (monitor information) about the specified storage node.

The performance information about high resolution monitors is internally collected at five second intervals. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.
- Immediately after Virtual Storage Software block is started or a storage node is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no performance information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

The high resolution monitor can obtain information only for five seconds until the REST API server receives a request. Basically, only the monitor information of the latest cycle is returned. However, depending on the timing for synchronizing NTP servers, monitor information of the latest cycle and the one immediately prior to the latest cycle might be returned.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Storage node ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/detail-performances/storage-nodes/dd5ed12b-cf39-47da-9a30-5ba53404050e
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 1236
Date: Fri, 30 Jul 2021 07:56:51 GMT

{"data":[{"timestamp":"2021-07-30T07:56:50Z","performanceObjects":[{"id":"dd5ed12b-
cf39-47da-9a30-5ba53404050e","volumeReadIOPS":0,"volumeWriteIOPS":0,
"volumeReadTransferRate":0.0,"volumeWriteTransferRate":0.0,"driveReadIOPS":98,
"driveWriteIOPS":391,"driveReadTransferRate":14.3287109375,
"driveWriteTransferRate":69.5798828125,"cpu":[{"index":3,
"process":"StorageController","usage":18.8},{"index":4,"process":"StorageController",
"usage":19.0},{"index":5,"process":"StorageController","usage":18.8},{"index":0,
"process":"SystemController","usage":2.9469548133595285},{"index":6,
"process":"SystemController","usage":3.292181069958848},{"index":7,
"process":"SystemController","usage":11.354581673306772},{"index":8,
"process":"SystemController","usage":58.116232464929865},{"index":9,
"process":"SystemController","usage":30.208333333333332},{"index":10,
"process":"SystemController","usage":60.6},{"index":11,"process":"SystemController",
"usage":44.88977955911824},{"index":12,"process":"SystemController",
"usage":9.475806451612904},{"index":13,"process":"SystemController",
"usage":32.72357723577235},{"index":1,"process":"StorageController","usage":0.0},
{"index":2,"process":"StorageController","usage":0.0}],"memory":{"usage":16.906}}]}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

storageNodePerformanceListResponse: object (on page 709)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/detail-performances/volumes

Required Role: Storage, Monitor, Resource, VpsStorage, or VpsMonitor

DESCRIPTION

Obtains the performance information (monitor information) of the volume specified by query parameter, ids.

The performance information about high resolution monitors is internally collected at five second intervals. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.
- Immediately after Virtual Storage Software block is started or a volume is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no performance information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

The high resolution monitor can obtain information only for five seconds until the REST API server receives a request. Basically, only the monitor information of the latest cycle is returned. However, depending on the timing for synchronizing NTP servers, monitor information of the latest cycle and the one immediately prior to the latest cycle might be returned.

REQUEST PARAMETERS

Name	Description	Type	Data type
ids	The volume ID. You can specify plural IDs up to 32 with comma (,) delimiter. Example: ids=<*id1*>,<*id2*>,<*id3*>… HTTP status code 400 (Bad Request) is returned when the following conditions are met: ▪ 33 or more IDs are specified. ▪ A character string other than UUID is specified (including when a space character exists before or after a comma, or when the string is an empty string).	query	*string* required
vpsId	The ID of the virtual private storage (VPS) that the acquisition-target resource belongs to. To filter out the resources that do not belong to the VPS, specify "system". To filter the resources by the VPS that the resources belong to, specify it in UUID format.	query	*string* , must match /^system $\|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/detail-performances/volumes?ids=ds=7e4eb895-48cc-4f48-b228-9224af6aa4f6,
35d16c28-b033-4c00-9956-67e8221493d3

HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 558
Date: Thu, 20 Oct 2022 05:49:36 GMT

{"data":[{"timestamp":"2022-10-20T05:49:34Z","performanceObjects":[{"id":"35d16c28-
b033-4c00-9956-67e8221493d3","vpsId":"00000000-0000-0000-0000-000000000000",
"readIOPS":0,"writeIOPS":0,"readTransferRate":0.0,"writeTransferRate":0.0,
"readResponseTime":0.0,"writeResponseTime":0.0}]},{"timestamp":"2022-10-20T05:49:35Z",
"performanceObjects":[{"id":"7e4eb895-48cc-4f48-b228-9224af6aa4f6","vpsId":"54665c29-
784a-4b71-9b61-2894d55697e9","readIOPS":0,"writeIOPS":0,"readTransferRate":0.0,
"writeTransferRate":0.0,"readResponseTime":0.0,"writeResponseTime":0.0}]}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

volumePerformanceListResponse: object (on page 742)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/detail-performances/volumes/<*id*>

Required Role: Storage, Monitor, Resource, VpsStorage, or VpsMonitor

DESCRIPTION

Obtains the performance information (monitor information) about the specified volume.

The performance information about high resolution monitors is internally collected at five second intervals. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.
- Immediately after Virtual Storage Software block is started or a volume is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no performance information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

The high resolution monitor can obtain information only for five seconds until the REST API server receives a request. Basically, only the monitor information of the latest cycle is returned. However, depending on the timing for synchronizing NTP servers, monitor information of the latest cycle and the one immediately prior to the latest cycle might be returned.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Volume ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/detail-performances/volumes/35d16c28-b033-4c00-9956-67e8221493d3

HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
```

```
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 284
Date: Thu, 20 Oct 2022 05:53:16 GMT

{"data":[{"timestamp":"2022-10-20T05:53:11Z","performanceObjects":[{"id":"35d16c28-
b033-4c00-9956-67e8221493d3","vpsId":"00000000-0000-0000-0000-000000000000",
"readIOPS":0,"writeIOPS":0,"readTransferRate":0.0,"writeTransferRate":0.0,
"readResponseTime":0.0,"writeResponseTime":0.0}]}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

volumePerformanceListResponse: object (on page 742)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

Chapter 11: License management

GET /v1/objects/license-setting

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains the license settings.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/license-setting
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 75
Date: Fri, 30 Jul 2021 07:56:26 GMT

{"warningThresholdSetting":{"remainingDays":30,"totalPoolCapacityRate":80}}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

licenseSetting: object (on page 636)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

PATCH /v1/objects/license-setting

Required Role: Storage

DESCRIPTION

Edits the license settings.

REQUEST BODY

Uses default content-types: application/json

patchLicenseSettingParam: object (on page 643)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d '{"warningThresholdSetting": {"remainingDays":-1,"totalPoolCapacityRate":-1}}'
https://192.168.1.100/ConfigurationManager/simple/v1/objects/license-setting
```

The following shows an example using a json body.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d @example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/license-
setting
```

example.json is defined as follows:

```
{
    "warningThresholdSetting":
        {
            "remainingDays":-1,
            "totalPoolCapacityRate":-1
        }
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/licenses

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains a list of license information.

REQUEST PARAMETERS

Name	Description	Type	Data type
programProductName	The name of the program product.	query	*string* (1 to 63 chars)
status	Status of the license.	query	*string* , x ∈ { "Active", "Warning", "Overwritten", "GracePeriod", "Invalid" }
statusSummary	License status summary.	query	*string* , x ∈ { "Normal", "Warning", "Error" }

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/licenses
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
```

```
Content-Type: application/json
Content-Length: 254
Date: Fri, 30 Jul 2021 07:56:27 GMT

{"data":[{"id":"222c8201-805f-453d-99ac-0b21b8a66bd6","programProductName":"Hitachi
SDS Core Base","status":"Active","statusSummary":"Normal","cause":null,
"keyType":"Trial","permittedCapacityInTiB":null,"totalPoolCapacityInGiB":null,
"remainingDays":60}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of license information.

PROPERTIES

data:*object[]*

ITEMS

license: object (on page 635)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/licenses

Required Role: Storage

DESCRIPTION

Registers a license.

REQUEST BODY

Uses default content-types: application/json

installLicenseParam: object (on page 619)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
'{"keyCode":"1LGB7VTDBH0J7QAQ3EEQM3O1VZLYBO1ER4HU7KXAN0DQ3GT6JW9NZCP1FS5IVAQ4BPF7DOO53
TN"}' https://192.168.1.100/ConfigurationManager/simple/v1/objects/licenses
```

The following shows an example using a json body.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
@example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/licenses
```

example.json is defined as follows:

```
{

"keyCode":"1LGB7VTDBH0J7QAQ3EEQM3O1VZLYBO1ER4HU7KXAN0DQ3GT6JW9NZCP1FS5IVAQ4BPF7DOO53TN
"
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 409 (on page 577)
Indicates that an inconsistent or impossible request was issued for the resource specified by URL (for example, creating a resource whose ID is same as the resource already created).

HTTP status code 411 (on page 577)
Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

DELETE /v1/objects/licenses/<*id*>

Required Role: Storage

DESCRIPTION

Deletes a license.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the license.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -X DELETE -u admin:password https://192.168.1.100/ConfigurationManager/simple/v1/
objects/licenses/222c8201-805f-453d-99ac-0b21b8a66bd6
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/licenses/<*id*>

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains the license information.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the license.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/licenses/222c8201-805f-453d-99ac-0b21b8a66bd6
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
```

```
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 243
Date: Fri, 30 Jul 2021 07:56:58 GMT

{"id":"222c8201-805f-453d-99ac-0b21b8a66bd6","programProductName":"Hitachi SDS Core
Base","status":"Active","statusSummary":"Normal","cause":null,"keyType":"Trial",
"permittedCapacityInTiB":null,"totalPoolCapacityInGiB":null,"remainingDays":60}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

license: object (on page 635)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

Chapter 12: Low resolution monitor

Note: Performance information described in this chapter includes internal I/Os such as volume migration.

GET /v1/objects/performances/control-ports

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains a list of control port performance information (monitor information). Information about control ports existing at the time of API execution is obtained.

Performance information is collected internally at one minute intervals. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned. To verify whether the information is the latest one, compare the timestamp contained in the response data with that obtained last.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.
- Immediately after Virtual Storage Software block is started or a control port is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no performance information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

If the number of control ports is large, in order to obtain all the information, it is necessary to obtain it separately in plural times. If some control ports are not obtained, hasNext in the response data shows true, and the token used to obtain the rest of the control ports is output as enumerateContext. You can obtain the next volume by executing this API again with enumerateContext specified as the query parameter.

Also, by setting a count parameter, you can determine the number of control ports you obtain. If lengthening the obtainment period (startTime/endTime), fewer number of the control ports might return than the value set in the count parameter. If you want to obtain a large number of the control port information at once, set (startTime/endTime) briefly during the obtainment period.

totalCount in the response data is the total number of control ports whose performance records were existing when the API was executed. The number might be changed with each API operation using enumerateContext. Similarly, as the information that is returned is the update at the time of API execution, you might obtain a record of a different timestamp value in a series of API operations using enumerateContext.

The following describes data to be output according to the specification of the startTime and endTime query parameters:

When both startTime and endTime are omitted

- It is assumed that the time when the REST API server received the request was specified as endTime.
- It is assumed that the time which is one minute earlier than the time when the REST API server received the request was specified as startTime. However, if the endTime is 1970-01-01T00:01:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When both startTime and endTime are specified

- The monitored information within the specified time period is returned.
- Due to addition or deletion of control ports, or no collection of information, the number of control ports at a certain point of time might be smaller than the number output for count.

When only startTime is specified

- It is assumed that the time when the REST API server received the request was specified as endTime.

When only endTime is specified

- It is assumed that the time which is one hour earlier than endTime was specified as startTime. However, if endTime is 1970-01-01T01:00:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.
- Due to addition or deletion of control ports, or no collection of information, the number of control ports at a certain point of time might be smaller than the number output for count.

When the range specified by startTime and endTime is invalid

- HTTP status code 400 (Bad Request) is returned when any of the following conditions is satisfied:
  - A date and time later than endTime is specified as startTime, including when endTime is not specified.
  - A date and time which is earlier than 1970-01-01T00:00:01Z is specified for startTime or endTime.
  - When all the following conditions are satisfied:
    - Specify true for startTimeExcluding, or false for endTimeIncluding.
    - The same date and time is specified for startTime and endTime, including the case when startTime is not specified and 1970-01-01T00:00:01Z is specified for endTime.

REQUEST PARAMETERS

Name	Description	Type	Data type
startTime	The start date and time of the monitor information to be obtained. If you specify startTimeExcluding, you can specify whether to include or exclude the specified time. By default, the specified time is included.	query	*string* (date-time)
startTimeExcluding	Determines whether to include information of the specified startTime. ▪ false (default): The information is included. ▪ true: The information is not included.	query	*boolean* (default: "false")
endTime	The end date and time of the monitor information to be obtained.	query	*string* (date-time)

Name	Description	Type	Data type
	If you specify endTimeIncluding, you can specify whether to include or exclude the specified time. By default, the specified time is excluded. If it is omitted, it is the request receipt time.		
endTimeIncluding	Specify whether to include the specified endTime. ▪ false (default): The specified time is not included. ▪ true: The specified time is included.	query	*boolean* (default: "false")
enumerateContext	Token that helps obtain the next batch records. Do not specify this parameter for initial information extraction.	query	*string* (uuid)
count	The number of control port records to be obtained.	query	*integer* , { x ∈ ℤ | 1 ≤ x ≤ 32768 } (default: "32768")

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/performances/control-ports
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 1204
Date: Fri, 30 Jul 2021 07:56:02 GMT
```

```
{"data":[{"timestamp":"2021-07-30T07:55:26Z","performanceObjects":[{"id":"f93b583a-
478d-413c-abe9-028d8ae4f67b","receiveTransferRate":0.004591456326571377,
"sendTransferRate":1.4565207741477273E-6}]},{"timestamp":"2021-07-30T07:55:27Z",
"performanceObjects":[{"id":"8f6d178b-fb20-497c-9825-c50f43113c5e",
"receiveTransferRate":0.004469590108902728,"sendTransferRate":2.7203169025358607E-
6}]},{"timestamp":"2021-07-30T07:55:28Z","performanceObjects":[{"id":"58c87902-5c16-
43aa-99f1-753fbb383f9f","receiveTransferRate":0.004469027284715996,
"sendTransferRate":2.7203169025358607E-6}]},{"timestamp":"2021-07-30T07:55:30Z",
"performanceObjects":[{"id":"4f4158c0-25dd-4fad-a0cb-245e1ac39cb4",
"receiveTransferRate":0.004559183120727539,"sendTransferRate":1.430511474609375E-6}]},
{"timestamp":"2021-07-30T07:55:31Z","performanceObjects":[{"id":"e2f1fea5-1abf-4288-
aa0c-84e0893332f5","receiveTransferRate":0.0045453071594238285,
"sendTransferRate":1.430511474609375E-6}]},{"timestamp":"2021-07-30T07:55:34Z",
"performanceObjects":[{"id":"755e2e4c-de8a-48cb-8a37-c7cbafc92686",
"receiveTransferRate":0.0071882088979085285,
"sendTransferRate":0.004561154047648112}]}],"totalCount":6,"hasNext":false,
"enumerateContext":null}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of control port performance information (monitor information) at the specified time and the information about performanceObjects.

PROPERTIES

data: *object[]*

A list of control port performance information (monitor information) at the specified time.

ITEMS

controlPortPerformanceListResponseData: object (on page 590)

totalCount: *integer*

The total number of control ports whose performance information exists within the specified period.

hasNext: *boolean*

Indicates if some control ports remain to be listed in performanceObjects.

enumerateContext: *string* (uuid)nullable

A token used to obtain the next value of performanceObjects for the applicable information. Null is output if the information is the last one.

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/performances/control-ports/<*id*>

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains the performance information (monitor information) about the specified control port.

Performance information is collected internally at one minute intervals. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned. To verify whether the information is the latest one, compare the timestamp contained in the response data with that obtained last.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.
- Immediately after Virtual Storage Software block is started or a control port is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no performance information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

The following describes data to be output according to the specification of the startTime and endTime query parameters:

When both startTime and endTime are omitted

- It is assumed that the time when the REST API server received the request was specified as endTime.
- It is assumed that the time which is one minute earlier than the time when the REST API server received the request was specified as startTime. However, if the endTime is 1970-01-01T00:01:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When both startTime and endTime are specified

- The monitored information within the specified time period is returned.

When only startTime is specified

- It is assumed that the time when the REST API server received the request was specified as endTime.

When only endTime is specified

- It is assumed that the time which is one hour earlier than endTime was specified as startTime. However, if endTime is 1970-01-01T01:00:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When the range specified by startTime and endTime is invalid

- HTTP status code 400 (Bad Request) is returned when any of the following conditions is satisfied:
  - A date and time later than endTime is specified as startTime, including when endTime is not specified.
  - A date and time which is earlier than 1970-01-01T00:00:01Z is specified for startTime or endTime.
  - When all the following conditions are satisfied:
    - Specify true for startTimeExcluding, or false for endTimeIncluding.
    - The same date and time is specified for startTime and endTime, including the case when startTime is not specified and 1970-01-01T00:00:01Z is specified for endTime.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the control port.	path	*string* (uuid) required
startTime	The start date and time of the monitor information to be obtained. If you specify startTimeExcluding, you can specify whether to include or exclude the specified time. By default, the specified time is included.	query	*string* (date-time)

Name	Description	Type	Data type
startTimeExcluding	Determines whether to include information of the specified startTime. ▪ false (default): The information is included. ▪ true: The information is not included.	query	*boolean* (default: "false")
endTime	The end date and time of the monitor information to be obtained. If you specify endTimeIncluding, you can specify whether to include or exclude the specified time. By default, the specified time is excluded. If it is omitted, it is the request receipt time.	query	*string* (date-time)
endTimeIncluding	Specify whether to include the specified endTime. ▪ false (default): The specified time is not included. ▪ true: The specified time is included.	query	*boolean* (default: "false")

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/performances/control-ports/f93b583a-478d-413c-abe9-028d8ae4f67b
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
```

```
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 199
Date: Fri, 30 Jul 2021 07:56:42 GMT

{"data":[{"timestamp":"2021-07-30T07:56:26Z","performanceObjects":[{"id":"f93b583a-
478d-413c-abe9-028d8ae4f67b","receiveTransferRate":0.003943777084350586,
"sendTransferRate":2.765655517578125E-6}]}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

controlPortPerformanceListResponse: object (on page 590)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/performances/drives

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains a list of drive performance information (monitor information). Information about drives existing at the time of API execution is obtained.

Performance information is collected internally at one minute intervals. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.
- Immediately after Virtual Storage Software block is started or a drive is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no performance information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

If the number of drives is large, in order to obtain all the information, it is necessary to obtain it separately in plural times. If some drives are not obtained, hasNext in the response data shows true, and the token used to obtain the rest of the drives is output as enumerateContext. You can obtain the next volume by executing this API again with enumerateContext specified as the query parameter.

Also, by setting a count parameter, you can determine the number of drives you obtain. If lengthening the obtainment period (startTime/endTime), fewer number of the drives might return than the value set in the count parameter. If you want to obtain a large number of the drive information at once, set (startTime/endTime) briefly during the obtainment period.

totalCount in the response data is the total number of drives whose performance records were existing when the API was executed. The number might be changed with each API operation using enumerateContext. Similarly, as the information that is returned is the update at the time of API execution, you might obtain a record of a different timestamp value in a series of API operations using enumerateContext.

The following describes data to be output according to the specification of the startTime and endTime query parameters:

When both startTime and endTime are omitted

- It is assumed that the time when the REST API server received the request was specified as endTime.
- It is assumed that the time which is one minute earlier than the time when the REST API server received the request was specified as startTime. However, if the endTime is 1970-01-01T00:01:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When both startTime and endTime are specified

- The monitored information within the specified time period is returned.
- Due to addition or deletion of drives, or no collection of information, the number of drives at a certain point of time might be smaller than the number output for count.

When only startTime is specified

- It is assumed that the time when the REST API server received the request was specified as endTime.

When only endTime is specified

- It is assumed that the time which is one hour earlier than endTime was specified as startTime. However, if endTime is 1970-01-01T01:00:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

- Due to addition or deletion of drives, or no collection of information, the number of drives at a certain point of time might be smaller than the number output for count.

When the range specified by startTime and endTime is invalid

- HTTP status code 400 (Bad Request) is returned when any of the following conditions is satisfied:
  - A date and time later than endTime is specified as startTime, including when endTime is not specified.
  - A date and time which is earlier than 1970-01-01T00:00:01Z is specified for startTime or endTime.
  - When all the following conditions are satisfied:
    - Specify true for startTimeExcluding, or false for endTimeIncluding.
    - The same date and time is specified for startTime and endTime, including the case when startTime is not specified and 1970-01-01T00:00:01Z is specified for endTime.

REQUEST PARAMETERS

Name	Description	Type	Data type
startTime	The start date and time of the monitor information to be obtained. If you specify startTimeExcluding, you can specify whether to include or exclude the specified time. By default, the specified time is included.	query	*string* (date-time)

Name	Description	Type	Data type
startTimeExcluding	Determines whether to include information of the specified startTime. ▪ false (default): The information is included. ▪ true: The information is not included.	query	*boolean* (default: "false")
endTime	The end date and time of the monitor information to be obtained. If you specify endTimeIncluding, you can specify whether to include or exclude the specified time. By default, the specified time is excluded. If it is omitted, it is the request receipt time.	query	*string* (date-time)
endTimeIncluding	Specify whether to include the specified endTime. ▪ false (default): The specified time is not included. ▪ true: The specified time is included.	query	*boolean* (default: "false")
enumerateContext	Token that helps obtain the next batch records. Do not specify this parameter for initial information extraction.	query	*string* (uuid)
count	The number of drive records to be obtained.	query	*integer* , { x ∈ ℤ | 1 ≤ x ≤ 32768 } (default: "32768")

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/performances/drives
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 1603
Date: Fri, 30 Jul 2021 07:56:03 GMT

{"data":[{"timestamp":"2021-07-30T07:55:26Z","performanceObjects":[{"id":"e426d5dc-
b86d-4e61-9643-891e002ed1dd","readIOPS":90,"writeIOPS":595,
"readTransferRate":12.189104352678571,"writeTransferRate":98.70378766741071,
"responseTime":0.00020800291204076858,"usage":0}]},{"timestamp":"2021-07-
30T07:55:27Z","performanceObjects":[{"id":"126f360e-c79e-4e75-8f7c-7d91bfd2f0b8",
"readIOPS":107,"writeIOPS":515,"readTransferRate":18.76904296875,
"writeTransferRate":100.03098958333334,"responseTime":0.00021371517110570888,
"usage":0}]},{"timestamp":"2021-07-30T07:55:28Z","performanceObjects":
[{"id":"31d5de2c-4c24-48e6-acb2-afe73c7ae950","readIOPS":111,"writeIOPS":528,
"readTransferRate":11.852197265625,"writeTransferRate":99.02229817708333,
"responseTime":0.0,"usage":0}]},{"timestamp":"2021-07-30T07:55:30Z",
"performanceObjects":[{"id":"fa480d4c-c906-4a38-b534-b8286010c6e9","readIOPS":101,
"writeIOPS":638,"readTransferRate":10.953245069159836,
"writeTransferRate":104.53008132684427,"responseTime":8.860142648296638e-05,
"usage":0}]},{"timestamp":"2021-07-30T07:55:31Z","performanceObjects":
[{"id":"f2d6ea88-70b6-4905-8dea-f85255c2237c","readIOPS":128,"writeIOPS":560,
"readTransferRate":19.243343098958334,"writeTransferRate":102.51724446614584,
"responseTime":0.001452643811737362,"usage":0}]},{"timestamp":"2021-07-30T07:55:34Z",
"performanceObjects":[{"id":"e63956f7-237d-43d7-a59b-2ba54d24094d","readIOPS":116,
"writeIOPS":410,"readTransferRate":16.966540727459016,
"writeTransferRate":79.21639664446721,"responseTime":0.0016195340725052947,
"usage":0}]}],"totalCount":6,"hasNext":false,"enumerateContext":null}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of drive performance information (monitor information) at the specified time and the information about performanceObjects.

PROPERTIES

data: *object[]*

A list of drive performance information (monitor information) at the specified time.

ITEMS

drivePerformanceListResponseData: object (on page 608)

totalCount: *integer*

The total number of drives whose performance information exists within the specified period.

hasNext: *boolean*

Indicates if some drives remain to be listed in performanceObjects.

enumerateContext: *string* (uuid)nullable

A token used to obtain the next value of performanceObjects for the applicable information. Null is output if the information is the last one.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/performances/drives/<*id*>

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains the performance information (monitor information) about the specified drive.

Performance information is collected internally at one minute intervals. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.
- Immediately after Virtual Storage Software block is started or a drive is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no performance information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

The following describes data to be output according to the specification of the startTime and endTime query parameters:

When both startTime and endTime are omitted

- It is assumed that the time when the REST API server received the request was specified as endTime.
- It is assumed that the time which is one minute earlier than the time when the REST API server received the request was specified as startTime. However, if the endTime is 1970-01-01T00:01:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When both startTime and endTime are specified

- The monitored information within the specified time period is returned.

When only startTime is specified

- It is assumed that the time when the REST API server received the request was specified as endTime.

When only endTime is specified

- It is assumed that the time which is one hour earlier than endTime was specified as startTime. However, if endTime is 1970-01-01T01:00:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When the range specified by startTime and endTime is invalid

- HTTP status code 400 (Bad Request) is returned when any of the following conditions is satisfied:
  - A date and time later than endTime is specified as startTime, including when endTime is not specified.
  - A date and time which is earlier than 1970-01-01T00:00:01Z is specified for startTime or endTime.
  - When all the following conditions are satisfied:
    - Specify true for startTimeExcluding, or false for endTimeIncluding.
    - The same date and time is specified for startTime and endTime, including the case when startTime is not specified and 1970-01-01T00:00:01Z is specified for endTime.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Drive ID.	path	*string* (uuid) required
startTime	The start date and time of the monitor information to be obtained. If you specify startTimeExcluding, you can specify whether to include or exclude the specified time. By default, the specified time is included.	query	*string* (date-time)

Name	Description	Type	Data type
startTimeExcluding	Determines whether to include information of the specified startTime. ▪ false (default): The information is included. ▪ true: The information is not included.	query	*boolean* (default: "false")
endTime	The end date and time of the monitor information to be obtained. If you specify endTimeIncluding, you can specify whether to include or exclude the specified time. By default, the specified time is excluded. If it is omitted, it is the request receipt time.	query	*string* (date-time)
endTimeIncluding	Specify whether to include the specified endTime. ▪ false (default): The specified time is not included. ▪ true: The specified time is included.	query	*boolean* (default: "false")

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/performances/drives/fa480d4c-c906-4a38-b534-b8286010c6e9
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
```

```
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 270
Date: Fri, 30 Jul 2021 07:56:43 GMT

{"data":[{"timestamp":"2021-07-30T07:56:30Z","performanceObjects":[{"id":"fa480d4c-
c906-4a38-b534-b8286010c6e9","readIOPS":102,"writeIOPS":656,
"readTransferRate":11.203800455729167,"writeTransferRate":107.43893229166666,
"responseTime":5.264658784302543E-4,"usage":0}]}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

drivePerformanceListResponse: object (on page 608)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/performances/internode-ports

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains a list of internode port performance information (monitor information). Information about internode ports existing at the time of API execution is obtained.

Performance information is collected internally at one minute intervals. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned. To verify whether the information is the latest one, compare the timestamp contained in the response data with that obtained last.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.
- Immediately after Virtual Storage Software block is started or a internode port is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no performance information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.
- Some information might not be obtained due to high load, including spikes.

If the number of internode ports is large, in order to obtain all the information, it is necessary to obtain it separately in plural times. If some internode ports are not obtained, hasNext in the response data shows true, and the token used to obtain the rest of the internode ports is output as enumerateContext. You can obtain the next volume by executing this API again with enumerateContext specified as the query parameter.

Also, by setting a count parameter, you can determine the number of internode ports you obtain. If lengthening the obtainment period (startTime/endTime), fewer number of the internode ports might return than the value set in the count parameter. If you want to obtain a large number of the internode port information at once, set (startTime/endTime) briefly during the obtainment period.

totalCount in the response data is the total number of internode ports whose performance records were existing when the API was executed. The number might be changed with each API operation using enumerateContext. Similarly, as the information that is returned is the update at the time of API execution, you might obtain a record of a different timestamp value in a series of API operations using enumerateContext.

The following describes data to be output according to the specification of the startTime and endTime query parameters:

When both startTime and endTime are omitted

- It is assumed that the time when the REST API server received the request was specified as endTime.
- It is assumed that the time which is one minute earlier than the time when the REST API server received the request was specified as startTime. However, if the endTime is 1970-01-01T00:01:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When both startTime and endTime are specified

- The monitored information within the specified time period is returned.
- Due to addition or deletion of internode ports, or no collection of information, the number of internode ports at a certain point of time might be smaller than the number output for count.

When only startTime is specified

- It is assumed that the time when the REST API server received the request was specified as endTime.

When only endTime is specified

- It is assumed that the time which is one hour earlier than endTime was specified as startTime. However, if endTime is 1970-01-01T01:00:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

- Due to addition or deletion of internode ports, or no collection of information, the number of internode ports at a certain point of time might be smaller than the number output for count.

When the range specified by startTime and endTime is invalid

- HTTP status code 400 (Bad Request) is returned when any of the following conditions is satisfied:
  - A date and time later than endTime is specified as startTime, including when endTime is not specified.
  - A date and time which is earlier than 1970-01-01T00:00:01Z is specified for startTime or endTime.
  - When all the following conditions are satisfied:
    - Specify true for startTimeExcluding, or false for endTimeIncluding.
    - The same date and time is specified for startTime and endTime, including the case when startTime is not specified and 1970-01-01T00:00:01Z is specified for endTime.

REQUEST PARAMETERS

Name	Description	Type	Data type
startTime	The start date and time of the monitor information to be obtained. If you specify startTimeExcluding, you can specify whether to include or exclude the specified time. By default, the specified time is included.	query	*string* (date-time)

Name	Description	Type	Data type
startTimeExcluding	Determines whether to include information of the specified startTime. ▪ false (default): The information is included. ▪ true: The information is not included.	query	*boolean* (default: "false")
endTime	The end date and time of the monitor information to be obtained. If you specify endTimeIncluding, you can specify whether to include or exclude the specified time. By default, the specified time is excluded. If it is omitted, it is the request receipt time.	query	*string* (date-time)
endTimeIncluding	Specify whether to include the specified endTime. ▪ false (default): The specified time is not included. ▪ true: The specified time is included.	query	*boolean* (default: "false")
enumerateContext	Token that helps obtain the next batch records. Do not specify this parameter for initial information extraction.	query	*string* (uuid)
count	The number of internode port records to be obtained.	query	*integer* , { x ∈ ℤ | 1 ≤ x ≤ 32768 } (default: "32768")

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/performances/internode-ports
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 1169
Date: Fri, 30 Jul 2021 07:56:04 GMT

{"data":[{"timestamp":"2021-07-30T07:55:26Z","performanceObjects":[{"id":"15b61129-
3a7a-48c0-b0c4-202ac29008ab","receiveTransferRate":28.348705603859642,
"sendTransferRate":42.630756568908694}]},{"timestamp":"2021-07-30T07:55:27Z",
"performanceObjects":[{"id":"6d9ea5ce-90bd-410e-971f-dc309fde7bdf",
"receiveTransferRate":37.975524558395634,"sendTransferRate":28.98244251188685}]},
{"timestamp":"2021-07-30T07:55:28Z","performanceObjects":[{"id":"3e748241-711c-49a8-
8678-1eefbd69745a","receiveTransferRate":23.233640139220192,
"sendTransferRate":38.45245092423236}]},{"timestamp":"2021-07-30T07:55:30Z",
"performanceObjects":[{"id":"f9713eb8-5c93-4150-a969-84f7f1ad9582",
"receiveTransferRate":36.95031840006511,"sendTransferRate":39.698774576187134}]},
{"timestamp":"2021-07-30T07:55:31Z","performanceObjects":[{"id":"3662d25d-4f0d-4727-
87e3-2823499c86c5","receiveTransferRate":34.33325111071269,
"sendTransferRate":30.25053777694702}]},{"timestamp":"2021-07-30T07:55:34Z",
"performanceObjects":[{"id":"385e0257-8759-4bfc-9414-48005a1e58cc",
"receiveTransferRate":38.46594311396281,"sendTransferRate":19.566099643707275}]}],
"totalCount":6,"hasNext":false,"enumerateContext":null}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION
A list of internode port performance information (monitor information) at the specified time and the information about performanceObjects.

PROPERTIES

data: *object[]*

A list of internode port performance information (monitor information) at the specified time.

ITEMS

internodePortPerformanceListResponseData: object (on page 622)

totalCount: *integer*

The total number of internode ports whose performance information exists within the specified period.

hasNext: *boolean*

Indicates if some internode ports remain to be listed in performanceObjects.

enumerateContext: *string* (uuid)nullable

A token used to obtain the next value of performanceObjects for the applicable information. Null is output if the information is the last one.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/performances/internode-ports/<*id*>

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains the performance information (monitor information) about the specified internode port.

Performance information is collected internally at one minute intervals. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned. To verify whether the information is the latest one, compare the timestamp contained in the response data with that obtained last.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.
- Immediately after Virtual Storage Software block is started or a internode port is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no performance information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

The following describes data to be output according to the specification of the startTime and endTime query parameters:

When both startTime and endTime are omitted

- It is assumed that the time when the REST API server received the request was specified as endTime.
- It is assumed that the time which is one minute earlier than the time when the REST API server received the request was specified as startTime. However, if the endTime is 1970-01-01T00:01:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When both startTime and endTime are specified

- The monitored information within the specified time period is returned.

When only startTime is specified

- It is assumed that the time when the REST API server received the request was specified as endTime.

When only endTime is specified

- It is assumed that the time which is one hour earlier than endTime was specified as startTime. However, if endTime is 1970-01-01T01:00:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When the range specified by startTime and endTime is invalid

- HTTP status code 400 (Bad Request) is returned when any of the following conditions is satisfied:
  - A date and time later than endTime is specified as startTime, including when endTime is not specified.
  - A date and time which is earlier than 1970-01-01T00:00:01Z is specified for startTime or endTime.
  - When all the following conditions are satisfied:
    - Specify true for startTimeExcluding, or false for endTimeIncluding.
    - The same date and time is specified for startTime and endTime, including the case when startTime is not specified and 1970-01-01T00:00:01Z is specified for endTime.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the internode port.	path	*string* (uuid) required
startTime	The start date and time of the monitor information to be obtained. If you specify startTimeExcluding, you can specify whether to include or exclude the specified time. By default, the specified time is included.	query	*string* (date-time)

Name	Description	Type	Data type
startTimeExcluding	Determines whether to include information of the specified startTime. ▪ false (default): The information is included. ▪ true: The information is not included.	query	*boolean* (default: "false")
endTime	The end date and time of the monitor information to be obtained. If you specify endTimeIncluding, you can specify whether to include or exclude the specified time. By default, the specified time is excluded. If it is omitted, it is the request receipt time.	query	*string* (date-time)
endTimeIncluding	Specify whether to include the specified endTime. ▪ false (default): The specified time is not included. ▪ true: The specified time is included.	query	*boolean* (default: "false")

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/performances/internode-ports/f9713eb8-5c93-4150-a969-84f7f1ad9582
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
```

```
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 194
Date: Fri, 30 Jul 2021 07:56:43 GMT

{"data":[{"timestamp":"2021-07-30T07:56:30Z","performanceObjects":[{"id":"f9713eb8-
5c93-4150-a969-84f7f1ad9582","receiveTransferRate":27.859156764921593,
"sendTransferRate":44.99646279069244}]}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

internodePortPerformanceListResponse: object (on page 622)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/performances/pool-capacities

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains a list of storage pool capacity information (monitor information). Information about storage pools existing at the time of API execution is obtained.

The capacity information is internally collected on a one-minute cycle. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.
- Immediately after Virtual Storage Software block is started or a storage pool is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no capacity information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

If the number of storage pools is large, in order to obtain all the information, it is necessary to obtain it separately in plural times. If some storage pools are not obtained, hasNext in the response data shows true, and the token used to obtain the rest of the pools is output as enumerateContext. You can obtain the next volume by executing this API again with enumerateContext specified as the query parameter.

Also, by setting a count parameter, you can determine the number of storage pools you obtain. If lengthening the obtainment period (startTime/endTime), fewer number of the storage pools might return than the value set in the count parameter. If you want to obtain a large number of the storage pool information at once, set (startTime/endTime) briefly during the obtainment period.

totalCount in the response data is the total number of storage pools whose capacity records were existing when the API was executed. The number might be changed with each API operation using enumerateContext. Similarly, as the information that is returned is the update at the time of API execution, you might obtain a record of a different timestamp value in a series of API operations using enumerateContext.

The following describes data to be output according to the specification of the startTime and endTime query parameters:

When both startTime and endTime are omitted

- It is assumed that the time when the REST API server received the request was specified as endTime.
- It is assumed that the time which is one minute earlier than the time when the REST API server received the request was specified as startTime. However, if the endTime is 1970-01-01T00:01:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When both startTime and endTime are specified

- The monitored information within the specified time period is returned.
- Due to addition or deletion of storage pools, or no collection of information, the number of storage pools at a certain point of time might be smaller than the number output for count.

When only startTime is specified

- It is assumed that the time when the REST API server received the request was specified as endTime.

When only endTime is specified

- It is assumed that the time which is one hour earlier than endTime was specified as startTime. However, if endTime is 1970-01-01T01:00:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

- Due to addition or deletion of storage pools, or no collection of information, the number of storage pools at a certain point of time might be smaller than the number output for count.

When the range specified by startTime and endTime is invalid

- HTTP status code 400 (Bad Request) is returned when any of the following conditions is satisfied:
  - A date and time later than endTime is specified as startTime, including when endTime is not specified.
  - A date and time which is earlier than 1970-01-01T00:00:01Z is specified for startTime or endTime.
  - When all the following conditions are satisfied:
    - Specify true for startTimeExcluding, or false for endTimeIncluding.
    - The same date and time is specified for startTime and endTime, including the case when startTime is not specified and 1970-01-01T00:00:01Z is specified for endTime.

REQUEST PARAMETERS

Name	Description	Type	Data type
startTime	The start date and time of the monitor information to be obtained. If you specify startTimeExcluding, you can specify whether to include or exclude the specified time. By default, the specified time is included.	query	*string* (date-time)

Name	Description	Type	Data type
startTimeExcluding	Determines whether to include information of the specified startTime. ▪ false (default): The information is included. ▪ true: The information is not included.	query	*boolean* (default: "false")
endTime	The end date and time of the monitor information to be obtained. If you specify endTimeIncluding, you can specify whether to include or exclude the specified time. By default, the specified time is excluded. If it is omitted, it is the request receipt time.	query	*string* (date-time)
endTimeIncluding	Specify whether to include the specified endTime. ▪ false (default): The specified time is not included. ▪ true: The specified time is included.	query	*boolean* (default: "false")
enumerateContext	Token that helps obtain the next batch records. Do not specify this parameter for initial information extraction.	query	*string* (uuid)
count	The number of storage pool records to be obtained.	query	*integer* , { x ∈ ℤ | 1 ≤ x ≤ 32768 } (default: "32768")

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/performances/pool-capacities
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 188
Date: Fri, 30 Jul 2021 07:56:04 GMT

{"data":[{"timestamp":"2021-07-30T07:55:23Z","performanceObjects":[{"id":"376ae3cc-
70ce-4ea2-9d37-d64311576739","usedCapacity":0}]}],"totalCount":1,"hasNext":false,
"enumerateContext":null}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of storage pool capacity information (monitor information) at the specified time and the information about performanceObjects.

PROPERTIES

data: *object[]*

A list of storage pool capacity information (monitor information) at the specified time.

ITEMS

poolCapacityListResponseData: object (on page 659)

totalCount: *integer*

The total number of storage pools whose capacity information exists within the specified period.

hasNext: *boolean*

Indicates if some storage pools remain to be listed in performanceObjects.

enumerateContext: *string* (uuid)nullable

A token used to obtain the next value of performanceObjects for the applicable information. Null is output if the information is the last one.

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/performances/pool-capacities/<*id*>

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains the capacity information (monitor information) about the specified storage pool.

The capacity information is internally collected on a one-minute cycle. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.
- Immediately after Virtual Storage Software block is started or a storage pool is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no capacity information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

The following describes data to be output according to the specification of the startTime and endTime query parameters:

When both startTime and endTime are omitted

- It is assumed that the time when the REST API server received the request was specified as endTime.
- It is assumed that the time which is one minute earlier than the time when the REST API server received the request was specified as startTime. However, if the endTime is 1970-01-01T00:01:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When both startTime and endTime are specified

- The monitored information within the specified time period is returned.

When only startTime is specified

- It is assumed that the time when the REST API server received the request was specified as endTime.

When only endTime is specified

- It is assumed that the time which is one hour earlier than endTime was specified as startTime. However, if endTime is 1970-01-01T01:00:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When the range specified by startTime and endTime is invalid

- HTTP status code 400 (Bad Request) is returned when any of the following conditions is satisfied:
  - A date and time later than endTime is specified as startTime, including when endTime is not specified.
  - A date and time which is earlier than 1970-01-01T00:00:01Z is specified for startTime or endTime.
  - When all the following conditions are satisfied:
    - Specify true for startTimeExcluding, or false for endTimeIncluding.
    - The same date and time is specified for startTime and endTime, including the case when startTime is not specified and 1970-01-01T00:00:01Z is specified for endTime.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the storage pool.	path	*string* (uuid) required
startTime	The start date and time of the monitor information to be obtained. If you specify startTimeExcluding, you can specify whether to include or exclude the specified time. By default, the specified time is included.	query	*string* (date-time)

Name	Description	Type	Data type
startTimeExcluding	Determines whether to include information of the specified startTime. ▪ false (default): The information is included. ▪ true: The information is not included.	query	*boolean* (default: "false")
endTime	The end date and time of the monitor information to be obtained. If you specify endTimeIncluding, you can specify whether to include or exclude the specified time. By default, the specified time is excluded. If it is omitted, it is the request receipt time.	query	*string* (date-time)
endTimeIncluding	Specify whether to include the specified endTime. ▪ false (default): The specified time is not included. ▪ true: The specified time is included.	query	*boolean* (default: "false")

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/performances/pool-capacities/376ae3cc-70ce-4ea2-9d37-d64311576739
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
```

```
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 133
Date: Fri, 30 Jul 2021 07:56:44 GMT

{"data":[{"timestamp":"2021-07-30T07:56:23Z","performanceObjects":[{"id":"376ae3cc-
70ce-4ea2-9d37-d64311576739","usedCapacity":0}]}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of storage pool capacity information (monitor information) at the specified time.

PROPERTIES

data: ***object[]***

ITEMS

poolCapacityListResponseData: object (on page 659)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/performances/pools

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains a list of storage pool performance information (monitor information). Information about storage pools existing at the time of API execution is obtained.

Performance information is collected internally at one minute intervals. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.

- Immediately after Virtual Storage Software block is started or a storage pool is added or deleted, performance information might not be returned because it is not collected yet.

- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.

- If the information that could not be obtained is an attribute, null is returned.

- If no performance information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

If the number of storage pools is large, in order to obtain all the information, it is necessary to obtain it separately in plural times. If some storage pools are not obtained, hasNext in the response data shows true, and the token used to obtain the rest of the pools is output as enumerateContext. You can obtain the next volume by executing this API again with enumerateContext specified as the query parameter.

Also, by setting a count parameter, you can determine the number of storage pools you obtain. If lengthening the obtainment period (startTime/endTime), fewer number of the storage pools might return than the value set in the count parameter. If you want to obtain a large number of the storage pool information at once, set (startTime/endTime) briefly during the obtainment period.

totalCount in the response data is the total number of storage pools whose performance records were existing when the API was executed. The number might be changed with each API operation using enumerateContext. Similarly, as the information that is returned is the update at the time of API execution, you might obtain a record of a different timestamp value in a series of API operations using enumerateContext.

The following describes data to be output according to the specification of the startTime and endTime query parameters:

When both startTime and endTime are omitted

- It is assumed that the time when the REST API server received the request was specified as endTime.

- It is assumed that the time which is one minute earlier than the time when the REST API server received the request was specified as startTime. However, if the endTime is 1970-01-01T00:01:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When both startTime and endTime are specified

- The monitored information within the specified time period is returned.

- Due to addition or deletion of storage pools, or no collection of information, the number of storage pools at a certain point of time might be smaller than the number output for count.

When only startTime is specified

- It is assumed that the time when the REST API server received the request was specified as endTime.

When only endTime is specified

- It is assumed that the time which is one hour earlier than endTime was specified as startTime. However, if endTime is 1970-01-01T01:00:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

- Due to addition or deletion of storage pools, or no collection of information, the number of storage pools at a certain point of time might be smaller than the number output for count.

When the range specified by startTime and endTime is invalid

- HTTP status code 400 (Bad Request) is returned when any of the following conditions is satisfied:
  - A date and time later than endTime is specified as startTime, including when endTime is not specified.
  - A date and time which is earlier than 1970-01-01T00:00:01Z is specified for startTime or endTime.
  - When all the following conditions are satisfied:
    - Specify true for startTimeExcluding, or false for endTimeIncluding.
    - The same date and time is specified for startTime and endTime, including the case when startTime is not specified and 1970-01-01T00:00:01Z is specified for endTime.

REQUEST PARAMETERS

Name	Description	Type	Data type
startTime	The start date and time of the monitor information to be obtained. If you specify startTimeExcluding, you can specify whether to include or exclude the specified time. By default, the specified time is included.	query	*string* (date-time)

Name	Description	Type	Data type
startTimeExcluding	Determines whether to include information of the specified startTime. ▪ false (default): The information is included. ▪ true: The information is not included.	query	*boolean* (default: "false")
endTime	The end date and time of the monitor information to be obtained. If you specify endTimeIncluding, you can specify whether to include or exclude the specified time. By default, the specified time is excluded. If it is omitted, it is the request receipt time.	query	*string* (date-time)
endTimeIncluding	Specify whether to include the specified endTime. ▪ false (default): The specified time is not included. ▪ true: The specified time is included.	query	*boolean* (default: "false")
enumerateContext	Token that helps obtain the next batch records. Do not specify this parameter for initial information extraction.	query	*string* (uuid)
count	The number of storage pool records to be obtained.	query	*integer* , { x ∈ ℤ | 1 ≤ x ≤ 32768 } (default: "32768")

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/performances/pools
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 269
Date: Fri, 30 Jul 2021 07:56:05 GMT

{"data":[{"timestamp":"2021-07-30T07:55:23Z","performanceObjects":[{"id":"376ae3cc-
70ce-4ea2-9d37-d64311576739","volumeReadIOPS":0,"volumeWriteIOPS":0,
"volumeReadTransferRate":0.0,"volumeWriteTransferRate":0.0}]}],"totalCount":1,
"hasNext":false,"enumerateContext":null}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of storage pool performance information (monitor information) at the specified time.

PROPERTIES

data: ***object[]***

ITEMS

poolPerformanceListResponseData: object (on page 660)

totalCount: *integer*

The total number of storage pools whose performance information exists within the specified period.

hasNext: *boolean*

Indicates if some storage pools remain to be listed in performanceObjects.

enumerateContext: *string* (uuid)nullable

A token used to obtain the next value of performanceObjects for the applicable information. Null is output if the information is the last one.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/performances/pools/<*id*>

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains the performance information (monitor information) about the specified storage pool.

Performance information is collected internally at one minute intervals. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.
- Immediately after Virtual Storage Software block is started or a storage pool is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no performance information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

The following describes data to be output according to the specification of the startTime and endTime query parameters:

When both startTime and endTime are omitted

- It is assumed that the time when the REST API server received the request was specified as endTime.
- It is assumed that the time which is one minute earlier than the time when the REST API server received the request was specified as startTime. However, if the endTime is 1970-01-01T00:01:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When both startTime and endTime are specified

- The monitored information within the specified time period is returned.

When only startTime is specified

- It is assumed that the time when the REST API server received the request was specified as endTime.

When only endTime is specified

- It is assumed that the time which is one hour earlier than endTime was specified as startTime. However, if endTime is 1970-01-01T01:00:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When the range specified by startTime and endTime is invalid

- HTTP status code 400 (Bad Request) is returned when any of the following conditions is satisfied:
  - A date and time later than endTime is specified as startTime, including when endTime is not specified.
  - A date and time which is earlier than 1970-01-01T00:00:01Z is specified for startTime or endTime.
  - When all the following conditions are satisfied:
    - Specify true for startTimeExcluding, or false for endTimeIncluding.
    - The same date and time is specified for startTime and endTime, including the case when startTime is not specified and 1970-01-01T00:00:01Z is specified for endTime.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the storage pool.	path	*string* (uuid) required
startTime	The start date and time of the monitor information to be obtained. If you specify startTimeExcluding, you can specify whether to include or exclude the specified time. By default, the specified time is included.	query	*string* (date-time)

Name	Description	Type	Data type
startTimeExcluding	Determines whether to include information of the specified startTime. ▪ false (default): The information is included. ▪ true: The information is not included.	query	*boolean* (default: "false")
endTime	The end date and time of the monitor information to be obtained. If you specify endTimeIncluding, you can specify whether to include or exclude the specified time. By default, the specified time is excluded. If it is omitted, it is the request receipt time.	query	*string* (date-time)
endTimeIncluding	Specify whether to include the specified endTime. ▪ false (default): The specified time is not included. ▪ true: The specified time is included.	query	*boolean* (default: "false")

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
 curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/
simple/v1/objects/performances/pools/376ae3cc-70ce-4ea2-9d37-d64311576739
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
```

```
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 214
Date: Fri, 30 Jul 2021 07:56:45 GMT

{"data":[{"timestamp":"2021-07-30T07:56:23Z","performanceObjects":[{"id":"376ae3cc-
70ce-4ea2-9d37-d64311576739","volumeReadIOPS":0,"volumeWriteIOPS":0,
"volumeReadTransferRate":0.0,"volumeWriteTransferRate":0.0}]}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of storage pool performance information (monitor information) at the specified time.

PROPERTIES

data: ***object[]***

ITEMS

poolPerformanceListResponseData: object (on page 660)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/performances/ports

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains a list of compute port performance information (monitor information). Information about compute ports existing at the time of API execution is obtained.

Performance information is collected internally at one minute intervals. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned. To verify whether the information is the latest one, compare the timestamp contained in the response data with that obtained last.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.
- Immediately after Virtual Storage Software block is started or a compute port is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no performance information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

If the number of compute ports is large, in order to obtain all the information, it is necessary to obtain it separately in plural times. If some compute ports are not obtained, hasNext in the response data shows true, and the token used to obtain the rest of the compute ports is output as enumerateContext. You can obtain the next volume by executing this API again with enumerateContext specified as the query parameter.

Also, by setting a count parameter, you can determine the number of compute ports you obtain. If lengthening the obtainment period (startTime/endTime), fewer number of the compute ports might return than the value set in the count parameter. If you want to obtain a large number of the compute port information at once, set (startTime/endTime) briefly during the obtainment period.

totalCount in the response data is the total number of compute ports whose performance records were existing when the API was executed. The number might be changed with each API operation using enumerateContext. Similarly, as the information that is returned is the update at the time of API execution, you might obtain a record of a different timestamp value in a series of API operations using enumerateContext.

The following describes data to be output according to the specification of the startTime and endTime query parameters:

When both startTime and endTime are omitted

- It is assumed that the time when the REST API server received the request was specified as endTime.

- It is assumed that the time which is one minute earlier than the time when the REST API server received the request was specified as startTime. However, if the endTime is 1970-01-01T00:01:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When both startTime and endTime are specified

- The monitored information within the specified time period is returned.

- Due to addition or deletion of compute ports, or no collection of information, the number of compute ports at a certain point of time might be smaller than the number output for count.

When only startTime is specified

- It is assumed that the time when the REST API server received the request was specified as endTime.

When only endTime is specified

- It is assumed that the time which is one hour earlier than endTime was specified as startTime. However, if endTime is 1970-01-01T01:00:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

- Due to addition or deletion of compute ports, or no collection of information, the number of compute ports at a certain point of time might be smaller than the number output for count.

When the range specified by startTime and endTime is invalid

- HTTP status code 400 (Bad Request) is returned when any of the following conditions is satisfied:
  - A date and time later than endTime is specified as startTime, including when endTime is not specified.
  - A date and time which is earlier than 1970-01-01T00:00:01Z is specified for startTime or endTime.
  - When all the following conditions are satisfied:
    - Specify true for startTimeExcluding, or false for endTimeIncluding.
    - The same date and time is specified for startTime and endTime, including the case when startTime is not specified and 1970-01-01T00:00:01Z is specified for endTime.

REQUEST PARAMETERS

Name	Description	Type	Data type
startTime	The start date and time of the monitor information to be obtained. If you specify startTimeExcluding, you can specify whether to include or exclude the specified time. By default, the specified time is included.	query	*string* (date-time)
startTimeExcluding	Determines whether to include information of the specified startTime. ▪ false (default): The information is included. ▪ true: The information is not included.	query	*boolean* (default: "false")
endTime	The end date and time of the monitor information to be obtained.	query	*string* (date-time)

Name	Description	Type	Data type
	If you specify endTimeIncluding, you can specify whether to include or exclude the specified time. By default, the specified time is excluded. If it is omitted, it is the request receipt time.		
endTimeIncluding	Specify whether to include the specified endTime. ▪ false (default): The specified time is not included. ▪ true: The specified time is included.	query	*boolean* (default: "false")
enumerateContext	Token that helps obtain the next batch records. Do not specify this parameter for initial information extraction.	query	*string* (uuid)
count	The number of compute port records to be obtained.	query	*integer* , { x ∈ ℤ | 1 ≤ x ≤ 32768 } (default: "32768")

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/performances/ports
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 1445
Date: Fri, 30 Jul 2021 07:56:06 GMT
```

```
{"data":[{"timestamp":"2021-07-30T07:55:26Z","performanceObjects":[{"id":"aae705d6-
764e-4d5a-8e03-6205fc90b267","fc":null,"iscsi":{"target":{"readIOPS":0,"writeIOPS":0,
"readTransferRate":0.0,"writeTransferRate":0.0,"responseTime":0.0}}}]},
{"timestamp":"2021-07-30T07:55:27Z","performanceObjects":[{"id":"4126bd41-e590-435e-
b572-8b38266004eb","fc":null,"iscsi":{"target":{"readIOPS":0,"writeIOPS":0,
"readTransferRate":0.0,"writeTransferRate":0.0,"responseTime":0.0}}}]},
{"timestamp":"2021-07-30T07:55:28Z","performanceObjects":[{"id":"ef7a2da3-abba-4d29-
94e6-657fac92c6e5","fc":null,"iscsi":{"target":{"readIOPS":0,"writeIOPS":0,
"readTransferRate":0.0,"writeTransferRate":0.0,"responseTime":0.0}}}]},
{"timestamp":"2021-07-30T07:55:30Z","performanceObjects":[{"id":"620aa11e-ca57-47a3-
9320-0c03e039a4c6","fc":null,"iscsi":{"target":{"readIOPS":0,"writeIOPS":0,
"readTransferRate":0.0,"writeTransferRate":0.0,"responseTime":0.0}}}]},
{"timestamp":"2021-07-30T07:55:31Z","performanceObjects":[{"id":"343db11a-21d0-45eb-
a66c-c31ab1ac6863","fc":null,"iscsi":{"target":{"readIOPS":0,"writeIOPS":0,
"readTransferRate":0.0,"writeTransferRate":0.0,"responseTime":0.0}}}]},
{"timestamp":"2021-07-30T07:55:34Z","performanceObjects":[{"id":"74478bb8-c99c-470b-
9fe5-55fa230aafc3","fc":null,"iscsi":{"target":{"readIOPS":0,"writeIOPS":0,
"readTransferRate":0.0,"writeTransferRate":0.0,"responseTime":0.0}}}]}],
"totalCount":6,"hasNext":false,"enumerateContext":null}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of compute port performance information (monitor information) at the specified time and the information about performanceObjects.

PROPERTIES

data: *object[]*

A list of compute port performance information (monitor information) at the specified time.

ITEMS

portPerformanceListResponseData: object (on page 664)

totalCount: *integer*

The total number of compute ports whose performance information exists within the specified period.

hasNext: *boolean*

Indicates if some compute ports remain to be listed in performanceObjects.

enumerateContext: *string* (uuid)nullable

A token used to obtain the next value of performanceObjects for the applicable information. Null is output if the information is the last one.

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/performances/ports/<*id*>

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains the performance information (monitor information) about the specified compute port.

Performance information is collected internally at one minute intervals. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned. To verify whether the information is the latest one, compare the timestamp contained in the response data with that obtained last.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.
- Immediately after Virtual Storage Software block is started or a compute port is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no performance information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

The following describes data to be output according to the specification of the startTime and endTime query parameters:

When both startTime and endTime are omitted

- It is assumed that the time when the REST API server received the request was specified as endTime.
- It is assumed that the time which is one minute earlier than the time when the REST API server received the request was specified as startTime. However, if the endTime is 1970-01-01T00:01:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When both startTime and endTime are specified

- The monitored information within the specified time period is returned.

When only startTime is specified

- It is assumed that the time when the REST API server received the request was specified as endTime.

When only endTime is specified

- It is assumed that the time which is one hour earlier than endTime was specified as startTime. However, if endTime is 1970-01-01T01:00:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When the range specified by startTime and endTime is invalid

- HTTP status code 400 (Bad Request) is returned when any of the following conditions is satisfied:
  - A date and time later than endTime is specified as startTime, including when endTime is not specified.
  - A date and time which is earlier than 1970-01-01T00:00:01Z is specified for startTime or endTime.
  - When all the following conditions are satisfied:
    - Specify true for startTimeExcluding, or false for endTimeIncluding.
    - The same date and time is specified for startTime and endTime, including the case when startTime is not specified and 1970-01-01T00:00:01Z is specified for endTime.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the compute port.	path	*string* (uuid) required
startTime	The start date and time of the monitor information to be obtained. If you specify startTimeExcluding, you can specify whether to include or exclude the specified time. By default, the specified time is included.	query	*string* (date-time)

Name	Description	Type	Data type
startTimeExcluding	Determines whether to include information of the specified startTime. ▪ false (default): The information is included. ▪ true: The information is not included.	query	*boolean* (default: "false")
endTime	The end date and time of the monitor information to be obtained. If you specify endTimeIncluding, you can specify whether to include or exclude the specified time. By default, the specified time is excluded. If it is omitted, it is the request receipt time.	query	*string* (date-time)
endTimeIncluding	Specify whether to include the specified endTime. ▪ false (default): The specified time is not included. ▪ true: The specified time is included.	query	*boolean* (default: "false")

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/performances/ports/343db11a-21d0-45eb-a66c-c31ab1ac6863
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
```

```
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 240
Date: Fri, 30 Jul 2021 07:56:45 GMT

{"data":[{"timestamp":"2021-07-30T07:56:27Z","performanceObjects":[{"id":"343db11a-
21d0-45eb-a66c-c31ab1ac6863","fc":null,"iscsi":{"target":{"readIOPS":0,"writeIOPS":0,
"readTransferRate":0.0,"writeTransferRate":0.0,"responseTime":0.0}}}]}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

portPerformanceListResponse: object (on page 663)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/performances/storage

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains the storage cluster performance information (monitor information).

Performance information is collected internally at one minute intervals. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.
- Immediately after Virtual Storage Software block is started, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no performance information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

The following describes data to be output according to the specification of the startTime and endTime query parameters:

When both startTime and endTime are omitted

- It is assumed that the time when the REST API server received the request was specified as endTime.
- It is assumed that the time which is one minute earlier than the time when the REST API server received the request was specified as startTime. However, if the endTime is 1970-01-01T00:01:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When both startTime and endTime are specified

- The monitored information within the specified time period is returned.

When only startTime is specified

- It is assumed that the time when the REST API server received the request was specified as endTime.

When only endTime is specified

- It is assumed that the time which is one hour earlier than endTime was specified as startTime. However, if endTime is 1970-01-01T01:00:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When the range specified by startTime and endTime is invalid

- HTTP status code 400 (Bad Request) is returned when any of the following conditions is satisfied:
  - A date and time later than endTime is specified as startTime, including when endTime is not specified.
  - A date and time which is earlier than 1970-01-01T00:00:01Z is specified for startTime or endTime.
  - When all the following conditions are satisfied:
    - Specify true for startTimeExcluding, or false for endTimeIncluding.
    - The same date and time is specified for startTime and endTime, including the case when startTime is not specified and 1970-01-01T00:00:01Z is specified for endTime.

REQUEST PARAMETERS

Name	Description	Type	Data type
startTime	The start date and time of the monitor information to be obtained. If you specify startTimeExcluding, you can specify whether to include or exclude the specified time. By default, the specified time is included.	query	*string* (date-time)

Name	Description	Type	Data type
startTimeExcluding	Determines whether to include information of the specified startTime. ▪ false (default): The information is included. ▪ true: The information is not included.	query	*boolean* (default: "false")
endTime	The end date and time of the monitor information to be obtained. If you specify endTimeIncluding, you can specify whether to include or exclude the specified time. By default, the specified time is excluded. If it is omitted, it is the request receipt time.	query	*string* (date-time)
endTimeIncluding	Specify whether to include the specified endTime. ▪ false (default): The specified time is not included. ▪ true: The specified time is included.	query	*boolean* (default: "false")

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/performances/storage
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
```

```
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 182
Date: Fri, 30 Jul 2021 07:56:06 GMT

{"data":[{"timestamp":"2021-07-30T07:55:23Z","performanceObjects":[{"id":"4d17fde1-
0cdb-4761-8d46-5cf5fb38c2ae","averageCpuUsage":22.368140738937356,
"averageMemoryUsage":16.7863}]}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of storage cluster performance information (monitor information) at the specified time.

PROPERTIES

data: *object[]*

ITEMS

storagePerformanceListResponseData: object (on page 710)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/performances/storage-nodes

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains a list of storage node performance information (monitor information). Information about storage nodes existing at the time of API execution is obtained.

Performance information is collected internally at one minute intervals. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.

- Immediately after Virtual Storage Software block is started or a storage node is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no performance information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

If the number of storage nodes is large, in order to obtain all the information, it is necessary to obtain it separately in plural times. If some storage nodes are not obtained, hasNext in the response data shows true, and the token used to obtain the rest of the nodes is output as enumerateContext. You can obtain the next volume by executing this API again with enumerateContext specified as the query parameter.

Also, by setting a count parameter, you can determine the number of storage nodes you obtain. If lengthening the obtainment period (startTime/endTime), fewer number of the storage nodes might return than the value set in the count parameter. If you want to obtain a large number of the storage node information at once, set (startTime/endTime) briefly during the obtainment period.

totalCount in the response data is the total number of storage nodes whose performance records were existing when the API was executed. The number might be changed with each API operation using enumerateContext. Similarly, as the information that is returned is the update at the time of API execution, you might obtain a record of a different timestamp value in a series of API operations using enumerateContext.

The following describes data to be output according to the specification of the startTime and endTime query parameters:

When both startTime and endTime are omitted

- It is assumed that the time when the REST API server received the request was specified as endTime.
- It is assumed that the time which is one minute earlier than the time when the REST API server received the request was specified as startTime. However, if the endTime is 1970-01-01T00:01:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When both startTime and endTime are specified

- The monitored information within the specified time period is returned.
- Due to addition or deletion of storage nodes, or no collection of information, the number of storage nodes at a certain point of time might be smaller than the number output for count.

When only startTime is specified

- It is assumed that the time when the REST API server received the request was specified as endTime.

When only endTime is specified

- It is assumed that the time which is one hour earlier than endTime was specified as startTime. However, if endTime is 1970-01-01T01:00:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

- Due to addition or deletion of storage nodes, or no collection of information, the number of storage nodes at a certain point of time might be smaller than the number output for count.

When the range specified by startTime and endTime is invalid

- HTTP status code 400 (Bad Request) is returned when any of the following conditions is satisfied:
  - A date and time later than endTime is specified as startTime, including when endTime is not specified.
  - A date and time which is earlier than 1970-01-01T00:00:01Z is specified for startTime or endTime.
  - When all the following conditions are satisfied:
    - Specify true for startTimeExcluding, or false for endTimeIncluding.
    - The same date and time is specified for startTime and endTime, including the case when startTime is not specified and 1970-01-01T00:00:01Z is specified for endTime.

REQUEST PARAMETERS

Name	Description	Type	Data type
startTime	The start date and time of the monitor information to be obtained. If you specify startTimeExcluding, you can specify whether to include or exclude the specified time. By default, the specified time is included.	query	*string* (date-time)

Name	Description	Type	Data type
startTimeExcluding	Determines whether to include information of the specified startTime. ▪ false (default): The information is included. ▪ true: The information is not included.	query	*boolean* (default: "false")
endTime	The end date and time of the monitor information to be obtained. If you specify endTimeIncluding, you can specify whether to include or exclude the specified time. By default, the specified time is excluded. If it is omitted, it is the request receipt time.	query	*string* (date-time)
endTimeIncluding	Specify whether to include the specified endTime. ▪ false (default): The specified time is not included. ▪ true: The specified time is included.	query	*boolean* (default: "false")
enumerateContext	Token that helps obtain the next batch records. Do not specify this parameter for initial information extraction.	query	*string* (uuid)
count	The number of storage node records to be obtained.	query	*integer* , { x ∈ ℤ | 1 ≤ x ≤ 32768 } (default: "32768")

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/performances/storage-nodes
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 7558
Date: Fri, 30 Jul 2021 07:56:07 GMT

{"data":[{"timestamp":"2021-07-30T07:55:26Z","performanceObjects":[{"id":"9f29d742-
9a75-4cef-9539-35b3e76c8a20","volumeReadIOPS":0,"volumeWriteIOPS":0,
"volumeReadTransferRate":0.0,"volumeWriteTransferRate":0.0,"driveReadIOPS":90,
"driveWriteIOPS":595,"driveReadTransferRate":12.189104352678571,
"driveWriteTransferRate":98.70378766741071,"cpu":[{"index":3,
"process":"StorageController","usage":20.2},{"index":4,"process":"StorageController",
"usage":19.8},{"index":5,"process":"StorageController","usage":19.2},{"index":0,
"process":"SystemController","usage":2.9697525206232815},{"index":6,
"process":"SystemController","usage":3.545572674958233},{"index":7,
"process":"SystemController","usage":37.03225806451613},{"index":8,
"process":"SystemController","usage":25.918212798182505},{"index":9,
"process":"SystemController","usage":31.561522898657344},{"index":10,
"process":"SystemController","usage":53.470437017994854},{"index":11,
"process":"SystemController","usage":38.08198849935077},{"index":12,
"process":"SystemController","usage":43.76379690949228},{"index":13,
"process":"SystemController","usage":23.74976933013471},{"index":1,
"process":"StorageController","usage":0.0},{"index":2,"process":"StorageController",
"usage":0.0}],"memory":{"usage":16.9712}}]},{"timestamp":"2021-07-30T07:55:27Z",
"performanceObjects":[{"id":"9d36c162-e379-4c85-bcc2-ccf98fe774a6","volumeReadIOPS":0,
"volumeWriteIOPS":0,"volumeReadTransferRate":0.0,"volumeWriteTransferRate":0.0,
"driveReadIOPS":107,"driveWriteIOPS":515,"driveReadTransferRate":18.76904296875,
"driveWriteTransferRate":100.03098958333334,"cpu":[{"index":3,
"process":"StorageController","usage":19.0},{"index":4,"process":"StorageController",
"usage":19.0},{"index":5,"process":"StorageController","usage":19.4},{"index":0,
"process":"SystemController","usage":3.0224075039082856},{"index":6,
"process":"SystemController","usage":13.068087231563178},{"index":7,
"process":"SystemController","usage":7.199723087573555},{"index":8,
"process":"SystemController","usage":99.96686547382373},{"index":9,
"process":"SystemController","usage":82.88707799767171},{"index":10,
"process":"SystemController","usage":4.885601238603l304},{"index":11,
"process":"SystemController","usage":20.480704129993228},{"index":12,
"process":"SystemController","usage":5.001689760054073},{"index":13,
"process":"SystemController","usage":9.536354056902002},{"index":1,
"process":"StorageController","usage":0.0},{"index":2,"process":"StorageController",
"usage":0.0}],"memory":{"usage":16.2665}}]},{"timestamp":"2021-07-30T07:55:28Z",
"performanceObjects":[{"id":"44f1d113-405e-448f-ad77-fd5554971c36","volumeReadIOPS":0,
"volumeWriteIOPS":0,"volumeReadTransferRate":0.0,"volumeWriteTransferRate":0.0,
"driveReadIOPS":111,"driveWriteIOPS":528,"driveReadTransferRate":11.852197265625,
"driveWriteTransferRate":99.02229817708333,"cpu":[{"index":3,
"process":"StorageController","usage":20.0},{"index":4,"process":"StorageController",
"usage":19.6},{"index":5,"process":"StorageController","usage":20.4},{"index":0,
"process":"SystemController","usage":0.4989190088142358},{"index":6,
"process":"SystemController","usage":0.9884402747528899},{"index":7,
"process":"SystemController","usage":13.298791018998275},{"index":8,
"process":"SystemController","usage":61.25504710632571},{"index":9,
"process":"SystemController","usage":37.81102891728312},{"index":10,
"process":"SystemController","usage":25.794871794871792},{"index":11,
"process":"SystemController","usage":55.40360240160107},{"index":12,
"process":"SystemController","usage":24.18716669533803 6},{"index":13,
"process":"SystemController","usage":21.139581928523263},{"index":1,
"process":"StorageController","usage":0.0},{"index":2,"process":"StorageController",
"usage":0.0}],"memory":{"usage":16.2983}}]},{"timestamp":"2021-07-30T07:55:30Z",
"performanceObjects":[{"id":"dd5ed12b-cf39-47da-9a30-5ba53404050e","volumeReadIOPS":0,
"volumeWriteIOPS":0,"volumeReadTransferRate":0.0,"volumeWriteTransferRate":0.0,
"driveReadIOPS":101,"driveWriteIOPS":638,"driveReadTransferRate":10.953245069159836,
"driveWriteTransferRate":104.53008132684427,"cpu":[{"index":3,
"process":"StorageController","usage":20.0},{"index":4,"process":"StorageController",
"usage":20.2},{"index":5,"process":"StorageController","usage":20.0},{"index":0,
"process":"SystemController","usage":2.627986348122867},{"index":6,
"process":"SystemController","usage":3.821548821548822},{"index":7,
"process":"SystemController","usage":34.89841215639406},{"index":8,
"process":"SystemController","usage":49.69818913480885},{"index":9,
"process":"SystemController","usage":41.59217405970653},{"index":10,
"process":"SystemController","usage":30.353817504655495},{"index":11,
"process":"SystemController","usage":40.27498323272971},{"index":12,
"process":"SystemController","usage":22.66144147214176},{"index":13,
"process":"SystemController","usage":32.13020745488278},{"index":1,
"process":"StorageController","usage":0.0},{"index":2,"process":"StorageController",
"usage":0.0}],"memory":{"usage":16.9205}}]},{"timestamp":"2021-07-30T07:55:31Z",
"performanceObjects":[{"id":"8fdbc008-ec6c-42b3-b930-146c94c5cef9","volumeReadIOPS":0,
"volumeWriteIOPS":0,"volumeReadTransferRate":0.0,"volumeWriteTransferRate":0.0,
"driveReadIOPS":128,"driveWriteIOPS":560,"driveReadTransferRate":19.243343098958334,
"driveWriteTransferRate":102.51724446614584,"cpu":[{"index":3,
"process":"StorageController","usage":19.6},{"index":4,"process":"StorageController",
"usage":19.0},{"index":5,"process":"StorageController","usage":19.4},{"index":0,
"process":"SystemController","usage":0.6439894319682958},{"index":6,
"process":"SystemController","usage":1.2248322147651007},{"index":7,
"process":"SystemController","usage":29.575051405071967},{"index":8,
"process":"SystemController","usage":66.52195692102187},{"index":9,
"process":"SystemController","usage":45.301899592944366},{"index":10,
"process":"SystemController","usage":16.746739876458477},{"index":11,
"process":"SystemController","usage":33.82227528568992},{"index":12,
"process":"SystemController","usage":30.38144849605108},{"index":13,
"process":"SystemController","usage":17.303447104771607},{"index":1,
"process":"StorageController","usage":0.0},{"index":2,"process":"StorageController",
"usage":0.0}],"memory":{"usage":16.2693}}]},{"timestamp":"2021-07-30T07:55:34Z",
"performanceObjects":[{"id":"a5b19d19-ca5d-4d19-988d-d625850c85de","volumeReadIOPS":0,
"volumeWriteIOPS":0,"volumeReadTransferRate":0.0,"volumeWriteTransferRate":0.0,
"driveReadIOPS":116,"driveWriteIOPS":410,"driveReadTransferRate":16.966540727459016,
"driveWriteTransferRate":79.21639664446721,"cpu":[{"index":3,
"process":"StorageController","usage":19.0},{"index":4,"process":"StorageController",
"usage":19.2},{"index":5,"process":"StorageController","usage":18.8},{"index":0,
"process":"SystemController","usage":7.8081969868783405},{"index":6,
"process":"SystemController","usage":7.868525896414343},{"index":7,
"process":"SystemController","usage":55.41088941494758},{"index":8,
"process":"SystemController","usage":45.48821548821549},{"index":9,
"process":"SystemController","usage":45.67713610266802},{"index":10,
"process":"SystemController","usage":55.56117290192113},{"index":11,
"process":"SystemController","usage":51.22937015830246},{"index":12,
"process":"SystemController","usage":43.19526627218935},{"index":13,
"process":"SystemController","usage":46.47506339814032},{"index":1,
"process":"StorageController","usage":0.0},{"index":2,"process":"StorageController",
"usage":0.0}],"memory":{"usage":18.1029}}]}],"totalCount":6,"hasNext":false,
"enumerateContext":null}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of storage node performance information (monitor information) at the specified time and the information about performanceObjects.

PROPERTIES

data: ***object[]***

ITEMS

storageNodePerformanceListResponseData: object (on page 709)

totalCount: *integer*

The total number of storage nodes whose performance information exists within the specified period.

hasNext: *boolean*

Indicates if some storage nodes remain to be listed in performanceObjects.

enumerateContext: *string* (uuid)nullable

A token used to obtain the next value of performanceObjects for the applicable information. Null is output if the information is the last one.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/performances/storage-nodes/<*id*>

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains the performance information (monitor information) about the specified storage node.

Performance information is collected internally at one minute intervals. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.
- Immediately after Virtual Storage Software block is started or a storage node is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no performance information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

The following describes data to be output according to the specification of the startTime and endTime query parameters:

When both startTime and endTime are omitted

- It is assumed that the time when the REST API server received the request was specified as endTime.
- It is assumed that the time which is one minute earlier than the time when the REST API server received the request was specified as startTime. However, if the endTime is 1970-01-01T00:01:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When both startTime and endTime are specified

- The monitored information within the specified time period is returned.

When only startTime is specified

- It is assumed that the time when the REST API server received the request was specified as endTime.

When only endTime is specified

- It is assumed that the time which is one hour earlier than endTime was specified as startTime. However, if endTime is 1970-01-01T01:00:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When the range specified by startTime and endTime is invalid

- HTTP status code 400 (Bad Request) is returned when any of the following conditions is satisfied:
  - A date and time later than endTime is specified as startTime, including when endTime is not specified.
  - A date and time which is earlier than 1970-01-01T00:00:01Z is specified for startTime or endTime.
  - When all the following conditions are satisfied:
    - Specify true for startTimeExcluding, or false for endTimeIncluding.
    - The same date and time is specified for startTime and endTime, including the case when startTime is not specified and 1970-01-01T00:00:01Z is specified for endTime.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Storage node ID.	path	*string* (uuid) required
startTime	The start date and time of the monitor information to be obtained. If you specify startTimeExcluding, you can specify whether to include or exclude the specified time. By default, the specified time is included.	query	*string* (date-time)

Name	Description	Type	Data type
startTimeExcluding	Determines whether to include information of the specified startTime. ▪ false (default): The information is included. ▪ true: The information is not included.	query	*boolean* (default: "false")
endTime	The end date and time of the monitor information to be obtained. If you specify endTimeIncluding, you can specify whether to include or exclude the specified time. By default, the specified time is excluded. If it is omitted, it is the request receipt time.	query	*string* (date-time)
endTimeIncluding	Specify whether to include the specified endTime. ▪ false (default): The specified time is not included. ▪ true: The specified time is included.	query	*boolean* (default: "false")

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/performances/storage-nodes/dd5ed12b-cf39-47da-9a30-5ba53404050e
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
```

```
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 1261
Date: Fri, 30 Jul 2021 07:56:46 GMT

{"data":[{"timestamp":"2021-07-30T07:56:30Z","performanceObjects":[{"id":"dd5ed12b-
cf39-47da-9a30-5ba53404050e","volumeReadIOPS":0,"volumeWriteIOPS":0,
"volumeReadTransferRate":0.0,"volumeWriteTransferRate":0.0,"driveReadIOPS":102,
"driveWriteIOPS":656,"driveReadTransferRate":11.203800455729167,
"driveWriteTransferRate":107.43893229166666,"cpu":[{"index":3,
"process":"StorageController","usage":20.0},{"index":4,"process":"StorageController",
"usage":20.2},{"index":5,"process":"StorageController","usage":19.6},{"index":0,
"process":"SystemController","usage":2.8387968908415004},{"index":6,
"process":"SystemController","usage":3.911735205616851},{"index":7,
"process":"SystemController","usage":25.239234449760765},{"index":8,
"process":"SystemController","usage":40.2974983096687},{"index":9,
"process":"SystemController","usage":48.660037080734874},{"index":10,
"process":"SystemController","usage":29.531568228105908},{"index":11,
"process":"SystemController","usage":41.719852200201544},{"index":12,
"process":"SystemController","usage":21.63673331074212},{"index":13,
"process":"SystemController","usage":45.30536288276095},{"index":1,
"process":"StorageController","usage":0.0},{"index":2,"process":"StorageController",
"usage":0.0}],"memory":{"usage":16.9268}}]}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

storageNodePerformanceListResponse: object (on page 709)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/performances/volume-capacities

Required Role: Storage, Monitor, Resource, VpsStorage, or VpsMonitor

DESCRIPTION

Obtains a list of volume capacity information (monitor information). Information about volumes existing at the time of API execution is obtained.

The capacity information is internally collected on a one-minute cycle. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.
- Immediately after Virtual Storage Software block is started or a volume is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no capacity information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

If the number of volumes is large, in order to obtain all the information, it is necessary to obtain it separately in plural times. If some volumes are not obtained, hasNext in the response data shows true, and the token used to obtain the rest of the volumes is output as enumerateContext. You can obtain the next volume by executing this API again with enumerateContext specified as the query parameter.

Also, by setting a count parameter, you can determine the number of volume records you obtain. If lengthening the obtainment period (startTime/endTime), fewer number of the volumes might return than the value set in the count parameter. If you want to obtain a large number of the volume information at once, set (startTime/endTime) briefly during the obtainment period.

totalCount in the response data is the total number of volumes whose capacity records were existing when the API was executed. The number might be changed with each API operation using enumerateContext. Similarly, as the information that is returned is the update at the time of API execution, you might obtain a record of a different timestamp value in a series of API operations using enumerateContext.

The following describes data to be output according to the specification of the startTime and endTime query parameters:

When both startTime and endTime are omitted

- It is assumed that the time when the REST API server received the request was specified as endTime.
- It is assumed that the time which is one minute earlier than the time when the REST API server received the request was specified as startTime. However, if the endTime is 1970-01-01T00:01:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When both startTime and endTime are specified

- The monitored information within the specified time period is returned.
- Due to addition or deletion of volumes, or no collection of information, the number of volumes at a certain point of time might be smaller than the number output for count.

When only startTime is specified

- It is assumed that the time when the REST API server received the request was specified as endTime.

When only endTime is specified

- It is assumed that the time which is one hour earlier than endTime was specified as startTime. However, if endTime is 1970-01-01T01:00:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.
- Due to addition or deletion of volumes, or no collection of information, the number of volumes at a certain point of time might be smaller than the number output for count.

When the range specified by startTime and endTime is invalid

- HTTP status code 400 (Bad Request) is returned when any of the following conditions is satisfied:
  - A date and time later than endTime is specified as startTime, including when endTime is not specified.
  - A date and time which is earlier than 1970-01-01T00:00:01Z is specified for startTime or endTime.
  - When all the following conditions are satisfied:
    - Specify true for startTimeExcluding, or false for endTimeIncluding.
    - The same date and time is specified for startTime and endTime, including the case when startTime is not specified and 1970-01-01T00:00:01Z is specified for endTime.

REQUEST PARAMETERS

Name	Description	Type	Data type
startTime	The start date and time of the monitor information to be obtained.	query	*string* (date-time)

Name	Description	Type	Data type
	If you specify startTimeExcluding, you can specify whether to include or exclude the specified time. By default, the specified time is included.		
startTimeExcluding	Determines whether to include information of the specified startTime. ▪ false (default): The information is included. ▪ true: The information is not included.	query	*boolean* (default: "false")
endTime	The end date and time of the monitor information to be obtained. If you specify endTimeIncluding, you can specify whether to include or exclude the specified time. By default, the specified time is excluded. If it is omitted, it is the request receipt time.	query	*string* (date-time)
endTimeIncluding	Specify whether to include the specified endTime. ▪ false (default): The specified time is not included. ▪ true: The specified time is included.	query	*boolean* (default: "false")
enumerateContext	Token that helps obtain the next batch records. Do not specify this parameter for initial information extraction.	query	*string* (uuid)
count	The number of volume records to be obtained.	query	*integer* ,{ x ∈ ℤ | 1 ≤ x ≤ 32768 } (default: "32768")

Name	Description	Type	Data type
vpsId	The ID of the virtual private storage (VPS) that the acquisition-target resource belongs to. To filter out the resources that do not belong to the VPS, specify "system". To filter the resources by the VPS that the resources belong to, specify it in UUID format.	query	*string* , must match /^system$\|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/performances/volume-capacities

HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 518
Date: Thu, 20 Oct 2022 05:26:07 GMT

{"data":[{"timestamp":"2022-10-20T05:25:11Z","performanceObjects":[{"id":"7e4eb895-
48cc-4f48-b228-9224af6aa4f6","vpsId":"54665c29-784a-4b71-9b61-2894d55697e9",
"capacityUsage":0}]},{"timestamp":"2022-10-20T05:25:15Z","performanceObjects":
[{"id":"27f8944a-e98b-490d-b8fb-14583ceabc65","vpsId":"00000000-0000-0000-0000-
000000000000","capacityUsage":0},{"id":"35d16c28-b033-4c00-9956-67e8221493d3",
"vpsId":"00000000-0000-0000-0000-000000000000","capacityUsage":0}]}],"totalCount":3,
"hasNext":false,"enumerateContext":null}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of capacity information of a volume (monitor information) at the specified time, and information about performanceObjects.

PROPERTIES

data: *object[]*

A list of capacity information of a volume (monitor information) at the specified time.

ITEMS

volumeCapacityListResponseData: object (on page 739)

totalCount: *integer*

The total number of volumes whose capacity information exists within the specified time period.

hasNext: *boolean*

Indicates if some volumes remain to be listed in performanceObjects.

enumerateContext: *string* (uuid) nullable

A token used to obtain the next value of performanceObjects for the applicable information. Null is output if the information is the last one.

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/performances/volume-capacities/<*id*>

Required Role: Storage, Monitor, Resource, VpsStorage, or VpsMonitor

DESCRIPTION

Obtains the capacity information (monitor information) about of the specified volume.

The capacity information is internally collected on a one-minute cycle. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.

- Immediately after Virtual Storage Software block is started or a volume is added or deleted, performance information might not be returned because it is not collected yet.

- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.

- If the information that could not be obtained is an attribute, null is returned.

- If no capacity information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

The following describes data to be output according to the specification of the startTime and endTime query parameters:

When both startTime and endTime are omitted

- It is assumed that the time when the REST API server received the request was specified as endTime.

- It is assumed that the time which is one minute earlier than the time when the REST API server received the request was specified as startTime. However, if the endTime is 1970-01-01T00:01:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When both startTime and endTime are specified

- The monitored information within the specified time period is returned.

When only startTime is specified

- It is assumed that the time when the REST API server received the request was specified as endTime.

When only endTime is specified

- It is assumed that the time which is one hour earlier than endTime was specified as startTime. However, if endTime is 1970-01-01T01:00:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When the range specified by startTime and endTime is invalid

- HTTP status code 400 (Bad Request) is returned when any of the following conditions is satisfied:
    - A date and time later than endTime is specified as startTime, including when endTime is not specified.
    - A date and time which is earlier than 1970-01-01T00:00:01Z is specified for startTime or endTime.
    - When all the following conditions are satisfied:
        - Specify true for startTimeExcluding, or false for endTimeIncluding.
        - The same date and time is specified for startTime and endTime, including the case when startTime is not specified and 1970-01-01T00:00:01Z is specified for endTime.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Volume ID.	path	*string* (uuid) required
startTime	The start date and time of the monitor information to be obtained. If you specify startTimeExcluding, you can specify whether to include or exclude the specified time. By default, the specified time is included.	query	*string* (date-time)

Name	Description	Type	Data type
startTimeExcluding	Determines whether to include information of the specified startTime. ▪ false (default): The information is included. ▪ true: The information is not included.	query	*boolean* (default: "false")
endTime	The end date and time of the monitor information to be obtained. If you specify endTimeIncluding, you can specify whether to include or exclude the specified time. By default, the specified time is excluded. If it is omitted, it is the request receipt time.	query	*string* (date-time)
endTimeIncluding	Specify whether to include the specified endTime. ▪ false (default): The specified time is not included. ▪ true: The specified time is included.	query	*boolean* (default: "false")

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/performances/volume-capacities/7e4eb895-48cc-4f48-b228-9224af6aa4f6

HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
```

```
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 181
Date: Thu, 20 Oct 2022 05:37:49 GMT

{"data":[{"timestamp":"2022-10-20T05:37:15Z","performanceObjects":[{"id":"7e4eb895-
48cc-4f48-b228-9224af6aa4f6","vpsId":"54665c29-784a-4b71-9b61-2894d55697e9",
"capacityUsage":0}]}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of capacity information of a volume (monitor information) at the specified time.

PROPERTIES

data: *object[]*

ITEMS

volumeCapacityListResponseData: object (on page 739)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/performances/volumes

Required Role: Storage, Monitor, Resource, VpsStorage, or VpsMonitor

DESCRIPTION

Obtains a list of volume performance information (monitor information). Information about volumes existing at the time of API execution is obtained.

Performance information is collected internally at one minute intervals. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.

- Immediately after Virtual Storage Software block is started or a volume is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no performance information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

If the number of volumes is large, in order to obtain all the information, it is necessary to obtain it separately in plural times. If some volumes are not obtained, hasNext in the response data shows true, and the token used to obtain the rest of the volumes is output as enumerateContext. You can obtain the next volume by executing this API again with enumerateContext specified as the query parameter.

Also, by setting a count parameter, you can determine the number of volume records you obtain. If lengthening the obtainment period (startTime/endTime), fewer number of the volumes might return than the value set in the count parameter. If you want to obtain a large number of the volume information at once, set (startTime/endTime) briefly during the obtainment period.

totalCount in the response data is the total number of volumes whose performance records were existing when the API was executed. The number might be changed with each API operation using enumerateContext. Similarly, as the information that is returned is the update at the time of API execution, you might obtain a record of a different timestamp value in a series of API operations using enumerateContext.

The following describes data to be output according to the specification of the startTime and endTime query parameters:

When both startTime and endTime are omitted

- It is assumed that the time when the REST API server received the request was specified as endTime.
- It is assumed that the time which is one minute earlier than the time when the REST API server received the request was specified as startTime. However, if the endTime is 1970-01-01T00:01:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When both startTime and endTime are specified

- The monitored information within the specified time period is returned.
- Due to addition or deletion of volumes, or no collection of information, the number of volumes at a certain point of time might be smaller than the number output for count.

When only startTime is specified

- It is assumed that the time when the REST API server received the request was specified as endTime.

When only endTime is specified

- It is assumed that the time which is one hour earlier than endTime was specified as startTime. However, if endTime is 1970-01-01T01:00:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

- Due to addition or deletion of volumes, or no collection of information, the number of volumes at a certain point of time might be smaller than the number output for count.

When the range specified by startTime and endTime is invalid

- HTTP status code 400 (Bad Request) is returned when any of the following conditions is satisfied:
  - A date and time later than endTime is specified as startTime, including when endTime is not specified.
  - A date and time which is earlier than 1970-01-01T00:00:01Z is specified for startTime or endTime.
  - When all the following conditions are satisfied:
    - Specify true for startTimeExcluding, or false for endTimeIncluding.
    - The same date and time is specified for startTime and endTime, including the case when startTime is not specified and 1970-01-01T00:00:01Z is specified for endTime.

REQUEST PARAMETERS

Name	Description	Type	Data type
startTime	The start date and time of the monitor information to be obtained. If you specify startTimeExcluding, you can specify whether to include or exclude the specified time. By default, the specified time is included.	query	*string* (date-time)

Name	Description	Type	Data type
startTimeExcluding	Determines whether to include information of the specified startTime. ▪ false (default): The information is included. ▪ true: The information is not included.	query	*boolean* (default: "false")
endTime	The end date and time of the monitor information to be obtained. If you specify endTimeIncluding, you can specify whether to include or exclude the specified time. By default, the specified time is excluded. If it is omitted, it is the request receipt time.	query	*string* (date-time)
endTimeIncluding	Specify whether to include the specified endTime. ▪ false (default): The specified time is not included. ▪ true: The specified time is included.	query	*boolean* (default: "false")
enumerateContext	Token that helps obtain the next batch records. Do not specify this parameter for initial information extraction.	query	*string* (uuid)
count	The number of volume records to be obtained.	query	*integer* ,{ x ∈ ℤ | 1 ≤ x ≤ 32768 } (default: "32768")
vpsId	The ID of the virtual private storage (VPS) that the acquisition-target resource belongs to. To filter out the resources that do not belong to the VPS, specify "system".	query	*string* , must match /^system $\|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

Name	Description	Type	Data type
	To filter the resources by the VPS that the resources belong to, specify it in UUID format.		

REQUEST EXAMPLE

Example 1:

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/performances/volumes

HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 827
Date: Thu, 20 Oct 2022 05:39:52 GMT

{"data":[{"timestamp":"2022-10-20T05:39:11Z","performanceObjects":[{"id":"7e4eb895-
48cc-4f48-b228-9224af6aa4f6","vpsId":"54665c29-784a-4b71-9b61-2894d55697e9",
"readIOPS":0,"writeIOPS":0,"readTransferRate":0.0,"writeTransferRate":0.0,
"readResponseTime":0.0,"writeResponseTime":0.0}]},{"timestamp":"2022-10-20T05:39:15Z",
"performanceObjects":[{"id":"27f8944a-e98b-490d-b8fb-14583ceabc65","vpsId":"00000000-
0000-0000-0000-000000000000","readIOPS":0,"writeIOPS":0,"readTransferRate":0.0,
"writeTransferRate":0.0,"readResponseTime":0.0,"writeResponseTime":0.0},
{"id":"35d16c28-b033-4c00-9956-67e8221493d3","vpsId":"00000000-0000-0000-0000-
000000000000","readIOPS":0,"writeIOPS":0,"readTransferRate":0.0,
"writeTransferRate":0.0,"readResponseTime":0.0,"writeResponseTime":0.0}]}],
"totalCount":3,"hasNext":false,"enumerateContext":null}
```

Example 2:

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password, and specifying the acquisition period (startTime / endTime).

```
curl -i -u admin:password -X GET 'https://192.168.1.100/ConfigurationManager/
simple/v1/objects/performances/volumes?startTime=2022-03-24T04:56:00Z&endTime=2022-03-
24T04:58:59Z'
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 746
Date: Thu, 24 Mar 2022 05:06:13 GMT

{"data":[{"timestamp":"2022-03-24T04:56:44Z","performanceObjects":[{"id":"6c4f42da-
dc94-4b3c-992c-0a84c244f88f","readIOPS":0,"writeIOPS":0,"readTransferRate":0.0,
"writeTransferRate":0.0,"readResponseTime":0.0,"writeResponseTime":0.0}]},
{"timestamp":"2022-03-24T04:57:44Z","performanceObjects":[{"id":"6c4f42da-dc94-4b3c-
992c-0a84c244f88f","readIOPS":0,"writeIOPS":0,"readTransferRate":0.0,
"writeTransferRate":0.0,"readResponseTime":0.0,"writeResponseTime":0.0}]},
{"timestamp":"2022-03-24T04:58:44Z","performanceObjects":[{"id":"6c4f42da-dc94-4b3c-
992c-0a84c244f88f","readIOPS":0,"writeIOPS":0,"readTransferRate":0.0,
"writeTransferRate":0.0,"readResponseTime":0.0,"writeResponseTime":0.0}]}],
"totalCount":1,"hasNext":false,"enumerateContext":null}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of volume performance information (monitor information) at the specified time and information about performanceObjects.

PROPERTIES

data: *object[]*

A list of volume performance information (monitor information) at the specified time.

ITEMS

volumePerformanceListResponseData: object (on page 742)

totalCount: *integer*

The total number of volumes whose performance information exists within the specified time period.

hasNext: *boolean*

Indicates if some volumes remain to be listed in performanceObjects.

enumerateContext: *string* (uuid) nullable

A token used to obtain the next value of performanceObjects for the applicable information. Null is output if the information is the last one.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/performances/volumes/<*id*>

Required Role: Storage, Monitor, Resource, VpsStorage, or VpsMonitor

DESCRIPTION

Obtains the performance information (monitor information) about the specified volume.

Performance information is collected internally at one minute intervals. Because of this, the following problems might occur. In such cases, if you want to obtain the latest information, wait for another cycle, and then call the API again.

- Depending on the timing, the same information as the last time might be returned.
- If the storage cluster's time changes drastically, past information might be returned, or information might not be returned because it is not collected yet.
- Immediately after Virtual Storage Software block is started or a volume is added or deleted, performance information might not be returned because it is not collected yet.
- Some information might not be obtained or duplicate information might be collected in one cycle due to high load, including spikes.

If the information could not be obtained, any of the following is returned:

- If the information that could not be obtained is an array element, an array without the element is returned.
- If the information that could not be obtained is an attribute, null is returned.
- If no performance information, including the one that satisfies the query parameter specification conditions, is found or obtained, an empty data object, "data": [] is returned.

The following describes data to be output according to the specification of the startTime and endTime query parameters:

When both startTime and endTime are omitted

- It is assumed that the time when the REST API server received the request was specified as endTime.
- It is assumed that the time which is one minute earlier than the time when the REST API server received the request was specified as startTime. However, if the endTime is 1970-01-01T00:01:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When both startTime and endTime are specified

- The monitored information within the specified time period is returned.

When only startTime is specified

- It is assumed that the time when the REST API server received the request was specified as endTime.

When only endTime is specified

- It is assumed that the time which is one hour earlier than endTime was specified as startTime. However, if endTime is 1970-01-01T01:00:00Z or earlier, it is assumed that 1970-01-01T00:00:01Z is specified as startTime.

When the range specified by startTime and endTime is invalid

- HTTP status code 400 (Bad Request) is returned when any of the following conditions is satisfied:
  - A date and time later than endTime is specified as startTime, including when endTime is not specified.
  - A date and time which is earlier than 1970-01-01T00:00:01Z is specified for startTime or endTime.
  - When all the following conditions are satisfied:
    - Specify true for startTimeExcluding, or false for endTimeIncluding.
    - The same date and time is specified for startTime and endTime, including the case when startTime is not specified and 1970-01-01T00:00:01Z is specified for endTime.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Volume ID.	path	*string* (uuid) required
startTime	The start date and time of the monitor information to be obtained.	query	*string* (date-time)

Name	Description	Type	Data type
	If you specify startTimeExcluding, you can specify whether to include or exclude the specified time. By default, the specified time is included.		
startTimeExcluding	Determines whether to include information of the specified startTime. ▪ false (default): The information is included. ▪ true: The information is not included.	query	*boolean* (default: "false")
endTime	The end date and time of the monitor information to be obtained. If you specify endTimeIncluding, you can specify whether to include or exclude the specified time. By default, the specified time is excluded. If it is omitted, it is the request receipt time.	query	*string* (date-time)
endTimeIncluding	Specify whether to include the specified endTime. ▪ false (default): The specified time is not included. ▪ true: The specified time is included.	query	*boolean* (default: "false")

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/performances/volumes/7e4eb895-48cc-4f48-b228-9224af6aa4f6
```

```
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 284
Date: Thu, 20 Oct 2022 05:41:54 GMT

{"data":[{"timestamp":"2022-10-20T05:41:12Z","performanceObjects":[{"id":"7e4eb895-
48cc-4f48-b228-9224af6aa4f6","vpsId":"54665c29-784a-4b71-9b61-2894d55697e9",
"readIOPS":0,"writeIOPS":0,"readTransferRate":0.0,"writeTransferRate":0.0,
"readResponseTime":0.0,"writeResponseTime":0.0}]}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

volumePerformanceListResponse: object (on page 742)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

Chapter 13: Power management

POST /v1/objects/storage/actions/shutdown/invoke

Required Role: Service

DESCRIPTION

Stops the storage cluster. Restart can be specified in the request body.

You can run this API only for the cluster master node (primary) if you specify "true" for force. If this API is executed for any node other than a cluster master node (primary), HTTP status code 412 (Precondition Failed) is returned.

REQUEST BODY

Uses default content-types: application/json

shutdownStorageParam: object (on page 687)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Length: 0" https://192.168.1.100/
ConfigurationManager/simple/v1/objects/storage/actions/shutdown/invoke
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)

Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)

Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

Chapter 14: Secure communication management

POST /v1/objects/server-certificate/actions/import/invoke

Required Role: Security

DESCRIPTION

Imports the server certificate.

You can run this API only for the cluster master node (primary). If you call this API for a node other than the cluster master node (primary), HTTP status code 412 (Precondition Failed) is returned.

REQUEST BODY

multipart/form-data

REQUEST PARAMETERS

Name	Description	Type	Data type
serverCertificate	The server certificate file (public key) to be transferred to the storage cluster.	formData	*file* required
secretKey	The server certificate file (private key) to be transferred to the storage cluster.	formData	*file* required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: multipart/form-data" -H "Expect:"
-F serverCertificate=@server.crt -F secretKey=@secret.pem https://192.168.1.100/
ConfigurationManager/simple/v1/objects/server-certificate/actions/import/invoke
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)

Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)

Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 412 (on page 577)

Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)

Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/web-server-access-setting

Required Role: Security

DESCRIPTION

Obtains the web server access settings.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/web-server-access-setting
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
```

```
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 57
Date: Fri, 30 Jul 2021 07:56:28 GMT

{"whitelistSetting":{"isEnabled":false,"clientNames":[]}}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

webServerAccessSetting: object (on page 749)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

PATCH /v1/objects/web-server-access-setting

Required Role: Security

DESCRIPTION

Edits the web server access settings.

REQUEST BODY

Uses default content-types: application/json

patchWebServerAccessSettingParam: object (on page 653)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d '{ "whitelistSetting" : { "isEnabled": true, "clientNames":
[ "192.168.1.100" ] } }' https://192.168.1.100/ConfigurationManager/simple/v1/objects/
web-server-access-setting
```

The following shows an example using a json body.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d @example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/web-
server-access-setting
```

example.json is defined as follows:

```
{
    "whitelistSetting" :
        {
            "isEnabled": true,
            "clientNames":
                [
                    "192.168.1.100"
                ]
        }
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

Chapter 15: Session management

GET /v1/objects/sessions

Required Role: Security or VpsSecurity

DESCRIPTION

Obtains a list of session information the user created.

REQUEST PARAMETERS

Name	Description	Type	Data type
vpsId	The ID of the virtual private storage (VPS) that the acquisition-target resource belongs to. To filter out the resources that do not belong to the VPS, specify "system". To filter the resources by the VPS that the resources belong to, specify it in UUID format.	query	*string* , must match /^system$\|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/sessions
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
```

```
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 283
Date: Mon, 02 Aug 2021 11:21:59 GMT

{"data":[{"sessionId":"6ac8fe9f-1224-4b93-8d8d-2da0263fa67a","userId":"admin",
"userObjectId":"admin","expirationTime":"2021-08-03T11:10:51Z","createdTime":"2021-08-
02T11:10:51Z","lastAccessTime":"2021-08-02T11:10:51Z","roleNames":["Storage",
"Service","Audit","Security","Monitor"]}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of session information

PROPERTIES

data:*object[]*

ITEMS

session: object (on page 685)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/sessions

Required Role: None

DESCRIPTION

Generates a session.

REQUEST BODY

Uses default content-types: application/json

createSessionParam: object (on page 594)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Length: 0" https://192.168.1.100/
ConfigurationManager/simple/v1/objects/sessions
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

createdSession: object (on page 592)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)

Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)

Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication

DELETE /v1/objects/sessions/<*sessionId*>

Required Role: None

DESCRIPTION

Abandons a generated session.

If you have the Security or VpsSecurity role, you can specify the session ID of a resource in a virtual private storage (VPS) that you can access in the API. If you do not have the Security or VpsSecurity role, you can specify only your own session ID. If you specify a session ID that is not yours, HTTP status code 403 (Forbidden) is returned regardless of whether the specified session ID exists.

REQUEST PARAMETERS

Name	Description	Type	Data type
sessionId	Session ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -X DELETE -u admin:password https://192.168.1.100/ConfigurationManager/simple/v1/
objects/sessions/6ac8fe9f-1224-4b93-8d8d-2da0263fa67a
```

RESPONSES

Uses default content-types: application/json

HTTP status code 204 (on page 575)

Indicates that the request was processed correctly, but no response data to be returned exists.

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)

Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/sessions/<*sessionId*>

Required Role: None

DESCRIPTION

Obtains the session information.

If you have the Security or VpsSecurity role, you can specify the session ID of a resource in a virtual private storage (VPS) that you can access in the API. If you do not have the Security or VpsSecurity role, you can specify only your own session ID. If you specify a session ID that is not yours, HTTP status code 403 (Forbidden) is returned regardless of whether the specified session ID exists.

REQUEST PARAMETERS

Name	Description	Type	Data type
sessionId	Session ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/sessions/6ac8fe9f-1224-4b93-8d8d-2da0263fa67a
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
```

```
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 272
Date: Mon, 02 Aug 2021 11:10:56 GMT

{"sessionId":"6ac8fe9f-1224-4b93-8d8d-2da0263fa67a","userId":"admin",
"userObjectId":"admin","expirationTime":"2021-08-03T11:10:51Z","createdTime":"2021-08-
02T11:10:51Z","lastAccessTime":"2021-08-02T11:10:51Z","roleNames":["Storage",
"Service","Audit","Security","Monitor"]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

session: object (on page 685)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

Chapter 16: Snapshot management

GET /v1/objects/volumes/<*id*>/master-volume

Required Role: Security, Storage, Monitor, Service, Resource, VpsSecurity, VpsStorage, or VpsMonitor

DESCRIPTION

Obtains the P-VOL information used to create this volume when snapshotAttribute of the target volume is S-VOL or P/S VOL. If this API is called when snapshotAttribute is a hyphen (-) or P-VOL, HTTP status code 404 (Not Found) is returned.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Volume ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/volumes/27f8944a-e98b-490d-b8fb-14583ceabc65/master-volume
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 215
Date: Thu, 20 Oct 2022 05:33:44 GMT

{"masterVolumeId":"35d16c28-b033-4c00-9956-67e8221493d3","vpsId":"(system)",
"vpsName":"(system)","qosParam":{"upperLimitForIops":-1,"upperLimitForTransferRate":-
1,"upperAlertAllowableTime":-1,"upperAlertTime":null}}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

masterVolume: object (on page 637)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/volumes/<*id*>/snapshot-volumes

Required Role: Security, Storage, Monitor, Service, Resource, VpsSecurity, VpsStorage, or VpsMonitor

DESCRIPTION

Obtains a list of S-VOL information items created from this volume when snapshotAttribute of the target volume is P-VOL or P/S VOL.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Volume ID.	path	*string* (uuid) required
vpsId	The ID of the virtual private storage (VPS) that the acquisition-target resource belongs to. To filter out the resources that do not belong to the VPS, specify "system". To filter the resources by the VPS that the resources belong to, specify it in UUID format.	query	*string* , must match /^system $\|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/volumes/35d16c28-b033-4c00-9956-67e8221493d3/snapshot-volumes

HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 555
Date: Thu, 20 Oct 2022 05:34:37 GMT

{"data":[{"snapshotVolumeId":"27f8944a-e98b-490d-b8fb-14583ceabc65",
"snapshotVolumeName":"Volume7","snapshotVolumeNickname":"system_vol",
"statusSummary":"Normal","status":"Normal","snapshotStatus":"Prepared",
"snapshotProgressRate":null,"snapshotTimestamp":null,"snapshotType":"Snapshot",
"snapshotConcordanceRate":100,"isWrittenInSvol":null,"vpsId":"(system)",
"vpsName":"(system)","qosParam":{"upperLimitForIops":-1,"upperLimitForTransferRate":-
1,"upperAlertAllowableTime":-1,"upperAlertTime":null}}],"totalCount":1,
"hasNext":false,"enumerateContext":null}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

DESCRIPTION

A list of S-VOL information items and information about data.

PROPERTIES

data:*object[]*

A list of S-VOL information items.

ITEMS

snapshotVolume: object (on page 690)

totalCount:*integer*

The total number of records.

hasNext:*boolean*

Indicates whether there is information dropped from the data.

enumerateContext:*string* (uuid)nullable

A token used to obtain the remaining items for the next batch. Null is output if there are no more items to be obtained.

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/volumes/actions/create-snapshot/invoke

Required Role: Storage or VpsStorage

DESCRIPTION

Creates a Snapshot.

REQUEST BODY

Uses default content-types: application/json

createSnapshotParam: object (on page 594)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
'{ "masterVolumeId": "f77278ae-c86a-451d-b309-605fa63b5ff5" }' https://192.168.1.100/
ConfigurationManager/simple/v1/objects/volumes/actions/create-snapshot/invoke
```

The following shows an example using a json body.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
@example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/volumes/
actions/create-snapshot/invoke
```

example.json is defined as follows:

```
{
    "masterVolumeId": "f77278ae-c86a-451d-b309-605fa63b5ff5"
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)

Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)

Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)

Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/volumes/actions/delete-snapshot/invoke

Required Role: Storage or VpsStorage

DESCRIPTION

Deletes a snapshot. You can delete a volume other than P-VOL, S-VOL or P/S-VOL only by the API for deleting a volume, not by this API.

By running the API with the S-VOL specified for snapshotVolumeId, you can delete the S-VOL. If you want to delete P/S-VOLs, perform deletion after deleting all the S-VOLs and P/S-VOLs created from the P/S-VOLs to be deleted.

When the snapshot status of the S-VOL is "Error" and deletion is unsuccessful with the S-VOL specified for snapshotVolumeId, specify the P-VOL corresponding to the S-VOL for masterVolumeId and "true" for snapshotTree, and then perform the deletion. When performing the deletion, note the following:

- You should verify the S-VOLs and P/S-VOLs created from the specified P-VOL beforehand so that you can confirm that all the S-VOLs and P/S-VOLs have been deleted after performing the deletion. For how to confirm which are the S-VOLs and P/S-VOLs created from the P-VOL, see *Obtaining a list of S-VOL information for the volume* in the *Hitachi Virtual Storage Software Block Storage Administrator Guide*.
- If a system configuration change operation conflicts when the API is run, the job might be unsuccessful. In this case, retry the operation.

REQUEST BODY

Uses default content-types: application/json

deleteSnapshotParam: object (on page 604)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
'{ "snapshotVolumeId": "f8a757a0-afbd-4354-ab9a-19b8a2487080" }' https://
192.168.1.100/ConfigurationManager/simple/v1/objects/volumes/actions/delete-snapshot/
invoke
```

The following shows an example using a json body.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
@example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/volumes/
actions/delete-snapshot/invoke
```

example.json is defined as follows:

```
{
    "snapshotVolumeId": "f8a757a0-afbd-4354-ab9a-19b8a2487080"
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)
Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/volumes/actions/restore-snapshot/invoke

Required Role: Storage or VpsStorage

DESCRIPTION

Restores the volume from the snapshot.

Performing restoration changes the snapshot status of the restoration-source S-VOL to Prepared. The status changes to Restoring first, and then to Prepared.

REQUEST BODY

Uses default content-types: application/json

restoreSnapshotParam: object (on page 675)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
'{ "snapshotVolumeId": "deb8971d-3c7a-4147-adc6-e20a2504b3e8" }' https://
192.168.1.100/ConfigurationManager/simple/v1/objects/volumes/actions/restore-snapshot/
invoke
```

The following shows an example using a json body.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
@example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/volumes/
actions/restore-snapshot/invoke
```

example.json is defined as follows:

```
{
    "snapshotVolumeId": "deb8971d-3c7a-4147-adc6-e20a2504b3e8"
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)

Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)

Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)

Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

Chapter 17: Software update

POST /v1/objects/storage/actions/stop-software-update/invoke

Required Role: Service

DESCRIPTION

Requests the stop of updating the storage software.

When this API is executed, a job requesting the stop of updating the storage software is submitted. When the stop request is completed, the state of the job changes to Succeeded, and the state of the storage software update job to be stopped changes to Stopping. Depending on the timing of executing the API, it might take time until the state of the target storage software update job changes to Stopping.

When the target storage software update job could be stopped, the state of this job changes to Stopped. However, depending on the timing when the API is executed, the target storage software update job might be executed to the end without being stopped.

In this case, the state of the target storage software update job changes to Succeeded (instead of Stopped) when the storage software update is successful or Failed when it is not successful.

The storage software update job to be stopped can be found using the attribute, affectedResources in the response data of the API.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Length: 0" https://192.168.1.100/
ConfigurationManager/simple/v1/objects/storage/actions/stop-software-update/invoke
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)
Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/storage/actions/update-software/invoke

Required Role: Service

DESCRIPTION

Updates the storage software.

You can run this API only for the cluster master node (primary). If you call this API for a node other than the cluster master node (primary), HTTP status code 412 (Precondition Failed) is returned.

REQUEST BODY

Uses default content-types: application/json

updateSoftwareParam: object (on page 721)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
'{ "mode": "Non-disruptive" }' https://192.168.1.100/ConfigurationManager/simple/v1/
objects/storage/actions/update-software/invoke
```

The following shows an example using a json body.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
@example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/storage/
actions/update-software/invoke
```

example.json is defined as follows:

```
{
    "mode": "Non-disruptive"
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)
Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 412 (on page 577)
Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/storage/actions/upload-software-update-file/invoke

Required Role: Service

DESCRIPTION

Transfers (uploads) the update file of the storage software to the storage cluster.

You can run this API only for the cluster master node (primary). If you call this API for a node other than the cluster master node (primary), HTTP status code 412 (Precondition Failed) is returned.

REQUEST BODY

multipart/form-data

REQUEST PARAMETERS

Name	Description	Type	Data type
softwareUpdateFile	The update file of the storage software to be transferred to the storage cluster.	formData	*file* required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H  "content-type: multipart/form-data" -H
"Expect:" -F softwareUpdateFile=@/home/user/update/hsds-update-01050000-2345.tar
https://192.168.1.100/ConfigurationManager/simple/v1/objects/storage/actions/upload-
software-update-file/invoke
```

RESPONSES

Uses default content-types: application/json

HTTP status code 204 (on page 575)
Indicates that the request was processed correctly, but no response data to be returned exists.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)
Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 412 (on page 577)
Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/storage/software-update-file

Required Role: Service

DESCRIPTION

Obtains the information of the update file of the storage software which performed transfer (upload) in the storage cluster.

If the update file of the storage software is not transferred to the storage cluster, returns HTTP status code 404 (Not Found).

In the case of the file that an update file of the storage software transferred to the storage cluster is unauthorized, returns HTTP status code 409 (Conflict).

You can run this API only for the cluster master node (primary). If you call this API for a node other than the cluster master node (primary), HTTP status code 412 (Precondition Failed) is returned.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/storage/software-update-file
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 25
Date: Fri, 30 Jul 2021 08:47:45 GMT

{"version":"01.10.02.00"}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

softwareUpdateFile: object (on page 695)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 409 (on page 577)
Indicates that an inconsistent or impossible request was issued for the resource specified by URL (for example, creating a resource whose ID is same as the resource already created).

HTTP status code 412 (on page 577)
Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

Chapter 18: Storage cluster management

GET /v1/objects/fault-domains

Required Role: Security, Storage, Monitor, Service, Audit, or Resource

DESCRIPTION

Obtains a list of fault domain information.

REQUEST PARAMETERS

Name	Description	Type	Data type
protectionDomainId	Protection domain ID.	query	*string* (uuid)

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/fault-domains
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 148
Date: Fri, 30 Jul 2021 07:55:56 GMT

{"data":[{"id":"4f538215-e568-4cc0-8bcc-16bc0b71b386","name":"SC01-PD01-FD01",
"statusSummary":"Normal","status":"Normal","numberOfStorageNodes":6}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of fault domain information.

PROPERTIES

data:*object[]*

ITEMS

faultDomainSummary: object (on page 615)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/fault-domains/<*id*>

Required Role: Security, Storage, Monitor, Service, Audit, or Resource

DESCRIPTION

Obtains fault domain information.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Fault domain ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/fault-domains/4f538215-e568-4cc0-8bcc-16bc0b71b386
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
```

```
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 390
Date: Fri, 30 Jul 2021 07:56:38 GMT

{"storageNodeIds":["44f1d113-405e-448f-ad77-fd5554971c36","8fdbc008-ec6c-42b3-b930-
146c94c5cef9","9d36c162-e379-4c85-bcc2-ccf98fe774a6","9f29d742-9a75-4cef-9539-
35b3e76c8a20","a5b19d19-ca5d-4d19-988d-d625850c85de","dd5ed12b-cf39-47da-9a30-
5ba53404050e"],"id":"4f538215-e568-4cc0-8bcc-16bc0b71b386","name":"SC01-PD01-FD01",
"statusSummary":"Normal","status":"Normal","numberOfStorageNodes":6}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

faultDomain: object (on page 615)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/health-status

Required Role: Security, Storage, Monitor, Service, Audit, Resource

DESCRIPTION

Obtains the health status.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/health-status
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
```

```
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 688
Date: Fri, 30 Jul 2021 07:55:57 GMT

{"resources":[{"type":"Storage","status":"Normal","protectionDomainId":null},
{"type":"License","status":"Normal","protectionDomainId":null},{"type":"Drive",
"status":"Normal","protectionDomainId":"0778a123-42e5-43ff-8fbc-c8580b79f2cf"},
{"type":"Pool","status":"Normal","protectionDomainId":"0778a123-42e5-43ff-8fbc-
c8580b79f2cf"},{"type":"StorageNode","status":"Normal","protectionDomainId":"0778a123-
42e5-43ff-8fbc-c8580b79f2cf"},{"type":"Port","status":"Normal",
"protectionDomainId":"0778a123-42e5-43ff-8fbc-c8580b79f2cf"},{"type":"Volume",
"status":"Normal","protectionDomainId":"0778a123-42e5-43ff-8fbc-c8580b79f2cf"},
{"type":"FaultDomain","status":"Normal","protectionDomainId":null}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

healthStatus: object (on page 618)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

PATCH /v1/objects/snmp-setting

Required Role: Security

DESCRIPTION

Edits SNMP settings.

If you run this API without specifying already set values as parameters, settings of corresponding attributes in SNMP settings are deleted.

REQUEST BODY

Uses default content-types: application/json

patchSnmpSettingParam: object (on page 645)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d '{"isSNMPAgentEnabled":true, "snmpVersion":"v2c", "sendingTrapSetting":
{"snmpv2cSettings":[{"community":"public","sendTrapTo":["192.168.16.117"]},
{"community":"private","sendTrapTo":["192.168.13.232"]}, {"community":"public",
"sendTrapTo":["192.168.13.235"]}]}, "requestAuthenticationSetting":{"snmpv2cSettings":
[{"community":"public","requestsPermitted":["192.168.16.117"]},
{"community":"private","requestsPermitted":["192.168.13.232"]}, {"community":"public",
"requestsPermitted":["192.168.13.235"]}]}, "systemGroupInformation":
{"storageSystemName":"Hitachi VSSB","contact":"Hitachi Vantara LLC.",
"location":"HSP3F-Rack0320"}}' https://192.168.13.100/ConfigurationManager/simple/v1/
objects/snmp-setting
```

The following shows an example using a json body.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d @example.json https://192.168.13.100/ConfigurationManager/simple/v1/objects/snmp-
setting
```

example.json is defined as follows:

```
{
    "isSNMPAgentEnabled":true,
    "snmpVersion":"v2c",
    "sendingTrapSetting":
        {
            "snmpv2cSettings":
                [
                    {
                        "community":"public",
                        "sendTrapTo":["192.168.16.117"]
                    },
                    {
                        "community":"private",
                        "sendTrapTo":["192.168.13.232"]
                    },
                    {
                        "community":"public",
                        "sendTrapTo":["192.168.13.235"]
                    }
                ]
        },
    "requestAuthenticationSetting":
        {
            "snmpv2cSettings":
                [
                    {
                        "community":"public",
                        "requestsPermitted":["192.168.16.117"]
                    },
                    {
                        "community":"private",
                        "requestsPermitted":["192.168.13.232"]
                    },
                    {
                        "community":"public",
                        "requestsPermitted":["192.168.13.235"]
                    }
                ]
        },
    "systemGroupInformation":
        {
            "storageSystemName":"Hitachi VSSB",
            "contact":"Hitachi Vantara LLC.",
            "location":"HSP3F-Rack0320"
        }
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/protection-domains

Required Role: Security, Storage, Monitor, Service, Audit, or Resource

DESCRIPTION

Obtains a list of protection domain information.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/protection-domains
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 462
Date: Fri, 30 Jul 2021 07:55:57 GMT

{"data":[{"id":"0778a123-42e5-43ff-8fbc-c8580b79f2cf","name":"SC01-PD01",
"redundantPolicy":"HitachiPolyphaseErasureCoding","redundantType":"4D+1P",
"driveDataRelocationStatus":"Stopped","driveDataRelocationProgressRate":null,
"rebuildStatus":"Stopped","rebuildProgressRate":null,"memoryMode":"VolatileMemory",
"asyncProcessingResourceUsageRate":"High","numberOfFaultSets":1,
"storageControllerClusteringPolicy":"OneRedundantStorageNode",
"minimumMemorySize":131072}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of summary information about protection domain.

PROPERTIES

data: ***object[]***

ITEMS

protectionDomainSummary: object (on page 669)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/protection-domains/<*id*>

Required Role: Security, Storage, Monitor, Service, Audit, or Resource

DESCRIPTION

Obtains the protection domain information.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Protection domain ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/protection-domains/0778a123-42e5-43ff-8fbc-c8580b79f2cf
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 511
Date: Fri, 30 Jul 2021 07:56:38 GMT

{"totalPhysicalCapacity":1118208,"isFastRebuildEnabled":true,"id":"0778a123-42e5-43ff-
8fbc-c8580b79f2cf","name":"SC01-PD01",
"redundantPolicy":"HitachiPolyphaseErasureCoding","redundantType":"4D+1P",
"driveDataRelocationStatus":"Stopped","driveDataRelocationProgressRate":null,
"rebuildStatus":"Stopped","rebuildProgressRate":null,"memoryMode":"VolatileMemory",
"asyncProcessingResourceUsageRate":"High","numberOfFaultSets":1,
"storageControllerClusteringPolicy":"OneRedundantStorageNode",
"minimumMemorySize":131072}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

protectionDomain: object (on page 667)

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

PATCH /v1/objects/protection-domains/<*id*>

Required Role: Storage or Service

DESCRIPTION

Edits the protection domain settings.

REQUEST BODY

Uses default content-types: application/json

patchProtectionDomainParam: object (on page 644)

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Protection domain ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d '{ "asyncProcessingResourceUsageRate":"Middle" }' https://192.168.1.10/
ConfigurationManager/simple/v1/objects/protection-domains/c482915f-cb5f-4311-a66b-
bd1c9281cdfb
```

The following shows an example using a json body.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d @example.json https://192.168.1.10/ConfigurationManager/simple/v1/objects/
protection-domains/c482915f-cb5f-4311-a66b-bd1c9281cdfb
```

example.json is defined as follows:

```
{
    "asyncProcessingResourceUsageRate":"Middle"
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)

Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)

Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/protection-domains/<*id*>/actions/resume-drive-data-relocation/invoke

Required Role: Storage or Service

DESCRIPTION

Requests resumption of drive data relocation.

driveDataRelocationStatus of the target protection domain changes to "Stopped" if the drive data relocation conditions are not met when the state of the API job becomes "Succeeded".

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Protection domain ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Length: 0" https://192.168.1.100/
ConfigurationManager/simple/v1/objects/protection-domains/12345678-1234-1234-1234-
1234567891b/actions/resume-drive-data-relocation/invoke
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)
Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/protection-domains/<*id*>/actions/suspend-drive-data-relocation/invoke

Required Role: Storage or Service

DESCRIPTION

Requests suspension of drive data relocation.

When the suspension request is completed, driveDataRelocationStatus of the target protection domain changes to "Suspended" after job status becomes "Succeeded". Depending on the timing when the API is executed, it might take time until driveDataRelocationStatus of the target protection domain becomes "Suspended".

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Protection domain ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Length: 0" https://192.168.1.100/
ConfigurationManager/simple/v1/objects/protection-domains/12345678-1234-1234-1234-
1234567891b/actions/suspend-drive-data-relocation/invoke
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)

Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)

Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)

Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/snmp-setting

Required Role: Security

DESCRIPTION

Obtains the SNMP settings.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/snmp-setting
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 455
Date: Fri, 30 Jul 2021 07:55:58 GMT

{"isSNMPAgentEnabled":false,"snmpVersion":"v2c","sendingTrapSetting":
{"snmpv2cSettings":[{"community":"","sendTrapTo":[""]},{"community":"","sendTrapTo":
[""]},{"community":"","sendTrapTo":[""]}]},"requestAuthenticationSetting":
```

```
{"snmpv2cSettings":[{"community":"","requestsPermitted":[""]},{"community":"",
"requestsPermitted":[""]},{"community":"","requestsPermitted":[""]}]},
"systemGroupInformation":{"storageSystemName":"-","contact":"-","location":"-"}}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

snmpSetting: object (on page 693)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/storage

Required Role: Security, Storage, Monitor, Service, Audit, or Resource

DESCRIPTION

Obtains the storage cluster information.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/storage
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 966
Date: Tue, 28 Dec 2021 02:51:16 GMT

{"storageDeviceId":"400000019322","id":"4d17fde1-0cdb-4761-8d46-5cf5fb38c2ae",
"modelName":"VSSB","internalId":"019322","nickname":"SC01","numberOfTotalVolumes":1,
"numberOfTotalServers":1,"numberOfTotalStorageNodes":6,"numberOfReadyStorageNodes":6,
"numberOfFaultDomains":1,"totalPoolRawCapacity":1054032,
"totalPoolPhysicalCapacity":1118208,"totalPoolCapacity":275940,"usedPoolCapacity":0,
"freePoolCapacity":275940,"savingEffects":{"efficiencyDataReduction":null,
"preCapacityDataReduction":null,"postCapacityDataReduction":null,
"totalEfficiencyStatus":"NoTargetData",
"dataReductionWithoutSystemDataStatus":"NoTargetData","totalEfficiency":null,
"dataReductionWithoutSystemData":null,
"preCapacityDataReductionWithoutSystemData":null,
"postCapacityDataReductionWithoutSystemData":null,"calculationStartTime":null,
"calculationEndTime":null},"softwareVersion":"01.10.02.00","statusSummary":"Normal",
"status":"Ready","systemRequirementsFileVersion":20211019,"serviceId":null}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

storage: object (on page 696)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/bmc-root-certificate/actions/delete/invoke (Bare metal)

Required Role: Security

DESCRIPTION

Deletes the root certificate that is used for communication with the BMC of the storage node from the storage cluster.

You can run this API only for the cluster master node (primary).

If you run this API for a node other than the cluster master node (primary), HTTP status code 412 (Precondition Failed) is returned.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Length: 0" https://192.168.1.100/
ConfigurationManager/simple/v1/objects/bmc-root-certificate/actions/delete/invoke
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)
Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 412 (on page 577)
Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/bmc-root-certificate/actions/import/invoke (Bare metal)

Required Role: Security

DESCRIPTION

Imports the root certificate that is used for communication with the BMC of the storage node into the storage cluster. Certificate files in PEM and DER formats are supported.

You can run this API only for the cluster master node (primary).

If you run this API for a node other than the cluster master node (primary), HTTP status code 412 (Precondition Failed) is returned.

REQUEST BODY

multipart/form-data

REQUEST PARAMETERS

Name	Description	Type	Data type
rootCertificate	Root certificate file to be imported into the storage cluster and used for communication with the BMC of the storage node.	formData	*file* required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: multipart/form-data" -H "Expect:
" -F rootCertificate=@ca-crt.pem https://192.168.1.100/ConfigurationManager/simple/v1/
objects/bmc-root-certificate/actions/import/invoke
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)
Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 412 (on page 577)
Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/bmc-root-certificate/download (Bare metal)

Required Role: Security or Service

DESCRIPTION

Obtains the root certificate imported into the storage cluster for use in communication with the BMC of the storage node.

The obtained root certificate is output as a DER-format file.

Using the API with no root certificate imported returns HTTP status code 404 (Not Found).

You can run this API only for the cluster master node (primary).

If you run this API for a node other than the cluster master node (primary), HTTP status code 412 (Precondition Failed) is returned.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -J -O -X GET https://192.168.1.100/ConfigurationManager/
simple/v1/objects/bmc-root-certificate/download
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   865  100   865    0     0    685      0  0:00:01  0:00:01 --:--:--   685
curl: Saved to filename 'bmcRootCertificate.crt'
```

RESPONSES

Uses default content-types: application/x-x509-ca-cert, application/json

200 OK

Root certificate that is used for connection with the BMC.

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 412 (on page 577)

Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/spare-nodes (Bare metal)

Required Role: Service

DESCRIPTION

Obtains a list of spare node information.

REQUEST PARAMETERS

Name	Description	Type	Data type
faultDomainId	The ID of the fault domain that the nodes belong to.	query	*string* (uuid)

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/spare-nodes
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
```

```
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 478
Date: Fri, 07 Oct 2022 11:55:19 GMT

{"data":[{"id":"cbcd1d3c-39db-4ebf-8860-022b07f5be96","name":"node04",
"faultDomainId":"f1e47bcc-ebcc-4240-891b-ab8fde4a6187","faultDomainName":"SC01-PD01-
FD01","controlPortIpv4Address":"192.168.13.104","softwareVersion":"01.12.00.40.3691",
"biosUuid":"7d113042-dc1f-9cfe-ad91-959645761c9c","modelName":"HA8000V/DL360 Gen10",
"serialNumber":"SGH829X6MJ","bmcName":"10.164.118.13","bmcUser":"administrator@local,
10.164.119.79_CLWS1-SN02"}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

spareNodeList: object (on page 696)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 412 (on page 577)

Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/spare-nodes (Bare metal)

Required Role: Service

DESCRIPTION

Registers spare node information.

REQUEST BODY

Uses default content-types: application/json

createSpareNodeParam: object (on page 595)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
'{"faultDomainId":"b5b7c613-6323-4950-af06-0cdf39b0a4cd",
"controlPortIpv4Address":"192.168.13.104","setupUserPassword":"node1007",
"bmcName":"10.164.118.13","bmcUser":"administrator@local,10.164.119.79_CLWS1-SN02",
"bmcPassword":"bpassword"}' https://192.168.1.100/ConfigurationManager/simple/v1/
objects/spare-nodes
```

The following shows an example using a json body.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
@example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/spare-nodes
```

example.json is defined as follows:

```
{
    "faultDomainId":"b5b7c613-6323-4950-af06-0cdf39b0a4cd",
    "controlPortIpv4Address":"192.168.13.104",
    "setupUserPassword":"node1007",
    "bmcName":"10.164.118.13",
    "bmcUser":"administrator@local,10.164.119.79_CLWS1-SN02",
    "bmcPassword":"bpassword"
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 409 (on page 577)

Indicates that an inconsistent or impossible request was issued for the resource specified by URL (for example, creating a resource whose ID is same as the resource already created).

HTTP status code 411 (on page 577)

Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 412 (on page 577)

Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)

Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

DELETE /v1/objects/spare-nodes/<*id*> (Bare metal)

Required Role: Service

DESCRIPTION

Deletes spare node information.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of a spare node.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X DELETE -H "Content-Length: 0" https://192.168.1.100/
ConfigurationManager/simple/v1/objects/spare-nodes/cbcd1d3c-39db-4ebf-8860-
022b07f5be96
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 412 (on page 577)
Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/spare-nodes/<*id*> (Bare metal)

Required Role: Service

DESCRIPTION

Obtains information about a spare node.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of a spare node.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/spare-nodes/cbcd1d3c-39db-4ebf-8860-022b07f5be96
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 467
Date: Fri, 07 Oct 2022 12:15:10 GMT

{"id":"cbcd1d3c-39db-4ebf-8860-022b07f5be96","name":"node04",
"faultDomainId":"f1e47bcc-ebcc-4240-891b-ab8fde4a6187","faultDomainName":"SC01-PD01-
FD01","controlPortIpv4Address":"192.168.13.104","softwareVersion":"01.12.00.40.3691",
"biosUuid":"7d113042-dc1f-9cfe-ad91-959645761c9c","modelName":"HA8000V/DL360 Gen10",
"serialNumber":"SGH829X6MJ","bmcName":"10.164.118.13","bmcUser":"administrator@local,
10.164.119.79_CLWS1-SN02"}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

spareNode: object (on page 695)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 412 (on page 577)

Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

PATCH /v1/objects/spare-nodes/<*id*> (Bare metal)

Required Role: Service

DESCRIPTION

Allows you to edit information about a spare node.

REQUEST BODY

Uses default content-types: application/json

patchSpareNodeParam: object (on page 646)

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of a spare node.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d '{"faultDomainId":"f1e47bcc-ebcc-4240-891b-ab8fde4a6187",
"controlPortIpv4Address":"192.168.13.104","setupUserPassword":"node1007",
"bmcName":"10.164.118.13","bmcUser":"administrator@local,10.164.119.79_CLWS1-SN02",
"bmcPassword":"bpassword"}' https://192.168.1.100/ConfigurationManager/simple/v1/
objects/spare-nodes/cbcd1d3c-39db-4ebf-8860-022b07f5be96
```

The following shows an example using a json body.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d @example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/spare-
nodes/cbcd1d3c-39db-4ebf-8860-022b07f5be96
```

example.json is defined as follows:

```
{
    "faultDomainId":"f1e47bcc-ebcc-4240-891b-ab8fde4a6187",
    "controlPortIpv4Address":"192.168.13.104",
    "setupUserPassword":"node1007",
    "bmcName":"10.164.118.13",
    "bmcUser":"administrator@local,10.164.119.79_CLWS1-SN02",
    "bmcPassword":"bpassword"
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 412 (on page 577)
Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

Chapter 19: Storage controller management

GET /v1/objects/storage-controllers

Required Role: Security, Storage, Monitor, Service, or Resource

DESCRIPTION

Obtains a list of storage controller information.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/storage-controllers
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 5152
Date: Fri, 30 Jul 2021 07:56:15 GMT

{"data":[{"id":"061e07ae-d015-4c69-be25-e4205dcfd0e3","allocatableCapacity":45990,
"currentlyAllocatableCapacity":45990,"usedCapacity":0,"logicalLimit":10188318,
"volumeMaximumCapacity":10187646,"freeCapacity":45990,"status":"Normal",
"activeStorageNodeId":"44f1d113-405e-448f-ad77-fd5554971c36",
"standbyStorageNodeId":"9d36c162-e379-4c85-bcc2-ccf98fe774a6",
"secondaryStandbyStorageNodeId":null,"isDetailedLoggingMode":false,
"allocatableCapacityUsageRate":0,"currentlyAllocatableCapacityUsageRate":0,
"capacityStatus":"Normal","dataRebalanceStatus":"Stopped",
"dataRebalanceProgressRate":null,"totalVolumeCapacity":0,
"provisionedVolumeCapacity":0,"otherVolumeCapacity":0,"temporaryVolumeCapacity":0,
"capacitiesExcludingSystemData":{"usedVolumeCapacity":0,"compressedCapacity":0,
"reclaimedCapacity":0,"systemDataCapacity":0,"preUsedCapacity":0,
"preCompressedCapacity":0}},{"id":"1aab5a59-f06a-4edd-abfc-ec8d737eb4c5",
"allocatableCapacity":45990,"currentlyAllocatableCapacity":45990,"usedCapacity":0,
"logicalLimit":10188570,"volumeMaximumCapacity":10187898,"freeCapacity":45990,
"status":"Normal","activeStorageNodeId":"9d36c162-e379-4c85-bcc2-ccf98fe774a6",
"standbyStorageNodeId":"8fdbc008-ec6c-42b3-b930-146c94c5cef9",
"secondaryStandbyStorageNodeId":null,"isDetailedLoggingMode":false,
"allocatableCapacityUsageRate":0,"currentlyAllocatableCapacityUsageRate":0,
"capacityStatus":"Normal","dataRebalanceStatus":"Stopped",
"dataRebalanceProgressRate":null,"totalVolumeCapacity":0,
"provisionedVolumeCapacity":0,"otherVolumeCapacity":0,"temporaryVolumeCapacity":0,
"capacitiesExcludingSystemData":{"usedVolumeCapacity":0,"compressedCapacity":0,
"reclaimedCapacity":0,"systemDataCapacity":0,"preUsedCapacity":0,
"preCompressedCapacity":0}},{"id":"307c7147-79fe-4533-ad4d-9f5b9eaf98af",
"allocatableCapacity":45990,"currentlyAllocatableCapacity":45990,"usedCapacity":0,
"logicalLimit":10188570,"volumeMaximumCapacity":10187898,"freeCapacity":45990,
"status":"Normal","activeStorageNodeId":"9f29d742-9a75-4cef-9539-35b3e76c8a20",
"standbyStorageNodeId":"dd5ed12b-cf39-47da-9a30-5ba53404050e",
"secondaryStandbyStorageNodeId":null,"isDetailedLoggingMode":false,
"allocatableCapacityUsageRate":0,"currentlyAllocatableCapacityUsageRate":0,
"capacityStatus":"Normal","dataRebalanceStatus":"Stopped",
"dataRebalanceProgressRate":null,"totalVolumeCapacity":0,
"provisionedVolumeCapacity":0,"otherVolumeCapacity":0,"temporaryVolumeCapacity":0,
"capacitiesExcludingSystemData":{"usedVolumeCapacity":0,"compressedCapacity":0,
"reclaimedCapacity":0,"systemDataCapacity":0,"preUsedCapacity":0,
"preCompressedCapacity":0}},{"id":"474635bc-1280-4899-b52c-45e2cf08a86a",
"allocatableCapacity":45990,"currentlyAllocatableCapacity":45990,"usedCapacity":0,
"logicalLimit":10188570,"volumeMaximumCapacity":10187898,"freeCapacity":45990,
"status":"Normal","activeStorageNodeId":"dd5ed12b-cf39-47da-9a30-5ba53404050e",
"standbyStorageNodeId":"a5b19d19-ca5d-4d19-988d-d625850c85de",
"secondaryStandbyStorageNodeId":null,"isDetailedLoggingMode":false,
"allocatableCapacityUsageRate":0,"currentlyAllocatableCapacityUsageRate":0,
"capacityStatus":"Normal","dataRebalanceStatus":"Stopped",
"dataRebalanceProgressRate":null,"totalVolumeCapacity":0,
"provisionedVolumeCapacity":0,"otherVolumeCapacity":0,"temporaryVolumeCapacity":0,
"capacitiesExcludingSystemData":{"usedVolumeCapacity":0,"compressedCapacity":0,
"reclaimedCapacity":0,"systemDataCapacity":0,"preUsedCapacity":0,
"preCompressedCapacity":0}},{"id":"97a0c318-dedf-474f-8896-486b90a9ad3f",
"allocatableCapacity":45990,"currentlyAllocatableCapacity":45990,"usedCapacity":0,
"logicalLimit":10188570,"volumeMaximumCapacity":10187898,"freeCapacity":45990,
"status":"Normal","activeStorageNodeId":"8fdbc008-ec6c-42b3-b930-146c94c5cef9",
"standbyStorageNodeId":"9f29d742-9a75-4cef-9539-35b3e76c8a20",
"secondaryStandbyStorageNodeId":null,"isDetailedLoggingMode":false,
"allocatableCapacityUsageRate":0,"currentlyAllocatableCapacityUsageRate":0,
"capacityStatus":"Normal","dataRebalanceStatus":"Stopped",
"dataRebalanceProgressRate":null,"totalVolumeCapacity":0,
"provisionedVolumeCapacity":0,"otherVolumeCapacity":0,"temporaryVolumeCapacity":0,
"capacitiesExcludingSystemData":{"usedVolumeCapacity":0,"compressedCapacity":0,
"reclaimedCapacity":0,"systemDataCapacity":0,"preUsedCapacity":0,
"preCompressedCapacity":0}},{"id":"fa9116ec-de89-4034-bc6e-902028bc3d55",
"allocatableCapacity":45990,"currentlyAllocatableCapacity":45990,"usedCapacity":0,
"logicalLimit":10188570,"volumeMaximumCapacity":10187898,"freeCapacity":45990,
"status":"Normal","activeStorageNodeId":"a5b19d19-ca5d-4d19-988d-d625850c85de",
"standbyStorageNodeId":"44f1d113-405e-448f-ad77-fd5554971c36",
"secondaryStandbyStorageNodeId":null,"isDetailedLoggingMode":false,
"allocatableCapacityUsageRate":0,"currentlyAllocatableCapacityUsageRate":0,
"capacityStatus":"Normal","dataRebalanceStatus":"Stopped",
"dataRebalanceProgressRate":null,"totalVolumeCapacity":0,
"provisionedVolumeCapacity":0,"otherVolumeCapacity":0,"temporaryVolumeCapacity":0,
"capacitiesExcludingSystemData":{"usedVolumeCapacity":0,"compressedCapacity":0,
"reclaimedCapacity":0,"systemDataCapacity":0,"preUsedCapacity":0,
"preCompressedCapacity":0}}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

DESCRIPTION

A list of storage controller information.

PROPERTIES

data:*object[]*

ITEMS

storageController: object (on page 699)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/storage-controllers/actions/configure/invoke

Required Role: Service

DESCRIPTION

Edit the settings for the storage controller.

REQUEST BODY

Uses default content-types: application/json

configureStorageControllerParam: object (on page 587)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -H 'Content-Type: application/json' -d
'{"isDetailedLoggingMode":false}' -X POST https://192.168.1.100/ConfigurationManager/
simple/v1/objects/storage-controllers/actions/configure/invoke
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)

Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)

Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)

Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/storage-controllers/<*id*>

Required Role: Security, Storage, Monitor, Service, or Resource

DESCRIPTION

Obtains the storage controller information.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Storage controller ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/storage-controllers/fa9116ec-de89-4034-bc6e-902028bc3d55
HTTP/1.1 200 OK
```

```
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 856
Date: Fri, 30 Jul 2021 07:56:54 GMT

{"id":"fa9116ec-de89-4034-bc6e-902028bc3d55","allocatableCapacity":45990,
"currentlyAllocatableCapacity":45990,"usedCapacity":0,"logicalLimit":10188570,
"volumeMaximumCapacity":10187898,"freeCapacity":45990,"status":"Normal",
"activeStorageNodeId":"a5b19d19-ca5d-4d19-988d-d625850c85de",
"standbyStorageNodeId":"44f1d113-405e-448f-ad77-fd5554971c36",
"secondaryStandbyStorageNodeId":null,"isDetailedLoggingMode":false,
"allocatableCapacityUsageRate":0,"currentlyAllocatableCapacityUsageRate":0,
"capacityStatus":"Normal","dataRebalanceStatus":"Stopped",
"dataRebalanceProgressRate":null,"totalVolumeCapacity":0,
"provisionedVolumeCapacity":0,"otherVolumeCapacity":0,"temporaryVolumeCapacity":0,
"capacitiesExcludingSystemData":{"usedVolumeCapacity":0,"compressedCapacity":0,
"reclaimedCapacity":0,"systemDataCapacity":0,"preUsedCapacity":0,
"preCompressedCapacity":0}}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

storageController: object (on page 699)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

Chapter 20: Storage node management

POST /v1/objects/storage/actions/stop-removing-storage-nodes/invoke

Required Role: Service

DESCRIPTION

Requests storage node removal to stop. When this API is executed, the job that requests storage node removal is performed. When the stop request is completed, the state of this job becomes "Succeeded", and the state of the storage node removal job to be stopped becomes "Stopping". Depending on the timing when the API is executed, it might take time until the state of the target storage node removal job becomes "Stopping".

When the target storage node removal job is stopped successfully, the state of the job becomes "Stopped". Depending on the timing when the API is executed, the target storage node removal job might not be stopped and might be executed to the end. In this case, the state of the target storage node removal job does not become "Stopped", and instead becomes "Succeeded" if storage node removal is successful, or "Failed" if removal did not succeed.

You can verify the storage node removal job to be stopped in attribute "affectedResources" in the response data for the API.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Length: 0" https://192.168.1.100/
ConfigurationManager/simple/v1/objects/storage/actions/stop-removing-storage-nodes/
invoke
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)
Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/storage-nodes

Required Role: Security, Storage, Monitor, Service, Audit, or Resource

DESCRIPTION

Obtains a list of information about storage nodes.

REQUEST PARAMETERS

Name	Description	Type	Data type
faultDomainId	Fault domain ID.	query	*string* (uuid)
name	Storage node name (partial match).	query	*string*
clusterRole	The role of a storage node in a storage cluster.	query	*string* , x ∈ { "Master" , "Worker" }
protectionDomainId	Protection domain ID.	query	*string* (uuid)

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/storage-nodes
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
```

```
Content-Length: 4312
Date: Fri, 30 Jul 2021 07:55:59 GMT

{"data":[{"id":"44f1d113-405e-448f-ad77-fd5554971c36","biosUuid":"423db8d0-c54b-4d81-
eac9-fccc6f05a965","protectionDomainId":"0778a123-42e5-43ff-8fbc-c8580b79f2cf",
"faultSetId":"4f538215-e568-4cc0-8bcc-16bc0b71b386","faultSetName":"SC01-PD01-FS01",
"name":"PTFM-ENV21-SN04","clusterRole":"Worker","storageNodeAttributes":[],
"statusSummary":"Normal","status":"Ready","driveDataRelocationStatus":"Stopped",
"controlPortIpv4Address":"192.168.1.214","internodePortIpv4Address":"192.168.73.214",
"softwareVersion":"01.10.02.00","modelName":"HA8000V/DL360 Gen10",
"serialNumber":"SGH829X4XQ","memory":131072,"insufficientResourcesForRebuildCapacity":
{"capacityOfDrive":0,"numberOfDrives":0},"rebuildableResources":{"numberOfDrives":1}},
{"id":"8fdbc008-ec6c-42b3-b930-146c94c5cef9","biosUuid":"423d1d1c-e470-c10c-c1e2-
0b6ab8f8b19e","protectionDomainId":"0778a123-42e5-43ff-8fbc-c8580b79f2cf",
"faultSetId":"4f538215-e568-4cc0-8bcc-16bc0b71b386","faultSetName":"SC01-PD01-FS01",
"name":"PTFM-ENV21-SN06","clusterRole":"Worker","storageNodeAttributes":[],
"statusSummary":"Normal","status":"Ready","driveDataRelocationStatus":"Stopped",
"controlPortIpv4Address":"192.168.1.216","internodePortIpv4Address":"192.168.73.216",
"softwareVersion":"01.10.02.00","modelName":"HA8000V/DL360 Gen10",
"serialNumber":"SGH829X4Y4","memory":131072,"insufficientResourcesForRebuildCapacity":
{"capacityOfDrive":0,"numberOfDrives":0},"rebuildableResources":{"numberOfDrives":1}},
{"id":"9d36c162-e379-4c85-bcc2-ccf98fe774a6","biosUuid":"423d3f78-56e6-194e-0d0f-
2f0507902b53","protectionDomainId":"0778a123-42e5-43ff-8fbc-c8580b79f2cf",
"faultSetId":"4f538215-e568-4cc0-8bcc-16bc0b71b386","faultSetName":"SC01-PD01-FS01",
"name":"PTFM-ENV21-SN05","clusterRole":"Worker","storageNodeAttributes":[],
"statusSummary":"Normal","status":"Ready","driveDataRelocationStatus":"Stopped",
"controlPortIpv4Address":"192.168.1.215","internodePortIpv4Address":"192.168.73.215",
"softwareVersion":"01.10.02.00","modelName":"HA8000V/DL360 Gen10",
"serialNumber":"SGH829X4XN","memory":131072,"insufficientResourcesForRebuildCapacity":
{"capacityOfDrive":0,"numberOfDrives":0},"rebuildableResources":{"numberOfDrives":1}},
{"id":"9f29d742-9a75-4cef-9539-35b3e76c8a20","biosUuid":"423d1534-ee45-f92d-f3b0-
662595e47bd7","protectionDomainId":"0778a123-42e5-43ff-8fbc-c8580b79f2cf",
"faultSetId":"4f538215-e568-4cc0-8bcc-16bc0b71b386","faultSetName":"SC01-PD01-FS01",
"name":"PTFM-ENV21-SN01","clusterRole":"Master","storageNodeAttributes":[],
"statusSummary":"Normal","status":"Ready","driveDataRelocationStatus":"Stopped",
"controlPortIpv4Address":"192.168.1.211","internodePortIpv4Address":"192.168.73.211",
"softwareVersion":"01.10.02.00","modelName":"HA8000V/DL360 Gen10",
"serialNumber":"SGH829X6MJ","memory":131072,"insufficientResourcesForRebuildCapacity":
{"capacityOfDrive":0,"numberOfDrives":0},"rebuildableResources":{"numberOfDrives":1}},
{"id":"a5b19d19-ca5d-4d19-988d-d625850c85de","biosUuid":"423d0e42-fcec-5725-a6f9-
165cc98907da","protectionDomainId":"0778a123-42e5-43ff-8fbc-c8580b79f2cf",
"faultSetId":"4f538215-e568-4cc0-8bcc-16bc0b71b386","faultSetName":"SC01-PD01-FS01",
"name":"PTFM-ENV21-SN03","clusterRole":"Master","storageNodeAttributes":[],
"statusSummary":"Normal","status":"Ready","driveDataRelocationStatus":"Stopped",
"controlPortIpv4Address":"192.168.1.213","internodePortIpv4Address":"192.168.73.213",
"softwareVersion":"01.10.02.00","modelName":"HA8000V/DL360 Gen10",
"serialNumber":"SGH829X4XC","memory":131072,"insufficientResourcesForRebuildCapacity":
{"capacityOfDrive":0,"numberOfDrives":0},"rebuildableResources":{"numberOfDrives":1}},
{"id":"dd5ed12b-cf39-47da-9a30-5ba53404050e","biosUuid":"423d271b-2866-16ca-62f5-
578f7f8d9b30","protectionDomainId":"0778a123-42e5-43ff-8fbc-c8580b79f2cf",
"faultSetId":"4f538215-e568-4cc0-8bcc-16bc0b71b386","faultSetName":"SC01-PD01-FS01",
"name":"PTFM-ENV21-SN02","clusterRole":"Master","storageNodeAttributes":[],
"statusSummary":"Normal","status":"Ready","driveDataRelocationStatus":"Stopped",
"controlPortIpv4Address":"192.168.1.212","internodePortIpv4Address":"192.168.73.212",
"softwareVersion":"01.10.02.00","modelName":"HA8000V/DL360 Gen10",
"serialNumber":"SGH829X4X9","memory":131072,"insufficientResourcesForRebuildCapacity":
{"capacityOfDrive":0,"numberOfDrives":0},"rebuildableResources":
{"numberOfDrives":1}}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of storage node information.

PROPERTIES

data: ***object[]***

ITEMS

storageNode: object (on page 703)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

DELETE /v1/objects/storage-nodes/<*id*>

Required Role: Service

DESCRIPTION

Deletes a storage node.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Storage node ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -X DELETE -u admin:password https://192.168.1.100/ConfigurationManager/simple/v1/
objects/storage-nodes/8fdbc008-ec6c-42b3-b930-146c94c5cef9
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/storage-nodes/<*id*>

Required Role: Security, Storage, Monitor, Service, Audit, or Resource

DESCRIPTION

Obtains the information about storage nodes.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Storage node ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/storage-nodes/dd5ed12b-cf39-47da-9a30-5ba53404050e
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 716
Date: Fri, 30 Jul 2021 07:56:39 GMT

{"id":"dd5ed12b-cf39-47da-9a30-5ba53404050e","biosUuid":"423d271b-2866-16ca-62f5-
578f7f8d9b30","protectionDomainId":"0778a123-42e5-43ff-8fbc-c8580b79f2cf",
"faultSetId":"4f538215-e568-4cc0-8bcc-16bc0b71b386","faultSetName":"SC01-PD01-FS01",
"name":"PTFM-ENV21-SN02","clusterRole":"Master","storageNodeAttributes":[],
"statusSummary":"Normal","status":"Ready","driveDataRelocationStatus":"Stopped",
"controlPortIpv4Address":"192.168.1.212","internodePortIpv4Address":"192.168.73.212",
"softwareVersion":"01.10.02.00","modelName":"HA8000V/DL360 Gen10",
"serialNumber":"SGH829X4X9","memory":131072,"insufficientResourcesForRebuildCapacity":
{"capacityOfDrive":0,"numberOfDrives":0},"rebuildableResources":{"numberOfDrives":1}}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

storageNode: object (on page 703)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/storage-nodes/<*id*>/actions/block-for-maintenance/invoke

Required Role: Service

DESCRIPTION

Places the storage node into the blockade state for maintenance operation.

REQUEST BODY

Uses default content-types: application/json

blockStorageNodeForMaintenance: object (on page 584)

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Storage node ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Length: 0" https://192.168.1.100/
ConfigurationManager/simple/v1/objects/storage-nodes/9014efa1-3a11-4687-8be4-
2d28d43200c9/actions/block-for-maintenance/invoke
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)
Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/storage-nodes/<*id*>/actions/recover/invoke

Required Role: Service

DESCRIPTION

Restores the storage node from the blockade state caused by maintenance operation.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Storage node ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Length: 0" https://192.168.1.100/
ConfigurationManager/simple/v1/objects/storage-nodes/7ba34e51-327e-4f28-b56b-
1e9930cc481e/actions/recover/invoke
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)
Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/storage-nodes/<*id*>/actions/replace/invoke (Bare metal)

Required Role: Service

DESCRIPTION

Replaces a storage node.

REQUEST BODY

Uses default content-types: application/json

ReplaceStorageNodeParam: object (on page 672)

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Storage node ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
'{"setupUserPassword": "hitachino1"}' https://192.168.1.100/ConfigurationManager/
simple/v1/objects/storage-nodes/021b437b-ce44-4a3c-b9d6-ef3ce9e0e406/actions/replace/
invoke
```

The following shows an example using a json body.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
@example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/storage-
nodes/021b437b-ce44-4a3c-b9d6-ef3ce9e0e406/actions/replace/invoke
```

example.json is defined as follows:

```
{
    "setupUserPassword":"hitachino1"
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)
Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 412 (on page 577)
Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/storage-nodes (Bare metal)

Required Role: Service

DESCRIPTION

Adds a storage node.

You can run this API only for the cluster master node (primary).

If you run this API for a node other than the cluster master node (primary), HTTP status code 412 (Precondition Failed) is returned.

REQUEST BODY

multipart/form-data

REQUEST PARAMETERS

Name	Description	Type	Data type
configurationFile	The configuration definition file to be transferred to the storage cluster.	formData	*file* required
setupUserPassword	Setup user password.	formData	*string* (1 to 256 chars) , must match /^[\-A-Za-z0-9!#\$%&'"\(\)*\+,\.\/:;<>=\?@\[\]\\\^_`\{\}|~]{1,256}$/ required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: multipart/form-data" -H "Expect:
" -F "setupUserPassword=hitachino1" -F configurationFile=@SystemConfigurationFile.csv
https://192.168.1.100/ConfigurationManager/simple/v1/objects/storage-nodes
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)
Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 412 (on page 577)
Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/storage-node-bmc-access-settings (Bare metal)

Required Role: Service

DESCRIPTION

Obtains a list of BMC connection information about storage nodes.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/storage-node-bmc-access-settings
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 410
Date: Fri, 07 Oct 2022 11:55:19 GMT

{"data":[{"id":"6ca229e4-4a60-45a3-aaa4-fa2ae5ed71cf","bmcName":"10.164.118.96",
"bmcUser":"administrator@local,10.164.118.96-ptfm-endo_SN03"},{"id":"9323fe40-5cad-
4cf6-b8a5-fdf6a0cb69b9","bmcName":"10.164.118.96","bmcUser":"administrator@local,
10.164.118.96-ptfm-endo_SN02"},{"id":"ad4f2244-0fac-45c6-af8e-15f2df52984f",
"bmcName":"10.164.118.96","bmcUser":"administrator@local,10.164.118.96-ptfm-
endo_SN01"}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

storageNodeBmcAccessSettingList: object (on page 707)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 412 (on page 577)
Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/storage-node-bmc-access-settings/<*id*> (Bare metal)

Required Role: Service

DESCRIPTION

Obtains the BMC connection information about a storage node.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Storage node ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/storage-node-bmc-access-settings/ad4f2244-0fac-45c6-af8e-15f2df52984f
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 133
Date: Fri, 07 Oct 2022 11:56:25 GMT

{"id":"ad4f2244-0fac-45c6-af8e-15f2df52984f","bmcName":"10.164.118.96",
"bmcUser":"administrator@local,10.164.118.96-ptfm-endo_SN01"}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

storageNodeBmcAccessSetting: object (on page 706)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 412 (on page 577)
Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

PATCH /v1/objects/storage-node-bmc-access-settings/<*id*> (Bare metal)

Required Role: Service

DESCRIPTION

Allows you to edit BMC connection information about a storage node.

REQUEST BODY

Uses default content-types: application/json

patchStorageNodeBmcAccessSettingParam: object (on page 647)

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Storage node ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d '{ "bmcName":"10.1.12.123", "bmcUser":"username", "bmcPassword":"password" }'
https://192.168.1.100/ConfigurationManager/simple/v1/objects/storage-node-bmc-access-
settings/ad4f2244-0fac-45c6-af8e-15f2df52984f
```

The following shows an example using a json body.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d @example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/storage-
node-bmc-access-settings/ad4f2244-0fac-45c6-af8e-15f2df52984f
```

example.json is defined as follows:

```
{
    "bmcName":"10.1.12.123",
    "bmcUser":"username",
    "bmcPassword":"password"
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 412 (on page 577)
Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

Chapter 21: Storage pool management

GET /v1/objects/pools

Required Role: Security, Storage, Monitor, Service, or Resource

DESCRIPTION

Obtains a list of storage pool information.

REQUEST PARAMETERS

Name	Description	Type	Data type
name	Storage pool name (exact match).	query	*string*
names	A list of the names of the storage pools (exact match). You can specify plural names up to 32 with comma (,) delimiter. Example: names=<*name1*>,<*name2*>,<*name3*>… If you enter a space character after a comma, the space character is recognized as a part of the name specified after the comma. Also, if you specify 33 or more names, HTTP status code 400 (Bad Request) is returned.	query	*string* , Each item must match /^[\-A-Za-z0-9_]{1,32}$/

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/pools
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 1453
Date: Fri, 30 Jul 2021 07:56:00 GMT

{"data":[{"id":"376ae3cc-70ce-4ea2-9d37-d64311576739","name":"SP01",
"protectionDomainId":"0778a123-42e5-43ff-8fbc-c8580b79f2cf","statusSummary":"Normal",
"status":"Normal","totalCapacity":275940,"totalRawCapacity":1054032,"usedCapacity":0,
"freeCapacity":275940,"totalPhysicalCapacity":1118208,
"metaDataPhysicalCapacity":64176,"reservedPhysicalCapacity":677592,
"usablePhysicalCapacity":376440,"blockedPhysicalCapacity":0,"capacityManage":
{"usedCapacityRate":0,"maximumReserveRate":-1,"thresholdWarning":70,
"thresholdDepletion":80,"thresholdStorageControllerDepletion":95},"savingEffects":
{"efficiencyDataReduction":null,"preCapacityDataReduction":null,
"postCapacityDataReduction":null,"totalEfficiencyStatus":"NoTargetData",
"dataReductionWithoutSystemDataStatus":"NoTargetData","totalEfficiency":null,
"dataReductionWithoutSystemData":null,
"preCapacityDataReductionWithoutSystemData":null,
"postCapacityDataReductionWithoutSystemData":null,"calculationStartTime":null,
"calculationEndTime":null},"numberOfVolumes":1,
"redundantPolicy":"HitachiPolyphaseErasureCoding","redundantType":"4D+1P",
"dataRedundancy":1,"storageControllerCapacitiesGeneralStatus":"Normal",
"totalVolumeCapacity":0,"provisionedVolumeCapacity":0,"otherVolumeCapacity":0,
"temporaryVolumeCapacity":0,"rebuildCapacityPolicy":"Variable",
"rebuildCapacityResourceSetting":{"numberOfTolerableDriveFailures":null},
"rebuildCapacityStatus":"Sufficient","rebuildableResources":{"numberOfDrives":1}}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of storage pool information.

PROPERTIES

data: *object[]*
ITEMS

pool: object (on page 655)

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/pools/<*id*>

Required Role: Security, Storage, Monitor, Service, or Resource

DESCRIPTION

Obtains the storage pool information.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the storage pool.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/pools/376ae3cc-70ce-4ea2-9d37-d64311576739
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 1442
Date: Tue, 28 Dec 2021 02:51:16 GMT

{"id":"376ae3cc-70ce-4ea2-9d37-d64311576739","name":"SP01",
"protectionDomainId":"0778a123-42e5-43ff-8fbc-c8580b79f2cf","statusSummary":"Normal",
"status":"Normal","totalCapacity":275940,"totalRawCapacity":1054032,"usedCapacity":0,
"freeCapacity":275940,"totalPhysicalCapacity":1118208,
"metaDataPhysicalCapacity":64176,"reservedPhysicalCapacity":677592,
"usablePhysicalCapacity":376440,"blockedPhysicalCapacity":0,"capacityManage":
{"usedCapacityRate":0,"maximumReserveRate":-1,"thresholdWarning":70,
"thresholdDepletion":80,"thresholdStorageControllerDepletion":95},"savingEffects":
{"efficiencyDataReduction":null,"preCapacityDataReduction":null,
"postCapacityDataReduction":null,"totalEfficiencyStatus":"NoTargetData",
"dataReductionWithoutSystemDataStatus":"NoTargetData","totalEfficiency":null,
"dataReductionWithoutSystemData":null,
"preCapacityDataReductionWithoutSystemData":null,
"postCapacityDataReductionWithoutSystemData":null,"calculationStartTime":null,
"calculationEndTime":null},"numberOfVolumes":1,
"redundantPolicy":"HitachiPolyphaseErasureCoding","redundantType":"4D+1P",
"dataRedundancy":1,"storageControllerCapacitiesGeneralStatus":"Normal",
"totalVolumeCapacity":0,"provisionedVolumeCapacity":0,"otherVolumeCapacity":0,
"temporaryVolumeCapacity":0,"rebuildCapacityPolicy":"Variable",
"rebuildCapacityResourceSetting":{"numberOfTolerableDriveFailures":null},
"rebuildCapacityStatus":"Sufficient","rebuildableResources":{"numberOfDrives":1}}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

pool: object (on page 655)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

PATCH /v1/objects/pools/<*id*>

Required Role: Storage

DESCRIPTION

Edit storage pool settings.

REQUEST BODY

Uses default content-types: application/json

patchPoolSettingParam: object (on page 648)

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Storage pool ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -d
'{"rebuildCapacityPolicy": "Fixed", "rebuildCapacityResourceSetting":
{"numberOfTolerableDriveFailures":3}}' https://192.168.1.100/ConfigurationManager/
simple/v1/objects/pools/7f294d33-5f9f-47ad-8a74-be7f251a8c54
```

The following shows an example using a json body.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -d
@example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/pools/
7f294d33-5f9f-47ad-8a74-be7f251a8c54
```

example.json is defined as follows:

```
{
    "rebuildCapacityPolicy": "Fixed",
    "rebuildCapacityResourceSetting":
        {
            "numberOfTolerableDriveFailures":3
        }
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/pools/<*id*>/actions/expand/invoke

Required Role: Storage

DESCRIPTION

Expands the storage pool capacity.

REQUEST BODY

Uses default content-types: application/json

expandPoolParam: object (on page 613)

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the storage pool.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
'{  "driveIds": [  "0437c9f8-ec5a-4527-900b-300519321f1d","cbf7b144-593e-451d-9a49-
d62e6b7e1334"  ]}' https://192.168.1.100/ConfigurationManager/simple/v1/objects/pools/
12345678-9abc-4def-8123-456789abcdef/actions/expand/invoke
```

The following shows an example using a json body.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
@example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/pools/
12345678-9abc-4def-8123-456789abcdef/actions/expand/invoke
```

example.json is defined as follows:

```
{
    "driveIds":
        [
            "0437c9f8-ec5a-4527-900b-300519321f1d",
            "cbf7b144-593e-451d-9a49-d62e6b7e1334"
        ]
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)
Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/storage-node-capacity-settings

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains a list of capacity management settings of a storage node.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/storage-node-capacity-settings
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 562
Date: Fri, 30 Jul 2021 07:56:00 GMT

{"data":[{"id":"44f1d113-405e-448f-ad77-fd5554971c36","capacityBalancingSetting":
{"isEnabled":true}},{"id":"8fdbc008-ec6c-42b3-b930-146c94c5cef9",
"capacityBalancingSetting":{"isEnabled":true}},{"id":"9d36c162-e379-4c85-bcc2-
ccf98fe774a6","capacityBalancingSetting":{"isEnabled":true}},{"id":"9f29d742-9a75-
4cef-9539-35b3e76c8a20","capacityBalancingSetting":{"isEnabled":true}},
{"id":"a5b19d19-ca5d-4d19-988d-d625850c85de","capacityBalancingSetting":
{"isEnabled":true}},{"id":"dd5ed12b-cf39-47da-9a30-5ba53404050e",
"capacityBalancingSetting":{"isEnabled":true}}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of capacity management settings of a storage node

PROPERTIES

data:*object[]*

ITEMS

storageNodeCapacitySetting: object (on page 707)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/storage-node-capacity-settings/<*id*>

Required Role: Storage, Monitor, or Resource

DESCRIPTION

Obtains the capacity management settings of a storage node.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Storage node ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/storage-node-capacity-settings/dd5ed12b-cf39-47da-9a30-5ba53404050e
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
```

```
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 91
Date: Fri, 30 Jul 2021 07:56:40 GMT

{"id":"dd5ed12b-cf39-47da-9a30-5ba53404050e","capacityBalancingSetting":
{"isEnabled":true}}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

storageNodeCapacitySetting: object (on page 707)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

PATCH /v1/objects/storage-node-capacity-settings/<*id*>

Required Role: Storage

DESCRIPTION

Edits the capacity management settings of a storage node.

REQUEST BODY

Uses default content-types: application/json

patchStorageNodeCapacitySettingParam: object (on page 647)

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Storage node ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d '{ "capacityBalancingSetting" :{ "isEnabled": false }}' https://192.168.1.100/
ConfigurationManager/simple/v1/objects/storage-node-capacity-settings/7ba34e51-327e-
4f28-b56b-1e9930cc481e
```

The following shows an example using a json body.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d @example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/storage-
node-capacity-settings/7ba34e51-327e-4f28-b56b-1e9930cc481e
```

example.json is defined as follows:

```
{
    "capacityBalancingSetting" :
        {
            "isEnabled": false
        }
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)
Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

Chapter 22: System management

GET /configuration/storage-master-node-primary-flag

Required Role: None

DESCRIPTION

Obtains the information indicating whether the storage node that received the API request is the cluster master node (primary). For the cluster master node (primary), HTTP status code 200 (OK) is returned. For other nodes, HTTP status code 404 (Not Found) is returned.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/
configuration/storage-master-node-primary-flag
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 35
Date: Fri, 30 Jul 2021 07:56:18 GMT

{"isStorageMasterNodePrimary":true}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

storageMasterNodePrimaryFlag: object (on page 702)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

GET /v1/objects/control-ports

Required Role: Security, Storage, Monitor, Service, Audit, or Resource

DESCRIPTION

Obtains a list of control port information.

REQUEST PARAMETERS

Name	Description	Type	Data type
storageNodeId	Storage node ID.	query	*string* (uuid)

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/control-ports
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 1498
Date: Fri, 30 Jul 2021 07:56:19 GMT

{"data":[{"id":"1d7fa4ae-9bc6-446c-8cf1-09de0ea01dcd","storageNodeId":"ad36eab7-20c3-
41cd-acac-d334ff5347d7","macAddress":"80:30:e0:39:b7:a5","mtuSize":1500,
"interfaceName":"eth0","deviceName":"NetXtreme BCM5719 Gigabit Ethernet PCIe
(Ethernet 1Gb 4-port 331i Adapter)","configuredPortSpeed":"Auto",
"portSpeedDuplex":"1Gbps Full","isTeamingEnabled":"false","ipv4Information":
{"address":"192.168.24.242","subnetMask":"255.255.192.0"},"teaming":null,
"redundancy":0,"status":"Normal","statusSummary":"Normal"},{"id":"74d48fce-bd5d-4ed9-
ab64-f7c86e213cde","storageNodeId":"3123caa9-f13a-4905-a6ee-815ffb7f98a2",
"macAddress":"80:30:e0:39:a7:1d","mtuSize":1500,"interfaceName":"eth0",
"deviceName":"NetXtreme BCM5719 Gigabit Ethernet PCIe (Ethernet 1Gb 4-port 331i
Adapter)","configuredPortSpeed":"Auto","portSpeedDuplex":"1Gbps Full",
"isTeamingEnabled":"false","ipv4Information":{"address":"192.168.24.243",
"subnetMask":"255.255.192.0"},"teaming":null,"redundancy":0,"status":"Normal",
"statusSummary":"Normal"},{"id":"bd4e17e1-57ba-4a16-bff8-b1d8923b8662",
"storageNodeId":"8e01e189-38bc-4002-a1fe-7a73633bce8c",
"macAddress":"80:30:e0:39:08:3d","mtuSize":1500,"interfaceName":"eth0",
"deviceName":"NetXtreme BCM5719 Gigabit Ethernet PCIe (Ethernet 1Gb 4-port 331i
Adapter)","configuredPortSpeed":"Auto","portSpeedDuplex":"1Gbps Full",
"isTeamingEnabled":"false","ipv4Information":{"address":"192.168.24.241",
"subnetMask":"255.255.192.0"},"teaming":null,"redundancy":0,"status":"Normal",
"statusSummary":"Normal"}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

DESCRIPTION

A list of control port information.

PROPERTIES

data:*object[]*

ITEMS

controlPort: object (on page 587)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/control-ports/<*id*>

Required Role: Security, Storage, Monitor, Service, Audit, or Resource

DESCRIPTION

Obtains the control port information.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Control port ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/control-ports/f93b583a-478d-413c-abe9-028d8ae4f67b
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
```

```
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 247
Date: Fri, 30 Jul 2021 07:56:55 GMT

{"id":"f93b583a-478d-413c-abe9-028d8ae4f67b","storageNodeId":"9f29d742-9a75-4cef-9539-
35b3e76c8a20","macAddress":"00:50:56:bd:a3:fb","mtuSize":1500,"interfaceName":"eth0",
"ipv4Information":{"address":"192.168.1.101","subnetMask":"255.255.192.0"}}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

controlPort: object (on page 587)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/internode-ports

Required Role: Security, Storage, Monitor, Service, Audit, or Resource

DESCRIPTION

Obtains a list of internode port information.

REQUEST PARAMETERS

Name	Description	Type	Data type
storageNodeId	Storage node ID.	query	*string* (uuid)

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/internode-ports
HTTP/1.1 200 OK
```

```
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 1498
Date: Fri, 30 Jul 2021 07:56:20 GMT

{"data":[{"id":"07ccc0b9-b9b2-4996-a63d-9aaaddc5b2e3","storageNodeId":"8e01e189-38bc-
4002-a1fe-7a73633bce8c","macAddress":"80:30:e0:14:6b:20","mtuSize":1500,
"interfaceName":"eth1","deviceName":"NetXtreme II BCM57810 10 Gigabit Ethernet
(FlexFabric 10Gb 2-port 533FLR-T Adapter)","configuredPortSpeed":"Auto",
"portSpeedDuplex":"10Gbps Full","isTeamingEnabled":"false","ipv4Information":
{"address":"192.168.78.241","subnetMask":"255.255.192.0"},"teaming":null,
"redundancy":0,"status":"Normal","statusSummary":"Normal"},{"id":"4c1a93b9-1eaa-42e5-
9893-3020f4ff93e3","storageNodeId":"ad36eab7-20c3-41cd-acac-d334ff5347d7",
"macAddress":"80:30:e0:14:63:28","mtuSize":1500,"interfaceName":"eth1",
"deviceName":"NetXtreme II BCM57810 10 Gigabit Ethernet (FlexFabric 10Gb 2-port
533FLR-T Adapter)","configuredPortSpeed":"Auto","portSpeedDuplex":"10Gbps Full",
"isTeamingEnabled":"false","ipv4Information":{"address":"192.168.78.242",
"subnetMask":"255.255.192.0"},"teaming":null,"redundancy":0,"status":"Normal",
"statusSummary":"Normal"},{"id":"67a671ae-c1aa-4471-887d-a25feeff180e",
"storageNodeId":"3123caa9-f13a-4905-a6ee-815ffb7f98a2",
"macAddress":"80:30:e0:14:4f:78","mtuSize":1500,"interfaceName":"eth1",
"deviceName":"NetXtreme II BCM57810 10 Gigabit Ethernet (FlexFabric 10Gb 2-port
533FLR-T Adapter)","configuredPortSpeed":"Auto","portSpeedDuplex":"10Gbps Full",
"isTeamingEnabled":"false","ipv4Information":{"address":"192.168.78.243",
"subnetMask":"255.255.192.0"},"teaming":null,"redundancy":0,"status":"Normal",
"statusSummary":"Normal"}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of internode port information.

PROPERTIES

data:*object[]*

ITEMS

internodePort: object (on page 619)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/internode-ports/<*id*>

Required Role: Security, Storage, Monitor, Service, Audit, or Resource

DESCRIPTION

Obtains the internode port information.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Internode port ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/internode-ports/f9713eb8-5c93-4150-a969-84f7f1ad9582
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 247
Date: Fri, 30 Jul 2021 07:56:55 GMT

{"id":"f9713eb8-5c93-4150-a969-84f7f1ad9582","storageNodeId":"dd5ed12b-cf39-47da-9a30-
5ba53404050e","macAddress":"00:50:56:bd:73:f1","mtuSize":1500,"interfaceName":"eth1",
"ipv4Information":{"address":"192.168.73.212","subnetMask":"255.255.192.0"}}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

internodePort: object (on page 619)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/storage/actions/resume-suppressed-start-processing/invoke

Required Role: Service

DESCRIPTION

Resumes the starting processing forcibly for the storage cluster whose starting was suppressed.

A storage node that is not started is blocked and the storage cluster is started.

This API may succeed if the storage cluster cannot be forcibly started. Check the success or failure of the storage cluster startup process after resuming by referring to the event log.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -H 'Content-Length: 0' -X POST https://192.168.1.100/
ConfigurationManager/simple/v1/objects/storage/actions/resume-suppressed-start-
processing/invoke
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)

Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)

Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/storage-auto-recovery-setting

Required Role: Service

DESCRIPTION

Obtains the setting of whether to enable or disable the auto-recovery function and the status of the function.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/storage-auto-recovery-setting
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 83
Date: Fri, 30 Jul 2021 07:56:21 GMT

{"isEnabled":true,"storageNodePersistentBlockingThresholdTime":0,"status":"Normal"}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

storageAutoRecoverySetting: object (on page 699)

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

PATCH /v1/objects/storage-auto-recovery-setting

Required Role: Service

DESCRIPTION

Enables or disables the auto-recovery function.

Enabling the auto-recovery function allows storage nodes in temporarily blocked status to be detected automatically and recovery processing to be performed. A maintenance recovery job is submitted for recovery.

REQUEST BODY

Uses default content-types: application/json

patchStorageAutoRecoverySettingParam: object (on page 646)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d '{"isEnabled":true}' https://192.168.13.100/ConfigurationManager/simple/v1/objects/
storage-auto-recovery-setting
```

The following shows an example using a json body.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d @example.json https://192.168.13.100/ConfigurationManager/simple/v1/objects/storage-
auto-recovery-setting
```

example.json is defined as follows:

```
{
    "isEnabled":true
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/storage-network-setting

Required Role: Security, Storage, Monitor, Service, Audit, or Resource

DESCRIPTION

Obtains the storage cluster network settings.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/storage-network-setting
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 102
Date: Fri, 30 Jul 2021 07:56:21 GMT

{"primaryDnsServerIpAddress":"","secondaryDnsServerIpAddress":"",
"virtualIpv4Address":"192.168.1.100"}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

storageNetworkSetting: object (on page 702)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/storage-node-network-settings

Required Role: Security, Storage, Monitor, Service, or Resource

DESCRIPTION

Obtains a list of storage node network settings.

REQUEST PARAMETERS

Name	Description	Type	Data type
storageNodeId	Storage node ID.	query	*string* (uuid)

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/storage-node-network-settings
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 790
Date: Fri, 30 Jul 2021 07:56:22 GMT

{"data":[{"id":"44f1d113-405e-448f-ad77-fd5554971c36","ipv4Route":
[{"destination":"default","gateway":"192.168.1.14","interface":"eth0"}]},
{"id":"8fdbc008-ec6c-42b3-b930-146c94c5cef9","ipv4Route":[{"destination":"default",
"gateway":"192.168.1.14","interface":"eth0"}]},{"id":"9d36c162-e379-4c85-bcc2-
ccf98fe774a6","ipv4Route":[{"destination":"default","gateway":"192.168.1.14",
"interface":"eth0"}]},{"id":"9f29d742-9a75-4cef-9539-35b3e76c8a20","ipv4Route":
[{"destination":"default","gateway":"192.168.1.14","interface":"eth0"}]},
{"id":"a5b19d19-ca5d-4d19-988d-d625850c85de","ipv4Route":[{"destination":"default",
"gateway":"192.168.1.14","interface":"eth0"}]},{"id":"dd5ed12b-cf39-47da-9a30-
5ba53404050e","ipv4Route":[{"destination":"default","gateway":"192.168.1.14",
"interface":"eth0"}]}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

> DESCRIPTION
>
> A list of storage node network settings.
>
> PROPERTIES
>
> **data:** ***object[]***
>
> ITEMS
>
> storageNodeNetworkSetting: object (on page 707)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/storage-node-network-settings/<*id*>

Required Role: Security, Storage, Monitor, Service, or Resource

DESCRIPTION

Obtains the storage node network settings.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Storage node ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/storage-node-network-settings/dd5ed12b-cf39-47da-9a30-5ba53404050e
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
```

```
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 129
Date: Fri, 30 Jul 2021 07:56:56 GMT

{"id":"dd5ed12b-cf39-47da-9a30-5ba53404050e","ipv4Route":[{"destination":"default",
"gateway":"192.168.1.14","interface":"eth0"}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

storageNodeNetworkSetting: object (on page 707)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/storage-time-setting

Required Role: Security, Storage, Monitor, Service, Audit, or Resource

DESCRIPTION

Obtains the storage cluster time settings.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/storage-time-setting
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
```

```
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 95
Date: Fri, 30 Jul 2021 07:56:23 GMT

{"systemTime":"2021-07-30T07:56:23Z","ntpServerNames":["192.168.1.14"],
"timezone":"Asia/Tokyo"}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

storageTimeSetting: object (on page 710)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/system-requirements-file/actions/import/invoke

Required Role: Service

DESCRIPTION

Imports a system requirements file.

Specify a system requirements file that describes authorized hardware and system requirements.

You can run this API only for the cluster master node (primary). If you call this API for a node other than the cluster master node (primary), HTTP status code 412 (Precondition Failed) is returned.

An import succeeds only when the version of the specified system requirements file is later than the version of the system requirements file existing in the storage cluster.

An import is skipped and an event log is output when the version of the specified system requirements file is equal to or older than the version of the system requirements file existing in the storage cluster.

REQUEST BODY

multipart/form-data

REQUEST PARAMETERS

Name	Description	Type	Data type
systemRequirementsFile	System requirements file to be imported to the storage cluster. Specify a system requirements file that describes system requirements to be updated.	formData	*file* required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H  "Content-Type: multipart/form-data" -H
"Expect:" -F systemRequirementsFile=@SystemRequirementsFile.yml https://192.168.1.100/
ConfigurationManager/simple/v1/objects/system-requirements-file/actions/import/invoke
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)
Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 412 (on page 577)
Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/configuration-file/actions/create/invoke (Bare metal)

Required Role: Service

DESCRIPTION

Restores a configuration definition file.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -X POST -u admin:password -H "Content-Length: 0" https://192.168.1.100/
ConfigurationManager/simple/v1/objects/configuration-file/actions/create/invoke
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)
Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 412 (on page 577)
Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/configuration-file/download (Bare metal)

Required Role: Service

DESCRIPTION

Downloads a restored configuration definition file.

You can run this API only for the cluster master node (primary).

If you run this API for a node other than the cluster master node (primary), HTTP status code 412 (Precondition Failed) is returned.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -J -O -X GET -u admin:password -H "Content-Length: 0" https://192.168.1.100/
ConfigurationManager/simple/v1/objects/configuration-file/download
```

RESPONSES

Uses default content-types: application/gzip, application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 412 (on page 577)
Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

Chapter 23: User role management

POST /v1/objects/external-auth-server-root-certificates/<*targetServer*>/actions/import/invoke

Required Role: Security

DESCRIPTION

Imports a root certificate (to be used in communication with the user's external authentication server) to a storage cluster.

PEM- and DER-format authentication files are supported. The root certificate is applied immediately if TLS communication with the server is enabled in the storage cluster settings for the external authentication server specified as "targetServer".

This API can be executed only for a cluster master node (primary). If this API is executed for any node other than a cluster master node (primary), HTTP status code 412 (Precondition Failed) is returned.

REQUEST BODY

multipart/form-data

REQUEST PARAMETERS

Name	Description	Type	Data type
rootCertificate	Root certificate file (used in communication with the external authentication server) to be imported to the storage cluster.	formData	*file* required
targetServer	Target external authentication server in user authentication.	path	*string* , x ∈ { "primary1", "secondary1" } required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: multipart/form-data" -H "Expect:"
-F rootCertificate=@win2016ca.cer https://192.168.1.100/ConfigurationManager/
simple/v1/objects/external-auth-server-root-certificates/primary1/actions/import/
invoke
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)
Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 412 (on page 577)
Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/external-auth-server-root-certificates/<*targetServer*>/download

Required Role: Security

DESCRIPTION

Obtains a root certificate (to be used in communication with the user's external authentication server) which is imported to a storage cluster.

A root certificate is output as a DER file.

If this API is called when no root certificate is imported, HTTP status code 404 (Not Found) is returned.

This API can be executed only for a cluster master node (primary). If this API is executed for any node other than a cluster master node (primary), HTTP status code 412 (Precondition Failed) is returned.

If you have not imported the root certificate for the external authentication server, you may see message ID: KARS15553-E. If this is the case, import the root certificate of the external authentication server.

REQUEST PARAMETERS

Name	Description	Type	Data type
targetServer	Target external authentication server in user authentication.	path	*string* , x ∈ { "primary1", "secondary1" } required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -J -O -X GET -u admin:password https://192.168.1.100/ConfigurationManager/
simple/v1/objects/external-auth-server-root-certificates/primary1/download
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   832  100   832    0     0   1165      0 --:--:-- --:--:-- --:--:--  1163
curl: Saved to filename 'download-external-auth-server-root-certificate.crt'
```

RESPONSES

Uses default content-types: application/x-x509-ca-cert, application/json

200 OK

Root certificate file corresponding to the server specified as "targetServer".

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 412 (on page 577)

Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/external-auth-server-setting

Required Role: Security

DESCRIPTION

Obtains the settings of the external authentication server.

This API can be executed only for a cluster master node (primary). If this API is executed for any node other than a cluster master node (primary), HTTP status code 412 (Precondition Failed) is returned.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/external-auth-server-setting
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 401
Date: Fri, 30 Jul 2021 07:56:23 GMT

{"isEnabled":false,"authProtocol":"LDAP","ldapSetting":{"mappingMode":"User",
"primaryLdapServerUrl":"","secondaryLdapServerUrl":"","isStartTlsEnabled":false,
"baseDn":"","bindDn":"","userIdAttribute":"cn","userTreeDn":"",
"userObjectClass":"person","externalGroupNameAttribute":"cn","userGroupTreeDn":"",
"userGroupObjectClass":"group","timeoutSeconds":-1,"retryIntervalMilliseconds":100,
"maxRetries":3}}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

externalAuthServerSetting: object (on page 614)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 412 (on page 577)
Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

PATCH /v1/objects/external-auth-server-setting

Required Role: Security

DESCRIPTION

Edits the settings of the external authentication server.

This API can be executed only for a cluster master node (primary). If this API is executed for any node other than a cluster master node (primary), HTTP status code 412 (Precondition Failed) is returned.

REQUEST BODY

Uses default content-types: application/json

patchExternalAuthServerSettingParam: object (on page 642)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d '{ "isEnabled": true, "authProtocol": "LDAP", "ldapSetting": { "mappingMode":
"User", "primaryLdapServerUrl": "ldap://192.168.19.174:389",
"secondaryLdapServerUrl": "", "isStartTlsEnabled": false, "baseDn": "DC=local",
"bindDn": "CN=binddn,CN=Users,DC=userrole,DC=local", "bindDnPassword":
"TestPassword", "userIdAttribute": "sAMAccountName", "userTreeDn": "OU=Test01,
DC=userrole,DC=local", "userObjectClass": "user", "userGroupIdAttribute":
"sAMAccountName", "userGroupTreeDn": "OU=Test01,DC=userrole,DC=local",
"userGroupObjectClass": "group" } }' https://192.168.1.100/ConfigurationManager/
simple/v1/objects/external-auth-server-setting
```

The following shows an example using a json body.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d @example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/external-
auth-server-setting
```

example.json is defined as follows:

```
{
    "isEnabled": true,
    "authProtocol": "LDAP",
    "ldapSetting":
        {
            "mappingMode": "User",
            "primaryLdapServerUrl": "ldap://192.168.19.174:389",
            "secondaryLdapServerUrl": "",
            "isStartTlsEnabled": false,
            "baseDn": "DC=local",
            "bindDn": "CN=binddn,CN=Users,DC=userrole,DC=local",
            "bindDnPassword": "TestPassword",
            "userIdAttribute": "sAMAccountName",
            "userTreeDn": "OU=Test01,DC=userrole,DC=local",
            "userObjectClass": "user",
            "userGroupIdAttribute": "sAMAccountName",
            "userGroupTreeDn": "OU=Test01,DC=userrole,DC=local",
            "userGroupObjectClass": "group"
        }
}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

externalAuthServerSetting: object (on page 614)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 412 (on page 577)

Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)

Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/external-auth-server-setting/actions/verify-connectivity/invoke

Required Role: Security or Service

DESCRIPTION

Verifies the connection with the external authentication server. As the connection destination, use the external authentication server which is already set.

This API can be executed only for a cluster master node (primary). If this API is executed for any node other than a cluster master node (primary), HTTP status code 412 (Precondition Failed) is returned.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Length: 0" https://192.168.1.100/
ConfigurationManager/simple/v1/objects/external-auth-server-setting/actions/verify-
connectivity/invoke
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

externalAuthServerConnectionVerification: object (on page 614)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)

Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 412 (on page 577)

Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)

Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/user-auth-setting

Required Role: None

DESCRIPTION

Obtains the user authentication settings.

This API can be executed only for a cluster master node (primary). If this API is executed for any node other than a cluster master node (primary), HTTP status code 412 (Precondition Failed) is returned.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/user-auth-setting
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
```

```
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 393
Date: Fri, 30 Jul 2021 07:56:24 GMT

{"passwordComplexitySetting":{"minLength":8,"minNumberOfUpperCaseChars":0,
"minNumberOfLowerCaseChars":0,"minNumberOfNumerals":0,"minNumberOfSymbols":0,
"numberOfPasswordHistory":1},"passwordAgeSetting":
{"requiresInitialPasswordReset":true,"minAgeDays":0,"maxAgeDays":42},"lockoutSetting":
{"maxAttempts":3,"lockoutSeconds":60},"sessionSetting":{"maxLifetimeSeconds":86400,
"maxIdleSeconds":1800}}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

userAuthSetting: object (on page 724)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 412 (on page 577)

Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

PATCH /v1/objects/user-auth-setting

Required Role: Security

DESCRIPTION

Edits the user authentication settings.

This API can be executed only for a cluster master node (primary). If this API is executed for any node other than a cluster master node (primary), HTTP status code 412 (Precondition Failed) is returned.

REQUEST BODY

Uses default content-types: application/json

patchUserAuthSettingParam: object (on page 648)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d '{ "passwordComplexitySetting":{ "minLength": 4, "minNumberOfUpperCaseChars": 0,
"minNumberOfLowerCaseChars": 0, "minNumberOfNumerals": 0, "minNumberOfSymbols": 0,
"numberOfPasswordHistory": 1 }, "passwordAgeSetting":
{ "requiresInitialPasswordReset":false, "minAgeDays": 0, "maxAgeDays": 42 },
"lockoutSetting":{ "maxAttempts": 3, "lockoutSeconds": 60 }, "sessionSetting":
{ "maxLifetimeSeconds": 86400, "maxIdleSeconds": 1800 } }' https://192.168.1.100/
ConfigurationManager/simple/v1/objects/user-auth-setting
```

The following shows an example using a json body.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d @example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/user-
auth-setting
```

example.json is defined as follows:

```
{
    "passwordComplexitySetting":
        {
            "minLength": 4,
            "minNumberOfUpperCaseChars": 0,
            "minNumberOfLowerCaseChars": 0,
            "minNumberOfNumerals": 0,
            "minNumberOfSymbols": 0,
            "numberOfPasswordHistory": 1
        },
    "passwordAgeSetting":
        {
            "requiresInitialPasswordReset":false,
            "minAgeDays": 0,
            "maxAgeDays": 42
        },
    "lockoutSetting":
        {
            "maxAttempts": 3,
            "lockoutSeconds": 60
        },
    "sessionSetting":
        {
            "maxLifetimeSeconds": 86400,
            "maxIdleSeconds": 1800
        }
}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

userAuthSetting: object (on page 724)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 412 (on page 577)

Indicates that the conditions for executing the API are not met.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)

Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/user-groups

Required Role: Security or VpsSecurity

DESCRIPTION

Obtains a list of user groups.

REQUEST PARAMETERS

Name	Description	Type	Data type
vpsId	The ID of the virtual private storage (VPS) that the acquisition-target resource belongs to. To filter out the resources that do not belong to the VPS, specify "system". To filter the resources by the VPS that the resources belong to, specify it in UUID format.	query	*string* , must match /^system $\|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/
ConfigurationManager/simple/v1/objects/user-groups

HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 2402
Date: Thu, 20 Oct 2022 06:00:00 GMT

{"data":[{"userGroupId":"SecurityAdministrators",
"userGroupObjectId":"SecurityAdministrators","roleNames":["Security","Monitor"],
"isBuiltIn":true,"externalGroupName":null,"vpsId":"(system)","scope":["system"]},
{"userGroupId":"AuditAdministrators","userGroupObjectId":"AuditAdministrators",
"roleNames":["Audit","Monitor"],"isBuiltIn":true,"externalGroupName":null,
"vpsId":"(system)","scope":["system"]},{"userGroupId":"MonitorUsers",
"userGroupObjectId":"MonitorUsers","roleNames":["Monitor"],"isBuiltIn":true,
"externalGroupName":null,"vpsId":"(system)","scope":["system"]},
{"userGroupId":"ServiceAdministrators","userGroupObjectId":"ServiceAdministrators",
"roleNames":["Storage","Service"],"isBuiltIn":true,"externalGroupName":null,
"vpsId":"(system)","scope":["system"]},{"userGroupId":"SystemAdministrators",
"userGroupObjectId":"SystemAdministrators","roleNames":["Audit","Security","Storage",
"Monitor","Service","Resource"],"isBuiltIn":true,"externalGroupName":null,
"vpsId":"(system)","scope":["system"]},{"userGroupId":"StorageAdministrators",
"userGroupObjectId":"StorageAdministrators","roleNames":["Storage","Resource"],
"isBuiltIn":true,"externalGroupName":null,"vpsId":"(system)","scope":["system"]},
{"userGroupId":"UG_ALL","userGroupObjectId":"UG_ALL","roleNames":["Security",
"Storage","Monitor","Service","Audit","Resource"],"isBuiltIn":false,
"externalGroupName":null,"vpsId":"(system)","scope":["system","54665c29-784a-4b71-
9b61-2894d55697e9"]},{"userGroupId":"SysSec","userGroupObjectId":"SysSec","roleNames":
["Security"],"isBuiltIn":false,"externalGroupName":null,"vpsId":"(system)","scope":
["system"]},{"userGroupId":"SysSto","userGroupObjectId":"SysSto","roleNames":
["Storage"],"isBuiltIn":false,"externalGroupName":null,"vpsId":"(system)","scope":
["system"]},{"userGroupId":"SysMo","userGroupObjectId":"SysMo","roleNames":
["Monitor"],"isBuiltIn":false,"externalGroupName":null,"vpsId":"(system)","scope":
["system"]},{"userGroupId":"SysAu","userGroupObjectId":"SysAu","roleNames":["Audit"],
"isBuiltIn":false,"externalGroupName":null,"vpsId":"(system)","scope":["system"]},
{"userGroupId":"SysSer","userGroupObjectId":"SysSer","roleNames":["Service"],
"isBuiltIn":false,"externalGroupName":null,"vpsId":"(system)","scope":["system"]},
{"userGroupId":"SysRe","userGroupObjectId":"SysRe","roleNames":["Resource"],
"isBuiltIn":false,"externalGroupName":null,"vpsId":"(system)","scope":["system"]}]}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of user group summary information.

PROPERTIES

data:*object[]*

ITEMS

userGroupSummary: object (on page 726)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/user-groups

Required Role: Security or VpsSecurity

DESCRIPTION

Creates a user group.

REQUEST BODY

Uses default content-types: application/json

createUserGroupParam: object (on page 596)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
'{"userGroupId":"testGroup", "roleNames": [ "Service"], "externalGroupName":
"testgroup"}' https://192.168.1.100/ConfigurationManager/simple/v1/objects/user-groups
```

The following shows an example using a json body.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
@example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/user-groups
```

example.json is defined as follows:

```
{
    "userGroupId":"testGroup",
    "roleNames":
        [
            "Service"
        ],
    "externalGroupName": "testgroup"
}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

userGroup: object (on page 724)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

DELETE /v1/objects/user-groups/<*userGroupId*>

Required Role: Security or VpsSecurity

DESCRIPTION

Deletes a user group.

REQUEST PARAMETERS

Name	Description	Type	Data type
userGroupId	User group ID.	path	*string* (1 to 64 chars), must match /^[a-zA-Z0-9!#\$%&'\-\.@\^_`\{\}~]{1,64}$/ required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -X DELETE -u admin:password https://192.168.1.100/ConfigurationManager/simple/v1/
objects/user-groups/SampleGroup
```

RESPONSES

Uses default content-types: application/json

HTTP status code 204 (on page 575)
Indicates that the request was processed correctly, but no response data to be returned exists.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 409 (on page 577)
Indicates that an inconsistent or impossible request was issued for the resource specified by URL (for example, creating a resource whose ID is same as the resource already created).

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/user-groups/<*userGroupId*>

Required Role: Security or VpsSecurity

DESCRIPTION

Obtains the user group information.

REQUEST PARAMETERS

Name	Description	Type	Data type
userGroupId	User group ID.	path	*string* (1 to 64 chars), must match /^[a-zA-Z0-9!#\$%&'\-\.@\^_`\{\}~]{1,64}$/ required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/user-groups/SystemAdministrators

HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 293
Date: Thu, 20 Oct 2022 06:23:31 GMT

{"memberUsers":[{"userId":"admin","userObjectId":"admin"}],
"userGroupId":"SystemAdministrators","userGroupObjectId":"SystemAdministrators",
"roleNames":["Audit","Security","Storage","Monitor","Service","Resource"],
"isBuiltIn":true,"externalGroupName":null,"vpsId":"(system)","scope":["system"]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

userGroup: object (on page 724)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

PATCH /v1/objects/user-groups/<*userGroupId*>

Required Role: Security or VpsSecurity

DESCRIPTION

Edits the user group information.

REQUEST BODY

Uses default content-types: application/json

patchUserGroupParam: object (on page 649)

REQUEST PARAMETERS

Name	Description	Type	Data type
userGroupId	User group ID.	path	*string* (1 to 64 chars), must match /^[a-zA-Z0-9!#\$%&'\-\.@\^_`\{\}~]{1,64}$/ required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d '{"roleNames": ["Security","Service","Storage","Audit"]}' https://192.168.1.100/
ConfigurationManager/simple/v1/objects/user-groups/groupA
```

The following shows an example using a json body.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d @example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/user-
groups/groupA
```

example.json is defined as follows:

```
{
    "roleNames":
        [
            "Security",
            "Service",
            "Storage",
            "Audit"
            ]
}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

userGroup: object (on page 724)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 409 (on page 577)

Indicates that an inconsistent or impossible request was issued for the resource specified by URL (for example, creating a resource whose ID is same as the resource already created).

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)

Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/users

Required Role: Security or VpsSecurity

DESCRIPTION

Obtains a list of users.

If external authentication is enabled and "mappingMode" is set to "Group", the users on the external authentication server are not included in the output list and only the users whose "authentication" is "local" are included in the output list.

REQUEST PARAMETERS

Name	Description	Type	Data type
vpsId	The ID of the virtual private storage (VPS) that the acquisition-target resource belongs to.	query	*string* , must match /^system$\|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

Name	Description	Type	Data type
	To filter out the resources that do not belong to the VPS, specify "system". To filter the resources by the VPS that the resources belong to, specify it in UUID format.		

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/users

HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 3194
Date: Thu, 20 Oct 2022 06:19:29 GMT

{"data":[{"userId":"admin","userObjectId":"admin","passwordExpirationTime":"2022-11-
30T07:21:21Z","isEnabled":true,"userGroups":[{"userGroupId":"SystemAdministrators",
"userGroupObjectId":"SystemAdministrators"}],"isBuiltIn":true,
"authentication":"local","roleNames":["Security","Storage","Monitor","Service",
"Audit","Resource"],"isEnabledConsoleLogin":null,"vpsId":"(system)","privileges":
[{"scope":"system","roleNames":["Audit","Security","Storage","Monitor","Service",
"Resource"]}]}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of user information.

PROPERTIES

data:*object[]*

ITEMS

user: object (on page 722)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/users

Required Role: Security or VpsSecurity

DESCRIPTION

Creates a user.

REQUEST BODY

Uses default content-types: application/json

createUserParam: object (on page 598)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
'{"userId":"testUser", "password":"testPassword", "userGroupIds":
[ "SecurityAdministrators"], "authentication": "external"}' https://192.168.1.100/
ConfigurationManager/simple/v1/objects/users
```

The following shows an example using a json body.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
@example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/users
```

example.json is defined as follows:

```
{
    "userId":"testUser",
    "password":"testPassword",
    "userGroupIds":
        [
            "SecurityAdministrators"
        ],
    "authentication": "external"
}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

user: object (on page 722)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 409 (on page 577)

Indicates that an inconsistent or impossible request was issued for the resource specified by URL (for example, creating a resource whose ID is same as the resource already created).

HTTP status code 411 (on page 577)

Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

DELETE /v1/objects/users/<*userId*>

Required Role: Security or VpsSecurity

DESCRIPTION

Deletes a user.

If you do not have access permission for the virtual private storage (VPS) that the specified user belongs to, HTTP status code 404 (Not Found) is returned.

REQUEST PARAMETERS

Name	Description	Type	Data type
userId	User ID.	path	*string* (5 to 255 chars), must match /^[\-A-Za-z0-9!#\$%&'\.@\^_`\{\}~]{5,255}$/ required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -X DELETE -u admin:password https://192.168.1.100/ConfigurationManager/simple/v1/
objects/users/sample-user
```

RESPONSES

Uses default content-types: application/json

HTTP status code 204 (on page 575)
Indicates that the request was processed correctly, but no response data to be returned exists.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 409 (on page 577)
Indicates that an inconsistent or impossible request was issued for the resource specified by URL (for example, creating a resource whose ID is same as the resource already created).

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)

Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/users/<*userId*>

Required Role: None

DESCRIPTION

Obtains the user information.

If you have the Security or VpsSecurity role, in this API, you can specify the user ID of any user in your accessible range. If you do not have the Security or VpsSecurity role, you can specify only your own user ID. If you specify a user ID other than your own, HTTP status code 403 (Forbidden) is returned regardless of whether the specified user ID exists.

If you specify a user on the external authentication server in the API when external authentication is enabled and "mappingMode" is set to "Group", HTTP status code 404 (Not Found) is returned.

If "self" is specified for userId, information about the user who was authenticated when this API was issued is returned.

REQUEST PARAMETERS

Name	Description	Type	Data type
userId	User ID.	path	*string* (up to 255 chars), must match /^self$\|^[\-A-Za-z0-9!#\$%&'\.@\^_`\{\}~]{5,255}$/ required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/users/admin

HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 475
Date: Thu, 20 Oct 2022 06:24:53 GMT

{"userId":"admin","userObjectId":"admin","passwordExpirationTime":"2022-11-
30T07:21:21Z","isEnabled":true,"userGroups":[{"userGroupId":"SystemAdministrators",
"userGroupObjectId":"SystemAdministrators"}],"isBuiltIn":true,
"authentication":"local","roleNames":["Security","Storage","Monitor","Service",
"Audit","Resource"],"isEnabledConsoleLogin":null,"vpsId":"(system)","privileges":
[{"scope":"system","roleNames":["Audit","Security","Storage","Monitor","Service",
"Resource"]}]}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

user: object (on page 722)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

PATCH /v1/objects/users/<*userId*>

Required Role: Security or VpsSecurity

DESCRIPTION

Edits the user information.

When the password is changed for the API, if requiresInitialPasswordReset of the user authentication settings is true, the password of the user expires.

The expired password can be recovered by using the password changing API(PATCH /v1/objects/users/<*userId*>/password) or CLI(user_password_set) of the local user.

If you do not have access permission for the virtual private storage (VPS) that the specified user belongs to, HTTP status code 404 (Not Found) is returned.

REQUEST BODY

Uses default content-types: application/json

patchUserParam: object (on page 649)

REQUEST PARAMETERS

Name	Description	Type	Data type
userId	User ID.	path	*string* (5 to 255 chars), must match /^[\-A-Za-z0-9!#\$%&'\.@\^_`\{\}~]{5,255}$/ required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d '{"isEnabled": true}' https://192.168.1.100/ConfigurationManager/simple/v1/objects/
users/admin
```

The following shows an example using a json body.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d @example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/users/
admin
```

example.json is defined as follows:

```
{
    "isEnabled": true
}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

user: object (on page 722)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 409 (on page 577)

Indicates that an inconsistent or impossible request was issued for the resource specified by URL (for example, creating a resource whose ID is same as the resource already created).

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/users/<*userId*>/actions/add-user-group/invoke

Required Role: Security or VpsSecurity

DESCRIPTION

Adds a user to a user group.

If you do not have access permission for the virtual private storage (VPS) that the specified user belongs to, HTTP status code 404 (Not Found) is returned.

REQUEST BODY

Uses default content-types: application/json

addUserToUserGroupParam: object (on page 581)

REQUEST PARAMETERS

Name	Description	Type	Data type
userId	User ID.	path	*string* (5 to 255 chars), must match /^[\-A-Za-z0-9!#\$%&'\.@\^_`\{\}~]{5,255}$/ required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
'{"userGroupIds": ["StorageAdministrators"]}' https://192.168.1.100/
ConfigurationManager/simple/v1/objects/users/userAA/actions/add-user-group/invoke
```

The following shows an example using a json body.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
@example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/users/
userAA/actions/add-user-group/invoke
```

example.json is defined as follows:

```
{
    "userGroupIds":
        [
            "StorageAdministrators"
        ]
}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

user: object (on page 722)

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 409 (on page 577)
Indicates that an inconsistent or impossible request was issued for the resource specified by URL (for example, creating a resource whose ID is same as the resource already created).

HTTP status code 411 (on page 577)
Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/users/<*userId*>/actions/delete-user-group/invoke

Required Role: Security or VpsSecurity

DESCRIPTION

Deletes a user from a user group.

If you do not have access permission for the virtual private storage (VPS) that the specified user belongs to, HTTP status code 404 (Not Found) is returned.

REQUEST BODY

Uses default content-types: application/json

deleteUserFromUserGroupParam: object (on page 605)

REQUEST PARAMETERS

Name	Description	Type	Data type
userId	User ID.	path	*string* (5 to 255 chars), must match /^[\-A-Za-z0-9!#\$%&'\.@\^_`\{\}~]{5,255}$/ required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
'{"userGroupIds": ["groupA"]}' https://192.168.1.100/ConfigurationManager/simple/v1/
objects/users/userAA/actions/delete-user-group/invoke
```

The following shows an example using a json body.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
@example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/users/
userAA/actions/delete-user-group/invoke
```

example.json is defined as follows:

```
{
    "userGroupIds":
        [
            "groupA"
        ]
}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

user: object (on page 722)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 409 (on page 577)

Indicates that an inconsistent or impossible request was issued for the resource specified by URL (for example, creating a resource whose ID is same as the resource already created).

HTTP status code 411 (on page 577)

Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)

Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

PATCH /v1/objects/users/<*userId*>/password

Required Role: None

DESCRIPTION

Changes the password of the local user.

You can execute this API only for a user whose "authentication" is set to "local". If you execute the API for a user whose "authentication" is set to "external", HTTP status code 401 (Unauthorized) is returned.

For security reasons, HTTP status code 401 (Unauthorized) is returned even if an invalid password is specified as the current password or a non-existent user is specified as userID.

REQUEST BODY

Uses default content-types: application/json

changePasswordParam: object (on page 585)

REQUEST PARAMETERS

Name	Description	Type	Data type
userId	user ID.	path	*string* (5 to 255 chars), must match /^[\-A-Za-z0-9!#\$%&'\.@\^_`\{\}~]{5,255}$/ required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d '{"currentPassword":"password","newPassword":"new-password"}' https://192.168.1.100/
ConfigurationManager/simple/v1/objects/users/admin/password
```

The following shows an example using a json body.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d @example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/users/
admin/password
```

example.json is defined as follows:

```
{
    "currentPassword":"password",
    "newPassword":"new-password"
}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

user: object (on page 722)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)

Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Chapter 24: Virtual Private Storage (VPS) management

GET /v1/objects/virtual-private-storages

Required Role: Security, Storage, Monitor, Service, Resource, VpsSecurity, VpsStorage, or VpsMonitor

DESCRIPTION

Obtains a list of information about a virtual private storage (VPS).

If you have the Security, Storage, Monitor, or Service role, this API command also outputs VPS summary information, in addition to a list of VPS information.

If you have the VpsSecurity, VpsStorage, or VpsMonitor role, only your own VPS information is output.

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/virtual-private-storages

HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 1204
Date: Tue, 18 Oct 2022 02:27:13 GMT

{"data":[{"id":"83747dfe-f347-453c-bd72-f3bad58ea96c","name":"VPS_01",
"upperLimitForNumberOfUserGroups":256,"numberOfUserGroupsCreated":3,
"upperLimitForNumberOfUsers":256,"numberOfUsersCreated":3,
"upperLimitForNumberOfSessions":436,"numberOfSessionsCreated":0,
"upperLimitForNumberOfServers":100,"numberOfServersCreated":0,
"upperLimitForNumberOfHbas":400,"numberOfHbasCreated":0,
"upperLimitForNumberOfVolumeServerConnections":100,
"numberOfVolumeServerConnectionsCreated":0,"volumeSettings":{"poolId":"66048bfd-96cc-
46aa-8e9b-67ac93c63451","upperLimitForNumberOfVolumes":50,"numberOfVolumesCreated":2,
"upperLimitForCapacityOfVolumes":100,"capacityOfVolumesCreated":94,
"upperLimitForCapacityOfSingleVolume":-1,"qosParam":{"upperLimitForIopsOfVolume":-1,
"upperLimitForTransferRateOfVolume":-1,"upperAlertAllowableTimeOfVolume":-1}}}],
"summaryInformation":{"totalCount":1,"totalUpperLimitForNumberOfUserGroups":256,
"totalUpperLimitForNumberOfUsers":256,"totalUpperLimitForNumberOfSessions":436,
"totalUpperLimitForNumberOfVolumes":50,"totalUpperLimitForCapacityOfVolumes":100,
"totalUpperLimitForNumberOfServers":100,"totalUpperLimitForNumberOfHbas":400,
"totalUpperLimitForNumberOfVolumeServerConnections":100}}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

virtualPrivateStorageList: object (on page 732)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/virtual-private-storages

Required Role: Resource

DESCRIPTION

Creates a virtual private storage (VPS).

REQUEST BODY

Uses default content-types: application/json

createVirtualPrivateStorageParam: object (on page 599)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
"{ "name": "VPS_01", "upperLimitForNumberOfServers": 100 , "volumeSettings":
[{ "poolId": "803eb9da-6da4-4b48-885c-1d495d992d2c",
"upperLimitForCapacityOfVolumes": 100, "upperLimitForNumberOfVolumes": 50,
"upperLimitForCapacityOfSingleVolume": -1, "upperLimitForIopsOfVolume": -1,
"upperLimitForTransferRateOfVolume": -1, "upperAlertAllowableTimeOfVolume": -1 }]}"
https://192.168.1.100/ConfigurationManager/simple/v1/objects/virtual-private-storages
```

The following shows an example using a json body.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
@example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/virtual-
private-storages
```

example.json is defined as follows:

```
{
    "name": "VPS_01",
    "upperLimitForNumberOfServers": 100 ,
    "volumeSettings":
        [
            {
                "poolId": "803eb9da-6da4-4b48-885c-1d495d992d2c",
                "upperLimitForCapacityOfVolumes": 100,
                "upperLimitForNumberOfVolumes": 50,
                "upperLimitForCapacityOfSingleVolume": -1,
                "upperLimitForIopsOfVolume": -1,
                "upperLimitForTransferRateOfVolume": -1,
                "upperAlertAllowableTimeOfVolume": -1
            }
        ]
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)

Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)

Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)

Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

DELETE /v1/objects/virtual-private-storages/<*id*>

Required Role: Resource

DESCRIPTION

Deletes a virtual private storage (VPS).

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the VPS.	path	*string* , must match /^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/ required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -X  DELETE -u admin:password https://192.168.1.100/ConfigurationManager/
simple/v1/objects/virtual-private-storages/83747dfe-f347-453c-bd72-f3bad58ea96c
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/virtual-private-storages/<*id*>

Required Role: Security, Storage, Monitor, Service, Resource, VpsSecurity, VpsStorage, or VpsMonitor

DESCRIPTION

Obtains information about a virtual private storage (VPS).

If you have the VpsSecurity, VpsStorage, or VpsMonitor role, you are not allowed to view a VPS that is not included in a scope.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	The ID of the VPS.	path	*string* , must match /^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/ required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/virtual-private-storages/83747dfe-f347-453c-bd72-f3bad58ea96c

HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 819
Date: Tue, 18 Oct 2022 02:31:43 GMT

{"id":"83747dfe-f347-453c-bd72-f3bad58ea96c","name":"VPS_01",
"upperLimitForNumberOfUserGroups":256,"numberOfUserGroupsCreated":3,
"upperLimitForNumberOfUsers":256,"numberOfUsersCreated":3,
"upperLimitForNumberOfSessions":436,"numberOfSessionsCreated":0,
"upperLimitForNumberOfServers":100,"numberOfServersCreated":0,
"upperLimitForNumberOfHbas":400,"numberOfHbasCreated":0,
"upperLimitForNumberOfVolumeServerConnections":100,
"numberOfVolumeServerConnectionsCreated":0,"volumeSettings":{"poolId":"66048bfd-96cc-
46aa-8e9b-67ac93c63451","upperLimitForNumberOfVolumes":50,"numberOfVolumesCreated":2,
"upperLimitForCapacityOfVolumes":100,"capacityOfVolumesCreated":94,
"upperLimitForCapacityOfSingleVolume":-1,"qosParam":{"upperLimitForIopsOfVolume":-1,
"upperLimitForTransferRateOfVolume":-1,"upperAlertAllowableTimeOfVolume":-1}}}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

virtualPrivateStorage: object (on page 728)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

PATCH /v1/objects/virtual-private-storages/<*id*>

Required Role: Resource

DESCRIPTION

Edits settings of a virtual private storage (VPS).

REQUEST BODY

Uses default content-types: application/json

patchVirtualPrivateStorageParam: object (on page 650)

REQUEST PARAMETERS

Name	Description	Type	Data type
id	VPS ID.	path	*string* , must match /^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/ required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
url -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -d
"{ "upperLimitForNumberOfServers": 1000 }" https://192.168.1.100/ConfigurationManager/
simple/v1/objects/virtual-private-storages/83747dfe-f347-453c-bd72-f3bad58ea96c
```

The following shows an example using a json body.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d @example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/virtual-
private-storages/83747dfe-f347-453c-bd72-f3bad58ea96c
```

example.json is defined as follows:

```
{
    "upperLimitForNumberOfServers": 1000
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

Chapter 25: Volume management

GET /v1/objects/volumes

Required Role: Security, Storage, Monitor, Service, Resource, VpsSecurity, VpsStorage, or VpsMonitor

DESCRIPTION

Obtains a list of volume information.

Volume information to be obtained can be narrowed down using a query parameter. Details are as follows:

- Using count, the number of volume information items to be obtained in a batch can be specified. By default, 500 is set.
- By specifying the last returned value for enumerateContext, you can obtain the succeeding information.

If you add a volume while obtaining a series of volume information using enumerateContext, the added volume information might not be obtained. In this case, obtain information from the beginning again without specifying enumerateContext.

totalCount of the response data is the total number of volumes when the API was executed. The number might be changed if volume information is obtained using enumerateContext, and volumes are added or deleted.

REQUEST PARAMETERS

Name	Description	Type	Data type
poolId	The ID of the storage pool to which the volume belongs.	query	*string* (uuid)
poolName	The name of the storage pool to which the volume belongs.	query	*string*
serverId	The ID of the compute node to which the volume is allocated.	query	*string* (uuid)

Name	Description	Type	Data type
serverNickname	The nickname of the compute node to which the volume is allocated (partial match). This parameter is ignored if serverId is specified.	query	*string*
name	Volume name (exact match).	query	*string*
names	Volume name (exact match). You can specify plural names up to 32 with comma (,) delimiter. Example: names=<*name1*>,<*name2*>,<*name3*>... If the name includes a comma, escape it using '\' (backslash). Example: if the name is "name,1", specify "names=name\,1". If you enter a space character after a comma, the space character is recognized as a part of the name specified after the comma. Also, if you specify 33 or more names, HTTP status code 400 (Bad Request) is returned.	query	*string* , Each item must match /^[\-A-Za-z0-9,\.:@_]{1,32}$/
nickname	The nickname of the volume (partial match).	query	*string*
minTotalCapacity	Minimum volume capacity (MiB).	query	*integer*
maxTotalCapacity	Maximum volume capacity (MiB).	query	*integer*
minUsedCapacity	Minimum occupied volume capacity (MiB).	query	*integer*

Name	Description	Type	Data type
maxUsedCapacity	Maximum occupied volume capacity (MiB).	query	*integer*
savingSetting	Settings of the data reduction function for volumes.	query	*string* , x ∈ { "Disabled" }
enumerateContext	Token that helps obtain the next batch records. Do not specify this parameter for initial information extraction.	query	*string* (uuid)
count	The maximum number of obtained volume information items.	query	*integer* , { x ∈ ℤ | 1 ≤ x ≤ 500 } (default: "500")
storageControllerId	Storage controller ID.	query	*string* (uuid)
vpsId	The ID of the virtual private storage (VPS) that the acquisition-target resource belongs to. To filter out the resources that do not belong to the VPS, specify "system". To filter the resources by the VPS that the resources belong to, specify it in UUID format.	query	*string* , must match /^system$\|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

REQUEST EXAMPLE

Example 1:

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/volumes

HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
```

```
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 2806
Date: Thu, 20 Oct 2022 05:31:45 GMT

{"data":[{"savingEffects":{"systemDataCapacity":0,
"preCapacityDataReductionWithoutSystemData":0,"postCapacityDataReduction":0},
"id":"27f8944a-e98b-490d-b8fb-14583ceabc65","name":"Volume7","nickname":"system_vol",
"volumeNumber":7,"poolId":"7bd0f11a-0d13-4f4e-8f28-e4b88c8ecc1a","poolName":"SP01",
"totalCapacity":47,"usedCapacity":0,"numberOfConnectingServers":0,
"numberOfSnapshots":0,"protectionDomainId":"b392fe52-bcc5-404a-87fa-668fdb55ad62",
"fullAllocated":false,"volumeType":"Snapshot","statusSummary":"Normal",
"status":"Normal","storageControllerId":"a57046ae-02c7-418b-b23c-5e38cf42b9c9",
"snapshotAttribute":"S-VOL","snapshotStatus":"Prepared","savingSetting":"Disabled",
"savingMode":null,"dataReductionStatus":"Disabled","dataReductionProgressRate":null,
"vpsId":"(system)","vpsName":"(system)","qosParam":{"upperLimitForIops":-1,
"upperLimitForTransferRate":-1,"upperAlertAllowableTime":-1,"upperAlertTime":null}},
{"savingEffects":{"systemDataCapacity":0,
"preCapacityDataReductionWithoutSystemData":0,"postCapacityDataReduction":0},
"id":"35d16c28-b033-4c00-9956-67e8221493d3","name":"system_vol",
"nickname":"system_vol","volumeNumber":2,"poolId":"7bd0f11a-0d13-4f4e-8f28-
e4b88c8ecc1a","poolName":"SP01","totalCapacity":47,"usedCapacity":0,
"numberOfConnectingServers":0,"numberOfSnapshots":1,"protectionDomainId":"b392fe52-
bcc5-404a-87fa-668fdb55ad62","fullAllocated":false,"volumeType":"Snapshot",
"statusSummary":"Normal","status":"Normal","storageControllerId":"a57046ae-02c7-418b-
b23c-5e38cf42b9c9","snapshotAttribute":"P-VOL","snapshotStatus":null,
"savingSetting":"Disabled","savingMode":null,"dataReductionStatus":"Disabled",
"dataReductionProgressRate":null,"vpsId":"(system)","vpsName":"(system)","qosParam":
{"upperLimitForIops":-1,"upperLimitForTransferRate":-1,"upperAlertAllowableTime":-1,
"upperAlertTime":null}},{"savingEffects":{"systemDataCapacity":0,
"preCapacityDataReductionWithoutSystemData":0,"postCapacityDataReduction":0},
"id":"7e4eb895-48cc-4f48-b228-9224af6aa4f6","name":"vps_vol","nickname":"vps_vol",
"volumeNumber":4,"poolId":"7bd0f11a-0d13-4f4e-8f28-e4b88c8ecc1a","poolName":"SP01",
"totalCapacity":47,"usedCapacity":0,"numberOfConnectingServers":0,
"numberOfSnapshots":0,"protectionDomainId":"b392fe52-bcc5-404a-87fa-668fdb55ad62",
"fullAllocated":false,"volumeType":"Normal","statusSummary":"Normal",
"status":"Normal","storageControllerId":"2f298c1f-9fdd-496d-9e2e-3c439d46d2da",
"snapshotAttribute":"-","snapshotStatus":null,"savingSetting":"Disabled",
"savingMode":null,"dataReductionStatus":"Disabled","dataReductionProgressRate":null,
"vpsId":"54665c29-784a-4b71-9b61-2894d55697e9","vpsName":"VPS_01","qosParam":
{"upperLimitForIops":100,"upperLimitForTransferRate":100,
"upperAlertAllowableTime":100,"upperAlertTime":null}}],"totalCount":3,"hasNext":false,
"enumerateContext":null}
```

Example 2:

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password, and specifying the volume name (name).

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/volumes?name=VOL001
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 803
Date: Thu, 24 Mar 2022 07:39:14 GMT

{"data":[{"savingEffects":{"systemDataCapacity":0,
"preCapacityDataReductionWithoutSystemData":0,"postCapacityDataReduction":0},
"id":"6c4f42da-dc94-4b3c-992c-0a84c244f88f","name":"VOL001","nickname":"VOL001",
"volumeNumber":0,"poolId":"ce221f32-10c6-46fc-b361-e9c30182e314","poolName":"SP01",
"totalCapacity":51200,"usedCapacity":0,"numberOfConnectingServers":0,
"numberOfSnapshots":0,"protectionDomainId":"636c33c6-56c6-448f-bef6-dbf7b9b79e3c",
"fullAllocated":false,"volumeType":"Normal","statusSummary":"Normal",
"status":"Normal","storageControllerId":"1624fa11-a368-4f3f-b37e-98e2dd04aaf1",
"snapshotAttribute":"-","snapshotStatus":null,"savingSetting":"Disabled",
"savingMode":null,"dataReductionStatus":"Disabled","dataReductionProgressRate":null}],
"totalCount":1,"hasNext":false,"enumerateContext":null}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when no information is obtained.

DESCRIPTION

A list of summary information of the volume and the information about data.

PROPERTIES

data: ***object[]***

ITEMS

volumeSummary: object (on page 744)

totalCount: *integer*

The total number of records.

hasNext: *boolean*

Indicates whether there is information dropped from the data.

enumerateContext: *string* (uuid)nullable

A token used to obtain the remaining items for the next batch. Null is output if there are no more items to be obtained.

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/volumes

Required Role: Storage or VpsStorage

DESCRIPTION

Creates a volume.

REQUEST BODY

Uses default content-types: application/json

createVolumeParam: object (on page 600)

REQUEST PARAMETERS

None

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
'{ "capacity":1024, "nicknameParam":{ "baseName": "test-volume" },"poolId":"12345678-
9abc-4def-8123-456789abcdef" }' https://192.168.1.100/ConfigurationManager/simple/v1/
objects/volumes
```

The following shows an example using a json body.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
@example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/volumes
```

example.json is defined as follows:

```
{
    "capacity":1024,
    "nicknameParam":
        {
            "baseName": "test-volume"
        },
    "poolId":"12345678-9abc-4def-8123-456789abcdef"
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 409 (on page 577)
Indicates that an inconsistent or impossible request was issued for the resource specified by URL (for example, creating a resource whose ID is same as the resource already created).

HTTP status code 411 (on page 577)
Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

DELETE /v1/objects/volumes/<*id*>

Required Role: Storage or VpsStorage

DESCRIPTION

Deletes the volume.

P-VOL, S-VOL, and P/S-VOL cannot be deleted by the API. Delete them by the following each procedure.

- P-VOL volume: After having deleted all S-VOLs which are created from the target P-VOL by the API for Snapshot delete, delete the volume by the API for volume delete.
- S-VOL volume: Delete the volume by the API for Snapshot delete.
- P/S-VOL volume: After having deleted all S-VOLs which are created from the target P/S-VOL by the API for Snapshot delete, delete the target P/S-VOL by the API for Snapshot delete.

REQUEST BODY

Uses default content-types: application/json

deleteVolumeParam: object (on page 605)

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Volume ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -X DELETE -u admin:password https://192.168.1.100/ConfigurationManager/simple/v1/
objects/volumes/fa476fba-7633-4b51-94f2-e50973782a5c
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

GET /v1/objects/volumes/<*id*>

Required Role: Security, Storage, Monitor, Service, Resource, VpsSecurity, VpsStorage, or VpsMonitor

DESCRIPTION

Obtains the volume information.

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Volume ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -i -u admin:password -X GET https://192.168.1.100/ConfigurationManager/simple/v1/
objects/volumes/35d16c28-b033-4c00-9956-67e8221493d3

HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Cache-Control: private, no-store, no-cache, must-revalidate
Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
Access-Control-Allow-Methods: GET, POST, DELETE, PATCH
Access-Control-Allow-Headers: Content-Type
Content-Type: application/json
Content-Length: 1177
Date: Thu, 20 Oct 2022 05:32:45 GMT

{"reservedCapacity":252,"freeCapacity":47,"luns":[],"snapshotProgressRate":null,
"snapshotTimestamp":null,"snapshotType":null,"savingEffects":{"dataReductionRate":0,
"dataReductionCapacity":0,"compressedCapacity":0,"reclaimedCapacity":0,
"systemDataCapacity":0,"preCapacityDataReductionWithoutSystemData":0,
"postCapacityDataReduction":0},"snapshotConcordanceRate":null,"isWrittenInSvol":null,
"id":"35d16c28-b033-4c00-9956-67e8221493d3","name":"system_vol",
"nickname":"system_vol","volumeNumber":2,"poolId":"7bd0f11a-0d13-4f4e-8f28-
e4b88c8ecc1a","poolName":"SP01","totalCapacity":47,"usedCapacity":0,
"numberOfConnectingServers":0,"numberOfSnapshots":1,"protectionDomainId":"b392fe52-
bcc5-404a-87fa-668fdb55ad62","fullAllocated":false,"volumeType":"Snapshot",
"statusSummary":"Normal","status":"Normal","storageControllerId":"a57046ae-02c7-418b-
b23c-5e38cf42b9c9","snapshotAttribute":"P-VOL","snapshotStatus":null,
"savingSetting":"Disabled","savingMode":null,"dataReductionStatus":"Disabled",
"dataReductionProgressRate":null,"vpsId":"(system)","vpsName":"(system)","qosParam":
{"upperLimitForIops":-1,"upperLimitForTransferRate":-1,"upperAlertAllowableTime":-1,
"upperAlertTime":null}}
```

RESPONSES

Uses default content-types: application/json

200 OK

Indicates that the request was processed correctly. This status code is also returned when the deletion request was executed successfully.

volume: object (on page 733)

HTTP status code 400 (on page 575)

Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)

Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)

Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)

Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)

Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)

Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

PATCH /v1/objects/volumes/<*id*>

Required Role: Storage, or VpsStorage

DESCRIPTION

Edits the volume settings.

REQUEST BODY

Uses default content-types: application/json

patchVolumeParam: object (on page 652)

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Volume ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d '{ "nickname": "newName" }' https://192.168.1.100/ConfigurationManager/simple/v1/
objects/volumes/f77278ae-c86a-451d-b309-605fa63b5ff5
```

The following shows an example using a json body.

```
curl -u admin:password -i -X PATCH -H "Content-Type: application/json" -H "Expect:" -
d @example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/volumes/
f77278ae-c86a-451d-b309-605fa63b5ff5
```

example.json is defined as follows:

```
{
    "nickname": "newName"
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 413 (on page 577)
Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)
Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)
Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)
Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)
Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)
Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)
Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

POST /v1/objects/volumes/<*id*>/actions/expand/invoke

Required Role: Storage or VpsStorage

DESCRIPTION

Expands the volume capacity.

You can execute this API only for a volume whose "status" is "Normal" or "ExpansionFailed", and "volumeType" is "Normal". The relationship between the status and execution content is as follows.

- If you execute this API for a volume whose "status" is "Normal", the API expands the volume by the capacity specified in the parameter.
- If you execute this API for a volume whose "status" is "ExpansionFailed", the API executes the volume expansion again.

REQUEST BODY

Uses default content-types: application/json

expandVolumeParam: object (on page 613)

REQUEST PARAMETERS

Name	Description	Type	Data type
id	Volume ID.	path	*string* (uuid) required

REQUEST EXAMPLE

The following example shows how a REST API is executed by using curl for Virtual Storage Software block at IP address 192.168.1.100 with the user name as admin and the password as password.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
'{ "additionalCapacity": 1000 }' https://192.168.1.100/ConfigurationManager/simple/v1/
objects/volumes/f77278ae-c86a-451d-b309-605fa63b5ff5/actions/expand/invoke
```

The following shows an example using a json body.

```
curl -u admin:password -i -X POST -H "Content-Type: application/json" -H "Expect:" -d
@example.json https://192.168.1.100/ConfigurationManager/simple/v1/objects/volumes/
f77278ae-c86a-451d-b309-605fa63b5ff5/actions/expand/invoke
```

example.json is defined as follows:

```
{
    "additionalCapacity": 1000
}
```

RESPONSES

Uses default content-types: application/json

HTTP status code 202 (on page 575)
: Indicates that an asynchronous process request was accepted.

HTTP status code 400 (on page 575)
: Indicates that the request header, query parameter, and request body are incorrect.

HTTP status code 401 (on page 575)
: Indicates that no Authorization header is specified for the request header, authentication using the information specified for the Authorization header was unsuccessful, or authentication using a parameter was unsuccessful.

HTTP status code 403 (on page 576)
: Indicates that you are not authorized to perform the operation.

HTTP status code 404 (on page 576)
: Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

HTTP status code 406 (on page 577)
: Indicates that an unsupported media type was specified for the Accept header.

HTTP status code 411 (on page 577)
: Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

HTTP status code 413 (on page 577)
: Indicates that the size of the request body exceeds the acceptable upper limit.

HTTP status code 415 (on page 578)
: Indicates that an unsupported media type was specified for the Content-Type header.

HTTP status code 417 (on page 578)

Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

HTTP status code 431 (on page 578)

Indicates that the size of the request header exceeds the acceptable upper limit.

HTTP status code 500 (on page 578)

Indicates that an internal error occurred on the REST API server or in the target storage cluster.

HTTP status code 502 (on page 578)

Indicates that an internal error occurred on the REST API server.

HTTP status code 503 (on page 579)

Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

HTTP status code 504 (on page 579)

Indicates that the REST API server did not respond.

Authentication schemes

- basic authentication
- session authentication

Appendix A: Response definitions

HTTP status code 200

200 OK: Indicates that the request was processed correctly.

HTTP status code 202

202 Accepted: Indicates that an asynchronous process request was accepted.

job: object (on page 627)

HTTP status code 204

204 No Content: Indicates that the request was processed correctly, but no response data to be returned exists.

HTTP status code 400

400 Bad Request: Indicates that the request header, query parameter, and request body are incorrect.

errorResponse: object (on page 611)

HTTP status code 401

401 Unauthorized: Indicates that the Authorization header is not specified in the request header, the authentication with the information specified in the Authorization header has failed, or the authentication has failed due to parameters.

errorResponse

Header	Description	Data Type
WWW-Authenticate	Indicates that authentication is required. ▪ When basic authentication did not succeed in addition to session creation failures: Basic realm="Block storage" ▪ When session authentication did not succeed: Session realm="Block storage"	*string*

HTTP status code 403

403 Forbidden: Indicates that you are not authorized to perform the operation.

errorResponse: object (on page 611)

HTTP status code 404

404 Not Found: Indicates that the resource specified by the URL was not found, or you are not authorized to see the resource.

errorResponse: object (on page 611)

HTTP status code 405

405 Method Not Allowed: Indicates that an unauthorized HTTP method is specified for the resource specified by the URL.

errorResponse: object (on page 611)

HTTP status code 406

406 Not Acceptable: Indicates that an unsupported media type was specified for the Accept header.

errorResponse: object (on page 611)

HTTP status code 409

409 Conflict: Indicates that an inconsistent or impossible request was issued for the resource specified by URL (for example, creating a resource whose ID is same as the resource already created).

errorResponse: object (on page 611)

HTTP status code 411

411 Length Required: Indicates that the Content-Length header must be specified. It is returned when the request body was not specified and "Content-Length: 0" was not specified as the header parameter.

errorResponse: object (on page 611)

HTTP status code 412

412 Precondition Failed: Indicates that the conditions for executing the API are not met.

errorResponse: object (on page 611)

HTTP status code 413

413 Payload Too Large: Indicates that the size of the request body exceeds the acceptable upper limit.

errorResponse: object (on page 611)

HTTP status code 415

415 Unsupported Media Type: Indicates that an unsupported media type was specified for the Content-Type header.

errorResponse: object (on page 611)

HTTP status code 417

417 Expectation Failed: Indicates that the Expect header is specified incorrectly, or the REST API server does not support the Expect header.

errorResponse: object (on page 611)

HTTP status code 431

431 Request Header Fields Too Large: Indicates that the size of the request header exceeds the acceptable upper limit.

errorResponse: object (on page 611)

HTTP status code 500

500 Internal Server Error: Indicates that an internal error occurred on the REST API server or in the target storage cluster.

errorResponse: object (on page 611)

HTTP status code 502

502 Bad Gateway: Indicates that an internal error occurred on the REST API server.

errorResponse: object (on page 611)

HTTP status code 503

503 Service Unavailable: Indicates that the request cannot be received because the REST API server or the target storage cluster is busy. This is a temporary problem, and the request might be processed normally if it is executed again after a while.

errorResponse: object (on page 611)

HTTP status code 504

504 Gateway Timeout: Indicates that the REST API server did not respond.

errorResponse: object (on page 611)

Appendix B: Schema definitions

addHbasServerParam: *object*

DESCRIPTION

A parameter for registering the initiator of the compute node.

PROPERTIES

protocol: *string* , x ∈ { "FC" , "iSCSI" } required

Initiator connection protocol.

hbaWwn: *string* (16 chars) , must match /^[a-f0-9]{16}$/

The WWN of the initiator. Required if FC is specified for protocol. HTTP status code 400 (Bad Request) is returned if FC is specified for protocol and the WWN is not specified for hbaWwn, or iSCSI is specified for protocol and the WWN is specified for hbaWwn.

iscsiName: *string* , must match /^((iqn\.[0-9]{4}\-[0-9]{2}\.[a-zA-Z0-9\-:\.]{0,211})|(eui\.[0-9a-fA-F]{16}))$/

The iSCSI name of the initiator. Required if iSCSI is specified for protocol. HTTP status code 400 (Bad Request) is returned if iSCSI is specified for protocol and iSCSI name is not specified for iscsiName, or FC is specified for protocol and the iSCSI name is specified for iscsiName.

vpsId: *string* , must match /^system$|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

The ID of the operation-target virtual private storage (VPS).

To specify a resource that does not belong to the VPS, set "system" for this property.

If this property is omitted, the VPS that the user who runs the API belongs to is assumed to be the operation target.

addPathServerParam: *object*

DESCRIPTION

A parameter for adding the compute node path information.

he operation depends on whether portId and hbaId are specified, as follows:

- When both portId and hbaId are omitted: All the initiators of the applicable compute nodes are allocated to all the compute ports.
- When only hbaId is specified: The specified initiator is allocated to all the compute ports.
- When only portId is specified: The specified compute port is allocated to all the initiators of the applicable compute nodes.
- When both portId and hbaId are specified: The initiator specified in hbaId is allocated to the compute port that is specified in portId.

PROPERTIES

hbaId: *string* (uuid)

The initiator ID of the compute node.

portId: *string* (uuid)

The ID of the allocation destination compute port for the target operation.

vpsId: *string* , must match /^system$|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

The ID of the operation-target virtual private storage (VPS).

To specify a resource that does not belong to the VPS, set "system" for this property.

If this property is omitted, the VPS that the user who runs the API belongs to is assumed to be the operation target.

addUserToUserGroupParam: *object*

DESCRIPTION

A parameter for adding a user to a user group.

PROPERTIES

userGroupIds: *string[]* (1 to 8 items) required

A list of user group IDs to which a user is to be added.

If the IDs of all virtual private storages (VPSs) that the specified user group belongs to and the IDs of all VPSs that the target user belongs to do not match, HTTP status code 400 (Bad Request) is returned.

ITEMS

string (1 to 64 chars) , must match /^[a-zA-Z0-9!#\$%&'\-\.@\^_`\{\}~]{1,64}$/

approveChapUserParam: *object*

DESCRIPTION

A parameter for permitting access to the compute port in CHAP authentication.

PROPERTIES

chapUserId: *string* (uuid) required

The ID of the CHAP user who is allowed to access the compute port in CHAP authentication.

approvedChapUser: *object*

DESCRIPTION

Information about a CHAP user who is allowed to access the compute port.

PROPERTIES

id: *string* (uuid)

The ID of a CHAP user who is allowed to access the compute port.

targetChapUserName: *string* (1 to 223 chars) , must match /^[a-zA-Z0-9\.:@_\-\+=\[\]~]{1,223}$/

CHAP user name used for CHAP authentication on the compute port (i.e., target side).

initiatorChapUserName: *string* (up to 223 chars) , must match /^$|^[a-zA-Z0-9\.:@_\-\+=\[\]~]{0,223}$/

CHAP user name used for CHAP authentication on the initiator side of the compute node.

An empty string "" is output if no address is set.

assignToServerVolumeParam: *object*

DESCRIPTION

A parameter for connecting a compute node and a volume.

The job succeeds only when this parameter is specified with the following property combinations.

If none of the following is specified, job does not succeed.

- volumeId, serverId, lun
- volumeIds, serverIds, startLun

If volumeIds and serverIds are specified, all the volumes specified in volumeIds are allocated to all the compute nodes specified in serverIds.

PROPERTIES

volumeId: *string* (uuid)

Volume ID to be allocated. This parameter is required if serverId is specified.

volumeIds: *string[]* (1 to 1000 items)

A list of volume IDs to be allocated. This parameter is required if serverIds is specified.

ITEMS

string (uuid)

serverId: *string* (uuid)

The ID of the allocation destination compute node. This parameter is required if volumeId is specified.

serverIds: *string[]* (1 to 100 items)

A list of IDs of the allocation destination compute nodes. This parameter is required if volumeIds is specified.

ITEMS

string (uuid)

lun: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 8191 }

LUN. If omitted, unused LUNs are allocated automatically in ascending order.

startLun: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 8191 }

The start number of the LUN to be allocated. Unused LUNs equal to or greater than the specified value are allocated. If omitted, unused LUNs are allocated automatically in ascending order.

When the sum of the number of volumeIds elements minus one and startLun exceeds 8191, HTTP status code 400 (Bad request) is returned.

vpsId: *string* , must match /^system$|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

The ID of the operation-target virtual private storage (VPS).

To specify a resource that does not belong to the VPS, set "system" for this property.

If this property is omitted, the VPS that the user who runs the API belongs to is assumed to be the operation target.

auditLogSetting: *object*

DESCRIPTION

Audit log settings.

PROPERTIES

syslogForwardingSetting: **syslogForwardingSettingOfAuditLogSetting: object (on page 713)**

blockStorageNodeForMaintenance: *object*

DESCRIPTION

A parameter for storage node maintenance blockade.

PROPERTIES

None

capacitiesExcludingSystemDataOfStorageController: *object*

DESCRIPTION

Pool usage by a storage controller.

When no monitoring information can be collected, null is output.

PROPERTIES

usedVolumeCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 6871947674 } nullable
Consumed capacity of volumes (unit: MiB).

compressedCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 6871947674 } nullable
Amount reduced by compression (unit: MiB).

reclaimedCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 6871947674 } nullable
Amount reduced by fixed pattern exclusion (unit: MiB).

systemDataCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 6871947674 } nullable
Amount of system data consumed by the data reduction function (amount of metadata and garbage data) (unit: MiB).

preUsedCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 6871947674 } nullable
Amount of data to be reduced (unit: MiB).

preCompressedCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 6871947674 } nullable
Amount of data to be compressed (unit: MiB).

capacityBalancingSetting: *object*

DESCRIPTION

Capacity balancing settings.

PROPERTIES

isEnabled: *boolean*

Enables or disables the capacity balancing. If enabled, capacity balancing applies to the storage node. If disabled, capacity balancing does not apply to the storage node.

Even if capacity balancing is enabled, snapshot volumes (P-VOL, P/S-VOL, or S-VOL) are not moved between storage nodes by capacity balancing.

For details, contact customer support.

capacityBalancingSettingParam: *object*

DESCRIPTION

A parameter for setting capacity balancing.

PROPERTIES

isEnabled: *boolean* required

Enables or disables the capacity balancing. If enabled, capacity balancing applies to the storage node. If disabled, capacity balancing does not apply to the storage node.

Even if capacity balancing is enabled, snapshot volumes (P-VOL, P/S-VOL, or S-VOL) are not moved between storage nodes by capacity balancing.

For details, contact customer support.

changePasswordParam: *object*

DESCRIPTION

A parameter for changing the password of the local user.

PROPERTIES

currentPassword: *string* (1 to 256 chars) , must match /^[\-A-Za-z0-9!#\$%&"'\(\)*\+,\.\/:;<>=\?@\[\]\\\^_`\{\}\|~]{1,256}$/ required

Current password.

newPassword: ***string*** **(1 to 256 chars) , must match /^[\-A-Za-z0-9!#\$%&"'\(\)*\+,\.\/:;<>=\?@\[\]\\\^_`\{\}\|~]{1,256}$/ required**

New password. The specifiable values can be restricted by policy. The regular expressions described here show the case where the input restrictions by policy are the least strict.

chapUser: *object*

DESCRIPTION

Information about the CHAP user.

PROPERTIES

portIds: ***string[]***

A list of compute port IDs to which the CHAP user is allowed to access in CHAP authentication.

ITEMS

string (uuid)

id: ***string*** **(uuid)**

The ID of the CHAP user.

targetChapUserName: ***string*** **(1 to 223 chars) , must match /^[a-zA-Z0-9\.:@_\-\+=\[\]~]{1,223}$/**

CHAP user name used for CHAP authentication on the compute port (i.e., target side).

initiatorChapUserName: ***string*** **(up to 223 chars) , must match /^$|^[a-zA-Z0-9\.:@_\-\+=\[\]~]{1,223}$/**

CHAP user name used for CHAP authentication on the initiator side of the compute node.

An empty string "" is output if no address is set.

chapUserSummary: *object*

DESCRIPTION

Summary information about the CHAP user.

PROPERTIES

id: ***string*** **(uuid)**

The ID of the CHAP user.

targetChapUserName: *string* (1 to 223 chars) , must match /^[a-zA-Z0-9\.:@_\-\+=\[\]~]{1,223}$/

CHAP user name used for CHAP authentication on the compute port (i.e., target side).

initiatorChapUserName: *string* (up to 223 chars) , must match /^$|^[a-zA-Z0-9\.:@_\-\+=\[\]~]{1,223}$/

CHAP user name used for CHAP authentication on the initiator side of the compute node.

An empty string "" is output if no address is set.

configureStorageControllerParam: *object*

DESCRIPTION

Parameters for editing storage controller settings.

PROPERTIES

storageControllerId: *string* (uuid)

The ID of the storage controller. If omitted, configure the settings for all storage controllers.

isDetailedLoggingMode: *boolean* required

Enable or disable the detailed logging mode. Specify true for such parameters only when instructed to do so by customer support.

> ⚠ **Caution:**
>
> Enabling the detailed logging mode degrades I/O performance.

controlPort: *object*

DESCRIPTION

Control port information.

PROPERTIES

id: *string* (uuid)

Control port ID.

storageNodeId: *string* (uuid)

The ID of the storage node that has control ports.

macAddress: *string*

MAC address used for communication.

(Bare metal) The MAC address of the primary port is always used when teaming is enabled.

mtuSize: *integer* (int32)

MTU size.

interfaceName: *string*

An interface name, which is unique within a storage node that contains compute ports, control ports, and ports between storage nodes. Example: eth0, eth1

deviceName: *string* (1 to 4096 chars)

Device name of the NIC.

configuredPortSpeed: *string* , x ∈ { "Auto" , "1G" , "10G" , "25G" , "40G" , "100G" }

Link speed setting of the physical port used for communication (unit: bps). This setting determines the actual link speed and duplex settings.

- Auto: The speed and duplex settings depend on the switch or SFP specifications.

portSpeedDuplex: *string* , x ∈ { "10Mbps Half" , "10Mbps Full" , "100Mbps Half" , "100Mbps Full" , "1Gbps Half" , "1Gbps Full" , "2.5Gbps Full" , "5Gbps Full" , "10Gbps Full" , "20Gbps Full" , "25Gbps Full" , "40Gbps Full" , "50Gbps Full" , "56Gbps Full" , "100Gbps Full" , "200Gbps Full" , "400Gbps Full" , "Unknown" , "LinkDown" , "DependsOnHypervisor" }

Actual link speed and duplex settings of the physical port used for communication.

If configuredPortSpeed is Auto, a value is output as per the actual switch or SFP specifications.

- Unknown: The status is unknown.
- LinkDown: Link down occurred.
- DependsOnHypervisor: Depends on the hypervisor setting.

(Virtual machine) DependsOnHypervisor is always output.

isTeamingEnabled: *string* , x ∈ { "true" , "false" , "DependsOnHypervisor" }

Indicates whether teaming is enabled/disabled.

- true: Teaming is enabled.
- false: Teaming is disabled.
- DependsOnHypervisor: Depends on the hypervisor setting.

(Virtual machine) DependsOnHypervisor is always output.

ipv4Information: ipv4InformationOfControlPort: object (on page 622)

teaming: teamingOfInternodePort (on page 622) nullable

(Virtual machine) Null is always output.

(Bare metal) The teaming information of the control port is output.

redundancy: *integer* (int32) , { x ∈ ℤ | -1 ≤ x ≤ 1 } nullable
Redundancy of the physical port. If isTeamingEnabled is set to "DependsOnHypervisor", null is output. If isTeamingEnabled is set to "false", 0 is output.

- -1: No redundancy. (Both systems are down.)
- 0: No redundancy. (One system is down.)
- 1: There is redundancy.

status: *string* , x ∈ { "Normal" , "Warning" , "Error" }
Status of the control port.

(Virtual machine) The output is always Normal.

(Bare metal)

- Normal: Operating normally.
- Warning: There is a problem in one port.
- Error: There is a problem in both ports.

statusSummary: *string* , x ∈ { "Normal" , "Warning" , "Error" }
Status summary of the control port.

(Virtual machine) The output is always Normal.

(Bare metal)

- Normal: No action by the user is required.
- Warning: Although immediate action by the user is not required, some action may have to be taken.
- Error: Immediate action by the user is required.

controlPortPerformance: *object*

DESCRIPTION

Performance information of the control port (monitor information). Infinite value, Infinity is not output as the double type value.

PROPERTIES

id: *string* (uuid)
The ID of the control port.

receiveTransferRate: *number* (double)
Amount of data received per second (unit: MiB/sec).

sendTransferRate: *number* (double)
Amount of data sent per second (unit: MiB/sec).

controlPortPerformanceListResponse: *object*

DESCRIPTION

A list of control port performance information (monitor information) at the specified time.

PROPERTIES

data: *object[]*
ITEMS
controlPortPerformanceListResponseData: object (on page 590)

controlPortPerformanceListResponseData: *object*

DESCRIPTION

The performance information (monitor information) of the control port when the information was collected. For performanceObjects, control ports of which performance information could be collected when the API was executed are output as an array.

PROPERTIES

timestamp: *string* (date-time)
The time when this information was collected.

performanceObjects: *object[]*
A list of control port performance information (monitor information) collected at the time specified for timestamp.

ITEMS
controlPortPerformance: object (on page 589)

cpuPerformance: *object*

DESCRIPTION

Usage of the CPU core.

PROPERTIES

index: ***number*** **(int32)**
CPU core number.

process: ***string*** **, x ∈ { "StorageController" , "SystemController" }**
Process which is using the CPU core. Either of the following is output:

- StorageController: Storage controller's process
- SystemController: System managing process

usage: ***number*** **(double)**
Usage of the CPU core (unit: %).

createChapUserParam: *object*

DESCRIPTION

A parameter for creating a CHAP user.

PROPERTIES

targetChapUserName: ***string*** **(1 to 223 chars) , must match /^[a-zA-Z0-9\.:@_\-\+=\[\]~]{1,223}$/ required**
CHAP user name used for CHAP authentication on the compute port (i.e., target side).

Specify a name different from other targetChapUserName.

targetChapSecret: ***string*** **(12 to 32 chars) , must match /^[a-zA-Z0-9\.:@_\-\+=\/\[\]~]{12,32}$/ required**
CHAP secret used for CHAP authentication on the compute port (i.e., target side).

initiatorChapUserName: ***string*** **(1 to 223 chars) , must match /^[a-zA-Z0-9\.:@_\-\+=\[\]~]{1,223}$/**
CHAP user name used for CHAP authentication on the initiator port of the compute node in mutual CHAP authentication.

initiatorChapSecret: ***string*** **(12 to 32 chars) , must match /^[a-zA-Z0-9\.:@_\-\+=\/\[\]~]{12,32}$/**
CHAP secret used for CHAP authentication on the initiator port of the compute node in mutual CHAP authentication.

createdSession: *object*

DESCRIPTION

Session information.

PROPERTIES

token: *string* (1 to 256 chars) , must match /^[A-Za-z0-9!"#\$%&'\(\)*\+,\-\.\/:;<=>\?@\[\]\\\^_`\{\|\}~]{1,256}$/

Authentication token.

sessionId: *string* (uuid)

Session ID.

userId: *string* (5 to 255 chars) , must match /^[\-A-Za-z0-9!#\$%&'\.@\^_`\{\}~]{5,255}$/

User ID.

userObjectId: *string* (5 to 765 chars) , must match /^[A-Za-z0-9%\-\._~]{5,765}$/

Object ID of the user. For the user ID, a percent-encoded reserved string defined in RFC 3986 is output.

expirationTime: *string* (date-time)

Expiration time of the session. After this time, the session expires.

createdTime: *string* (date-time)

Date and time when the session is generated.

lastAccessTime: *string* (date-time)

Date and time when the session was last used.

roleNames: *string[]*

A list of roles assigned to the user who retains this session.

ITEMS

- For system administrators: (One to six items)

  string , x ∈ { "Security", "Storage", "Monitor", "Service", "Audit", "Resource" }

- For VPS administrators: (One to three items)

  string , x ∈ { "VpsSecurity", "VpsStorage", "VpsMonitor" }

vpsId: *string* , must match /^\(system\)$|[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

The ID of the virtual private storage (VPS) that the user belongs to.

If the user is a system administrator, the string "(system)" is displayed.

privileges: *object[]* (1 to 65 items)

A list of information about a virtual private storage (VPS) that the user can access.

ITEMS

userPrivileges: object (on page 727)

createDumpFileParam: *object*

DESCRIPTION

A parameter for creating a dump log file of the storage node.

PROPERTIES

label: ***string*** **(default: "") (up to 64 chars) , must match /^$|^[a-zA-Z0-9!#$%&\'\-\.@\^_`\{\}~]{1,64}$/**

Label name assigned to the file name as dump log file identifying information.

mode: ***string*** **, x ∈ { "Base" (default) , "Memory" , "Monitor" , "All" }**

Determines information to be included in the dump log files. For modes to be selected, see *Procedure for collecting Virtual Storage Software blockdump log files* in the *Hitachi Virtual Storage Software Block System Administrator Guide*

- Base: Obtains log files required for initial analysis. Information common to all storage nodes, such as configuration information, is obtained only by cluster master node (primary).

- Memory: Obtains memory dump files.

- Monitor: Obtains monitor information. This mode can be specified only for cluster master node (primary) or cluster master node (secondary).

- All: Obtains all the log files. Information common to all storage nodes, such as configuration information and monitor information, is obtained only by cluster master node (primary).

createServerParam: *object*

DESCRIPTION

A parameter for registering compute node information.

PROPERTIES

serverNickname: ***string*** **(1 to 229 chars) , must match /^[a-zA-Z0-9,\.:@_]([a-zA-Z0-9,\.:@_\-]{0,228})$/ required**

The compute node nickname. The same nickname cannot be set for other compute nodes.

osType: *string* , x ∈ { "Linux" , "VMware" , "Windows" } required
The OS type of the compute node.

vpsId: *string* , must match /^system$|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/
The ID of the operation-target virtual private storage (VPS).

To specify a resource that does not belong to the VPS, set "system" for this property.

If this property is omitted, the VPS that the user who runs the API belongs to is assumed to be the operation target.

createSessionParam: *object*

DESCRIPTION

A parameter for generating a session.

PROPERTIES

aliveTime: *integer* (int64) , { x ∈ ℤ | 1 ≤ x ≤ 300 }
The time until a session times out (unit: seconds).

If omitted, the value specified for maxIdleSeconds in user authentication setting is applied.

createSnapshotParam: *object*

DESCRIPTION

A parameter for the Snapshot creation.

masterVolumeId and snapshotVolumeId cannot be specified at the same time. If you specify both masterVolumeId and snapshotVolumeId, HTTP status code 400 (Bad Request) is returned.

PROPERTIES

name: *string* (1 to 32 chars) , must match /^[\-A-Za-z0-9,\.:@_]{1,32}$/
Volume name. Set a name unique throughout the volumes.

If omitted, "Volume<*volumeNumber*>" is registered based on volumeNumber of the created volume. However, if the "Volume<*volumeNumber*>" already exists, "Volume<*volumeNumber*>_<*unique-value*>" is registered. This unique value is the smallest in the range of 1 to 1048576 that does not overlap with an existing volume name when the volume is created.

masterVolumeId: *string* (uuid)
The ID of the volume becoming P-VOL.

This parameter must be specified when nothing, Prepare, or PrepareAndFinalize is specified for operationType. Otherwise, HTTP status code 400 (Bad Request) is returned.

snapshotVolumeId: *string* (uuid)

The ID of the volume that is an S-VOL. You can specify only volumes whose snapshotAttribute and snapshotStatus are S-VOL and Prepared, respectively. If you specify Finalize for operationType, this parameter must be specified. Otherwise, HTTP status code 400 (Bad Request) is returned.

operationType: *string* , x ∈ { "Prepare" , "Finalize" , "PrepareAndFinalize" (default) }

Operation type.

- Prepare: Creates an empty S-VOL corresponding to the volume specified for masterVolumeId as a preliminary operation.
- Finalize: Starts recording difference data in relation to the volume specified for masterVolumeId using the S-VOL created in the preliminary operation.
- PrepareAndFinalize: Performs both Prepare and Finalize operations.

qosParam: volumeQosParam: object (on page 743)

vpsId: *string* , must match /^system$|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

The ID of the operation-target virtual private storage (VPS).

To specify a resource that does not belong to the VPS, set "system" for this property.

If this property is omitted, the VPS that the user who runs the API belongs to is assumed to be the operation target.

createSpareNodeParam: *object*

DESCRIPTION

A parameter for registering spare node information.

PROPERTIES

faultDomainId: *string* (uuid) required

The ID of the fault domain that the nodes belong to.

controlPortIpv4Address: *string* (7 to 15 chars) , must match /^(([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])$/ required

The IP address (IPv4) of the control port.

setupUserPassword: *string* (1 to 256 chars) , must match /^[\-A-Za-z0-9!#\$%&"'\(\)*\+,\.\/:;<>=\?@\[\]\\\^_`\{\}\|~]{1,256}$/

Setup user password.

bmcName: *string* (1 to 253 chars) , must match /^([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9])\.)*([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9]))$/

The host name or IP address (IPv4) of the BMC.

bmcUser: *string* (1 to 512 chars) , must match /^[a-zA-Z0-9!"#\$%&'\(\)*\+,\-\.\/:;<=>\?@\[\\\]\^_`\{\|\}~]{1,512}$/

The user name for BMC connection.

bmcPassword: *string* (1 to 512 chars) , must match /^[a-zA-Z0-9!"#\$%&'\(\)*\+,\-\.\/:;<=>\?@\[\\\]\^_`\{\|\}~]{1,512}$/

The password for BMC connection.

createTicketParam: *object*

DESCRIPTION

A parameter for issuing an authentication ticket.

PROPERTIES

maxAgeDays: *integer* (int32) , { x ∈ ℤ | 1 ≤ x ≤ 365 }

Valid period of the ticket (unit: number of days). Date and time of ticket issuance + valid period becomes the expiration time.

If omitted, the ticket expiration date and time are the same as that for the password of the user who issued the ticket.

However, if the user's password does not have an expiration time, the ticket will be valid for 365 days.

createUserGroupParam: *object*

DESCRIPTION

A parameter for creating a user group.

PROPERTIES

userGroupId: *string* (1 to 64 chars) , must match /^[a-zA-Z0-9!#\$%&'\-\.@\^_`\{\}~]{1,64}$/

User Group ID.

roleNames: *string[]* required

Role of the user group. At least one role must be specified. If no role was specified, HTTP status code 400 (Bad Request) is returned.

ITEMS

- For system administrators: (Specify one to six items)

  string , x ∈ { "Security", "Storage", "Monitor", "Service", "Audit", "Resource" }

- For VPS administrators: (Specify one to three items)

  string , x ∈ { "VpsSecurity", "VpsStorage", "VpsMonitor" }

externalGroupName: *string* (1 to 4096 chars)

Name of the group registered with an external authorization server when the external authorization server is linked. When assigning a role to a group in an external authorization server, specify a group name in the external authorization server for this attribute. If omitted, this user group is not linked to the external authorization server.

vpsId: *string* , must match /^system$|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

The ID of the virtual private storage (VPS) that the user group belongs to.

To create a system-administrator user group, set "system".

If this property is omitted, the ID of the VPS that the user who runs the API belongs to is automatically set. If the user who runs the API is a system administrator, "system" is set.

scope: *string[]* (1 to 65 items)

An array of the IDs of virtual private storages (VPSs) that the user group can access.

This item can be omitted. If this property is omitted, a single-element array consisting of only the ID of the VPS that the user who runs the API belongs to is automatically set. If the user who runs the API is a system administrator, "system" is set.

Because a system administrator can access multiple VPSs, if this parameter has the "system" element, multiple IDs of VPSs can be specified. A VPS administrator, who can access only one VPS, can specify only one ID.

ITEMS

string , must match /^system$|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

The ID of the VPS that the user group belongs to cannot be omitted. If it is not included, HTTP status code 400 (Bad Request) is returned.

If the vpsId property is omitted and its setting is automatically specified, the ID of the VPS that the user who runs the API belongs to must be included. If it is not included, HTTP status code 400 (Bad Request) is returned.

In the case of a system administrator group where the ID of the VPS that the user group belongs to is "system", any VPSs can be specified. In the case of a VPS administrator group where the ID of the VPS that the user group belongs to is not "system", only one array element can be specified. If two or more elements are specified, HTTP status code 400 (Bad Request) is returned.

createUserParam: *object*

DESCRIPTION

A parameter for creating a user.

PROPERTIES

userId: *string* (6 to 255 chars) , must match /^[\-A-Za-z0-9!#\$%&'\.@\^_`\{\}~]{6,255}$/ required

User ID.

isEnabledConsoleLogin: *boolean* (6 to 28 chars) , must match /^[A-Za-z_][-A-Za-z0-9._]{5,27}$/

If isEnabledConsoleLogin is set to true, the user ID must be 6 to 28 characters.

password: *string* (1 to 256 chars) , must match /^[\-A-Za-z0-9!#\$%&"'\(\)*\+,\.\/:;<>=\?@\[\]\\\^_`\{\}\|~]{1,256}$/

User password. The specifiable values can be restricted by policy. The regular expressions described here show the case where the input restrictions by policy are the least strict.

Either of the following conditions must be met in combination with authentication:

- Nothing is specified for authentication, or local is specified for authentication and password is specified.
- external is specified for authentication, and password is specified.

HTTP status code 400 (Bad Request) is returned when none of the conditions is satisfied.

userGroupIds: *string[]* (1 to 8 items) required

A list of user group IDs to which the user belongs.

The user will be created on the virtual private storages (VPSs) that the specified user groups belong to.

In the following cases, HTTP status code 400 (Bad Request) is returned:

- The IDs of the all VPSs that the specified user groups belong to are not the same.
- Access permission for the VPSs that the specified user group belongs to are not assigned.
- The maximum allowable number of users on the VPSs that the specified user group belongs to is exceeded.

ITEMS

string (1 to 64 chars) , must match /^[a-zA-Z0-9!#\$%&'\-\.@\^_`\{\}~]{1,64}$/

authentication: *string* , x ∈ { "local"(default), "external" }
Authentication type.

- local: Authenticated locally.
- external: Authenticated by the external authentication server.

isEnabledConsoleLogin: *boolean*
(Virtual machine) This option will be ignored.

(Bare metal) Whether the use of console interface is permitted.

- true: The console interface can be used.
- false: The console interface cannot be used.

If this option is set to true when the specified user group belongs to a virtual private storage (VPS) is other than "system", HTTP status code 400 (Bad Request) is returned.

createVirtualPrivateStorageParam: *object*

DESCRIPTION

A parameter for creating a virtual private storage (VPS).

PROPERTIES

name: *string* (1 to 32 chars) , must match /^(?!system$)[\-A-Za-z0-9,\.:@_]{1,32}$/ required
A VPS name.

The string "system" cannot be specified, and an existing VPS cannot be named.

upperLimitForNumberOfUserGroups: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 256 }
The maximum number of user groups that can belong to the VPS. If this property is omitted, 256 is assumed to be specified.

upperLimitForNumberOfUsers: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 256 }
The maximum number of users that can belong to the VPS. If this property is omitted, 256 is assumed to be specified.

upperLimitForNumberOfSessions: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 436 }
The maximum number of sessions that can be established for the VPS. If this property is omitted, 436 is assumed to be specified.

upperLimitForNumberOfServers: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 1024 } required
The maximum allowable number of compute nodes on the VPS.

volumeSettings: *object[]* required

The volume settings for the VPS.

ITEMS

VirtualPrivateStorageDefaultVolumeSettings: object (on page 730)

createVolumeParam: *object*

DESCRIPTION

A parameter for creating volumes.

Either of nameParam and nicknameParam must be specified. If neither of them is specified, HTTP status code 400 (Bad Request) is returned.

If nameParam is specified and nicknameParam is omitted, the same character string as name is registered in nickname of each volume.

If nicknameParam is specified and nameParam is omitted, the following is stored in name of each volume:

- The same character string as nickname if there is no other volume whose name is the same as the nickname string.
- "Volume<*volumeNumber-of-created-volume*>" if a volume whose name is the same as the nickname string exists.

  However, "Volume<*volumeNumber-of-created-volume*>_<*unique-value*>" if "<*volumeNumber-of-created-volume*>" already exists.

  This unique value is the smallest in the range of 1 to 1048576 that does not overlap with an existing volume name when the volume is created.

PROPERTIES

capacity: *integer* (int32) , { x ∈ ℤ | 47 ≤ x ≤ 268435456 } required

The (logical) capacity of the volume (unit: MiB).

For the value to be specified in "capacity", contact customer support.

For the value to be specified in "capacity", see *Creating volumes* in the *Hitachi Virtual Storage Software Block Storage Administrator Guide*.

number: *integer* (int32) , { x ∈ ℤ | 1 ≤ x ≤ 1000 } (default: "1")

The number of volumes to be created.

nameParam: volumeNameParam: object (on page 739)

nicknameParam: volumeNicknameParam: object (on page 740)

poolId: *string* (uuid)

The ID of the storage pool where the volume is created.

If the user who runs the API has a system administrator role, either the poolId or vpsId property must be specified. Both of these properties cannot be specified at the same time.

If the user who runs the API has a VPS administrator role, this property cannot be specified.

vpsId: ***string*** **, must match /^system$|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/**

The ID of the operation-target virtual private storage (VPS).

To specify a resource that does not belong to a VPS, specify it by using the poolId property without specifying the vpsId property.

If this property is omitted, the VPS that the user who runs the API belongs to is assumed to be the operation target.

If the user who runs the API has a system administrator role, either the poolId or vpsId property must be specified. Both of these properties cannot be specified at the same time.

If the user who runs the API has a VPS administrator role, this property can be omitted.

storageControllerId: ***string*** **(uuid)**

The ID of the storage controller that manages the volume.

qosParam: **createVolumeQosParam: object (on page 601)**

createVolumeQosParam: *object*

DESCRIPTION

A QoS-related parameter.

PROPERTIES

upperLimitForIops: ***integer*** **(int64) , { x ∈ ℤ | -1 ≤ x ≤ 2147483647 }**

The upper limit of volume performance (in IOPS).

To set the upper limit of volume performance (in IOPS), specify a value in the range from 100 to 2147483647. To set no upper limit, specify -1. If you specify values from 0 to 99, jobs will be unsuccessful.

If you make both of the upperLimitForIops and upperLimitForTransferRate settings unavailable, the setting of upperAlertAllowableTime is also made unavailable.

If this property is omitted, following valiue is specified.

- If the volume belongs to a virtual private storage (VPS): Default value set for the VPS
- If the volume does not belong to a VPS: -1

upperLimitForTransferRate: ***integer*** **(int64) , { x ∈ ℤ | -1 ≤ x ≤ 2097151 }**

The upper limit of volume performance (in MiB/s).

To set the upper limit of volume performance (in MiB/s), specify a value in the range from 1 to 2097151. To set no upper limit, specify -1. If you specify 0, jobs will be unsuccessful.

If you make both of the upperLimitForIops and upperLimitForTransferRate settings unavailable, the setting of upperAlertAllowableTime is also made unavailable.

If this property is omitted, following valiue is specified.

- If the volume belongs to a virtual private storage (VPS): Default value set for the VPS
- If the volume does not belong to a VPS: -1

upperAlertAllowableTime: ***integer*** **(int32) , { x ∈ ℤ | -1 ≤ x ≤ 600 }**

The alert threshold value (in seconds) for the upper limit of volume performance.

The VPS administrator cannot specify this property.

To set the alert threshold, specify a value in the range from 1 to 600. To set no alert threshold, specify -1. If you specify 0, jobs will be unsuccessful.

A message is output to the event log when restriction of the upper limit of performance specified by upperLimitForIops or upperLimitForTransferRate continues for the specified length of time.

This property can be specified if either upperLimitForIops or upperLimitForTransferRate, or both, is set.

If this property is omitted, following valiue is specified.

- If the volume belongs to a virtual private storage (VPS): Default value set for the VPS
- If the volume does not belong to a VPS: -1

deleteHbasServerParam: *object*

DESCRIPTION

A parameter for deleting the initiator information of compute nodes.

PROPERTIES

vpsId: *string* , must match /^system$|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

The ID of the operation-target virtual private storage (VPS).

To specify a resource that does not belong to the VPS, set "system" for this property.

If this property is omitted, the VPS that the user who runs the API belongs to is assumed to be the operation target.

deletePathServerParam: *object*

DESCRIPTION

A parameter for deleting the initiator information of compute nodes.

PROPERTIES

vpsId: *string* , must match /^system$|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

The ID of the operation-target virtual private storage (VPS).

To specify a resource that does not belong to the VPS, set "system" for this property.

If this property is omitted, the VPS that the user who runs the API belongs to is assumed to be the operation target.

deleteServerParam: *object*

DESCRIPTION

A parameter for deleting the compute node information.

PROPERTIES

vpsId: *string* , must match /^system$|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

The ID of the operation-target virtual private storage (VPS).

To specify a resource that does not belong to the VPS, set "system" for this property.

If this property is omitted, the VPS that the user who runs the API belongs to is assumed to be the operation target.

deleteSnapshotParam: *object*

DESCRIPTION

A parameter for deleting snapshots.

Either masterVolumeId or snapshotVolumeId must be specified, but they cannot be specified together. If you omit this parameter, or specify both parameters, HTTP status code 400 (Bad Request) is returned.

PROPERTIES

masterVolumeId: *string* (uuid)

Specify the ID of the volume which is P-VOL. Only the volume of which snapshotAttribute is P-VOL can be specified. At the time of masterVolumeId specified, specifying snapshotTree becomes essential.

snapshotVolumeId: *string* (uuid)

Specify the ID of the volume which is S-VOL. Only the volume of which snapshotAttribute is S-VOL can be specified. Deletes the volume which is S-VOL of the parameter specified.

snapshotTree: *boolean* (default: "false")

Specify whether to forcibly delete all S-VOLs and P/S-VOLs that have been created from the P-VOL specified for masterVolumeId.

- When masterVolumeId is specified:
  - true: Deletes all S-VOLs and P/S-VOLs created from the masterVolumeId specified P-VOL.
  - false or unspecified: Returns HTTP status code 400 (Bad Request).
- When snapshotVolumeId is specified:
  - true: Returns HTTP status code 400 (Bad Request).
  - false or unspecified: Deletes the S-VOLs for which snapshotVolumeId is specified.

vpsId: *string* , must match /^system$|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

The ID of the operation-target virtual private storage (VPS).

To specify a resource that does not belong to the VPS, set "system" for this property.

If this property is omitted, the VPS that the user who runs the API belongs to is assumed to be the operation target.

deleteUserFromUserGroupParam: *object*

DESCRIPTION

A parameter for deleting a user from a user group.

PROPERTIES

userGroupIds: *string[]* (1 to 8 items) required
A list of user group IDs from which a user is to be deleted.

ITEMS
string **(1 to 64 chars)** , must match /^[a-zA-Z0-9!#\$%&'\-\.@\^_`\{\}~]{1,64}$/

deleteVolumeParam: *object*

DESCRIPTION

A parameter for deleting volumes.

PROPERTIES

vpsId: *string* , must match /^system$|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/
The ID of the operation-target virtual private storage (VPS).

To specify a resource that does not belong to the VPS, set "system" for this property.

If this property is omitted, the VPS that the user who runs the API belongs to is assumed to be the operation target.

drive: *object*

DESCRIPTION

Drive information.

PROPERTIES

id: *string* (uuid)
Drive ID.

wwid: *string* (1 to 128 chars)

Drive WWID (WWN).

statusSummary: *string* , x ∈ { "Normal" , "Warning" , "Error" }

Drive status summary.

- Normal: No action by the user is required.
- Warning: Although immediate action by the user is not required, some action may have to be taken.
- Error: Immediate action by the user is required.

status: *string* , x ∈ { "Offline" , "Normal" , "TemporaryBlockage" , "Blockage" }

The drive status. Any one of the following statuses is output:

- Offline: Not used. This drive can be removed.
- Normal: Operating normally.
- TemporaryBlockage: Temporarily blocked.
- Blockage: Blocked.

typeCode: *string* (1 to 128 chars)

Drive type code (drive model name).

serialNumber: *string* (1 to 128 chars)

Drive serial number.

storageNodeId: *string* (uuid)

Storage node ID.

deviceFileName: *string*

Device file name for a drive.

vendorName: *string* (1 to 128 chars)

The vendor name.

firmwareRevision: *string* (1 to 128 chars)

Firmware revision.

locatorLedStatus: *string* , x ∈ { "Off ", "On" }

On/off state of the locator LED.

- Off: The LED is off.
- On: The LED is on.

driveType: *string* , x ∈ { "SSD" , "HDD" , "Unknown" }

Drive type.

driveCapacity: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 256000 }
The drive capacity (Unit: GB). 1GB=1,000,000,000 bytes.

driveLocatorLedParam: *object*

DESCRIPTION

Controls on and off of the locator LED of a drive.

PROPERTIES

operationType: *string* , x ∈ { "TurnOff ", "TurnOn" } required
Specifies operation for the locator LED.

- TurnOff: Turns off the LED.
- TurnOn: Turns on the LED.

drivePerformance: *object*

DESCRIPTION

Performance information of the drive (monitor information). Null is set if the information could not be obtained. The infinite value (Infinity) is not output for double type values.

PROPERTIES

id: *string* (uuid)
Drive ID.

readIOPS: *integer* (int32) nullable
The number of read I/Os per second (unit: IOPS).

writeIOPS: *integer* (int32) nullable
The number of write I/Os per second (unit: IOPS).

readTransferRate: *number* (double) nullable
Read data transfer amount per second (unit: MiB/sec).

writeTransferRate: *number* (double) nullable
Write data transfer amount per second (unit: MiB/sec).

responseTime: *number* (double)
Average response time (unit: msec).

usage: *integer* (int32)
Ratio of I/O operating time of the drive against elapsed time (unit: %).

drivePerformanceListResponse: *object*

DESCRIPTION

A list of drive performance information (monitor information) at the specified time.

PROPERTIES

data: *object[]*

ITEMS

drivePerformanceListResponseData: object (on page 608)

drivePerformanceListResponseData: *object*

DESCRIPTION

The performance information (monitor information) of the drive when the information was collected. For performanceObjects, drives of which performance information could be collected when the API was executed are output as an array.

PROPERTIES

timestamp: *string* (date-time)

The time when this information was collected.

performanceObjects: *object[]*

A list of drive performance information (monitor information) collected at the time specified for timestamp.

ITEMS

drivePerformance: object (on page 607)

dumpStatus: *object*

DESCRIPTION

Status of a dump log file creation request.

PROPERTIES

startedTime: *string* (date-time) nullable

Time when dump log file creation started.

completedTime: *string* (date-time) nullable

Time when dump log file creation ended.

label: *string* (up to 64 chars) , must match /^$|^[a-zA-Z0-9!#$%&\'\-\.@\^_`\{\}~]{1,64}$/ nullable

Label name assigned to the file name as dump log file identifying information.

status: *string* , x ∈ { "NotCreated" , "Creating" , "Created" , "Failed" }

Progress of dump log file creation.

- NotCreated: Status in which a dump log file is not created (the dump creation API has never been run).
- Creating: Status in which a dump log file is being created.
- Created: Status in which a dump log file is created successfully and is ready to be downloaded.
- Failed: Status in which dump log file creation did not succeed.

size: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 147456 } nullable

File size that can be downloaded (unit: MiB).

triggerType: *string* , x ∈ { "Manual " , "Auto " } nullable

Trigger type for dump log file creation.

- Manual: A dump log file was created by user's request.
- Auto: A dump log file was automatically created by the system.

mode: *string* , x ∈ { "Base" , "Memory" , "Monitor" , "All" , "AutoCollection" } nullable

Information to be included in the dump log files.

- Base: Obtains log files required for initial analysis.
- Memory: Obtains memory dump files.
- Monitor: Obtains monitor information.
- All: Obtains all the log files.
- AutoCollection: The system automatically obtains log files.

fileName: *string* (1 to 255 chars) , must match /^[a-zA-Z0-9!#$%&\'\-\.@\^_`\{\}~]{1,255}$/ nullable

Name of the dump log file.

numberOfSplitFiles: *integer* (int32) , { x ∈ ℤ | 1 ≤ x ≤ 369 } nullable

The number that the dump log files are split when the files are downloaded separately.

error: errorResponse: object (on page 611) nullable

dumpStatusList: *object*

DESCRIPTION

A list of dump log file creation statuses.

PROPERTIES

data: ***object[]***

ITEMS

dumpStatus: object (on page 608)

emailReportSettingOfEditEventLogSetting: *object*

DESCRIPTION

A parameter for editing the event log transfer destination in email notification settings.

PROPERTIES

smtpSettings: ***object[]*** **(1 item) required**

SMTP setting for event log transfer destination.

ITEMS

smtpSettingOfEditEventLogSetting: object (on page 688)

emailReportSettingOfEventLogSetting: *object*

DESCRIPTION

Event log email notification setting.

PROPERTIES

smtpSettings: ***object[]*** **(1 item)**

A list of SMTP settings.

ITEMS

smtpSettingOfEventLogSetting: object (on page 689)

errorInformationOfLdapServerConnectionVerification: *object*

DESCRIPTION

Information about an error returned by the LDAP server.

PROPERTIES

primaryServer: primaryServerOfLdapServerConnectionVerificationError: object (on page 666) nullable

secondaryServer: secondaryServerOfLdapServerConnectionVerificationError: object (on page 681) nullable

errorResponse: *object*

DESCRIPTION

Error information.

This object is returned with job attribute "error" if attribute "state" is "Failed".

PROPERTIES

errorSource: *string* (link) nullable
URL on which the error occurred.

messageId: *string*
Message ID.

message: *string* (1 to 512 chars)
Error message.

cause: *string* (1 to 512 chars) nullable
Error factor.

solution: *string* (1 to 512 chars) nullable
Error solution.

solutionType: *string* , x ∈ { "RETRY" , "SEE_ERROR_DETAIL" }
Error solution type.

errorCode: *object* nullable
Internal error code. This information is used to contact customer support or for maintenance.

eventLog: *object*

DESCRIPTION

Event log.

PROPERTIES

id: *string* (uuid)
Event ID.

time: *string* (date-time)
The date and time when the event was detected.

timeInMicroseconds: *integer* (int64) , { x ∈ ℤ | x ≥ 0 }
The time (in microseconds) elapsed from January 1, 1970, 00:00:00:00 until when the event was detected.

If more than one event occurred at the same time, which event occurred first can be verified by comparing their "timeInMicroseconds" values.

category: *string*
Event category (for example, drive, storage pool).

eventName: *string*
Event name uniquely given.

messageId: *string*
Message ID.

severity: *string* , x ∈ { "Info" , "Warning" , "Error" , "Critical" , "Unknown" }
The severity of the event. The following severity is output:

- Info: Notified for your information.
- Warning: Immediate action is not required, but this event might cause a major failure in the future.
- Error: Corrective action is required.
- Critical: Fatal failure. Immediate action is required.
- Unknown: Unknown event.

message: *string*
Event description.

solution: *string*
Recommended action for the generated event.

nodeLocation: *string*
Information about a node on which the event occurred.

eventType: ***string***
Event type. This item is reserved for future use. "Informational" is always output.

severityLevel: ***string***
Degree of urgency of the event. This item is reserved for future use. "1" is always output.

eventLogSetting: *object*

DESCRIPTION

Event log settings.

PROPERTIES

syslogForwardingSetting: syslogForwardingSettingOfEventLogSetting: object (on page 715)

emailReportSetting: emailReportSettingOfEventLogSetting: object (on page 610)

expandPoolParam: *object*

DESCRIPTION

A parameter for expanding the storage pool capacity.

PROPERTIES

driveIds: ***string[]*** **(1 to 1024 items) required**
A list of IDs of the drives to be added to the storage pool. HTTP status code 400 (Bad Request) is returned when an empty array ([]) is specified.

ITEMS
string (uuid)

DESCRIPTION
The ID of a drive to be added to the storage pool.

expandVolumeParam: *object*

DESCRIPTION

A parameter for expanding the volume.

PROPERTIES

additionalCapacity: *integer* (int32) , { x ∈ ℤ | 1 ≤ x ≤ 268435456 }

The logical capacity (in MiB) added to the volume.

Execute this API by specifying a value if the status of the value is "Normal", or without specifying "additionalCapacity" if the status is "ExpansionFailed".

For the value of "additionalCapacity" that you specify, see *Expanding volumes* in the *Hitachi Virtual Storage Software Block Storage Administrator Guide*.

vpsId: *string* , must match /^system$|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

The ID of the operation-target virtual private storage (VPS).

To specify a resource that does not belong to the VPS, set "system" for this property.

If this property is omitted, the VPS that the user who runs the API belongs to is assumed to be the operation target.

externalAuthServerConnectionVerification: *object*

DESCRIPTION

Information about the connection with the external authentication server.

PROPERTIES

authProtocol: *string* , x ∈ { "LDAP" }

Authentication protocol used for external authentication.

ldap: ldapServerConnectionVerification: object (on page 630) nullable

externalAuthServerSetting: *object*

DESCRIPTION

Setting of the external authentication server.

PROPERTIES

isEnabled: *boolean*

Enables or disables external authentication.

authProtocol: *string* , x ∈ { "LDAP" }

Authentication protocol used for external authentication.

ldapSetting: ldapSettingOfExternalAuthServerSetting: object (on page 633) nullable

faultDomain: *object*

DESCRIPTION

Fault domain information.

PROPERTIES

storageNodeIds: *string[]* (1 to 32 items)
A list of storage node IDs which belong to the fault domain.

ITEMS
string (uuid)

id: *string* (uuid)
Fault domain ID.

name: *string* (1 to 32 chars) , must match /^[\-A-Za-z0-9_]{1,32}$/
Fault domain name.

statusSummary: *string* , x ∈ { "Normal", "Error"}
Fault domain status summary.

- Normal: No action by the user is required.
- Error: Immediate action by the user is required.

status: *string* , x ∈ { "Normal", "Error"}
The fault domain status.

- Normal: Normal status.
- Error: A failure occurred in at least one storage node in the fault domain.

numberOfStorageNodes: *integer* (int32) , { x ∈ ℤ | 1 ≤ x ≤ 64 }
The number of storage nodes which belong to the fault domain.

faultDomainSummary: *object*

DESCRIPTION

Summary information about the fault domain.

PROPERTIES

id: *string* (uuid)
Fault domain ID.

name: ***string*** **(1 to 32 chars) , must match /^[\-A-Za-z0-9_]{1,32}$/**

Fault domain name.

statusSummary: ***string*** **, x ∈ { "Normal", "Error"}**

Fault domain status summary.

- Normal: No action by the user is required.
- Error: Immediate action by the user is required.

status: ***string*** **, x ∈ { "Normal", "Error"}**

The fault domain status.

- Normal: Normal status.
- Error: A failure occurred in at least one storage node in the fault domain.

numberOfStorageNodes: ***integer*** **(int32) , { x ∈ ℤ | 1 ≤ x ≤ 64 }**

The number of storage nodes which belong to the fault domain.

fcInformationOfEditPort: *object*

DESCRIPTION

A parameter for editing FC settings.

Each value remains as is if nothing is specified for it. FC settings are disabled if iSCSI connection is used.

PROPERTIES

connectionType: ***string*** **, x ∈ { "PointToPoint" }**

Network topology for FC connection.

fcPortPerformance: *object*

DESCRIPTION

FC port performance information. Output only if the target port uses FC connection. In this case, iscsi is null.

PROPERTIES

target: fcTargetPortPerformance: object (on page 617) nullable

fcTarget: *object*

DESCRIPTION

FC port information. Output only if the compute port uses FC connection. In this case, iscsiInformation is null.

PROPERTIES

connectionType: ***string*** **, x ∈ { "PointToPoint" }**
Network connection type.

sfpDataTransferRate: ***string*** **, x ∈ { "8G" , "16G" , "32G" , "Unknown" }**
SFP data transfer rate (unit: bps). "Unknown" is output if the SFP extension port cannot be recognized or nothing is connected to the SFP extension port.

physicalWwn: ***string***
Physical WWN of the compute port if "name" of the compute port is a logical WWN.

fcTargetPortPerformance: *object*

DESCRIPTION

Performance information of the FC port for the target operation. Output only if the compute port is for the target operation.

PROPERTIES

readIOPS: ***integer*** **(int32)**
The number of read I/Os per second (unit: IOPS).

writeIOPS: ***integer*** **(int32)**
The number of write I/Os per second (unit: IOPS).

readTransferRate: ***number*** **(double)**
Read data transfer amount per second (unit: MiB/sec).

writeTransferRate: ***number*** **(double)**
Write data transfer amount per second (unit: MiB/sec).

responseTime: ***number*** **(double)**
Average response time (unit: msec).

hba: *object*

DESCRIPTION

Initiator information of the compute node.

PROPERTIES

id: *string* (uuid)
The ID of the initiator.

serverId: *string* (uuid)
The ID of the compute node.

name: *string*
The initiator WWN for FC connection, or the iSCSI name for iSCSI connection.

protocol: *string* , x ∈ { "FC" , "iSCSI" }
Initiator connection protocol.

portIds: *string[]*
A list of IDs of the allocation destination compute port for the target operation.

ITEMS

string (uuid)

DESCRIPTION

The ID of the allocation destination compute port for the target operation.

vpsId: *string* , must match /^\(system\)$|[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/
The ID of the virtual private storage (VPS) that the acquisition-target resource belongs to.

If the resource does not belong to a VPS, the reserved word "(system)", which indicates that the resource is independent, is output instead of an ID.

vpsName: *string* (1 to 32 chars) , must match /^\(system\)$|[\-A-Za-z0-9,\.:@_]{1,32}$/
The name of the virtual private storage (VPS) that the acquisition-target resource belongs to.

If the resource does not belong to a VPS, the reserved word "(system)", which indicates that the resource is independent, is output.

healthStatus: *object*

DESCRIPTION

Health status.

PROPERTIES

resources: *object[]*

ITEMS

resourceStatusOfHealthStatus: object (on page 674)

installLicenseParam: *object*

DESCRIPTION

A parameter for registering a license.

PROPERTIES

keyCode: *string* , must match /^[A-Z0-9]{75}$/ required

License key.

internodePort: *object*

DESCRIPTION

Internode port information.

PROPERTIES

id: *string* (uuid)

Internode port ID.

storageNodeId: *string* (uuid)

The ID of the storage node that has an internode port.

macAddress: *string*

MAC address used for communication.

(Bare metal) The MAC address of the primary port is always used when teaming is enabled.

mtuSize: *integer* (int32)

MTU size.

interfaceName: *string*

An interface name, which is unique within a storage node that contains compute ports, control ports, and ports between storage nodes. Example: eth0, eth1

deviceName: *string* (1 to 4096 chars)

Device name of the NIC.

configuredPortSpeed: ***string*** **, x ∈ { "Auto" , "1G" , "10G" , "25G" , "40G" , "100G" }**

Link speed setting of the physical port used for communication (unit: bps). This setting determines the actual link speed and duplex settings.

- Auto: The speed and duplex settings depend on the switch or SFP specifications.

portSpeedDuplex: ***string*** **, x ∈ { "10Mbps Half" , "10Mbps Full" , "100Mbps Half" , "100Mbps Full" , "1Gbps Half" , "1Gbps Full" , "2.5Gbps Full" , "5Gbps Full" , "10Gbps Full" , "20Gbps Full" , "25Gbps Full" , "40Gbps Full" , "50Gbps Full" , "56Gbps Full" , "100Gbps Full" , "200Gbps Full" , "400Gbps Full" , "Unknown" , "LinkDown" , "DependsOnHypervisor" }**

Actual link speed and duplex settings of the physical port used for communication.

If configuredPortSpeed is Auto, a value is output as per the actual switch or SFP specifications.

- Unknown: The status is unknown.

- LinkDown: Link down occurred.

- DependsOnHypervisor: Depends on the hypervisor setting.

(Virtual machine) DependsOnHypervisor is always output.

isTeamingEnabled: ***string*** **, x ∈ { "true" , "false" , "DependsOnHypervisor" }**

Indicates whether teaming is enabled/disabled.

- true: Teaming is enabled.

- false: Teaming is disabled.

- DependsOnHypervisor: Depends on the hypervisor setting.

(Virtual machine) DependsOnHypervisor is always output.

ipv4Information: ipv4InformationOfInternodePort: object (on page 623)

teaming: teamingOfInternodePort: object (on page 719) nullable

(Virtual machine) Null is always output.

(Bare metal) The teaming information of the internode port.

redundancy: ***integer*** **(int32) , { x ∈ ℤ | -1 ≤ x ≤ 1 } nullable**

Redundancy of the physical port. If isTeamingEnabled is set to "DependsOnHypervisor", null is output. If isTeamingEnabled is set to "false", 0 is output.

- -1: No redundancy. (Both systems are down.)

- 0: No redundancy. (One system is down.)

- 1: There is redundancy.

status: ***string*** **, x ∈ { "Normal" , "Warning" , "Error" }**
Status of the internode port.

(Virtual machine) The output is always Normal.

(Bare metal)

- Normal: Operating normally.
- Warning: There is a problem in one port.
- Error: There is a problem in both ports.

statusSummary: ***string*** **, x ∈ { "Normal" , "Warning" , "Error" }**
Status summary of the internode port.

(Virtual machine) The output is always Normal.

(Bare metal)

- Normal: No action by the user is required.
- Warning: Although immediate action by the user is not required, some action may have to be taken.
- Error: Immediate action by the user is required.

internodePortPerformance: *object*

DESCRIPTION

Performance information of the internode port (monitor information). The infinite value (Infinity) is not output for double type values.

PROPERTIES

id: ***string*** **(uuid)**
The ID of the internode port.

receiveTransferRate: ***number*** **(double)**
Amount of data received per second (unit: MiB/sec).

sendTransferRate: ***number*** **(double)**
Amount of data sent per second (unit: MiB/sec).

internodePortPerformanceListResponse: *object*

DESCRIPTION

A list of internode port performance information (monitor information) at the specified time.

PROPERTIES

data: ***object[]***

ITEMS

internodePortPerformanceListResponseData: object (on page 622)

internodePortPerformanceListResponseData: *object*

DESCRIPTION

The performance information (monitor information) of the internode port when the information was collected. For performanceObjects, inter nodes of which performance information could be collected when the API was executed are output as an array.

PROPERTIES

timestamp: ***string*** **(date-time)**

The time when this information was collected.

performanceObjects: ***object[]***

A list of internode port performance information (monitor information) collected at the time specified for timestamp.

ITEMS

internodePortPerformance: object (on page 621)

ipv4InformationOfControlPort: *object*

DESCRIPTION

IPv4 setting information of the control port.

PROPERTIES

address: ***string***

IP address (IPv4).

subnetMask: ***string***

Subnet mask (IPv4).

ipv4InformationOfInternodePort: *object*

DESCRIPTION

IPv4 setting information of the internode port.

PROPERTIES

address: *string*
IP address (IPv4).

subnetMask: *string*
Subnet mask (IPv4).

ipv4InformationOfUniversal: *object*

DESCRIPTION

IPv4 setting information of the iSCSI port.

PROPERTIES

address: *string* (7 to 15 chars)
IP address (IPv4).

subnetMask: *string* (7 to 15 chars)
Subnet mask (IPv4).

defaultGateway: *string* (up to 15 chars)
The IP address of the default gateway (IPv4).

ipv4RouteOfStorageNodeNetworkSetting: *object*

DESCRIPTION

Routing table (IPv4).

PROPERTIES

destination: *string*
Destination network.

gateway: *string*
Gateway (NextHop).

interface: *string*
The interface name.

ipv6InformationOfUniversal: *object*

DESCRIPTION

iSCSI port IPv6 information.

PROPERTIES

linklocalAddressMode: *string* , x ∈ { "Auto" , "Manual" }
Setting mode for link local addresses.

linklocalAddress: *string* (up to 39 chars)
IPv6 link local address.

An empty string "" is output if no address is set.

globalAddressMode: *string* , x ∈ { "Auto" , "Manual" }
Setting mode for IPv6 global addresses.

globalAddress1: *string* (up to 39 chars)
IPv6 global address 1.

An empty string "" is output if no address is set.

subnetPrefixLength1: *integer* , { x ∈ ℤ | 0 ≤ x ≤ 128 }
Subnet prefix length of IPv6 global address 1.

defaultGateway: *string* (up to 39 chars)
The IP address of the default gateway (IPv6).

An empty string "" is output if no address is set.

iscsiInformationOfEditPort: *object*

DESCRIPTION

A parameter for editing the iSCSI settings.

Each value remains as is if nothing is specified for it. Setting is impossible in an FC connection.

PROPERTIES

isIsnsClientEnabled: *boolean*
iSNS client function. Specifying "true" enables the iSNS client function.

isnsServers: *object[]* (1 item)
iSNS server as the connection destination in the iSNS client function.

ITEMS

isnsServerOfEditIscsiInformation: object (on page 626)

Note:

The iSNS client function can be set only in iSNS connection.

iscsiPortPerformance: *object*

DESCRIPTION

Performance information of the iSCSI port.

Output only if the compute port uses iSCSI connection. In this case, fc is null.

PROPERTIES

target: iscsiTargetPortPerformance: object (on page 625) nullable

iscsiTargetPortPerformance: *object*

DESCRIPTION

Performance information of the iSCSI port when the target is running.

PROPERTIES

readIOPS: *integer* (int32)
The number of read I/Os per second (unit: IOPS).

writeIOPS: *integer* (int32)
The number of write I/Os per second (unit: IOPS).

readTransferRate: *number* (double)
Read data transfer amount per second (unit: MiB/sec).

writeTransferRate: *number* (double)
Write data transfer amount per second (unit: MiB/sec).

responseTime: *number* (double)
Average response time (unit: msec).

iscsiUniversal: *object*

DESCRIPTION

iSCSI port information.

Output only if the compute port uses iSCSI connection. In this case, fcInformation is null.

PROPERTIES

ipMode: *string* , x ∈ { "ipv4" , "ipv4v6" }

Enables or disables IPv4/IPv6.

- ipv4: Enables IPv4 only.
- ipv4v6: Enables both IPv4 and IPv6.

ipv4Information: ipv4InformationOfUniversal: object (on page 623)

ipv6Information: ipv6InformationOfUniversal: object (on page 624)

delayedAck: *boolean*

Whether TCP delayed ACKs are used. When "true" is specified, TCP delayed ACKs are used.

mtuSize: *integer* (int32) , { x ∈ ℤ | 1500 ≤ x ≤ 9000 }

The MTU size of Ethernet (unit: byte).

macAddress: *string* (17 chars) , must match /^([a-f0-9]{2}:){5}[a-f0-9]{2}$/

MAC address.

isIsnsClientEnabled: *boolean*

iSNS client function. Specifying "true" enables the iSNS client function.

isnsServers: *object[]* (1 item)

iSNS server as the connection destination in the iSNS client function.

ITEMS

isnsServerOfIscsiUniversal: object (on page 627)

isnsServerOfEditIscsiInformation: *object*

DESCRIPTION

A parameter for editing the iSNS server settings.

Each value remains as is if nothing is specified for it.

However, either serverName or port must be specified. If neither of them is specified, HTTP status code 400 (Bad Request) is returned.

PROPERTIES

index: *integer* (int32) , { x ∈ ℤ | 1 ≤ x ≤ 1 } (default: "1")

The ID of the iSNS server.

serverName: *string* (2 to 45 chars) , must match /^(([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])$|^(((([0-9a-fA-F]{1,4}:){7}([0-9a-fA-F]{1,4}|:))|(([0-9a-fA-F]{1,4}:){6}(:[0-9a-fA-F]{1,4}|((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|0?[0-9]?[0-9])(\.(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|0?[0-9]?[0-9])){3})|:))|(([0-9a-fA-F]{1,4}:){5}(((:[0-9a-

fA-F]{1,4}){1,2})|:((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|0?[0-9]?[0-9])(\.(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|0?[0-9]?[0-9])){3})|:))|((([0-9a-fA-F]{1,4}:){4}(((:[0-9a-fA-F]{1,4}){1,3})|((:[0-9a-fA-F]{1,4})?:((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|0?[0-9]?[0-9])(\.(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|0?[0-9]?[0-9])){3}))|:))|((([0-9a-fA-F]{1,4}:){3}(((:[0-9a-fA-F]{1,4}){1,4})|((:[0-9a-fA-F]{1,4}){0,2}:((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|0?[0-9]?[0-9])(\.(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|0?[0-9]?[0-9])){3}))|:))|((([0-9a-fA-F]{1,4}:){2}(((:[0-9a-fA-F]{1,4}){1,5})|((:[0-9a-fA-F]{1,4}){0,3}:((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|0?[0-9]?[0-9])(\.(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|0?[0-9]?[0-9])){3}))|:))|((([0-9a-fA-F]{1,4}:){1}(((:[0-9a-fA-F]{1,4}){1,6})|((:[0-9a-fA-F]{1,4}){0,4}:((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|0?[0-9]?[0-9])(\.(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|0?[0-9]?[0-9])){3}))|:))|(:(((:[0-9a-fA-F]{1,4}){1,7})|((:[0-9a-fA-F]{1,4}){0,5}:((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|0?[0-9]?[0-9])(\.(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|0?[0-9]?[0-9])){3}))|:)))(\%.+)?\s*)$/

IP address (IPv4 or IPv6) setting of the iSNS server.

port: *integer* (int32) , { x ∈ ℤ | 1 ≤ x ≤ 65536 }

TCP port number of the iSNS server.

isnsServerOfIscsiUniversal: *object*

DESCRIPTION

Setting information of the iSNS server.

PROPERTIES

index: *integer* (int32) , { x ∈ ℤ | 1 ≤ x ≤ 1 }

The ID of the iSNS server.

serverName: *string* (up to 45 chars) , must match /^$|^(([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])$|^((([0-9a-f]{1,4}:){7}([0-9a-f]{1,4}))|(([0-9a-f]{1,4}:){6}:)|(([0-9a-f]{1,4}:){5}(((:[0-9a-f]{1,4}))|:))|(([0-9a-f]{1,4}:){4}(((:[0-9a-f]{1,4}){1,2})|:))|(([0-9a-f]{1,4}:){3}(((:[0-9a-f]{1,4}){1,3})|:))|(([0-9a-f]{1,4}:){2}(((:[0-9a-f]{1,4}){1,4})|:))|(([0-9a-f]{1,4}:){1}(((:[0-9a-f]{1,4}){1,5})|:))|(:(((:[0-9a-f]{1,4}){1,6})|:)))$/

IP address (IPv4 or IPv6) setting of the iSNS server. An empty string "" is output if no address is set.

port: *integer* (int32) , { x ∈ ℤ | 1 ≤ x ≤ 65536 }

TCP port number of the iSNS server.

job: *object*

DESCRIPTION

Job information.

PROPERTIES

jobId: *string* (uuid)
Job ID.

self: *string* (link)
URL to access a target job record.

EXAMPLE
"/ConfigurationManager/simple/v1/objects/jobs/56c698c5-7263-4996-90f7-b0ddc840b410"

userId: *string* (5 to 255 chars) , must match /^[\-A-Za-z0-9!#\$%&'\.@\^_`\{\}~]{5,255}$/
The ID of the user who issued the API that triggers creation of the job.

status: *string* , x ∈ { "Initializing" , "Running" , "Completed" }
The progress of the job.

- Initializing: Being initialized.
- Running: Being executed.
- Completed

state: *string* , x ∈ { "Queued" , "Started" , "StorageAccepted" , "Stopping" , "Succeeded" , "Failed" , "Stopped" , "Unknown" }
The job status.

- Queued: The job is queued.
- Started: The job is started.
- StorageAccepted: The request is accepted by the storage cluster.
- Stopping: The job is in the progress of being stopped.
- Succeeded: The job succeeded.
- Failed: The job could not be executed.
- Stopped: The job is stopped.
- Unknown: The job status is unknown.

createdTime: *string* (date-time)
The time when the job was created.

updatedTime: *string* (date-time)
The time when the job state was updated.

completedTime: *string* (date-time) nullable
The time when the job was completed. Returns null if the job is not completed.

request: request: object (on page 673)

affectedResources: ***string[]*** **nullable**

URLs used to access the target resource for the job.

Basically, if one API uses multiple resources, URLs of all target resources are returned. If a job is unsuccessful, only URLs of the resources of which jobs are confirmed to be completed are returned.

If resources are deleted successfully, the URLs of the deleted resources are returned. In this case, access the returned URLs. If 404 Not Found appears, the resource is deleted successfully.

As exceptions, for APIs listed below for which a large amount of resources might be returned, the following URL is returned in affectedResources.

API	Execution conditions	URL returned in affectedResources
POST /v1/objects/volume-server-connections	When only one volume is connected.	/ConfigurationManager/simple/v1/objects/volume-server-connections/<*volumeId*>,<*serverId*>
	When multiple volumes are connected.	/ConfigurationManager/simple/v1/objects/volume-server-connections?serverId=<*serverId*>
POST /v1/objects/volume-server-connections/actions/release/invoke	When only one volume is disconnected.	/ConfigurationManager/simple/v1/objects/volume-server-connections/<*volumeId*>,<*serverId*>
	When multiple volumes are disconnected.	/ConfigurationManager/simple/v1/objects/volume-server-connections?serverId=<*serverId*>
POST /v1/objects/servers/<*id*>/paths	When both hbaId and portId are specified, and the connection to the resource succeeded.	/ConfigurationManager/simple/v1/objects/servers/<*id*>/paths/<*hbaId*>,<*portId*>
	When portId is omitted or both hbaId and portId are omitted, and connection to one or more resources succeeded.	/ConfigurationManager/simple/v1/objects/servers/<*id*>/paths

API	Execution conditions	URL returned in affectedResources
DELETE /v1/objects/servers/<*id*>/hbas/<*hbaId*>	When one or more connections between the target initiators and compute ports are released.	/ConfigurationManager/simple/v1/objects/servers/<*id*>/paths
	When the initiator is deleted.	/ConfigurationManager/simple/v1/objects/servers/<*id*>/hbas/<*hbaId*>
DELETE /v1/objects/servers/<*id*>	When one or more target initiators and compute ports are released.	/ConfigurationManager/simple/v1/objects/servers/<*id*>/paths
	When one or more initiators are deleted.	/ConfigurationManager/simple/v1/objects/servers/<*id*>/hbas
	When one or more connections between the compute nodes and volumes are released.	/ConfigurationManager/simple/v1/objects/volume-server-connections/<*volumeId*>,<*serverId*>
	When the compute node is deleted.	/ConfigurationManager/simple/v1/objects/servers/<*id*>

ITEMS

string (link)

error: errorResponse: object (on page 611) nullable

ldapServerConnectionVerification: *object*

DESCRIPTION

Connection with the LDAP server.

PROPERTIES

result: resultOfLdapServerConnectionVerification: object (on page 676)

error: errorInformationOfLdapServerConnectionVerification: object (on page 611)

ldapSettingOfEditExternalAuthServerSetting: *object*

DESCRIPTION

A parameter for editing the LDAP authentication settings.

PROPERTIES

mappingMode: *string* , x ∈ { "User" , "Group" }

Unit of mapping to the LDAP server.

- User: Mapping for each user. Grants permission to individual users in the LDAP server.

- Group: Mapping for each user group. Grants permission to individual user groups in the LDAP server.

primaryLdapServerUrl: *string* (up to 267 chars) , must match /^$|^ldaps?:\/\/((([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])|([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9])\.)*([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9])))(:([1-9]|[1-9][0-9]{1,3}|[1-5][0-9]{4}|6[0-4][0-9]{3}|65[0-4][0-9]{2}|655[0-2][0-9]|6553[0-5]))?/?$/

URL of the primary LDAP server.

Specify it as "ldap(s)://{<*IPv4-address*> | <*host-name*>}:<*port-number*>". ":<*port-number*>/" can be omitted.

Specify an empty string "" to initialize it.

secondaryLdapServerUrl: *string* (up to 267 chars) , must match /^$|^ldaps?:\/\/((([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])|([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9])\.)*([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9])))(:([1-9]|[1-9][0-9]{1,3}|[1-5][0-9]{4}|6[0-4][0-9]{3}|65[0-4][0-9]{2}|655[0-2][0-9]|6553[0-5]))?/?$/

URL of the secondary LDAP server.

Specify it as "ldap(s)://{<*IPv4-address*> | <*host-name*>}:<*port-number*>". ":<*port-number*>/" can be omitted.

Specify an empty string "" to initialize it.

isStartTlsEnabled: *boolean*

Enables or disables StartTLS communication for LDAP authentication.

baseDn: *string* (up to 8192 chars)

Base Distinguished Name used as the point from where a user or user group will be searched for LDAP authentication. Specify a distinguished name that includes all the users and user groups to be LDAP authenticated.

Specify it following the RFC 4514 stipulations.

bindDn: *string* (up to 8192 chars)

Bind Distinguished Name used for performing a search on a tree specified in baseDn.

Specify it following the RFC 4514 stipulations.

bindDnPassword: *string* (up to 8192 chars)

Password for Distinguished Name specified in bindDn.

userIdAttribute: *string* (up to 8192 chars)

LDAP Attribute Type mapped as the user ID. Specify it following the RFC 4514 stipulations.

For example, specify "sAMAccountName" when using the Windows logon ID when linked with an AD server.

The following conditions must be met.

- The value of the specified attribute is of the character type and it has the character length that can be used as userId.
- It is unique in the search range of baseDn or userTreeDn.

userTreeDn: *string* (up to 8192 chars)

Base Distinguished Name used as the point from where a user will be searched for LDAP authentication. Specify it following the RFC 4514 stipulations.

Search with userTreeDn has priority when it is specified at the same time as baseDn.

The default of this item is an empty string. If the value is an empty string, a string of "OU=users," followed by baseDn is handled as userTreeDn.

userObjectClass: *string* (up to 8192 chars)

LDAP object class to be mapped as a user. Only the LDAP entry which is the applicable object class is mapped.

externalGroupNameAttribute: *string* (up to 8192 chars)

LDAP Attribute Type mapped as externalGroupName in the user group.

Specify it following the RFC 4514 stipulations.

To enable a search as externalGroupName, it must be unique in the search range of baseDn or userGroupTreeDn.

userGroupTreeDn: *string* (up to 8192 chars)

Base Distinguished Name used as the point from where a user will be searched for LDAP authentication. Specify it following the RFC 4514 stipulations.

Search with userGroupTreeDn has priority when it is specified at the same time as baseDn.

The default of this item is an empty string. If the value is an empty string, a string of "OU=userGroups," followed by baseDn is handled as userGroupTreeDn.

userGroupObjectClass: *string* (up to 8192 chars)

LDAP object class to be mapped as a user group. Only the LDAP entry which is the applicable object class is mapped.

timeoutSeconds: *integer* (int32) , { x ∈ ℤ | -1 ≤ x ≤ 65535 }

Timeout time (in seconds) applied to the connection to the LDAP server. -1 means that a session never times out. The default is -1. It is recommended that you use the default value without changing it.

retryIntervalMilliseconds: *integer* (int32) , { x ∈ ℤ | 1 ≤ x ≤ 3000 }

Retry interval (in milliseconds) in communication with the LDAP server. The default is 100. It is recommended that you use the default value without changing it.

maxRetries: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 65535 }

Number of retries in communication with the LDAP server. 0 means that no retry is performed. The default is 3. It is recommended that you use the default value without changing it.

ldapSettingOfExternalAuthServerSetting: *object*

DESCRIPTION

Setting information about LDAP authentication.

PROPERTIES

mappingMode: *string* , x ∈ { "User" , "Group" }

Unit of mapping to the LDAP server.

- User: Mapping for each user. Grants permission to individual users in the LDAP server.
- Group: Mapping for each user group. Grants permission to individual user groups in the LDAP server.

primaryLdapServerUrl: *string* (up to 267 chars) , must match /^$|^ldaps?:\/\/((([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])|([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9])\.)*([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9])))(:([1-9]|[1-9][0-9]{1,3}|[1-5][0-9]{4}|6[0-4][0-9]{3}|65[0-4][0-9]{2}|655[0-2][0-9]|6553[0-5]))?/?$/

URL of the primary LDAP server. An empty string "" is returned if no URL is set.

secondaryLdapServerUrl: *string* (up to 267 chars) , must match /^$|^ldaps?:\/\/((([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])|([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9])\.)*([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9])))(:([1-9]|[1-9][0-9]{1,3}|[1-5][0-9]{4}|6[0-4][0-9]{3}|65[0-4][0-9]{2}|655[0-2][0-9]|6553[0-5]))?/?$/

URL of the secondary LDAP server. An empty string "" is returned if no URL is set.

isStartTlsEnabled: ***boolean***

Enables or disables the StartTLS communication.

baseDn: ***string*** **(up to 8192 chars)**

Base Distinguished Name used as the point from where a user or user group will be searched for external authentication.

bindDn: ***string*** **(up to 8192 chars)**

Bind Distinguished Name used for performing a search on a tree specified in baseDn.

userIdAttribute: ***string*** **(up to 8192 chars)**

LDAP Attribute Type mapped as the user ID.

userTreeDn: ***string*** **(up to 8192 chars)**

Base Distinguished Name used as the point from where a user will be searched for external authentication.

Search with userTreeDn has priority when it is specified at the same time as baseDn.

The default of this item is an empty string. If the value is an empty string, a string of "OU=users," followed by baseDn is handled as userTreeDn.

userObjectClass: ***string*** **(up to 8192 chars)**

LDAP object class to be mapped as a user. Only the LDAP entry which is the applicable object class is mapped.

externalGroupNameAttribute: ***string*** **(up to 8192 chars)**

LDAP Attribute Type mapped as externalGroupName in the user group.

userGroupTreeDn: ***string*** **(up to 8192 chars)**

Base Distinguished Name used as the point from where a user group will be searched for external authentication.

Search with userGroupTreeDn has priority when it is specified at the same time as baseDn.

The default of this item is an empty string. If the value is an empty string, a string of "OU=userGroups," followed by baseDn is handled as userGroupTreeDn.

userGroupObjectClass: ***string*** **(up to 8192 chars)**

LDAP object class to be mapped as a user group. Only the LDAP entry which is the applicable object class is mapped.

timeoutSeconds: ***integer*** (int32) **,** $\{ x \in \mathbb{Z} \mid -1 \le x \le 65535 \}$

Timeout time (in seconds) applied to the connection to the external authentication server. -1 means that a session never times out.

retryIntervalMilliseconds: ***integer*** (int32) **,** $\{ x \in \mathbb{Z} \mid 1 \le x \le 3000 \}$

Retry interval (in milliseconds) in communication with the external authentication server.

maxRetries: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 65535 }

Number of retries in communication with the external authentication server. 0 means that no retry is performed.

license: *object*

DESCRIPTION

License information.

PROPERTIES

id: *string* (uuid)

The ID of the license.

programProductName: *string* (1 to 63 chars)

Name of the program product.

"_NotSupported" is displayed for an unsupported license.

status: *string* , x ∈ { "Active" , "Warning" , "Overwritten" , "GracePeriod" , "Invalid" }

Status of the license.

- Active: Status in which a license is applied to the system.
- Warning: Status in which a license is applied to the system, but a warning threshold is exceeded.
- Overwritten: The license is invalid because a conflicting license exists.
- GracePeriod: The license has expired or the contract capacity is exceeded.
- Invalid: The license is invalid.

statusSummary: *string* , x ∈ { "Normal" , "Warning" , "Error" }

License status summary.

- Normal: No action by the user is required.
- Warning: Although immediate action by the user is not required, some action may have to be taken.
- Error: Immediate action by the user is required.

cause: *string[]* nullable

Cause of Warning or Error in the license status summary.

ITEMS

string , x ∈ { "Capacity" , "Term" }

- Capacity: The licensed capacity is exceeded (including the license threshold setting).
- Term: The license has expired (including the license threshold setting).

keyType: *string* , x ∈ { "Trial" , "Perpetual" , "Subscription" , "Utility" , "Emergency" , "NonProductionTestAndDevelopment" }

License key type.

permittedCapacityInTiB:*integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 268435455 } nullable

Total logical capacity (in TiB) of all storage pools allowed by the license. Null is output if no limitation is applied to the capacity.

totalPoolCapacityInGiB: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 274877906943 } nullable

Total logical capacity (in GiB) of all storage pools. Null is output if no limitation is applied to the capacity.

remainingDays: *integer* (int64) , { x ∈ ℤ | -120 ≤ x ≤ 1000000 } nullable

Number of remaining days. Null is output if the license can be used indefinitely.

If the status is Active, Warning, or Overwritten, this number indicates the number of remaining days.

If the status is GracePeriod or Invalid, this value is a minus value, indicating the number of days since the license has expired or the licensed capacity was exceeded. For a license whose status is GracePeriod, the status becomes Invalid when remainingDays is -31 or less.

licenseSetting: *object*

DESCRIPTION

License setting.

PROPERTIES

warningThresholdSetting: warningThresholdSettingOfLicenseSetting: object (on page 749)

lockoutSettingOfEditUserAuthSetting: *object*

DESCRIPTION

A parameter for setting lockout.

This setting is applied only to users whose authentication is local.

PROPERTIES

maxAttempts: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 10 }**
Number of consecutive login failures until the account is locked. 0 means that the function is disabled (the user can be unsuccessful an unlimited number of times).

lockoutSeconds: ***integer*** **(int32) , { x ∈ ℤ | 60 ≤ x ≤ 600 }**
Duration (in seconds) after the account is locked due to consecutive login failures until the account is unlocked.

lockoutSettingOfUserAuthSetting: *object*

DESCRIPTION

Lockout settings.

This setting is applied only to users whose authentication is local.

PROPERTIES

maxAttempts: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 10 }**
Number of consecutive login failures until the account is locked. 0 means that the function is disabled (the user can be unsuccessful an unlimited number of times).

lockoutSeconds:***integer*** **(int32) , { x ∈ ℤ | 60 ≤ x ≤ 600 }**
Duration (in seconds) after the account is locked due to consecutive login failures until the account is unlocked.

loginMessage: *object*

DESCRIPTION

Message to be displayed in the GUI login window and CLI warning banner.

PROPERTIES

message: ***string*** **(up to 6144 chars) , must match /^[a-zA-Z0-9!"#\$%&'"\(\)*\+,\-\.\/:;<=>\?@\[\\\]\^_`\{\|\}~\t\r\n]{0,6144}$/**
Message body to be displayed in the GUI login window and CLI warning banner. An empty string "" means that nothing is displayed.

masterVolume: *object*

DESCRIPTION

P-VOL information.

PROPERTIES

masterVolumeId: *string* (uuid)

The ID of the volume which is P-VOL.

vpsId: *string* , must match /^\(system\)$|[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

The ID of the virtual private storage (VPS) that the acquisition-target resource belongs to.

If the resource does not belong to a VPS, the reserved word "(system)", which indicates that the resource is independent, is output instead of an ID.

vpsName: *string* (1 to 32 chars) , must match /^\(system\)$|[\-A-Za-z0-9,\.:@_]{1,32}$/

The name of the virtual private storage (VPS) that the acquisition-target resource belongs to.

If the resource does not belong to a VPS, the reserved word "(system)", which indicates that the resource is independent, is output.

qosParam: *object*

A QoS-related parameter.

PROPERTIES

upperLimitForIops: *integer* (int64) , { x ∈ ℤ | -1 ≤ x ≤ 2147483647 }

The upper limit of volume performance (in IOPS).

The value -1 indicates that no upper limit is placed on volume performance (in IOPS).

upperLimitForTransferRate: *integer* (int64) , { x ∈ ℤ | -1 ≤ x ≤ 2097151 }

The upper limit of volume performance (in MiB/s).

The value -1 indicates that no upper limit is placed on volume performance (in MiB/s).

upperAlertAllowableTime: *integer* (int32) , { x ∈ ℤ | -1 ≤ x ≤ 600 }

The alert threshold value (in seconds) for the upper limit of volume performance.

The value -1 indicates that no entries are output to the event log.

VPS administrators cannot specify this property when creating or editing volumes.

upperAlertTime: *string* (date-time) nullable

The last time the upper limit of volume performance was continuously exceeded and the conditions for the alert threshold of the performance upper limit were met (UTC).

memoryPerformance: *object*

DESCRIPTION

Performance information of the memory.

PROPERTIES

usage: ***number*** **(double)**
Memory usage (unit: %).

passwordAgeSettingOfEditUserAuthSetting: *object*

DESCRIPTION

A parameter for setting the expiration time of a password. This setting is applied only to users whose authentication is local.

HTTP status code 400 (Bad Request) is returned if both minAgeDays and maxAgeDays are not 0 and minAgeDays ≥ maxAgeDays.

PROPERTIES

requiresInitialPasswordReset: ***boolean***
Whether a new user is forced to change the default password before the initial login. If true, a new user is forced to change the default password before the initial login.

minAgeDays: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 10 }**
Number of days after which you can change the password again after you changed the password last. 0 means that the expiration time is disabled (the user can change the password immediately).

maxAgeDays: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 365 }**
Number of days during which you can use the password after you changed the password last. The password is invalid after this period has elapsed. 0 means that this limit is disabled (the user can use the password indefinitely).

passwordAgeSettingOfUserAuthSetting: *object*

DESCRIPTION

Password expiration time setting. This setting is applied only to users whose authentication is local.

PROPERTIES

requiresInitialPasswordReset: ***boolean***

Whether a new user is forced to change the default password before the initial login. If true, a new user is forced to change the default password before the initial login.

minAgeDays: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 10 }**

Number of days after which you can change the password again after you changed the password last. 0 means that the expiration time is disabled (the user can change the password immediately).

maxAgeDays: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 365 }**

Number of days during which you can use the password after you changed the password last. The password is invalid after this period has elapsed. 0 means that this limit is disabled (the user can use the password indefinitely).

passwordComplexitySettingOfEditUserAuthSetting: *object*

DESCRIPTION

A parameter for setting the password complexity. This setting is applied only to users whose authentication is local.

In such a case, if you specify a value to make (sum of minNumberOfUpperCaseChars, minNumberOfLowerCaseChars, minNumberOfNumerals, and minNumberOfSymbols) greater than minLength, HTTP status code 400 (Bad Request) is returned.

PROPERTIES

minLength: ***integer*** **(int32) , { x ∈ ℤ | 1 ≤ x ≤ 256 }**

Minimum password length.

minNumberOfUpperCaseChars: ***integer*****(int32) , { x ∈ ℤ | 0 ≤ x ≤ 256 }**

Minimum number of uppercase alphabetical characters contained in a password.

minNumberOfLowerCaseChars: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 256 }**

Minimum number of lowercase alphabetical characters contained in a password.

minNumberOfNumerals: ***integer*** **(int32), { x ∈ ℤ | 0 ≤ x ≤ 256 }**

Minimum number of numerals (0 to 9) contained in a password.

minNumberOfSymbols: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 256 }**

Minimum number of symbols (excluding alphanumeric characters) contained in a password.

numberOfPasswordHistory: ***integer*****(int32) , { x ∈ ℤ | 1 ≤ x ≤ 10 }**

Number of generations from generation 1 (when the password was changed) for which use of a previously used password is prohibited. 1 means that this limit is disabled (the user can set the same password as a past one).

passwordComplexitySettingOfUserAuthSetting: *object*

DESCRIPTION

Password complexity settings.

This setting is applied only to users whose authentication is local.

PROPERTIES

minLength: ***integer*** **(int32) , { x ∈ ℤ | 1 ≤ x ≤ 256 }**
Minimum password length.

minNumberOfUpperCaseChars: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 256 }**
Minimum number of uppercase alphabetical characters contained in a password.

minNumberOfLowerCaseChars: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 256 }**
Minimum number of lowercase alphabetical characters contained in a password.

minNumberOfNumerals: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 256 }**
Minimum number of numerals (0 to 9) contained in a password.

minNumberOfSymbols: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 256 }**
Minimum number of symbols (excluding alphanumeric characters) contained in a password.

numberOfPasswordHistory: ***integer*** **(int32) , { x ∈ ℤ | 1 ≤ x ≤ 10 }**
Number of generations from generation 1 (when the password was changed) for which use of a previously used password is prohibited. 1 means that this limit is disabled (the user can set the same password as a past one).

patchAuditLogSettingParam: *object*

DESCRIPTION

A parameter for editing the audit log settings.

PROPERTIES

syslogForwardingSetting: syslogForwardingSettingOfEditAuditLogSetting: object (on page 713) required

patchChapUserParam: *object*

DESCRIPTION

A parameter for editing CHAP user information. If none of the following is specified, job does not succeed.

PROPERTIES

targetChapUserName: ***string*** **(1 to 223 chars), must match /^[a-zA-Z0-9\.:@_\-\+=\[\]~]{1,223}$/**
CHAP user name used for CHAP authentication on the compute port (i.e., target side).

targetChapSecret: ***string*** **(12 to 32 chars) , must match /^[a-zA-Z0-9\.:@_\-\+=\/\[\]~]{12,32}$/**
CHAP secret used for CHAP authentication on the compute port (i.e., target side).

initiatorChapUserName: ***string*** **(1 to 223 chars) , must match /^$|^[a-zA-Z0-9\.:@_\-\+=\[\]~]{1,223}$/**
CHAP user name used for CHAP authentication on the initiator port of the compute node in mutual CHAP authentication.

initiatorChapSecret: ***string*** **(12 to 32 chars) , must match /^$|^[a-zA-Z0-9\.:@_\-\+=\/\[\]~]{12,32}$/**
CHAP secret used for CHAP authentication on the initiator port of the compute node in mutual CHAP authentication.

patchEventLogSettingParam: *object*

DESCRIPTION

A parameter for editing the event log settings.

Either of syslogForwardingSetting or emailReportSetting must be specified. Specifying both is not allowed. If you specified both or neither of them, HTTP status code 400 (Bad Request) is returned.

PROPERTIES

syslogForwardingSetting: syslogForwardingSettingOfEditEventLogSetting: object (on page 714)

emailReportSetting: emailReportSettingOfEditEventLogSetting: object (on page 610)

patchExternalAuthServerSettingParam: *object*

DESCRIPTION

A parameter for editing external authentication server settings.

If nothing is specified, HTTP status code 400 (Bad Request) is returned.

PROPERTIES

isEnabled: ***boolean***
Enables or disables external authentication.

authProtocol: *string* , x ∈ { "LDAP" }
Authentication protocol used for external authentication.

ldapSetting: ldapSettingOfEditExternalAuthServerSetting: object (on page 631)

patchLicenseSettingParam: *object*

DESCRIPTION

A parameter for setting a license.

Each value remains as is if nothing is specified for it. If none of the following is specified, HTTP status code 400 (Bad Request) is returned.

PROPERTIES

warningThresholdSetting: warningThresholdSettingOfEditLicenseSetting: object (on page 748)

patchLoginMessageParam: *object*

DESCRIPTION

A parameter for editing a message to be displayed in the GUI login window and CLI warning banner.

PROPERTIES

message: *string* (up to 6144 chars) , must match /^[a-zA-Z0-9!"#\$%&"\(\)*\+,\-\.\/:;<=>\?@\[\\\]\^_`\{\|\}~\t\r\n]{0,6144}$/ required
Message body to be displayed in the GUI login window and CLI warning banner. Specify an empty string "" if nothing is to be displayed.

patchPortAuthSettingParam: *object*

DESCRIPTION

A parameter for editing the authentication settings for the compute port for the target operation.

Each value remains as is if nothing is specified for it. If none of the following is specified, HTTP status code 400 (Bad Request) is returned.

PROPERTIES

authMode: *string* , x ∈ { "CHAP" , "CHAPComplyingWithInitiatorSetting" , "None" }
Authentication scheme of the compute port.

- CHAP: CHAP authentication.

- CHAPComplyingWithInitiatorSetting: Complies with the setting of the compute node. If the setting is "CHAP", CHAP authentication is performed. If the setting is "None", no authentication is required.

- None: No authentication is performed.

isDiscoveryChapAuth: *boolean*
Enables or disables CHAP authentication at the time of discovery in iSCSI connection. Enables CHAP authentication at the time of discovery when true is specified.

isMutualChapAuth: *boolean*
Enables or disables mutual CHAP authentication. Enables mutual CHAP authentication when true is specified.

patchPortParam: *object*

DESCRIPTION

A parameter for editing compute port settings. The current values are not changed if parameter setting is omitted. If none of the following is specified, job does not succeed.

PROPERTIES

nickname: *string* , must match /^[a-zA-Z0-9!#\$%&'\+\-\.=@\^_\{\}~\(\)\[\]:]{1,32}$/
The compute port nickname. Each compute port must have its own unique nickname.

name: *string* , must match /^((iqn\.[0-9]{4}\-[0-9]{2}\.[a-zA-Z0-9\-:\.]{0,211})|(eui\.[0-9a-fA-F]{16}))$/
The iSCSI name of the compute port. For FC connection, this property cannot be specified.

The same name cannot be used for multiple compute ports.

fcInformation: fcInformationOfEditPort: object (on page 616)

iscsiInformation: iscsiInformationOfEditPort: object (on page 624)

patchProtectionDomainParam: *object*

DESCRIPTION

A parameter for editing the protection domain settings.

PROPERTIES

asyncProcessingResourceUsageRate: ***string*** **, x ∈ { "VeryHigh", "High" , "Middle" , "Low" } required**

Controls the resource usage rate of the internal processing I/O.

To select VeryHigh, prerequisites must be met. For details, contact customer support.

patchServerParam: *object*

DESCRIPTION

A parameter for editing the compute node.

PROPERTIES

nickname: ***string*** **(1 to 229 chars) , must match /^[a-zA-Z0-9,\.:@_]([a-zA-Z0-9,\.:@_\-]{0,228})$/**

The nickname of the compute node. The same nickname cannot be set for multiple compute nodes.

osType: ***string*** **, x ∈ { "Linux" , "VMware" , "Windows" }**

The OS type of the compute node.

vpsId: ***string*** **, must match /^system$|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/**

The ID of the operation-target virtual private storage (VPS).

To specify a resource that does not belong to the VPS, set "system" for this property.

If this property is omitted, the VPS that the user who runs the API belongs to is assumed to be the operation target.

patchSnmpSettingParam: *object*

DESCRIPTION

A parameter for editing SNMP settings.

PROPERTIES

isSNMPAgentEnabled: ***boolean***

Enables or disables SNMP.

snmpVersion: ***string*** **, x ∈ { "v2c" } required**

SNMP version.

sendingTrapSetting: **sendingTrapSettingOfEditSnmpSetting: object (on page 682)**

requestAuthenticationSetting: requestAuthenticationSettingOfEditSnmpSetting: object (on page 673)

systemGroupInformation: systemGroupInformationOfEditSnmpSetting: object (on page 717) required

patchSpareNodeParam: *object*

DESCRIPTION

A parameter for editing spare node information.

For omitted properties, the current values are maintained. However, if no properties are specified, HTTP status code 400 (Bad Request) is returned.

PROPERTIES

faultDomainId: *string* (uuid)
The ID of the fault domain that the nodes belong to.

controlPortIpv4Address: *string* (7 to 15 chars) , must match /^(([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])$/
The IP address (IPv4) of the control port.

setupUserPassword: *string* (1 to 256 chars) , must match /^[\-A-Za-z0-9!#\$%&'"\(\)*\+,\.\/:;<>=\?@\[\]\\\^_`\{\}\|~]{1,256}$/
Setup user password.

bmcName: *string* (1 to 253 chars) , must match /^([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9])\.)*([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9]))$/
The host name or IP address (IPv4) of the BMC.

bmcUser: *string* (1 to 512 chars) , must match /^[a-zA-Z0-9!"#\$%&'\(\)*\+,\-\.\/:;<=>\?@\[\\\]\^_`\{\|\}~]{1,512}$/
The user name for BMC connection.

bmcPassword: *string* (1 to 512 chars) , must match /^[a-zA-Z0-9!"#\$%&'\(\)*\+,\-\.\/:;<=>\?@\[\\\]\^_`\{\|\}~]{1,512}$/
The password for BMC connection.

patchStorageAutoRecoverySettingParam: *object*

DESCRIPTION

A parameter for setting the auto-recovery function. Either "isEnabled" or "storageNodePersistentBlockingThresholdTime" must be specified. If neither of them is specified, HTTP status code 400 (Bad Request) is returned.

PROPERTIES

isEnabled: ***boolean***

Enables or disables the auto-recovery function. Specifying "true" enables the auto-recovery function.

storageNodePersistentBlockingThresholdTime: ***integer*(int32) , { x ∈ ℤ | 0 ≤ x ≤ 8784 }**

The time to determine if the temporarily blocked storage node that was recovered previously is changed to persistent blockage status (unit: hours).

If a temporarily blocked storage node was recovered previously at least once in the period specified for this parameter, the storage node is changed to persistent blockage status. When the value is 0, the storage node is not changed to persistent blockage status.

patchStorageNodeCapacitySettingParam: *object*

DESCRIPTION

A parameter for editing the capacity management settings of a storage node.

PROPERTIES

capacityBalancingSetting: capacityBalancingSettingParam: object (on page 585)

patchStorageNodeBmcAccessSettingParam: *object*

DESCRIPTION

A parameter for editing the BMC connection information about storage nodes.

PROPERTIES

bmcName: ***string*** **(1 to 253 chars) , must match /^([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9])\.)*([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9]))$/ required**

The host name or IP address (IPv4) of the BMC.

bmcUser: ***string*** **(1 to 512 chars) , must match /^[a-zA-Z0-9!"#\$%&'\(\)*\+,\-\.\/:;<=>\?@\[\\\]\^_`\{\|\}~]{1,512}$/ required**

The user name for BMC connection.

bmcPassword: ***string*** **(1 to 512 chars) , must match /^[a-zA-Z0-9!"#\$%&'\(\)*\+,\-\.\/:;<=>\?@\[\\\]\^_`\{\|\}~]{1,512}$/ required**

The password for BMC connection.

patchPoolSettingParam: *object*

DESCRIPTION

A parameter for setting rebuild capacity of the storage pool.

When the following combinations of parameters are used, HTTP status code 400 (Bad Request) is returned.

- When specifying Variable for rebuildCapacityPolicy and specifying numberOfTolerableDriveFailures of rebuildCapacityResourceSetting
- When specifying Fixed for rebuildCapacityPolicy without specifying numberOfTolerableDriveFailures of rebuildCapacityResourceSetting

PROPERTIES

rebuildCapacityPolicy: *string* , x ∈ { "Fixed " , "Variable " } required
Rebuild capacity policy.

- Fixed: Secures capacity required for Rebuild as a fixed Rebuild-dedicated capacity.
- Variable: Secures a part of user data capacity as rebuild capacity when the storage pool usage is low, and uses the capacity entirely as user data capacity when the storage pool usage increases.

rebuildCapacityResourceSetting: rebuildCapacityResourceSettingParam: object (on page 671)

patchUserAuthSettingParam: *object*

DESCRIPTION

A parameter for setting user authentication. Each value is not changed if nothing is specified. If none of the following is specified, HTTP status code 400 (Bad Request) is returned.

PROPERTIES

passwordComplexitySetting: passwordComplexitySettingOfEditUserAuthSetting: object (on page 640)

passwordAgeSetting: passwordAgeSettingOfEditUserAuthSetting: object (on page 639)

lockoutSetting: lockoutSettingOfEditUserAuthSetting: object (on page 636)

sessionSetting: sessionSettingOfEditUserAuthSetting: object (on page 686)

patchUserGroupParam: *object*

DESCRIPTION

A parameter for editing user group information.

PROPERTIES

roleNames: ***string[]***

Role of the user group. At least one role must be specified. Specify at least one role if you specify this property. If no role is specified, HTTP status code 400 (Bad Request) is returned.

ITEMS

- For system administrators: (Specify one to six items)

  string , x ∈ { "Security", "Storage", "Monitor", "Service", "Audit", "Resource" }

- For VPS administrators: (Specify one to three items)

  string , x ∈ { "VpsSecurity", "VpsStorage", "VpsMonitor" }

scope: ***string[]*** **(1 to 65 items)**

An array of the IDs of virtual private storages (VPSs) that the user group can access.

Because a system administrator can access multiple VPSs, if this parameter has the "system" element, multiple IDs of VPSs can be specified. A VPS administrator, who can access only one VPS, can specify only one ID.

ITEMS

string , must match /^system$|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

The ID of the VPS that the user group belongs to cannot be omitted. If it is not included, HTTP status code 400 (Bad Request) is returned.

In the case of a system administrator group (if the ID of the VPS that the user group belongs to is "system"), any VPSs can be specified. In the case of a VPS administrator group (if the ID of the VPS that the user group belongs to is not "system"), only one array element can be specified. If two or more elements are specified, HTTP status code 400 (Bad Request) is returned.

patchUserParam: *object*

DESCRIPTION

A parameter for editing user information. Either "password" or "isEnabled" must be specified. If neither of them is specified, HTTP status code 400 (Bad Request) is returned.

PROPERTIES

password: *string* (1 to 256 chars) , must match /^[\-A-Za-z0-9!#\$%&"'\(\)*\+,\.\/:;<>=\?@\[\]\\\^_`\{\}\|~]{1,256}$/

New password.

The specifiable values can be restricted by policy. The regular expressions described here show the case where the input restrictions by policy are the least strict.

If you specify a password for a user whose authentication is not local, HTTP status code 400 (Bad Request) is returned.

isEnabled: *boolean*

Enables or disables the user.

patchVirtualPrivateStorageParam: *object*

DESCRIPTION

A parameter for editing a virtual private storage (VPS).

If no properties are specified, the job will be unsuccessful.

PROPERTIES

name: *string* (1 to 32 chars) , must match /^(?!system$)[\-A-Za-z0-9,\.:@_]{1,32}$/

The new VPS name.

The string "system" cannot be specified, and an existing VPS cannot be named.

upperLimitForNumberOfUserGroups: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 256 }

The upper limit of the number of user groups that belong to the VPS.

upperLimitForNumberOfUsers: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 256 }

The upper limit of the number of users that belong to the VPS.

upperLimitForNumberOfSessions: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 436 }

The upper limit of the number of sessions on the VPS.

upperLimitForNumberOfServers: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 1024 }

The maximum allowable number of compute nodes on the VPS.

upperLimitForNumberOfVolumes: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 32768 }

The upper limit of the number of volumes on the VPS.

For details about the values to be specified for upperLimitForNumberOfVolumes, contact customer support.

upperLimitForCapacityOfVolumes: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 9223372036854775807 }

The upper limit of the total volume capacity (MiB) of the VPS.

For details about the values to be specified for upperLimitForCapacityOfVolumes, contact customer support.

upperLimitForCapacityOfSingleVolume: ***integer*** **(int64) , { x ∈ ℤ | -1 ≤ x ≤ 6871947674 }**
The upper limit of the capacity (MiB) of a single volume of the VPS.

To place no limit on the capacity of a single volume, specify -1.

For details about the values to be specified for upperLimitForCapacityOfSingleVolume, contact customer support.

upperLimitForIopsOfVolume: ***integer*** **(int64) , { x ∈ ℤ | -1 ≤ x ≤ 2147483647 }**
The upper limit of volume performance (in IOPS) of the VPS.

This is used as the default value for the upper limit of performance (in IOPS) of volumes created on the VPS. To set the upper limit of volume performance (in IOPS), specify a value in the range from 100 to 2147483647. To set no upper limit, specify -1. If you specify values from 0 to 99, jobs will be unsuccessful.

The VPS administrator who creates volumes can set a value that is no more than this value as the upper limit of performance (in IOPS) for each volume. If you make both of the upperLimitForIopsOfVolume and upperLimitForTransferRateOfVolume settings unavailable, the setting of upperAlertAllowableTimeOfVolume is also made unavailable. If this property is omitted, -1 is assumed to be specified.

upperLimitForTransferRateOfVolume: ***integer*** **(int64) , { x ∈ ℤ | -1 ≤ x ≤ 2097151 }**
The upper limit of volume performance (in MiB/s) for the VPS.

This is used as the default value for the upper limit of performance (in MiB/s) of volumes created on the VPS. To set the upper limit of volume performance (in MiB/s), specify a value in the range from 1 to 2097151. To set no upper limit, specify -1. If you specify 0, jobs will be unsuccessful.

The VPS administrator who creates volumes can set a value that is no more than this value as the upper limit of performance (in MiB/s) for each volume. If you make both of the upperLimitForIopsOfVolume and upperLimitForTransferRateOfVolume settings unavailable, the setting of upperAlertAllowableTimeOfVolume is also made unavailable. If this property is omitted, -1 is assumed to be specified.

upperAlertAllowableTimeOfVolume: ***integer*** **(int32) , { x ∈ ℤ | -1 ≤ x ≤ 600 }**
The alert threshold value (in seconds) for the upper limit of volume performance for the VPS.

This is used as the default value for the alert threshold for the upper limit of performance of volumes created on the VPS. To set the alert threshold, specify a value in the range from 1 to 600. To set no alert threshold, specify -1. If you specify 0, jobs will be unsuccessful.

A message is output to the event log when restriction of the upper limit of performance specified by upperLimitForIopsOfVolume or upperLimitForTransferRateOfVolume continues for the specified length of time. This property can be specified if either upperLimitForIopsOfVolume or upperLimitForTransferRateOfVolume, or both, is set. If this property is omitted, -1 is assumed to be specified.

patchVolumeParam: *object*

DESCRIPTION

A parameter for editing volume settings.

Each value remains as is if nothing is specified for it.

You can specify values depending on the volume status as shown further on.

If you specify parameters with combinations that are not indicated in the following, jobs will be unsuccessful.

- When the status is other than "UpdateFailed", specify any value.
- When the status is "UpdateFailed", no value can be specified.

PROPERTIES

name: *string* (1 to 32 chars) , must match /^[\-A-Za-z0-9,\.:@_]{1,32}$/
Volume name. Set a name unique throughout the volumes.

nickname: *string* (1 to 32 chars) , must match /^[\-A-Za-z0-9,\.:@_]{1,32}$/
The nickname of the volume. The same nickname can be specified for multiple volumes.

qosParam: patchVolumeQosParam: object (on page 652)

vpsId: *string* , must match //^system$|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/
The ID of the operation-target virtual private storage (VPS).

To specify a resource that does not belong to the VPS, set "system" for this property.

If this property is omitted, the VPS that the user who runs the API belongs to is assumed to be the operation target.

patchVolumeQosParam: *object*

DESCRIPTION

A QoS-related parameter.

PROPERTIES

upperLimitForIops: *integer* (int64) , { x ∈ ℤ | -1 ≤ x ≤ 2147483647 }
The upper limit of volume performance (in IOPS).

To set the upper limit of volume performance (in IOPS), specify a value in the range from 100 to 2147483647. To set no upper limit, specify -1. If you specify values from 0 to 99, jobs will be unsuccessful.

If you make both of the upperLimitForIops and upperLimitForTransferRate settings unavailable, the setting of upperAlertAllowableTime is also made unavailable.

upperLimitForTransferRate: *integer* (int64) , { x ∈ ℤ | -1 ≤ x ≤ 2097151 }
The upper limit of volume performance (in MiB/s).

To set the upper limit of volume performance (in MiB/s), specify a value in the range from 1 to 2097151. To set no upper limit, specify -1. If you specify 0, jobs will be unsuccessful.

If you make both of the upperLimitForIops and upperLimitForTransferRate settings unavailable, the setting of upperAlertAllowableTime is also made unavailable.

upperAlertAllowableTime: *integer* (int32) , { x ∈ ℤ | -1 ≤ x ≤ 600 }
The alert threshold value (in seconds) for the upper limit of volume performance.

The VPS administrator cannot specify this property.

To set the alert threshold, specify a value in the range from 1 to 600. To set no alert threshold, specify -1. If you specify 0, jobs will be unsuccessful.

A message is output to the event log when restriction of the upper limit of performance specified by upperLimitForIops or upperLimitForTransferRate continues for the specified length of time.

This property can be specified if either upperLimitForIops or upperLimitForTransferRate, or both, is set.

patchWebServerAccessSettingParam: *object*

DESCRIPTION

A parameter for web server access setting.

The setting remains as is if nothing is specified. If nothing is specified, HTTP status code 400 (Bad Request) is returned.

PROPERTIES

whitelistSetting: whitelistSettingOfEditWebServerAccessSetting: object (on page 749)

path: *object*

DESCRIPTION

Path information.

PROPERTIES

id: ***string***

The ID of the path. Format: "<*hbaId*>,<*portId*>" Where hbaId is the initiator ID of the computer node, and portId is the ID of the compute port for the target operation.

serverId: ***string*** **(uuid)**

The ID of the compute node.

hbaName: ***string***

The initiator WWN of the compute node for FC connection, or iSCSI name for iSCSI connection.

hbaId: ***string*** **(uuid)**

The initiator ID of the compute node.

portId: ***string*** **(uuid)**

The ID of the compute port to which the initiator of the compute node is allocated.

portName: ***string***

The WWN (for FC connection) or iSCSI Name (for iSCSI connection) of the compute port to which the initiator of the compute node is allocated.

portNickname: ***string*** **, must match /^[a-zA-Z0-9!#\$%&'\+\-\.=@\^_\{\}~\(\)\[\]:]{1,32}$/**

The nickname of the compute port to which the initiator of the compute node is allocated.

vpsId: ***string*** **, must match /^\(system\)$|[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/**

The ID of the virtual private storage (VPS) that the acquisition-target resource belongs to.

If the resource does not belong to a VPS, the reserved word "(system)", which indicates that the resource is independent, is output instead of an ID.

vpsName: ***string*** **(1 to 32 chars) , must match /^\(system\)$|[\-A-Za-z0-9,\.:@_]{1,32}$/**

The name of the virtual private storage (VPS) that the acquisition-target resource belongs to.

If the resource does not belong to a VPS, the reserved word "(system)", which indicates that the resource is independent, is output.

pathOfServer: *object*

DESCRIPTION

Information about registered paths.

PROPERTIES

hbaName: ***string***

The initiator WWN of the compute node for FC connection, or iSCSI name for iSCSI connection.

portIds: ***string[]***

A list of IDs of the allocation destination compute port for the target operation.

ITEMS

string (uuid)

DESCRIPTION

The ID of the allocation destination compute port for the target operation.

pool: *object*

DESCRIPTION

Storage pool information.

PROPERTIES

id: ***string*** **(uuid)**

The ID of the storage pool.

name: ***string*** **(1 to 32 chars)**

The name of the storage pool.

protectionDomainId: ***string*** **(uuid)**

The ID of the protection domain to which the volume is belonging.

statusSummary: ***string*** **, x ∈ { "Normal" , "Warning" , "Error" }**

The summary of the storage pool status.

- Normal: No action by the user is required.
- Warning: Although immediate action by the user is not required, some action may have to be taken.
- Error: Immediate action by the user is required.

status: ***string*** **, x ∈ { "Normal" , "ExceededThreshold" , "Error" }**

The storage pool status. The following status is returned:

- Normal: Operating normally.
- ExceededThreshold: The threshold is exceeded.
- Error: Cannot be used because the storage pool is full or blocked.

totalCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 6871947674 }
The logical capacity of the storage pool (unit: MiB).

totalRawCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 9223372036854775807 }
The effective physical capacity in the total physical capacity of the storage pool (Unit: MiB)

usedCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 6871947674 } nullable
The used capacity of the storage pool (unit: MiB).

freeCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 6871947674 } nullable
The available capacity of the storage pool (unit: MiB).

totalPhysicalCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 9223372036854775807 }
The total physical capacity of the storage pool (unit: MiB).

metaDataPhysicalCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 9223372036854775807 }
The capacity for control information in the total physical capacity of the storage pool (unit: MiB).

reservedPhysicalCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 9223372036854775807 }
The internally reserved area in the effective physical capacity of the storage pool. This area cannot be configured redundantly (unit: MiB).

usablePhysicalCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 6871947674 }
The capacity which can be used as the logical capacity in the effective physical capacity of the storage pool (unit: MiB).

blockedPhysicalCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 9223372036854775807 }
The blocked capacity in the effective physical capacity of the storage pool (unit: MiB).

totalVolumeCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 9223372036854775807 }
The total capacity of volumes that have been created on this storage controller (unit: MiB).

provisionedVolumeCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 9223372036854775807 }
The total capacity of provisioned volumes that have been created on this storage controller (unit: MiB).

otherVolumeCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 9223372036854775807 }
The total capacity of other volumes that have been created on this storage controller (unit: MiB).

temporaryVolumeCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 9223372036854775807 }
The total capacity of temporary volumes that have been created on this storage controller (unit: MiB).

capacityManage: *object*
Capacity management.

PROPERTIES

usedCapacityRate: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 100 } nullable**

Usage rate (unit: %).

maximumReserveRate: ***integer*** **(int32) , { x ∈ ℤ | -1 ≤ x ≤ 65534 }**

Maximum reserve rate (%). -1 indicates the capacity is unlimited.

maximumReserveRate is currently fixed to -1.

thresholdWarning: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 100 }**

The warning threshold (unit: %).

thresholdWarning is currently fixed to 70.

thresholdDepletion: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 100 }**

The Depletion threshold (unit: %).

thresholdDepletion is currently fixed to 80.

thresholdStorageControllerDepletion: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 100 }**

The depletion threshold of the storage controller managing the storage pool (unit: %).

thresholdDepletion is currently fixed to 95.

savingEffects: savingEffectOfPool: object (on page 676)

numberOfVolumes: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 32768 }**

The number of volumes belonging to the storage pool.

redundantPolicy: ***string*** **, x ∈ { "Mirroring" , "HitachiPolyphaseErasureCoding" }**

User data protection method.

redundantType: ***string*** **, x ∈ { "Duplication" , "2D+1P" , "4D+1P" , "4D+2P" }**

User data protection type.

dataRedundancy: ***integer*** **(int32) , { x ∈ ℤ | -1 ≤ x ≤ 2 }**

Redundancy of user data. User data will not be lost for this number of storage nodes in the event of the following failures.

- Failure of one or more drives in the storage node.
- Failure of the storage node.

Value of 0 means no redundancy, and -1 means that the user data has been lost.

storageControllerCapacitiesGeneralStatus: ***string*** **, x ∈ { "Normal" , "Warning" , "Error" }**

Summary information about the capacity status (capacityStatus) of all the storage controllers managing the storage pool.

- Normal: capacityStatus of all the storage controllers managing the storage pool is Normal.
- Warning: At least one of the storage controllers managing the storage pool has Warning capacityStatus and none have Error capacityStatus.
- Error: At least one of the storage controllers managing the storage pool has Error capacityStatus.

rebuildCapacityPolicy: ***string*** **, x ∈ { "Fixed" , "Variable" }**

Rebuild capacity policy.

- Fixed: Secures capacity required for Rebuild as a fixed Rebuild-dedicated capacity.
- Variable: Secures a part of user data capacity as rebuild capacity when the storage pool usage is low, and uses the capacity entirely as user data capacity when the storage pool usage increases.

rebuildCapacityResourceSetting: ***object***

Resource type and the number of resources of rebuild capacity.

PROPERTIES

numberOfTolerableDriveFailures: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 23 } nullable**

The number of drive failures that can be tolerated.

rebuildCapacityStatus: ***string*** **, x ∈ { "Sufficient" , "PartialShortage" , "None" }**

The status of securing rebuild capacity.

- Sufficient: The capacity required for Rebuild is already secured.
- PartialShortage: A part of the capacity required for Rebuild is already secured.
- None: None of the capacity required for Rebuild is secured.

rebuildableResources: **rebuildableResourcesOfPool: object (on page 671)**

Resource for which Rebuild is possible.

poolCapacity: *object*

DESCRIPTION

Capacity information (monitor information) for each storage pool. Null is output if the information could not be obtained.

PROPERTIES

id: ***string*** **(uuid)**
The ID of the storage pool.

usedCapacity: ***integer*** **(int64) nullable**
Total occupied capacity (unit: MiB).

poolCapacityListResponseData: *object*

DESCRIPTION

The capacity information (monitor information) of the storage pool when the information was collected. For performanceObjects, storage pools of which capacity information could be collected when the API was executed are output as an array.

PROPERTIES

timestamp: ***string*** **(date-time)**
The time when this information was collected.

performanceObjects: ***object[]***
A list of storage pool capacity information (monitor information) collected at the time specified for timestamp.

ITEMS
poolCapacity: object (on page 658)

poolPerformance: *object*

DESCRIPTION

Performance information (monitor information) for each storage pool. Null is output if the information could not be obtained. The infinite value (Infinity) is not output as the double type value.

PROPERTIES

id: ***string*** **(uuid)**
The ID of the storage pool.

volumeReadIOPS: ***integer*** **(int64) nullable**
The amount of write transfer across all volumes in the storage pool per second (unit: MiB/sec).

volumeWriteIOPS: ***integer*** **(int64) nullable**
The number of write I/Os across all volumes in the storage pool per second (unit: IOPS).

volumeReadTransferRate: *number* (double) nullable

The amount of read transfer across all volumes in the storage pool per second (unit: MiB/sec).

volumeWriteTransferRate: *number* (double) nullable

The amount of write transfer across all volumes in the storage pool per second (unit: MiB/sec).

poolPerformanceListResponseData: *object*

DESCRIPTION

The performance information (monitor information) of the storage pool when the information was collected. For performanceObjects, storage pools of which performance information could be collected when the API was executed are output as an array.

PROPERTIES

timestamp: *string* (date-time)

The time when this information was collected.

performanceObjects: *object[]*

A list of storage pool performance information (monitor information) collected at the time specified for timestamp.

ITEMS

poolPerformance: object (on page 659)

port: *object*

DESCRIPTION

Compute port information.

PROPERTIES

id: *string* (uuid)

The ID of the compute port.

protocol: *string* , x ∈ { "FC" , "iSCSI" }

The protocol for connecting compute ports.

type: *string* , x ∈ { "Target" , "Initiator" , "Universal" }

The type of the compute port.

nickname: *string* , must match /^[a-zA-Z0-9!#\$%&'\+\-\.=@\^_\{\}~\(\)\[\]:]{1,32}$/

The compute port nickname. Each compute port must have its own unique nickname.

name: ***string***

The WWN of the allocation destination compute port of the target operation for FC connection, or the iSCSI name for iSCSI connections.

The same name cannot be used for multiple compute ports.

configuredPortSpeed: ***string*** **, x ∈ { "16G" , "32G" , "Auto" }**

Link speed setting. The actual link speed is determined based on this setting.

- Auto: The speed is automatically set according to the switch and FC cable specifications.
- 16G, 32G: The compute port runs at the set speed even if communication is possible at a speed faster than the set value.

portSpeed: ***string*** **, x ∈ { "1G" , "2G" , "4G" , "8G" , "10G" , "16G" , "25G" , "32G" , "40G" , "Unknown" , "LinkDown", "DependsOnHypervisor" }**

Actual link speed (unit: bps).

If configuredPortSpeed is Auto, a value is output as per the actual cable or switch specifications.

- Unknown: The status is unknown.
- LinkDown: Link down occurred.
- DependsOnHypervisor: Depends on the hypervisor setting.

(Virtual machine) DependsOnHypervisor is always output for iSCSI connection configuration.

portSpeedDuplex: ***string*** **, x ∈ { "10Mbps Half" , "10Mbps Full" , "100Mbps Half" , "100Mbps Full" , "1Gbps Half" , "1Gbps Full" , "2.5Gbps Full" , "5Gbps Full" , "10Gbps Full" , "20Gbps Full" , "25Gbps Full" , "40Gbps Full" , "50Gbps Full" , "56Gbps Full" , "100Gbps Full" , "200Gbps Full" , "400Gbps Full" , "1G" , "8G" , "10G" , "16G" , "25G" , "32G" , "40G" , "Unknown" , "LinkDown" , "DependsOnHypervisor" }**

Actual link speed and duplex settings of the physical port used for communication. Only link speed is displayed for FC connection configuration.

If configuredPortSpeed is Auto, a value is output as per the actual cable or switch specifications.

- Unknown: The status is unknown.
- LinkDown: Link down occurred.
- DependsOnHypervisor: Depends on the hypervisor setting.

(Virtual machine) DependsOnHypervisor is always output for iSCSI connection configuration.

protectionDomainId: *string* (uuid)
The ID of the protection domain to which the volume is belonging.

storageNodeId: *string* (uuid)
The ID of the storage node that has compute ports.

interfaceName: *string*
An interface name, which is unique within a storage node that contains compute ports, control ports, and ports between storage nodes. Example: eth0, eth1

statusSummary: *string* , x ∈ { "Normal", "Warning", "Error" }
The summary of the compute port status.

- Normal: No action by the user is required.
- Warning: Although immediate action by the user is not required, some action may have to be taken.
- Error: Immediate action by the user is required.

status: *string* , x ∈ { "Normal" , "Error" , "MaintenanceBlockage" }
The status of the compute port.

- Normal: Available.
- Error: Unavailable.
- MaintenanceBlockage: Unavailable (during maintenance blockade).

fcInformation: fcTarget: object (on page 617) nullable

iscsiInformation: iscsiUniversal: object (on page 625) nullable

portAuthSetting: *object*

DESCRIPTION

Information about the authentication settings for the compute port for the target operation.

PROPERTIES

id: *string* (uuid)
The ID of the compute port.

authMode: ***string*** **, x ∈ { "CHAP" , "CHAPComplyingWithInitiatorSetting" , "None" }**

Authentication scheme of the compute port.

- CHAP: CHAP authentication.
- CHAPComplyingWithInitiatorSetting: Complies with the setting of the compute node. If the setting is "CHAP", CHAP authentication is performed. If the setting is "None", no authentication is required.
- None: No authentication is performed.

isDiscoveryChapAuth: ***boolean***

Enables or disables CHAP authentication at the time of discovery in iSCSI connection. Enables CHAP authentication at the time of discovery when true is specified.

isMutualChapAuth: ***boolean***

Enables or disables mutual CHAP authentication. Enables mutual CHAP authentication when true is specified.

portPerformance: *object*

DESCRIPTION

Performance information of the compute port (monitor information). Null is set if the information could not be obtained. The infinite value (Infinity) is not output for double type values.

PROPERTIES

id: ***string*** **(uuid)**

The ID of the compute port.

fc: fcPortPerformance: object (on page 616) nullable

iscsi: iscsiPortPerformance: object (on page 625) nullable

portPerformanceListResponse: *object*

DESCRIPTION

A list of compute port performance information (monitor information) at the specified time.

PROPERTIES

data: ***object[]***

ITEMS

poolPerformanceListResponseData: object (on page 660)

portPerformanceListResponseData: *object*

DESCRIPTION

The performance information (monitor information) of the compute port when the information was collected. For performanceObjects, compute ports of which performance information could be collected when the API was executed are output as an array.

PROPERTIES

timestamp: *string* (date-time)
The time when this information was collected.

performanceObjects: *object[]*
A list of compute port performance information (monitor information) collected at the time specified for timestamp.

ITEMS
portPerformance: object (on page 663)

portSummary: *object*

DESCRIPTION

Summary information about the compute port.

PROPERTIES

id: *string* (uuid)
The ID of the compute port.

protocol: *string* , x ∈ { "FC" , "iSCSI" }
The protocol for connecting compute ports.

type: *string* , x ∈ { "Target" , "Initiator" , "Universal" }
The type of the compute port.

nickname: *string* , must match /^[a-zA-Z0-9!#\$%&'\+\-\.=@\^_\{\}~\(\)\[\]:]{1,32}$/
The compute port nickname. Each compute port must have its own unique nickname.

name: *string*
The WWN of the allocation destination compute port of the target operation for FC connection, or the iSCSI name for iSCSI connections.

The same name cannot be used for multiple compute ports.

configuredPortSpeed: ***string*** **, x ∈ { "16G" , "32G" , "Auto" }**

Link speed setting. The actual link speed is determined based on this setting.

- Auto: The speed is automatically set according to the switch and FC cable specifications.
- 16G, 32G: The compute port runs at the set speed even if communication is possible at a speed faster than the set value.

portSpeed: ***string*** **, x ∈ { "1G" , "2G" , "4G" , "8G" , "10G" , "16G" , "25G" , "32G" , "40G" , "Unknown" , "LinkDown", "DependsOnHypervisor" }**

Actual link speed (unit: bps).

If configuredPortSpeed is Auto, a value is output as per the actual cable or switch specifications.

- Unknown: The status is unknown.
- LinkDown: Link down occurred.
- DependsOnHypervisor: Depends on the hypervisor setting.

(Virtual machine) DependsOnHypervisor is always output for iSCSI connection configuration.

portSpeedDuplex: ***string*** **, x ∈ { "10Mbps Half" , "10Mbps Full" , "100Mbps Half" , "100Mbps Full" , "1Gbps Half" , "1Gbps Full" , "2.5Gbps Full" , "5Gbps Full" , "10Gbps Full" , "20Gbps Full" , "25Gbps Full" , "40Gbps Full" , "50Gbps Full" , "56Gbps Full" , "100Gbps Full" , "200Gbps Full" , "400Gbps Full" , "1G" , "8G" , "10G" , "16G" , "25G" , "32G" , "40G" , "Unknown" , "LinkDown" , "DependsOnHypervisor" }**

Actual link speed and duplex settings of the physical port used for communication. Only link speed is displayed for FC connection configuration.

If configuredPortSpeed is Auto, a value is output as per the actual cable or switch specifications.

- Unknown: The status is unknown.
- LinkDown: Link down occurred.
- DependsOnHypervisor: Depends on the hypervisor setting.

(Virtual machine) DependsOnHypervisor is always output for iSCSI connection configuration.

protectionDomainId: ***string*** **(uuid)**

The ID of the protection domain to which the volume is belonging.

storageNodeId: ***string*** **(uuid)**

The ID of the storage node that has compute ports.

interfaceName: ***string***

An interface name, which is unique within a storage node that contains compute ports, control ports, and ports between storage nodes. Example: eth0, eth1

statusSummary: *string* , x ∈ { "Normal", "Warning", "Error" }
The summary of the compute port status.

- Normal: No action by the user is required.
- Warning: Although immediate action by the user is not required, some action may have to be taken.
- Error: Immediate action by the user is required.

status: *string* , x ∈ { "Normal" , "Error" , "MaintenanceBlockage" }
The status of the compute port.

- Normal: Available.
- Error: Unavailable.
- MaintenanceBlockage: Unavailable (during maintenance blockade).

fcInformation: fcTarget: object (on page 617) nullable

iscsiInformation: iscsiUniversal: object (on page 625) nullable

primaryServerOfLdapServerConnectionVerificationError: *object*

DESCRIPTION

Error information in the primary LDAP server.

If the primary LDAP server did not return an error, primaryServer returns a null instead of the object.

If the primary LDAP server is not registered, primaryServer also returns a null.

PROPERTIES

code:*integer* (int32)
LDAP protocol error code when connecting the primary LDAP server.

message: *string*
Description of the LDAP protocol error code when connecting the primary LDAP server.

primaryServerOfLdapServerConnectionVerificationResult: *object*

DESCRIPTION

Result of primary LDAP server connection verification.

If the primary LDAP server could not be reached or it returned an error, primaryServer returns a null instead of the object.

If the primary LDAP server is not registered, primaryServer also returns a null.

PROPERTIES

numberOfExternalUsers: *integer* (int64)
The number of users who could be searched on the primary LDAP server and can be externally authenticated.

0 means that communication was unsuccessful or no user could be found because the DN was set incorrectly.

numberOfExternalUserGroups: *integer* (int64)
The number of user groups who could be searched on the primary LDAP server and can be externally authenticated.

0 means that communication was unsuccessful or no user group could be found because the DN was set incorrectly.

protectionDomain: *object*

DESCRIPTION

Information about the protection domain.

PROPERTIES

totalPhysicalCapacity: *integer* (int64)
Total drive capacity in a protection domain (unit: MiB)

isFastRebuildEnabled: *boolean*
Whether fast rebuild is enabled or disabled. Only "true" is output.

id: *string* (uuid)
Protection domain ID.

name: *string* (1 to 32 chars)
The name of the protection domain.

redundantPolicy: *string* , x ∈ { "Mirroring" , "HitachiPolyphaseErasureCoding" }
User data protection method.

redundantType: *string* , x ∈ { "Duplication" , "2D+1P" , "4D+1P", "4D+2P" }
User data protection type.

driveDataRelocationStatus: *string* , x ∈ { "Stopped" , "Running" , "Error" , "Suspended" }
Status of drive data relocation.

- Stopped: Status in which drive data relocation is not being performed.
- Running: Status in which drive data relocation is being performed.
- Error: Status in which drive data relocation cannot be performed due to an error.
- Suspended: Status in which drive data relocation is suspended at the request of the user.

driveDataRelocationProgressRate: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 100 } nullable
Progress rate (%) of drive data relocation.

Null is output if driveDataRelocationStatus is "Stopped".

rebuildStatus: *string* , x ∈ { "Stopped" , "Running" , "Error" }
Rebuild status.

- Stopped: Status in which rebuild processing is not being performed.
- Running: Status in which rebuild processing is being performed.
- Error: Status in which rebuild processing cannot be performed due to an error.

rebuildProgressRate: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 100 } nullable
Progress rate (%) of rebuild processing.

Null is output if rebuildStatus is "Stopped".

memoryMode: *string* , x ∈ { "VolatileMemory" }
Information about memory in use.

- VolatileMemory: Volatile memory.

asyncProcessingResourceUsageRate: *string* , x ∈ { "VeryHigh", "High" , "Middle" , "Low" }
Controls the resource usage rate of the internal processing I/O.

numberOfFaultDomains: *integer*(int32) , { x ∈ ℤ | 1 ≤ x ≤ 1024 }
The total number of fault domains in the protection domain.

storageControllerClusteringPolicy: *string* , x ∈ { "OneRedundantStorageNode" , "TwoRedundantStorageNodes" }

Cluster policy of the storage controller. This represents the number of storage nodes on which the standby storage controller node runs when a storage controller cluster is configured.

- OneRedundantStorageNode: The standby storage controller node runs on a single storage node.

- TwoRedundantStorageNodes: The standby storage controller node runs on two storage nodes.

minimumMemorySize: *integer*(int64), { x ∈ ℤ | 0 ≤ x ≤ 268435456 }

Smallest value of memory among storage nodes belonging to the protection domain (unit: MiB).

protectionDomainSummary: *object*

DESCRIPTION

Summary information about protection domain.

PROPERTIES

id: *string* (uuid)

Protection domain ID.

name: *string* (1 to 32 chars)

The name of the protection domain.

redundantPolicy: *string* , x ∈ { "Mirroring" , "HitachiPolyphaseErasureCoding" }

User data protection method.

redundantType: *string* , x ∈ { "Duplication" , "2D+1P" , "4D+1P", "4D+2P" }

User data protection type.

driveDataRelocationStatus: *string* , x ∈ { "Stopped" , "Running" , "Error" , "Suspended" }

Status of drive data relocation.

- Stopped: Status in which drive data relocation is not being performed.

- Running: Status in which drive data relocation is being performed.

- Error: Status in which drive data relocation cannot be performed due to an error.

- Suspended: Status in which drive data relocation is suspended at the request of the user.

driveDataRelocationProgressRate: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 100 } nullable

Progress rate (%) of drive data relocation.

Null is output if driveDataRelocationStatus is "Stopped".

rebuildStatus: *string* , x ∈ { "Stopped" , "Running" , "Error" }

Rebuild status.

- Stopped: Status in which rebuild processing is not being performed.
- Running: Status in which rebuild processing is being performed.
- Error: Status in which rebuild processing cannot be performed due to an error.

rebuildProgressRate: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 100 } nullable

Progress rate (%) of rebuild processing.

Null is output if rebuildStatus is "Stopped".

memoryMode: *string* , x ∈ { "VolatileMemory" }

Information about memory in use.

- VolatileMemory: Volatile memory.

asyncProcessingResourceUsageRate: *string* , x ∈ { "VeryHigh", "High" , "Middle" , "Low" }

Controls the resource usage rate of the internal processing I/O.

numberOfFaultDomains: *integer* (int32) , { x ∈ ℤ | 1 ≤ x ≤ 1024 }

The total number of fault domains in the protection domain.

storageControllerClusteringPolicy: *string* , x ∈ { "OneRedundantStorageNode" , "TwoRedundantStorageNodes" }

Cluster policy of the storage controller. This represents the number of storage nodes on which the standby storage controller node runs when a storage controller cluster is configured.

- OneRedundantStorageNode: The standby storage controller node runs on a single storage node.
- TwoRedundantStorageNodes: The standby storage controller node runs on two storage nodes.

minimumMemorySize: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 268435456 }

Smallest value of memory among storage nodes belonging to the protection domain (unit: MiB).

rebuildableResourcesOfPool: *object*

DESCRIPTION

Information about resources for which Rebuild is possible.

PROPERTIES

numberOfDrives: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 23 } nullable**
The number of drives for which Rebuild is possible.

rebuildableResourcesOfStorageNode: *object*

DESCRIPTION

Information about resources for which Rebuild is possible.

PROPERTIES

numberOfDrives: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 23 } nullable**
The number of drives for which Rebuild is possible.

rebuildCapacityResourceSettingParam: *object*

DESCRIPTION

A parameter for the resource type and number of resources of rebuild capacity.

PROPERTIES

numberOfTolerableDriveFailures: ***integer*****(int32), { x ∈ ℤ | 0 ≤ x ≤ 23 }**
The number of drive failures that can be tolerated.

releaseFromServerVolumeParam: *object*

DESCRIPTION

A parameter for releasing multiple connections between the volumes and compute nodes. Deallocate all the volumes specified in volumeIds from all the compute nodes that are specified in serverIds.

PROPERTIES

volumeIds: *string[]* (1 to 1000 items) required

A list of volume IDs to be deallocated.

ITEMS

string (uuid)

vpsId: *string* , must match /^system$|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

The ID of the operation-target virtual private storage (VPS).

To specify a resource that does not belong to the VPS, set "system" for this property.

If this property is omitted, the VPS that the user who runs the API belongs to is assumed to be the operation target.

serverIds: *string[]* (1 to 100 items) required

A list of compute node IDs whose volume is to be deallocated.

ITEMS

string (uuid)

releaseFromSingleServerSingleVolumeParam: *object*

DESCRIPTION

A parameter for releasing the connection between a volume and a compute node.

PROPERTIES

vpsId: *string* , must match /^system$|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

The ID of the operation-target virtual private storage (VPS).

To specify a resource that does not belong to the VPS, set "system" for this property.

If this property is omitted, the VPS that the user who runs the API belongs to is assumed to be the operation target.

replaceStorageNodeParam: *object*

DESCRIPTION

A parameter for storage node replacement.

PROPERTIES

setupUserPassword: *string* (1 to 255 chars) , must match /^[\-A-Za-z0-9!#\$%&'"\(\)*\+,\.\/:;<>=\?@\[\]\\\^_`\{\}\|~]{1,256}$/ required

Setup user password.

request: *object*

DESCRIPTION

Request information.

PROPERTIES

requestUrl: *string* (link)

A URL requested by an asynchronous processing API. If a character string is longer than 2048 bytes, it is displayed in abbreviated form (in-between characters of the string are trimmed).

requestMethod: *string* , x ∈ { "GET" , "PATCH" , "POST" , "DELETE" }

HTTP method requested by an asynchronous processing API.

requestBody: *string* nullable

Request body when requested by an asynchronous processing API. If a character string is longer than 1024 bytes, it is displayed in abbreviated form (in-between characters of the string are trimmed).

A parameter which is not specified by the user or a parameter for which its default is specified is not output. Only parameters specified in the application/json form are output. Parameters in the multipart/form-data form are not output.

requestAuthenticationSettingOfEditSnmpSetting: *object*

DESCRIPTION

A parameter for editing the setting of whether to permit SNMP requests.

If "requestAuthenticationSetting" is omitted, all requestAuthenticationSetting settings are deleted.

PROPERTIES

snmpv2cSettings: *object[]* (1 to 3 items) required

A list of settings for whether to permit SNMPv2c requests.

ITEMS

snmpv2cSettingOfEditRequestAuthenticationSetting: object (on page 693)

requestAuthenticationSettingOfSnmpSetting: *object*

DESCRIPTION

Setting for whether to permit SNMP requests.

PROPERTIES

snmpv2cSettings: *object[]* (up to 3 items)
A list of settings for whether to permit SNMPv2c requests.

ITEMS
snmpv2cSettingOfRequestAuthenticationSetting: object (on page 694)

resourceStatusOfHealthStatus: *object*

DESCRIPTION

Status information of each resource in the health status.

PROPERTIES

type: *string* , x ∈ { "Storage", "StorageNode", "Pool", "Port", "Drive", "Volume" , "License" , "FaultDomain" , "ControlPort" , "InternodePort" }
Resource type.

Object in "/v1/objects/<*object*>" in the URI.

- Storage: Storage cluster. You can obtain resource information by "GET /v1/objects/storage".
- StorageNode: Storage node. You can obtain resource information by "GET /v1/objects/storage-nodes".
- Pool: Storage pool. You can obtain resource information by "GET /v1/objects/pools".
- Port: Compute port. You can obtain resource information by "GET /v1/objects/ports".
- Drive: Drive. You can obtain resource information by "GET /v1/objects/drives".
- Volume: Volume. You can obtain resource information by "GET /v1/objects/volumes".
- License: License. You can obtain resource information by "GET /v1/objects/licenses".

- FaultDomain: Fault domain. You can obtain resource information by "GET /v1/objects/fault-domains".
- ControlPort: Control port. You can obtain resource information by "GET /v1/objects/control-ports".
- InternodePort: Internode port. You can obtain resource information by "GET /v1/objects/internode-ports".

status:*string* , x ∈ { "Normal" , "Alerting" }

Resource status of the type which belongs to the protection domain of protectionDomainId in the health status.

Status of the resource indicated by "type" managed on a cluster-wide basis (not on each protection domain basis) if protectionDomainId is null.

statusSummary of the applicable resource contains one or more warnings or errors, the Alerting status applies.

protectionDomainId: *string* (uuid) nullable

The ID of the protection domain.

Null is output if the type indicates a resource which is managed on a storage cluster basis (not for each protection domain).

The types for which null is output are "Storage", "License", or "FaultDomain".

restoreSnapshotParam: *object*

DESCRIPTION

A parameter for the Snapshot restoration.

PROPERTIES

snapshotVolumeId: *string* (uuid) required

The ID of the volume which is S-VOL. Restores S-VOL of the parameter specified to P-VOL.

vpsId: *string* , must match /^system$|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

The ID of the operation-target virtual private storage (VPS).

To specify a resource that does not belong to the VPS, set "system" for this property.

If this property is omitted, the VPS that the user who runs the API belongs to is assumed to be the operation target.

resultOfLdapServerConnectionVerification: *object*

DESCRIPTION

Result of verification of the connection with the LDAP server.

PROPERTIES

primaryServer: primaryServerOfLdapServerConnectionVerificationResult: object (on page 667) nullable

secondaryServer: secondaryServerOfLdapServerConnectionVerificationResult: object (on page 682) nullable

savingEffectOfPool: *object*

DESCRIPTION

Effect of the data reduction function on the storage pool.

PROPERTIES

efficiencyDataReduction: *integer* (int32) , { x ∈ ℤ | 100 ≤ x ≤ 100 } nullable

Ratio of the data capacity before and after data reduction (unit: %).

efficiencyDataReduction is currently fixed to 100.

-1 indicates that the value is invalid.

Reduction effect by the Snapshot function is not included.

preCapacityDataReduction: *integer* (int64) , { x ∈ ℤ | -1 ≤ x ≤ 6871947674 } nullable

Data capacity before reduction by the data reduction function (unit: MiB).

-1 indicates that the value is invalid.

postCapacityDataReduction: *integer* (int64) , { x ∈ ℤ | -1 ≤ x ≤ 6871947674 } nullable

Data capacity after reduction by the data reduction function (unit: MiB).

-1 indicates that the value is invalid.

totalEfficiencyStatus: *string* , x ∈ { "Valid" , "CalculationInProgress" , "NoTargetData" , "Unknown" } nullable

Status of total efficiency.

- Valid: Valid.
- CalculationInProgress: Calculation is in progress.

- NoTargetData: No data to be calculated exists.
- Unknown: Indicates that the calculated value is invalid. This status is set when the storage pool capacity is not expanded. This status is also set when volumes are created under all storage nodes and the consumed amount of the storage pool is 0.

dataReductionWithoutSystemDataStatus: *string* , x ∈ { "Valid" , "NotSupported" , "CalculationInProgress" , "NoTargetData" , "Unknown" } nullable

Status of the reduction effect of the data reduction function.

- Valid: Valid.
- NotSupported: Not supported.
- CalculationInProgress: Calculation is in progress.
- NoTargetData: No data to be calculated exists.
- Unknown: Indicates that the calculated value is invalid. This status is set when the storage pool capacity is not expanded.

totalEfficiency: *integer*(int64) , { x ∈ ℤ | -1 ≤ x ≤ 9223372036854775807 } nullable

Indicates the effect of volume creation and snapshot functions on capacity consumption.

Indicates weighted average of the ratio (according to total capacity of created volumes for each storage controller) of total capacity of created volumes (totalVolumeCapacity) to used capacity of the storage pool (usedCapacity) for each storage controller (unit: %).

The greater the total capacity of created volumes of a storage controller, the greater the effect of saving capacity reflected in the totalEfficiency value.

For example, if the total capacity of created volumes is 1,000 and used capacity of the storage pool is 50 for each storage controller, 2,000 (unit: %) is output.

dataReductionWithoutSystemData: *integer* (int64) , { x ∈ ℤ | -1 ≤ x ≤ 9223372036854775807 } nullable

Ratio of reduction efficiency in the entire storage pool after reduction by the data reduction function (unit: %).

The value before data reduction is output when the value after data reduction is assumed to be 100.

For example, 138 is output if the ratio of the value before data reduction to the value after data reduction is 138:100.

preCapacityDataReductionWithoutSystemData: *integer* (int64) , { x ∈ ℤ | -1 ≤ x ≤ 6871947674 } nullable

Capacity of the entire storage pool before reduction by the data reduction function (MiB).

There may be an error with the sum of the properties with the same name, which is displayed in the effect of the data reduction function for each volume (savingEffectOfVolume, savingEffectOfVolumeSummary).

postCapacityDataReductionWithoutSystemData: *integer* (int64) , { x ∈ ℤ | -1 ≤ x ≤ 6871947674 } nullable

Capacity of the entire storage pool after reduction by the data reduction function (unit: MiB).

calculationStartTime: *string* (date-time) nullable

Date and time the calculation started.

calculationEndTime: *string* (date-time) nullable

Date and time the calculation ended.

savingEffectOfStorage: *object*

DESCRIPTION

Effect of the data reduction function on the entire storage cluster.

PROPERTIES

efficiencyDataReduction: *integer* (int32) , { x ∈ ℤ | 100 ≤ x ≤ 100 } nullable

Ratio of the data capacity before and after data reduction (unit: %).

efficiencyDataReduction is currently fixed to 100.

-1 indicates that the value is invalid.

Reduction effect by the Snapshot function is not included.

preCapacityDataReduction: *integer* (int64) , { x ∈ ℤ | -1 ≤ x ≤ 6871947674 } nullable

Data capacity before reduction by the data reduction function (unit: MiB).

-1 indicates that the value is invalid.

postCapacityDataReduction: *integer* (int64) , { x ∈ ℤ | -1 ≤ x ≤ 6871947674 } nullable

Data capacity after reduction by the data reduction function (unit: MiB).

-1 indicates that the value is invalid.

totalEfficiencyStatus: *string* , x ∈ { "Valid" , "CalculationInProgress" , "NoTargetData" , "Unknown" } nullable

Status of total efficiency.

- Valid: Valid.
- CalculationInProgress: Calculation is in progress.

- NoTargetData: No data to be calculated exists.
- Unknown: Indicates that the calculated value is invalid. This status is set when the storage pool capacity is not expanded. This status is also set when volumes are created under all storage nodes and the consumed amount of the storage pool is 0.

dataReductionWithoutSystemDataStatus: *string* , x ∈ { "Valid" , "NotSupported" , "CalculationInProgress" , "NoTargetData" , "Unknown" } nullable

Status of the reduction effect of the data reduction function.

- Valid: Valid.
- NotSupported: Not supported.
- CalculationInProgress: Calculation is in progress.
- NoTargetData: No data to be calculated exists.
- Unknown: Indicates that the calculated value is invalid. This status is set when the storage pool capacity is not expanded.

totalEfficiency: *integer* (int64) , { x ∈ ℤ | -1 ≤ x ≤ 9223372036854775807 } nullable

Indicates the effect of volume creation and snapshot functions on capacity consumption.

Indicates weighted average of the ratio (according to total capacity of created volumes for each storage controller) of total capacity of created volumes (totalVolumeCapacity) to used capacity of the storage pool (usedCapacity) for each storage controller (unit: %).

The greater the total capacity of created volumes of a storage controller, the greater the effect of saving capacity reflected in the totalEfficiency value.

For example, if the total capacity of created volumes is 1,000 and used capacity of the storage pool is 50 for each storage controller, 2,000 (unit: %) is output.

dataReductionWithoutSystemData: *integer* (int64) , { x ∈ ℤ | -1 ≤ x ≤ 9223372036854775807 } nullable

Ratio of reduction efficiency in the entire storage pool after reduction by the data reduction function (unit: %).

The value before data reduction is output when the value after data reduction is assumed to be 100.

For example, 138 is output if the ratio of the value before data reduction to the value after data reduction is 138:100.

preCapacityDataReductionWithoutSystemData: *integer* (int64) , { x ∈ ℤ | -1 ≤ x ≤ 6871947674 } nullable

Capacity of the entire storage pool before reduction by the data reduction function (MiB).

There may be an error with the sum of the properties with the same name, which is displayed in the effect of the data reduction function for each volume (savingEffectOfVolume, savingEffectOfVolumeSummary).

postCapacityDataReductionWithoutSystemData: *integer* (int64) , { x ∈ ℤ | -1 ≤ x ≤ 6871947674 } nullable
Capacity of the entire storage pool after reduction by the data reduction function (unit: MiB).

calculationStartTime: *string* (date-time) nullable
Date and time the calculation started.

calculationEndTime: *string* (date-time) nullable
Date and time the calculation ended.

savingEffectOfVolume: *object*

DESCRIPTION

Effect of the data reduction function.

PROPERTIES

dataReductionRate: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 2147483647 } nullable
Ratio of the data capacity before and after data reduction (unit: %).

dataReductionRate is currently fixed to null or 0.

dataReductionCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 268435456 } nullable
Amount reduced by the data reduction function (unit: MiB).

The amount of metadata and garbage data by the data reduction function are included.

dataReductionCapacity is currently fixed to null or 0.

compressedCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 268435456 } nullable
Amount reduced by compression (unit: MiB).

compressedCapacity is currently fixed to null or 0.

reclaimedCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 268435456 } nullable
Amount reduced by fixed pattern exclusion (unit: MiB).

reclaimedCapacity is currently fixed to null or 0.

systemDataCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 268435456 } nullable
The amount of system data consumed by the data reduction function (amount of metadata and garbage data) (unit: MiB).

systemDataCapacity is currently fixed to null or 0.

preCapacityDataReductionWithoutSystemData: ***integer*** **(int64) , { x ∈ ℤ | 0 ≤ x ≤ 268435456 } nullable**

Consumed amount of the storage pool before reduction by the data reduction function (unit: MiB).

preCapacityDataReductionWithoutSystemData is currently fixed to null or 0.

postCapacityDataReduction: ***integer*** **(int64) , { x ∈ ℤ | 0 ≤ x ≤ 268435456 } nullable**

Consumed amount of the storage pool after reduction by the data reduction function (unit: MiB).

postCapacityDataReduction is currently fixed to null or 0.

savingEffectOfVolumeSummary: *object*

DESCRIPTION

Effect of the data reduction function on the volume summary information.

PROPERTIES

systemDataCapacity: ***integer*** **(int64) , { x ∈ ℤ | 0 ≤ x ≤ 268435456 } nullable**

The amount of system data consumed by the data reduction function (amount of metadata and garbage data) (unit: MiB).

systemDataCapacity is currently fixed to null or 0.

preCapacityDataReductionWithoutSystemData: ***integer*** **(int64) , { x ∈ ℤ | 0 ≤ x ≤ 268435456 } nullable**

Consumed amount of the storage pool before reduction by the data reduction function (unit: MiB).

preCapacityDataReductionWithoutSystemData is currently fixed to null or 0.

postCapacityDataReduction: ***integer*** **(int64) , { x ∈ ℤ | 0 ≤ x ≤ 268435456 } nullable**

Consumed amount of the storage pool after reduction by the data reduction function (unit: MiB).

PostCapacityDataReduction is currently fixed to null or 0.

secondaryServerOfLdapServerConnectionVerificationError: *object*

DESCRIPTION

Error information in the secondary LDAP server.

If the secondary LDAP server did not return an error, the secondary server returns a null instead of the object.

If the secondary LDAP server is not registered, secondaryServer also returns a null.

PROPERTIES

code: ***integer*** **(int32)**

LDAP protocol error code when connecting the secondary LDAP server.

message: ***string***

Description of the LDAP protocol error code when connecting the secondary LDAP server.

secondaryServerOfLdapServerConnectionVerificationResult: *object*

DESCRIPTION

Result of secondary LDAP server connection verification.

If the secondary LDAP server could not be reached or it returned an error, secondaryServer returns a null instead of the object.

If the secondary LDAP server is not registered, secondaryServer also returns a null.

PROPERTIES

numberOfExternalUsers: ***integer*** **(int64)**

The number of users who could be searched on the secondary LDAP server and can be externally authenticated.

0 means that communication was unsuccessful or no user could be found because the DN was set incorrectly.

numberOfExternalUserGroups: ***integer*** **(int64)**

The number of user groups who could be searched on the secondary LDAP server and can be externally authenticated.

0 means that communication was unsuccessful or no user group could be found because the DN was set incorrectly.

sendingTrapSettingOfEditSnmpSetting: *object*

DESCRIPTION

A parameter for editing SNMP trap transmission destination settings.

If "sendingTrapSetting" is omitted, all sendingTrapSetting settings are deleted.

PROPERTIES

snmpv2cSettings: ***object[]*** **(1 to 3 items) required**

A list of SNMPv2c trap transmission destination settings.

ITEMS

snmpv2cSettingOfEditSendingTrapSetting: object (on page 694)

sendingTrapSettingOfSnmpSetting: *object*

DESCRIPTION

SNMP trap transmission destination settings.

PROPERTIES

snmpv2cSettings: *object[]* **(up to 3 items)**

A list of SNMPv2c trap transmission destination settings.

ITEMS

snmpv2cSettingOfSendingTrapSetting: object (on page 694)

server: *object*

DESCRIPTION

Compute node information.

PROPERTIES

numberOfVolumes: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 8192 }

The number of volumes already allocated.

paths: *object[]* nullable

A list of information items of registered paths.

ITEMS

pathOfServer: object (on page 654)

id: *string* (uuid)

The ID of the compute node.

nickname: *string* (1 to 229 chars)

The compute node nickname. The same nickname must be unique across compute nodes.

osType: *string* , x ∈ { "Linux" , "VMware" , "Windows" }

The OS type of the compute node.

totalCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 2199023255552 }

Total capacity of the volumes on the storage pool allocated to the compute node (unit: MiB).

Any volume whose "volumeType" is "ExternalMigrationOrigin" is not included. 0 is output if volumes whose "volumeType" is "ExternalMigrationOrigin" are allocated to the compute node.

usedCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 2199023255552 } nullable

Consumed amount of the volumes on the storage pool allocated to the compute node (unit: MiB).

Any volume whose "volumeType" is "ExternalMigrationOrigin" is not included. 0 is output if volumes whose "volumeType" is "ExternalMigrationOrigin" are allocated to the compute node.

numberOfPaths: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 8192 }

The number of registered paths.

vpsId: *string* , must match /^\(system\)$|[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

The ID of the virtual private storage (VPS) that the acquisition-target resource belongs to.

If the resource does not belong to a VPS, the reserved word "(system)", which indicates that the resource is independent, is output instead of an ID.

vpsName: *string* (1 to 32 chars) , must match /^\(system\)$|[\-A-Za-z0-9,\.:@_]{1,32}$/

The name of the virtual private storage (VPS) that the acquisition-target resource belongs to.

If the resource does not belong to a VPS, the reserved word "(system)", which indicates that the resource is independent, is output.

serverSummary: *object*

DESCRIPTION

Summary information of the compute node.

PROPERTIES

id: *string* (uuid)

The ID of the compute node.

nickname: *string* (1 to 229 chars)

The compute node nickname. The same nickname must be unique across compute nodes.

osType: *string* , x ∈ { "Linux" , "VMware" , "Windows" }

The OS type of the compute node.

totalCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 2199023255552 }

Total capacity of the volumes on the storage pool allocated to the compute node (unit: MiB).

Any volume whose "volumeType" is "ExternalMigrationOrigin" is not included. 0 is output if volumes whose "volumeType" is "ExternalMigrationOrigin" are allocated to the compute node.

usedCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 2199023255552 } nullable

Consumed amount of the volumes on the storage pool allocated to the compute node (unit: MiB).

Any volume whose "volumeType" is "ExternalMigrationOrigin" is not included. 0 is output if volumes whose "volumeType" is "ExternalMigrationOrigin" are allocated to the compute node.

numberOfPaths: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 8192 }

The number of registered paths.

vpsId: *string* , must match /^\(system\)$|[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

The ID of the virtual private storage (VPS) that the acquisition-target resource belongs to.

If the resource does not belong to a VPS, the reserved word "(system)", which indicates that the resource is independent, is output instead of an ID.

vpsName: *string* (1 to 32 chars) , must match /^\(system\)$|[\-A-Za-z0-9,\.:@_]{1,32}$/

The name of the virtual private storage (VPS) that the acquisition-target resource belongs to.

If the resource does not belong to a VPS, the reserved word "(system)", which indicates that the resource is independent, is output.

session: *object*

DESCRIPTION

Session information.

PROPERTIES

sessionId: *string* (uuid)

Session ID.

userId: *string* (5 to 255 chars) , must match /^[\-A-Za-z0-9!#\$%&'\.@\^_`\{\}~]{5,255}$/

User ID.

userObjectId: *string* (5 to 765 chars)

Object ID of the user. For the user ID, a percent-encoded reserved string defined in RFC 3986 is output.

expirationTime: *string* (date-time)

Expiration time of the session. After this time, the session expires.

createdTime: *string* (date-time)

Date and time when the session is generated.

lastAccessTime: *string* (date-time)

Date and time when the session was last used.

roleNames:*string[]*

A list of roles assigned to the user who retains this session.

ITEMS

- For system administrators: (One to six items)

  string , x ∈ { "Security", "Storage", "Monitor", "Service", "Audit", "Resource" }

- For VPS administrators: (One to three items)

  string , x ∈ { "VpsSecurity", "VpsStorage", "VpsMonitor" }

vpsId: *string* , must match /^\(system\)$|[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

The ID of the virtual private storage (VPS) that the user belongs to.

If the user is a system administrator, the string "(system)" is displayed.

privileges:*object[]* (1 to 65 items)

A list of information about a virtual private storage (VPS) that the user can access.

ITEMS

userPrivileges: object (on page 727)

sessionSettingOfEditUserAuthSetting: *object*

DESCRIPTION

A parameter for setting a session.

PROPERTIES

maxLifetimeSeconds: *integer* (int32) , { x ∈ ℤ | 1800 ≤ x ≤ 604800 }

Token lifetime (unit: seconds). After authentication, authentication by the authentication token is enabled during this period.

maxIdleSeconds: *integer* (int32) , { x ∈ ℤ | 300 ≤ x ≤ 86400 }

Time until the session times out (unit: seconds). The session is disabled if the REST API server is not accessed using the session during this specified time after the server is accessed using the session.

sessionSettingOfUserAuthSetting: *object*

DESCRIPTION

Session settings.

PROPERTIES

maxLifetimeSeconds: *integer* (int32) , { x ∈ ℤ | 1800 ≤ x ≤ 604800 }

Token lifetime (unit: seconds). After authentication, authentication by the authentication token is enabled during this period.

maxIdleSeconds: *integer* (int32) , { x ∈ ℤ | 300 ≤ x ≤ 86400 }

Time until the session times out (unit: seconds). The session is disabled if the REST API server is not accessed using the session during this specified time after the server is accessed using the session.

shutdownStorageParam: *object*

DESCRIPTION

A parameter for stopping a storage cluster.

"true" cannot be specified for both "force" and "reboot" simultaneously. If "true" is specified for both, HTTP status code 400 (Bad Request) is returned.

PROPERTIES

force: *boolean* (default: "false")

Specify “true” for this parameter only when instructed to do so in a manual or by customer support.

Specifies whether to perform the operation forcibly. When true is specified, the operation is performed forcibly.

When true is specified for reboot or configParameterSettingMode, true cannot be specified here.

reboot: *boolean* (default: "false")

Specifies whether to restart the storage cluster. Specify false to stop it, or true to restart it.

When true is specified for force, true cannot be specified.

configParameterSettingMode: *boolean* (default: "false")

Specify “true” for this parameter only when instructed to do so in a manual or by customer support.

(Virtual machine) This parameter will be ignored if it is specified.

(Bare metal) Specifies whether to start the storage cluster in the configuration parameter setting mode after shutdown. If true is specified, the storage node cluster will start in the configuration parameter setting mode. When true is specified for force, true cannot be specified here.

smtpSettingOfEditEventLogSetting: *object*

DESCRIPTION

A parameter for editing the event log transfer destination in SMTP settings. One of toAddress1, toAddress2, or toAddress3 must be specified. If nothing is specified, HTTP status code 400 (Bad Request) is returned.

PROPERTIES

index: ***integer*** **(int32), { x ∈ ℤ | 1 ≤ x ≤ 1 } (default: "1") required**

The ID of the SMTP server.

isEnabled: ***boolean*** **required**

Whether event logs are transferred using the SMTP setting.

smtpServerName: ***string*** **(up to 253 chars) , must match /^$|^([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9])\.)*([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9]))$/ required**

Host name or IP address (IPv4) of the SMTP server.

smtpAuthAccount: ***string*** **(up to 255 chars) , must match /^$|^[0-9a-zA-Z!\$%'\(\)\-\.~_@]+$/ required**

SMTP authentication account.

smtpAuthPassword: ***string*** **(up to 255 chars) , must match /^$|^[0-9a-zA-Z!\$%'\(\)\-\.~_@]+$/ required**

SMTP authentication password.

fromAddress: ***string*** **(up to 255 chars) , must match /^$|^[0-9a-zA-Z@!#\$%&`\+\-*\/'\^\{\}_\.]+$/ required**

Source E-mail address.

toAddress1: ***string*** **(up to 255 chars) , must match /^$|^[0-9a-zA-Z@!#\$%&`\+\-*\/'\^\{\}_\.]+$/ required**

Destination E-mail address 1.

toAddress2: ***string*** **(up to 255 chars) , must match /^$|^[0-9a-zA-Z@!#\$%&`\+\-*\/'\^\{\}_\.]+$/ required**

Destination E-mail address 2.

toAddress3: ***string*** **(up to 255 chars) , must match /^$|^[0-9a-zA-Z@!#\$%&`\+\-*\/'\^\{\}_\.]+$/ required**

Destination E-mail address 3.

smtpSettingOfEventLogSetting: *object*

DESCRIPTION

A parameter for editing the event log transfer destination in SMTP settings.

PROPERTIES

index: *integer* (int32) , { x ∈ ℤ | 1 ≤ x ≤ 1 }
The ID of the SMTP server.

isEnabled: *boolean*
Whether event logs are transferred using the SMTP setting.

smtpServerName: *string* (up to 253 chars) , must match /^$|^([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9])\.)*([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9]))$/
Host name or IP address (IPv4) of the SMTP server.

port: *integer* (int32) , { x ∈ ℤ | 1 ≤ x ≤ 65535 }
Port number of the SMTP server.

connectionEncryptionType: *string* , x ∈ { "STARTTLS" , "SMTPS" , "None" }
Encryption type when the SMTP server is connected.

isSmtpAuthEnabled: *boolean*
Enables or disables SMTP authentication.

smtpAuthAccount: *string* (up to 255 chars) , must match /^$|^[0-9a-zA-Z!\$%'\(\)\-\.~_@]+$/
SMTP authentication account.

fromAddress: *string* (up to 255 chars) , must match /^$|^[0-9a-zA-Z@!#\$%&`\+\-*\/'\^\{\}_\.]+$/
Source E-mail address.

toAddress1: *string* (up to 255 chars) , must match /^$|^[0-9a-zA-Z@!#\$%&`\+\-*\/'\^\{\}_\.]+$/
Destination E-mail address 1.

toAddress2: *string* (up to 255 chars) , must match /^$|^[0-9a-zA-Z@!#\$%&`\+\-*\/'\^\{\}_\.]+$/
Destination E-mail address 2.

toAddress3: *string* (up to 255 chars) , must match /^$|^[0-9a-zA-Z@!#\$%&`\+\-*\/'\^\{\}_\.]+$/
Destination E-mail address 3.

snapshotVolume: *object*

DESCRIPTION

S-VOL information.

PROPERTIES

snapshotVolumeId: *string* (uuid)
The ID of the volume which is S-VOL.

snapshotVolumeName: *string* (1 to 32 chars)
The name of the volume which is S-VOL. The same name cannot be used in multiple volumes.

snapshotVolumeNickname: *string* (1 to 32 chars)
The nickname of the volume which is S-VOL. The same nickname can be used in multiple volumes.

statusSummary: *string* , x ∈ { "Normal" , "Warning" , "Error" }
The summary of the S-VOL status.

- Normal: No action by the user is required.
- Warning: Although immediate action by the user is not required, some action may have to be taken.
- Error: Immediate action by the user is required.

status: *string* , x ∈ { "Normal" , "Deleting" , "Updating" , "Expanding" , "CreationFailed" , "DeletionFailed" , "UpdateFailed" , "ExpansionFailed" , "IOSuppressed" }
The volume status. Any one of the following statuses is output:

- Normal: Operating normally.
- Deleting: Being deleted.
- Updating: The setting is being updated.
- Expanding: The capacity is being expanded.
- CreationFailed: Creation ended abnormally.
- DeletionFailed: Deleting ended abnormally.
- UpdateFailed: The setting update ended abnormally.
- ExpansionFailed: Capacity expansion did not succeed.
- IOSuppressed: I/O is suppressed.

snapshotStatus: *string* , x ∈ { "Normal" , "Deleting" , "Restoring" , "Empty " , "Preparing" , "Prepared" , "Error" }

The snapshot status. Any one of the following statuses is output:

- Normal: Operating normally.
- Deleting: Being deleted.
- Restoring: Being restored.
- Empty: Status in which snapshots are empty without metadata. This is a temporary status when creating or deleting S-VOLs. After waiting for a while, the status transitions to Preparing when creating S-VOLs and the applicable volumes are deleted when deleting S-VOLs.
- Preparing: Being prepared for creating snapshots.
- Prepared: Completed preparation for creating snapshots.
- Error: Operating abnormally.

snapshotProgressRate: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 100 } nullable

Progress rate of snapshot operation (unit: %). The progress rate is output only when snapshotStatus is one of Preparing, Deleting, or Restoring. In the case of other statuses, null is output.

snapshotTimestamp: *string* (date-time) nullable

The time when snapshots are obtained.

When snapshotStatus is other than Normal, null is output.

snapshotType: *string* , x ∈ { "Snapshot" }

The type of the snapshot.

snapshotConcordanceRate: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 100 } nullable

The matching rate between the snapshot creation-source volume (P-VOL) and the applicable volume (unit: %).

- The matching rate is output only when snapshotAttribute is "S-VOL" or "P/S-VOL".
- The matching rate is output when snapshotStatus is Normal or Prepared. In the case of other statuses, null is output.

isWrittenInSvol: *boolean* nullable

When snapshotStatus is Normal, whether writing is performed from the compute node to the volume is output.

- true: Writing to the volume is performed.
- false: Writing to the volume is not performed.

When snapshotStatus is other than Normal, null is output.

vpsId: ***string*** **, must match /^\(system\)$|[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/**

The ID of the virtual private storage (VPS) that the acquisition-target resource belongs to.

If the resource does not belong to a VPS, the reserved word "(system)", which indicates that the resource is independent, is output instead of an ID.

vpsName: ***string*** **(1 to 32 chars) , must match /^\(system\)$|[\-A-Za-z0-9,\.:@_]{1,32}$/**

The name of the virtual private storage (VPS) that the acquisition-target resource belongs to.

If the resource does not belong to a VPS, the reserved word "(system)", which indicates that the resource is independent, is output.

qosParam: ***object***

A QoS-related parameter.

PROPERTIES

upperLimitForIops: ***integer*** **(int64) , { x ∈ ℤ | -1 ≤ x ≤ 2147483647 }**

The upper limit of volume performance (in IOPS).

The value -1 indicates that no upper limit is placed on volume performance (in IOPS).

upperLimitForTransferRate: ***integer*** **(int64) , { x ∈ ℤ | -1 ≤ x ≤ 2097151 }**

The upper limit of volume performance (in MiB/s).

The value -1 indicates that no upper limit is placed on volume performance (in MiB/s).

upperAlertAllowableTime: ***integer*** **(int32) ,{ x ∈ ℤ | -1 ≤ x ≤ 600 }**

The alert threshold value (in seconds) for the upper limit of volume performance.

The value -1 indicates that no entries are output to the event log.

VPS administrators cannot specify this property when creating or editing volumes.

upperAlertTime: ***string*** **(date-time) nullable**

The last time the upper limit of volume performance was continuously exceeded and the conditions for the alert threshold of the performance upper limit were met (UTC).

snmpSetting: *object*

DESCRIPTION

SNMP settings.

PROPERTIES

isSNMPAgentEnabled: *boolean*
Enables or disables SNMP.

snmpVersion: *string* , x ∈ { "v2c" }
SNMP version.

sendingTrapSetting: sendingTrapSettingOfSnmpSetting: object (on page 683)

requestAuthenticationSetting: requestAuthenticationSettingOfSnmpSetting: object (on page 674)

systemGroupInformation: systemGroupInformationOfSnmpSetting: object (on page 717)

snmpv2cSettingOfEditRequestAuthenticationSetting: *object*

DESCRIPTION

A parameter for editing the settings of whether to permit SNMPv2c requests.

PROPERTIES

community: *string* (up to 180 chars) , must match /^[A-Za-z0-9!#\$'\(\)\+\-\.@_`\{\}~]$|^[A-Za-z0-9!#\$'\(\)\+\-\.@_`\{\}~][A-Za-z0-9!#\$'\(\)\+\-\.@_`\{\}~ ]{0,178}[A-Za-z0-9!#\$'\(\)\+\-\.@_`\{\}~]$/ required
Name of the community which accepts requests.

requestsPermitted: *string[]* (1 item) required
P address (IPv4) or host name of the SNMP manager which accepts requests. For an empty array, all SNMP managers accept requests.

ITEMS
string **(up to 253 chars)** , must match /^([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9])\.)*([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9]))$/

snmpv2cSettingOfEditSendingTrapSetting: *object*

DESCRIPTION

A parameter for editing SNMPv2c trap transmission destination settings.

PROPERTIES

community: ***string*** **(up to 180 chars) , must match /^[A-Za-z0-9!#\$'\(\)\+\-\.@_`\{\}~]$|^[A-Za-z0-9!#\$'\(\)\+\-\.@_`\{\}~][A-Za-z0-9!#\$'\(\)\+\-\.@_`\{\}~ ]{0,178}[A-Za-z0-9!#\$'\(\)\+\-\.@_`\{\}~]$/required**

Community name to be used when reporting SNMP traps.

sendTrapTo: ***string[]*** **(1 item) required**

IP address (IPv4) or host name of the SNMP trap transmission destination.

ITEMS

string **(up to 253 chars)**, must match /^([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9])\.)*([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9]))$/

snmpv2cSettingOfRequestAuthenticationSetting: *object*

DESCRIPTION

Setting information about whether to permit SNMPv2c requests.

PROPERTIES

community: ***string*** **(up to 180 chars) ,must match /^$|^[A-Za-z0-9!#\$'\(\)\+\-\.@_`\{\}~]$|^[A-Za-z0-9!#\$'\(\)\+\-\.@_`\{\}~][A-Za-z0-9!#\$'\(\)\+\-\.@_`\{\}~ ]{0,178}[A-Za-z0-9!#\$'\(\)\+\-\.@_`\{\}~]$/**

Name of the community which accepts requests.

requestsPermitted: ***string[]*** **(1 item)**

IP address (IPv4) or host name of the SNMP manager which accepts requests. For an empty array, all SNMP managers accept requests.

ITEMS

string **(up to 253 chars)** , must match /^$|^([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9])\.)*([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9]))$/

snmpv2cSettingOfSendingTrapSetting: *object*

DESCRIPTION

Setting information about the SNMPv2c trap transmission destination.

PROPERTIES

community: *string* (up to 180 chars) , must match /^$|^[A-Za-z0-9!#\$'\(\)\+\-\.@_`\{\}~]$|^[A-Za-z0-9!#\$'\(\)\+\-\.@_`\{\}~][A-Za-z0-9!#\$'\(\)\+\-\.@_`\{\}~ ]{0,178}[A-Za-z0-9!#\$'\(\)\+\-\.@_`\{\}~]$/

Community name to be used when reporting SNMP traps.

sendTrapTo: *string[]* 1 item

IP address (IPv4) or host name of the SNMP trap transmission destination.

ITEMS

string **(up to 253 chars)** , must match /^$|^([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9])\.)*([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9]))$/

softwareUpdateFile: *object*

DESCRIPTION

The information about the update file of the storage software transferred to the storage cluster.

PROPERTIES

version: *string* (7 to 11 chars) , must match /^[0-9]{1,2}\.[0-9]{1,2}\.[0-9]{1,2}\.[0-9]{1,2}$/

The version of the update file of the storage software.

spareNode: *object*

DESCRIPTION

Spare node information.

PROPERTIES

id: *string* (uuid)

The ID of a spare node.

name: *string*

A spare node name.

faultDomainId: *string* (uuid)

The ID of the fault domain that the nodes belong to.

faultDomainName: *string*

The name of the fault domain that the nodes belong to.

controlPortIpv4Address: *string* (7 to 15 chars) , must match /^(([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])$/

The IP address (IPv4) of the control port.

softwareVersion: *string*

The version of storage software.

biosUuid: *string* (uuid)

The BIOS ID.

modelName: *string*

The model name of the server on which the spare node is operating.

serialNumber: *string*

The serial number of the server on which the spare node is operating.

bmcName: *string* (1 to 253 chars) , must match /^([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9])\.)*([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9]))$/

The host name or IP address (IPv4) of the BMC.

bmcUser: *string* (1 to 512 chars) , must match /^[a-zA-Z0-9!"#\$%&'\(\)*\+,\-\.\/:;<=>\?@\[\\\]\^_`\{\|\}~]{1,512}$/

The user name for BMC connection.

spareNodeList: *object*

DESCRIPTION

A list of spare node information.

PROPERTIES

data: *object[]*

ITEMS

spareNode: object (on page 695)

storage: *object*

DESCRIPTION

Storage cluster information.

PROPERTIES

storageDeviceId: *string* (12 chars) , must match /^[0-9]{12}$/

ID to identify a storage system type.

id: *string* (uuid)
The UUID of the storage cluster.

modelName: *string* (3 to 23 chars)
Product model name.

internalId: *string* (6 chars) , must match /^[0-9]{6}$/
The ID used in the storage cluster.

nickname: *string* (1 to 180 chars)
The nickname of the storage cluster.

numberOfTotalVolumes: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 32768 }
The number of volumes already created.

Any volume whose "volumeType" is "ExternalMigrationOrigin" is not included.

numberOfTotalServers: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 1024 }
The number of registered compute nodes.

numberOfTotalStorageNodes: *integer* (int32) , { x ∈ ℤ | 1 ≤ x ≤ 64 }
The total number of storage nodes contained in the storage cluster.

numberOfReadyStorageNodes: *integer* (int32) , { x ∈ ℤ | 1 ≤ x ≤ 64 }
The number of storage nodes operating normally in the storage cluster.

numberOfFaultDomains: *integer* (int32) , { x ∈ ℤ | 1 ≤ x ≤ 64 }
The total number of fault domains that comprise the storage cluster.

totalPoolRawCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 9223372036854775807 }
The total effective physical capacity of all storage pools contained in the storage cluster (unit: MiB).

totalPoolPhysicalCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 9223372036854775807 }
The total physical capacity of all storage pools contained in the storage cluster (unit: MiB).

totalPoolCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 281474976710655 }
The total logical capacity of all storage pools contained in the storage cluster (unit: MiB).

usedPoolCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 281474976710655 } nullable
The total used capacity of all storage pools contained in the storage cluster (unit: MiB).

freePoolCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 281474976710655 } nullable
The total available capacity of all storage pools contained in the storage cluster (unit: MiB).

savingEffects: savingEffectOfStorage: object (on page 678)

softwareVersion: *string* (7 to 11 chars) , must match /^[0-9]{1,2}\.[0-9]{1,2}\.[0-9]{1,2}\.[0-9]{1,2}$/
The version of storage software.

statusSummary: *string* , x ∈ { "Normal" , "Warning" , "Error" }

Summary of the storage cluster status.

- Normal: No action by the user is required.
- Warning: Although immediate action by the user is not required, some action may have to be taken.
- Error: Immediate action by the user is required.

status: *string* , x ∈ { "Installing" , "Starting" , "Ready" , "NondisruptiveUpdating" , "InstallationFailed" , "StartFailed" , "Stopping" , "Stopped", "Blockage" , "BlockageDueToFailuresExceedingRedundancyLimit" }

The management status of the storage cluster. The following status is output:

- Installing: Initial installation is in progress.
- Starting: Being started.
- Ready: Normal status. The storage cluster can be stopped, or storage software can be upgraded.
- NondisruptiveUpdating: Storage software is being updated without stopping I/Os.
- InstallationFailed: Installation was unsuccessful.
- StartFailed: Startup was unsuccessful.
- Stopping: Planned shutdown is in progress.
- Stopped: Planned shutdown.
- Blockage: Blocked status.
- BlockageDueToFailuresExceedingRedundancyLimit: Blocked status because multiple failures exceeding redundancy occurred.

systemRequirementsFileVersion: *integer* (int32) , { x ∈ ℤ | -1 ≤ x ≤ 99991231 }

Version of the system requirements file registered in the storage cluster.

-1 is returned if an invalid system requirements file is registered.

serviceId: *string* (13 chars) , must match /^[A-Z]{4}[0-9]{9}$/ nullable

Service ID of the storage cluster.

The value set by service personnel or maintenance personnel is displayed. When no value is set, null is output.

storageAutoRecoverySetting: *object*

DESCRIPTION

The setting of whether to enable or disable the auto-recovery function and the status of the function.

PROPERTIES

isEnabled: *boolean*
Enables or disables the auto-recovery function.

storageNodePersistentBlockingThresholdTime: *integer* (int32)
The time to determine if the temporarily blocked storage node that was recovered previously is changed to persistent blockage status (unit: hours).

If a temporarily blocked storage node was recovered previously at least once in the period specified for this parameter, the storage node is changed to persistent blockage status. When the value is 0, the storage node is not changed to persistent blockage status.

status: *string* , x ∈ { "Disabled" , "Normal" , "Conflict" , "Error" }
The status of the auto-recovery function.

- Disabled: The auto-recovery function is disabled.
- Normal: Status in which the auto-recovery function is operating normally. Included cases are those where no storage node is blocked temporarily or the maintenance recovery job by auto recovery is being performed.
- Conflict: The auto-recovery function is temporarily unavailable due to conflict with another task that is in progress.
- Error: The auto-recovery function is unavailable.

storageController: *object*

DESCRIPTION

Storage controller information.

PROPERTIES

id: *string* (uuid)
Storage controller ID.

allocatableCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 6871947674 }
The maximum logical capacity that can be managed on this storage controller (unit: MiB).

currentlyAllocatableCapacity: *integer* (int64)

The logical capacity that is managed on this storage controller (unit: MiB).

usedCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 6871947674 } nullable

The used capacity of the storage pool managed by this storage controller (unit: MiB).

logicalLimit: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 6871947674 }

The maximum logical capacity of volumes that can be created on this storage controller (unit: MiB). The capacity includes the area for control information of the volume.

volumeMaximumCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 6871947674 }

The maximum capacity of the single volume that can be created on this storage controller (unit: MiB). The capacity does not include the area for control information of the volume.

This is the maximum value that can be specified in "capacity" for the volume-creating API when creating a single volume. When creating multiple volumes, specify a value that includes the areas for control information of all the volumes and is smaller than "logicalLimit" for "capacity".

freeCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 6871947674 } nullable

The available capacity of the storage pool managed by this storage controller (unit: MiB).

totalVolumeCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 9223372036854775807 }

The total capacity of volumes that have been created on this storage controller (unit: MiB).

provisionedVolumeCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 9223372036854775807 }

The total capacity of provisioned volumes that have been created on this storage controller (unit: MiB).

otherVolumeCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 9223372036854775807 }

The total capacity of other volumes that have been created on this storage controller (unit: MiB).

temporaryVolumeCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 9223372036854775807 }

The total capacity of temporary volumes that have been created on this storage controller (unit: MiB).

status: *string* , x ∈ { "Normal" , "OneNodeDown" , "RecoveringOneNode" , "Blockage" , "Switching" , "TwoNodesDown" , "BlockageDueToNoActiveNodes" }

The status of the storage controller. The following status is output:

- Normal: Operating normally.
- OneNodeDown: One of the storage controllers went down.
- RecoveringOneNode: One of the storage controllers is recovering.
- Blockage: Blocked.

- Switching: The active and standby storage controllers are switching.
- TwoNodesDown: Two of the storage controller nodes went down.
- BlockageDueToNoActiveNodes: The storage nodes represented in standbyStorageNodeId and secondaryStandbyStorageNodeId could not be failed over and are blocked.

activeStorageNodeId: *string* (uuid)

The ID of the storage node in which the active storage controller is running.

standbyStorageNodeId: *string* (uuid)

The ID of the storage node in which the standby storage controller is running.

secondaryStandbyStorageNodeId: *string* (uuid) nullable

The ID of the storage node in which the second standby storage controller is running.

If storageControllerClusteringPolicy of the protection domain in which the storage controller runs is set to OneRedundantStorageNode, null is output.

isDetailedLoggingMode: *boolean*

Enables or disables the detailed logging mode. If "true" is set, the detailed logging mode is enabled.

allocatableCapacityUsageRate: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 100 }

Usage rate to the maximum logical capacity that can be managed on the storage controller (unit: %).

currentlyAllocatableCapacityUsageRate: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 100 }

Usage rate to the logical capacity managed on the storage controller (unit: %).

capacityStatus: *string* , x ∈ { "Normal" , "Warning" , "Error" }

Capacity status managed by the storage controller.

- Normal: Normal.
- Warning: Capacity usage rate of the storage controller is high.
- Error: Capacity usage rate of the storage controller is high, which might lead to capacity depletion.

dataRebalanceStatus: *string* , x ∈ { "Stopped" , "Running" , "Waiting" }

Status of balancing processing of user data capacity managed by each storage controller.

- Stopped: Status in which the above processing is not being performed.
- Running: Status in which the above processing is being performed.
- Waiting: Status in which the above processing is waiting to be performed.

dataRebalanceProgressRate: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 99 }

Progress rate of balancing processing of user data capacity managed by each storage controller (unit: %).

If dataRebalanceStatus is "Stopped" or "Waiting", null is output.

capacitiesExcludingSystemData: capacitiesExcludingSystemDataOfStorageController: object (on page 584)

storageMasterNodePrimaryFlag: *object*

DESCRIPTION

Indicates whether the storage node is the cluster master node (primary).

PROPERTIES

isStorageMasterNodePrimary: *boolean*

Indicates whether the storage node that received the target API is the cluster master node (primary). If another node receives the API request, HTTP status code 404 (Not Found) is returned. Therefore, this attribute is always true.

storageNetworkSetting: *object*

DESCRIPTION

Network settings of the storage cluster.

PROPERTIES

primaryDnsServerIpAddress: *string*

The IP address of the DNS server for the first name resolution request destination.

secondaryDnsServerIpAddress: *string*

The IP address of the DNS server for the second name resolution request destination.

virtualIpv4Address: *string*

The representative IP address (IPv4) of the storage cluster. If a failure occurs on the storage node to which this IP address is assigned, the IP address is inherited by another storage node.

If the subnet of the control port on all storage nodes is not common, this property cannot be set. In this case, an empty string "" is stored for virtualIpv4Address.

storageNode: *object*

DESCRIPTION

Storage node information.

PROPERTIES

id: *string* (uuid)
Storage node ID.

biosUuid: *string* (uuid)
The storage node UUID, which is registered in the SMBIOS.

protectionDomainId: *string* (uuid)
The ID of the protection domain to which the volume is belonging.

faultDomainId: *string* (uuid)
The ID of a fault domain to which the volume belongs.

faultDomainName: *string*
Name of a fault domain to which the volume belongs.

name: *string* (1 to 64 chars)
Storage node name.

clusterRole: *string* , x ∈ { "Master" , "Worker" }
The role of the storage node in the storage cluster.

storageNodeAttributes: *string[]* (up to 1 items)
Storage node attribute. An empty array ([]) means a storage node which has no attribute.

- Initiator: Initiator node.

ITEMS

string , x ∈ { "Initiator" }

statusSummary: *string* , x ∈ { "Normal", "Warning", "Error" }
The summary of the storage node status.

- Normal: No action by the user is required.
- Warning: Although immediate action by the user is not required, some action may have to be taken.
- Error: Immediate action by the user is required.

status: *string* , x ∈ { "Installing" , "Starting" , "Ready" , "NondisruptiveUpdating" , "TemporaryBlockageProcessInProgress" , "TemporaryBlockage" , "PersistentBlockage" , "TemporaryBlockageFailed" , "MaintenanceBlockageProcessInProgress" , "MaintenanceBlockage" ,

"MaintenanceBlockageFailed" , "Recovering" , "InstallationFailed" , "Removing" , "BlockageAndRemoving" , "Stopping" , "Stopped" , "RemovalFailedAndStarting" , "RemovalFailed" , "RemovalFailedAndNondisruptiveUpdating" , "RemovalFailedAndTemporaryBlockageProcessInProgress" , "RemovalFailedAndTemporaryBlockage" , "RemovalFailedAndPersistentBlockage" , "RemovalFailedAndMaintenanceBlockageProcessInProgress" , "RemovalFailedAndMaintenanceBlockage" , "RemovalFailedAndRecovering" , "RemovalFailedAndStopping" , "RemovalFailedAndStopped" , "RemovalFailedAndMultipleFailures" , "Unknown" }

The status of the storage node. The following status is output:

- Installing: Being set up.
- Starting: Being started.
- Ready: Running normally.
- NondisruptiveUpdating: The storage software on this storage node is being updated without stopping I/Os.
- TemporaryBlockageProcessInProgress: In the process of being blocked temporarily.
- TemporaryBlockage: Blocked temporarily. If the auto-recovery function is enabled, the storage node is subject to auto recovery.
- PersistentBlockage: Blocked(including the case of a recovery failure). The storage node is not subject to auto recovery.
- TemporaryBlockageFailed: Could not be blocked temporarily.
- MaintenanceBlockageProcessInProgress: In the process of being blocked for maintenance.
- MaintenanceBlockage: Blocked for maintenance.
- MaintenanceBlockageFailed: Could not be blocked for maintenance (including the case of a recovery failure).
- Recovering: Being restored.
- InstallationFailed: Could not be set up.
- Removing: Being removed.
- BlockageAndRemoving: Being removed and ready to be disconnected.
- Stopping: Being stopped.
- Stopped: Stopped.

- RemovalFailedAndStarting: Removal did not succeed and starting.
- RemovalFailed: Removal did not succeed and running.
- RemovalFailedAndNondisruptiveUpdating: Removal did not succeed and updating the storage software in the storage node without disrupting I/O.
- RemovalFailedAndTemporaryBlockageProcessInProgress: Removal did not succeed and in the process of being blocked temporarily.
- RemovalFailedAndTemporaryBlockage: Removal did not succeed and blocked temporarily. The storage node is not subject to auto recovery.
- RemovalFailedAndPersistentBlockage: Removal did not succeed and blocked. The storage node is not subject to auto recovery.
- RemovalFailedAndMaintenanceBlockageProcessInProgress: Removal did not succeed and in the process of being blocked for maintenance.
- RemovalFailedAndMaintenanceBlockage: Removal did not succeed and blocked for maintenance.
- RemovalFailedAndRecovering: Removal did not succeed and recovering from the storage node blocked state.
- RemovalFailedAndStopping: Removal did not succeed and stopping.
- RemovalFailedAndStopped: Removal did not succeed and stopped.
- RemovalFailedAndMultipleFailures: Removal did not succeed and multiple failures occurred.
- Unknown: Unknown

driveDataRelocationStatus: *string* , x ∈ { "Stopped" , "Running" }
Status of drive data relocation.

- Stopped: Status in which drive data relocation is not performed.
- Running: Status in which drive data relocation is being performed.

controlPortIpv4Address: *string* , must match /^(([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])$/
The IP address (IPv4) of the control port.

internodePortIpv4Address: *string* , must match /^(([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])$/
The IP address (IPv4) of the internode port.

softwareVersion: *string* (7 to 11 chars) , must match /^[0-9]{1,2}\.[0-9]{1,2}\.[0-9]{1,2}\.[0-9]{1,2}$/
The version of storage software.

modelName: ***string*** **(0 to 128 chars)**

Model name of the server on which the storage node is running.

serialNumber: ***string*** **(0 to 128 chars)**

Serial number of the server on which the storage node is running.

memory: ***integer*** **(int64) , { x ∈ ℤ | 0 ≤ x ≤ 268435456 }**

Amount of storage node memory (unit: MiB).

(Virtual machine) Memory capacity of VM

(Bare metal) Memory capacity available in the physical server

insufficientResourcesForRebuildCapacity: ***object***

PROPERTIES

capacityOfDrive: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 256000 } nullable**

Lacking drive capacity of rebuild capacity (unit: GB, 1 GB = 1,000,000,000 byte).

numberOfDrives: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 23 } nullable**

The number of lacking drives of rebuild capacity.

rebuildableResources: **rebuildableResourcesOfStorageNode: object (on page 671)**

Resource for which Rebuild is possible.

storageNodeBmcAccessSetting: *object*

DESCRIPTION

The BMC connection information about storage nodes.

PROPERTIES

id: ***string*** **(uuid)**

Storage node ID.

bmcName: ***string*** **(up to 253 chars) , must match /^$|^([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9])\.)*([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9]))$/**

The host name or IP address (IPv4) of the BMC.

An empty string "" is output if nothing is set.

bmcUser: ***string*** **(up to 512 chars) , must match /^$|^[a-zA-Z0-9!"#\$%&'\(\)*\+,\-\.\/:;<=>\?@\[\\\]\^_`\{\|\}~]{1,512}$/**

The user name for BMC connection.

An empty string "" is output if nothing is set.

storageNodeBmcAccessSettingList: *object*

DESCRIPTION

A list of BMC connection information about storage nodes.

PROPERTIES

data: ***object[]***

ITEMS

storageNodeBmcAccessSetting: object (on page 706)

storageNodeCapacitySetting: *object*

DESCRIPTION

Capacity management settings of a storage node.

PROPERTIES

id: ***string*** **(uuid)**

Storage node ID.

capacityBalancingSetting: capacityBalancingSetting: object (on page 585)

storageNodeNetworkSetting: *object*

DESCRIPTION

Storage node network settings.

PROPERTIES

id: ***string*** **(uuid)**

Storage node ID.

ipv4Route: ***object[]***

ITEMS

ipv4RouteOfStorageNodeNetworkSetting: object (on page 623)

storageNodePerformance: *object*

DESCRIPTION

Performance information of the storage node (monitor information). Null is set if the information could not be obtained. Infinite value, Infinity is not output as the double type value.

PROPERTIES

id: *string* (uuid)
Storage node ID.

volumeReadIOPS: *integer* (int32) nullable
The number of read I/Os processed for a volume (unit: IOPS).

volumeWriteIOPS: *integer* (int32) nullable
The number of write I/Os processed for a volume (unit: IOPS).

volumeReadTransferRate: *number* (double) nullable
Read data transfer amount in a volume per second (unit: MiB/sec).

volumeWriteTransferRate: *number* (double) nullable
Write data transfer amount in a volume per second (unit: MiB/sec).

driveReadIOPS: *integer* (int32) nullable
The number of read I/Os processed by a drive (unit: IOPS).

driveWriteIOPS: *integer* (int32) nullable
The number of write I/Os processed by a drive (unit: IOPS).

driveReadTransferRate: *number* (double) nullable
Read data transfer amount per second (unit: MiB/sec).

driveWriteTransferRate: *number* (double) nullable
Write data transfer amount per second (unit: MiB/sec).

cpu: *object[]*
A list of CPU performance information of the storage node. If no information could be collected about the usage of a specific CPU core, an array without information about the CPU core is returned. If no information could be collected about the usage of all CPU cores, an empty array ([]) is returned.

ITEMS

cpuPerformance: object (on page 591)

memory: memoryPerformance: object (on page 639) nullable

storageNodePerformanceListResponse: *object*

DESCRIPTION

A list of storage node performance information (monitor information) at the specified time.

PROPERTIES

data: ***object[]***

ITEMS

storageNodePerformanceListResponseData: object (on page 709)

storageNodePerformanceListResponseData: *object*

DESCRIPTION

The performance information (monitor information) of the storage node when the information was collected. For performanceObjects, storage nodes of which performance information could be collected when the API was executed are output as an array.

PROPERTIES

timestamp: ***string*** **(date-time)**

The time when this information was collected.

performanceObjects: ***object[]***

A list of storage node performance information (monitor information) collected at the time specified for timestamp.

ITEMS

storageNodePerformance: object (on page 708)

storagePerformance: *object*

DESCRIPTION

Performance information (monitor information) for each storage cluster. Infinite value, Infinity is not output as a double type value.

PROPERTIES

id: ***string*** **(uuid)**

The UUID of the storage cluster.

averageCpuUsage: *number* (double)
Average CPU usage for all storage nodes (unit: %).

averageMemoryUsage: *number* (double)
Average memory usage for all storage nodes (unit: %).

storagePerformanceListResponseData: *object*

DESCRIPTION

The performance information (monitor information) of the storage cluster when the information was collected. For performanceObjects, storage clusters of which performance information could be collected when the API was executed are output as an array.

PROPERTIES

timestamp: *string* (date-time)
The time when this information was collected.

performanceObjects: *object[]*
A list of storage cluster performance information (monitor information) collected at the time specified for timestamp.

ITEMS
storagePerformance: object (on page 709)

storageTimeSetting: *object*

DESCRIPTION

Time settings of the storage cluster.

PROPERTIES

systemTime: *string* (date-time)
The UTC time of the storage cluster.

ntpServerNames: *string[]*
A list of NTP servers. NTP servers are stored in the order of priority.

ITEMS
string

DESCRIPTION
Host name or IP address (IPv4) of the NTP server.

timezone: *string* , x ∈ { "Africa/Abidjan" , "Africa/Accra" , "Africa/Addis_Ababa" , "Africa/Algiers" , "Africa/Asmara" , "Africa/Bamako" , "Africa/Bangui" , "Africa/Banjul" , "Africa/Bissau" , "Africa/Blantyre" , "Africa/Brazzaville" , "Africa/Bujumbura" , "Africa/Cairo" , "Africa/Casablanca" , "Africa/Ceuta" , "Africa/Conakry" , "Africa/Dakar" , "Africa/Dar_es_Salaam" , "Africa/Djibouti" , "Africa/Douala" , "Africa/El_Aaiun" , "Africa/Freetown" , "Africa/Gaborone" , "Africa/Harare" , "Africa/Johannesburg" , "Africa/Juba" , "Africa/Kampala" , "Africa/Khartoum" , "Africa/Kigali" , "Africa/Kinshasa" , "Africa/Lagos" , "Africa/Libreville" , "Africa/Lome" , "Africa/Luanda" , "Africa/Lubumbashi" , "Africa/Lusaka" , "Africa/Malabo" , "Africa/Maputo" , "Africa/Maseru" , "Africa/Mbabane" , "Africa/Mogadishu" , "Africa/Monrovia" , "Africa/Nairobi" , "Africa/Ndjamena" , "Africa/Niamey" , "Africa/Nouakchott" , "Africa/Ouagadougou" , "Africa/Porto-Novo" , "Africa/Sao_Tome" , "Africa/Tripoli" , "Africa/Tunis" , "Africa/Windhoek" , "America/Adak" , "America/Anchorage" , "America/Anguilla" , "America/Antigua" , "America/Araguaina" , "America/Argentina/Buenos_Aires" , "America/Argentina/Catamarca" , "America/Argentina/Cordoba" , "America/Argentina/Jujuy" , "America/Argentina/La_Rioja" , "America/Argentina/Mendoza" , "America/Argentina/Rio_Gallegos" , "America/Argentina/Salta" , "America/Argentina/San_Juan" , "America/Argentina/San_Luis" , "America/Argentina/Tucuman" , "America/Argentina/Ushuaia" , "America/Aruba" , "America/Asuncion" , "America/Atikokan" , "America/Bahia" , "America/Bahia_Banderas" , "America/Barbados" , "America/Belem" , "America/Belize" , "America/Blanc-Sablon" , "America/Boa_Vista" , "America/Bogota" , "America/Boise" , "America/Cambridge_Bay" , "America/Campo_Grande" , "America/Cancun" , "America/Caracas" , "America/Cayenne" , "America/Cayman" , "America/Chicago" , "America/Chihuahua" , "America/Costa_Rica" , "America/Creston" , "America/Cuiaba" , "America/Curacao" , "America/Danmarkshavn" , "America/Dawson" , "America/Dawson_Creek" , "America/Denver" , "America/Detroit" , "America/Dominica" , "America/Edmonton" , "America/Eirunepe" , "America/El_Salvador" , "America/Fort_Nelson" , "America/Fortaleza" , "America/Glace_Bay" , "America/Godthab" , "America/Goose_Bay" , "America/Grand_Turk" , "America/Grenada" , "America/Guadeloupe" , "America/Guatemala" , "America/Guayaquil" , "America/Guyana" , "America/Halifax" , "America/Havana" , "America/Hermosillo" , "America/Indiana/Indianapolis" , "America/Indiana/Knox" , "America/Indiana/Marengo" , "America/Indiana/Petersburg" , "America/Indiana/Tell_City" , "America/Indiana/Vevay" , "America/Indiana/Vincennes" , "America/Indiana/Winamac" , "America/Inuvik" , "America/Iqaluit" , "America/Jamaica" , "America/Juneau" , "America/Kentucky/Louisville" , "America/Kentucky/Monticello" , "America/Kralendijk" , "America/La_Paz" , "America/Lima" , "America/Los_Angeles" , "America/Lower_Princes" , "America/Maceio" , "America/Managua" , "America/Manaus" , "America/Marigot" , "America/Martinique" , "America/Matamoros" , "America/Mazatlan" , "America/Menominee" , "America/Merida" , "America/Metlakatla" , "America/Mexico_City" , "America/Miquelon" , "America/Moncton" , "America/Monterrey" , "America/Montevideo" , "America/Montserrat" , "America/Nassau" , "America/New_York" , "America/Nipigon" , "America/Nome" , "America/Noronha" , "America/North_Dakota/Beulah" , "America/North_Dakota/Center" , "America/North_Dakota/New_Salem" , "America/Ojinaga" , "America/Panama" , "America/Pangnirtung" , "America/Paramaribo" , "America/Phoenix" , "America/Port-au-Prince" , "America/Port_of_Spain" , "America/Porto_Velho" , "America/Puerto_Rico" , "America/Punta_Arenas" , "America/Rainy_River" , "America/Rankin_Inlet" , "America/Recife" , "America/Regina" , "America/Resolute" , "America/Rio_Branco" , "America/Santarem" , "America/Santiago" , "America/Santo_Domingo" , "America/Sao_Paulo" ,

"America/Scoresbysund" , "America/Sitka" , "America/St_Barthelemy" , "America/St_Johns" , "America/St_Kitts" , "America/St_Lucia" , "America/St_Thomas" , "America/St_Vincent" , "America/Swift_Current" , "America/Tegucigalpa" , "America/Thule" , "America/Thunder_Bay" , "America/Tijuana" , "America/Toronto" , "America/Tortola" , "America/Vancouver" , "America/Whitehorse" , "America/Winnipeg" , "America/Yakutat" , "America/Yellowknife" , "Antarctica/Casey" , "Antarctica/Davis" , "Antarctica/DumontDUrville" , "Antarctica/Macquarie" , "Antarctica/Mawson" , "Antarctica/McMurdo" , "Antarctica/Palmer" , "Antarctica/Rothera" , "Antarctica/Syowa" , "Antarctica/Troll" , "Antarctica/Vostok" , "Arctic/Longyearbyen" , "Asia/Aden" , "Asia/Almaty" , "Asia/Amman" , "Asia/Anadyr" , "Asia/Aqtau" , "Asia/Aqtobe" , "Asia/Ashgabat" , "Asia/Atyrau" , "Asia/Baghdad" , "Asia/Bahrain" , "Asia/Baku" , "Asia/Bangkok" , "Asia/Barnaul" , "Asia/Beijing" , "Asia/Beirut" , "Asia/Bishkek" , "Asia/Brunei" , "Asia/Chita" , "Asia/Choibalsan" , "Asia/Colombo" , "Asia/Damascus" , "Asia/Dhaka" , "Asia/Dili" , "Asia/Dubai" , "Asia/Dushanbe" , "Asia/Famagusta" , "Asia/Gaza" , "Asia/Hebron" , "Asia/Ho_Chi_Minh" , "Asia/Hong_Kong" , "Asia/Hovd" , "Asia/Irkutsk" , "Asia/Jakarta" , "Asia/Jayapura" , "Asia/Jerusalem" , "Asia/Kabul" , "Asia/Kamchatka" , "Asia/Karachi" , "Asia/Kathmandu" , "Asia/Khandyga" , "Asia/Kolkata" , "Asia/Krasnoyarsk" , "Asia/Kuala_Lumpur" , "Asia/Kuching" , "Asia/Kuwait" , "Asia/Macau" , "Asia/Magadan" , "Asia/Makassar" , "Asia/Manila" , "Asia/Muscat" , "Asia/Nicosia" , "Asia/Novokuznetsk" , "Asia/Novosibirsk" , "Asia/Omsk" , "Asia/Oral" , "Asia/Phnom_Penh" , "Asia/Pontianak" , "Asia/Pyongyang" , "Asia/Qatar" , "Asia/Qyzylorda" , "Asia/Riyadh" , "Asia/Sakhalin" , "Asia/Samarkand" , "Asia/Seoul" , "Asia/Shanghai" , "Asia/Singapore" , "Asia/Srednekolymsk" , "Asia/Taipei" , "Asia/Tashkent" , "Asia/Tbilisi" , "Asia/Tehran" , "Asia/Thimphu" , "Asia/Tokyo" , "Asia/Tomsk" , "Asia/Ulaanbaatar" , "Asia/Urumqi" , "Asia/Ust-Nera" , "Asia/Vientiane" , "Asia/Vladivostok" , "Asia/Yakutsk" , "Asia/Yangon" , "Asia/Yekaterinburg" , "Asia/Yerevan" , "Atlantic/Azores" , "Atlantic/Bermuda" , "Atlantic/Canary" , "Atlantic/Cape_Verde" , "Atlantic/Faroe" , "Atlantic/Madeira" , "Atlantic/Reykjavik" , "Atlantic/South_Georgia" , "Atlantic/St_Helena" , "Atlantic/Stanley" , "Australia/Adelaide" , "Australia/Brisbane" , "Australia/Broken_Hill" , "Australia/Currie" , "Australia/Darwin" , "Australia/Eucla" , "Australia/Hobart" , "Australia/Lindeman" , "Australia/Lord_Howe" , "Australia/Melbourne" , "Australia/Perth" , "Australia/Sydney" , "Europe/Amsterdam" , "Europe/Andorra" , "Europe/Astrakhan" , "Europe/Athens" , "Europe/Belgrade" , "Europe/Berlin" , "Europe/Bratislava" , "Europe/Brussels" , "Europe/Bucharest" , "Europe/Budapest" , "Europe/Busingen" , "Europe/Chisinau" , "Europe/Copenhagen" , "Europe/Dublin" , "Europe/Gibraltar" , "Europe/Guernsey" , "Europe/Helsinki" , "Europe/Isle_of_Man" , "Europe/Istanbul" , "Europe/Jersey" , "Europe/Kaliningrad" , "Europe/Kiev" , "Europe/Kirov" , "Europe/Lisbon" , "Europe/Ljubljana" , "Europe/London" , "Europe/Luxembourg" , "Europe/Madrid" , "Europe/Malta" , "Europe/Mariehamn" , "Europe/Minsk" , "Europe/Monaco" , "Europe/Moscow" , "Europe/Oslo" , "Europe/Paris" , "Europe/Podgorica" , "Europe/Prague" , "Europe/Riga" , "Europe/Rome" , "Europe/Samara" , "Europe/San_Marino" , "Europe/Sarajevo" , "Europe/Saratov" , "Europe/Simferopol" , "Europe/Skopje" , "Europe/Sofia" , "Europe/Stockholm" , "Europe/Tallinn" , "Europe/Tirane" , "Europe/Ulyanovsk" , "Europe/Uzhgorod" , "Europe/Vaduz" , "Europe/Vatican" , "Europe/Vienna" , "Europe/Vilnius" , "Europe/Volgograd" , "Europe/Warsaw" , "Europe/Zagreb" , "Europe/Zaporozhye" , "Europe/Zurich" , "Indian/Antananarivo" , "Indian/Chagos" , "Indian/Christmas" , "Indian/Cocos" , "Indian/Comoro" , "Indian/Kerguelen" , "Indian/Mahe" , "Indian/Maldives" , "Indian/Mauritius" , "Indian/Mayotte" , "Indian/Reunion" , "Pacific/Apia" , "Pacific/Auckland" , "Pacific/Bougainville" , "Pacific/Chatham" , "Pacific/Chuuk" , "Pacific/Easter" , "Pacific/Efate" , "Pacific/Enderbury" , "Pacific/Fakaofo" , "Pacific/

Fiji" , "Pacific/Funafuti" , "Pacific/Galapagos" , "Pacific/Gambier" , "Pacific/Guadalcanal" , "Pacific/Guam" , "Pacific/Honolulu" , "Pacific/Kiritimati" , "Pacific/Kosrae" , "Pacific/Kwajalein" , "Pacific/Majuro" , "Pacific/Marquesas" , "Pacific/Midway" , "Pacific/Nauru" , "Pacific/Niue" , "Pacific/Norfolk" , "Pacific/Noumea" , "Pacific/Pago_Pago" , "Pacific/Palau" , "Pacific/Pitcairn" , "Pacific/Pohnpei" , "Pacific/Port_Moresby" , "Pacific/Rarotonga" , "Pacific/Saipan" , "Pacific/Tahiti" , "Pacific/Tarawa" , "Pacific/Tongatapu" , "Pacific/Wake" , "Pacific/Wallis" , "UTC" }

The timezone of the storage cluster.

syslogForwardingSettingOfAuditLogSetting: *object*

DESCRIPTION

Syslog transfer settings of audit logs.

PROPERTIES

locationName: ***string*** **(up to 32 chars) , must match /^$|^[A-Za-z0-9!"#\$%&'\(\)*\+\-\.\/:;<=>\?@\[\\\]\^_`\{\|\}~]{1,32}$/**

Location information.

syslogServers: ***object[]*** **(1 to 2 items)**

Syslog server to which audit logs are transferred. You can set up to two servers.

ITEMS

syslogServerSettingOfAuditLogSetting: object (on page 715)

syslogForwardingSettingOfEditAuditLogSetting: *object*

DESCRIPTION

A parameter for editing the Syslog transfer settings of audit logs.

PROPERTIES

locationName: ***string*** **(1 to 32 chars) , must match /^[A-Za-z0-9!"#\$%&'\(\)*\+\-\.\/:;<=>\?@\[\\\]\^_`\{\|\}~]{1,32}$/**

Location information for Syslog transfer.

syslogServers: ***object[]*** **(1 to 2 items) required**

Syslog server to which audit logs are transferred. You can set up to two servers. If you specified three or more servers, HTTP status code 400 (Bad Request) is returned.

ITEMS

syslogServerSettingOfEditAuditLogSetting: object (on page 714)

syslogServerSettingOfEditAuditLogSetting: *object*

DESCRIPTION

A parameter for editing the settings of the Syslog server to which audit logs are transferred.

HTTP status code 400 (Bad Request) is returned if the index was specified with multiple elements of syslogServers.

PROPERTIES

index: ***integer*****(int32), { x ∈ ℤ | 1 ≤ x ≤ 2 } required**
The ID of the Syslog server.

isEnabled: ***boolean*** **required**
Whether audit logs are transferred to the Syslog server specified in serverName.

serverName: ***string*** **(1 to 253 chars) , must match /^([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9])\.)*([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9]))$/ required**
Host name or IP address (IPv4) of the Syslog server.

port: ***integer*** **(int32) , { x ∈ ℤ | 1 ≤ x ≤ 65535 } required**
Port number of the Syslog server.

transportProtocol: ***string*** **, x ∈ { "UDP" } required**
Communications protocol.

syslogForwardingSettingOfEditEventLogSetting: *object*

DESCRIPTION

A parameter for editing the Syslog transfer settings of event logs.

PROPERTIES

locationName: ***string*** **(1 to 32 chars) , must match /^[A-Za-z0-9!"#\$%&'\(\)*\+\-\.\/:;<=>\?@\[\\\]\^_`\{\|\}~]{1,32}$/**
Location information for Syslog transfer.

syslogServers: *object[]* **(1 to 2 items) required**

Syslog server to which event logs are transferred. You can set up to two servers. If you specified three or more servers, HTTP status code 400 (Bad Request) is returned.

ITEMS

syslogServerSettingOfEditEventLogSetting: object (on page 715)

syslogForwardingSettingOfEventLogSetting: *object*

DESCRIPTION

Syslog transfer settings of event logs.

PROPERTIES

locationName: *string* (up to 32 chars) , must match /^$|^[A-Za-z0-9!"#\$%&'\(\)*\+\-\.\/:;<=>\?@\[\\\]\^_`\{\|\}~]{1,32}$/
Location information.

syslogServers: *object[]* (1 to 2 items)
Syslog server to which event logs are transferred. You can set up to two servers.

ITEMS
syslogServerSettingOfEventLogSetting: object (on page 716)

syslogServerSettingOfAuditLogSetting: *object*

DESCRIPTION

Settings of the Syslog server to which audit logs are transferred.

PROPERTIES

index: *integer* (int32)
The ID of the Syslog server.

isEnabled: *boolean*
Whether audit logs are transferred to the Syslog server specified in serverName.

serverName: *string* (up to 253 chars) , must match /^$|^([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9])\.)*([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9]))$/
Host name or IP address (IPv4) of the Syslog server.

port: *integer* (int32) , { x ∈ ℤ | 1 ≤ x ≤ 65535 }
Port number of the Syslog server.

transportProtocol: *string* , x ∈ { "UDP" }
Communications protocol.

syslogServerSettingOfEditEventLogSetting: *object*

DESCRIPTION

A parameter for editing the settings of the Syslog server to which event logs are transferred.

HTTP status code 400 (Bad Request) is returned if the index was specified with multiple elements of syslogServers.

PROPERTIES

index: ***integer*** **(int32) , { x ∈ ℤ | 1 ≤ x ≤ 2 } required**
The ID of the Syslog server.

isEnabled: ***boolean*** **required**
Whether event logs are transferred to the Syslog server specified in serverName.

serverName: ***string*** **(1 to 253 chars) , must match /^([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9])\.)*([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9]))$/ required**
Host name or IP address (IPv4) of the Syslog server.

port: ***integer*** **(int32) , { x ∈ ℤ | 1 ≤ x ≤ 65535 } required**
Port number of the Syslog server.

transportProtocol: ***string*** **, x ∈ { "UDP" } required**
Communications protocol.

syslogServerSettingOfEventLogSetting: *object*

DESCRIPTION

Settings of the Syslog server to which event logs are transferred.

PROPERTIES

index: ***integer*** **(int32)**
The ID of the Syslog server.

isEnabled: ***boolean***
Whether event logs are transferred to the Syslog server specified in serverName.

serverName: ***string*** **(up to 253 chars) , must match /^$|^([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9])\.)*([a-zA-Z0-9](|[a-zA-Z0-9\-]{0,61}[a-zA-Z0-9]))$/**
Host name or IP address (IPv4) of the Syslog server.

port: ***integer*** **(int32) , { x ∈ ℤ | 1 ≤ x ≤ 65535 }**
Port number of the Syslog server.

transportProtocol: ***string*** **, x ∈ { "UDP" }**
Communications protocol.

systemGroupInformationOfEditSnmpSetting: *object*

DESCRIPTION

A parameter for editing system group information.

PROPERTIES

storageSystemName: *string* (up to 180 chars) , must match /^[A-Za-z0-9!#\$'\(\)\+\-\.=@\[\]_`\{\}~]$|^[A-Za-z0-9!#\$'\(\)\+\-\.=@\[\]_`\{\}~][A-Za-z0-9!#\$'\(\)\+\-\.=@\[\]_`\{\}~ ]{0,178}[A-Za-z0-9!#\$'\(\)\+\-\.=@\[\]_`\{\}~]$/ required

System name. Output as "sysName" in the SNMP agent MIB.

contact: *string* (up to 180 chars) , must match /^[A-Za-z0-9!#\$'\(\)\+\-\.=@\[\]_`\{\}~]$|^[A-Za-z0-9!#\$'\(\)\+\-\.=@\[\]_`\{\}~][A-Za-z0-9!#\$'\(\)\+\-\.=@\[\]_`\{\}~ ]{0,178}[A-Za-z0-9!#\$'\(\)\+\-\.=@\[\]_`\{\}~]$/ required

Administrator name or contact. Output as "sysContact" in the SNMP agent MIB.

location: *string* (up to 180 chars) , must match /^[A-Za-z0-9!#\$'\(\)\+\-\.=@\[\]_`\{\}~]$|^[A-Za-z0-9!#\$'\(\)\+\-\.=@\[\]_`\{\}~][A-Za-z0-9!#\$'\(\)\+\-\.=@\[\]_`\{\}~ ]{0,178}[A-Za-z0-9!#\$'\(\)\+\-\.=@\[\]_`\{\}~]$/ required

Installation location. Output as "sysLocation" in the SNMP agent MIB.

systemGroupInformationOfSnmpSetting: *object*

DESCRIPTION

System group information.

PROPERTIES

storageSystemName: *string* (up to 180 chars) , must match /^[A-Za-z0-9!#\$'\(\)\+\-\.=@\[\]_`\{\}~]$|^[A-Za-z0-9!#\$'\(\)\+\-\.=@\[\]_`\{\}~][A-Za-z0-9!#\$'\(\)\+\-\.=@\[\]_`\{\}~ ]{0,178}[A-Za-z0-9!#\$'\(\)\+\-\.=@\[\]_`\{\}~]$/

System name. Output as "sysName" in the SNMP agent MIB. If this is not set, a hyphen (-) will be output.

contact: *string* (up to 180 chars) , must match /^[A-Za-z0-9!#\$'\(\)\+\-\.=@\[\]_`\{\}~]$|^[A-Za-z0-9!#\$'\(\)\+\-\.=@\[\]_`\{\}~][A-Za-z0-9!#\$'\(\)\+\-\.=@\[\]_`\{\}~ ]{0,178}[A-Za-z0-9!#\$'\(\)\+\-\.=@\[\]_`\{\}~]$/

Administrator name or contact. Output as "sysContact" in the SNMP agent MIB. If this is not set, a hyphen (-) will be output.

location: *string* (up to 180 chars) , must match /^[A-Za-z0-9!#\$'\(\)\+\-\.=@\[\]_`\{\}~]$|^[A-Za-z0-9!#\$'\(\)\+\-\.=@\[\]_`\{\}~][A-Za-z0-9!#\$'\(\)\+\-\.=@\[\]_`\{\}~ ]{0,178}[A-Za-z0-9!#\$'\(\)\+\-\.=@\[\]_`\{\}~]$/

Installation location. Output as "sysLocation" in the SNMP agent MIB. If this is not set, a hyphen (-) will be output.

teamingOfControlPort: *object*

DESCRIPTION

Teaming information of the control port. This is output only when teaming is enabled.

PROPERTIES

primaryMacAddress: *string* (17 chars) , must match /^([a-f0-9]{2}:){5}[a-f0-9]{2}$/

The physical MAC address of the teaming information of the primary port.

secondaryMacAddress: *string* (17 chars) , must match /^([a-f0-9]{2}:){5}[a-f0-9]{2}$/

The physical MAC address of the teaming information of the secondary port.

primaryConfiguredPortSpeed: *string* , x ∈ { "Auto" , "1G" , "10G" , "25G" , "40G" , "100G" }

Link speed setting of the primary port (unit: bps). This setting determines the actual link speed.

- Auto: The speed setting depends on the switch or SFP specifications.

secondaryConfiguredPortSpeed: *string* , x ∈ { "Auto" , "1G" , "10G" , "25G" , "40G" , "100G" }

Link speed setting of the secondary port (unit: bps). This setting determines the actual link speed.

- Auto: The speed setting depends on the switch or SFP specifications.

primaryPortSpeedDuplex: *string* , x ∈ { "10Mbps Half" , "10Mbps Full" , "100Mbps Half" , "100Mbps Full" , "1Gbps Half" , "1Gbps Full" , "2.5Gbps Full" , "5Gbps Full" , "10Gbps Full" , "20Gbps Full" , "25Gbps Full" , "40Gbps Full" , "50Gbps Full" , "56Gbps Full" , "100Gbps Full" , "200Gbps Full" , "400Gbps Full" , "Unknown" , "LinkDown" }

Actual link speed and duplex settings of the physical port used for communication through the primary port.

- Unknown: The status is unknown.
- LinkDown: Link down occurred.

secondaryPortSpeedDuplex: *string* , x ∈ { "10Mbps Half" , "10Mbps Full" , "100Mbps Half" , "100Mbps Full" , "1Gbps Half" , "1Gbps Full" , "2.5Gbps Full" , "5Gbps Full" , "10Gbps Full" , "20Gbps Full" , "25Gbps Full" , "40Gbps Full" , "50Gbps Full" , "56Gbps Full" , "100Gbps Full" , "200Gbps Full" , "400Gbps Full" , "Unknown" , "LinkDown" }

Actual link speed and duplex settings of the physical port used for communication through the secondary port.

- Unknown: The status is unknown.
- LinkDown: Link down occurred.

primaryDeviceName: ***string*** **(1 to 4096 chars)**
Device name of the NIC of the primary port.

secondaryDeviceName: ***string*** **(1 to 4096 chars)**
Device name of the NIC of the secondary port.

primaryInterfaceName: ***string***
Interface name of the primary port. This name is unique within a storage node for the compute ports, control ports, and internode ports.

secondaryInterfaceName: ***string***
Interface name of the secondary port. This name is unique within a storage node for the compute ports, control ports, and internode ports.

activePort: ***string*** **, x ∈ { "primary" , "secondary" }**
Information on the teaming active port.

- primary: The primary port is the active port.
- secondary: The secondary port is the active port.

mode: ***string*** **, x ∈ { "active-standby" }**
Teaming mode.

- active-standby: Active-standby configuration

autoFailBackEnabled: ***boolean***
Indicates whether automatic failback for teaming is enabled/disabled.

- true: Automatic failback is enabled.
- false: Automatic failback is disabled.

teamingOfInternodePort: *object*

DESCRIPTION

Teaming information of the internode port. This is output only when teaming is enabled.

PROPERTIES

primaryMacAddress: ***string*** **(17 chars) , must match /^([a-f0-9]{2}:){5}[a-f0-9]{2}$/**
The physical MAC address of the teaming information of the primary port.

secondaryMacAddress: ***string*** **(17 chars) , must match /^([a-f0-9]{2}:){5}[a-f0-9]{2}$/**
The physical MAC address of the teaming information of the secondary port.

primaryConfiguredPortSpeed: ***string*** **, x ∈ { "Auto" , "1G" , "10G" , "25G" , "40G" , "100G" }**

Link speed setting of the primary port (unit: bps). This setting determines the actual link speed.

- Auto: The speed setting depends on the switch or SFP specifications.

secondaryConfiguredPortSpeed: ***string*** **, x ∈ { "Auto" , "1G" , "10G" , "25G" , "40G" , "100G" }**

Link speed setting of the secondary port (unit: bps). This setting determines the actual link speed.

- Auto: The speed setting depends on the switch or SFP specifications.

primaryPortSpeedDuplex: ***string*** **, x ∈ { "10Mbps Half" , "10Mbps Full" , "100Mbps Half" , "100Mbps Full" , "1Gbps Half" , "1Gbps Full" , "2.5Gbps Full" , "5Gbps Full" , "10Gbps Full" , "20Gbps Full" , "25Gbps Full" , "40Gbps Full" , "50Gbps Full" , "56Gbps Full" , "100Gbps Full" , "200Gbps Full" , "400Gbps Full" , "Unknown" , "LinkDown" }**

Actual link speed and duplex settings of the physical port used for communication through the primary port.

- Unknown: The status is unknown.
- LinkDown: Link down occurred.

secondaryPortSpeedDuplex: ***string*** **, x ∈ { "10Mbps Half" , "10Mbps Full" , "100Mbps Half" , "100Mbps Full" , "1Gbps Half" , "1Gbps Full" , "2.5Gbps Full" , "5Gbps Full" , "10Gbps Full" , "20Gbps Full" , "25Gbps Full" , "40Gbps Full" , "50Gbps Full" , "56Gbps Full" , "100Gbps Full" , "200Gbps Full" , "400Gbps Full" , "Unknown" , "LinkDown" }**

Actual link speed and duplex settings of the physical port used for communication through the secondary port.

- Unknown: The status is unknown.
- LinkDown: Link down occurred.

primaryDeviceName: ***string*** **(1 to 4096 chars)**

Device name of the NIC of the primary port.

secondaryDeviceName: ***string*** **(1 to 4096 chars)**

Device name of the NIC of the secondary port.

primaryInterfaceName: ***string***

Interface name of the primary port. This name is unique within a storage node for the compute ports, control ports, and internode ports.

secondaryInterfaceName: ***string***

Interface name of the secondary port. This name is unique within a storage node for the compute ports, control ports, and internode ports.

activePort: *string* , x ∈ { "primary" , "secondary" }

Information on the teaming active port.

- primary: The primary port is the active port.
- secondary: The secondary port is the active port.

mode: *string* , x ∈ { "active-standby" }

Teaming mode.

- active-standby: Active-standby configuration

autoFailBackEnabled: *boolean*

Indicates whether automatic failback for teaming is enabled/disabled.

- true: Automatic failback is enabled.
- false: Automatic failback is disabled.

ticket: *object*

DESCRIPTION

Information of the authentication ticket.

PROPERTIES

ticket: *string* (1 to 12288 chars) , must match /^[\-A-Za-z0-9!\"#\$%&'\(\)*\+,\.\/:;<=>\?@\[\\\]\^_`\{\|\}~]{1,12288}$/

Authentication ticket. Specify applicable information to the Authorization header when using ticket authentication.

expirationTime: *string* (date-time)

Expiration time of the authentication ticket. If the valid period is omitted, the authentication ticket expiration date and time are the same as that for the password of the user who issued the ticket.

However, if the user's password does not have an expiration time, the ticket will be valid for 365 days.

updateSoftwareParam: *object*

DESCRIPTION

A parameter for the update of the storage software.

PROPERTIES

mode: ***string*** **, x ∈ { "Non-disruptive" } required**

Modes for updating the software:

- Non-disruptive: I/Os are not stopped.

user: *object*

DESCRIPTION

User information.

A null is returned for the following values if (1) External authentication is enabled, (2) mappingMode is set to Group, (3) The API for obtaining user information is run with "self" specified for userId, and (4) (3) is run by using the user on the external authentication server:

- userObjectId
- isEnabled
- userGroups

PROPERTIES

userId: ***string*** **(5 to 255 chars) , must match /^[\-A-Za-z0-9!#\$%&'\.@\^_`\{\}~]{5,255}$/**

User ID.

userObjectId: ***string*** **(5 to 765 chars) , must match /^[A-Za-z0-9%\-\._~]{5,765}$/ nullable**

Object ID of the user. For the user ID, a percent-encoded reserved string defined in RFC 3986 is output.

passwordExpirationTime: ***string*** **(date-time) nullable**

Expiration time of the password. After this time, the password expires.

If authentication is external, null is returned.

isEnabled: ***boolean*** **nullable**

Enables or disables the user.

userGroups: ***object[]*** **(1 to 8 items) nullable**

A list of user group IDs to which the user belongs.

ITEMS

object

PROPERTIES

userGroupId: *string* (1 to 64 chars) , must match /^[a-zA-Z0-9!#\$%&'\-\.@\^_`\{\}~]{1,64}$/

User Group ID.

userGroupObjectId: *string* (1 to 192 chars) , must match /^[A-Za-z0-9%\-\._~]{1,192}$/

Object ID of the user group. For the user group ID, a percent-encoded reserved string defined in RFC 3986 is output.

isBuiltIn: *boolean*

Whether the user is a built-in user.

authentication: *string* , x ∈ { "local" , "external" }

Authentication type.

- local: Authenticated locally.
- external: Authenticated by the external authentication server.

roleNames: *string[]*

Role of the user group.

ITEMS

- For system administrators: (One to six items)

  string , x ∈ { "Security", "Storage", "Monitor", "Service", "Audit", "Resource" }

- For VPS administrators: (One to three items)

  string , x ∈ { "VpsSecurity", "VpsStorage", "VpsMonitor" }

isEnabledConsoleLogin: *boolean* nullable

Whether the use of console interface is permitted.

- true: The console interface can be used.
- false: The console interface cannot be used.

(Virtual machine) A null is output.

(Bare metal) The set value is valid.

vpsId: *string* , must match /^\(system\)$|[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

The ID of the virtual private storage (VPS) that the user belongs to.

If the user is a system administrator, the string "(system)" is displayed.

privileges: *object[]* (1 to 65 items)

A list of information about a virtual private storage (VPS) that the user can access.

ITEMS

userPrivileges: object (on page 727)

userAuthSetting: *object*

DESCRIPTION

User authentication settings.

PROPERTIES

passwordComplexitySetting: passwordComplexitySettingOfUserAuthSetting: object (on page 641)

passwordAgeSetting: passwordAgeSettingOfUserAuthSetting: object (on page 639)

lockoutSetting: lockoutSettingOfUserAuthSetting: object (on page 637)

sessionSetting: sessionSettingOfUserAuthSetting: object (on page 687)

userGroup: *object*

DESCRIPTION

User group information.

PROPERTIES

memberUsers: *object[]* (1 to 32 items)

A list of users who belong to the user group.

For a user group on an external authentication server, no user on that server is output.

ITEMS

object

PROPERTIES

userId: *string* (5 to 255 chars) , must match /^[\-A-Za-z0-9!#\$%&'\.@\^_`\{\}~]{5,255}$/

User ID.

userObjectId: *string* (5 to 765 chars) , must match /^[A-Za-z0-9%\-\._~]{5,765}$/

Object ID of the user. For the user ID, a percent-encoded reserved string defined in RFC 3986 is output.

userGroupId: *string* (1 to 64 chars) , must match /^[a-zA-Z0-9!#\$%&'\-\.@\^_`\{\}~]{1,64}$/

User Group ID.

userGroupObjectId: *string* (1 to 192 chars) , must match /^[A-Za-z0-9%\-\._~]{1,192}$/

Object ID of the user group. For the user group ID, a percent-encoded reserved string defined in RFC 3986 is output.

roleNames: *string[]*

Role of the user group.

ITEMS

- For system administrators: (One to six items)

  string , x ∈ { "Security", "Storage", "Monitor", "Service", "Audit", "Resource" }

- For VPS administrators: (One to three items)

  string , x ∈ { "VpsSecurity", "VpsStorage", "VpsMonitor" }

isBuiltIn: *boolean*

Whether the user is a built-in user.

externalGroupName: *string* (1 to 4096 chars) nullable

Name of the user group registered with an external authorization server when the external authorization server is linked.

Null is returned for a user group on the storage cluster (not on an external authentication server).

vpsId: *string* , must match /^\(system\)$|[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

The ID of the virtual private storage (VPS) that the user group belongs to.

For a system administrator group, the string "(system)" is displayed.

scope: *string[]* (1 to 65 items)

An array of the IDs of virtual private storages (VPSs) that the user group can access.

ITEMS

must match /^system$|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

The IDs of VPSs that the user group can access.

For a system administrator group, the string "system" is displayed instead of an ID.

userGroupSummary: *object*

DESCRIPTION

User group summary information.

PROPERTIES

userGroupId: *string* (1 to 64 chars) , must match /^[a-zA-Z0-9!#\$%&'\-\.@\^_`\{\}~]{1,64}$/

User Group ID.

userGroupObjectId: *string* (1 to 192 chars) , must match /^[A-Za-z0-9%\-\._~]{1,192}$/

Object ID of the user group. For the user group ID, a percent-encoded reserved string defined in RFC 3986 is output.

roleNames: *string[]*

Role of the user group.

ITEMS

- For system administrators: (One to six items)

  string , x ∈ { "Security", "Storage", "Monitor", "Service", "Audit", "Resource" }

- For VPS administrators: (One to three items)

  string , x ∈ { "VpsSecurity", "VpsStorage", "VpsMonitor" }

isBuiltIn: *boolean*

Whether the user is a built-in user.

externalGroupName: *string* (1 to 4096 chars) nullable

Name of the user group registered with an external authorization server when the external authorization server is linked.

Null is returned for a user group on the storage cluster (not on an external authentication server).

vpsId: *string* , must match /^\(system\)$|[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

The ID of the virtual private storage (VPS) that the user group belongs to.

For a system administrator group, the string "(system)" is displayed.

scope: *string[]* (1 to 65 items)

An array of the IDs of virtual private storages (VPSs) that the user group can access.

ITEMS

must match /^system$|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

The ID of VPS that the user group can access.

For a system administrator group, the string "system" is displayed instead of an ID.

userPrivileges: *object*

DESCRIPTION

The information about the virtual private storage (VPS) that the user can access.

PROPERTIES

scope: *string* , must match /^system$|^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

The ID of the virtual private storage (VPS) that the user can access.

roleNames: *string[]*

A list of roles for the virtual private storage (VPS) that the user can access.

ITEMS

- For system administrators: (One to six items)

  string , x ∈ { "Security", "Storage", "Monitor", "Service", "Audit", "Resource" }

- For VPS administrators: (One to three items)

  string , x ∈ { "VpsSecurity", "VpsStorage", "VpsMonitor" }

version: *object*

DESCRIPTION

API version information.

PROPERTIES

apiVersion: *string* (7 to 11 chars) , must match /^[0-9]{1,2}\.[0-9]{1,2}\.[0-9]{1,2}\.[0-9]{1,2}$/

API version.

productName: *string*

API name.

virtualPrivateStorage: *object*

DESCRIPTION

The information about a virtual private storage (VPS).

PROPERTIES

id: ***string*** **, must match /^[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/**
The ID of the VPS.

name: ***string*** **(1 to 32 chars) , must match /^(?!system$)[\-A-Za-z0-9,\.:@_]{1,32}$/**
A VPS name.

upperLimitForNumberOfUserGroups: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 256 }**
The upper limit of the number of user groups that belong to the VPS.

numberOfUserGroupsCreated: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 256 }**
The number of user groups that have been created on the VPS.

upperLimitForNumberOfUsers: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 256 }**
The upper limit of the number of users that belong to the VPS.

numberOfUsersCreated: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 256 }**
The number of users that have been created on the VPS.

upperLimitForNumberOfSessions: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 436 }**
The upper limit of the number of sessions on the VPS.

numberOfSessionsCreated: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 436 }**
The number of sessions that have been created on the VPS.

upperLimitForNumberOfServers: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 1024 }**
The maximum allowable number of compute nodes on the VPS.

numberOfServersCreated: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 1024 }**
The number of compute nodes that have been created on the VPS.

upperLimitForNumberOfHbas: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 4096 }**
The upper limit of the number of initiators for the VPS.

numberOfHbasCreated: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 4096 }**
The number of initiators that have been created on the VPS.

upperLimitForNumberOfVolumeServerConnections: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 65536 }**
The upper limit of the number of information items about connections between volumes and compute nodes on the VPS.

numberOfVolumeServerConnectionsCreated: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 65536 }**
The number of information items about connections between volumes and compute nodes on the VPS.

volumeSettings: ***object***

The information about volumes on the VPS.

PROPERTIES

poolId: ***string*** **(uuid)**

The ID of the storage pool to be used.

upperLimitForNumberOfVolumes: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 32768 }**

The upper limit of the number of volumes on the VPS.

numberOfVolumesCreated: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 32768 }**

The number of volumes that have been created on the VPS.

upperLimitForCapacityOfVolumes: ***integer*** **(int64) , { x ∈ ℤ | 0 ≤ x ≤ 9223372036854775807 }**

The upper limit of the total volume capacity (MiB) of the VPS.

capacityOfVolumesCreated: ***integer*** **(int64) , { x ∈ ℤ | 0 ≤ x ≤ 9223372036854775807 }**

The total capacity (in MiB) of volumes that have been created on the VPS.

upperLimitForCapacityOfSingleVolume: ***integer*** **(int64) , { x ∈ ℤ | -1 ≤ x ≤ 6871947674 }**

The upper limit of the capacity (MiB) of a single volume of the VPS.

If the capacity of a single volume is not limited, -1 is displayed.

qosParam: ***object***

PROPERTIES

upperLimitForIopsOfVolume: ***integer*** **(int64) , { x ∈ ℤ | -1 ≤ x ≤ 2147483647 }**

The upper limit of volume performance (in IOPS) of the VPS.

This is used as the default value for the upper limit of performance (in IOPS) of volumes created on the VPS. If the upper limit of volume performance (in IOPS) is set, a value in the range from 100 to 2147483647 is displayed. The VPS administrator who creates volumes can set a value that is no more than this value as the upper limit of performance (in IOPS) for each volume. If the upper limit of volume performance (in IOPS) is not set, -1 is displayed.

upperLimitForTransferRateOfVolume: ***integer*** **(int64) , { x ∈ ℤ | -1 ≤ x ≤ 2097151 }**

The upper limit of volume performance (in MiB/s) for the VPS.

This is used as the default value for the upper limit of performance (in MiB/s) of volumes created on the VPS. If the upper limit of volume performance (in MiB/s) is set, a value in the range from 1 to 2097151 is displayed. The VPS administrator who creates volumes can set a value that is no more than this value as the upper limit of performance (in MiB/s) for each volume. If the upper limit of volume performance (in MiB/s) is not set, -1 is displayed.

upperAlertAllowableTimeOfVolume: *integer* (int32) , { x ∈ ℤ | -1 ≤ x ≤ 600 }
The alert threshold value (in seconds) for the upper limit of volume performance for the VPS.

This is used as the default value for the alert threshold for the upper limit of performance of volumes created on the VPS. If the alert threshold for the upper limit of volume performance is set, a value in the range from 1 to 600 is displayed. A message is output to the event log when restriction of the upper limit of performance specified by upperLimitForIopsOfVolume or upperLimitForTransferRateOfVolume continues for the specified length of time. If the alert threshold for the upper limit of volume performance is not set, -1 is displayed.

VirtualPrivateStorageDefaultVolumeSettings: *object*

PROPERTIES

poolId: *string* (uuid) required
The ID of the storage pool to be used on the virtual private storage (VPS).

upperLimitForNumberOfVolumes: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 32768 } required
The upper limit of the number of volumes on the VPS.

For details about the values to be specified for upperLimitForNumberOfVolumes, contact customer support.

upperLimitForCapacityOfVolumes: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 9223372036854775807 } required
The upper limit of the total volume capacity (MiB) of the VPS.

For details about the values to be specified for upperLimitForCapacityOfVolumes, contact customer support.

upperLimitForCapacityOfSingleVolume: ***integer*** **(int64) , { x ∈ ℤ | -1 ≤ x ≤ 6871947674 }**

The upper limit of the capacity (MiB) of a single volume of the VPS.

To place no limit on the capacity of a single volume, specify -1.

For details about the values to be specified for upperLimitForCapacityOfSingleVolume, contact customer support.

upperLimitForIopsOfVolume: ***integer*** **(int64) , { x ∈ ℤ | -1 ≤ x ≤ 2147483647 }**

The upper limit of volume performance (in IOPS) of the VPS.

This is used as the default value for the upper limit of performance (in IOPS) of volumes created on the VPS. To set the upper limit of volume performance (in IOPS), specify a value in the range from 100 to 2147483647. To set no upper limit, specify -1. If you specify values from 0 to 99, jobs will be unsuccessful.

The VPS administrator who creates volumes can set a value that is no more than this value as the upper limit of performance (in IOPS) for each volume. If you make both of the upperLimitForIopsOfVolume and upperLimitForTransferRateOfVolume settings unavailable, the setting of upperAlertAllowableTimeOfVolume is also made unavailable. If this property is omitted, -1 is assumed to be specified.

upperLimitForTransferRateOfVolume: *integer* (int64) , { x ∈ ℤ | -1 ≤ x ≤ 2097151 }

The upper limit of volume performance (in MiB/s) for the VPS.

This is used as the default value for the upper limit of performance (in MiB/s) of volumes created on the VPS. To set the upper limit of volume performance (in MiB/s), specify a value in the range from 1 to 2097151. To set no upper limit, specify -1. If you specify 0, jobs will be unsuccessful.

The VPS administrator who creates volumes can set a value that is no more than this value as the upper limit of performance (in MiB/s) for each volume. If you make both of the upperLimitForIopsOfVolume and upperLimitForTransferRateOfVolume settings unavailable, the setting of upperAlertAllowableTimeOfVolume is also made unavailable. If this property is omitted, -1 is assumed to be specified.

upperAlertAllowableTimeOfVolume: ***integer*** **(int32) , { x ∈ ℤ | -1 ≤ x ≤ 600 }**

The alert threshold value (in seconds) for the upper limit of volume performance for the VPS.

This is used as the default value for the alert threshold for the upper limit of performance of volumes created on the VPS. To set the alert threshold, specify a value in the range from 1 to 600. To set no alert threshold, specify -1.If you specify 0, jobs will be unsuccessful.

A message is output to the event log when restriction of the upper limit of performance specified by upperLimitForIopsOfVolume or upperLimitForTransferRateOfVolume continues for the specified length of time. This property can be specified if either upperLimitForIopsOfVolume or upperLimitForTransferRateOfVolume, or both, is set. If this property is omitted, -1 is assumed to be specified.

virtualPrivateStorageList: *object*

DESCRIPTION

A list of summary information about virtual private storages (VPSs) and information about data.

PROPERTIES

data: *object[]* (up to 64 items)
A list of summary information about virtual private storages (VPSs).

ITEMS
virtualPrivateStorage: object (on page 728)

summaryInformation: virtualPrivateStorageSummaryInformation: object (on page 732)

virtualPrivateStorageSummaryInformation: *object*

DESCRIPTION

PROPERTIES

totalCount: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 64 }
The number of virtual private storages (VPSs) that have been created.

totalUpperLimitForNumberOfUserGroups: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 16384 }
The total value of the upperLimitForNumberOfUserGroups properties set for the VPSs that have been created.

totalUpperLimitForNumberOfUsers: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 16384 }
The total value of the upperLimitForNumberOfUsers properties set for the VPSs that have been created.

totalUpperLimitForNumberOfSessions: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 16384 }
The total value of the upperLimitForNumberOfSessions properties set for the VPSs that have been created.

totalUpperLimitForNumberOfVolumes: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 2097152 }
The total value of the upperLimitForNumberOfVolumes properties set for the VPSs that have been created.

totalUpperLimitForCapacityOfVolumes: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 9223372036854775807 }
The total value (in MiB) of the upperLimitForCapacityOfVolumes properties set for the VPSs that have been created.

totalUpperLimitForNumberOfServers: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 65536 }

The total value of the upperLimitForNumberOfServers properties set for the VPSs that have been created.

totalUpperLimitForNumberOfHbas: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 262144 }

The total value of the upperLimitForNumberOfHbas properties set for the VPSs that have been created.

totalUpperLimitForNumberOfVolumeServerConnections: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 4194304 }

The total value of the upperLimitForNumberOfVolumeServerConnections properties set for the VPSs that have been created.

volume: *object*

DESCRIPTION

Volume information.

PROPERTIES

reservedCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 268435456 }

The reserved (logical) capacity (unit: MiB).

Null is output if "volumeType" is "ExternalMigrationOrigin".

freeCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 268435456 } nullable

Available (logical) capacity (unit: MiB).

Null is output if "volumeType" is "ExternalMigrationOrigin".

luns: *object[]* nullable

A list of volume LUNs

ITEMS

object

PROPERTIES

lun: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 8191 }

LUN

serverId: *string* (uuid)

The ID of the compute node.

snapshotProgressRate: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 100 } nullable

The progress of preparing, deleting, or restoring snapshots (unit: %).

- Null is output when snapshotAttribute is a hyphen (-) or P-VOL.
- The progress rate is output only when snapshotStatus is one of Preparing, Deleting, or Restoring. In the case of other statuses, null is output.

snapshotTimestamp: *string* (date-time) nullable

The time when snapshots are obtained is output when snapshotAttribute is "S-VOL" or "P/S-VOL".

When snapshotStatus is other than Normal, null is output.

snapshotType: *string* , x ∈ { "Snapshot" } nullable

The type of the snapshot. Null is output when snapshotAttribute is a hyphen (-) or P-VOL.

savingEffects: savingEffectOfVolume: object (on page 680)

snapshotConcordanceRate: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 100 } nullable

The matching rate between the snapshot creation-source volume (P-VOL) and the applicable volume (unit: %).

- The matching rate is output only when snapshotAttribute is "S-VOL" or "P/S-VOL". When snapshotAttribute is "-" or "P-VOL", null is output.
- The matching rate is output when snapshotStatus is Normal or Prepared. In the case of other statuses, null is output.

isWrittenInSvol: *boolean* nullable

When snapshotStatus is Normal, whether writing is performed from the compute node to the volume is output.

- true: Writing to the volume is performed.
- false: Writing to the volume is not performed.

When snapshotStatus is other than Normal, null is output.

id: *string* (uuid)

Volume ID.

name: *string* (1 to 32 chars)

Volume name. The same name cannot be used in multiple volumes.

nickname: *string* (1 to 32 chars)

The nickname of the volume. The same nickname can be used in multiple volumes.

volumeNumber: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 131071 }

Volume number.

poolId: *string* (uuid) nullable

The ID of the storage pool.

Null is output if "volumeType" is "ExternalMigrationOrigin".

poolName: *string* (1 to 32 chars) nullable

The name of the storage pool.

Null is output if "volumeType" is "ExternalMigrationOrigin".

totalCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 268435456 }

The total (logical) capacity (unit: MiB).

(Virtual machine) An external volume or a volume after data migration might not be in 1-MiB units depending on the capacity of the external volume. In this case, the value is rounded down to 1 MiB.

usedCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 268435456 } nullable

The used (logical) capacity (unit: MiB).

Null is output if "volumeType" is "ExternalMigrationOrigin".

Equals totalCapacity if fullAllocated is true.

numberOfConnectingServers: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 4096 }

The number of connected compute nodes.

numberOfSnapshots: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 1023 }

The number of Snapshots.

protectionDomainId: *string* (uuid)

The ID of the protection domain to which the volume is belonging.

fullAllocated: *boolean*

Specifies whether all the area for writing user data is pre-allocated.

fullAllocated is currently fixed to false.

- false: The area is allocated according to the amount of written user data.

volumeType: *string* , x ∈ { "Normal", "Snapshot", "MigrationDestination", "ExternalMigrationOrigin" }

A list of volume types (attributes)

- Normal: Normal volume
- Snapshot: Volume which is used in Snapshot
- MigrationDestination: Volume being used as the migration destination
- (Virtual machine) ExternalMigrationOrigin: The migration source volume (virtual volume)

statusSummary: *string* , x ∈ { "Normal" , "Warning" , "Error" }

Summary of the volume status.

- Normal: No action by the user is required.
- Warning: Although immediate action by the user is not required, some action may have to be taken.
- Error: Immediate action by the user is required.

status: *string* , x ∈ { "Normal" , "Deleting" , "Updating" , "Expanding" , "CreationFailed" , "DeletionFailed" , "UpdateFailed" , "ExpansionFailed" , "IOSuppressed" , "MetaDataConsistencyError" }

The volume status. The following status is output:

- Normal: Operating normally.
- Deleting: Being deleted.
- Updating: The setting is being updated.
- Expanding: The capacity is being expanded.
- CreationFailed: Creation ended abnormally.
- DeletionFailed: Deleting ended abnormally.
- UpdateFailed: The setting update ended abnormally.
- ExpansionFailed: Capacity expansion did not succeed.
- IOSuppressed: I/O is suppressed.
- MetaDataConsistencyError: Metadata used by the data reduction function is inconsistent.

storageControllerId: *string* (uuid) nullable

The ID of the storage controller that manages this volume.

Null is output if "volumeType" is "ExternalMigrationOrigin".

snapshotAttribute: *string* , x ∈ { "-" , "P-VOL" , "P/S-VOL" , "S-VOL" }

The snapshot attribute. A hyphen (-) indicates that the volume is other than P-VOL, P/S-VOL, or S-VOL.

snapshotStatus: *string* , x ∈ { "Normal" , "Deleting" , "Restoring" , "Empty " , "Preparing" , "Prepared" , "Error" } nullable

The snapshot status. Any one of the following statuses is output:

- Normal: Operating normally.
- Deleting: Being deleted.
- Restoring: Being restored.
- Empty: Status in which snapshots are empty without metadata. This is a temporary status when creating or deleting S-VOLs. After waiting for a while, the status transitions to Preparing when creating S-VOLs and the applicable volumes are deleted when deleting S-VOLs.
- Preparing: Being prepared for creating snapshots.
- Prepared: Completed preparation for creating snapshots.
- Error: Operating abnormally.

The status is output only when snapshotAttribute is "S-VOL" or "P/S-VOL". When snapshotAttribute is "-" or "P-VOL", null is output.

savingSetting: *string* , x ∈ { "Disabled" }

Settings of the data reduction function.

- Disabled: The data reduction function is disabled.

savingMode: *string* , x ∈ { "null" } nullable

Processing mode of the data reduction function.

savingMode is currently fixed to null.

dataReductionStatus: *string* , x ∈ { "Disabled" }

Status of data reduction.

- Disabled: The data reduction function is disabled.

dataReductionProgressRate: *integer* (int32) , { x ∈ ℤ | null } nullable

Progress of the data reduction function (unit: %).

dataReductionStatus is currently fixed to null.

vpsId: *string* , must match /^\(system\)$|[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/

The ID of the virtual private storage (VPS) that the acquisition-target resource belongs to.

If the resource does not belong to a VPS, the reserved word "(system)", which indicates that the resource is independent, is output instead of an ID.

vpsName: *string* (1 to 32 chars) , must match /^\(system\)$|[\-A-Za-z0-9,\.:@_]{1,32}$/

The name of the virtual private storage (VPS) that the acquisition-target resource belongs to.

If the resource does not belong to a VPS, the reserved word "(system)", which indicates that the resource is independent, is output.

qosParam: *object*

PROPERTIES

upperLimitForIops: *integer* (int64) , { x ∈ ℤ | -1 ≤ x ≤ 2147483647 }

The upper limit of volume performance (in IOPS).

The value -1 indicates no upper limit is placed on volume performance (in IOPS).

upperLimitForTransferRate: *integer* (int64) , { x ∈ ℤ | -1 ≤ x ≤ 2097151 }

The upper limit of volume performance (in MiB/s).

The value -1 indicates that no upper limit is placed on volume performance (in MiB/s).

upperAlertAllowableTime: *integer* (int32) , { x ∈ ℤ | -1 ≤ x ≤ 600 }

The alert threshold value (in seconds) for the upper limit of volume performance.

The value -1 indicates that no entries are output to the event log.

VPS administrators cannot specify this property when creating or editing volumes.

upperAlertTime: *string* (date-time) nullable

The last time the upper limit of volume performance was continuously exceeded and the conditions for the alert threshold of the performance upper limit were met (UTC).

naaId: *string* (up to 32 chars) , must match /^[0-9a-f]{32}$/ nullable

NAA ID of volumes.

If the volumes were created by the software version earlier than 01.12.0x.xx, null is output.

volumeCapacity: *object*

DESCRIPTION

The capacity information of the volume (monitor information). Null is set if the information could not be obtained.

PROPERTIES

id: *string* (uuid)
Volume ID.

vpsId: *string*
The ID of the virtual private storage (VPS) that the acquisition-target resource belongs to.

If the resource does not belong to a VPS, the reserved word "(system)", which indicates that the resource is independent, is output instead of an ID.

capacityUsage: *integer* (int64) nullable
Usage amount (unit: MiB).

volumeCapacityListResponseData: *object*

DESCRIPTION

The capacity information (monitor information) of the volume when the information was collected. For performanceObjects, volumes of which capacity information could be collected when the API was executed are output as an array.

PROPERTIES

timestamp: *string* (date-time)
The time when this information was collected.

performanceObjects: *object[]*
A list of volume capacity information (monitor information) collected at the time specified for timestamp.

ITEMS
volumeCapacity: object (on page 738)

volumeNameParam: *object*

DESCRIPTION

A parameter for creating a name.

name = baseName + <*serial-number*> (the number of digits is set by numberOfDigits).

The maximum number of characters for a name is 32.

The name must be unique throughout the volumes. If a volume with the name according to the formula above already exists, creation of all the volumes does not succeed.

PROPERTIES

baseName: ***string*** **(1 to 32 chars) , must match /^[\-A-Za-z0-9,\.:@_]{1,32}$/required**
A name which is set for a volume.

startNumber: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 32767 }**
The first sequential number suffixed to a name.

This parameter is required if the parameter for the number of volumes to be created is 2 or more.

numberOfDigits: ***integer*** **(int32) , { x ∈ ℤ | 1 ≤ x ≤ 5 } (default: "1")**
The number of digits of a serial number suffixed to a name.

volumeNicknameParam: *object*

DESCRIPTION

A parameter for creating a nickname.

The following formula is valid only when startNumber is set. Specify the number of digits using numberOfDigits.

The maximum number of characters for a nickname is 32.

The same nickname can be set for volumes.

nickname = baseName + <*serial-number*>

PROPERTIES

baseName: ***string*** **(1 to 32 chars) , must match /^[\-A-Za-z0-9,\.:@_]{1,32}$/required**
A nickname which is set for a volume.

startNumber: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 32767 }**
The first sequential number suffixed to a nickname. If omitted, no sequential number is used for nicknames.

numberOfDigits: ***integer*** **(int32) , { x ∈ ℤ | 1 ≤ x ≤ 5 } (default: "1")**
The number of digits of a serial number added to the end of a nickname. This value can be specified only when startNumber is set.

volumePath: *object*

DESCRIPTION

Connection information between the volume and the compute node.

PROPERTIES

id: ***string***

The ID of connection information between the volume and the compute node.

Format: "*(volumeId)* , *(serverId)*"

Where volumeID is the ID of the volume, and serverID is the ID of the compute node.

serverId: ***string*** **(uuid)**

The ID of the compute node.

volumeId: ***string*** **(uuid)**

Volume ID.

lun: ***integer*** **(int32) , { x ∈ ℤ | 0 ≤ x ≤ 8191 }**

LUN

vpsId: ***string*** **, must match /^\(system\)$|[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/**

The ID of the virtual private storage (VPS) that the acquisition-target resource belongs to.

If the resource does not belong to a VPS, the reserved word "(system)", which indicates that the resource is independent, is output instead of an ID.

vpsName: ***string*** **(1 to 32 chars) , must match /^\(system\)$|[\-A-Za-z0-9,\.:@_]{1,32}$/**

The name of the virtual private storage (VPS) that the acquisition-target resource belongs to.

If the resource does not belong to a VPS, the reserved word "(system)", which indicates that the resource is independent, is output.

volumePerformance: *object*

DESCRIPTION

Performance information of the volume (monitor information). Null is set if the information could not be obtained. The infinite value (Infinity) is not output for double type values.

PROPERTIES

id: ***string*****(uuid)**

Volume ID.

vpsId: ***string*** **(uuid)**

The ID of the virtual private storage (VPS) that the acquisition-target resource belongs to.

If the resource does not belong to a VPS, the reserved word "(system)", which indicates that the resource is independent, is output instead of an ID.

readIOPS: *integer* (int32) nullable
The number of read I/Os per second (unit: IOPS).

writeIOPS: *integer* (int32) nullable
The number of write I/Os per second (unit: IOPS).

readTransferRate: *number* (double) nullable
Read data transfer amount per second (unit: MiB/sec).

writeTransferRate: *number* (double) nullable
Write data transfer amount per second (unit: MiB/sec).

readResponseTime: *number* (double) nullable
Average response time of the volume (unit: msec).

writeResponseTime: *number* (double) nullable
Average write response time of the volume (unit: msec).

volumePerformanceListResponse: *object*

DESCRIPTION

A list of volume performance information (monitor information) at the specified time.

PROPERTIES

data: *object[]*
ITEMS
volumePerformanceListResponseData: object (on page 742)

volumePerformanceListResponseData: *object*

DESCRIPTION

The performance information (monitor information) of the volume when the information was collected. For performanceObjects, volumes of which performance information could be collected when the API was executed are output as an array.

PROPERTIES

timestamp: *string* (date-time)
The time when this information was collected.

performanceObjects: *object[]*
A list of volume performance information (monitor information) collected at the time specified for timestamp.

ITEMS

volumePerformance: object (on page 741)

volumeQosParam: *object*

DESCRIPTION

A QoS-related parameter.

PROPERTIES

upperLimitForIops: *integer* (int64) , { x ∈ ℤ | -1 ≤ x ≤ 2147483647 }

The upper limit of volume performance (in IOPS).

To set the upper limit of volume performance (in IOPS), specify a value in the range from 100 to 2147483647. To set no upper limit, specify -1. If you specify values from 0 to 99, jobs will be unsuccessful.

If you make both of the upperLimitForIops and upperLimitForTransferRate settings unavailable, the setting of upperAlertAllowableTime is also made unavailable.

If this property is omitted, following valiue is specified.

- If the volume belongs to a virtual private storage (VPS): Default value set for the VPS
- If the volume does not belong to a VPS: -1

upperLimitForTransferRate: *integer* (int64) , { x ∈ ℤ | -1 ≤ x ≤ 2097151 }

The upper limit of volume performance (in MiB/s).

To set the upper limit of volume performance (in MiB/s), specify a value in the range from 1 to 2097151. To set no upper limit, specify -1. If you specify 0, jobs will be unsuccessful.

If you make both of the upperLimitForIops and upperLimitForTransferRate settings unavailable, the setting of upperAlertAllowableTime is also made unavailable.

If this property is omitted, following valiue is specified.

- If the volume belongs to a virtual private storage (VPS): Default value set for the VPS
- If the volume does not belong to a VPS: -1

upperAlertAllowableTime: *integer* (int32) , { x ∈ ℤ | -1 ≤ x ≤ 600 }

The alert threshold value (in seconds) for the upper limit of volume performance.

The VPS administrator cannot specify this property.

To set the alert threshold, specify a value in the range from 1 to 600. To set no alert threshold, specify -1. If you specify 0, jobs will be unsuccessful.

A message is output to the event log when restriction of the upper limit of performance specified by upperLimitForIops or upperLimitForTransferRate continues for the specified length of time.

This property can be specified if either upperLimitForIops or upperLimitForTransferRate, or both, is set.

If this property is omitted, following valiue is specified.

- If the volume belongs to a virtual private storage (VPS): Default value set for the VPS
- If the volume does not belong to a VPS: -1

volumeSummary: *object*

DESCRIPTION

Summary information of the volume.

PROPERTIES

savingEffects: savingEffectOfVolumeSummary: object (on page 681)

id: *string* (uuid)
Volume ID.

name: *string* (1 to 32 chars)
The name of the volume. The same name cannot be used in multiple volumes.

nickname: *string* (1 to 32 chars)
The nickname of the volume. The same nickname can be used in multiple volumes.

volumeNumber: *integer* (int32) , { x $\in \mathbb{Z}$ | 0 ≤ x ≤ 131071 }
Volume number.

poolId: *string* (uuid) nullable
The ID of the storage pool.

Null is output if "volumeType" is "ExternalMigrationOrigin".

poolName: *string* (1 to 32 chars) nullable
The name of the storage pool.

Null is output if "volumeType" is "ExternalMigrationOrigin".

totalCapacity: *integer* (int64) , { x $\in \mathbb{Z}$ | 0 ≤ x ≤ 268435456 }
The total (logical) capacity (unit: MiB).

(Virtual machine) An external volume or a volume after data migration might not be in 1-MiB units depending on the capacity of the external volume. In this case, the value is rounded down to 1 MiB.

usedCapacity: *integer* (int64) , { x ∈ ℤ | 0 ≤ x ≤ 268435456 } nullable

The used (logical) capacity (unit: MiB).

Null is output if "volumeType" is "ExternalMigrationOrigin".

Equals totalCapacity if fullAllocated is true.

numberOfConnectingServers: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 4096 }

The number of connected compute nodes.

numberOfSnapshots: *integer* (int32) , { x ∈ ℤ | 0 ≤ x ≤ 1023 }

The number of Snapshots.

protectionDomainId: *string* (uuid)

The ID of the protection domain to which the volume is belonging.

fullAllocated: *boolean*

Specifies whether all the area for writing user data is pre-allocated.

fullAllocated is currently fixed to false.

- false: The area is allocated according to the amount of written user data.

volumeType: *string* , x ∈ { "Normal", "Snapshot", "MigrationDestination", "ExternalMigrationOrigin" }

A list of volume types (attributes)

- Normal: Normal volume
- Snapshot: Volume which is used in Snapshot
- MigrationDestination: Volume being used as the migration destination
- (Virtual machine) ExternalMigrationOrigin: The migration source volume (virtual volume)

statusSummary: *string* , x ∈ { "Normal" , "Warning" , "Error" }

Summary of the volume status.

- Normal: No action by the user is required.
- Warning: Although immediate action by the user is not required, some action may have to be taken.
- Error: Immediate action by the user is required.

status: *string* , x ∈ { "Normal" , "Deleting" , "Updating" , "Expanding" , "CreationFailed" , "DeletionFailed" , "UpdateFailed" , "ExpansionFailed" , "IOSuppressed" , "MetaDataConsistencyError" }

The volume status. The following status is output:

- Normal: Operating normally.
- Deleting: Being deleted.
- Updating: The setting is being updated.
- Expanding: The capacity is being expanded.
- CreationFailed: Creation ended abnormally.
- DeletionFailed: Deleting ended abnormally.
- UpdateFailed: The setting update ended abnormally.
- ExpansionFailed: Capacity expansion did not succeed.
- IOSuppressed: I/O is suppressed.
- MetaDataConsistencyError: Metadata used by the data reduction function is inconsistent.

storageControllerId: *string* (uuid) nullable

The ID of the storage controller that manages this volume.

Null is output if "volumeType" is "ExternalMigrationOrigin".

snapshotAttribute: *string* , x ∈ { "-", "P-VOL", "P/S-VOL", "S-VOL" }

The snapshot attribute. A hyphen (-) indicates that the volume is other than P-VOL, P/S-VOL, or S-VOL.

snapshotStatus: *string* , x ∈ { "Normal" , "Deleting" , "Restoring" , "Empty" , "Preparing" , "Prepared" , "Error" } nullable

The snapshot status. Any one of the following statuses is output:

- Normal: Operating normally.
- Deleting: Being deleted.
- Restoring: Being restored.
- Empty: Status in which snapshots are empty without metadata. This is a temporary status when creating or deleting S-VOLs. After waiting for a while, the status transitions to Preparing when creating S-VOLs and the applicable volumes are deleted when deleting S-VOLs.
- Preparing: Being prepared for creating snapshots.

- Prepared: Completed preparation for creating snapshots.
- Error: Operating abnormally.

The status is output only when snapshotAttribute is "S-VOL" or "P/S-VOL". When snapshotAttribute is "-" or "P-VOL", null is output.

savingSetting: *string* , x ∈ { "Disabled" }
Settings of the data reduction function.

- Disabled: The data reduction function is disabled.

savingMode: *string* , x ∈ { "null" } nullable
Processing mode of the data reduction function.

savingMode is currently fixed to null.

dataReductionStatus: *string* , x ∈ { "Disabled" }
Status of data reduction.

- Disabled: The data reduction function is disabled.

dataReductionProgressRate: *integer* (int32) , { null }
Progress of the data reduction function (unit: %).

dataReductionStatus is currently fixed to null.

vpsId: *string* , must match /^\(system\)$|[A-Fa-f0-9]{8}(-[A-Fa-f0-9]{4}){3}-[A-Fa-f0-9]{12}$/
The ID of the virtual private storage (VPS) that the acquisition-target resource belongs to.

If the resource does not belong to a VPS, the reserved word "(system)", which indicates that the resource is independent, is output instead of an ID.

vpsName: *string* (1 to 32 chars) , must match /^\(system\)$|[\-A-Za-z0-9,\.:@_]{1,32}$/
The name of the virtual private storage (VPS) that the acquisition-target resource belongs to.

If the resource does not belong to a VPS, the reserved word "(system)", which indicates that the resource is independent, is output.

qosParam: *object*
A QoS-related parameter.

PROPERTIES

upperLimitForIops: *integer* (int64) , { x ∈ ℤ | -1 ≤ x ≤ 2147483647 }
The upper limit of volume performance (in IOPS).

The value -1 indicates that no upper limit is placed on volume performance (in IOPS).

upperLimitForTransferRate: *integer* (int64) , { x ∈ ℤ | -1 ≤ x ≤ 2097151 }
The upper limit of volume performance (in MiB/s).

The alert threshold value (in seconds) for the upper limit of volume performance.

upperAlertAllowableTime: *integer* (int32) , { x ∈ ℤ | -1 ≤ x ≤ 600 }
The alert threshold value (in seconds) for the upper limit of volume performance.

The value -1 indicates that no entries are output to the event log.

VPS administrators cannot specify this property when creating or editing volumes.

upperAlertTime: *string* (date-time) nullable
The last time the upper limit of volume performance was continuously exceeded and the conditions for the alert threshold of the performance upper limit were met (UTC).

naaId: *string* (up to 32 chars) , must match /^[0-9a-f]{32}$/ nullable
NAA ID of volumes.

If the volumes were created by the software version earlier than 01.12.0x.xx, null is output.

warningThresholdSettingOfEditLicenseSetting: *object*

DESCRIPTION

A parameter for setting a license warning threshold.

PROPERTIES

remainingDays: *integer* (int32) , { x ∈ ℤ | -1 ≤ x ≤ 60 }
Number of remaining days. A warning is issued if the number of remaining days becomes less than the specified number of days.

-1 means that no warning is issued.

totalPoolCapacityRate: *integer* (int32) , { x ∈ ℤ | -1 ≤ x ≤ 100 }
Capacity rate warning. A warning is issued if the current total logical capacity of all storage pools exceeds the specified capacity rate in the total logical capacity of all storage pools permitted by the license.

-1 means that no warning is issued.

warningThresholdSettingOfLicenseSetting: *object*

DESCRIPTION

License warning threshold setting.

PROPERTIES

remainingDays: *integer* (int32) , { x ∈ ℤ | -1 ≤ x ≤ 60 }

Number of remaining days. A warning is issued if the number of remaining days becomes less than the specified number of days.

-1 means that no warning is issued.

totalPoolCapacityRate: *integer* (int32) , { x ∈ ℤ | -1 ≤ x ≤ 100 }

Warned capacity rate. A warning is issued if the current total logical capacity of all storage pools exceeds the specified capacity rate in the total logical capacity of all storage pools permitted by the license.

-1 means that no warning is issued.

webServerAccessSetting: *object*

DESCRIPTION

Web server access settings.

PROPERTIES

whitelistSetting: whitelistSettingOfWebServerAccessSetting: object (on page 750)

whitelistSettingOfEditWebServerAccessSetting: *object*

DESCRIPTION

A parameter for setting a whitelist for the web server.

When the whitelist function is enabled or a transmission source list of the whitelist is set when the function is enabled, whether the transmission source IP address of the request is included in the transmission source list is verified. If the IP address is not included, the job does not succeed.

PROPERTIES

isEnabled: *boolean*

Enables or disables the whitelist function for the web server.

Enables the function when the value is true. Disables the function if the value is false.

clientNames: ***string[]*** **(1 to 10 items)**

Transmission source list which is set in the whitelist for the web server.

HTTP status code 400 (Bad Request) is returned if:

- A character string which is not an IPv4 address is specified.
- A net mask is specified.

clientNames which were originally set are overwritten if this parameter is specified.

ITEMS

string **, must match /^(([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])$/**

whitelistSettingOfWebServerAccessSetting: *object*

DESCRIPTION

Web server whitelist settings.

PROPERTIES

isEnabled: ***boolean***

Enables or disables the whitelist function for the web server.

clientNames: ***string[]*** **(up to 10 items)**

Transmission source list which is set in the whitelist for the web server.

ITEMS

string **(7 to 15 chars) , must match /^(([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])$/**

Glossary

Auto recovery

See *Storage node auto-recovery* in the Glossary.

base license

A license that provides basic functionality.

blocked, blocking, blockage

A state for a storage or resources that comprise a storage where I/O operations cannot be performed.

BMC network

Network that connects the storage node BMC and the controller node. This network is used to operate the BMC from the controller node.

BMC port

The port that is on a storage node and is used for connection to the BMC network.

capacity balancing

Function of moving volumes automatically from high capacity usage storage controllers to low capacity usage storage controllers when capacity usage is not balanced among storage controllers.

cluster master node (primary)

A storage node within the storage cluster that has the role of managing the entire storage cluster.

cluster master node (secondary)

A storage node in the storage cluster that is responsible for managing the entire storage cluster in the event of failure of the cluster master node (primary).

cluster worker node

A storage node in the storage cluster that does not have the role of managing the entire storage cluster.

compute network

A network between a compute node and a storage node. Used for input / output of user data.

compute node

A node that the application of the user operates and instructs input / output of user data to the storage node. A host connected to the compute port.

compute port

(Virtual machine) The virtual port that is on a storage node and connects to the compute network.

(Bare metal) The port that is on a storage node and connects to the compute network.

configuration backup file

Backup file of storage cluster configuration information.

Configuration file

(Virtual machine) Generic term for VSS block configuration file and VM configuration file.

(Bare metal) A synonym for the VSS block configuration file.

Console interface

The interface of a storage node console (such as a virtual console via BMC).

control network

(Virtual machine) The network between the controller node and the storage node or maintenance node. It is used for Virtual Storage Software block management operation and communication with external service such as SNMP and NTP.

(Bare metal) The network between the controller node and the storage node. It is used for Virtual Storage Software block management operation and communication with external service such as SNMP and NTP.

control port

(Virtual machine) The virtual port that is on a storage node and connects to the control network.

(Bare metal) The port that is on a storage node and connects to the control network.

controller node

A management node used to instruct Virtual Storage Software block's management function (volume creation, etc.).

data migration

A functionality to migrate data from an external storage system into Virtual Storage Software block in volume units.

disk controller

Hardware required to use a drive.

drive

A physical device that stores user data and the OS. Common name for SSDs and HDDs.

drive data relocation

Function of balancing data capacity among storage nodes (to optimize capacity efficiency of each storage node) when capacity becomes unbalanced among storage nodes due to storage node addition or removal.

event log

A file that records the operation of the system. In Virtual Storage Software block, it refers to the log for the purpose of fault notification.

Failover

Switching the cluster master (secondary) to the cluster master (primary) in the event of failure of the cluster master (primary).

fault domain

A group of storage nodes sharing power system and network switch. A configuration for making it possible to continue the operation of storage even if the storage nodes in a group collectively become abnormal.

initiator

An endpoint on the compute node side when accessing a volume from a compute node.

internode network

Network between storage nodes. Used for communication of user data and management information between storage nodes.

internode port

(Virtual machine) The virtual port that is on a storage node and connects to the internode network.

(Bare metal) The port that is on a storage node and connects to the internode network.

license key

Key to activate the corresponding license in Virtual Storage Software block.

maintenance blockage

See *Storage node maintenance blocking* in the Glossary.

maintenance node

VM that is configured inside some of the storage nodes, and which is used to configure and manage Virtual Storage Software block.

maintenance recovery

See *Storage node maintenance recovery* in the Glossary.

multi-tenancy function

Function to allow resources of a storage in a large storage system to be distributed to and shared by multiple tenants (companies and divisions). A storage distributed to each tenant is called VPS (Virtual Private Storage).

normal volume

Volume that is neither P-VOL, S-VOL, nor P/S-VOL.

other volume capacity

Total capacity of snapshot volumes (S-VOLs and P/S-VOLs).

OVA

An acronym for the Open Virtualization Appliance/Application. The following files are bundled into one tar ball.

1. OVF file contains the virtual machine attributes etc.
2. Disk image or ISO image created by certain Hypervisor software.
3. Manifest file contains hash value for each file (mf option).
4. Certification file for digital signage for Manifest files (cert option).

OVF

Acronym for Open Virtualization Format. OVF is a standard format designed to allow different virtualization software to exchange virtual machine image files with each other.

P-VOL

Volume of the copy source.

P/S-VOL

Volume having both the P-VOL and S-VOL attributes in a snapshot tree in cascade configuration.

physical node

In an environment where storage is used, a physical server that belongs to that environment.

program product license

A license provided on a per-function basis.

protection domain

Setting for limiting the range of failure if an error occurs in a storage node or the network between storage nodes.

provisioned volume capacity

Total capacity of normal volumes and snapshot volumes (P-VOLs).

rebuild

Function of automatically restoring redundancy of data whose redundancy was reduced due to a drive failure or storage node failure.

Rebuild capacity

Capacity in a storage pool secured for Data rebuild at the time of drive failure.

Representative storage node

A storage node that is used to configure a storage cluster in the setup procedure for the bare metal model. This node is different from a cluster master node (primary).

S-VOL

The copy destination volume.

scale out

A method of increasing the number of CPUs, memory capacity, and the number of drives by adding storage nodes to improve system performance and capacity.

scope

The range of resources that users can operate. A scope is set for a user group. A scope for a user is determined according to the user group to which the user belongs.

snapshot volume

Volume that is either a P-VOL, S-VOL, or P/S-VOL.

spare node

Standby storage node used for the spare node function.

spare node function

Function to allow restoration of redundancy by performing spare node switchover. Spare node switchover from a faulty storage node to a storage node that is registered as a standby storage node in the storage cluster is performed when the faulty storage node cannot be restored by the auto-recovery function.

storage cluster

A virtual storage system built from multiple storage nodes.

storage controller

Part of Virtual Storage Software block processes that manage storage node capacities and volumes.

storage controller relocation

Function of optimizing the number of the storage controllers of each storage node when the number of the storage controllers becomes unbalanced among storage nodes due to storage node addition or removal.

storage node

Physical server to which the CPU, memory, and drives that comprise Virtual Storage Software block are assigned. Alternatively, this term refers to a process group of Virtual Storage Software block software running on storage nodes.

storage node addition

A process of adding a storage node to a storage cluster.

Storage node auto-recovery

Function to execute self-diagnosis and self-recovery by a storage node to recover the storage node from server failures due to software factors (firmware, driver, and so on) or due to temporary network problems between storage nodes.

Storage node maintenance blocking

Process of separating a storage node from a storage cluster temporarily and placing the storage node in a status that allows for part replacement or other maintenance.

Storage node maintenance recovery

Process of returning a storage node to the available status again after it was blocked by manual operation or due to a failure.

storage node removal

A process of removing a storage node from a storage cluster.

storage node replacement

A functionality or process that manually recovers a blocked storage node.

Replace the following to recover the blocked storage node.

(Virtual machine) Storage node VM

(Bare metal) Physical node

storage pool

Logical user data storage area that combines multiple drives.

storage software

The Virtual Storage Software block software that realizes a storage cluster.

system administrator

Administrator who manages the entire system.

target

An endpoint on the storage cluster side when accessing a volume from a compute node.

temporary volume capacity

Total capacity of volumes created temporarily by Data migration and Capacity balance.

thin provisioning

Method of creating a virtual storage in which the minimum required capacity is initially secured, and then expanded as required.

virtual machine (VM)

Virtual machine.

virtual private storage

Virtual storage logically divided from a storage cluster in a multi-tenancy configuration.

volume

A logical device that mounts on a compute node to read or write user data.

volume migration

Moving volumes (existing on a storage node to be removed) to another storage node.

volume path

Connection information between a compute node and a volume. One of the setting information necessary for using a volume from a compute node.

VPS

Acronym for Virtual Private Storage. See *virtual private storage* in the Glossary.

VPS administrator

Administrator who manages a virtual private storage (VPS) in a multi-tenancy configuration.

Hitachi Vantara

Corporate Headquarters
2535 Augustine Drive
Santa Clara, CA 95054 USA

HitachiVantara.com/contact