Hitachi NAS File Storage

NAS File OS v14.2.7100 or higher

API v7.2.1

REST API Reference

This document describes the File Storage Native REST API for the Hitachi NAS Platform.

Contents

Contents

Contents

Contents

Contents

Preface

You can use the Hitachi NAS REST API server to administer Hitachi storage systems using standard HTTP protocol operations.

This REST API provides a consistent interface for client applications.

Intended audience

This document is for REST API server application developers who want to integrate with supported Hitachi storage systems.

Application developers are expected to have a basic knowledge of Hitachi storage system administration and a functional knowledge of file storage data services such as thin provisioning, snapshots, directory cloning, and other features.

Software version

The version of the REST API is 07.2.1.

Document revision history

Table 1

Revision	Date	Description
7.2.1	January 2022	Added output paging, added tier information to filesystem and storagepool API responses and added "Get multiple file system storage statistices" API call.
7.2.0	February 2021	Deprecated Async option, corrected some documentation errors, and added ability to supply parameters as part of the URL. Added read-only support for the API.
7.1.3	April 2020	Added <u>Get virus scan info of a virtual server (on page 298)</u>
7.1.2	December 2019	No new APIs.

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.)

Convention	Description
`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: [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.

Conventions for storage capacity values

Physical storage capacity values (for example, disk drive capacity) are calculated based on the following values:

Physical capacity unit	Value
1 kilobyte (KB)	1,000 (10^3) bytes
1 megabyte (MB)	1,000 KB or $1,000^2$ bytes
1 gigabyte (GB)	1,000 MB or $1,000^3$ bytes
1 terabyte (TB)	1,000 GB or $1,000^4$ bytes
1 petabyte (PB)	1,000 TB or $1,000^5$ bytes
1 exabyte (EB)	1,000 PB or $1,000^6$ bytes

Logical capacity values (for example, logical device capacity, cache memory capacity) are calculated based on the following values:

Logical capacity unit	Value
1 block	512 bytes
1 cylinder	Mainframe: 870 KB Open-systems: ▪ OPEN-V: 960 KB ▪ Others: 720 KB
1 KB	1,024 (2^{10}) bytes
1 MB	1,024 KB or $1,024^2$ bytes
1 GB	1,024 MB or $1,024^3$ bytes
1 TB	1,024 GB or $1,024^4$ bytes
1 PB	1,024 TB or $1,024^5$ bytes
1 EB	1,024 PB or $1,024^6$ bytes

Comments

Please send us your comments on this document 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: Hardware and software requirements

This section describes the hardware and software required to support the REST API server.

Enabling Native REST API

This document covers API version 7 supported by the HNAS native REST API. Native API mode has been introduced in software release 13.9.6800, and will be used for future API improvements and features, replacing the previous legacy API versions.

To check which mode is enabled, use the rest-server-status command:

```
G800-443037-1:$ rest-server-status


Node    Rest Server
----    -----------
1       running
2       running


RestAPI server current mode is legacy.
```

To switch to native mode, stop the REST API server, enable native mode, and then restart the REST API server, using the following commands:

```
G800-443037-1:$ rest-server-stop
The RestAPI server has been successfully stopped.
G800-443037-1:$ rest-server-mode --native
The RestAPI server mode change will not take affect until the RestAPI server is
restarted.
G800-443037-1:$ rest-server-start
The RestAPI server has been successfully started.
G800-443037-1:$ rest-server-status

RestAPI server is running
 supported API versions:
   07.2.1
RestAPI server current mode is native.
G800-443037-1:$
```

Authenticating with the REST API server

Credentials can be either a user/password combination or an API Key. Access via API Key is the preferred method, and keys can be generated as follows:

1. Log in to the HNAS CLI as an administrative user.

2. Create a new API key with the `apikey-create` command. API keys are created with full access to all APIs by default.

```
m1-merc-metro:$ apikey-create "nicktest"
Please make a note of this new API Key, as it is not possible to display the
full key again.
Only the prefix and description can be displayed in the future.

New key:  xIAdbgTNVP.Nj2TOgxiOYgpTu2kjzEGS4QmIJIeLmF3aXKg6FhY9vC
```

> **Note:**
>
> It is user's responsibility to store this created key to be used in all API requests. There is no way to recover the API key if it is lost except to create a new one.

3. To list, enable/disable or delete an API key, use the `apikey-list`, `apikey-update`, and `apikey-delete` commands. API keys can be restricted to read-only using the `apikey-update` command.

4. Subsequently, you can use the created API key to replace user password authentication.

   For example, suppose your API key is "`QA3RIPLHbk.B6.pfg8YCeeK07BhTy3UC48VlJ8teTg28nNvMWEHl77`". Then, you can use a custom HTTP header "`X-Api-Key: QA3RIPLHbk.B6.pfg8YCeeK07BhTy3UC48VlJ8teTg28nNvMWEHl77`" in your request for authentication. The API key is the recommended authentication method because it provides better security by enforcing authentication for each request. See more details at .

   > **Note:**
   >
   > Traditional user/password authentication is still supported for backward compatibility purposes. Create a new user name User1 with the password Password through the following CLI command:
   >
   > **`user add User1 Password SUPERVISOR`**
   >
   > To restrict any new users to read-only access, create them at "USER" level instead of "SUPERVISOR". This is only possible on HNAS systems and not VSP Unified systems.

   A user cannot be deleted unless and until all valid login sessions associated with that user have ended. The typical idle timeout session time is about 30 minutes.

Ports used by the REST API server

The REST API server uses the TCP port 8444 for HTTPS communication.

Hitachi file storage firmware requirements

The firmware requirements are the following.

- VSP Fx00 models, VSP Gx00 models, and Nx00 with NAS OS 14.2.7100 or higher

- HNAS with NAS OS 14.2.7100 or higher

Adding a file storage system

There are no specific REST API server configuration requirements for file storage other than the REST API server host having network connectivity to the file storage system.

Acquiring the correct admin EVS IP address

This section provides information about acquiring the correct admin Enterprise Virtual System (EVS) IP address.

📄 **Note:**

This IP is the IP of the REST API server in each API request.

Hitachi NAS Platform

When you are in the HNAS environment, the admin EVS IP address is required to invoke any file API. Do not confuse this management IP address with the system management unit (SMU) IP address.

Procedure

1. Log in to the NAS Manager (this is the GUI software within the HNAS product). From the server settings region click **EVS Management**. NAS Manager displays the EVS Management window:

2. In the **Type** column of the EVS list, identify the EVS that has the "admin services" type. This is the administrative EVS.

3. Examine the IP address of this EVS in the **First IP Address** column.

4. If the administrative EVS is a reachable public server, record the IP address in the **First IP Address** column and exit the EVS Management window. If the administrative EVS is not a reachable public server, click **Details** at the far right of the administrative server row to display the EVS Details window as shown in the following figure. Record the IP address of the reachable public server in the IP addresses list.

You now have a valid admin EVS IP address that you can use for HNAS API requests.

VSP N series and VSP Gx00 and Fx00 with NAS modules

To acquire an HNAS management IP or GUM IP address using Hitachi Device Manager - Storage Navigator to log into the service processor (SVP), perform the following steps.

Procedure

1. Log in as system administrator in Hitachi Device Manager - Storage Navigator. The URL for a system admin is different from the URL for a user.
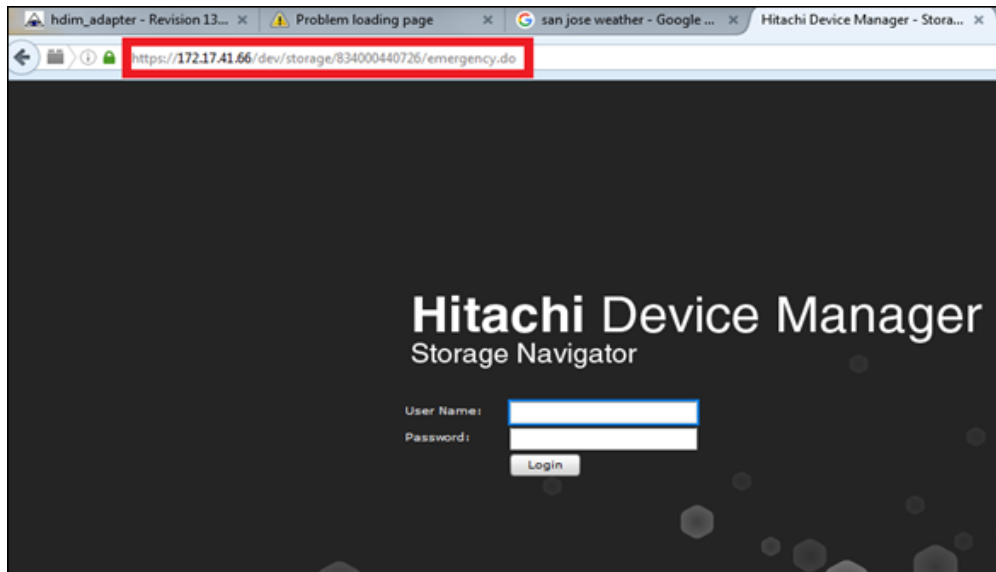
2. In SVP, look for the controller IP addresses to acquire the GUM IP addresses that will be used as the HNAS management IP address for the file API parameter. Note that either GUM IP address can be used for the file API to return the identical result. If any HNAS cluster node fails, the other GUM IP can be used as a backup.
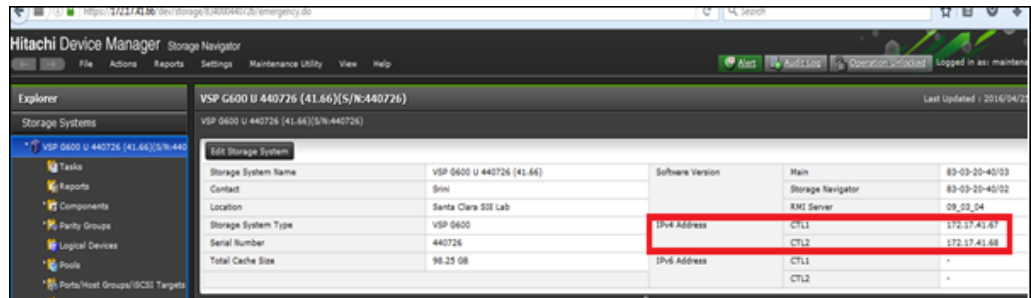
Chapter 2: API common specifications

The chapter describes various API topics such as deprecation and purging policy and an inventory of supported APIs.

Resource URI format

Resources can be accessed using the following URI formats.

- *<host>*:= IP address / host name of the REST API server.

- *<port>*:= port number of the REST API server.

- *<resources>*:= resources that are managed by the REST API server. They are always in plural format; for example, ports, logical-units.

- *<resource-id>*:= if specified, identifies the resource as managed by the REST API server. REST API server supports applicable create, read, update, and delete (CRUD) operations for each resource.

Example of a URI:

```
https://172.17.11.11:8444/v7/storage/filesystems/865A1E306B0D4EE70000000000000000
```

HTTP methods

The REST API server supports the following HTTP methods.

Operation	Guidelines
GET	Retrieves resource information.
POST	▪ Creates new resources. ▪ Performs actions on resources.
PUT/PATCH	Updates existing resource.
DELETE	Deletes a resource instance.

> **Note:** When the credentials supplied allow read-only access, only HTTP GET requests are supported.

Non-CRUD operations

Some storage actions cannot be completed using the create, read, update, and delete (CRUD) operations. In general, a POST operation is performed for non-CRUD actions on a resource. For example, to format a file system, the request URI is:

```
POST <Base_URI>/v7/storage/filesystems/{filesystemId}/format
```

The request body is:

```
{"blockSize":4}
```

Required custom HTTP headers

The following custom headers are required by all operations.

Name	Description
X-Api-Key	API key to access the storage system. This is the recommended authentication method that provides better security. To create and manage an API key, check details at API authentication.
(optional) X-Subsystem-User X-Subsystem-Password	User name and password to access the storage system. This method is considered optional. It works when the X-Api-Key header is not present. Users are encouraged to switch to the new API key method. The user password method is provided for backward compatibility purposes. When both API key and user/password headers are included, the user/password is ignored. Only one of these two methods is required.

Optional custom HTTP headers

The following custom headers are optional by all operations and the main purpose is for auditing and profiling.

Name	Description
X-Forwarded-for	IP of the REST API client. This is mainly for the purposes of auditing and profiling. Even though it is not required, users are encouraged to voluntarily provide their IP in this header.

Chapter 2: API common specifications

X-Fowarded-for-user	REST API user name. This is not necessarily the user used in authentication. It is only for the purposes of auditing and profiling.

HTTP status codes

The REST API server responds with the following status codes, which are defined in RFC 2616.

Status code	Status name	Description
200	OK	Resource retrieved successfully.
201	Created	Resource retrieved successfully.
204	Edited/Deleted	Resource edited or deleted successfully.
301	Move permanently	The API has been moved permanently.
303	See Other	The resource ID has been changed.
400	Bad Request	Missing or invalid request contents.
404	Not Found	Requested resource not found.
500	Internal Server Error	Internal error originated from storage sub-system or REST API server.
501	Not implemented	Server has not implemented the request operation on the resource.

Storage system error response

In cases when an error is thrown by the storage system, the REST API server displays the error code and error message in the response body as shown in the example below.

`errorDetail` shows low-level subsystem error details for debugging purposes.

```
{
  "errorCode": 1081419,
  "errorDetail": {
    "detail": "",
    "fault": "",
    "fileName": "RestApiQuotas.cpp",
    "function": "GetVirtualVolumeQuota",
    "lineNumber": 446,
    "message": "The quota does not exist",
    "reason": "The quota does not exist",
    "returnedValue": 0,
    "subCode": 2095
  },
  "errorMsg": "Quota not found"
}
```

Basic data types

The data types are JSON data types.

Data type	Definition
boolean	True or false
number	Signed/unsigned integer, fraction, exponent
string	Text
array	Array of JSON objects
object	Arbitrary JSON object

Inventory of supported APIs

A GET operation to https://*<host>*:*<port>/v7/* will return an HTML page that lists all supported REST API server operations.

```
curl -k -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/
```

Input and output format

The REST API server supports JSON input/output format.

As a general rule, if objects have an ObjectID associated with them, and they are to be used as input parameters, either the ID or the ObjectID will be accepted as the parameter. For example, any functions that require a virtualServerId parameter should accept a value of "1" or "313a3a3a3a3a3a3a303a3a3a4f49445f24232140255f56", as they represent the same object within the NAS system.

Where API calls require input parameters, in some cases, these can now also be supplied as query parameters, rather than needing to supply them within the request body. For example, the following two examples are equivalent:

```
curl -k -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/object-replications/last-report -X GET -d
'{"objectReplicationPolicyId":"ca2e73f2-d5d7-11d3-9cff-49e1bb864b23"}'
```

```
curl -k -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/object-replications/last-report?
objectReplicationPolicyId=ca2e73f2-d5d7-11d3-9cff-49e1bb864b23
```

This approach is only appropriate for simple input parameters, and should not be used for parameters that contain spaces, or other characters that need to be encoded or escaped within a URL, as they may produce unexpected results

Paging output

Output paging is supported for GET ALL API operations (for example, Get all file storage system statistics). It is possible to specify the page size (number of items) returned, and the page (an offset into the total results) that is to be returned. This functionality should be used if the expected number of items to be returned may be excessive for a single response.

The following examples demonstrate the use:

```
GET <base_URI>/v7/storage/statistics/CONTEXT_ALL?pageSize=4&page=1
```

The first example would return the first 4 items associated with the "CONTEXT_ALL" statistics. While the next example would return items 5 to 8 associated with the "CONTEXT_ALL" statistics.:

```
GET <base_URI>/v7/storage/statistics/CONTEXT_ALL?pageSize=4&page=2
```

Sort output

Sorting is controlled by the top-level scalar (of type boolean, number, or string) attributes. REST API server does not support either multi-level or nesting sort operations. REST API server only supports sorting for the GET ALL API operations (for example, Get all storage pools).

The format of ascending order sort has the following syntax:

```
sort=<attribute_name>
```

The format of descending order sort has the following syntax:

```
sort=-<attribute_name>
```

> **Note:** If output sorting is specified when output paging is being used, the sorting will only be applied to the returned results, and not to the full set of results.

Request

```
curl -vk -H "X-Api-Key:Td5qNSpXX4.732uVwjjuN1Wgmxw7yJwL5nygQk79k6pbVg.wvMFqH2"
https://172.17.23.10:8444/v7/storage/filesystems?sort=usedCapacity
```

Response

```
{
        "filesystems" :
        [
                {
                        "blockSize" : 0,
                        "capacity" : 38537265152,
                        "expansionLimits" : 18446744073709551615,
                        "filesystemId" : "3B7C8C94C8CC42790000000000000000",
                        "freeCapacity" : 0,
                        "isDedupeEnabled" : false,
                        "isDedupeSupported" : false,
                        "isLogicalCapacityFreeCapacityValid" : true,
                        "isLogicalCapacityValid" : true,
                        "isNDMPRecoveryTarget" : false,
                        "isNonStrictWORM" : false,
                        "isReadCached" : false,
                        "isReadOnly" : false,
                        "isSysLocked" : false,
                        "isThinProvisioningEnabled" : false,
                        "isThinProvisioningEnabledValid" : true,
                        "isTrueSparseFileEnabled" : false,
                        "isTrueSparseFileEnabledValid" : true,
                        "isUnlimitedExpansion" : true,
                        "isWORM" : false,
```

```
                    "label" : "xyz-createfs-0126-2",
                    "logicalCapacity" : 0,
                    "logicalFreeCapacity" : 0,
                    "objectId" :
"334237433384339344338434334323739303030303030303030303030303030303a3a3a303a3a3a4f494
45f24232140255f56",
                    "status" : "NOT_MOUNTED",
                    "storagePoolId" : 4286815075241083603,
                    "usedCapacity" : 0,
                    "virtualServerId" : 3
                },
                {
                    "blockSize" : 0,
                    "capacity" : 385670447104,
                    "expansionLimits" : 18446744073709551615,
                    "filesystemId" : "3B7C8CD40A200F630000000000000000",
                    "freeCapacity" : 0,
                    "isDedupeEnabled" : false,
                    "isDedupeSupported" : false,
                    "isLogicalCapacityFreeCapacityValid" : true,
"isLogicalCapacityValid" : true,
                    "isNDMPRecoveryTarget" : false,
                    "isNonStrictWORM" : false,
                    "isReadCached" : false,
                    "isReadOnly" : false,
                    "isSysLocked" : false,
                    "isThinProvisioningEnabled" : false,
                    "isThinProvisioningEnabledValid" : true,
                    "isTrueSparseFileEnabled" : false,
                    "isTrueSparseFileEnabledValid" : true,
                    "isUnlimitedExpansion" : true,
                    "isWORM" : false,
                    "label" : "xyz-createfs-0126",
                    "logicalCapacity" : 0,
                    "logicalFreeCapacity" : 0,
                    "objectId" :
"334237433384344343041323030463633303030303030303030303030303030303a3a3a303a3a3a4f494
45f24232140255f56",
                    "status" : "NOT_MOUNTED",
                    "storagePoolId" : 4286815075241083603,
                    "usedCapacity" : 0,
                    "virtualServerId" : 3
                },
                {
                    "blockSize" : 0,
                    "capacity" : 19268632576,
                    "expansionLimits" : 18446744073709551615,
                    "filesystemId" : "3B7CC79F0C7812B60000000000000000",
                    "freeCapacity" : 0,
                    "isDedupeEnabled" : false,
                    "isDedupeSupported" : false,
```

Chapter 2: API common specifications

```
                        "isLogicalCapacityFreeCapacityValid" : true,
                        "isLogicalCapacityValid" : true,
                        "isNDMPRecoveryTarget" : false,
                        "isNonStrictWORM" : false,
                        "isReadCached" : false,
                        "isReadOnly" : false,
                        "isSysLocked" : false,
                        "isThinProvisioningEnabled" : false,
                        "isThinProvisioningEnabledValid" : true,
                        "isTrueSparseFileEnabled" : false,
                        "isTrueSparseFileEnabledValid" : true,
                        "isUnlimitedExpansion" : true,
                        "isWORM" : false,
                        "label" : "xyz-create-fs-test3",
                        "logicalCapacity" : 0,
                        "logicalFreeCapacity" : 0,
                        "objectId" :
"334237434337394630433738313242363030303030303030303030303030303030303a3a3a303a3a3a4f494
45f24232140255f56",
                        "status" : "NOT_MOUNTED",
                        "storagePoolId" : 4286815075241083603,
                        "usedCapacity" : 0,
                        "virtualServerId" : 4
                },
                {
                        "blockSize" : 0,
                        "capacity" : 38465961984,
                        "expansionLimits" : 18446744073709551615,
                        "filesystemId" : "864C623AF020227C0000000000000000",
                        "freeCapacity" : 0,
                        "isDedupeEnabled" : false,
                        "isDedupeSupported" : false,
                        "isLogicalCapacityFreeCapacityValid" : true,
                        "isLogicalCapacityValid" : true,
                        "isNDMPRecoveryTarget" : false,
                        "isNonStrictWORM" : false,
                        "isReadCached" : false,
                        "isReadOnly" : false,
                        "isSysLocked" : false,
                        "isThinProvisioningEnabled" : false,
                        "isThinProvisioningEnabledValid" : true,
                        "isTrueSparseFileEnabled" : false,
                        "isTrueSparseFileEnabledValid" : true,
                        "isUnlimitedExpansion" : true,
                        "isWORM" : false,
                        "label" : "dontremove-fs-tieredpool",
                        "logicalCapacity" : 0,
                        "logicalFreeCapacity" : 0,
                        "objectId" :
"383634433632334146303230323237433030303030303030303030303030303030303a3a3a303a3a3a4f494
45f24232140255f56",
```

Chapter 2: API common specifications

```
                    "status" : "NOT_MOUNTED",
                    "storagePoolId" : 9678053102318196702,
                    "usedCapacity" : 0,
                    "virtualServerId" : 2
                }
        ]
}
```

Filter output

Filtering is used to narrow the result set, and can be used at the same time as result sorting.

REST API server supports only top-level scalar attributes. REST API server does not support multi-level or nesting filters. REST API server supports only GET ALL API operations. The following filter operators are supported: =, != <, >, >=, and <=. To filter multiple attributes, use the AND (&) operator to connect them.

Request

```
curl --insecure -H "X-Api-Key:
zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC" https://
172.17.23.10:8444/v7/storage/filesystems?usedCapacity>=39755415550
```

Response

```
HTTP/1.1 200 OK
{
        "filesystems" :
        [
                {
                        "blockSize" : 32768,
                        "capacity" : 1097095708672,
                        "expansionLimits" : 1099511627776,
                        "filesystemId" : "3B73894023F7F81B0000000000000000",
                        "freeCapacity" : 9733373952,
                        "isDedupeEnabled" : false,
                        "isDedupeSupported" : false,
                        "isLogicalCapacityFreeCapacityValid" : true,
                        "isLogicalCapacityValid" : true,
                        "isNDMPRecoveryTarget" : false,
                        "isNonStrictWORM" : false,
                        "isReadCached" : false,
                        "isReadOnly" : false,
                        "isSysLocked" : false,
                        "isThinProvisioningEnabled" : false,
                        "isThinProvisioningEnabledValid" : true,
                        "isTrueSparseFileEnabled" : true,
                        "isTrueSparseFileEnabledValid" : true,
                        "isUnlimitedExpansion" : false,
                        "isWORM" : false,
```

Chapter 2: API common specifications

```
                "label" : "ErinPerfFs",
                "logicalCapacity" : 1097095708672,
                "logicalFreeCapacity" : 9733373952,
                "objectId" :
"3342373333839343032334637463831423030303030303030303030303030303030303a3a3a303a3a3a4f49445
f24232140255f56",
                "status" : "MOUNTED",
                "storagePoolId" : 4283918657438155176,
                "usedCapacity" : 1087362334720,
                "virtualServerId" : 4
        },
        {
                "blockSize" : 32768,
                "capacity" : 192468221952,
                "expansionLimits" : 268435456000,
                "filesystemId" : "3B7D39EA82A4BA220000000000000000",
                "freeCapacity" : 46612545536,
                "isDedupeEnabled" : false,
                "isDedupeSupported" : false,
                "isLogicalCapacityFreeCapacityValid" : true,
                "isLogicalCapacityValid" : true,
                "isNDMPRecoveryTarget" : false,
                "isNonStrictWORM" : false,
                "isReadCached" : false,
                "isReadOnly" : false,
                "isSysLocked" : false,
                "isThinProvisioningEnabled" : false,
                "isThinProvisioningEnabledValid" : true,
                "isTrueSparseFileEnabled" : true,
                "isTrueSparseFileEnabledValid" : true,
                "isUnlimitedExpansion" : false,
                "isWORM" : false,
                "label" : "xyzDevProdFS",
                "logicalCapacity" : 192468221952,
                "logicalFreeCapacity" : 46612545536,
                "objectId" :
"3342374433339454138324134424132323030303030303030303030303030303030303a3a3a303a3a3a4f49445
f24232140255f56",
                "status" : "MOUNTED",
                "storagePoolId" : 4286815075241083603,
                "usedCapacity" : 145855676416,
                "virtualServerId" : 4
        },

        {
                "blockSize" : 32768,
                "capacity" : 58518929408,
                "expansionLimits" : 107374182400,
                "filesystemId" : "864B1228ABB144AC0000000000000000",
                "freeCapacity" : 18763513856,
                "isDedupeEnabled" : true,
```

Chapter 2: API common specifications

```
                     "isDedupeSupported" : true,
                     "isLogicalCapacityFreeCapacityValid" : true,
                     "isLogicalCapacityValid" : true,
                     "isNDMPRecoveryTarget" : false,
                     "isNonStrictWORM" : false,
                     "isReadCached" : false,
                     "isReadOnly" : false,
                     "isSysLocked" : true,
                     "isThinProvisioningEnabled" : false,
                     "isThinProvisioningEnabledValid" : true,
                     "isTrueSparseFileEnabled" : false,
                     "isTrueSparseFileEnabledValid" : true,
                     "isUnlimitedExpansion" : false,
                     "isWORM" : false,
                     "label" : "xyz-obj-rep-target-new",
                     "logicalCapacity" : 0,
                     "logicalFreeCapacity" : 0,
                     "objectId" :
"383634423132323841424231343441433030303030303030303030303030303030303a3a3a303a3a3a4f49445
f24232140255f56",
                     "status" : "MOUNTED",
                     "storagePoolId" : 9676880324761524863,
                     "usedCapacity" : 39755415552,
                     "virtualServerId" : 3
             }
     ]
}
```

Output embedded

REST API server supports embedding for single-level associations.

For example, system drives have associations with a storage pool (as shown in the following example).

📄 **Note:** Without the embed keyword or with the notation embed=false, the output does not display data associated with the systemDrives keyword). REST API server supports embedding only for the GET API operation of a single resource (for example, a storage pool).

Request

```
curl --insecure -H "X-Api-Key:
zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC" https://
172.43.12.33:8444/v7/storage/filesystems/
383634423132323841424231343441433030303030303030303030303030303030303a3a3a303a3
a3a4f49445f24232140255f56?embed=true
```

Response

```
HTTP/1.1 200 OK
{
   "filesystem" :
        {
                "blockSize" : 32768,
                "capacity" : 58518929408,
                "expansionLimits" : 107374182400,
                "filesystemId" : "864B1228ABB144AC0000000000000000",
                "freeCapacity" : 18763186176,
                "isDedupeEnabled" : true,
                "isDedupeSupported" : true,
                "isLogicalCapacityFreeCapacityValid" : true,
                "isLogicalCapacityValid" : true,
                "isNDMPRecoveryTarget" : false,
                "isNonStrictWORM" : false,
                "isReadCached" : false,
                "isReadOnly" : false,
                "isSysLocked" : true,
                "isThinProvisioningEnabled" : false,
                "isThinProvisioningEnabledValid" : true,
                "isTrueSparseFileEnabled" : false,
                "isTrueSparseFileEnabledValid" : true,
                "isUnlimitedExpansion" : false,
                "isWORM" : false,
                "label" : "xyz-obj-rep-target-new",
                "logicalCapacity" : 0,
                "logicalFreeCapacity" : 0,
                "objectId" :
"38363442313232384142423134344143303030303030303030303030303030303a3a3a303a3a3a4f49445
f24232140255f56",
                "status" : "MOUNTED",
                "storagePoolId" : 9676880324761524863,
                "storagePools" :
                [
                        {
                                "chunkSize" : 19327352832,
                                "freeCapacity" : 175728754688,
                                "isAssignedToLocalCluster" : true,
                                "isFilesystemExpansionAllowed" : true,
                                "isHealthy" : true,
                                "isTiered" : false,
                                "label" : "historDevStoragePool",
                                "objectId" :
"39363736383830333234373631353234383633333a3a3a3a3a3a303a3a3a4f49445f24232140255f56",
                                "storagePoolId" : 9676880324761524863,
                                "totalCapacity" : 429492535296,
                                "usedCapacity" : 253763780608
                        }
                ],
```

```
            "usedCapacity" : 39755743232,
            "virtualServerId" : 3,
            "virtualServers" :
            [
                    {
                            "UUID" : "b70d82b2-0386-11d0-9043-49e1bb864b23",
                            "ipAddresses" :
                            [
                                    "172.17.58.122"
                            ],
                            "isEnabled" : true,
                            "name" : "xyz-evs-58-122",
                            "objectId" :
"333a3a3a3a3a3a303a3a3a4f49445f24232140255f56",
                            "status" : "ONLINE",
                            "type" : "File services",
                            "virtualServerId" : 3
                    }
            ]
    }
}
```

Chapter 3: Device resource

The device resource enables you to obtain important information about the storage system. Key information includes the cluster unique identifier, number of nodes, and the firmware version. In addition, the resource enables you to retrieve a list of licenses applied on the storage system.

Device object model

The device object model describing this resource contains the following objects.

route

Attribute	JSON Type	Data Type	Description
MTU	number	uint	MTU of the route.
destinaton	string	string	Route destination IP address.
mask	string	string	IP address mask.
flags	number	ushort	Flags of the route.
gateway	string	string	IP address of the route gateway.
isClusterNodeRoute	boolean	boolean	Whether the cluster node route has been enable.d
isStaticRoute	boolean	boolean	Whether the route is static.
nodeId	number	int	Cluster node ID. When route by cluster node is not enabled, it is always 0.
objectId	string	string	Unique object ID which assigned by the web services. This ID is not storage identifier.
type	string	string	Route type. Valid values are: ▪ ROUTE_TYPE_HOST ▪ ROUTE_TYPE_NETWORK ▪ ROUTE_TYPE_GATEWAY

fileStorageDeviceInfo

Attribute	JSON Type	Data Type	Description
name	string	string	Device name.
vendor	string	string	Vendor name.
isCluster	boolean	boolean	True for an HNAS cluster.
node count	number	int	Number of nodes in a cluster.
storageHealth	string	string	Health of the cluster. Possible values are: ▪ ROBUST ▪ DEGRADED ▪ CRITICAL ▪ UNKNOWN
clusterUUID	string	string	HNAS cluster UUID.
model	string	string	HNAS model.
firmwareVersion	string	string	HNAS firmware version.
licenses	array	array	Contains the names of the licenses applied for this particular device. Possible values are: ▪ CIFS ▪ NFS ▪ SFM ▪ WORM ▪ ISCSI ▪ DATA_MIGRATOR ▪ CLUSTER ▪ TB ▪ FS_ROLLBACK ▪ CNS ▪ READ_CACHE ▪ HDS ▪ EVS_SECURITY ▪ EVS

Attribute	JSON Type	Data Type	Description
			▪ METRO_CLUSTER
			▪ REPLICATION
			▪ XVL
			▪ UNKNOWN
			▪ FS_RECOVER_FROM_SNAP
			▪ MACCRC
			▪ HSR
			▪ FILE_CLONE
			▪ BLUEARC_RS
			▪ PERF_ACCELERATOR
			▪ BASE_DEDUPLICATION
			▪ PREMIUM_DEDUPLICATION
			▪ MODEL_TYPE
			▪ EXT_PACK_SECURE_FTP
			▪ UNIVERSAL_NAS_VIRTUAL_CAPACITY

ethernetInterface

Attribute	JSON type	Data type	Description
isAdminVirtualServerIpAllowed	Boolean	Boolean	True if the admin virtual server IP address is allowed; false otherwise.
isAdvancedIpConfigAllowed	Boolean	Boolean	True if the advanced IP address configuration is allowed; false otherwise.
isAggregationAllowed	Boolean	Boolean	True if aggregation is allowed; false otherwise.
isClusterNodeIpAllowed	Boolean	Boolean	True if the cluster node address is allowed; false otherwise.
isVirtualServerIpAllowed	Boolean	Boolean	True if the virtual server IP address is allowed; false otherwise.

Chapter 3: Device resource

Attribute	JSON type	Data type	Description
name	string	string	Name of the Ethernet interface.

Get a file device

Retrieves basic information about the storage system.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/file-devices
```

Return codes

Code	Data	Description
200	fileStorageDeviceInfo	Device information retrieved successfully.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/file-devices
```

Response example

```
HTTP/1.1 200 OK
   {
   "clusterUUID" : "2bb93988-246e-11cf-9000-5ec00375458d",
   "firmwareVersion" : "13.1.4416.00",
   "isCluster" : true,
   "licenses" : [
      "CIFS",
      "NFS",
      "SFM",
      "WORM",
      "ISCI",
      "DATA_MIGRATOR",
      "CLUSTER",
```

Chapter 3: Device resource

```
        "TB",
        "CNS",
        "HDS",
        "EVS_SECURITY",
        "EVS",
        "REPLICATION",
        "XVL",
        "FILE_CLONE",
        "PERF_ACCELERATOR",
        "BASE_DEDUPLICATION"
    ],
    "model" : "HNAS 4060",
    "name" : "Gizmo-CLUSTER",
    "nodeCount" : 2,
    "storageHealth" : "CRITICAL",
    "vendor" : "HITACHI"
}
```

Get network routes

Retrieves all network routes.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/file-devices/network-routes
```

Return codes

Code	Data	Description
200	routes	Array of route objects.
400	No Data	Missing or invalid request contents.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/file-devices/network-routes
```

Response example

```
HTTP/1.1 200 OK
{
   "routes" : [
      {
         "MTU" : 0,
         "destination" : "1.2.3.4/32",
         "flags" : 68,
         "gateway" : "1.2.3.4",
         "isClusterNodeRoute" : false,
         "isStaticRoute" : true,
         "nodeId" : 0,
         "objectId" : "486f73743a3a3a312e322e332e343a3a3a312e322e332e342f3332",
         "type" : "Host"
      },
      {
         "MTU" : 0,
         "destination" : "172.17.41.0/32",
         "flags" : 68,
         "gateway" : "172.17.239.1",
         "isClusterNodeRoute" : false,
         "isStaticRoute" : true,
         "nodeId" : 0,
         "objectId" :
"486f73743a3a3a3137322e31372e3233392e313a3a3a3137322e31372e34312e302f3332",
         "type" : "Host"
      }
   ]
}
```

Get a network route

Retrieves a network route.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/file-devices/network-routes/{routeObjectId}
```

Parameters

Name	Type	Required	Values	Description
routeObjectId	URI_PARAM	Y	string	Route object ID.

Return codes

Code	Data	Description
200	route	Route object successfully retrieved.
400	No Data	Missing or invalid request contents.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/file-devices/network-routes/
486f73743a3a3a312e322e332e343a3a3a312e322e332e342f3332
```

Response example

```
HTTP/1.1 200 OK
{
    "route" : {
        "MTU" : 0,
        "destination" : "1.2.3.4/32",
        "flags" : 68,
        "gateway" : "1.2.3.4",
        "isClusterNodeRoute" : false,
        "isStaticRoute" : true,
        "nodeId" : 0,
        "objectId" : "486f73743a3a3a312e322e332e343a3a3a312e322e332e342f3332",
        "type" : "Host"
    }
}
```

Create a network route

Creates a network route.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/file-devices/network-routes
```

Parameters

Name	Type	Required	Values	Description
destination	string	Y	string	Destination IP address.
gateway	string	Y	string	Gateway IP address.
type	string	Y	string	Route type. Valid values are: ▪ ROUTE_TYPE_HOST ▪ ROUTE_TYPE_NETWO RK ▪ ROUTE_TYPE_GATEWA Y **Note:** For more details, see https:// knowledge.hitachivantara.co m/Documents/Storage/ NAS_Platform.
mask	string	Y	string	IP address mask. Valid values are: ▪ ROUTE_MASK_24 ▪ ROUTE_MASK_25 ▪ ROUTE_MASK_26 ▪ ROUTE_MASK_27 ▪ ROUTE_MASK_28 ▪ ROUTE_MASK_29 ▪ ROUTE_MASK_30 ▪ ROUTE_MASK_31 ▪ ROUTE_MASK_32 **Note:** For more details, see https:// knowledge.hitachivantara.co m/Documents/Storage/ NAS_Platform.

Return codes

Code	Data	Description
201	route	Route object successfully created.

Chapter 3: Device resource

Code	Data	Description
400	No Data	Missing or invalid request contents.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 201 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/file-devices/network-routes -d
'{ "destination":"1.2.3.4", "mask":"ROUTE_MASK_30",
"type":"ROUTE_TYPE_HOST", "gateway":"1.2.3.4"}' -X POST
```

Response example

```
HTTP/1.1 201 Created
{
    "route" : {
        "MTU" : 0,
        "destination" : "1.2.3.4/32",
        "flags" : 68,
        "gateway" : "1.2.3.4",
        "isClusterNodeRoute" : false,
        "isStaticRoute" : true,
        "nodeId" : 0,
        "objectId" : "486f73743a3a3a312e322e332e343a3a3a312e322e332e342f3332",
        "type" : "Host"
    },
    "uri" : "https://172.17.58.117:8444/v7/storage/file-devices/network-routes/
486f73743a3a3a312e322e332e343a3a3a312e322e332e342f3332"
}
```

Delete a network route

Deletes a network route.

HTTP request syntax (URI)

```
DELETE <base_URI>/v7/storage/file-devices/network-routes/{routeObjectId}
```

Chapter 3: Device resource

Parameters

Name	Type	Required	Values	Description
routeObjectId	URI_PARAM	Y	string	Route object ID.

Return codes

Code	Data	Description
204	No Data	Route object successfully deleted.
400	No Data	Missing or invalid request contents.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/file-devices/network-routes/
486f73743a3a3a312e322e332e343a3a3a312e322e332e342f3332 -X DELETE
```

Response example

```
HTTP/1.1 204 No Content
```

Flush network routes

Flushes all network routes.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/file-devices/network-routes/flush
```

Return codes

Code	Data	Description
204	No Data	No Data.

Code	Data	Description
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/file-devices/network-routes/flush -X POST
```

Response example

```
HTTP/1.1 204 No Content
```

Get Ethernet interfaces

Gets all Ethernet interfaces.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/file-devices/ethernet-interfaces
```

Return codes

Code	Data	Description
200	ethernetInterfaces	Ethernet interfaces retrieved successfully.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/file-devices/ethernet-interfaces
```

Response example

```
HTTP/1.1 200 OK
{
  "ethernetInterfaces" : [
    {
        "isAdminVirtualServerIpAllowed" : true,
        "isAdvancedIpConfigAllowed" : true,
        "isAggregationAllowed" : false,
        "isClusterNodeIpAllowed" : false,
        "isVirtualServerIpAllowed" : true,
        "name" : "ag1"
        },
        {
        "isAdminVirtualServerIpAllowed" : false,
        "isAdvancedIpConfigAllowed" : false,
        "isAggregationAllowed" : false,
        "isClusterNodeIpAllowed" : false,
        "isVirtualServerIpAllowed" : false,
        "name" : "c1"
        }
                    ]
}
```

Chapter 4: File storage statistics resource

The file storage statistics resource represents various file storage statistical objects. The genericStatistics object is a nested data structure to represent a generic statistics object. Note that most statistics reports activity for a preset time interval. This is the interval of time between the reset time and the current timestamp. Each statistical type may have its own reset time point. The storageStatistics object is a simple data structure that describes statistics data about a file storage.

File storage statistics object model

The file storage statistics object model contains the following objects.

genericStatistics

Attribute	JSON Type	Data Type	Description
errorMessage	string	string	Error message from HNAS.
resetTime	number	int64	Unix time of last reset time for measurement.
statistics	array	array	Array of statistics objects.

statistic

Attribute	JSON Type	Data Type	Description
label	string	string	Name of the statistics group.
resetTime	number	int64	Unix time of last reset time for measurement.
elements	array	array	Array of element objects.
statsType	string	string	Type of statistics data to retrieve.

element

Attribute	JSON Type	Data Type	Description
text	string	string	Name of the statistics element.
value	string	string	Value of the statistics element.

storageStatistics

Attribute	JSON Type	Data Type	Description
nodeUUID	string	string	UUID of the cluster node.
name	string	string	The name of the statistical data.
value	number	uint	The value of the statistics. Depending on the type of statistics, this number can be in any units.

Get file storage node statistics

Retrieves statistics data for file storage from a specific node. Some statistics data is node-specific. For other types of statistics data, the node parameter is ignored.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/statistics/{id}/{statsType}
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string/number	Node object ID or storage node ID. If the statsType parameter is "Virus", then the id parameter is used to specify a virtual server ID and not the node ID

Name	Type	Required	Values	Description
statsType	URI_PARAM	Y	string	Type of statistics data to retrieve. Possible values are: ▪ CIFS ▪ iSCSI ▪ NFS ▪ NVRAM ▪ TCPIP ▪ DetailedTCPIP ▪ TCPIPPerPort ▪ FTP ▪ SNMP ▪ Virus ▪ FibreChannel ▪ FibreChannelPerPort ▪ EtherNet ▪ EtherNetPerPort ▪ HTTPSManagement ▪ SSCManagement ▪ VSSManagement ▪ ReadCache ▪ FileSystemOpsPerSec

Return codes

Code	Data	Description
200	genericStatistics	A genericStatistics object is retrieved successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.

Chapter 4: File storage statistics resource

Code	Data	Description
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example 1

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/statistics/313a3a3a/nfs
```

Response example 1

```
HTTP/1.1 200 OK
{
   "genericStatistics" : {
      "errorMessage" : "",
      "resetTime" : 1462000840,
      "statistics" : [
         {
            "elements" : [
               {
                  "text" : "Null",
                  "value" : "0"
               },
               {
                  "text" : "GetAttr",
                  "value" : "0"
               },
               {
                  "text" : "SetAttr",
                  "value" : "0"
               },
               {
                  "text" : "Link",
                  "value" : "0"
               },
               {
                  "text" : "SymLink",
                  "value" : "0"
               },
               {
                  "text" : "MkDir",
                  "value" : "0"
               },
               {
                  "text" : "RmDir",
```

```
            "value" : "0"
          },
          {
            "text" : "ReadDir",
            "value" : "0"
          },
          {
            "text" : "StatFS/FSStat",
            "value" : "0"
          }
        ],
        "label" : "Version 2",
        "resetTime" : 0
      },
      {
        "elements" : [
          {
            "text" : "Null",
            "value" : "1"
          },
          {
            "text" : "GetAttr",
            "value" : "20"
          }
        ],
        "label" : "Version 4",
        "resetTime" : 0
      }
    ]
  }
}
```

Request example 2

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/statistics/313a3a3a/nvram
```

Response example 2

```
HTTP/1.1 200 OK
    {
  "genericStatistics" : {
    "errorMessage" : "",
    "resetTime" : 0,
    "statistics" : [
      {
        "elements" : [
          {
            "text" : "NVRAM size",
            "value" : "3.981 GB"
```

Chapter 4: File storage statistics resource

```
          },
          {
            "text" : "Maximum used",
            "value" : "48 MB"
          },
          {
            "text" : "Currently in use",
            "value" : "48 MB"
          }
        ],
        "label" : "",
        "resetTime" : 0
      }
    ]
  }
}
```

Get file storage system statistics

Retrieves system statistics data for file storage.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/statistics/{systemStatsType}
```

Parameters

Name	Type	Required	Values	Description
systemStatsType	URI_PARAM	Y	string	It can be either an arbitrary filter string or the following pre-defined type of statistics data to retrieve. In case of an arbitrary string, certain restrictions such as minimum length and excluding some special characters will apply(see example).

Name	Type	Required	Values	Description
				Possible pre-defined values are: ▪ TOTAL_OPERATION_PER_SEC ▪ FILESYSTEM_OPERATIONS_VALUE ▪ NVRAM_WAITED_ALLOCS ▪ FSB_FILESYSTEM_LOAD ▪ FSA_FILESYSTEM_LOAD ▪ FSI_CACHE_USAGE ▪ HEAP_USAGE ▪ FIBRECHANNEL_THROUGHPUT_RX ▪ FIBRECHANNEL_THROUGHPUT_TX ▪ ETHERNET_THROUGHPUT_RX ▪ ETHERNET_THROUGHPUT_TX ▪ MFB_LOAD ▪ MMB_LOAD ▪ DISK_READ_LATENCY ▪ DISK_WRITE_LATENCY ▪ DISK_STRIPE_WRITE_LATENCY ▪ RUNNING_BOSSOCK_FIBRES ▪ PI_TCP_SOCKETS_RECEIVE_FIBRES ▪ CONTEXT_NODE

Chapter 4: File storage statistics resource

Name	Type	Required	Value s	Description
				▪ CONTEXT_FILESY STEMS ▪ CONTEXT_ALL

Return codes

Code	Data	Description
200	storageStatistics	A storageStatistics object is retrieved successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example using a pre-defined search string

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.239.120:8444/v7/storage/statistics/HEAP_USAGE
```

Response example using a pre-defined search string

```
HTTP/1.1 200 OK
        {
        "storageStatistics" : [
         {
                "name" : "Heap Usage (%)",
                "nodeUUID" : "328511ce-680f-11d1-9001-040400070206",
                "value" : 62
                },
                {
                "name" : "Heap Usage (%)",
                "nodeUUID" : "31533498-680f-11d1-9001-8188be748157",
                "value" : 62
          }
          ]
       }
```

Chapter 4: File storage statistics resource

Request example using an arbitrary search string "111"

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.239.120:8444/v7/storage/statistics/111
```

Response example using an arbitrary search string "111"

```
HTTP/1.1 200 OK
{
    "storageStatistics" : [
        {
            "name" : "LongTermSizeBasedPool-4398046511120: pool current",
            "nodeUUID" : "dc1da44a-f532-11d4-9001-11b384af0bfe",
            "value" : 0
        },
        {
            "name" : "LongTermSizeBasedPool-4398046511120: pool hwm since reboot",
            "nodeUUID" : "dc1da44a-f532-11d4-9001-11b384af0bfe",
            "value" : 0
        }
    ]
}
```

Get multiple file storage system statistics

Allows multiple search queries to be specified when retrieving file storage system statistics. This API call works in the same way as "Get file storage system statistics", but takes a list of query string values, allowing more statistics to be returned in a single API call.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/statistics
```

Parameters

Name	Type	Required	Values	Description
systemStatsType	BODY	Y	array	Each array item can be either an arbitrary filter string or the following pre-defined type of statistics data to retrieve

Name	Type	Required	Values	Description
				In case of an arbitrary string, certain restrictions such as minimum length and excluding some special characters will apply(see example).
				Possible pre-defined values are:
				▪ TOTAL_OPERATION_PER_SEC
				▪ FILESYSTEM_OPERATIONS_VALUE
				▪ NVRAM_WAITED_ALLOCS
				▪ FSB_FILESYSTEM_LOAD
				▪ FSA_FILESYSTEM_LOAD
				▪ FSI_CACHE_USAGE
				▪ HEAP_USAGE
				▪ FIBRECHANNEL_THROUGHPUT_RX
				▪ FIBRECHANNEL_THROUGHPUT_TX
				▪ ETHERNET_THROUGHPUT_RX
				▪ ETHERNET_THROUGHPUT_TX
				▪ MFB_LOAD

Name	Type	Required	Value s	Description
				▪ MMB_LOAD ▪ DISK_READ_LATE

Return codes

Code	Data	Description
200	storageStatistics	A storageStatistics object is retrieved successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example using multiple search strings

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.239.120:8444/v7/storage/statistics -X GET -d '{"systemStatsType":
["HEAP_USAGE", "MFB_LOAD", "111"]}'
```

Response example using multiple search strings

```
HTTP/1.1 200 OK
{
  "storageStatistics": [
    {
      "name": "Heap Usage (%)",
      "nodeUUID": "48fbe624-4c33-11d0-9001-9c5547075e75",
      "value": 77
    },
    {
      "name": "LongTermSizeBasedPool-4398046511120: pool current",
      "nodeUUID": "48fbe624-4c33-11d0-9001-9c5547075e75",
      "value": 0
    },
    {
      "name": "LongTermSizeBasedPool-4398046511120: pool hwm since reboot",
      "nodeUUID": "48fbe624-4c33-11d0-9001-9c5547075e75",
      "value": 0
```

```
    },
    {
      "name": "MFB Load (%)",
      "nodeUUID": "48fbe624-4c33-11d0-9001-9c5547075e75",
      "value": 3
    }
  ]
}
```

Chapter 5: File system resource

The file system resource enables you to obtain detailed information about the HNAS file system on the storage system. A file system is a namespace in which NFS/CIFS exports may be created and accessed. In addition, it allows you to perform key management operations related to file systems, such as creation, deletion, mount, and unmount operations.

File system object model

The file system object model contains the following objects.

filesystem

Attribute	JSON Type	Data Type	Description
objectId	string	string	File system object unique identifier. This ID is not the HNAS storage file system ID.
filesystemId	string	string	HNAS file system ID.
status	string	string	Status of the file system. Current supported values are: ▪ CHECKING ▪ FAILING ▪ FIXING ▪ MOUNTING ▪ FORMATTING ▪ MOUNTED ▪ NOT_MOUNTED ▪ NOT_FOUND ▪ VOLUME_NOT_ AVAILABLE_TO_ BS

Attribute	JSON Type	Data Type	Description
label	string	string	Label of the file system.
blockSize	number	uint64	Block size of the file system in bytes.
capacity	number	uin64	Capacity of the file system in bytes.
usedCapacity	number	uint64	Used capacity in bytes.
freeCapacity	number	uint64	Free capacity in bytes.
expansionLimits	number	uint64	Expansion limits in bytes. The expansion limit is unlimited if the value is 0xFFFFFFFFFFFFFFFF.
isDedupeEnabled	boolean	boolean	True if dedupe is enabled on the file system.
isDedupeSupported	boolean	boolean	True if dedupe is supported on the file system.
isNDMPRecoveryTarget	boolean	boolean	True if the file system is an NDMP target.
isNonStrictWORM	boolean	boolean	True if the file system is non-strict write once read many.
isReadCached	boolean	boolean	True if the file system is a read cache.
isObjectReplicationTarget	boolean	boolean	True if the filesystem is object replication target.
isReadOnly	boolean	boolean	True if the file system is read only.

Chapter 5: File system resource

Attribute	JSON Type	Data Type	Description
isSysLocked	boolean	boolean	True if the file system is sys locked.
isWORM	boolean	boolean	True if the file system is write once read many.
isUnlimitedExpansion	boolean	boolean	True if the file system is set for unlimited expansion
isLogicalCapacityFreeCapacityValid	boolean	boolean	True if the logical free capacity is valid.
isLogicalCapacityValid	boolean	boolean	True if the logical capacity is valid.
isThinProvisioningEnabled	boolean	boolean	True if thin provisioning is enabled.
isThinProvisioningEnabledValid	boolean	boolean	True if thin provisioning enabled is valid.
isTrueSparseFileEnabled	boolean	boolean	True if true sparse file is enabled.
isTrueSparseFileEnabledValid	boolean	boolean	True if true sparse file enabled is valid.
logicalCapacity	number	uint64	Logical capacity of the file system in bytes.
logicalFreeCapacity	number	uint64	Logical free capacity of the file system in bytes.
storagePoolId	number	uint64	Storage pool ID of the file system
virtualServerId	number	integer	Virtual server ID of the file system

Attribute	JSON Type	Data Type	Description
tierInformation	object	array	Array of tierInformation objects related to the file system, if the file system is made up from multiple tiers. Empty if file system is only a single tier.
isTiered	boolean	boolean	True if the file system is made up of multiple tiers.

tierinformation

Attribute	JSON Type	Data Type	Description
tierNumber	number	integer	Tier number.
tierCapacity	number	uint64	Capacity of the tier in bytes.
tierFreeSpace	number	uint64	Free capacity of the tier in bytes.
tierExpansionLimit	number	uint64	Expansion limit of the tier in bytes. The expansion limit is unlimited if the value is 0xFFFFFFFFFFFFFFFF.
isUnlimitedExpansion	boolean	boolean	True if the tier is set for unlimited expansion.

storagePool

Attribute	JSON Type	Data Type	Description
objectId	string	string	Storage pool unique identifier. This ID is not the HNAS storage pool ID.
storagePoolId	number	unit64	HNAS storage pool ID.
label	string	string	Storage pool label.
totalCapacity	number	uint64	Total capacity in bytes.

Chapter 5: File system resource

Attribute	JSON Type	Data Type	Description
usedCapacity	number	uint64	Used capacity in bytes.
freeCapacity	number	uint64	Free capacity in bytes.
chunkSize	number	uint64	Chunk size in bytes.
isHealthy	boolean	boolean	True if the storage pool is in a healthy condition.
isTiered	boolean	boolean	True if the storage pool is tiered.
isFilesystemExpansionAllowed	boolean	boolean	True if allowed.
isAssignedToLocalCluster	boolean	boolean	True if assigned.
tierInformation	object	array	Array of tierInformation objects related to the storage pool, if the storage pool is made up from multiple tiers. Empty if storage pool is only a single tier.

dedupFilesystem

Attribute	JSON Type	Data Type	Description
filesystemId	string	string	HNAS dedup file system ID.
lastRun	number	integer	Number of seconds since the last time deduplication service started running
percentageReclaimed	number	integer	Percentage of disk space reclaimed by the deduplication service
reclaimedCapacity	number	integer	Size of reclaimed capacity in bytes.
isEnabled	boolean	boolean	True if the dedup file system is enabled; False otherwise.
virtualServerId	number	integer	ID of virtual server of the dedup file system.

replicationSnapshotDetail

Attribute	JSON Type	Data Type	Description
replicationSnapshotStatus	string	string	Status of replication snapshot. Current supported values: ▪ INVALID ▪ NON_REPLICATION_TARGET ▪ REPLICATION_TARGET_INCOMPLETE ▪ REPLICATION_TARGET_COMPLETE ▪ UNKNOWN
replicationType	string	string	Replication type. Current supported values: ▪ NONE ▪ INCREMENTAL ▪ FULL ▪ UNKNOWN
sourceFilesystemId	string	string	Source file system ID.
sourceSnapshotName	string	string	Source snapshot name.
sourceSnapshotTime	number	integer	Source snapshot time.
targetFilesystemId	string	string	Target file system.
targetSnapshotName	string	string	Target snapshot name.

filesystemQuota

Attribute	JSON Type	Data Type	Description
virtualServerId	number	uint	Virtual server ID of the quota object.
filesystemId	string	string	File system ID of the virtual volume.
quota	object	object	Object that describes the quota setup.

filesystemDRStateChangeReport

Attribute	JSON Type	Data Type	Description
endTime	number	integer	End time.
fileSystemId	string	string	File system ID.
isActive	boolean	boolean	Disaster recovery state change report is active or not.
logName	string	string	File name of the report.
startTime	number	integer	Start time.
statistics	object	array	List of key-value pairs of the statistics report.
status	string	string	Status of DR state change.
transitionId	string	string	ID of DR state transition.

Get file systems

Retrieves file systems on the managed storage system.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/filesystems
```

Return codes

Code	Data	Description
200	filesystem	Array of file systems retrieved successfully.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems
```

Response example

```
HTTP/1.1 200 OK
                {
                "filesystems" :
                [
                    {
                        "blockSize" : 32768,
                        "capacity" : 19327352832,
                        "expansionLimits" : 18446744073709551615,
                        "filesystemId" : "3B510107663A94D30000000000000000",
                        "freeCapacity" : 16794255360,
                        "isDedupeEnabled" : false,
                        "isDedupeSupported" : false,
                        "isLogicalCapacityFreeCapacityValid" : true,
                        "isLogicalCapacityValid" : true,
                        "isNDMPRecoveryTarget" : false,
                        "isNonStrictWORM" : false,
                        "isObjectReplicationTarget" : true
                        "isReadCached" : false,
                        "isReadOnly" : false,
                        "isSysLocked" : false,
                        "isThinProvisioningEnabled" : false,
                        "isThinProvisioningEnabledValid" : true,
                        "isTrueSparseFileEnabled" : false,
                        "isTrueSparseFileEnabledValid" : true,
                        "isUnlimitedExpansion" : true,
                        "isWORM" : false,
                        "label" : "testC19847",
                        "logicalCapacity" : 0,
                        "logicalFreeCapacity" : 0,
                        "objectId" :
"334235313031303736363341393444333030303030303030303030303030303030303a3a3a303a3a3a4f49445
f24232140255f56",
                        "status" : "NOT_MOUNTED",
                        "storagePoolId" : 9677726511474388510,
                        "usedCapacity" : 2533097472,
                        "virtualServerId" : 2
                    },
                    {
                        "blockSize" : 32768,
                        "capacity" : 19327352832,
                        "expansionLimits" : 18446744073709551615,
                        "filesystemId" : "3B51199C3C1B8C7A0000000000000000",
                        "freeCapacity" : 16666853376,
                        "isDedupeEnabled" : false,
                        "isDedupeSupported" : false,
                        "isLogicalCapacityFreeCapacityValid" : true,
                        "isLogicalCapacityValid" : true,
                        "isNDMPRecoveryTarget" : false,
                        "isNonStrictWORM" : false,
```

Chapter 5: File system resource

```
                        "isObjectReplicationTarget" : false,
                        "isReadCached" : false,
                        "isReadOnly" : false,
                        "isSysLocked" : false,
                        "isThinProvisioningEnabled" : false,
                        "isThinProvisioningEnabledValid" : true,
                        "isTrueSparseFileEnabled" : true,
                        "isTrueSparseFileEnabledValid" : true,
                        "isUnlimitedExpansion" : true,
                        "isWORM" : false,
                        "label" : "rtat_concur_API_FS",
                        "logicalCapacity" : 19327352832,
                        "logicalFreeCapacity" : 16666853376,
                        "objectId" :
"3342353131393943333433314238433741303030303030303030303030303030303a3a3a303a3a3a4f49445
f24232140255f56",
                        "status" : "MOUNTED",
                        "storagePoolId" : 9677726511474388510,
                        "usedCapacity" : 2660499456,
                        "virtualServerId" : 2
                }
        ]
    }
```

Get a file system

Retrieves detailed information about a file system on the storage system.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/filesystems/{id}
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Either the object ID or HNAS file system ID.

Return codes

Code	Data	Description
200	filesystem	File system object retrieved successfully.

Chapter 5: File system resource

Code	Data	Description
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example: Get a file system using the object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.12.44.55:8444/v7/storage/filesystems/
383634423132323841424231343441433030303030303030303030303030303a3a3a303a3a3a4f4944
5f24232140255f56
```

Response example: Get a file system using the object ID

```
HTTP/1.1 200 OK
{
    "filesystem" :
        {
                "blockSize" : 32768,
                "capacity" : 58518929408,
                "expansionLimits" : 107374182400,
                "filesystemId" : "864B1228ABB144AC0000000000000000",
                "freeCapacity" : 18763251712,
                "isDedupeEnabled" : true,
                "isDedupeSupported" : true,
                "isLogicalCapacityFreeCapacityValid" : true,
                "isLogicalCapacityValid" : true,
                "isNDMPRecoveryTarget" : false,
                "isNonStrictWORM" : false,
                "isObjectReplicationTarget" : false,
                "isReadCached" : false,
                "isReadOnly" : false,
                "isSysLocked" : true,
                "isThinProvisioningEnabled" : false,
                "isThinProvisioningEnabledValid" : true,
                "isTrueSparseFileEnabled" : false,
                "isTrueSparseFileEnabledValid" : true,
                "isUnlimitedExpansion" : false,
                "isWORM" : false,
                "label" : "xyz-obj-rep-target-new",
                "logicalCapacity" : 0,
                "logicalFreeCapacity" : 0,
```

Chapter 5: File system resource

```
             "objectId" :
"38363442313232384142423134344143303030303030303030303030303030303a3a3a303a3a3a4f49445
f24232140255f56",
             "status" : "MOUNTED",
             "storagePoolId" : 9676880324761524863,
             "usedCapacity" : 39755677696,
             "virtualServerId" : 3
              }
        }
```

Request example: Get a file system using the HNAS storage file system ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/864B1228ABB144AC0000000000000000
```

Response example: Get a file system using the HNAS storage file system ID

```
HTTP/1.1 200 OK
        {
         "filesystem" :
          {
          "blockSize" : 32768,
          "capacity" : 58518929408,
          "expansionLimits" : 107374182400,
          "filesystemId" : "864B1228ABB144AC0000000000000000",
          "freeCapacity" : 18763317248,
          "isDedupeEnabled" : true,
          "isDedupeSupported" : true,
          "isLogicalCapacityFreeCapacityValid" : true,
          "isLogicalCapacityValid" : true,
          "isNDMPRecoveryTarget" : false,
          "isNonStrictWORM" : false,
          "isObjectReplicationTarget" : true,
          "isReadCached" : false,
          "isReadOnly" : false,
          "isSysLocked" : true,
          "isThinProvisioningEnabled" : false,
          "isThinProvisioningEnabledValid" : true,
          "isTrueSparseFileEnabled" : false,
          "isTrueSparseFileEnabledValid" : true,
          "isUnlimitedExpansion" : false,
          "isWORM" : false,
          "label" : "xyz-obj-rep-target-new",
          "logicalCapacity" : 0,
          "logicalFreeCapacity" : 0,
          "objectId" :
"38363442313232384142423134344143303030303030303030303030303030303a3a3a303a3a3a4f49445
f24232140255f56",
             "status" : "MOUNTED",
             "storagePoolId" : 9676880324761524863,
```

Chapter 5: File system resource

```
            "usedCapacity" : 39755612160,
            "virtualServerId" : 3
            }
        }
```

Create a file system

Creates a file system on a virtual server using a storage pool. A label is assigned to the file system during the creation process. The operation returns a URI that can be accessed to retrieve detailed information about the file system.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/filesystems
```

Parameters

Name	Type	Required	Values	Description
label	BODY	Y	string	Label of the file system to be created.
virtualServerId	BODY	Y	number	ID of the virtual server to be used.
storagePoolId	BODY	Y	number	ID of the storage pool to be used.
capacity	BODY	Y	number	Capacity of the file system to be created in bytes.

Return codes

Code	Data	Description
201	filesystem	File system object successfully created.
400	No Data	Missing or invalid request contents.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 201 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems -d '{"label":"fs-test",
"virtualServerId":4, "storagePoolId":4286815075241083603, "capacity":19327352832}' -
X POST
```

Response example

```
HTTP/1.1 201 Created
{
   "filesystem" : {
      "blockSize" : 0,
      "capacity" : 19268632576,
      "expansionLimits" : 18446744073709551615,
      "filesystemId" : "866260F047C882190000000000000000",
      "freeCapacity" : 0,
      "isDedupeEnabled" : false,
      "isDedupeSupported" : false,
      "isNDMPRecoveryTarget" : false,
      "isNonStrictWORM" : false,
      "isObjectReplicationTarget" : false,
      "isReadCached" : false,
      "isReadOnly" : false,
      "isSysLocked" : false,
      "isUnlimitedExpansion" : true,
      "isWORM" : false,
      "label" : "fs-test",
      "objectId" : "38363632363046303437433838323139303030303030303030303030303030303030",
      "status" : "NOT_MOUNTED",
      "usedCapacity" : 0
   },
   "uri" : "https://172.17.11.11:8444/v7/storage/filesystems/
38363632363046303437433838323139303030303030303030303030303030303030"
```

Get the virtual server associated with a file system

Retrieves detailed information about a virtual server on which a file system resides.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/filesystems/{id}/virtual-servers
```

Chapter 5: File system resource

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Either file system object ID or HNAS storage file system ID.

Return codes

Code	Data	Description
200	virtualServer	Virtual server object retrieved successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example: Using the file system object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/
38363632363743313235414337314142303030303030303030303030303030/virtual-servers
```

Response example: Using the file system object ID

```
HTTP/1.1 200 OK
{
    "virtualServer" :
        {
                "UUID" : "b70d82b2-0386-11d0-9043-49e1bb864b23",
                "ipAddresses" :
                [
                        "172.17.58.122"
                ],
                "isEnabled" : true,
                "name" : "xyz-evs-58-122",
                "objectId" : "333a3a3a3a3a3a303a3a3a4f49445f24232140255f56",
                "status" : "ONLINE",
                "type" : "File services",
                "virtualServerId" : 3
```

Chapter 5: File system resource

```
        }
    }
```

Request example: Using the HNAS storage file system ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/866267C125AC71AB0000000000000000/
virtual-servers
```

Response example: Using the HNAS storage file system ID

```
HTTP/1.1 200 OK
{
    "virtualServer" :
        {
                "UUID" : "b70d82b2-0386-11d0-9043-49e1bb864b23",
                "ipAddresses" :
                [
                        "172.17.58.122"
                ],
                "isEnabled" : true,
                "name" : "xyz-evs-58-122",
                "objectId" : "333a3a3a3a3a3a303a3a3a4f49445f24232140255f56",
                "status" : "ONLINE",
                "type" : "File services",
                "virtualServerId" : 3
        }
}
```

Get the storage pool associated with a file system

Retrieves detailed information about a storage pool used to create a file system.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/filesystems/{id}/storage-pools
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Specifies either the file system object ID or the HNAS storage file system ID.

Chapter 5: File system resource

Return codes

Code	Data	Description
200	storagePool	Storage pool object retrieved successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example: Using the file system object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/
38363632363743313235414337314142303030303030303030303030303030/storage-pools
```

Response example: Using the file system object ID

```
HTTP/1.1 200 OK
    {
     "storagePool" :
        {
                "chunkSize" : 19327352832,
                "freeCapacity" : 175728754688,
                "isAssignedToLocalCluster" : true,
                "isFilesystemExpansionAllowed" : true,
                "isHealthy" : true,
                "isTiered" : false,
                "label" : "historDevStoragePool",
                "objectId" :
"39363736383830333234373631353234383336333a3a3a3a3a3a303a3a3a4f49445f24232140255f56",
                "storagePoolId" : 9676880324761524863,
                "totalCapacity" : 429492535296,
                "usedCapacity" : 253763780608
        }
    }
```

Request example: Using the HNAS storage file system ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/866267C125AC71AB0000000000000000/
storage-pools
```

Chapter 5: File system resource

Response example: Using the HNAS storage file system ID

```
HTTP/1.1 200 OK
   {
    "storagePool" :
       {
             "chunkSize" : 19327352832,
             "freeCapacity" : 175728754688,
             "isAssignedToLocalCluster" : true,
             "isFilesystemExpansionAllowed" : true,
             "isHealthy" : true,
             "isTiered" : false,
             "label" : "historDevStoragePool",
             "objectId" :
"39363736383830333234373631353234383633333a3a3a3a3a3a303a3a3a4f49445f24232140255f56",
             "storagePoolId" : 9676880324761524863,
             "totalCapacity" : 429492535296,
             "usedCapacity" : 253763780608
       }
   }
```

Get file system snapshots associated with a file system

Retrieves detailed information about file system snapshots associated with a file system on the storage system. A unique identifier value identifies the file system.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/filesystems/{id}/snapshots
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Specifies either the file system object ID or the HNAS storage file system ID.

Return codes

Code	Data	Description
200	snapshot	Storage file system snapshots retrieved successfully.

Chapter 5: File system resource

Code	Data	Description
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example: Using the file system object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/
3836353945344445384532464634454130303030303030303030303030303030/snapshots
```

Response example: Using the file system object ID

```
HTTP/1.1 200 OK
  {
   "snapshots" : [
      {
         "creationReason" : "SNAPSHOT_CREATED_BY_RULE",
         "creationTime" : 1477006200,
         "displayName" : "2016-10-20_1630-0700.rickSSRule",
         "filesystemId" : "8659E4DE8E2FF4EA0000000000000000",
         "objectId" :
"3836353945344445384532464634454130303030303030303030303030303030303a3a3a323031362d31302
d32305f313633302d303730302e7269636b535352756c65",
         "state" : "SNAPSHOT_MANAGEMENT_STATE_USABLE"
      },
      {
         "creationReason" : "SNAPSHOT_CREATED_BY_APPLICATION",
         "creationTime" : 1459455905,
         "displayName" : "snap1-rickProdFS",
         "filesystemId" : "8659E4DE8E2FF4EA0000000000000000",
         "objectId" :
"3836353945344445384532464634454130303030303030303030303030303030303a3a3a736e6170312d726
9636b50726f644653",
         "state" : "SNAPSHOT_MANAGEMENT_STATE_USABLE"
      },
      {
         "creationReason" : "SNAPSHOT_CREATED_MANUALLY",
         "creationTime" : 1426638564,
         "displayName" : "xyz-test-snapshots-0315",
         "filesystemId" : "8659E4DE8E2FF4EA0000000000000000",
         "objectId" :
```

Chapter 5: File system resource

```
"38363539453444453845324646344541303030303030303030303030303030303a3a3a6e69636b2d74657
3742d736e617073686f74732d30333135",
        "state" : "SNAPSHOT_MANAGEMENT_STATE_USABLE"
      }
    ]
 }
```

Request example: Using the HNAS storage file system ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/8659E4DE8E2FF4EA0000000000000000/
snapshots
```

Response example: Using the HNAS storage file system ID

```
HTTP/1.1 200 OK
  {
   "snapshots" : [
      {
        "creationReason" : "SNAPSHOT_CREATED_BY_RULE",
        "creationTime" : 1477006200,
        "displayName" : "2016-10-20_1630-0700.rickSSRule",
        "filesystemId" : "8659E4DE8E2FF4EA0000000000000000",
        "objectId" :
"38363539453444453845324646344541303030303030303030303030303030303a3a3a323031362d31302
d32305f313633302d303730302e7269636b535352756c65",
        "state" : "SNAPSHOT_MANAGEMENT_STATE_USABLE"
      },
      {
        "creationReason" : "SNAPSHOT_CREATED_BY_APPLICATION",
        "creationTime" : 1459455905,
        "displayName" : "snap1-rickProdFS",
        "filesystemId" : "8659E4DE8E2FF4EA0000000000000000",
        "objectId" :
"38363539453444453845324646344541303030303030303030303030303030303a3a3a736e6170312d726
9636b50726f644653",
        "state" : "SNAPSHOT_MANAGEMENT_STATE_USABLE"
      },
      {
        "creationReason" : "SNAPSHOT_CREATED_MANUALLY",
        "creationTime" : 1426638564,
        "displayName" : "xyz-test-snapshots-0315",
        "filesystemId" : "8659E4DE8E2FF4EA0000000000000000",
        "objectId" :
"38363539453444453845324646344541303030303030303030303030303030303a3a3a6e69636b2d74657
3742d736e617073686f74732d30333135",
        "state" : "SNAPSHOT_MANAGEMENT_STATE_USABLE"
      }
    ]
  }
```

Chapter 5: File system resource

Expand a file system

Expands the file system. You can expand the file system by specifying the desired capacity of the file system in bytes.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/filesystems/{id}/expand
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Specifies either the file system object ID or the HNAS storage file system ID to expand.
capacity	BODY	Y	number	Desired capacity of the file system in bytes.

Return codes

Code	Data	Description
204	No Data	File system successfully expanded.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example: Using the file system object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/
38363632363743313235414337314142303030303030303030303030303030/expand -d
'{"capacity":108447924224}' -X POST
```

Response example: Using the file system object ID

```
HTTP/1.1 204 No Content
```

Request example: Using the HNAS storage file system ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/8659E4DE8E2FF4EA0000000000000000/
expand -d '{"capacity":108447924224}' -X POST
```

Response example: Using the HNAS storage file system ID

```
HTTP/1.1 204 No Content
```

Rename a file system

Renames a file system. Also, you can rename the file system by changing the label associated with the file system.

HTTP request syntax (URI)

```
PATCH <base_URI>/v7/storage/filesystems/{id}
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Specifies either the file system object ID or the HNAS storage file system ID.
label	BODY	Y	string	The new label for the file system.

Return codes

Code	Data	Description
204	No Data	File system successfully renamed.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.

Code	Data	Description
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example: Using the file system object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/
38363632363743313235414337314142303030303030303030303030303030 -d
'{"label":"TestFS"}' -X PATCH
```

Response example: Using the file system object ID

```
HTTP/1.1 204 No Content
```

Request example: Using the HNAS storage file system ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/8659E4DE8E2FF4EA0000000000000000 -
d '{"label":"TestFS"}' -X PATCH
```

Response example: Using the HNAS storage file system ID

```
HTTP/1.1 204 No Content
```

Format a file system

Formats a file system. Specify the file system block size.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/filesystems/{id}/format
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Specifies either the file system object ID or the HNAS storage file system ID to format.

Name	Type	Required	Values	Description
blockSize	BODY	Y	number	Specifies the block size of the file system in kilobytes (KB). Accepted values can be either 4 or 32.

Return codes

Code	Data	Description
204	No Data	File system successfully formatted.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example: Using the file system object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/
38363632363743313235414337314142303030303030303030303030303030/format -d
'{"blockSize":4}' -X POST
```

Response example: Using the file system object ID

```
HTTP/1.1 204 No Content
```

Request example: Using the HNAS storage file system ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/8659E4DE8E2FF4EA0000000000000000/
format -d '{"blockSize":4}' -X POST
```

Response example: Using the HNAS storage file system ID

```
HTTP/1.1 204 No Content
```

Mount a file system

Mounts a file system.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/filesystems/{id}/mount
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Specifies either the file system object ID or the HNAS storage file system ID to mount.

Return codes

Code	Data	Description
204	No Data	File system mounted successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example: Using the file system object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/
38363632363743313235414337314142303030303030303030303030303030/mount -X POST
```

Response example: Using the file system object ID

```
HTTP/1.1 204 No Content
```

Request example: Using the HNAS storage file system ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/8659E4DE8E2FF4EA0000000000000000/
mount -X POST
```

Response example: Using the HNAS storage file system ID

```
HTTP/1.1 204 No Content
```

Unmount a file system

Unmounts a file system.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/filesystems/{id}/unmount
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Specifies either the file system object ID or the HNAS storage file system ID to unmount.

Return codes

Code	Data	Description
204	No Data	File system unmounted successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example: Using the file system object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/
3836363236374331323541433731414230303030303030303030303030303030/unmount -X POST
```

Response example: Using the file system object ID

```
HTTP/1.1 204 No Content
```

Request example: Using the HNAS storage file system ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/8659E4DE8E2FF4EA0000000000000000/
unmount -X POST
```

Response example: Using the HNAS storage file system ID

```
HTTP/1.1 204 No Content
```

Delete a file system

Deletes a file system. You cannot delete a file system if it is in a mounted state.

HTTP request syntax (URI)

```
DELETE <base_URI>/v7/storage/filesystems/{id}
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Specifies either the file system object ID or the HNAS storage file system ID.

Return codes

Code	Data	Description
204	No Data	File system object deleted successfully.
400	No Data	Missing or invalid request contents.

Chapter 5: File system resource

Code	Data	Description
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example: Using the file system object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/
38363632363046303437433838323139303030303030303030303030303030 -X DELETE
```

Response example: Using the file system object ID

```
HTTP/1.1 204 OK
```

Request example: Using the HNAS storage file system ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/8659E4DE8E2FF4EA0000000000000000 -
X DELETE
```

Response example: Using the HNAS storage file system ID

```
HTTP/1.1 204 OK
```

Set a file system as object replication target

Sets a file system as an object replication target.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/filesystems/{id}
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Specifies either the file system object ID or the HNAS storage file system ID.
enable	BODY	Y	boolean	True to enable the file system as the object replication target. False to disable.
enableTransferAccessPoint	BODY	Y	boolean	True to enable the transfer of shares and exports that have been configured to use filesystem default settings. If the enable parameter is set to true, then enableTransferAccessPoint is ignored.

Return codes

Code	Data	Description
204	No Data	File system object set as object replication target successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example: Using the file system object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/
38363646463637454244453645434133303030303030303030303030303030303a3a3a303a3a3a4f4944
5f24232140255f56 -X POST -d '{"enable":false, "enableTransferAccessPoint":false}'
```

Response example: Using the file system object ID

```
HTTP/1.1 204 No Content
```

Request example: Using the HNAS storage file system ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/8659E4DE8E2FF4EA0000000000000000 -
X POST -d '{"enable":true,"enableTransferAccessPoint":false}'
```

Response example: Using the HNAS storage file system ID

```
HTTP/1.1 204 No Content
```

Get status of disaster recovery state change of a file system

Gets the status of a disaster recovery state change of a file system.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/filesystems/{id}/dr-state-change-status
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Specifies either the file system object ID or the HNAS storage file system ID.
isMostRecent	BODY	N	boolean	True returns only the most recent status. False returns a list of previous statuses. Default is False if the parameter is not supplied.

Return codes

Code	Data	Description
200	filesystemDRStateChangeReport	Status of disaster recovery state change of the file system object retrieved successfully
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example: Using the file system object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/
865a1e306b0d4ee700000000000000000/dr-state-change-status -X GET -d
'{"isMostRecent":true}'
```

Response example: Using the file system object ID

```
HTTP/1.1 200 OK
{
  "filesystemDRStateChangeReport" :
  [
    {
      "endTime" : 1545960264,
      "fileSystemId" : "865A1E306B0D4EE70000000000000000",
      "isActive" : false,
      "logName" : "/2/simm/DR_2018-12-27_172422-0800.bin",
      "startTime" : 1545960262,
      "statistics" :
      [
        {
          "name" : "numShareRecoveredEvents",
          "value" : 0
        },
        {
          "name" : "numShareRecoveryFailureEvents",
          "value" : 0
        },
```

Chapter 5: File system resource

```
    {
      "name" : "numShareRecoveryFailureDueToNameClashEvents",
      "value" : 0
    },
    {
      "name" : "numShareNameClashFixedEvents",
      "value" : 0
    },
    {
      "name" : "numShareSkippedDueToNameClashEvents",
      "value" : 0
    },
    {
      "name" : "numShareSkippedDueToDoNotTransferEvents",
      "value" : 0
    },
    {
      "name" : "numShareSkippedDueToUseFsDefaultEvents",
      "value" : 0
    },
    {
      "name" : "numShareSkippedDueToNameClashIdenticalShareEvents",
      "value" : 0
    },
    {
      "name" : "numShareDeletedEvents",
      "value" : 0
    },
    {
      "name" : "numShareDeletionFailureEvents",
      "value" : 0
    },
    {
      "name" : "numExportRecoveredEvents",
      "value" : 0
    },
    {
      "name" : "numExportRecoveryFailureEvents",
      "value" : 0
    },
    {
      "name" : "numExportRecoveryFailureDueToNameClashEvents",
      "value" : 0
    },
    {
      "name" : "numExportNameClashFixedEvents",
      "value" : 0
    },
    {
      "name" : "numExportSkippedDueToNameClashEvents",
      "value" : 0
```

Chapter 5: File system resource

```
      },
      {
        "name" : "numExportSkippedDueToDoNotTransferEvents",
        "value" : 0
      },
      {
        "name" : "numExportSkippedDueToUseFsDefaultEvents",
        "value" : 0
      },
      {
        "name" : "numExportSkippedDueToNameClashIdenticalShareEvents",
        "value" : 0
      },
      {
        "name" : "numExportIdClashEvents",
        "value" : 0
      },
      {
        "name" : "numExportDeletedEvents",
        "value" : 0
      },
      {
        "name" : "numExportDeletionFailureEvents",
        "value" : 0
      },
      {
        "name" : "numFailedToLogToDrLogEvents",
        "value" : 0
      }
    ],
    "status" : "SUCCESSFULLY",
    "transitionId" : "0ff38b10-9e4c-11d4-9645-49e1bb864b23"
    }
  ]
}
```

Request example: Using the HNAS storage file system ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/
8659E4DE8E2FF4EA0000000000000000/dr-state-change-status -X GET -d
'{"isMostRecent":true}'
```

Response example: Using the HNAS storage file system ID

```
HTTP/1.1 200 OK
{
  "filesystemDRStateChangeReport" :
  [
    {
```

```
    "endTime" : 1545960264,
    "fileSystemId" : "865A1E306B0D4EE70000000000000000",
    "isActive" : false,
    "logName" : "/2/simm/DR_2018-12-27_172422-0800.bin",
    "startTime" : 1545960262,
    "statistics" :
    [
      {
        "name" : "numShareRecoveredEvents",
        "value" : 0
      },
      {
        "name" : "numShareRecoveryFailureEvents",
        "value" : 0
      },
      {
        "name" : "numShareRecoveryFailureDueToNameClashEvents",
        "value" : 0
      },
      {
        "name" : "numShareNameClashFixedEvents",
        "value" : 0
      },
      {
        "name" : "numShareSkippedDueToNameClashEvents",
        "value" : 0
      },
      {
        "name" : "numShareSkippedDueToDoNotTransferEvents",
        "value" : 0
      },
      {
        "name" : "numShareSkippedDueToUseFsDefaultEvents",
        "value" : 0
      },
      {
        "name" : "numShareSkippedDueToNameClashIdenticalShareEvents",
        "value" : 0
      },
      {
        "name" : "numShareDeletedEvents",
        "value" : 0
      },
      {
        "name" : "numShareDeletionFailureEvents",
        "value" : 0
      },
      {
        "name" : "numExportRecoveredEvents",
        "value" : 0
      },
```

Chapter 5: File system resource

```
        {
          "name" : "numExportRecoveryFailureEvents",
          "value" : 0
        },
        {
          "name" : "numExportRecoveryFailureDueToNameClashEvents",
          "value" : 0
        },
        {
          "name" : "numExportNameClashFixedEvents",
          "value" : 0
        },
        {
          "name" : "numExportSkippedDueToNameClashEvents",
          "value" : 0
        },
        {
          "name" : "numExportSkippedDueToDoNotTransferEvents",
          "value" : 0
        },
        {
          "name" : "numExportSkippedDueToUseFsDefaultEvents",
          "value" : 0
        },
        {
          "name" : "numExportSkippedDueToNameClashIdenticalShareEvents",
          "value" : 0
        },
        {
          "name" : "numExportIdClashEvents",
          "value" : 0
        },
        {
          "name" : "numExportDeletedEvents",
          "value" : 0
        },
        {
          "name" : "numExportDeletionFailureEvents",
          "value" : 0
        },
        {
          "name" : "numFailedToLogToDrLogEvents",
          "value" : 0
        }
      ],
      "status" : "SUCCESSFULLY",
      "transitionId" : "0ff38b10-9e4c-11d4-9645-49e1bb864b23"
    }
  ]
}
```

Chapter 5: File system resource

Change disaster recovery state of a file system

Changes the disaster recovery state of a file system.

HTTP request syntax (URI)

```
PATCH <base_URI>/v7/storage/filesystems/{id}/change-dr-state
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Specifies either the file system object ID or the HNAS storage file system ID.
transitionOption	BODY	Y	string	Transition state of the disaster recovery. Current supported values are: ▪ READ_WRITE ▪ READ_ONLY ▪ REPLICATION_TARGET
snapshotName	BODY	N	string	Snapshot name.
isRecoverSharesOnPromoted	BODY	N	boolean	True if shares are recovered on a promoted file system. The default value is True.
isRecoverExportsOnPromoted	BODY	N	boolean	True if exports are recovered on a promoted file system. The default value is True.

Name	Type	Required	Values	Description
isDeleteSharesFromDemoted	BODY	N	boolean	True if shares are deleted from a promoted file system. The default value is True.
isDeleteExportsFromDemoted	BODY	N	boolean	True if exports are deleted from a promoted file system. The default value is True.

Return codes

Code	Data	Description
204	No Data	Changed disaster recovery state of the file system object successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example: Using the file system object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/
38363646463637454244453645434133303030303030303030303030303030303a3a303a3a3a4f4944
5f24232140255f56/change-dr-state -X PATCH -d '{"transitionOption":"READ_ONLY"}'
```

Response example: Using the file system object ID

```
HTTP/1.1 204 No Content
```

Chapter 5: File system resource

Request example: Using the HNAS storage file system ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/8659E4DE8E2FF4EA0000000000000000/
change-dr-state -X PATCH -d '{"transitionOption":"READ_ONLY"}'
```

Response example: Using the HNAS storage file system ID

```
HTTP/1.1 204 No Content
```

Get deduplication file systems

Retrieves deduplication file systems on the managed storage system. A deduplication file system is a file system that has duplicate information removed.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/filesystems/type/dedup
```

Return codes

Code	Data	Description
200	dedupFilesystems	Array of dedup file systems retrieved successfully.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/type/dedup
```

Response example

```
HTTP/1.1 200 OK
 {
"dedupFilesystems" :
        [
             {
                   "filesystemId" : "3B70ACE241339156000000000000000000",
                   "isEnabled" : true,
                   "lastRun" : 1478550139,
```

Chapter 5: File system resource

```
                         "percentageReclaimed" : 0,
                         "reclaimedCapacity" : 0,
                         "virtualServerId" : 3
                },
                {
                         "filesystemId" : "8643CABC964FEEB30000000000000000",
                         "isEnabled" : true,
                         "lastRun" : 1523989308,
                         "percentageReclaimed" : 13,
                         "reclaimedCapacity" : 2373353472,
                         "virtualServerId" : 4
                },
                {
                         "filesystemId" : "864B1204593F504E0000000000000000",
                         "isEnabled" : true,
                         "lastRun" : 0,
                         "percentageReclaimed" : 0,
                         "reclaimedCapacity" : 0,
                         "virtualServerId" : 1
                },
                {
                         "filesystemId" : "864B1228ABB144AC0000000000000000",
                         "isEnabled" : true,
                         "lastRun" : 0,
                         "percentageReclaimed" : 0,
                         "reclaimedCapacity" : 0,
                         "virtualServerId" : 3
                }
        ]
}
```

Get replication snapshots from a target (destination)

Gets replication snapshots from a target (destination) of a file system.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/filesystems/{id}/replication-snapshots
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Specifies either the file system object ID or the HNAS storage file system ID.

Chapter 5: File system resource

Return codes

Code	Data	Description
200	replicationSnapshotDetails	A list of replication snapshot details from a file system successfully retrieved.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystems/3B7B298BBCE736650000000000000000/
replication-snapshots
```

Response example

```
HTTP/1.1 200 OK
{

        "replicationSnapshotDetails" :
        [
            {
                    "replicationSnapshotStatus" : "REPLICATION_TARGET_COMPLETE",
                    "replicationType" : "INCREMENTAL",
                    "sourceFilesystemId" : "3B6BB41DC647FC060000000000000000",
                    "sourceSnapshotName" : "AUTO_SNAPSHOT_ec0545e4-a3c9-11d4-9797-
49e1bb864b23_2",
                    "sourceSnapshotTime" : 1546817620,
                    "targetFilesystemId" : "86589FF84F28BA590000000000000000",
                    "targetSnapshotName" : "AUTO_SNAPSHOT_TARGET_2"
            }
        ]

}
```

Get user and group quotas of a file System

Gets user and group quotas of a file system.

Chapter 5: File system resource

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/filesystems/{id}/quotas
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Specifies either the object ID or the HNAS file system ID.
virtualServerId	BODY	N	number	Virtual server ID where the file system resides. This parameter is ignored, as the virtual server ID can be derived from the filesystem ID. It is maintained for backward compatability.

Return codes

Code	Data	Description
200	fileSystemQuotas	User and groups quotas successfully retrieved.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystems/864553F647F8647D0000000000000000/
quotas -X GET -d '{"virtualServerId":3}
```

Response example

```
HTTP/1.1 200 OK
{
        "filesystemQuotas" :
        [
                {
                        "filesystemId" : "864553F647F8647D0000000000000000",
                        "quota" :
                        {
                                "diskUsage" : 32768,
                                "diskUsageThreshold" :
                                {
                                        "isHard" : true,
                                        "limit" : 1073741824,
                                        "reset" : 5,
                                        "severe" : 90,
                                        "warning" : 70
                                },
                                "fileCountThreshold" :
                                {
                                        "isHard" : true,
                                        "limit" : 20,
                                        "reset" : 5,
                                        "severe" : 70,
                                        "warning" : 40
                                },
                                "fileUsage" : 9,
                                "logEvent" : false,
                                "targetDomain" : "",
                                "targetName" : "root",
                                "targetType" : "USER"
                        },
                        "virtualServerId" : 3
                },
                {
                        "filesystemId" : "864553F647F8647D0000000000000000",
                        "quota" :
                        {
                                "diskUsage" : 0,
                                "diskUsageThreshold" :
                                {
                                        "isHard" : true,
                                        "limit" : 1234567890,
                                        "reset" : 65,
                                        "severe" : 95,
                                        "warning" : 75
                                },
                                "fileCountThreshold" :
                                {
                                        "isHard" : false,
```

Chapter 5: File system resource

```
                              "limit" : 1000,
                              "reset" : 85,
                              "severe" : 95,
                              "warning" : 90
                      },
                      "fileUsage" : 5,
                      "logEvent" : true,
                      "targetDomain" : "BUILTIN",
                      "targetName" : "Current Owner",
                      "targetType" : "USER"
                },
                "virtualServerId" : 3
        },
        {
                "filesystemId" : "864553F647F8647D0000000000000000",
                "quota" :
                {
                        "diskUsage" : 0,
                        "diskUsageThreshold" :
                        {
                                "isHard" : true,
                                "limit" : 1234567890,
                                "reset" : 65,
                                "severe" : 95,
                                "warning" : 75
                        },
                        "fileCountThreshold" :
                        {
                                "isHard" : false,
                                "limit" : 1000,
                                "reset" : 85,
                                "severe" : 95,
                                "warning" : 90
                        },
                        "fileUsage" : 0,
                        "logEvent" : true,
                        "targetDomain" : "BUILTIN",
                        "targetName" : "Current Owner",
                        "targetType" : "GROUP"
                },
                "virtualServerId" : 3
        }
    ]
}
```

Add a user quota

Adds a user quota by file system ID.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/filesystems/{id}/user-quota
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Either the object ID or the HNAS file system ID
diskUsageThreshold	BODY	Y	object	quotaThresholdPerc entages object of the disk space to be added to the virtual volume.
fileCountThreshold	BODY	Y	object	quotaThresholdPerc entages object of the file count to be added to the virtual volume.
logEvent	BODY	Y	boolean	Indicates whether to log an event.

Name	Type	Required	Values	Description
userId	BODY	Y	string	User can be either a NFS user or a CIFS user. For an NFS user, it can be NFS v2/3 format with its UNIX user ID optionally registered in the HNAS user mappings. For example, if user NFSuser with its UNIX user ID 501 is added to User Mappings, then you can add a user quota with the user named "NFSuser." For an NFS v4 format, the user must attach its host name in format of user@host (NFSv4User@deb50 10-57-75). A CIFS user can be discovered from The Windows Domain Controller or input by an HNAS user. Usually the user is in format of Domain \user, such as BUILTIN\Current User.
virtualServerId	BODY	N	number	Virtual server ID where the file system resides. This parameter is ignored, as the virtual server ID can be derived from the filesystem ID. It is maintained for backward compatability.

Chapter 5: File system resource

Return codes

Code	Data	Description
201	userQuota	User quota successfully added
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 201 indicates that the API did not complete successfully.

Request example of a NFS user

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystems/864553F647F8647D0000000000000000/
user-quota -d '{"diskUsageThreshold":{"limit":1234567890,"isHard":true,"reset":65,
"warning":75,"severe":95}, "fileCountThreshold":{"limit":1000,"isHard":false,
"reset":85,"warning":90, "severe":95},"logEvent":true, "userId":"xyz",
"virtualServerId":3}' -X POST
```

Response example

```
HTTP/1.1 201 Created
{
            "userQuota" :
     {
          "filesystemId" : "864553F647F8647D0000000000000000",
          "quota" :
          {
                 "diskUsage" : 0,
                 "diskUsageThreshold" :
                 {
                          "isHard" : true,
                          "limit" : 1234567890,
                          "reset" : 65,
                          "severe" : 95,
                          "warning" : 75
                 },
                 "fileCountThreshold" :
                 {
                          "isHard" : false,
                          "limit" : 1000,
                          "reset" : 85,
                          "severe" : 95,
                          "warning" : 90
```

Chapter 5: File system resource

```
                    },
                    "fileUsage" : 0,
                    "logEvent" : true,
                    "targetDomain" : "",
                    "targetName" : "xyz",
                    "targetType" : "USER"
                },
                "virtualServerId" : 3
        }
}
```

Request example of a CIFS user

```
curl -vk -H "X-Api-Key: Td5qNSpXX4.732uVwjjuN1Wgmxw7yJwL5nygQk79k6pbVg.wvMFqH2"
https://172.27.250.210:8444/v7/storage/filesystems/864553F647F8647D0000000000000000/
user-quota -X POST -d'{"diskUsageThreshold": {"limit":1234567890, "isHard":true,
"reset":65,"warning":75,"severe":95},"fileCountThreshold":{"limit":1000,
"isHard":false,"reset":85,"warning":90,"severe":95}, "userId": "BUILTIN\\current
owner","virtualVolumeName":"nickvvol-test1","logEvent":true}'
```

Response example of a CIFS user

```
HTTP/1.1 201 Created
{
   "uri" : "https://172.27.250.210:8444/v7/storage/virtual-volumes/
323a3a3a41463042464544433143463342434337303030303030303030303030303a3a3a323a3a3a
303a3a3a4f49445f24232140255f56/user-quotas",
   "userQuota" : {
      "filesystemId" : "AF0BFEDC1CF3BCC70000000000000000",
      "objectId" : "nickvvol-test1",
      "quota" : {
         "diskUsage" : 0,
         "diskUsageThreshold" : {
            "isHard" : true,
            "limit" : 1234567890,
            "reset" : 65,
            "severe" : 95,
            "warning" : 75
         },
         "fileCountThreshold" : {
            "isHard" : false,
            "limit" : 1000,
            "reset" : 85,
            "severe" : 95,
            "warning" : 90
         },
         "fileUsage" : 0,
         "logEvent" : true,
         "targetDomain" : "BUILTIN",
         "targetName" : "Current Owner",
```

Chapter 5: File system resource

```
        "targetType" : "USER"
    },
    "userId" : "BUILTIN\\current owner",
    "virtualServerId" : 2,
    "virtualVolumeId" : 2,
    "virtualVolumeName" : "nickvvol-test1"
   }
}
```

Modify a user quota

Modifies a user quota by file system ID.

HTTP request syntax (URI)

```
PATCH <base_URI>/v7/storage/filesystems/{id}/user-quota
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Either the object ID or the HNAS file system ID.
diskUsageThreshold	BODY	Y	object	quotaThresholdPerc entages object of the disk space to be added to the virtual volume.
fileCountThreshold	BODY	Y	object	quotaThresholdPerc entages object of the file count to be added to the virtual volume.
logEvent	BODY	Y	boolean	Indicates whether to log an event.
userId	BODY	Y	string	The user can be either a NFS (v2/v3/v4) user or a Windows user name.

Name	Type	Required	Values	Description
virtualServerId	BODY	N	number	Virtual server ID where the file system resides. This parameter is ignored, as the virtual server ID can be derived from the filesystem ID. It is maintained for backward compatability.

Return codes

Code	Data	Description
204	No Data	User quota successfully modified
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystems/864553F647F8647D0000000000000000/
user-quota -d '{"diskUsageThreshold":{"limit":1234567890,"isHard":true,"reset":65,
"warning":75,"severe":95}, "fileCountThreshold":{"limit":1000,"isHard":false,
"reset":85,"warning":90, "severe":95},"logEvent":true, "userId":"xyz",
"virtualServerId":3}' -X PATCH
```

Response example

```
HTTP/1.1 204 No Content
```

Delete a user quota

Deletes a user quota.

> 📄 **Note:** This operation can remove only user quotas created by adding a user quota.

HTTP request syntax (URI)

```
DELETE <base_URI>/v7/storage/filesystems/{id}/user-quota
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Either the object ID or the HNAS file system ID.
userId	BODY	Y	string	Target name of the quota.
virtualServerId	BODY	N	number	Virtual server ID where the file system resides. This parameter is ignored, as the virtual server ID can be derived from the filesystem ID. It is maintained for backward compatability.

Return codes

Code	Data	Description
204	No Data	User quota successfully deleted
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystems/864553F647F8647D0000000000000000/
user-quota -d '{"userId":"xyz", "virtualServerId":3}' -X DELETE
```

Response example

```
HTTP/1.1 204 No Content
```

Add a group quota

Adds a group quota by file system ID.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/filesystems/{id}/group-quota
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Either the object ID or the HNAS file system ID
diskUsageThreshold	BODY	Y	object	quotaThresholdPercentages object of the disk space to be added to the virtual volume.
fileCountThreshold	BODY	Y	object	quotaThresholdPercentages object of the file count to be added to the virtual volume
logEvent	BODY	Y	boolean	Indicates whether to log an event.

Name	Type	Required	Values	Description
groupId	BODY	Y	string	Group can be either a NFS group or a CIFS group. For an NFS group, it can be the NFS v2/3 format with its UNIX group ID optionally registered in the HNAS group mappings. For example, if group NFSGroup with its UNIX group ID 501 is added to Group Mappings, then you can add a group quota with a group named "NFSGroup." For an NFS v4 format, the group must attach its host name in the format of group@host (NFSv4Group@ deb5010-57-75).A CIFS group can be discovered from The Windows Domain Controller or inputted by an HNAS user. Usually the group is in the format of Domain\group, such as BUILTIN\Current Group.
virtualServerId	BODY	N	number	Virtual server ID where the file system resides. This parameter is ignored, as the virtual server ID can be derived from the filesystem ID. It is maintained for backward compatability.

Return codes

Code	Data	Description
201	groupQuota	Group quota successfully added
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 201 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystems/864553F647F8647D0000000000000000/
group-quota -d `{"diskUsageThreshold":{"limit":1234567890,"isHard":true,"reset":65,
"warning":75,"severe":95},"fileCountThreshold":{"limit":1000,"isHard":false,
"reset":85,"warning":90, "severe":95},"logEvent":true, "groupId":"xyz",
"virtualServerId":3}' -X POST
```

Response example

```
HTTP/1.1 201 Created
{
        "groupQuota" :
        {
                "filesystemId" : "864553F647F8647D0000000000000000",
                "quota" :
                {
                        "diskUsage" : 0,
                        "diskUsageThreshold" :
                        {
                                "isHard" : true,
                                "limit" : 1234567890,
                                "reset" : 65,
                                "severe" : 95,
                                "warning" : 75
                        },
                        "fileCountThreshold" :
                        {
                                "isHard" : false,
                                "limit" : 1000,
                                "reset" : 85,
                                "severe" : 95,
                                "warning" : 90
```

```
                },
                "fileUsage" : 0,
                "logEvent" : true,
                "targetDomain" : "",
                "targetName" : "xyz",
                "targetType" : "GROUP"
            },
            "virtualServerId" : 3
        }
}
```

Modify a group quota

Modifies a group quota by file system ID.

HTTP request syntax (URI)

```
PATCH {base_URI}/v7/storage/filesystems/{id}/group-quota
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Either the object ID or the HNAS file system ID
diskUsageThreshold	BODY	Y	object	quotaThresholdPerc entages object of the disk space to be added to the virtual volume.
fileCountThreshold	BODY	Y	object	quotaThresholdPerc entages object of the file count to be added to the virtual volume.
logEvent	BODY	Y	boolean	Indicates whether to log an event.

Name	Type	Required	Values	Description
groupId	BODY	Y	string	Group can be either a NFS group or a CIFS group. For an NFS group, it can be the NFS v2/3 format with its UNIX group ID optionally registered in the HNAS group mappings. For example, if group NFSgroup with its UNIX group ID 501 is added to Group Mappings, then you can add a group quota with the group named "NFSGroup." For an NFS v4 format, the group must attach its host name in format of group@host (NFSv4Group@deb5 010-57-75).A CIFS group can be discovered from The Windows Domain Controller or inputted by an HNAS user. Usually the group is in the format of Domain\group, such as BUILTIN\Current Group. When inputted as an HNAS group, a windows Security Identifier (SID) may also be supplied. In this case, the group can have an empty domain name at the beginning. That is, CIFSTest can be inputted in Group Mappings with a

Name	Type	Required	Values	Description
				windows SID of S-1-5-21-2073948960-1606239773-750479623-1121 under the Windows group name \CIFSTest. In this case, the parameter takes the format of CIFSTest.
virtualServerId	BODY	N	number	Virtual server ID where the file system resides. This parameter is ignored, as the virtual server ID can be derived from the filesystem ID. It is maintained for backward compatability.

Return codes

Code	Data	Description
204	No Data	Group quota successfully modified.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystems/864553F647F8647D0000000000000000/
group-quota -d `{"diskUsageThreshold":{"limit":1234567890,"isHard":true,"reset":65,
"warning":75,"severe":95}, "fileCountThreshold":{"limit":1000,"isHard":false,
"reset":85,"warning":90, "severe":95},"logEvent":true, "groupId":"xyz",
"virtualServerId":3}' -X PATCH
```

Chapter 5: File system resource

Response example

```
HTTP/1.1 204 No Content
```

Delete a group quota

Deletes a group quota by file system ID.

HTTP request syntax (URI)

```
DELETE <base_URI>/v7/storage/filesystems/{id}/group-quota
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Either the object ID or the HNAS file system ID.
groupId	BODY	Y	string	Target name of the quota.
virtualServerId	BODY	N	number	Virtual server ID where the file system resides. This parameter is ignored, as the virtual server ID can be derived from the filesystem ID. It is maintained for backward compatability.

Return codes

Code	Data	Description
204	No Data	Group quota successfully deleted
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.

Chapter 5: File system resource

Code	Data	Description
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystems/864553F647F8647D0000000000000000/
group-quota -d '{"groupId":"xyz", "virtualServerId":3}' -X DELETE
```

Response example

```
HTTP/1.1 204 No Content
```

Get a user quota template of a file system

Gets a user quota template of a file system.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/filesystems/{id}/user-quota-template
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Either the object ID or the HNAS file system ID.
virtualServerId	BODY	N	number	Virtual server ID where the file system resides. This parameter is ignored, as the virtual server ID can be derived from the filesystem ID. It is maintained for backward compatability.

Return codes

Code	Data	Description
200	userQuotaTemplate	User quota template successfully retrieved.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystems/864553F647F8647D0000000000000000/
user-quota-template -X GET -d '{"virtualServerId":3}'
```

Response example

```
HTTP/1.1 200 OK
{
        "userQuotaTemplate" :
        {
                "filesystemId" : "864553F647F8647D0000000000000000",
                "quota" :
                {
                        "diskUsage" : 0,
                        "diskUsageThreshold" :
                        {
                                "isHard" : true,
                                "limit" : 1073741824,
                                "reset" : 5,
                                "severe" : 90,
                                "warning" : 70
                        },
                        "fileCountThreshold" :
                        {
                                "isHard" : true,
                                "limit" : 20,
                                "reset" : 5,
                                "severe" : 70,
                                "warning" : 40
                        },
                        "fileUsage" : 0,
                        "logEvent" : false,
```

Chapter 5: File system resource

```
                "targetDomain" : "",
                "targetName" : "",
                "targetType" : "USER"
        },
        "virtualServerId" : 3
    }
            }
```

Get a group quota template of a file system

Gets a group quota template of a file system.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/filesystems/{id}/group-quota-template
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Either the object ID or the HNAS file system ID.
virtualServerId	BODY	N	number	Virtual server ID where the file system resides. This parameter is ignored, as the virtual server ID can be derived from the filesystem ID. It is maintained for backward compatability.

Return codes

Code	Data	Description
200	groupQuotaTemplate	Group quota template successfully retrieved.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.

Code	Data	Description
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystems/864553F647F8647D0000000000000000/
group-quota-template -X GET -d '{"virtualServerId":3}'
```

Response example

```
HTTP/1.1 200 OK
{
        "groupQuotaTemplate" :
        {
                "filesystemId" : "864553F647F8647D0000000000000000",
                "quota" :
                {
                        "diskUsage" : 0,
                        "diskUsageThreshold" :
                        {
                                "isHard" : true,
                                "limit" : 0,
                                "reset" : 0,
                                "severe" : 0,
                                "warning" : 0
                        },
                        "fileCountThreshold" :
                        {
                                "isHard" : true,
                                "limit" : 0,
                                "reset" : 0,
                                "severe" : 0,
                                "warning" : 0
                        },
                        "fileUsage" : 0,
                        "logEvent" : false,
                        "targetDomain" : "",
                        "targetName" : "",
                        "targetType" : "GROUP"
                },
                "virtualServerId" : 3
        }
                }
```

Chapter 5: File system resource

Chapter 6: File system directory resource

The file system directory resource provides directory management for a file system. A tree clone is a recursive clone of a directory with all its sub-directories and files. The clone is produced instantly like a snapshot, even though the finishing of cloning may take some time.

File system directory object model

The object model describing this resource contains the following objects.

directory

Attribute	JSON Type	Data Type	Description
objectId	string	string	Unique identifier of a file system directory assigned by web services. This ID is not a storage identifier.
displayName	array	array	String array of directory names.

Get root file system directory

Gets a root file system directory.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/filesystems/{id}/directories
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Specifies either the object ID of the file system or the HNAS file system ID to create a directory.

Return codes

Code	Data	Description
200	directory	Array of immediate sub-directories.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example: Using the file system object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/
41413942394239453446443835414644303030303030303030303030303030/directories
```

Response example: Using the file system object ID

```
HTTP/1.1 200 OK
{
    "directories" : [
        {
            "displayName" : [ "//$__NDMP__" ],
            "objectId" : "245f5f4e444d505f5f"
        },
        {
            "displayName" : [ "//$__CFN__" ],
            "objectId" : "245f5f43464e5f5f"
        },
        {
            "displayName" : [ "//sharenamed" ],
            "objectId" : "736573686172656e616d6564"
        }
    ]
}
```

Request example: Using the HNAS storage file system ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/864DC6E1DFB68C210000000000000000/
directories
```

Chapter 6: File system directory resource

Response example: Using the HNAS storage file system ID

```
HTTP/1.1 200 OK
{
   "directories" : [
      {
         "displayName" : [ "//$__NDMP__" ],
         "objectId" : "245f5f4e444d505f5f"
      },
      {
         "displayName" : [ "//$__CFN__" ],
         "objectId" : "245f5f43464e5f5f"
      },
      {
         "displayName" : [ "//sharenamed" ],
         "objectId" : "736573686172656e616d6564"
      }
   ]
}
```

Get a file system directory

Gets a file system directory.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/filesystems/{id}/directories/{directoryObjectId}
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Specifies either the object ID of the file system or the HNAS storage file system ID.
directoryObjectId	URI_PARAM	Y	string	Directory ID.

Return codes

Code	Data	Description
200	directory	Array of immediate subdirectories

Code	Data	Description
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example: Using the file system object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/
41413942394239453446443835414644303030303030303030303030303030/directories/
73657368616e6e6577362f74657374
```

Response example: Using the file system object ID

```
HTTP/1.1 200 OK
{
    "directories" : [
        {
            "displayName" : [ "//seshanew6/test" ],
            "objectId" : "73657368616e6e6577362f74657374"
        }
    ]
        }
```

Request example: Using the HNAS storage file system ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/864DC6E1DFB68C210000000000000000/
directories/73657368616e6e6577362f74657374
```

Response example: Using the HNAS storage file system ID

```
HTTP/1.1 200 OK
{
    "directories" : [
        {
            "displayName" : [ "//seshanew6/test" ],
            "objectId" : "73657368616e6e6577362f74657374"
        }
    ]
          }
```

Chapter 6: File system directory resource

Create a file system directory

Creates a file system directory.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/filesystems/{id}/directories
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Specifies either the object ID of the file system or the HNAS file system ID to create a directory.
directory	BODY	Y	string	Full path of the directory to be created. The path should be in UNIX format.
ensureCreate	BODY	Y	boolean	When set to TRUE, the end result is similar to the mkdir -p command, which allows creating a directory even if its parent directory does not exist. For example, if /example does not exist in the file system, and directory has the value /example/subdirectory and the ensureCreate parameter is set to TRUE, then this operation can succeed.

Return codes

Code	Data	Description
201	directory	File system directory has been successfully created.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 201 indicates that the API did not complete successfully.

Request example: Using the file system object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystems/
4141394239423945344644383541464430303030303030303030303030303030/directories -d
'{"directory":"/new6/test/test4", "ensureCreate":true}' -X POST
```

Response example: Using the file system object ID

```
HTTP/1.1 201 Created
{
   "directory" : {
      "displayName" : [ "//new6/test/test4" ],
      "objectId" : "73657368616e6577362f746573742f7465737434"
   },
   "uri" : "https://172.17.58.108.8444/v7/storage/filesystems/
4141394239423945344644383541464430303030303030303030303030303030/directories/
73657368616e6577362f746573742f7465737434"
}
```

Request example: Using the HNAS storage file system ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystems/864DC6E1DFB68C210000000000000000/
directories -d '{"directory":"/new6/test/test4", "ensureCreate":true}' -X POST
```

Response example: Using the HNAS storage file system ID

```
HTTP/1.1 201 Created
{
   "directory" : {
      "displayName" : [ "//new6/test/test4" ],
      "objectId" : "73657368616e6577362f746573742f7465737434"
```

Chapter 6: File system directory resource

```
    },
   "uri" : "https://172.17.58.108:8444/v7/storage/filesystems/
41413942394239453446443835414644303030303030303030303030303030/directories/
73657368616e6577362f746573742f7465737434"
}
```

Rename a file system directory

Renames a file system directory.

HTTP request syntax (URI)

```
PATCH <base_URI>/v7/storage/filesystems/{id}/directories/{directoryObjectId}
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Specifies either the object ID of the file system or the HNAS file system ID.
directoryObjectId	URI_PARAM	Y	string	Directory object ID.
newDirectory	BODY	Y	string	New absolute path of the directory. The path should be in UNIX format. If this directory exists or is not empty, the operation will fail.

Name	Type	Required	Values	Description
ensureExist	BODY	Y	boolean	When set to TRUE, the end result is similar to the mkdir -p, which allows a directory to be created even though its parent directory does not exist. In this case, if the / destination object does not exist, and the newDirectory variable has the values /destination/ subdirectory and ensureExist is TRUE, the operation will succeed.

Return codes

Code	Data	Description
303	directory	File system directory has been successfully renamed.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 303 indicates that the operation did not complete successfully.

Request example: Using the file system object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/
303730374139433336463844384237423030303030303030303030303030303030/directories/
6d6574726f5f64697232 -X PATCH -d '{"newDirectory": "/xyz_rename",
"ensureExist":true}'
```

Response example: Using the file system object ID

```
HTTP/1.1 303 See Other
{
   "directory" : {
      "displayName" : [ "//xyz_rename" ],
      "objectId" : "6d6574726f5f72656e616d65"
   },
   "uri" :
"https://172.17.11.11:8444/v7/storage/filesystems/
30373037413934333364363844438423742303030303030303030303030303030/directories/
6d6574726f5f72656e616d65"
}
```

Request example: Using the HNAS storage file system ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/filesystems/864DC6E1DFB68C210000000000000000/
directories/6d6574726f5f64697232 -X PATCH -d '{"newDirectory":"/xyz_rename",
"ensureExist":true}'
```

Response example: Using the HNAS storage file system ID

```
HTTP/1.1 303 See Other
{
   "directory" : {
      "displayName" : [ "//xyz_rename" ],
      "objectId" : "6d6574726f5f72656e616d65"
   },
   "uri" :
"https://172.17.11.11:8444/v7/storage/filesystems/
30373037413934333364363844438423742303030303030303030303030303030/directories/
6d6574726f5f72656e616d65"
}
```

Delete a file system directory

Deletes a file system directory using the built in tree-delete functionality, which makes the delete operation appear to happen instantaneously.

HTTP request syntax (URI)

```
DELETE <base_URI>/v7/storage/filesystems/{id}/directories/{directoryObjectId}
```

Chapter 6: File system directory resource

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Specifies either object ID of the file system or HNAS file system ID.
directoryObjectId	URI_PARAM	Y	string	Directory object ID.

Return codes

Code	Data	Description
204	No Data	File system directory has been successfully deleted.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example: Using the file system object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystems/
41414431453441353943463530324534303030303030303030303030303030303030/directories/
2f2f2f2f73657368616e6577362f74657374 -X DELETE
```

Response example: Using the file system object ID

```
HTTP/1.1 204 No Content
```

Request example: Using the HNAS storage file system ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystems/864DC6E1DFB68C210000000000000000/
directories/2f2f2f2f73657368616e6577362f74657374 -X DELETE
```

Response example: Using the HNAS storage file system ID

```
HTTP/1.1 204 No Content
```

Chapter 6: File system directory resource

Get a tree clone job state of a file system directory

Gets a tree clone job state of a file system directory.

HTTP request syntax (URI)

GET *<base_URI>*/v7/storage/filesystems/clone-jobs/{*jobId*}

Parameters

Name	Type	Required	Values	Description
jobId	URI_PARAM	Y	string	ID of the file system directory tree clone job submitted by the Submit a tree clone job of a file system directory operation.

Return codes

Code	Data	Description
200	jobState	File system directory clone job state has been successfully retrieved. The possible values are: ▪ INVALID ▪ QUEUED ▪ RUNNING ▪ ABORTING ▪ FINISHED ▪ ABORTED ▪ ERRORED ▪ ACTIVE ▪ INACTIVE ▪ ALL
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.

Code	Data	Description
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example: Using the file system object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystems/clone-jobs/ba542618-e055-11d0-9dce-
4bee3faaca3b
```

Response example: Using the file system object ID

```
HTTP/1.1 200 Ok
{
    "jobState" : "FINISHED"
}
```

Submit a tree clone job of a file system directory

Submits a tree clone job of a file system directory.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/filesystems/{id}/clone-jobs
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Specifies either the object ID of the file system or the HNAS file system ID.
srcDirectory	BODY	Y	string	Source directory absolute path of the tree clone job. The path should be in UNIX format.

Chapter 6: File system directory resource

Name	Type	Required	Values	Description
dstDirectory	BODY	Y	string	Destination directory absolute path of the tree clone job. The path should be in UNIX format.
ensureCreate	BODY	Y	boolean	Whether to create the destination directory path if it does not exist. When it is false and the destination directory path exists, the operation results in failure.

Return codes

Code	Data	Description
201	jobId	File system directory clone job has been successfully submitted. The value is the ID of the tree clone job.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 201 indicates that the API did not complete successfully.

Request example: Using the file system object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystems/
41414530433937464138383933313036303030303030303030303030303030303030/clone-jobs -d
'{"srcDirectory":"/xyz-src-directory", "dstDirectory":"/xyz-create-directory-0825-
submit-tree-clone-job-dest", "ensureCreate": true}' -X POST
```

Using the file system object ID

```
HTTP/1.1 201 Created
{
```

Chapter 6: File system directory resource

```
    "jobId" : "3019a7b6-f590-11d0-91bb-4bee3faaca3b",
     "uri" : "https://172.17.57.75:8444/v7/storage/filesystems/clone-jobs/3019a7b6-
f590-11d0-91bb-4bee3faaca3b"
}
```

Request example: Using the HNAS storage file system ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystems/864DC6E1DFB68C210000000000000000/
clone-jobs -d '{"srcDirectory":"/xyz-src-directory", "dstDirectory":"/xyz-create-
directory-0825-submit-tree-clone-job-dest", "ensureCreate": true}' -X POST
```

Response example: Using the HNAS storage file system ID

```
HTTP/1.1 201 Created
{
    "jobId" : "3019a7b6-f590-11d0-91bb-4bee3faaca3b",
     "uri" : "https://172.17.57.77:8444/v7/storage/filesystems/clone-jobs/3019a7b6-
f590-11d0-91bb-4bee3faaca3b"
}
```

Abort a tree clone job of a file system

Aborts a tree clone job of a file system.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/filesystems/clone-jobs/{jobId}/abort
```

Parameters

Name	Type	Required	Values	Description
jobId	URI_PARAM	Y	string	ID of the file system directory tree clone job returned by the submit a tree clone job of a file system operation.

Return codes

Code	Data	Description
204	No Data	Clone job has been successfully aborted.
400	No Data	Missing or invalid request contents.

Chapter 6: File system directory resource

Code	Data	Description
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example: Using the file system object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystems/clone-jobs/ba542618-e055-11d0-9dce-
4bee3faaca3b/abort -X POST
```

Response example: Using the file system object ID

```
HTTP/1.1 204 No Content
```

Clone a file system directory

Clones a file system directory.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/filesystems/{id}/clone-directory
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Specifies the object ID of the file system or the HNAS storage file system ID to clone a file system directory.
srcDirectory	BODY	Y	string	Source directory absolute path of the directory to be cloned. The path should be in UNIX format.

Chapter 6: File system directory resource

Name	Type	Required	Values	Description
dstDirectory	BODY	Y	string	Destination directory absolute path of the cloned copy. The path should be in UNIX format.
ensureCreate	BODY	Y	boolean	Indicates whether to create the destination directory if it does not exist. The operation fails when it is false and the destination directory exists.

Return codes

Code	Data	Description
204	No Data	File system directory clone has been done successfully
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example: Using the file system object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystems/
41414530433937464138383933313036303030303030303030303030303030/clone-directory -d
'{"srcDirectory":"/xyz-src-submit-tree-clone-job", "dstDirectory":"/xyz-create-
directory-0825-submit-tree-clone-job-dest", "ensureCreate": true}' -X POST
```

Response example: Using the file system object ID

```
HTTP/1.1 204 No Content
```

Request example: Using the HNAS storage file system ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystems/864DC6E1DFB68C210000000000000000/
clone-directory -d '{"srcDirectory":"/xyz-src-submit-tree-clone-job",
"dstDirectory":"/xyz-create-directory-0825-submit-tree-clone-job-dest",
"ensureCreate": true}' -X POST
```

Response example: Using the HNAS storage file system ID

```
HTTP/1.1 204 No Content
```

Clone a file system file

Clones a file system file.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/filesystems/{id}/clone-file
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string	Specifies either the object ID of the file system or the HNAS storage file system ID to clone a file system file.
srcFile	BODY	Y	string	Source absolute path of the file to be cloned. The path should be in UNIX format.
dstFile	BODY	Y	string	Destination absolute path of the cloned fiile. The path should be in UNIX format.

Return codes

Code	Data	Description
204	No Data	File system file clone completed successfully
400	No Data	Missing or invalid request contents.

Chapter 6: File system directory resource

Code	Data	Description
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example: Using the file system object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystems/
4141453043393746413838393331303630303030303030303030303030303030/clone-file -d
'{"srcFile":"/xyz-src-submit-tree-clone-job/readme","dstFile":"/xyz-0825-clone-file-
dest"}' -X POST
```

Response example: Using the file system object ID

```
HTTP/1.1 204 No Content
```

Request example: Using the HNAS storage file system ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystems/864DC6E1DFB68C210000000000000000/
clone-file -d '{"srcFile":"/xyz-src-submit-tree-clone-job/readme","dstFile":"/xyz-
0825-clone-file-dest"}' -X POST
```

Response example: Using the HNAS storage file system ID

```
HTTP/1.1 204 No Content
```

Chapter 7: File system share resource

The file system share resource provides NFS (network file system) and CIFS (common Internet file system) application layer protocols that enable you to mount a remote file system across a network.

File system share object model

The file system share object model describing this resource contains the following objects.

fileSystemShare

Attribute	JSON Type	Data Type	Description
objectId	string	string	File system share unique identifier.
shareId	string	string	Either the NFS export ID or the CIFS share ID.
name	string	string	File system share name.
virtualServerId	number	ushort	Virtual server ID.
filesystemId	string	string	File system unique identity.
path	string	string	File system path.
settings	object	object	Depending on the share type, can be either CIFSSettings or NFSSettings.

CIFSSettings

Attribute	JSON Type	Data Type	Description
accessConfig	string	string	CIFS share config options (typically it is the IP addresses of the allowed hosts).

Attribute	JSON Type	Data Type	Description
comment	string	string	File system share comment.
userHomeDirectoryPath	string	string	User home directory path.
isScanForVirusesEnabled	boolean	boolean	Enable to scan for virus on this CIFS share.
maxConcurrentUsers	number	int	Maximum concurrent sessions to access the share. The -1 value indicates unlimited concurrent values.
snapshotOption	string	string	Snapshot option. Possible values are: ▪ HIDE_AND_DISABLE_ACCESS ▪ HIDE_AND_ALLOW_ACCESS ▪ SHOW_AND_ALLOW_ACCESS
cacheOption	string	string	Cache option. Possible values are: ▪ MANUAL_CACHING_DOCS ▪ AUTO_CACHING_DOCS ▪ AUTO_CACHING_PROGS ▪ CACHING_OFF
transferToReplicationTargetSetting	string	string	Transfer to replication target setting. Possible values are: ▪ DO_NOT_TRANSFER ▪ TRANSFER

Attribute	JSON Type	Data Type	Description
			▪ USE_FS_DEFAU LT ▪ INVALID
userHomeDirectoryMode	string	string	User home directory mode. Possible values are: ▪ OFF ▪ ADS ▪ USER ▪ HIDDEN_USER ▪ DOMAIN_AND_U SER ▪ UNIX
isFollowSymbolicLinks	boolean	boolean	Indicates whether or not to follow the symbolic links.
isFollowGlobalSymbolicLinks	boolean	boolean	Indicates whether or not to follow the global symbolic links.
isForceFileNameToLowercase	boolean	boolean	Flag to force file name on the share to lower case.
isABEEnabled	boolean	boolean	Indicates whether access-based enumeration (ABE) is enabled. When enabled, ABE filters the contents of an SMB share. Note that enabling ABE may have a negative impact on SMB performance.

NFSSettings

Attribute	JSON Type	Data Type	Description
accessConfig	string	string	NFS export config options. Typically it is the IP addresses of allowed hosts.
snapshotOption	string	string	NFS snap shot option. Possible values are: ▪ SHOW_AND_ALLOW_ACCESS ▪ HIDE_AND_DISABLE_ACCESS ▪ HIDE_AND_ALLOW_ACCESS
localReadCacheOption	string	string	NFS local read cache option. Possible values are: ▪ DISABLED ▪ ENABLED_FOR_ALL_FILES ▪ ENABLED_FOR_TAGGED_FILES ▪ ENABLED_FOR_CVLS
transferToReplicationTargetSetting	string	string	NFS transfer to replication target setting. Possible values are: ▪ DO_NOT_TRANSFER ▪ TRANSFER ▪ USE_FS_DEFAULT ▪ INVALID

CIFSAuthentication

Attribute	JSON Type	Data Type	Description
name	string	string	Name of CIFS access authentication. It is typically a windows user or group name or SID. Example values are: ▪ current owner ▪ everyone ▪ guests ▪ guest ▪ users ▪ administrators ▪ administrator ▪ S-1-0 ▪ S-1-3 ▪ S-1-1-0
encodedName	string	string	Encoded string of name that allows the user access through the URI. The user should not be concerned with encoding methods that might be subject to change in the future.
type	string	string	Type of authentication. Possible values are: ▪ ALIAS ▪ COMPUTER ▪ DELETED ▪ DOMAIN ▪ GROUP ▪ INVALID ▪ UNKNOWN ▪ USER ▪ WELLKNOWN
permission	number	uint	Bit representation of permission for access authentication. Use decimal number. See the explanation in the table below.

CIFS Authentication Permission Explanation

Permission	Binary	Decimal
No permission	0	0
Grant read	1000	8
Grant read+change	11000	24
Grant read+change+full control	111000	56
Deny read	1	1
Deny read+change	11	3
Deny read+change+full control	111	7

Get a file system share

Retrieves a file system share.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/filesystem-shares/{type}/{filesystemShareObjectId}
```

Parameters

Name	Type	Required	Values	Description
filesystemShareObjectId	URI_PARAM	Y	string	ID of the file system share object.
type	URI_PARAM	Y	string	Possible values are cifs or nfs.

Return codes

Code	Data	Description
200	filesystemShare	File system share objects retrieved successfully
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.

Code	Data	Description
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example 1

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystem-shares/nfs/
353a3a3a66366233396131652d303562362d313164302d393136382d35656330333373534353864
```

Response example 1

```
HTTP/1.1 200 OK
{
  "filesystemShare" : {
    "filesystemId" : "75412709E1CB9AAB0000000000000000",
    "objectId" :
"353a3a3a66366233396131652d303562362d313164302d393136382d35656330333373534353864",
    "name" : "/xyz-docu-test-nfs",
    "path" : "/xyz/doc/test/path/0426",
    "settings" : {
      "accessConfig" : "*",
      "localReadCacheOption" : "DISABLED",
      "snapshotOption" : "SHOW_AND_ALLOW_ACCESS",
      "transferToReplicationTargetSetting" : "DO_NOT_TRANSFER"
    },
    "shareId" : "91c13a68-54ba-11d1-911a-4bee3faaca3b",
    "virtualServerId" : 5
  }
}
```

Request example 2

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystem-shares/
cifs/5/353a3a3a64383163313665382d313163642d313164312d393234352d34626565533366616163613
362
```

Response example 2

```
HTTP/1.1 200 OK
{
  "filesystemShare" :{
    "filesystemId" : "7547FEE51E4FEC080000000000000000",
```

```
    "objectId" :
"353a3a3a64383163313665382d313163642d313164312d393234352d3462656533666616163613362",
    "name" : "xyz-test-cifs5",
    "path" : "\\",
    "settings" : {
      "shareId" : "91c13a68-54ba-11d1-911a-4bee3faaca3b",
      "isABEEnabled" : false,
      "accessConfig" : "*",
      "cacheOption" : "MANUAL_CACHING_DOCS",
      "comment" : "xyz comment",
      "isFollowGlobalSymbolicLinks" : false,
      "isFollowSymbolicLinks" : false,
      "isForceFileNameToLowercase" : false,
      "maxConcurrentUsers" : 100,
      "isScanForVirusesEnabled" : false,
      "snapshotOption" : "SHOW_AND_ALLOW_ACCESS",
      "transferToReplicationTargetSetting" : "DO_NOT_TRANSFER",
      "userHomeDirectoryMode" : "OFF",
      "userHomeDirectoryPath" : "",
       "virtualServerId" : 5
    }
  }
}
```

Get file system shares associated with a virtual server

Retrieves file system shares associated with a virtual server.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/virtual-servers/{virtualServerObjectId}/{type}
```

Parameters

Name	Type	Required	Values	Description
virtualServerObjectId	URI_PARAM	Y	number	ID of the virtual server hosting the shares.
type	URI_PARM	Y	string	Type of the file system share. Possible values are cifs or nfs.

Chapter 7: File system share resource

Return codes

Code	Data	Description
200	fileSystemShares	Array of fileSystemShare objects retrieved successfully
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example 1

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/virtual-servers/383a3a3a/cifs
```

Response example 1

```
HTTP/1.1 200 OK
{
  "fileSystemShares":  [
    {
      "filesystemId" : "75412709E1CB9AAB0000000000000000",
      "objectId" :
"353a3a3a36623833613537382d353439342d313164312d393064302d34626565533366616163613362",
      "name" : "user-xyz-test-cifs-01-21-3",
      "path" : "\\mypath-cifs-1",
      "settings" : {
        "accessConfig" : "*",
        "cacheOption" : "MANUAL_CACHING_DOCS",
        "comment" : "xyz comment",
        "isABEEnabled" : false,
        "isFollowGlobalSymbolicLinks" : false,
        "isFollowSymbolicLinks" : false,
        "isForceFileNameToLowercase" : false,
        "isScanForVirusesEnabled" : false,
        "maxConcurrentUsers" : 100,
        "snapshotOption" : "SHOW_AND_ALLOW_ACCESS",
        "transferToReplicationTargetSetting" : "DO_NOT_TRANSFER",
        "userHomeDirectoryMode" : "OFF",
        "userHomeDirectoryPath" : ""
      },
      "shareId" : "6b83a578-5494-11d1-90d0-4bee3faaca3b",
      "virtualServerId" : 5
```

Chapter 7: File system share resource

```
      },
      {
        "filesystemId" : "75412709E1CB9AAB0000000000000000",
        "objectId" :
"353a3a3a36626434336466652d323062382d313164312d393031302d35656333303033373534353864",
        "name" : "user-xyz-test-cifs-11-16-1",
        "path" : "\\mypath-cifs-1",
        "settings" : {
          "accessConfig" : "*",
          "cacheOption" : "MANUAL_CACHING_DOCS",
          "comment" : "xyz comment",
          "isABEEnabled" : false,
          "isFollowGlobalSymbolicLinks" : false,
          "isFollowSymbolicLinks" : false,
          "isForceFileNameToLowercase" : false,
          "isScanForVirusesEnabled" : false,
          "maxConcurrentUsers" : 100,
          "snapshotOption" : "SHOW_AND_ALLOW_ACCESS",
          "transferToReplicationTargetSetting" : "DO_NOT_TRANSFER",
          "userHomeDirectoryMode" : "OFF",
          "userHomeDirectoryPath" : ""
        },
        "shareId" : "6bd43dfe-20b8-11d1-9010-5ec00375458d",
        "virtualServerId" : 5
      }
    ]
}
```

Request example 2

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/virtual-servers/383a3a3a/nfs
```

Response example

```
HTTP/1.1 200 OK
{
  "filesystemShares" : [
    {
      "filesystemId" : "75412709E1CB9AAB0000000000000000",
      "objectId" :
"353a3a3a66366233396131652d303562362d313164302d393136382d35656333303033373534353864",
      "name" : "/xyz*name*with*star",
      "path" : "/xyz-*path",
      "settings" : {
        "accessConfig" : "",
        "localReadCacheOption" : "DISABLED",
        "snapshotOption" : "SHOW_AND_ALLOW_ACCESS",
        "transferToReplicationTargetSetting" : "USE_FS_DEFAULT"
      },
```

Chapter 7: File system share resource

```
      "virtualServerId" : 5
    },
    {
      "filesystemId" : "75412709E1CB9AAB0000000000000000",
      "objectId" :
"353a3a3a65653163393135382d653130322d313164302d396464322d34626565533666616163613362",
      "name" : "/xyz-path-test-exp",
      "path" : "/xyz/test/path",
      "shareId" : "91c13a78-54ba-11d1-911a-4bee3faaca3b",
      "settings" : {
        "accessConfig" : "*",
        "localReadCacheOption" : "DISABLED",
        "snapshotOption" : "SHOW_AND_ALLOW_ACCESS",
        "transferToReplicationTargetSetting" : "USE_FS_DEFAULT"
      },
      "shareId" : "91c13a68-54ba-11d1-931a-4bee3faaca3b",
      "virtualServerId" : 5
    }
  ]
}
```

Create a file system share

Creates a file system share of type CIFS or NFS.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/filesystem-shares/{type}
```

Parameters

Name	Type	Required	Values	Description
name	BODY	Y	string	Share name.
type	URI_PARAM	Y	string	File system share type. The only possible values are cifs or nfs.
virtualServerId	BODY	Y	number	ID of the virtual server to which the file system is assigned.

Chapter 7: File system share resource

Name	Type	Required	Values	Description
filesystemId	BODY	Y	string	ID of the file system that the share belongs to.
fileSystemPath	BODY	Y	string	Absolute filesystem path where the share resides - for NFS exports, the path should be UNIX format, for CIFS shares the path should be Windows format, and the \ character needs to be escaped.
settings	BODY	Y	object	Share setting. Depending on what value type is, the value for a share setting is either the object of the CIFS settings or the object of the NFS settings.

Return codes

Code	Data	Description
201	filesystemShare	File system share object created successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 201 indicates that the API did not complete successfully.

Request example 1

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystem-shares/nfs    -d '{"name":"xyz-test-
nfs-01-21-3","virtualServerId":5, "filesystemId":"75412709E1CB9AAB0000000000000000",
```

Chapter 7: File system share resource

```
"filesystemPath":"/xyz/ test/path/1204","settings":{"accessConfig":"*",
"snapshotOption":"SHOW_AND_ALLOW_ACCESS","localReadCacheOption":"DISABLED",
"transferToReplicationTargetSetting":"DO_NOT_TRANSFER"}}' -X POST
```

Response example 1

```
HTTP/1.1 201 Created
{
  "filesystemShare" : {
    "filesystemId" : "75412709E1CB9AAB0000000000000000",
    "objectId" :
"353a3a3a65646430653862322d353462612d313164312d393131622d3462265653366616163613362",
    "name" : "/xyz-test-nfs-01-21-3",
    "path" : "/xyz/test/path/1204",
    "settings" : {
      "accessConfig" : "*",
      "localReadCacheOption" : "DISABLED",
      "snapshotOption" : "SHOW_AND_ALLOW_ACCESS",
      "transferToReplicationTargetSetting" : "DO_NOT_TRANSFER"
    },
    "shareId" : "edd0e8b2-54ba-11d1-911b-4bee3faaca3b",
    "virtualServerId" : 5
  },
  "uri" : "https://172.17.57.75:8444/v7/storage/filesystem-shares/nfs/
353a3a3a65646430653862322d353462612d313164312d393131622d3462265653366616163613362"
}
```

Request example 2

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystem-shares/cifs -d '{"name":"xyz-test-
cifs-01-21-4","virtualServerId":5,
"filesystemId":"75412709E1CB9AAB0000000000000000","filesystemPath":"\\mypath-cifs-
1","settings":{"accessConfig":"*","comment":"xyz comment",
"userHomeDirectoryPath":"","isScanForVirusesEnabled":false,
"maxConcurrentUsers":100,"snapshotOption":"SHOW_AND_ALLOW_ACCESS",
"cacheOption":"MANUAL_CACHING_DOCS",
"transferToReplicationTargetSetting":"DO_NOT_TRANSFER",
"userHomeDirectoryMode":"OFF","isFollowSymbolicLinks":false,
"isFollowGlobalSymbolicLinks":false,"isForceFileNameToLowercase":false,
"isABEEnabled":false}}' -X POST
```

Response example 2

```
HTTP/1.1 201 Created
{
  "filesystemShare" : {
    "filesystemId" : "75412709E1CB9AAB0000000000000000",
    "objectId" :
"353a3a3a39316331336136382d353462612d313164312d393131612d3462265653366616163613362",
```

Chapter 7: File system share resource

```
      "name" : "xyz-test-cifs-01-21-4",
      "path" : "\\mypath-cifs-1",
      "settings" : {
        "accessConfig" : "*",
        "cacheOption" : "MANUAL_CACHING_DOCS",
        "comment" : "xyz comment",
        "isABEEnabled" : false,
        "isFollowGlobalSymbolicLinks" : false,
        "isFollowSymbolicLinks" : false,
        "isForceFileNameToLowercase" : false,
        "isScanForVirusesEnabled" : false,
        "maxConcurrentUsers" : 100,
        "snapshotOption" : "SHOW_AND_ALLOW_ACCESS",
        "transferToReplicationTargetSetting" : "DO_NOT_TRANSFER",
        "userHomeDirectoryMode" : "OFF",
        "userHomeDirectoryPath" : ""
      },
      "shareId" : "91c13a68-54ba-11d1-911a-4bee3faaca3b",
      "virtualServerId" : 5
    },
  "uri" : "https://172.17.57.75:8444/v7/storage/filesystem-shares/cifs/
353a3a3a39316331336136382d353462612d313164312d393131612d34626565533366616163613362"
}
```

Update a file system share

Updates a file system share.

HTTP request syntax (URI)

```
PATCH <base_URI>/v7/storage/filesystem-shares/{type}/{filesystemshareObjectId}
```

Parameters

Name	Type	Required	Values	Description
filesystemshareObjectId	URI_PARAM	Y	string	Object ID of the file system share.
type	URI_PARAM	Y	string	Possible values are cifs or nfs.
filesystemId	BODY	Y	string	ID of the file system where the share resides.

Name	Type	Required	Values	Description
fileSystemPath	BODY	Y	string	Absolute filesystem path where the share resides - for NFS exports, the path should be UNIX format, for CIFS shares the path should be Windows format, and the \ character needs to be escaped
settings	BODY	Y	object	Share setting. Depending which file system type specified, the object is either NFS settings or CIFS settings.

Return codes

Code	Data	Description
204	No Data	File system share was successfully updated
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example 1

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystem-shares/cifs/
353a3a3a65646430653862322d353462612d313164312d393131622d346265565336661616363362 -d
'{"filesystemId":"7547FEE51E4FEC080000000000000000","filesystemPath":"\\",
"settings":{"accessConfig":"*","comment":"xyz comment","userHomeDirectoryPath":"",
"isScanForVirusesEnabled":false, "maxConcurrentUsers":100,
```

```
"snapshotOption":"SHOW_AND_ALLOW_ACCESS", "cacheOption":"MANUAL_CACHING_DOCS",
"transferToReplicationTargetSetting":"DO_NOT_TRANSFER",
"userHomeDirectoryMode":"OFF","isFollowSymbolicLinks":false,
"isFollowGlobalSymbolicLinks":false,"isForceFileNameToLowerCase":false,
"isABEEnabled":true}}' -X PATCH
```

Response example 1

```
HTTP/1.1 204 No Content
```

Request example 2

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystem-shares/nfs/
353a3a3a65646430653862322d353462612d313164312d393131622d346265653533666616163613362 -d
'{"filesystemId":"75412709E1CB9AAB0000000000000000","filesystemPath":"/xyz/doc/test/
path/0426","settings":{"accessConfig":"*", "snapshotOption":"SHOW_AND_ALLOW_ACCESS",
"localReadCacheOption":"DISABLED",
"transferToReplicationTargetSetting":"DO_NOT_TRANSFER"}}' -X PATCH
```

Response example 2

```
HTTP/1.1 204 No Content
```

Delete a file system share

Deletes a file system share.

HTTP request syntax (URI)

```
DELETE <base_URI>/v7/storage/filesystem-shares/{type}/{filesystemShareObjectId}
```

Parameters

Name	Type	Required	Values	Description
filesystemshareObjectId	URI_PARAM	Y	string	Object ID of the file system share.
type	URI_PARAM	Y	string	Type of the file system share. Possible values are cifs or nfs.

Return codes

Code	Data	Description
204	No Data	File system share deleted successfully
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example 1

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystem-shares/nfs/
353a3a3a656464306653862322d353462612d313164312d393131622d34626565533366616163613362 -X
DELETE
```

Response example 1

```
HTTP/1.1 204 No Content
```

Request example 2

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystem-shares/cifs/
353a3a3a656464306653862322d353462612d513164312d393131622d34626565533366616163613362 -X
DELETE
```

Response example 2

```
HTTP/1.1 204 No Content
```

Get CIFS share access authentications

Retrieves CIFS share access authentications.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/filesystem-shares/cifs/{cifsObjectId}/authentications
```

Parameters

Name	Type	Required	Values	Description
cifsObjectId	URI_PARAM	Y	string	ID of the CIFS share.

Return codes

Code	Data	Description
200	cifsAuthentication	Array of CIFS authentication objects returned which are attached with CIFS share
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystem-shares/cifs/
353a3a3a65646430653862322d353462612d313164312d393131622d346265653366616163613362/
authentications
```

Response example

```
HTTP/1.1 200 OK
{
    "cifsAuthentications" : [
        {
            "encodedName" : "4255494c54494e5c477565737473",
            "name" : "BUILTIN\\Guests",
            "permission" : 56,
            "type" : "ALIAS"
        },
        {
            "encodedName" : "4255494c54494e5c5573657273",
            "name" : "BUILTIN\\Users",
            "permission" : 56,
            "type" : "ALIAS"
        }
```

Chapter 7: File system share resource

```
   ]
 }
```

Add CIFS share access authentications

Adds CIFS share access authentications.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/filesystem-shares/cifs/{cifsObjectId}/authentications
```

Parameters

Name	Type	Required	Values	Description
cifsObjectId	URI_PARAM	Y	string	Object ID of the CIFS share.
cifsAuthentications	BODY	Y	object	Array of CIFS authentication objects to be added to the CIFS share. Each CIFS authentication object should omit the encodedName attribute. See the examples for the correct syntax

Return codes

Code	Data	Description
201	No Data	CIFS share access authentications are added successfully
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 201 indicates that the API did not complete successfully.

Chapter 7: File system share resource

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystem-shares/cifs/
353a3a3a65646430653862322d353462612d313164312d393131622d34626565533666616163613362/
authentications -d '{"cifsAuthentications":[{"type":"GROUP", "permission":56,
"name":"current owner"},{"type":"GROUP", "permission":56,
"name":"Administrators"}]}' -X POST
```

Response example

```
HTTP/1.1 201 Created
    {
    "cifsAuthentications" : [
      {
         "encodedName" : "4255494c54494e5c41646d696e6973747261746f7273",
         "name" : "BUILTIN\\Administrators",
         "permission" : 56,
         "type" : "ALIAS"
      },
      {
         "encodedName" : "4255494c54494e5c43757272656e74204f776e6572",
         "name" : "BUILTIN\\Current Owner",
         "permission" : 56,
         "type" : "ALIAS"
      }
   ],
   "uri" : "172.17.57.88:8444/v7/storage/filesystem-shares/cifs/
363a3a3a39353764393634652d383566302d313164312d393336342d34626565533666616163613362/
authentications"
    }
```

Delete CIFS share access authentication

Deletes a CIFS share access authentication.

HTTP request syntax (URI)

```
DELETE <base_URI>/v7/storage/filesystem-shares/cifs/{cifsObjectId}/authentications/
{encodedName}
```

Parameters

Name	Type	Required	Values	Description
cifsObjectId	URI_PARAM	Y	string	ID of the CIFS share.

Name	Type	Required	Values	Description
encodedName	URI_PARAM	Y	string	Encoded name of the CIFS authentication to be deleted.

Return codes

Code	Data	Description
204	No Data	CIFS share access authentication is deleted successfully
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystem-shares/cifs/
353a3a3a65646430653862322d353462612d313164312d393131622d34626565533666616163613362/
authentications/4255494c54494e5c5573657273 -X DELETE
```

Response example

```
HTTP/1.1 204 No Content
```

Chapter 8: File system snapshot resource

The file system snapshot resource provides management of a snapshot of a file system.

File system snapshot object model

The file system snapshot object model describing this resource contains the following objects.

snapshot

Attribute	JSON Type	Data Type	Description
objectId	string	string	Web services unique assigned identifier of the snapshot. This object ID is not the ID in the storage system.
displayName	string	string	Display name of the snapshot.
filesystemId	string	string	ID of the file system.
state	string	string	State of snapshot. Possible values are: ▪ SNAPSHOT_MANAGEMENT_ST ATE_USABLE ▪ SNAPSHOT_MANAGEMENT_ST ATE_DELETE _PENDING ▪ SNAPSHOT_MANAGEMENT_ST ATE_DELETE _ONGOING ▪ SNAPSHOT_MANAGEMENT_ST ATE_DEAD ▪ SNAPSHOT_MANAGEMENT_ST ATE_DELETE _ROOT_ONODE_PASS

Attribute	JSON Type	Data Type	Description
creationReason	string	string	Reason for snapshot creation. Possible values are: ▪ SNAPSHOT_CREATED_MANUA LLY ▪ SNAPSHOT_CREATED_BY_RUL E ▪ SNAPSHOT_CREATED_FOR_BA CKUP ▪ SNAPSHOT_CREATED_FOR_DA TA_MIGRATION _TARGET ▪ SNAPSHOT_CREATED_BY_VSS ▪ SNAPSHOT_CREATED_BY_CFN ▪ SNAPSHOT_CREATED_BY_HSR ▪ SNAPSHOT_CREATED_BY_JET _API ▪ SNAPSHOT_CREATED_FOR_FS _UTILITY ▪ SNAPSHOT_CREATED_FOR_DE DUPE ▪ SNAPSHOT_CREATED_RESERV ED ▪ SNAPSHOT_CREATED_FOR_MI GRATION _RECOVERY ▪ SNAPSHOT_CREATED_BY_APP LICATION
creationTime	number	int64	Unix time of the creation of the snapshot.

Get file system snapshots

Retrieves all file system snapshots by file system ID filtered by appSearchId.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/filesystem-snapshots/{filesystemId}/{appSearchId}
```

Parameters

Name	Type	Required	Values	Description
filesystemId	URI_PARAM	Y	string	Unique identity of the file system.
appSearchId	URI_PARAM	Y	string	A filter string to be used to filter out snapshots. When it is null, all available snapshots are retrieved. This is the appSearchId used to create a snapshot. User is recommended to give a specific appSearchId to speed up search. If appSearchId is not known or you do not care, a special wildcard value "null" can be used to retrieve all available snapshots, which includes snapshots not created by application. (see example) Therefore caution should be taken when managing them. Note that when the file system is not in the mounted state, an empty result may be returned without errors. When it is not null, only snapshots created with this appSearchId are retrieved. When the file system is not in the mounted state, an error is returned. Warning: This API allows you to retrieve all snapshots, which includes system created ones and modifying or deleting automatically created system snapshots may result in unexpected behavior.

Return codes

Code	Data	Description
200	snapshot	File system snapshot successfully retrieved
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example using special wildcard appSearchId "null" to retrieve all snapshots.

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystem-snapshots/
75412709E1CB9AAB0000000000000000/null -X GET
```

Response example

```
HTTP/1.1 200 OK
{
  "snapshots":
  [
    {
        "creationReason" : "SNAPSHOT_CREATED_BY_HSR",
        "creationTime" : 1469740945,
        "displayName" : "AUTO_SNAPSHOT_22c97d50-e916-11d1-9b15-5ec00375458d_1",
        "filesystemId" : "75412709E1CB9AAB0000000000000000",
        "objectId" :
"373534313237303945314342394141423030303030303030303030303030303a3a3a4155544f5f534e4
15053484f545f32326339376435302d653931362d313164312d396231352d3565563330303337353435386
45
f31",
        "state" : "SNAPSHOT_MANAGEMENT_STATE_USABLE"
    },
    {
        "creationReason" : "SNAPSHOT_CREATED_BY_HSR",
        "creationTime" : 1469740349,
        "displayName" : "AUTO_SNAPSHOT_bfbac044-e914-11d1-9b13-5ec00375458d_1",
        "filesystemId" : "75412709E1CB9AAB0000000000000000",
        "objectId" :
"373534313237303945314342394141423030303030303030303030303030303a3a3a4155544f5f534e4
15053484f545f62666261633034342d653931342d313164312d396231332d3565563330303337353435386
45
f31",
        "state" : "SNAPSHOT_MANAGEMENT_STATE_USABLE"
    }
```

Chapter 8: File system snapshot resource

```
        ]
}
```

Request example with specific appSearchId "Metro" to retrieve all snapshots with "Metro" prefix

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystem-snapshots/
75412709E1CB9AAB0000000000000000/Metro -X GET
```

Response example with specific appSearchId "Metro" to retrieve all snapshots with "Metro" prefix

```
{
   "snapshots" : [
      {
         "creationReason" : "SNAPSHOT_CREATED_BY_APPLICATION",
         "creationTime" : 1585155275,
         "displayName" : "Metro-xyz-fs2-ss-test2",
         "filesystemId" : "AF0BFEDC1CF3BCC70000000000000000",
         "objectId" :
"414630424264544433143463342434337303030303030303030303030303030303a3a3a4d6574726f2d6e6
9636b2d6673322d73732d74657374323a3a3a3a3a3a303a3a3a4f49445f24232140255f56",
         "state" : "SNAPSHOT_MANAGEMENT_STATE_USABLE"
      },
      {
         "creationReason" : "SNAPSHOT_CREATED_BY_APPLICATION",
         "creationTime" : 1585153895,
         "displayName" : "Metro-xyz-fs2-ss-test1",
         "filesystemId" : "AF0BFEDC1CF3BCC70000000000000000",
         "objectId" :
"414630424264544433143463342434337303030303030303030303030303030303a3a3a4d6574726f2d6e6
9636b2d6673322d73732d74657374313a3a3a3a3a3a303a3a3a4f49445f24232140255f56",
         "state" : "SNAPSHOT_MANAGEMENT_STATE_USABLE"
      }
   ]
}
```

Get a file system snapshot

Retrieves a file system snapshot.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/filesystem-snapshots/{snapshotObjectId}
```

Parameters

Name	Type	Required	Values	Description
snapshotObjectId	URI_PARAM	Y	string	Unique object identity of the file system snapshot.

Return codes

Code	Data	Description
200	snapshot	File system snapshot successfully retrieved
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.239.123:8444/v7/storage/filesystem-snapshots/
3836344231323238414242313434414330303030303030303030303030303030303a3a3a4155544f5f534e
415053484f545f5441524745545f353a3a3a3a3a3a303a3a3a4f49445f24232140255f56
```

Response example

```
HTTP/1.1 200 OK
 {
    "snapshot" :
        {
                "creationReason" : "SNAPSHOT_CREATED_BY_HSR",
                "creationTime" : 1466555788,
                "displayName" : "AUTO_SNAPSHOT_TARGET_5",
                "filesystemId" : "864B1228ABB144AC0000000000000000",
                "objectId" :
"3836344231323238414242313434414330303030303030303030303030303030303a3a3a4155544f5f534e4
15053484f545f5441524745545f353a3a3a3a3a3a303a3a3a4f49445f24232140255f56",
                "state" : "SNAPSHOT_MANAGEMENT_STATE_USABLE"
        }
 }
```

Chapter 8: File system snapshot resource

Create a file system snapshot

Creates a file system snapshot.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/filesystem-snapshots
```

Parameters

Name	Type	Required	Values	Description
filesystemId	BODY	Y	string	ID of the file system to take the snapshot.
displayName	BODY	Y	string	Display name of the snapshot to create.
appSearchId	BODY	Y	string	Application specific filter ID to speed up searching for future retrieval. TIP: This user-defined ID becomes the prefix of the snapshot display name when created. And it will speed up in subsequent queries.

Return codes

Code	Data	Description
201	snapshot	File system snapshot has been successfully created.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 201 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystem-snapshots -d
'{"displayName":"snapshot-0917", "filesystemId":"748154E1A75B56600000000000000000",
"appSearchId":"xyz"}' -X POST
```

Response example

```
HTTP/1.1 201 Created
{
   "snapshot" : {
      "creationReason" : "SNAPSHOT_CREATED_BY_APPLICATION",
      "creationTime" : 1476398487,
      "displayName" : "xyz-snapshot-0917",
      "filesystemId" : "748154E1A75B56600000000000000000",
      "objectId" :
"373438313534453141373542353636303030303030303030303030303030303030303a3a3a4d6574726f6f2d787
97a2d736e617073686f742d30393137",
      "state" : "SNAPSHOT_MANAGEMENT_STATE_USABLE"
   },
   "uri" : "https://172.17.57.75:8444/v7/storage/filesystem-snapshots/
373438313534453141373542353636303030303030303030303030303030303030303a3a3a4d6574726f6f2d7879
7a2d736e617073686f742d30393137"
}
```

Delete a file system snapshot

Deletes a file system snapshot.

HTTP request syntax (URI)

```
DELETE <base_URI>/v7/storage/filesystem-snapshots/{snapshotObjectId}
```

Parameters

Name	Type	Required	Values	Description
snapshotObjectId	URI_PARAM	Y	number	Object ID of the snapshot.

Return codes

Code	Data	Description
204	No Data	File system snapshot has been deleted successfully
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/filesystem-snapshots/
37343831353445314137354235363630303030303030303030303030303a3a3a4d6574726f6f2d78
797a2d736e617073686f6f742d30393137 -X DELETE
```

Response example

```
HTTP/1.1 204 No Content
```

Get a file system snapshot size

Retrieves the preserved space that is in use by a snapshot. Note that if the snapshot is deleted, all of the space preserved for it may not be freed since some of the space may also be shared with older snapshots.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/filesystem-snapshots/{snapshotObjectId}/snapshot-size
```

Parameters

Name	Type	Required	Values	Description
snapshotObjectId	URI_PARAM	Y	string	Unique object identity of the file system snapshot.

Chapter 8: File system snapshot resource

Return codes

Code	Data	Description
200	snapshotSize	File system snapshot size successfully retrieved
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.239.123:8444/v7/storage/filesystem-snapshots/
41414531443338333030314638343733330303030303030303030303030303030303a3a3a4d657
4726f2d617473656e672d736e617073686f742d30303031/snapshot-size
```

Response example

```
HTTP/1.1 200 OK
 {
     "snapshotSize" : 8388608
 }
```

Chapter 9: iSCSI resource

The iSCSI resource enables you to create and manage iSCSI targets and logical units. iSCSI is an IP storage networking standard that links storage systems. By forwarding iSCSI commands over the network, iSCSI enables data transfer both over local area and wide area networks.

An iSCSI target is a node or instance running on an iSCSI server. Typically, the target is a disk drive. Targets act as destinations to which management settings and information is sent by an iSCSI initiator. An iSCSI logical unit, or LUN, is a logical representation of a physical device on the iSCSI target, identified by the LUN number. An iSCSI initiator establishes a connection with the LUN through a series of sessions with the target.

iSCSI object model

The iSCSI object model describing this resource contains the following objects.

iSCSI

Attribute	JSON Type	Data Type	Description
objectId	string	string	ID of the resource.
iSCSIId	string	string	iSCSI identifier name.
virtualServerId	number	ushort	Virtual server ID.
comment	string	string	Note about the iSCSI object.
accessConfig	string	string	Access configuration of the iSCSI.
isAuthenticationEnabled	boolean	boolean	Authentication flag enabled.
secret	string	string	iSCSI secret.
iSCSILogicalUnits	object	object	Array of iSCSI logical unit objects.
globalUniqueName	string	string	Global unique name.
isRegenerateGUN	boolean	boolean	IsRegenerateGUN flag.

iSCSILogicalUnit

Attribute	JSON Type	Data Type	Description
logicalUnitId	string	string	Logical unit ID.
logicalUnit	number	ushort	The LUN that is assigned to the logical unit.

Get iSCSI targets

Retrieves all iSCSI targets.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/iscsi-targets
```

Return codes

Code	Data	Description
200	iscsi	iSCSI target retrieved successfully
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.58.119:8444/v7/storage/iscsi-targets
```

Response example

```
HTTP/200 OK
{
   "iSCSITargets" : [
      {
         "accessConfig" : "172.17.58.37.0",
         "comment" : "test Modisy ISCSI Target",
```

```
        "globalUniqueName" : "iqn.2015-01.com.hds.sie:rickevs-58-112.iscsidev-0",
        "iSCSIId" : "iSCSIDEV-0-new",
        "iSCSILogicalUnits" : [],
        "objectId" : "313a3a3a69534353494445562d302d6e6577",
        "isAuthenticationEnabled" : true,
        "isRegenerateGUN" : false,
        "secret" : "testISCSILDEV0",
        "virtualServerId" : 1
    },
    {
        "accessConfig" : "172.17.37.0\n172.17.58.0",
        "comment" : "testModify ISCSI",
        "globalUniqueName" : "iqn.2015-01.com.hds.sie:rickevs-58-112.iscsidev-1",
        "iSCSIId" : "iSCSIDEV-1",
        "iSCSILogicalUnits" : [],
        "objectId" : "313a3a3a69534353494445562d31",
        "isAuthenticationEnabled" : true,
        "isRegenerateGUN" : false,
        "secret" : "testISCSIDEV11",
        "virtualServerId" : 1
    },
    {
        "accessConfig" : "*",
        "comment" : "test delete iscsilu",
        "globalUniqueName" : "iqn.2016-04.com.hds.sie:rickevs-58-112.iscsi-test",
        "iSCSIId" : "iscsi-test",
        "iSCSILogicalUnits" : [],
        "objectId" : "313a3a3a69736373692d74657374",
        "isAuthenticationEnabled" : true,
        "isRegenerateGUN" : false,
        "secret" : "secret",
        "virtualServerId" : 1
    },
    {
        "accessConfig" : "172.17.239.171",
        "comment" : "test for windows",
        "globalUniqueName" : "iqn.2016-04.com.hds.sie:rickevs-58-112.xyz-iscsi-for-
windows",
        "iSCSIId" : "xyz-iscsi-for-windows",
        "iSCSILogicalUnits" : [
            {
                "logicalUnitId" : "xyz-iscsi-lu-test0111",
                "logicalUnit" : 0
            }
        ],
        "objectId" : "313a3a3a6e69636b2d69736373692d666f722d77696e646f7773",
        "isAuthenticationEnabled" : true,
        "isRegenerateGUN" : false,
        "secret" : "windows",
        "virtualServerId" : 1
    }
```

Chapter 9: iSCSI resource

```
    ]
  }
```

Get an iSCSI target

Retrieves an iSCSI target.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/iscsi-targets/{iscsiObjectId}
```

Parameters

Name	Type	Required	Values	Description
iscsiObjectId	URI_PARAM	Y	string	iSCSI object ID.

Return codes

Code	Data	Description
200	iscsi	iSCSI target retrieved successfully
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.58.117:8444/v7/storage/iscsi-targets/
383a3a3a767361746973682d69736373692d746573742d35
```

Response example

```
HTTP/200 OK
{
    "iSCSITarget" : {
      "accessConfig" : "",
      "comment" : "IscsiTargetCreate2",
```

Chapter 9: iSCSI resource

```
      "globalUniqueName" : "iqn.2016-09.xyzdo1875:xyzvs4014.xyzis6554",
      "iSCSIId" : "xyzIS6554",
      "iSCSILogicalUnits" : [],
      "objectId" : "31323a3a3a6d6574726f495336353534",
      "isAuthenticationEnabled" : true,
      "isRegenerateGUN" : false,
      "secret" : "HitachiHitachi",
      "virtualServerId" : 12
   }
}
```

Create an iSCSI target

Creates an iSCSI target.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/iscsi-targets/
```

Parameters

Name	Type	Required	Values	Description
secret	BODY	Y	string	iSCSI secret.
comment	BODY	Y	string	Comment at iSCSI creation.
iSCSIName	BODY	Y	string	Name of the iSCSI target.
accessConfig	BODY	Y	string	Access configuration.
enableAuthentication	BODY	Y	boolean	Enable authentication flag.
iSCSIlogicalUnit	BODY	N	array	Array of iSCSILogicalUnitId values. Note that failure to add the logical units to the target will not stop creation of the iSCSI target.

Name	Type	Required	Values	Description
virtualServerId	BODY	Y	number	Virtual server ID.

Return codes

Code	Data	Description
201	iscsi	iSCSI target object created successfully
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 201 indicates that the API did not complete successfully.

> **Note:** The underlying virtual server should have an iSCSI domain name assigned to create an iSCSI Target.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.58.119:8444/v7/storage/iscsi-targets -d'{"secret":"HitachiHitachi",
"comment":"IscsiTargetCreate2","iSCSIName":"xyz-iscsi-test-5", "accessConfig":"",
"enableAuthentication":true, "iSCSIlogicalUnit":[], "virtualServerId":8}' -X POST
```

Response example

```
HTTP/1.1 201 Created
{
        "iSCSITarget" : {
            "accessConfig" : "",
            "comment" : "IscsiTargetCreate2",
            "globalUniqueName" : "iqn.2016-09.xyzdo1875:xyzvs4014.xyzis6554",
            "iSCSIId" : "xyz-iscsi-test-5",
            "iSCSIlogicalUnit" : [],
            "objectId" : "31323a3a3a6d6574726f495336353534",
            "isAuthenticationEnabled" : true,
                    "isRegenerateGUN" : false,
            "secret" : "HitachiHitachi",
            "virtualServerId" : 8
             },
          "uri" : "https://172.17.57.178:8444/v7/storage/iscsi-targets/
```

Chapter 9: iSCSI resource

```
31323a3a3a6d6574726f6f495336353534"
}
```

Update an iSCSI target

Updates an iSCSI target.

HTTP request syntax (URI)

```
PATCH <base_URI>/v7/storage/iscsi-targets/{iscsiObjectId}
```

Parameters

Name	Type	Required	Values	Description
iscsiObjectId	URI_PARAM	Y	string	iSCSI object ID.
secret	BODY	Y	string	iSCSI secret.
comment	BODY	Y	string	Comment at iSCSI creation.
accessConfig	BODY	Y	string	Access configuration.
enableAuthentication	BODY	Y	boolean	Enable authentication flag.
newAlias	BODY	Y	string	New iSCSI alias name.

Return codes

Code	Data	Description
303	iscsi	iSCSI target updated successfully
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 303 indicates that the operation did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.58.117:8444/v7/storage/iscsi-targets/
383a3a3a767361746973682d69736373692d746573742d32 -X PATCH -d'{"secret":"hellohere",
"comment":"user1-testing","newAlias":"vinh-iscsi- 255","accessConfig":"",
"enableAuthentication":true}'
```

Response example

```
HTTP/1.1 303 See Other
{
   "iSCSITarget" : {
      "accessConfig" : "",
      "comment" : "user1-testing",
      "globalUniqueName" : "iqn.2016-10.vsatish-iscsi-domain-1:satish.vinh-iscsi-
testing",
      "iSCSIId" : "vinh-iscsi-255",
      "iSCSILogicalUnits" : [],
      "isAuthenticationEnabled" : true,
      "isRegenerateGUN" : false,
      "objectId" : "383a3a3a76696e682d69736373692d323535",
      "secret" : "hellohere",
      "virtualServerId" : 8
   },
   "uri" : "https://172.17.239.120:8444/v7/storage/iscsi-targets/
383a3a3a76696e682d69736373692d323535"
}
```

Delete an iSCSI target

Deletes an iSCSI target.

HTTP request syntax (URI)

```
DELETE <base_URI>/v7/storage/iscsi-targets/{iscsiObjectId}
```

Parameters

Name	Type	Required	Values	Description
iscsiObjectId	URI_PARAM	Y	string	ID of the iSCSI target.

Return codes

Code	Data	Description
204	No Data	iSCSI target deleted successfully
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.58.117:8444/v7/storage/iscsi-targets/
383a3a3a767361746973682d69736373692d746573742d35 -X DELETE
```

Response example

```
HTTP/1.1 204 No Content
```

Get all iSCSI logical units associated with an iSCSI target

Obtains all iSCSI logical units associated with an iSCSI target.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/iscsi-targets/{iscsiObjectId}/iscsi-logical-units
```

Parameters

Name	Type	Required	Values	Description
iscsiObjectId	URI_PARAM	Y	string	iSCSI object ID.

Return codes

Code	Data	Description
200	iscsi	Get all iSCSI logical unit associated with an iSCSI target successfully
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.58.117:8444/v7/storage/iscsi-targets/
383a3a3a767361746973682d69736373692d746573742d37/iscsi-logical-units
```

Response example

```
HTTP/1.1 201 OK
{
      "iSCSILogicalUnits" : [
        {
          "logicalUnitId" : "xyz-LU-2000",
          "logicalUnit" : 1
        }
      ]
}
```

Add an iSCSI logical unit to an iSCSI target

Adds an iSCSI logical unit to an iSCSI target. The logical unit is assigned the next free LUN ID associated with the iSCSI target.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/iscsi-targets/{iscsiObjectId}/iscsi-logical-units
```

Chapter 9: iSCSI resource

Parameters

Name	Type	Required	Values	Description
iscsiObjectId	URI_PARAM	Y	string	iSCSI object ID.
logicalUnitId	BODY	Y	string	iSCSI logical unit ID.

Return codes

Code	Data	Description
201	No Data	iSCSI logical unit added to an iSCSI target successfully
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 201 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.58.117:8444/v7/storage/iscsi-targets/
383a3a3a767361746973682d69736373692d746573742d37/iscsi-logical-units -X POST -d
'{"logicalUnitId":"xyz-iscsi-lu-test0"}'
```

Response example

```
HTTP/1.1 201 No Content
```

Delete an iSCSI logical unit from an iSCSI target

Deletes the iSCSI logical unit association with the iSCSI target. The actual logical unit it not deleted.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/iscsi-targets/{iscsiObjectId}/iscsi-logical-units/
{iscsiLuId}
```

Parameters

Name	Type	Required	Values	Description
iscsiObjectId	URI_PARAM	Y	string	iSCSI object ID.
iscsiLuId	URI_PARAM	Y	string	iSCSI logical unit ID.

Return codes

Code	Data	Description
204	No Data	iSCSI logical unit successfully deleted from iSCSI target
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.58.117:8444/v7/storage/iscsi-targets/
383a3a3a767361746973682d69736373692d746573742d37/iscsi-logical-units/xyz- iscsi-lu-
test0 -X DELETE
```

Response example

```
HTTP/1.1 204 No Content
```

Chapter 10: iSCSI logical unit resource

The iSCSI logical unit (LUN) resource enables you to create and manage iSCSI logical units. iSCSI is an IP storage networking standard that links storage systems. By forwarding iSCSI commands over the network, iSCSI enables data transfer both over local area and wide area networks.

An iSCSI logical unit, or LUN, is a logical representation of a physical device on the iSCSI target, identified by the LUN number. An iSCSI initiator establishes a connection with the LUN through a series of sessions with the target.

iSCSI logical unit object model

The iSCSI logical unit object model describing this resource contains the following objects.

iSCSILogicalUnit

Attribute	JSON Type	Data Type	Description
objectId	string	string	Resource ID.
iSCSILogicalUnitId	string	string	iSCSI logical unit identifier name.
virtualServerId	number	uint	Virtual server ID.
iSCSITargetIds	array	array	iSCSI ID.
comment	string	string	Comment.
filesystemId	string	string	File system ID.
path	string	string	Path of the iSCSI logical unit file.
sizeInBytes	number	uint	Size of the iSCSI logical unit in bytes.
isMounted	boolean	boolean	Flag to indicate if the logical unit is mounted.
isBoundToTarget	boolean	boolean	Flag to indicate if the logical unit is bound to the iSCSI target.

Attribute	JSON Type	Data Type	Description
percentageCreatedx100	number	uint	A value that is 100 times the percent value of the object that has been created.

Get iSCSI logical units

Retrieves all iSCSI Logical Units.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/iscsi-logical-units
```

Return codes

Code	Data	Description
200	iscsiLogicalUnit	Array of the iSCSI logical unit objects
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.58.117:8444/v7/storage/iscsi-logical-units
```

Response example

```
HTTP/1.1 200 OK
{
  "iSCSILogicalUnits": [
    {
      "comment": "new lu",
      "filesystemId": "075E75858FF8164E0000000000000000",
      "iSCSILogicalUnitId": "new bob",
      "iSCSITargetIds": [
```

Chapter 10: iSCSI logical unit resource

```
      "updatedAlias"
    ],
    "isBoundToTarget": true,
    "isMounted": true,
    "objectId": "313a3a3a6e657720626f623a3a3a3a3a303a3a3a4f49445f24232140255f56",
    "path": "/hellobob.iscsi",
    "percentageCreatedx100": 10000,
    "sizeInBytes": 567890,
    "virtualServerId": 1
  },
  {
    "comment": "hello",
    "filesystemId": "075E75858FF8164E0000000000000000",
    "iSCSILogicalUnitId": "clonedLu",
    "iSCSITargetIds": [
      "updatedName"
    ],
    "isBoundToTarget": true,
    "isMounted": true,
    "objectId":
"313a3a3a636c6f6e65644c753a3a3a3a3a303a3a3a4f49445f24232140255f56",
    "path": "/folder/clonedlun.iscsi",
    "percentageCreatedx100": 10000,
    "sizeInBytes": 524288000,
    "virtualServerId": 1
  }
]
}
```

Get an iSCSI logical unit

Retrieves an iSCSI logical unit.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/iscsi-logical-units/{iscsiLuObjectId}
```

Parameters

Name	Type	Required	Values	Description
iscsiLuObjectId	URI_PARAM	Y	string	iSCSI logical unit object ID.

Return codes

Code	Data	Description
200	iscsiLogicalUnit	iSCSI logical unit object
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.58.117:8444/v7/storage/iscsi-logical-units/
383a3a3a767361746973682d69736373692d6c752d7465737430
```

Response example

```
HTTP/1.1 200 OK
{
    "iSCSILogicalUnit" : {
        "comment" : "xyz-curl-test",
        "filesystemId" : "AACF4FA49143751B00000000000000000",
        "iSCSILogicalUnitId" : "xyz-iscsi-lu-test0",
        "iSCSITargetIds" : [],
        "objectId" : "383a3a3a767361746973682d69736373692d6c752d7465737430",
        "isBountToTarget" : false,
        "isMounted" : true,
        "path" : "/folder/xyz-lu.iscsi",
        "percentageCreatedx100" : 10000,
        "sizeInBytes" : 2000,
        "virtualServerId" : 8
    }
}
```

Create an iSCSI logical unit

Creates an iSCSI logical unit.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/iscsi-logical-units
```

Parameters

Name	Type	Required	Values	Description
path	BODY	Y	string	Absolute path to the iSCSI logical unit file, in UNIX format.
comment	BODY	Y	string	Comment at the iSCSI logical unit creation.
sizeInBytes	BODY	Y	number	Size of the iSCSI logical unit in bytes.
filesystemId	BODY	Y	string	File system ID.
virtualServerId	BODY	Y	number	Virtual server ID.
iscsiLogicalUnitId	BODY	Y	string	iSCSI logical unit identifier name.

Return codes

Code	Data	Description
201	iscsiLogicalUnit	iSCSI logical unit object created successfully
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 201 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.58.117:8444/v7/storage/iscsilogical-units -d'{ "comment": "xyz-curl-
test", "filesystemId":"AACF4FA49143751B0000000000000000", "iSCSILogicalUnitId":
"xyz-iscsi-lutest0", "path": "/folder/xyz-lu.iscsi", "sizeInBytes": 2000,
"virtualServerId": 8}' -X POST
```

Response example

```
HTTP/1.1 201 Created
{
   "iSCSILogicalUnit" : {
      "comment" : "xyz-curl-test",
      "filesystemId" : "AACF4FA49143751B0000000000000000",
      "iSCSILogicalUnitId" : "xyz-iscsi-lu-test0",
      "iSCSITargetIds" : [],
      "objectId" : "383a3a3a767361746973682d69736373692d6c752d7465737430",
      "isBoundToTarget" : false,
      "isMounted" : false,
      "path" : "/folder/xyz-lu.iscsi",
      "percentageCreatedx100" : 0,
      "sizeInBytes" : 2000,
      "virtualServerId" : 8
   },
   "uri" : "https://172.17.58.117:8444/v7/storage/iscsi-logical-units/
383a3a3a767361746973682d69736373692d6c752d7465737430"
}
```

Update an iSCSI logical unit

Updates an iSCSI logical unit.

HTTP request syntax (URI)

```
PATCH <base_URI>/v7/storage/iscsi-logical-units/{iscsiLuObjectId}
```

Parameters

Name	Type	Required	Values	Description
iscsiLuObjectId	URI_PARAM	Y	string	iSCSI logical unit object ID.
comment	BODY	N	string	Comment for update.
newAlias	BODY	N	string	New alias for the iSCSI logical unit.
sizeInBytes	BODY	N	number	Size of the iSCSI logical unit in bytes.

Chapter 10: iSCSI logical unit resource

Return codes

Code	Data	Description
303	iscsiLogicalUnit	iSCSI logical unit successfully updated
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 303 indicates that the operation did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.58.117:8444/v7/storage/iscsi-logical- units/
383a3a3a767361746973682d69736373692d6c752d7465737430 -X PATCH -d '{"newAlias":"vinh-
logical-units-1"}'
```

Response example

```
HTTP/1.1 303 See Other
{
   "iSCSILogicalUnit" : {
      "comment" : "xyz-logical-units-4352",
      "filesystemId" : "75435847AB2D8EDA0000000000000000",
      "iSCSILogicalUnitId" : "vinh-logical-units-1",
      "iSCSITargetIds" : [],
      "isBountToTarget" : false,
      "isMounted" : true,
      "objectId" : "373a3a3a76696e682d6c6f676963616c2d756e6974732d31",
      "path" : "/folder/xyz-lu.iscsi",
      "precentageCreatedx100" : 10000,
      "sizeInBytes" : 2000,
      "virtualServerId" : 7
   },
   "uri" : "https://172.17.239.120:8444/v7/storage/iscsi-logical-units/
373a3a3a76696e682d6c6f676963616c2d756e6974732d31"
}
```

Mount an iSCSI logical unit

Mounts an iSCSI logical unit.

Chapter 10: iSCSI logical unit resource

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/iscsi-logical-units/{iscsiLuObjectId}/mount
```

Parameters

Name	Type	Required	Values	Description
iscsiLuObjectId	URI_PARAM	Y	string	iSCSI logical unit object ID.

Return codes

Code	Data	Description
204	No Data	N/A
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC" -X
POST https://172.17.58.117:8444/v7/storage/iscsi-logical-units/
383a3a3a767361746973682d69736373692d6c752d7465737430/mount
```

Response example

```
HTTP/1.1 204 OK
```

Unmount an iSCSI logical unit

Unmounts an iSCSI logical unit.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/iscsi-logical-units/{iscsiLuObjectId}/unmount
```

Parameters

Name	Type	Required	Values	Description
iscsiLuObjectId	URI_PARAM	Y	string	iSCSI logical unit object ID.

Return codes

Code	Data	Description
204	No Data	N/A
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC" -X
POST https://172.17.58.117:8444/v7/storage/iscsi-logical-units/
383a3a3a767361746973682d69736373692d6c752d7465737430/unmount
```

Response example

```
HTTP/1.1 204 OK
```

Clone an iSCSI logical unit

Clones an iSCSI logical unit.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/iscsi-logical-units/{iscsiLuObjectId}/clone
```

Parameters

Name	Type	Required	Values	Description
iscsiLuObjectId	URI_PARAM	Y	string	iSCSI logical unit object ID.
newLogicalUnitAlias	BODY	Y	string	Alias of the iSCSI LU being cloned.
newLogicalUnitPath	BODY	Y	string	Path of iSCSI LU being cloned The new path name must not exist, and the directory of the path must exist. That is, if newLogicalUnitPath has a value of /absolute/path/clone_file, then directory /absolute/path must exist, and absolute/path/clone_file must not exist. Add the .iscsi extension (lower case) to the filename of the pathname to avoid accidental user deletion.

Return codes

Code	Data	Description
204	No Data	N/A
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.58.117:8444/v7/storage/iscsi-logical-units/
383a3a3a767361746973682d69736373692d74657374/clone -X POST -
d'{ "newLogicalUnitAlias":"vsatish-clone-test", "newLogicalUnitPath":"/folder2/xyz-
lu.iscsi"}'
```

Response example

```
HTTP/1.1 204 OK
```

Delete an iSCSI logical unit

Deletes an iSCSI logical unit, and the file associated with it. Note that the file contains any data that may have been written to the iSCSI logical unit.

HTTP request syntax (URI)

```
DELETE <base_URI>/v7/storage/iscsi-logical-units/{iscsiLuObjectId}
```

Parameters

Name	Type	Required	Values	Description
iscsiLuObjectId	BODY	Y	string	iSCSI logical unit object ID.

Return codes

Code	Data	Description
204	No Data	N/A
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.58.117:8444/v7/storage/iscsi-logical-units/
383a3a3a767361746973682d69736373692d6c752d7465737430 -X DELETE
```

Response example

```
HTTP/1.1 204 OK
```

Chapter 11: Node resource

Node resources are the physical servers that make up the Hitachi NAS Platform. Nodes can be standalone or part of a cluster. The node resource enables you to obtain detailed information about a cluster node. In addition, it allows you to retrieve information about the events that have occurred on the node.

Node object model

The object model describing this resource contains the following objects.

node

Attribute	JSON Type	Data Type	Description
objectId	string	string	Node object unique identifier. This ID is not the HNAS storage node ID.
nodeId	number	uint	Node ID on a device. The node ID starts from 1.
status	string	string	Status of the node. Possible values are: ▪ UNKNOWN ▪ ONLINE ▪ NOT_UP ▪ DEAD ▪ INVALID ▪ UP ▪ DORMANT
UUID	string	string	UUID of the node.
firmwareVersion	string	string	Firmware version of the node.
model	string	string	Model of the node.
name	string	string	Name of the node.

Attribute	JSON Type	Data Type	Description
ipAddresses	array	array	IP addresses of the node.
serial	string	string	Serial number of the node

event

Attribute	JSON Type	Data Type	Description
eventId	number	int64	Event ID.
severity	string	string	Event severity, one of the following levels: ▪ INFO ▪ SEVERE ▪ WARNING ▪ CRITICAL
category	string	string	Event category, one of the following levels: ▪ SYSTEM ▪ SECURITY ▪ FILESYSTEM
timeStamp	number	int64	Time of the event in clock ticks since an epoch.
text	string	string	Event description.
clusterNodeId	number	ushort	Cluster node ID.

Get nodes

Retrieves nodes in a cluster.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/nodes
```

Return codes

Code	Data	Description
200	node	Array of nodes retrieved successfully.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/nodes
```

Response example

```
HTTP/1.1 200 OK
{
  "nodes": [
    {
      "UUID": "bbd85dd2-3016-11d5-9001-040403000307",
      "firmwareVersion": "13.9.6809.00",
      "ipAddresses": [
        "192.168.0.190",
        "10.251.48.1"
      ],
      "model": "N800",
      "name": "G800-443037-1",
      "nodeId": 1,
      "objectId": "313a3a3a3a3a3a303a3a3a4f49445f24232140255f56",
      "serial": "443037",
      "status": "ONLINE"
    },
    {
      "UUID": "b2e122e0-3016-11d5-9001-f03c9491cf19",
      "firmwareVersion": "13.9.6809.00",
      "ipAddresses": [
        "192.168.0.191",
        "10.251.52.1"
      ],
      "model": "N800",
      "name": "G800-443037-2",
      "nodeId": 2,
      "objectId": "323a3a3a3a3a3a303a3a3a4f49445f24232140255f56",
      "serial": "443037",
      "status": "ONLINE"
```

Chapter 11: Node resource

```
    }
  ]
}
```

Get a node

Retrieves detailed information about a node in a cluster. A node identifier identifies the node.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/nodes/{id}
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string/number	Specifies either a node object ID or an HNAS storage cluster node ID.

Return codes

Code	Data	Description
200	node	Individual node information retrieved successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example: Using a node object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/nodes/
313a3a3a3a3a3a303a3a3a4f49445f24232140255f56
```

Response example: Using a node object ID

```
HTTP/1.1 200 OK
{
  "node": {
    "UUID": "bbd85dd2-3016-11d5-9001-040403000307",
    "firmwareVersion": "13.9.6809.00",
    "ipAddresses": [
      "192.168.0.190",
      "10.251.48.1"
    ],
    "model": "N800",
    "name": "G800-443037-1",
    "nodeId": 1,
    "objectId": "313a3a3a3a3a3a3a303a3a3a4f49445f24232140255f56",
    "serial": "443037",
    "status": "ONLINE"
  }
}
```

Request example: Using an HNAS storage node ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/nodes/1
```

Response example: Using an HNAS storage node ID

```
HTTP/1.1 200 OK
{
  "node": {
    "UUID": "bbd85dd2-3016-11d5-9001-040403000307",
    "firmwareVersion": "13.9.6809.00",
    "ipAddresses": [
      "192.168.0.190",
      "10.251.48.1"
    ],
    "model": "N800",
    "name": "G800-443037-1",
    "nodeId": 1,
    "objectId": "313a3a3a3a3a3a3a303a3a3a4f49445f24232140255f56",
    "serial": "443037",
    "status": "ONLINE"
  }
}
```

Get node events

Retrieves a node's events. Events include violated thresholds, alarms, traps, notifications, floods, reboots, and various errors. Note that this function can potentially return a lot of data, depending on the number of items in the event log.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/nodes/{id}/events
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string/number	Specifies either a node object ID or an HNAS storage cluster node ID.

Return codes

Code	Data	Description
200	event	Array of node events retrieved successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example: Using a node object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/nodes/313a3a3a4f49445f24232140255f56/events
```

Response example: Using a node object ID

```
HTTP/1.1 200 OK
{
    "events" : [
        {
            "category" : "SYSTEM",
            "clusterNodeId" : 1,
```

Chapter 11: Node resource

```
      "eventId" : 7217,
      "severity" : "INFO",
      "text" : "An administrator has licensed (allowed access to) SD 121.",
      "timeStamp" : 1485977870
    },
    {
      "category" : "SYSTEM",
      "clusterNodeId" : 1,
      "eventId" : 8517,
      "severity" : "INFO",
      "text" : "Unspanned SD 121 is healthy, primary and licensed.",
      "timeStamp" : 1485977870
    }
  ]
}
```

Request example: Using an HNAS storage node ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/nodes/1/events
```

Response example: Using an HNAS storage node ID

```
HTTP/1.1 200 OK
{
  "events" : [
    {
      "category" : "SYSTEM",
      "clusterNodeId" : 1,
      "eventId" : 7217,
      "severity" : "INFO",
      "text" : "An administrator has licensed (allowed access to) SD 121.",
      "timeStamp" : 1485977870
    },
    {
      "category" : "SYSTEM",
      "clusterNodeId" : 1,
      "eventId" : 8517,
      "severity" : "INFO",
      "text" : "Unspanned SD 121 is healthy, primary and licensed.",
      "timeStamp" : 1485977870
    }
  ]
}
```

Set user-defined event

Sets a user-defined event.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/nodes/user-event
```

Parameters

Name	Type	Required	Values	Description
severity	BODY	Y	string	Severity of event. Current supported values are: ▪ INFORMATION ▪ WARNING ▪ SEVERE
message	BODY	Y	string	The event message.

Return codes

Code	Data	Description
204	No Data	User-defined event posted successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/nodes/user-event -X POST -d
'{"severity":"INFORMATION","message":"test1"}'
```

Response example

```
HTTP/1.1 204 No Content
```

Chapter 12: Object replication resource

Object replication replicates a snapshot of a file system to another server, typically remote, to provide backup and recovery of the source data.

Object replication object model

The object model describing this resource contains the following objects.

objectReplicationReport

Attribute	JSON Type	Data Type	Description
bytesRemaining	number	integer	Number of remaining bytes.
bytesTransferred	number	integer	Number of transferred bytes.
destinationSnapshotName	string	string	Destination snapshot name.
endTime	number	integer	End time.
extraStatusMessage	string	string	Extra status message.
log	string	string	Log message.
objectsRemaining	number	integer	Number of remaining objects.
objectsTransferred	number	integer	Number of transferred objects.
policyId	string	string	Object replication policy ID.
policyName	string	string	Object replication policy name.
policyTarget	string	string	IP address of target of object replication policy.

Attribute	JSON Type	Data Type	Description
policyTargetFilesystemLabel	string	string	Filesystem name of target of object replication policy.
sourceIncrementalBaseSnapshotName	string	string	Source incremental base snapshot name.
sourceSnapshotName	string	string	Source snapshot name.
startTime	number	integer	Start time.
status	string	string	Status. Current supported values are: ▪ COMPLETE ▪ RUNNING ▪ FAILED ▪ DRY_RUN_CO MPLETED ▪ DRY_RUN_RU NNING ▪ DRY_RUN_FAI LED ▪ FAILED_SERV ER_RESET ▪ DRY_RUN_FAI LED_SERVER_ RESET

Start an object replication

Starts an object replication.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/object-replications/start
```

Chapter 12: Object replication resource

Parameters

Name	Type	Required	Values	Description
objectReplicationPolicyId	BODY	Y	string	ID of an object replication policy.
type	BODY	Y	string	Run type of this operation. Current supported values are: ▪ NORMAL ▪ DRY ▪ DETAILED ▪ SIMPLE (NORMAL means a real object replication running type. All the rest are pseudo running tests with different level of detail of output information).

Return codes

Code	Data	Description
204	No Data	An object replication started successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/object-replications/start -X POST -d
```

Chapter 12: Object replication resource

```
'{"objectReplicationPolicyId":"c7d44f66-767c-11d2-906a-49e1bb864b23",
"type":"NORMAL"}'
```

Response example

```
HTTP/1.1 204 No Data
```

Stop an object replication

Stops an object replication.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/object-replications/stop
```

Parameters

Name	Type	Required	Values	Description
objectReplicationPolicyId	BODY	Y	string	ID of an object replication policy.

Return codes

Code	Data	Description
204	No Data	An object replication stopped successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/object-replications/stop -X POST  -d
'{"objectReplicationPolicyId":"c7d44f66-767c-11d2-906a-49e1bb864b23"}'
```

Response example

```
HTTP/1.1 204 No Data
```

Get an object replication last report

Gets an object replication last report.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/object-replications/last-report
```

Parameters

Name	Type	Required	Values	Description
objectReplicationPolicyId	BODY	Y	string	ID of an object replication policy.
logIncluded	BODY	N	boolean	True for log included.

Return codes

Code	Data	Description
200	objectReplicationReport	An object replication last report retrieved successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/object-replications/last-report -X GET -d
'{"objectReplicationPolicyId":"ca2e73f2-d5d7-11d3-9cff-49e1bb864b23"}'
```

Response example

```
HTTP/1.1 200 OK
   {
        "objectReplicationReport" :
                {
                        "bytesRemaining" : 0,
                        "bytesTransferred" : 0,
                        "destinationSnapshotName" : "",
                        "endTime" : 1524783484,
                        "extraStatusMessage" : "Failed to negotiate object
replication with target",
                        "log" : "",
                        "objectsRemaining" : 0,
                        "objectsTransferred" : 0,
                        "policyId" : "ca2e73f2-d5d7-11d3-9cff-49e1bb864b23",
                        "policyName" : "xyz_orp2",
                        "policyTarget" : "172.17.239.168",
                        "policyTargetFilesystemLabel" : "fakeFs222",
                        "sourceIncrementalBaseSnapshotName" : "",
                        "sourceSnapshotName" : "AUTO_SNAPSHOT_ca2e73f2-d5d7-11d3-
9cff-49e1bb864b23_1",
                        "startTime" : 1524783454,
                        "status" : "FAILED"
                }
    }
```

Get all object replication reports

Gets all object replication reports.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/object-replications/reports
```

Parameters

Name	Type	Required	Values	Description
objectReplicationPolicyId	BODY	Y	string	ID of an object replication policy.
logIncluded	BODY	N	boolean	True for log included.

Chapter 12: Object replication resource

Return codes

Code	Data	Description
200	objectReplicationReports	All object replication reports retrieved successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/object-replications/reports -X GET -d
'{"objectReplicationPolicyId":"ca2e73f2-d5d7-11d3-9cff-49e1bb864b23"}'
```

Response example

```
HTTP/1.1 200 OK
{
        "objectReplicationReports" :
          [
                {
                  "bytesRemaining" : 0,
                  "bytesTransferred" : 0,
                  "destinationSnapshotName" : "",
                  "endTime" : 1524612161,
                  "extraStatusMessage" : "Failed to negotiate object
replication with target",
                  "log" : "",
                  "objectsRemaining" : 0,
                  "objectsTransferred" : 0,
                  "policyId" : "ca2e73f2-d5d7-11d3-9cff-49e1bb864b23",
                  "policyName" : "xyz_orp2",
                  "policyTarget" : "172.17.239.168",
                  "policyTargetFilesystemLabel" : "fakeFs222",
                  "sourceIncrementalBaseSnapshotName" : "",
                  "sourceSnapshotName" : "",
                  "startTime" : 1524612131,
                  "status" : "DRY_RUN_FAILED"
                },
                {
                  "bytesRemaining" : 0,
```

Chapter 12: Object replication resource

```
                    "bytesTransferred" : 0,
                    "destinationSnapshotName" : "",
                    "endTime" : 1524620621,
                    "extraStatusMessage" : "Failed to negotiate object
replication with target",
                    "log" : "",
                    "objectsRemaining" : 0,
                    "objectsTransferred" : 0,
                    "policyId" : "ca2e73f2-d5d7-11d3-9cff-49e1bb864b23",
                    "policyName" : "xyz_orp2",
                    "policyTarget" : "172.17.239.168",
                    "policyTargetFilesystemLabel" : "fakeFs222",
                    "sourceIncrementalBaseSnapshotName" : "",
                    "sourceSnapshotName" : "",
                    "startTime" : 1524620591,
                    "status" : "DRY_RUN_FAILED"
                }
            ]
}
```

Get the object replication listening port

Gets the object replication listening port.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/object-replications/port
```

Return codes

Code	Data	Description
200	objectReplicationListeningPort	The object replication listening port retrieved successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/object-replications/port
```

Response example

```
HTTP/1.1 200 OK
{
          "objectReplicationListeningPort" : 59550
}
```

Set the object replication listening port

Sets an object replication listening port.

HTTP request syntax (URI)

```
PUT <base_URI>/v7/storage/object-replications/port
```

Parameters

Name	Type	Required	Values	Description
port	BODY	Y	number	An unsigned short value (0 – 65535).

Return codes

Code	Data	Description
204	No Data	The object replication listening port set successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/object-replications/port -X PUT -d
'{"port":1004}'
```

Response example

```
HTTP/1.1 204 No Content
```

Chapter 13: Object replication policy resource

Object replication policy resources are policies for object replication.

Object replication policy object model

The object model describing this resource contains the following objects.

objectReplicationPolicy

Attribute	JSON Type	Data Type	Description
objectId	string	string	Object replication policy object unique identifier.
destinationFilesystemName	string	string	Destination file system name.
destinationIPAddress	string	string	Destination IP address.
destinationPort	number	integer	Destination port number.
destinationSnapshotRuleName	string	string	Destination snapshot rule name.
id	string	string	Object replication policy ID.
name	string	string	Object replication policy name.
sourceFilesystemId	string	string	Source filesystem ID.
sourceSnapshotRuleName	string	string	Source snapshot rule name.
sourceVirtualServerId	number	integer	Source virtual server ID.

Get object replication policies

Gets object replication policies.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/object-replication-policies
```

Return codes

Code	Data	Description
200	objectReplicationPolicies	A list of object replication policies retrieved successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/object-replication-policies
```

Response example

```
HTTP/1.1 200 OK
{
        "objectReplicationPolicies" :
        [
                {
                        "destinationFilesystemName" : "ErinObjDest2",
                        "destinationIPAddress" : "172.17.59.48",
                        "destinationPort" : 59550,
                        "destinationSnapshotRuleName" : "",
                        "id" : "1c2e1672-872c-11d2-9519-49e1bb864b23",
                        "name" : "objRmtPolice",
                        "objectId" :
"31633265313637322d383732632d313164322d393531392d343965531626238636346232333a3a3a303a3a3
a4f49445f24232140255f56",
                        "sourceFilesystemId" : "3B7A998FB7FA575B0000000000000000",
                        "sourceSnapshotRuleName" : "",
                        "sourceVirtualServerId" : 3
                },
                {
                        "destinationFilesystemName" : "ErinDestObj3",
                        "destinationIPAddress" : "172.17.58.122",
                        "destinationPort" : 59550,
                        "destinationSnapshotRuleName" : "",
                        "id" : "13be959c-729c-11d2-9665-49e1bb864b23",
                        "name" : "objRmtPolice",
```

Chapter 13: Object replication policy resource

```
                            "objectId" :
"31336265393539632d373239632d313164322d393636352d343965316262383836346232333a3a3a303a3a3
a4f49445f24232140255f56",
                            "sourceFilesystemId" : "3B6BB41DC647FC060000000000000000",
                            "sourceSnapshotRuleName" : "",
                            "sourceVirtualServerId" : 4
                },
            {
                            "destinationFilesystemName" : "fakeFs123",
                            "destinationIPAddress" : "172.17.239.141",
                            "destinationPort" : 65535,
                            "destinationSnapshotRuleName" : "",
                            "id" : "ca2e73f2-d5d7-11d3-9cff-49e1bb864b23",
                            "name" : "user1_objRepPolicy2",
                            "objectId" :
"63613265373366322d643564372d313164332d396366662d343965316262383836346232333a3a3a303a3a3
a4f49445f24232140255f56",
                            "sourceFilesystemId" : "3B6E4F23B465547700000000000000000",
                            "sourceSnapshotRuleName" : "",
                            "sourceVirtualServerId" : 4
                }
        ]
}
```

Get an object replication policy

Gets an object replication policy.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/object-replication-policies/{id}
```

Parameters

Name	Type	Required	Values	Description
id	URI	Y	string/number	ID of an object replication policy object.

Return codes

Code	Data	Description
200	objectReplicationPolicy	An object replication policy retrieved successfully.

Chapter 13: Object replication policy resource

Code	Data	Description
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/object-replication-policies/
63326261363039652d643564382d313164332d396430332d3439653162623836346232333a3
a3a303a3a3a4f49445f24232140255f56
```

Response example

```
HTTP/1.1 200 OK
{
        "objectReplicationPolicy" :
        {
                "destinationFilesystemName" : "fakeFs123",
                "destinationIPAddress" : "172.17.239.141",
                "destinationPort" : 65535,
                "destinationSnapshotRuleName" : "",
                "id" : "c2ba609e-d5d8-11d3-9d03-49e1bb864b23",
                "name" : "abc_objRepPolicy3",
                "objectId" :
"63326261363039652d643564382d313164332d396430332d3439653162623836346232333a3a3a303a3a3
a4f49445f24232140255f56",
                "sourceFilesystemId" : "3B6E4F23B4655477000000000000000",
                "sourceSnapshotRuleName" : "",
                "sourceVirtualServerId" : 4
        }
}
```

Create an object replication policy

Creates an object replication policy.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/object-replication-policies
```

Parameters

Name	Type	Required	Values	Description
filesystemId	BODY	Y	string	File system ID.
name	BODY	Y	string	Name of the object replication policy.
destinationIPAddress	BODY	Y	string	Destination host IPv4 address.
destinationFilesystemName	BODY	Y	string	Destination host filesystem name.
destinationPort	BODY	N	number	Destination host port. If ommitted, 59550 is used as the default port.
sourceSnapshotRuleName	BODY	N	string	Source host snapshot rule name.
destinationSnapshotRuleName	BODY	N	string	Destination host snapshot rule name.

Return codes

Code	Data	Description
201	objectReplicationPolicy	An object replication policy created successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 201 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/object-replication-policies -X POST -d
'{"filesystemId":"3B6E4F23B46554770000000000000000","name":"abc_objRepPolicy3",
"destinationIPAddress":"172.17.239.141", "destinationFilesystemName":"fakeFs123",
"destinationPort":65535}'
```

Response example

```
HTTP/1.1 201 Created
{
        "objectReplicationPolicy" :
        {
                "destinationFilesystemName" : "fakeFs123",
                "destinationIPAddress" : "172.17.239.141",
                "destinationPort" : 65535,
                "destinationSnapshotRuleName" : "",
                "id" : "c2ba609e-d5d8-11d3-9d03-49e1bb864b23",
                "name" : "abc_objRepPolicy3",
                "objectId" :
"63326261363039652d643564382d313164332d396430332d34396531162623836346232333a3a3a303a3a3
a4f49445f24232140255f56",
                "sourceFilesystemId" : "3B6E4F23B46554770000000000000000",
                "sourceSnapshotRuleName" : "",
                "sourceVirtualServerId" : 4
        },
        "uri" : "https://172.17.11.11:8444/v7/storage/object-replication-policies/
63326261363039652d643564382d313164332d396430332d34396531162623836346232333a3a3a303a3a3a
4f49445f24232140255f56"
}
```

Modify an object replication policy

Modifies an object replication policy.

HTTP request syntax (URI)

```
PATCH <base_URI>/v7/storage/object-replication-policies/{id}
```

Parameters

Name	Type	Required	Values	Description
id	URI	Y	string	ID of an object replication policy object.

Name	Type	Required	Values	Description
name	BODY	Y	string	Object replication policy name.
destinationPort	BODY	N	number	Destination host port. A new value will invalidate all existing policies.
sourceSnapshotRuleName	BODY	N	string	Source host snapshot rule name.
destinationSnapshotRuleName	BODY	N	string	Destination host snapshot rule name.
destinationIPAddress	BODY	Y	string	Destination host IPv4 address.
destinationFilesystemName	BODY	Y	string	Destination host filesystem name.

Return codes

Code	Data	Description
200	objectReplicationPolicy	An object replication policy updated successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/object-replication-policies/
63326261363039652d643564382d313164332d396430332d34396531626238363462323333a3a3a303a3a
```

```
3a4f49445f24232140255f56 -X PATCH -d '{"name":"xyz_orp2", "destinationPort":63452,
"destinationIPAddress":"172.17.239.168", "destinationFilesystemName":"fakeFs222"}'
```

Response example

```
HTTP/1.1 200 OK
{
        "objectReplicationPolicy" :
        {
                "destinationFilesystemName" : "fakeFs222",
                "destinationIPAddress" : "172.17.239.168",
                "destinationPort" : 63452,
                "destinationSnapshotRuleName" : "",
                "id" : "ca2e73f2-d5d7-11d3-9cff-49e1bb864b23",
                "name" : "xyz_orp2",
                "objectId" :
"63613265373366322d643564372d313164332d396366662d3439653162623836346232333a3a3a303a3a3
a4f49445f24232140255f56",
                "sourceFilesystemId" : "3B6E4F23B465547700000000000000000",
                "sourceSnapshotRuleName" : "",
                "sourceVirtualServerId" : 4
        }
}
```

Delete an object replication policy

Deletes an object replication policy.

HTTP request syntax (URI)

```
DELETE <base_URI>/v7/stroage/object-replication-policies/{id}
```

Parameters

Name	Type	Required	Values	Description
id	URI	Y	string/number	ID of an object replication policy object.

Return codes

Code	Data	Description
204	No Data	An object replication policy was deleted successfully.

Chapter 13: Object replication policy resource

Code	Data	Description
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/object-replication-policies/
63326261363039652d643564382d313164332d396430332d34396531626238366346232333a3
a3a303a3a3a4f49445f24232140255f56 -X DELETE
```

Response example

```
HTTP/1.1 204 No Content
```

Modify snapshot rule name of an object replication policy

Modifies the snapshot rule name of an object replication policy.

HTTP request syntax (URI)

```
PATCH <base_URI>/v7/storage/object-replication-policies/{id}/snapshot-rule-name
```

Parameters

Name	Type	Required	Values	Description
id	URI	Y	string	ID of an object replication policy object.
sourceSnapshotRuleName	BODY	Y	string	Source snapshot rule name.
destinationSnapshotRuleName	BODY	Y	string	Destination snapshot rule name.

Return codes

Code	Data	Description
200	objectReplicationPolicy	An object replication policy was updated successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/object-replication-policies/
63326261363039652d643564382d313164332d396430332d34396531626238363462323333a3a3a303a3a
3a4f49445f24232140255f56/snapshot-rule-name -X PATCH -d
'{"sourceSnapshotRuleName":"abc","destinationSnapshotRuleName":"xyz"}'
```

Response example

```
HTTP/1.1 200 OK
 {
      "objectReplicationPolicy" :
      {
            "destinationFilesystemName" : "fakeFs222",
            "destinationIPAddress" : "172.17.239.168",
            "destinationPort" : 63452,
            "destinationSnapshotRuleName" : "xyz",
            "id" : "ca2e73f2-d5d7-11d3-9cff-49e1bb864b23",
            "name" : "xyz_orp2",
            "objectId" :
"63613265373366322d643564372d313164332d396366662d34396531626238363462323333a3a3a303a3a
3a4f49445f24232140255f56",
            "sourceFilesystemId" : "3B6E4F23B4655477000000000000000000",
            "sourceSnapshotRuleName" : "abc",
            "sourceVirtualServerId" : 4
      }
}
```

Chapter 14: Object replication schedule resource

Object replication schedule resources are schedules of object replication.

Object replication schedule object model

The object model describing this resource contains the following objects.

objectReplicationSchedule

Attribute	JSON Type	Data Type	Description
objectId	string	string	Object replication schedule object unique identifier.
enabled	boolean	boolean	Object replication schedule enabled or not.
finalRunSchedule	number	unsigned integer	Time of final run schedule.
nextRunSchedule	number	unsigned integer	Time of next run schedule.
interval	number	integer	Time interval between schedules.
unit	string	string	Interval unit. Current supported values are: ▪ MINUTE ▪ HOUR ▪ DAY ▪ WEEK ▪ MONTH
id	number	integer	Object replication schedule ID.
policyId	string	string	Object replication policy ID which the schedule is associated with.

Attribute	JSON Type	Data Type	Description
type	string	string	Interval schedule type. Current supported values are: ▪ PERIODIC ▪ CONTINUOUS ▪ ONCE_ONLY ▪ DRY_RUN_DETAILED

Get object replication schedules

Gets object replication schedules.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/object-replication-schedules
```

Parameters

Name	Type	Required	Values	Description
virtualServerId	BODY	Y	number	ID of the virtual server where object replication policies and schedules reside.

Return codes

Code	Data	Description
200	objectReplicationSchedules	A list of object replication schedules was retrieved successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Chapter 14: Object replication schedule resource

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/object-replication-schedules -X GET -d
'{"virtualServerId":4}'
```

Response example

```
HTTP/1.1 200 OK
{
        "objectReplicationSchedules" :
        [
                {
                        "enabled" : false,
                        "finalRunSchedule" : 1524785999,
                        "id" : 1,
                        "interval" : 7,
                        "nextRunSchedule" : 1525380799,
                        "objectId" :
"33666134326565342d373338612d313164322d393664662d34396531626238363462232333a3a3a343a3a3
a313a3a3a303a3a3a4f49445f24232140255f56",
                        "policyId" : "3fa42ee4-738a-11d2-96df-49e1bb864b23",
                        "type" : "PERIODIC",
                        "unit" : "DAY"
                },
                {
                        "enabled" : true,
                        "finalRunSchedule" : 4294967295,
                        "id" : 2,
                        "interval" : 0,
                        "nextRunSchedule" : 4294967295,
                        "objectId" :
"34323834326264382d643262302d313164332d396138392d34396531626238363462232333a3a3a343a3a3
a323a3a3a303a3a3a4f49445f24232140255f56",
                        "policyId" : "42842bd8-d2b0-11d3-9a89-49e1bb864b23",
                        "type" : "ONCE_ONLY",
                        "unit" : "MINUTE"
                }
        ]
}
```

Get an object replication schedule

Gets an object replication schedule.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/object-replication-schedules/{id}
```

Parameters

Name	Type	Required	Values	Description
id	URI	Y	string/number	ID of an object replication schedule object.

Return codes

Code	Data	Description
200	objectReplicationSchedule	An object replication schedule was retrieved successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/object-replication-schedules/
63613265373366322d643564372d313164332d396366662d3439653316262383 6346232333a3
a3a343a3a3a313a3a3a303a3a3a4f49445f24232140255f56
```

Response example

```
HTTP/1.1 200 OK
{
        "objectReplicationSchedule" :
        {
                "enabled" : true,
                "finalRunSchedule" : 4294967295,
                "id" : 2,
                "interval" : 0,
                "nextRunSchedule" : 4294967295,
                "objectId" :
"6337643434663636362d373637632d313164322d393036612d3439653316262383 6346232333a3a3a343a3a3
```

Chapter 14: Object replication schedule resource

```
a323a3a3a303a3a3a4f49445f24232140255f56",
                "policyId" : "c7d44f66-767c-11d2-906a-49e1bb864b23",
                "type" : "DRY_RUN_DETAILED",
                "unit" : "MINUTE"
        }
}
```

Create an object replication run-once schedule

Creates an object replication run-once schedule.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/object-replication-schedules/run-once-schedule
```

Parameters

Name	Type	Required	Values	Description
objectReplicationPolicyObjectId	BODY	Y	string	Object ID of an object replication policy.
nextRunSchedule	BODY	Y	number	Time of next run schedule.
testOnly	BODY	N	boolean	If true, no real replication schedule is run. Default value is false.

Return codes

Code	Data	Description
201	objectReplicationSchedule	An object replication run-once schedule was created successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.

Chapter 14: Object replication schedule resource

Code	Data	Description
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 201 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/object-replication-schedules/run-once-schedule
-X POST -d
'{"objectReplicationPolicyObjectId":"343a3a3a63376434346636362d373637632d31
3164322d393036612d343965316262383836346232333a3a3a303a3a3a4f49445f24232140255f56",
"nextRunSchedule":167777}'
```

Response example

```
HTTP/1.1 201 Created
{
        "objectReplicationSchedule" :
        {
                "enabled" : true,
                "finalRunSchedule" : 4294967295,
                "id" : 4,
                "interval" : 0,
                "nextRunSchedule" : 167777,
                "objectId" :
"63376434346636362d373637632d313164322d393036612d343965316262383836346232333a3a3a343a3a3a303a3a3a4f49445f24232140255f56",
                "policyId" : "c7d44f66-767c-11d2-906a-49e1bb864b23",
                "type" : "ONCE_ONLY",
                "unit" : "MINUTE"
        }
}
```

Create an object replication periodic schedule

Creates an object replication periodic schedule.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/object-replication-schedules/periodic-schedule
```

Parameters

Name	Type	Required	Values	Description
objectReplicationPolicyObjectId	BODY	Y	string	Object ID of an object replication policy.
interval	BODY	Y	number	Periodic interval.
unit	BODY	Y	string	Interval unit. Current supports values are: ▪ MINUTE ▪ HOUR ▪ DAY ▪ WEEK ▪ MONTH
noEarlyThan	BODY	N	number	Equivalent to the nextRunSchedule value, which indicates when the policy will next be run. If omitted a default value of 0 will be used.
noLaterThan	BODY	N	number	Equivalent to the finalRunSchedule value, which indicates the final time the policy will be execuated. If omitted, the schedule will continue indefinately.

Chapter 14: Object replication schedule resource

Name	Type	Required	Values	Description
isPeriodic	BODY	N	boolean	True indicates periodic; False indicates continuous. Default is TRUE.

Return codes

Code	Data	Description
201	objectReplicationSchedule	An object replication periodic schedule was created successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 201 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/object-replication-schedules/periodic-schedule
-X POST -d '{"interval":1, "unit":"DAY",
"objectReplicationPolicyObjectId":"343a3a3a63376434346636362d373637632d3131
64322d393036612d343965316262383636346232333a3a3a303a3a3a4f49445f24232140255f56"}'
```

Response example

```
HTTP/1.1 201 Created
{
        "objectReplicationSchedule" :
        {
                "enabled" : true,
                "finalRunSchedule" : 4294967295,
                "id" : 5,
                "interval" : 1,
                "nextRunSchedule" : 4294967295,
                "objectId" :
"63376434346636362d373637632d313164322d393036612d343965316262383836346232333a3a3a343a3a3
a353a3a3a303a3a3a4f49445f24232140255f56",
```

```
            "policyId" : "c7d44f66-767c-11d2-906a-49e1bb864b23",
            "type" : "PERIODIC",
            "unit" : "DAY"
        }
}
```

Update an object replication schedule

Updates an object replication schedule.

HTTP request syntax (URI)

```
PATCH <base_URI>/v7/storage/object-replication-schedules/{id}
```

Parameters

Name	Type	Required	Values	Description
id	URI	Y	string/ number	ID of an object replication schedule object.
enabled	BODY	Y	boolean	True to enable; false otherwise.
interval	BODY	Y	number	Schedule interval.
unit	BODY	Y	string	Unit of time of schedule interval. Current supported values are: ▪ MINUTE ▪ HOUR ▪ DAY ▪ WEEK ▪ MONTH
nextRunSchedule	BODY	Y	number	Time of the next run schedule.
finalRunSchedule	BODY	Y	number	Time of the final run schedule.

Chapter 14: Object replication schedule resource

Name	Type	Required	Values	Description
type	BODY	Y	string	Type of schedule. Current supported values are: ▪ PERIODIC ▪ CONTINUOUS ▪ ONCE_ONLY ▪ DRY_RUN_DETAILED If type is ONCE_ONLY, the interval and unit values are not used so they can be anything.

Return codes

Code	Data	Description
204	No Data	An object replication schedule was updated successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/object-replication-schedules/
63376434346636362d373637632d313164322d393036612d343965316262383634623233333a3
a3a343a3a3a353a3a3a303a3a3a4f49445f24232140255f56 -X PATCH -d '{"interval":2,
"enabled":false, "unit":"MONTH", "type":"ONCE_ONLY", "nextRunSchedule":1234567,
"finalRunSchedule":2234568}'
```

Chapter 14: Object replication schedule resource

Response example

```
HTTP/1.1 204 No Content
```

Delete an object replication schedule

Deletse an object replication schedule.

HTTP request syntax (URI)

```
DELETE <base_URI>/v7/storage/object-replication-schedules/{id}
```

Parameters

Name	Type	Required	Values	Description
id	URI	Y	string/number	ID of an object replication schedule object.

Return codes

Code	Data	Description
204	No Data	An object replication schedule was deleted successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/object-replication-schedules/
63326261363039652d643564382d313164332d396430332d343965533162623836346232333a3
a3a303a3a3a4f49445f24232140255f56 -X DELETE
```

Response example

```
HTTP/1.1 204 No Content
```

Chapter 15: Snapshot rules resource

Rules for creating new snapshots. Snapshots can be automatically created using snapshot rules associated with snapshot schedules.

Snapshot rules object model

The object model describing this resource contains the following objects.

filesystemSnapshotRule

Attribute	JSON Type	Data Type	Description
objectId	string	string	ID of the snapshot rule.
filesystemId	string	string	File system ID that the snapshot rule is created for.
isValid	boolean	boolean	TRUE if the snapshot rule is valid.
name	string	string	Name of the snapshot rule.
queueSize	number	uint	Maximum number of snapshots can be created by this rule.
virtualServerId	number	integer	ID of the virtual server where the snapshot rule belongs.

filesystemSnapshotSchedule

Attribute	JSON Type	Data Type	Description
objectId	string	string	Object ID of the snapshot schedule.
emails	string	string	Emails to receive notification.
isValid	boolean	boolean	TRUE if the snapshot schedule is valid.
time	string	string	UNIX cron job time format.

Attribute	JSON Type	Data Type	Description
handle	number	uint	Index into an array of schedules associated with a snapshot rule.
virtualServerId	number	integer	ID of the virtual server where the snapshot schedule belongs.

Get snapshot rules

Gets all snapshot rules that belong to a virtual server ID.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/virtual-servers/{virtualServerObjectId}/snapshot-rules
```

Parameters

Name	Type	Required	Values	Description
virtualServerObjectId	URI_PARAM	Y	string	ID of the virtual server object to which the snapshot rules belong.

Return codes

Code	Data	Description
200	filesystemSnapshotRule	Retrieved one or more file system snapshot rules successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.239.120:8444/v7/storage/virtual-servers/373a3a3a/snapshot-rules
```

Response example

```
HTTP/1.1 200 OK
{
    "filesystemSnapshotRules" :
        [
                {
                        "filesystem" :
                        {
                                "blockSize" : 0,
                                "capacity" : 0,
                                "expansionLimits" : 0,
                                "filesystemId" : "",
                                "freeCapacity" : 0,
                                "isDedupeEnabled" : false,
                                "isDedupeSupported" : false,
                                "isLogicalCapacityFreeCapacityValid" : false,
                                "isLogicalCapacityValid" : false,
                                "isNDMPRecoveryTarget" : false,
                                "isNonStrictWORM" : false,
                                "isReadCached" : false,
                                "isReadOnly" : false,
                                "isSysLocked" : false,
                                "isThinProvisioningEnabled" : false,
                                "isThinProvisioningEnabledValid" : false,
                                "isTrueSparseFileEnabled" : false,
                                "isTrueSparseFileEnabledValid" : false,
                                "isUnlimitedExpansion" : false,
                                "isWORM" : false,
                                "label" : "",
                                "logicalCapacity" : 0,
                                "logicalFreeCapacity" : 0,
                                "objectId" : "3a3a3a303a3a3a4f49445f24232140255f56",
                                "status" : "FAILING",
                                "storagePoolId" : 18446744073709551615,
                                "usedCapacity" : 0,
                                "virtualServerId" : 65535
                        },
                        "filesystemId" : "3B6E4F23B4655477000000000000000000",
                        "isValid" : true,
                        "name" : "rule1",
                        "objectId" :
"343a3a3a334236453446323342343635353437373030303030303030303030303030303030303a3a3a6e69636
b2d6f626a2d7265702d7372632d73732d72756c65313a3a3a303a3a3a4f49445f24232140255f56",
                        "queueSize" : 4,
                        "virtualServerId" : 4
```

Chapter 15: Snapshot rules resource

```
                },
                {
                        "filesystem" :
                        {
                                "blockSize" : 0,
                                "capacity" : 0,
                                "expansionLimits" : 0,
                                "filesystemId" : "",
                                "freeCapacity" : 0,
                                "isDedupeEnabled" : false,
                                "isDedupeSupported" : false,
                                "isLogicalCapacityFreeCapacityValid" : false,
                                "isLogicalCapacityValid" : false,
                                "isNDMPRecoveryTarget" : false,
                                "isNonStrictWORM" : false,
                                "isReadCached" : false,
                                "isReadOnly" : false,
                                "isSysLocked" : false,
                                "isThinProvisioningEnabled" : false,
                                "isThinProvisioningEnabledValid" : false,
                                "isTrueSparseFileEnabled" : false,
                                "isTrueSparseFileEnabledValid" : false,
                                "isUnlimitedExpansion" : false,
                                "isWORM" : false,
                                "label" : "",
                                "logicalCapacity" : 0,
                                "logicalFreeCapacity" : 0,
                                "objectId" : "3a3a3a303a3a3a4f49445f24232140255f56",
                                "status" : "FAILING",
                                "storagePoolId" : 18446744073709551615,
                                "usedCapacity" : 0,
                                "virtualServerId" : 65535
                        },
                        "filesystemId" : "8664E653C9F413BF0000000000000000",
                        "isValid" : true,
                        "name" : "rule4",
                        "objectId" :
"343a3a3a383636344453635334339463431334246303030303030303030303030303030303a3a3a6170692
d7372632d736e617073686f742d72756c653a3a3a303a3a3a4f49445f24232140255f56",
                        "queueSize" : 25,
                        "virtualServerId" : 4
            }
    ]
}
```

Get a snapshot rule

Gets a specific snapshot rule.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/snapshot-rules/{snapshotRuleObjectID}
```

Parameters

Name	Type	Required	Values	Description
snapshotRuleObjectId	URI_PARAM	Y	string	ID of the snapshot rule object.

Return codes

Code	Data	Description
200	filesystemSnapshotRule	Retrieved a filesystem snapshot rule successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.239.120:8444/v7/storage/snapshot-rules/
343a3a3a383634333838393139931313336393443303030303030303030303030303030303a3a3a617069
2d6473742d736e617073686f742d72756c653a3a3a303a3a3a4f49445f24232140255f56
```

Response example

```
HTTP/1.1 200 OK
{
   "filesystemSnapshotRule" :
        {
               "filesystem" :
               {
                      "blockSize" : 0,
                      "capacity" : 0,
                      "expansionLimits" : 0,
                      "filesystemId" : "",
```

Chapter 15: Snapshot rules resource

```
                     "freeCapacity" : 0,
                     "isDedupeEnabled" : false,
                     "isDedupeSupported" : false,
                     "isLogicalCapacityFreeCapacityValid" : false,
                     "isLogicalCapacityValid" : false,
                     "isNDMPRecoveryTarget" : false,
                     "isNonStrictWORM" : false,
                     "isReadCached" : false,
                     "isReadOnly" : false,
                     "isSysLocked" : false,
                     "isThinProvisioningEnabled" : false,
                     "isThinProvisioningEnabledValid" : false,
                     "isTrueSparseFileEnabled" : false,
                     "isTrueSparseFileEnabledValid" : false,
                     "isUnlimitedExpansion" : false,
                     "isWORM" : false,
                     "label" : "",
                     "logicalCapacity" : 0,
                     "logicalFreeCapacity" : 0,
                     "objectId" : "3a3a3a303a3a3a4f49445f24232140255f56",
                     "status" : "FAILING",
                     "storagePoolId" : 18446744073709551615,
                     "usedCapacity" : 0,
                     "virtualServerId" : 65535
                },
                "filesystemId" : "864388919113694C0000000000000000",
                "isValid" : true,
                "name" : "api-dst-snapshot-rule",
                "objectId" :
"343a3a3a38363433388393139313133363934433030303030303030303030303030303030303a3a3a6170692
d6473742d736e617073686f742d72756c653a3a3a303a3a3a4f49445f24232140255f56",
                "queueSize" : 25,
                "virtualServerId" : 4
        }
}
```

Create a snapshot rule

Creates a snapshot rule for a filesystem ID.

HTTP request syntax (URI)

```
POST <base_URL>/v7/storage/snapshot-rules
```

Parameters

Name	Type	Required	Values	Description
filesystemId	BODY	Y	string	ID of the file system for which the snapshot rule is created.
snapshotRuleName	BODY	Y	string	Name of the snapshot rule.
queueSize	BODY	Y	number	The maximum number of snapshots that can be created by this rule. When the maximum number has been reached, the newest snapshot overwrites the oldest one. The maximum is 1024 snapshots per rule.

Return codes

Code	Data	Description
201	filesystemSnapshotRule	Retrieves a file system snapshot rule.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 201 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.239.120:8444/v7/storage/snapshot-rules -d '{"filesystemId":
"8659E4DE8E2FF4EA0000000000000000", "snapshotRuleName":"rickSSRule",
"queueSize":1}' -X POST
```

Response example

```
HTTP/1.1 201 Created
{
    "filesystemSnapshotRule" : {
        "filesystemId" : "8659E4DE8E2FF4EA0000000000000000",
        "objectId" :
"343a3a3a3836353945344445384532464634454130303030303030303030303030303030303a3a3a7269636
b535352756c65",
        "isValid" : true,
        "name" : "rickSSRule",
        "queueSize" : 1,
        "virtualServerId" : 4
    },
    "uri" : "https://172.17.239.120:8444/v7/storage/snapshot-rules/
343a3a3a3836353945344445384532464634454130303030303030303030303030303030303a3a3a7269636b
535352756c65"
}
```

Modify a snapshot rule

Modifies a snapshot rule for a filesystem ID.

HTTP request syntax (URI)

```
PATCH <base_URI>/v7/storage/snapshot-rules/{snapshotRuleObjectId}
```

Parameters

Name	Type	Required	Values	Description
snapshotRuleObjectId	URI_PARAM	Y	string	ID of the snapshot rule object.
snapshotRuleName	BODY	Y	string	Name of the snapshot rule.

Name	Type	Required	Values	Description
queueSize	BODY	Y	number	The maximum number of snapshots that can be created by this rule. When the maximum number has been reached, the newest snapshot overwrites the oldest one.

Return codes

Code	Data	Description
303	filesystemSnapshotRule	Updates a file system snapshot rule.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 303 indicates that the operation did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.239.120:8444/v7/storage/snapshot-rules/
343a3a3a3836353945344445384532464634454130303030303030303030303030303030303a3a3a726963
6b535352756c65 -d '{"snapshotRuleName":"user1-Snapshot-Rule", "queueSize":3}' -X
PATCH
```

Response example

```
HTTP/1.1 303 See Other
{
   "filesystemSnapshotRule" : {
      "filesystemId" : "8659E4DE8E2FF4EA0000000000000000",
      "objectId" :
```

Chapter 15: Snapshot rules resource

```
"343a3a3a38363539453444453845324646344541303030303030303030303030303030303030303a3a3a4875616
e2d536e617073686f742d52756c65",
      "isValid" : true,
      "name" : "user1-Snapshot-Rule",
      "queueSize" : 3,
      "virtualServerId" : 4
   },
   "uri" : "https://172.17.239.120:8444/v7/storage/snapshot-rules/
343a3a3a38363539453444453845324646344541303030303030303030303030303030303030303030303a3a3a4875616e
2d536e617073686f742d52756c65"
}
```

Delete snapshot rules

Deletes all snapshot rules that belong to a virtual server.

HTTP request syntax (URI)

```
DELETE <base_URI>/v7/storage/virtual-servers/{virtualServerObjectId}/snapshot-rules
```

Parameters

Name	Type	Required	Values	Description
virtualServerObjectId	URI_PARAM	Y	string	ID of the virtual server object to which the snapshot rules belong.

Return codes

Code	Data	Description
204	No Data.	Deleted successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Chapter 15: Snapshot rules resource

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.239.120:8444/v7/storage/virtual-servers/373a3a3a/snapshot-rules -X
DELETE
```

Response example

```
HTTP/1.1 204 No Content
```

Delete a snapshot rule

Deletes a snapshot rule, and all snapshots associated with the rule.

HTTP request syntax (URI)

```
DELETE <base_URI>/v7/storage/snapshot-rules/{snapshotRuleObjectId}
```

Parameters

Name	Type	Required	Values	Description
snapshotRuleObjectId	URI_PARAM	Y	string	ID of the snapshot rule object.

Return codes

Code	Data	Description
204	No Data	Deleted successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.239.120:8444/v7/storage/snapshot-rules/
```

```
343a3a3a38363539453444453845324646344541303030303030303030303030303030303a3
a3a4875616e2d536e617073686f742d52756c65 -X DELETE
```

Response example

```
HTTP/1.1 204 No Content
```

Get a snapshot schedule associated with a snapshot sule

Gets a snapshot schedule associated with a snapshot rule.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/snapshot-rules/{snapshotRuleObjectId}/schedules/
{snapshotScheduleObjectId}
```

Parameters

Name	Type	Required	Values	Description
snapshotRuleObjectId	URI_PARAM	Y	string	ID of the snapshot rule object.
snapshotScheduleObjectId	URI_PARAM	Y	string	ID of the snapshot schedule object.

Return codes

Code	Data	Description
200	filesystemSnapshotSchedule	A filesystemSnapshotSchedule object is returned.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.239.120:8444/v7/storage/snapshot-rules/
343a3a3a38363539453444453845324646344541303030303030303030303030303030303a3a3a726963
6b535352756c65/schedules/
3330203136202a202a202a7377616d6d79406864732e636f6d3b687472696e68406864732e636f6d3a3a3
a343a3a3a31
```

Response example

```
{
   "filesystemSnapshotSchedule" : {
      "emails" : "user1@xyz.com;user2@xyz.com",
      "handle" : 1,
      "objectId" :
"3330203136202a202a202a7377616d6d79406864732e636f6d3b687472696e68406864732e636f6d3a3a3
a343a3a3a31",
      "isValid" : true,
      "time" : "30 16 * * *",
      "virtualServerId" : 7
   }
}
```

Add a snapshot schedule to a snapshot rule

Adds a snapshot schedule to a snapshot rule.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/snapshot-rules/{snapshotRuleObjectId}/schedules
```

Parameters

Name	Type	Required	Values	Description
snapshotRuleObjectId	URI_PARAM	Y	string	ID of the snapshot rule object.
emails	BODY	Y	string	List of emails separated by ';'.
time	BODY	Y	string	UNIX cron job time specification format.

Chapter 15: Snapshot rules resource

Return codes

Code	Data	Description
201	filesystemSnapshotSchedule	filesystemSnapshotSchedule object is returned.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 201 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.239.120:8444/v7/storage/snapshot-rules/
343a3a3a3836353945344445384532464634454130303030303030303030303030303a3a3a726963
6b535352756c65/schedules -d '{"time":"30 16 * * *", "emails":"user1@xyz.com;
user2@xyz.com"}' -X POST
```

Response example

```
HTTP/1.1 201 Created
{
   "filesystemSnapshotSchedule" : {
       "emails" : "user1@xyz.com;user2@xyz.com",
       "handle" : 1,
       "objectId" :
"3330203136202a202a202a7377616d6d79406864732e636f6d3b687472696e68406864732e636f6d3a3a3a
a343a3a3a31",
       "isValid" : true,
       "time" : "30 16 * * *",
       "virtualServerId" : 4
   },
   "uri" : "https://172.17.239.120:8444/v7/storage/snapshot-rules/
343a3a3a3836353945344445384532464634454130303030303030303030303030303a3a3a7269636b
535352756c65/schedules/
3330203136202a202a202a7377616d6d79406864732e636f6d3b687472696e68406864732e636f6d3a3a3a
343a3a3a31"
}
```

Delete a snapshot schedule associated with a snapshot rule

Deletes a snapshot schedule associated with a snapshot rule.

HTTP request syntax (URI)

```
DELETE <base_URI>/v7/storage/snapshot-rules/{snapshotRuleObjectId}/schedules/
{snapshotScheduleObjectId}
```

Parameters

Name	Type	Required	Values	Description
snapshotRuleObjectId	URI_PAR AM	Y	string	ID of the snapshot rule object.
snapshotscheduleObjectId	URI_PAR AM	Y	string	ID of the snapshot schedule object.

Return codes

Code	Data	Description
204	No Data	The object was successfully deleted.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.239.120:8444/v7/storage/snapshot-rules/
343a3a3a38363539453444453845324646344541303030303030303030303030303a3a3a726963
6b535352756c65/schedules/
3330203136202a202a202a7377616d6d79406864732e636f6d3b687472696e68406864732e636f6d3a3a
3a343a3a3a31 -X DELETE
```

Response example

```
HTTP/1.1 204 No Content
```

Chapter 16: Storage pool resource

The storage pool resource refers to the virtual containers that provide storage for file systems. File systems consume storage pool space when they are created or as they expand.

Storage pool object model

The object model describing this resource contains the following objects.

storagePool

Attribute	JSON Type	Data Type	Description
objectId	string	string	Storage pool object unique identifier.
storagePoolId	number	uint64	Storage pool ID.
label	string	string	Storage pool label.
totalCapacity	number	uint64	Total capacity in bytes.
usedCapacity	number	uint64	Used capacity in bytes.
freeCapacity	number	uint64	Free capacity in bytes.
chunkSize	number	uint64	Chunk size in bytes.
isHealthy	boolean	boolean	True if storage pool is in healthy condition.
isTiered	boolean	boolean	True if storage pool is tiered.
isFilesystemExpansionAllowed	boolean	boolean	True if allowed.
isAssignedToLocalCluster	boolean	boolean	True if assigned.
tierInformation	object	array	Array of tierInformation objects related to the storage pool, if the storage pool is made up from multiple tiers. Empty if storage pool is only a single tier

tierinformation

Attribute	JSON Type	Data Type	Description
tierNumber	number	integer	Tier number.
tierCapacity	number	uint64	Capacity of the tier in bytes.
tierFreeSpace	number	uint64	Free capacity of the tier in bytes.

storagePoolHdpPoolInfo

Attribute	JSON Type	Data Type	Description
arraySerial	string	string	Serial number of the Hitachi Dynamic Provisioning (HDP) pool storage system.
freeSpace	number	integer	Free space (in bytes) of the HDP pool.
poolIds	array	array	List of other storage pools that are included in this HDP pool.
poolNumber	number	integer	HDP pool number.

Get storage pools

Retrieves storage pools on the storage system.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/storage-pools
```

Return codes

Code	Data	Description
200	storagePool	Array of storage pools retrieved successfully.
500	Error message	Error associated with the storage system.

Code	Data	Description
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/storage-pools
```

Response example

```
HTTP/1.1 200 OK
{
   "storagePools" :
    [
      {
         "chunkSize" : 19327352832,
         "freeCapacity" : 21458059264,
         "isAssignedToLocalCluster" : true,
         "isFilesystemExpansionAllowed" : true,
         "isHealthy" : true,
         "isTiered" : false,
         "label" : "xyz-uneven-lun-SP-test",
         "objectId" : "353036333333336373939313433133373639393a3a3a",
         "storagePoolId" : 506333679914137699,
         "totalCapacity" : 21458059264,
         "usedCapacity" : 0
      },
      {
         "chunkSize" : 1073741824,
         "freeCapacity" : 8455716864,
         "isAssignedToLocalCluster" : true,
         "isFilesystemExpansionAllowed" : true,
         "isHealthy" : true,
         "isTiered" : false,
         "label" : "test-sanjeev",
         "objectId" : "353035373133303539383738363939323232353a3a3a",
         "storagePoolId" : 505713059878699225,
         "totalCapacity" : 85882568704,
         "usedCapacity" : 77426851840
      }
    ]
}
```

Chapter 16: Storage pool resource

Get a storage pool

Retrieves a storage pool on the storage system.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/storage-pools/{id}
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string/number	Specifies either storage pool object ID or HNAS storage pool ID.

Return codes

Code	Data	Description
200	storagePool	Storage pool information retrieved successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example: Using the storage pool object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/storage-pools/
35303537393330313031303633323303038383a3a3a
```

Response example: Using the storage pool object ID

```
HTTP/1.1 200 OK
{
    "storagePool" : {
        "chunkSize" : 19327352832,
        "freeCapacity" : 42932895744,
        "isAssignedToLocalCluster" : true,
```

```
        "isFilesystemExpansionAllowed" : true,
        "isHealthy" : true,
        "isTiered" : false,
        "label" : "sp-test",
        "objectId" : "3530353739333303130313036333232303038383a3a3a",
        "storagePoolId" : 505793010106320088,
        "totalCapacity" : 42932895744,
        "usedCapacity" : 0
    }
}
```

Request example: Using the HNAS storage pool ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/storage-pools/505793010106320088
```

Response example: Using the HNAS storage pool ID

```
HTTP/1.1 200 OK
{
    "storagePool" : {
        "chunkSize" : 19327352832,
        "freeCapacity" : 42932895744,
        "isAssignedToLocalCluster" : true,
        "isFilesystemExpansionAllowed" : true,
        "isHealthy" : true,
        "isTiered" : false,
        "label" : "sp-test",
        "objectId" : "3530353739333303130313036333232303038383a3a3a",
        "storagePoolId" : 505793010106320088,
        "totalCapacity" : 42932895744,
        "usedCapacity" : 0
    }
}
```

Create a storage pool

Creates a non-tiered storage pool using a set of system drives. A label is assigned to the
storage pool during the creation process. The operation returns a URI that can be accessed
to obtain detailed information about the storage pool immediately after the URI is returned.

HTTP request syntax (URI)

```
POST <base_URI>/v7/stroage/storage-pools
```

Parameters

Name	Type	Required	Values	Description
label	BODY	Y	string	Label for the storage pool to be created.
chunkSize	BODY	Y	number	Chunk size in bytes.
systemDrives	BODY	Y	array	System drives to be used to create the storage pool. To create a storage pool, the number of system drives must be at least 4 but cannot exceed 32.

Return codes

Code	Data	Description
201	storagePool	Storage pool object successfully created.
400	No Data	Missing or invalid request contents.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 201 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/storage-pools -d '{"label":"sp-test",
"chunkSize":19327352832, "systemDrives":[10, 11, 12, 13]}' -X POST
```

Response example

```
HTTP/1.1 201 Created
{
   "storagePool" : {
      "chunkSize" : 19327352832,
      "freeCapacity" : 42932895744,
      "isAssignedToLocalCluster" : true,
      "isFilesystemExpansionAllowed" : true,
      "isHealthy" : true,
      "isTiered" : false,
      "label" : "sp-test",
      "objectId" : "353035373933330313031303633323303038383a3a3a",
```

Chapter 16: Storage pool resource

```
      "storagePoolId" : 505793010106320088,
      "totalCapacity" : 42932895744,
      "usedCapacity" : 0
   },
   "uri" : "https://172.17.11.11:8444/v7/storage/storage-pools/
3530353739333031303130363332303038383a3a3a"
}
```

Create a tiered storage pool

Creates a tiered storage pool using a set of system drives grouped by tiers. A label is assigned to the storage pool during the creation process. The operation returns a URI that can be accessed to obtain detailed information about the storage pool immediately after the URI is returned.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/storage-pools
```

Parameters

Name	Type	Required	Values	Description
label	BODY	Y	string	Label for the storage pool to be created.
chunkSize	BODY	Y	number	Chunk size in bytes.
systemDrives	BODY	Y	object	Object that contains the array of tier 0 and 1 system drives.
tier0SystemDrives	BODY	Y	array	Tier 0 system drives to be used to create the storage pool. The number of system drives must be at least 4 but cannot exceed 32.
tier1SystemDrives	BODY	Y	array	Tier 1 system drives to be used to create the storage pool. The number of system drives must be at least 4 but cannot exceed 32.

Chapter 16: Storage pool resource

Return codes

Code	Data	Description
201	storagePool	Storage pool object successfully created.
400	No Data	Missing or invalid request contents.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 201 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/storage-pools -d '{"label": "sp-test",
"chunkSize": 19327352832, "systemDrives": {"tier0SystemDrives": [10, 11,12, 13],
"tier1SystemDrives": [31, 32, 33, 34]}}' -X POST
```

Response example

```
HTTP/1.1 201 Created
{
    "storagePool" : {
        "chunkSize" : 19327352832,
        "freeCapacity" : 42932895744,
        "isAssignedToLocalCluster" : true,
        "isFilesystemExpansionAllowed" : true,
        "isHealthy" : true,
        "isTiered" : true,
        "label" : "sp-test",
        "objectId" : "35303537393330313031303633332303038383a3a3a",
        "storagePoolId" : 505793010106320088,
        "totalCapacity" : 42932895744,
        "usedCapacity" : 0
    },
    "uri" : "https://172.17.11.11:8444/v7/storage/storage-pools/
35303537393330313031303633332303038383a3a3a"
}
```

Get system drives associated with a storage pool

Retrieves system drives used to create a storage pool on the storage system.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/storage-pools/{id}/system-drives
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string/number	Specifies either the storage pool object ID or HNAS storage pool ID.

Return codes

Code	Data	Description
200	systemDrive	Array of system drives was retrieved successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example: Using the storage pool object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/storage-pools/
393637373036323034343433363639353430343a3a3a/system-drives
```

Response example: Using the storage pool object ID

```
HTTP/1.1 200 OK
{
    "systemDrives" :
        [
            {
                "capacity" : 429496729600,
                "comment" : "",
                "controllerPort" : "0000",
                "externalLUN" : "0",
                "internalLUN" : "00E2",
                "isAccessAllowed" : true,
```

```
                            "isAssignedToStoragePool" : true,
                            "isMirrorPrimary" : true,
                            "isMirrored" : false,
                            "label" : "00E2",
                            "mirrorDriveUniqueId" : "",
                            "model" : "OPEN-V",
                            "objectId" : "343a3a3a3a3a3a303a3a3a4f49445f24232140255f56",
                            "queueDepth" : -1,
                            "rackName" : "10076",
                            "status" : "OK",
                            "subModel" : "R800",
                            "systemDriveId" : 4,
                            "tier" : "NO_TIER",
                            "uniqueId" : "31096614-1287399200483992000-30",
                            "vendor" : "HITACHI"
                    }
            ]
}
```

Request example: Using the HNAS storage pool ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/storage-pools/505793010106320088/system-drives
```

Response example: Using the HNAS storage pool ID

```
HTTP/1.1 200 OK
{
    "systemDrives" :
        [
                {
                        "capacity" : 429496729600,
                        "comment" : "",
                        "controllerPort" : "0000",
                        "externalLUN" : "0",
                        "internalLUN" : "00E2",
                        "isAccessAllowed" : true,
                        "isAssignedToStoragePool" : true,
                        "isMirrorPrimary" : true,
                        "isMirrored" : false,
                        "label" : "00E2",
                        "mirrorDriveUniqueId" : "",
                        "model" : "OPEN-V",
                        "objectId" : "343a3a3a3a3a3a303a3a3a4f49445f24232140255f56",
                        "queueDepth" : -1,
                        "rackName" : "10076",
                        "status" : "OK",
                        "subModel" : "R800",
                        "systemDriveId" : 4,
                        "tier" : "NO_TIER",
```

Chapter 16: Storage pool resource

```
                            "uniqueId" : "31096614-1287399200483992000-30",
                            "vendor" : "HITACHI"
                    }
            ]
}
```

Get file systems associated with a storage pool

Retrieves file systems associated with the storage pool.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/storage-pools/{id}/filesystems
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string/number	Specifies either the storage pool object ID or HNAS storage pool ID.

Return codes

Code	Data	Description
200	filesystem	Array of file systems associated with storage pool was retrieved successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example: Using the storage pool object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/storage-pools/
35303537313330353938373836393932353a3a3a/filesystems
```

Response example: Using the storage pool object ID

```
HTTP/1.1 200 OK
 {
   "filesystems" :
       [
               {
                       "blockSize" : 32768,
                       "capacity" : 19327352832,
                       "expansionLimits" : 53687091200,
                       "filesystemId" : "8642B2069D9F4F8A0000000000000000",
                       "freeCapacity" : 16714563584,
                       "isDedupeEnabled" : false,
                       "isDedupeSupported" : false,
                       "isLogicalCapacityFreeCapacityValid" : true,
                       "isLogicalCapacityValid" : true,
                       "isNDMPRecoveryTarget" : false,
                       "isNonStrictWORM" : false,
                       "isReadCached" : false,
                       "isReadOnly" : false,
                       "isSysLocked" : false,
                       "isThinProvisioningEnabled" : false,
                       "isThinProvisioningEnabledValid" : true,
                       "isTrueSparseFileEnabled" : true,
                       "isTrueSparseFileEnabledValid" : true,
                       "isUnlimitedExpansion" : false,
                       "isWORM" : false,
                       "label" : "DimitriDevFS",
                       "logicalCapacity" : 19327352832,
                       "logicalFreeCapacity" : 16714563584,
                       "objectId" :
"383634324232303639443946344638413030303030303030303030303030303030303a3a3a303a3a3a4f49445
f24232140255f56",
                       "status" : "MOUNTED",
                       "storagePoolId" : 9676880324761524863,
                       "usedCapacity" : 2612789248,
                       "virtualServerId" : 4
               },
               {
                       "blockSize" : 32768,
                       "capacity" : 19327352832,
                       "expansionLimits" : 19327352832,
                       "filesystemId" : "3B6BB41DC647FC060000000000000000",
                       "freeCapacity" : 1158905856,
                       "isDedupeEnabled" : false,
                       "isDedupeSupported" : false,
                       "isLogicalCapacityFreeCapacityValid" : true,
                       "isLogicalCapacityValid" : true,
                       "isNDMPRecoveryTarget" : false,
                       "isNonStrictWORM" : false,
                       "isReadCached" : false,
```

Chapter 16: Storage pool resource

```
                        "isReadOnly" : false,
                        "isSysLocked" : false,
                        "isThinProvisioningEnabled" : false,
                        "isThinProvisioningEnabledValid" : true,
                        "isTrueSparseFileEnabled" : true,
                        "isTrueSparseFileEnabledValid" : true,
                        "isUnlimitedExpansion" : false,
                        "isWORM" : false,
                        "label" : "ErinObjSrc",
                        "logicalCapacity" : 19327352832,
                        "logicalFreeCapacity" : 1158905856,
                        "objectId" :
"3342364242343144433634374643303630303030303030303030303030303030303a3a3a303a3a3a4f49445
f24232140255f56",
                        "status" : "MOUNTED",
                        "storagePoolId" : 9676880324761524863,
                        "usedCapacity" : 18168446976,
                        "virtualServerId" : 4
                }
        ]
}
```

Request example: Using the HNAS storage pool ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/storage-pools/505793010106320088/filesystems
```

Response example: Using the HNAS storage pool ID

```
HTTP/1.1 200 OK
 {
   "filesystems" :
        [
                {
                        "blockSize" : 32768,
                        "capacity" : 19327352832,
                        "expansionLimits" : 53687091200,
                        "filesystemId" : "8642B2069D9F4F8A0000000000000000",
                        "freeCapacity" : 16714563584,
                        "isDedupeEnabled" : false,
                        "isDedupeSupported" : false,
                        "isLogicalCapacityFreeCapacityValid" : true,
                        "isLogicalCapacityValid" : true,
                        "isNDMPRecoveryTarget" : false,
                        "isNonStrictWORM" : false,
                        "isReadCached" : false,
                        "isReadOnly" : false,
                        "isSysLocked" : false,
                        "isThinProvisioningEnabled" : false,
                        "isThinProvisioningEnabledValid" : true,
```

Chapter 16: Storage pool resource

```
                        "isTrueSparseFileEnabled" : true,
                        "isTrueSparseFileEnabledValid" : true,
                        "isUnlimitedExpansion" : false,
                        "isWORM" : false,
                        "label" : "DimitriDevFS",
                        "logicalCapacity" : 19327352832,
                        "logicalFreeCapacity" : 16714563584,
                        "objectId" :
"38363432423230363944394634463841303030303030303030303030303030303a3a3a303a3a3a4f49445
f24232140255f56",
                        "status" : "MOUNTED",
                        "storagePoolId" : 9676880324761524863,
                        "usedCapacity" : 2612789248,
                        "virtualServerId" : 4
                },
                {
                        "blockSize" : 32768,
                        "capacity" : 19327352832,
                        "expansionLimits" : 19327352832,
                        "filesystemId" : "3B6BB41DC647FC060000000000000000",
                        "freeCapacity" : 1158905856,
                        "isDedupeEnabled" : false,
                        "isDedupeSupported" : false,
                        "isLogicalCapacityFreeCapacityValid" : true,
                        "isLogicalCapacityValid" : true,
                        "isNDMPRecoveryTarget" : false,
                        "isNonStrictWORM" : false,
                        "isReadCached" : false,
                        "isReadOnly" : false,
                        "isSysLocked" : false,
                        "isThinProvisioningEnabled" : false,
                        "isThinProvisioningEnabledValid" : true,
                        "isTrueSparseFileEnabled" : true,
                        "isTrueSparseFileEnabledValid" : true,
                        "isUnlimitedExpansion" : false,
                        "isWORM" : false,
                        "label" : "ErinObjSrc",
                        "logicalCapacity" : 19327352832,
                        "logicalFreeCapacity" : 1158905856,
                        "objectId" :
"33423642423431444336343746433036303030303030303030303030303030303a3a3a303a3a3a4f49445
f24232140255f56",
                        "status" : "MOUNTED",
                        "storagePoolId" : 9676880324761524863,
                        "usedCapacity" : 18168446976,
                        "virtualServerId" : 4
                }
        ]
}
```

Chapter 16: Storage pool resource

Rename a storage pool

Renames a storage pool. To rename a storage pool, change its associated label.

HTTP request syntax (URI)

```
PATCH <base_URI>/v7/storage/storage-pools/{id}
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string/number	Specifies either the storage pool object ID or HNAS storage pool ID to rename.
label	BODY	Y	string	Storage pool label.

Return codes

Code	Data	Description
204	No data	Storage pool renamed successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example: Using the storage pool object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/storage-pools/
35303537393330313031303363332303038383a3a3a -d '{"label":"sp-test1"}' -X PATCH
```

Response example: Using the storage pool object ID

```
HTTP/1.1 204 No Content
```

Request example: Using the HNAS storage pool ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/storage-pools/505793010106320088 -d
'{"label":"sp-test1"}' -X PATCH
```

Response example: Using the HNAS storage pool ID

```
HTTP/1.1 204 No Content
```

Expand a storage pool

Expands a storage pool. The operation accepts a list of system drives to be used during storage pool expansion.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/storage-pools/{id}/expand
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string/number	Specifies either the storage pool object ID or HNAS storage pool ID.
systemDrives	BODY	Y	array	System drives to be used to expand the storage pool. For rules on expansion, see examples in the note below.
tier	BODY	N	string	Only required when expanding a tiered storage pool. Possible values are: ▪ TIER0 ▪ TIER1

Chapter 16: Storage pool resource

> 📄 **Note:** Example of systemDrives values:
>
> - Example 1 System drive number expansion requirement: If the storage pool was initially created with 4 system drives, you need at least 4 or more system drives to expand.
>
> - Example 2 Capacity expansion requirement: List capacity refers to the amount of storage of the drive with the lowest capacity value in the grouping. When you expand the storage pool of a series of system drives, you must increment each by the list value. As an example, consider the capacity of each drive in a grouping of 4 system drives is as follows: Then to expand the storage pool, you must expand each of the system drives by at least 1 GB, the list value.
>   - 1 GB for drive 1
>   - 1 GB for drive 2
>   - 1 GB for drive 3
>   - 5 GB for drive 4

Return codes

Code	Data	Description
204	No Data	Storage pool successfully expanded.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example: To expand a non-tiered storage pool by specifying the storage pool object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/storage-pools/
353035373933303130313036333230303038383a3a3a/expand -d '{"systemDrives":[15,16,17,
18]}' -X POST
```

Response example: To expand a non-tiered storage pool by specifying the storage pool object ID

```
HTTP/1.1 204 No Content
```

Request example: To expand a tiered storage pool by specifying the storage pool object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/storage-pools/
3530353233335393636343933343938333343313a3a3a/expand -d '{"tier": "TIER1",
"systemDrives":[114,115,116,117,118,119]}' -X POST
```

Response example: To expand a tiered storage pool by specifying the storage pool object ID

```
HTTP/1.1 204 No Content
```

Request example: To expand a non-tiered storage pool by specifying the HNAS storage pool ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/storage-pools/505793010106320088/expand -d
'{"systemDrives":[15,16,17,18]}' -X POST
```

Response example: To expand a non-tiered storage pool by specifying the HNAS storage pool ID

```
HTTP/1.1 204 No Content
```

Delete a storage pool

Deletes a storage pool identified by a unique identifier. A storage pool can be deleted only after the file systems that were created using the storage pool have been deleted.

HTTP request syntax (URI)

```
DELETE <base_URI>/v7/storage/storage-pools/{id}
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string/number	Specifies either the storage pool object ID or HNAS storage pool ID.

Return codes

Code	Data	Description
204	No Data	Storage pool successfully deleted.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example: Using the storage pool object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/storage-pools/
35303537393330313031303633332303038383a3a3a -X DELETE
```

Response example: Using the storage pool object ID

```
HTTP/1.1 204 No Content
```

Request example: Using the HNAS storage pool ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/storage-pools/505793010106320088 -X DELETE
```

Response example: Using the HNAS storage pool ID

```
HTTP/1.1 204 No Content
```

Get HDP storage pool info

Retrieves HDP storage pool information on the storage system. A unique identifier identifies the storage pool.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/storage-pools/{id}/hdp
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string/number	Specifies either the storage pool object ID or HNAS storage pool ID.

Return codes

Code	Data	Description
200	storagePoolHdpPoolInfo	HDP pool information retrieved successfully
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example: Using storage pool object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/storage-pools/
3530353739333031303130363332303038383a3a3a/hdp
```

Response example: Using storage pool object ID

```
HTTP/1.1 200 OK
{
  "storagePoolHdpPoolInfo": [
    {
      "arraySerial": "91250531",
      "freeSpace": 9516573786112,
      "poolIds": [
        {
          "poolId": 530991019344831940
        },
        {
          "poolId": 530991019272706240
        },
        {
```

```
          "poolId": 530991019443607550
        },
        {
          "poolId": 530991019380918000
        }
      ],
      "poolNumber": 0
    }
  ]
}
```

Chapter 17: System drive resource

The system drive resource represents a logical drive on the file storage system on block storage physical logical units. The system drive resource enables you to retrieve detailed information about system drives on the storage system.

System drive object model

The object model describing this resource contains the following objects.

systemDrive

Attribute	JSON Type	Data Type	Description
objectId	string	string	System drive unique identifier.
systemDriveId	number	uint	System drive ID.
status	string	string	Status of the system drive. Possible values are: ▪ FORMATTING ▪ SECONDARY ▪ WRITE_PROTECTED ▪ DISCONNECTED ▪ FAILED ▪ FIRST_INVALID ▪ INITIALIZING ▪ NOT_PRESENT ▪ OFFLINE ▪ OK
uniqueId	string	string	Unique identifier of the storage system drive.
label	string	string	System drive label.

Attribute	JSON Type	Data Type	Description
comment	string	string	Comment associated with the system drive.
capacity	number	int	Capacity of the drive in bytes.
rackName	string	string	The serial number of the storage system to which the system drive belongs
queueDepth	number	int	The queue depth of a system drive. The default value is -1. The value range for setting is 16 to 512.
mirrorDriveUniqueId	string	string	Unique ID of the mirror drive associated with the system drive. If there is no mirror drive, the string is empty.
isAssignedToStoragePool	boolean	boolean	True if the drive is assigned to a storage pool.
isMirrored	boolean	boolean	True if the drive has a mirror drive associated with it.
isAccessAllowed	boolean	boolean	True if access to the drive is enabled.
isMirrorPrimary	boolean	boolean	True if the system drive is mirrored and it is the primary mirror.
vendor	string	string	Manufacturer of the storage system.
model	string	string	Model of the storage system.
tier	string	string	The tier status that the system drive belongs to. Possible values are: ▪ TIER0 ▪ TIER1 ▪ NO_TIER ▪ BAD_TIER ▪ UNKNOWN_TIER
controllerPort	string	string	Controller port.

Chapter 17: System drive resource

Attribute	JSON Type	Data Type	Description
externalLUN	string	string	External LUN.
internalLUN	string	string	Internal LUN.
subModel	string	string	Sub model.

Get system drives

Retrieves system drives on the storage system.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/system-drives
```

Return codes

Code	Data	Description
200	systemDrives	Array of system drives retrieved successfully.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/system-drives
```

Response example

```
HTTP/1.1 200 OK
{
  "systemDrives" : [
      {
          "capacity" : 214748364800,
          "comment" : "",
          "controllerPort" : "0000",
          "externalLUN" : "10",
```

Chapter 17: System drive resource

```
            "internalLUN" : "0064",
            "isAccessAllowed" : true,
            "isAssignedToStoragePool" : false,
            "isMirrorPrimary" : true,
            "isMirrored" : false,
            "label" : "0064",
            "mirrorDriveUniqueId" : "31096614-12822110-17001110-17000100",
            "model" : "OPEN-V",
            "objectId" : "303a3a3a4f49445f24232140255f56",
            "queueDepth" : -1,
            "rackName" : "93935",
            "status" : "OK",
            "subModel" : "R700",
            "systemDriveId" : 0,
            "tier" : "NO_TIER",
            "uniqueId" : "31096614-12822110-17001110-17000100",
            "vendor" : "HITACHI"
        },
        {
            "capacity" : 214749020160,
            "comment" : "",
            "controllerPort" : "0000",
            "externalLUN" : "0",
            "internalLUN" : "00A5",
            "isAccessAllowed" : true,
            "isAssignedToStoragePool" : true,
            "isMirrorPrimary" : true,
            "isMirrored" : false,
            "label" : "00A5",
            "mirrorDriveUniqueId" : "31096614-128193966080323966000-91",
            "model" : "OPEN-V",
            "objectId" : "313a3a3a4f49445f24232140255f56",
            "queueDepth" : -1,
            "rackName" : "210050",
            "status" : "OK",
            "subModel" : "HM70",
            "systemDriveId" : 1,
            "tier" : "NO_TIER",
            "uniqueId" : "31096614-128193966080323966000-91",
            "vendor" : "HITACHI"
        }
    ]
}
```

Get a system drive

Retrieves a system drive on the storage system.

Chapter 17: System drive resource

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/system-drives/{id}
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string/number	Specifies either the system drive object ID or HNAS storage system drive ID.

Return codes

Code	Data	Description
200	systemDrive	System drive successfully retrieved.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example: Using the system drive object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/system-drives/
3132313a3a3a4f49445f24232140255f56
```

Response example: Using the system drive object ID

```
HTTP/1.1 200 OK
 {
   "systemDrive" : {
      "capacity" : 21474836480,
      "comment" : "",
      "controllerPort" : "0000",
      "externalLUN" : "27",
      "internalLUN" : "0234",
      "isAccessAllowed" : true,
      "isAssignedToStoragePool" : false,
      "isMirrorPrimary" : true,
```

Chapter 17: System drive resource

```
      "isMirrored" : false,
      "label" : "0234",
      "mirrorDriveUniqueId" : "31096614-128739910048399100252",
      "model" : "OPEN-V",
      "objectId" : "3132313a3a3a4f49445f24232140255f56",
      "queueDepth" : 39,
      "rackName" : "10075",
      "status" : "OK",
      "subModel" : "R800",
      "systemDriveId" : 21,
      "tier" : "NO_TIER",
      "uniqueId" : "31096614-128739910048399100252",
      "vendor" : "HITACHI"
   }
}
```

Request example: Using the HNAS storage system drive ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/system-drives/21
```

Response example: Using the HNAS storage system drive ID

```
HTTP/1.1 200 OK
{
   "systemDrive" : {
      "capacity" : 21474836480,
      "comment" : "",
      "controllerPort" : "0000",
      "externalLUN" : "27",
      "internalLUN" : "0234",
      "isAccessAllowed" : true,
      "isAssignedToStoragePool" : false,
      "isMirrorPrimary" : true,
      "isMirrored" : false,
      "label" : "0234",
      "mirrorDriveUniqueId" : "31096614-128739910048399100252",
      "model" : "OPEN-V",
      "objectId" : "3132313a3a3a4f49445f24232140255f56",
      "queueDepth" : 39,
      "rackName" : "10075",
      "status" : "OK",
      "subModel" : "R800",
      "systemDriveId" : 21,
      "tier" : "NO_TIER",
      "uniqueId" : "31096614-128739910048399100252",
      "vendor" : "HITACHI"
   }
}
```

Chapter 17: System drive resource

Set system drive

Sets the queue depth for a system drive on the storage system. This operation can also be used to enable or disable access to a system drive.

HTTP request syntax (URI)

```
PATCH <base_URI>/v7/storage/system-drives/{id}
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string/number	Specifies either the system drive object ID or HNAS storage system drive ID.
queueDepth	BODY	N	number	Queue depth. Value range is 16 to 512. A value of -1 resets the queue depth to the default for the particular drive type.
enableAccess	BODY	N	boolean	True to allow the server access to the system drive or false to deny access.

Return codes

Code	Data	Description
204	No Data	Queue depth of a system drive successfully set.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Chapter 17: System drive resource

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example: Using the system drive object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/system-drives/
32313a3a3a4f49445f24232140255f56 -d '{"queueDepth": 49, "enableAccess": true}' -X
PATCH
```

Response example: Using the system drive object ID

```
HTTP/1.1 204 No Content
```

Request example: Using the HNAS storage system drive ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/system-drives/21 -d '{"queueDepth": 49,
"enableAccess": true}' -X PATCH
```

Response example: Using the HNAS storage system drive ID

```
HTTP/1.1 204 No Content
```

Refresh SCSI system drives

Refreshes all SCSI system drives of all nodes in a storage system. System drives should be refreshed when changes have been made, or new storage attached to a system.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/system-drives/scsi-refresh
```

Return codes

Code	Data	Description
204	No Data	iSCSI system drives were refreshed successfully.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/system-drives/scsi-refresh -X POST
```

Response example

```
HTTP/1.1 204 No Content
```

Chapter 18: Virtual server resource

The virtual server resource, also called EVS, consists of virtualized file services or administrative services. Each virtual server is assigned unique network settings and storage resources, providing the flexibility to logically partition access to shared storage resources. To ensure maximum data availability, you can configure virtual servers to automatically migrate between nodes in a cluster when failures occur. You can retrieve virtual servers in the system.

Virtual server object model

The object model describing this resource contains the following objects.

virtualServer

Attribute	JSON Type	Data Type	Description
objectId	string	string	Virtual server object unique identifier. Note: this ID is not HNAS storage virtual server ID.
virtualServerId	number	ushort	HNAS virtual server ID.
status	string	string	Status of the virtual server. Possible values are: ▪ ONLINE ▪ NOT_CONFIGURED ▪ DISABLED ▪ OFFLINE
UUID	string	string	UUID of the virtual server.
ipAddresses	array	array	IP addresses for the virtual server.
name	string	string	Name of the virtual server.

Attribute	JSON Type	Data Type	Description
type	string	string	Type of the virtual server. Possible values can be: ▪ File Services ▪ Administrative Services
isEnabled	boolean	boolean	True if the virtual server is enabled.

scanServer (RPC mode)

Attribute	JSON Type	Data Type	Description
name	string	string	Name of the virus scan engine server
inUse	boolean	boolean	True if the virus scan server is in use
ipAddress	string	string	IP of the virus scan server
domain	string	string	Domain name of the virus scan service
scanStatus	number	integer	Status of the virus scan server

scanServer (ICAP mode)

Attribute	JSON Type	Data Type	Description
hostName	string	string	Host name of the virus scan engine server
enabled	boolean	boolean	True if the virus scan server is enabled
ipAddress	string	string	IP of the virus scan server
port	number	integer	Port of the virus scan server

Chapter 18: Virtual server resource

Attribute	JSON Type	Data Type	Description
serviceName	string	string	Service name of the virus scan service
scanStatus	number	integer	Status of the virus scan server

virusScan

Attribute	JSON Type	Data Type	Description
fileExtensions	array	array	A list of file extensions the virus scan run against
areAllFileTypesScanned	boolean	boolean	Virus scan run for all file types or not
isSuccessful	boolean	boolean	Virus scan successful or not
isVirusScanEnabled	boolean	boolean	Virus scan enabled or not
mode	string	string	Virus scan mode: RPC or ICAP
scanServers	array	array	A list of objects of scanServer that describes the virus scan server configuration of either RPC mode or ICAP mode.

Get virtual servers

Retrieves virtual servers in the system.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/virtual-servers
```

Return codes

Code	Data	Description
200	virtualServers	Array of virtual servers successfully retrieved.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/virtual-servers
```

Response example

```
HTTP/1.1 200 OK
    {
  "virtualServers" : [
      {
        "UUID" : "34c45abc-680f-11d1-901c-040400070206",
        "ipAddresses" : [ "192.168.48.2" ],
        "isEnabled" : true,
        "name" : "G600-440726-a",
        "objectId" : "303a3a3a4f49445f24232140255f56",
        "status" : "ONLINE",
        "type" : "Administrative services",
        "virtualServerId" : 0
      },
       {
        "UUID" : "a0751a20-7bf6-11d2-9511-040400070206",
        "ipAddresses" : [ "172.17.59.229" ],
        "isEnabled" : true,
        "name" : "xyz11-59-229",
        "objectId" : "31313a3a3a4f49445f24232140255f56",
        "status" : "ONLINE",
        "type" : "File services",
        "virtualServerId" : 11
      }
    ]
  }
```

Chapter 18: Virtual server resource

Get a virtual server

Retrieves a virtual server in the system. A unique value identifies the virtual server.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/virtual-servers/{id}
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string/number	Specifies either the virtual server object ID or the HNAS storage virtual server ID.

Return codes

Code	Data	Description
200	virtualServer	Virtual server successfully created.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example: Using the virtual server object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/virtual-servers/383a3a3a4f49445f24232140255f56
```

Response example: Using the virtual server object ID

```
HTTP/1.1 200 OK
{
   "virtualServer" : {
      "UUID" : "6efbf72c-b287-11d1-901e-040400070206",
      "ipAddresses" : [ "111.111.111.111" ],
      "isEnabled" : true,
      "name" : "testing",
```

Chapter 18: Virtual server resource

```
        "objectId" : "383a3a3a4f49445f24232140255f56",
        "status" : "ONLINE",
        "type" : "File services",
        "virtualServerId" : 8
    }
}
```

Request example: Using the HNAS storage virtual server ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/virtual-servers/8
```

Response example: Using the HNAS storage virtual server ID

```
HTTP/1.1 200 OK
{
    "virtualServer" : {
        "UUID" : "6efbf72c-b287-11d1-901e-040400070206",
        "ipAddresses" : [ "111.111.111.111" ],
        "isEnabled" : true,
        "name" : "testing1",
        "objectId" : "383a3a3a4f49445f24232140255f56",
        "status" : "ONLINE",
        "type" : "File services",
        "virtualServerId" : 8
    }
}
```

Create a virtual server

Creates a virtual server on the storage system. The virtual server is assigned an IP address when you create the server. It uses the Ethernet link aggregation port available on the physical cluster node. The operation returns a URI you can access to retrieve a virtual server immediately after the operation returns.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/virtual-servers
```

Chapter 18: Virtual server resource

Parameters

Name	Type	Required	Values	Description
name	BODY	Y	string	Name (only alphanumeric and "-") for the virtual server to be created.
clusterNodeId	BODY	Y	number	ID of the cluster node in which to create the virtual server.
ipAddress	BODY	Y	string	Either IPv4 or IPv6 address of the virtual server.
netmask	BODY	Y	string	Netmask must be of the form x.x.x.x for an IPv4 address or specify a prefix length for an IPv6 address.
ethernetLinkAggregation	BODY	Y	string	Name of the Ethernet link aggregation for the virtual server.

Return codes

Code	Data	Description
201	virtualServer	Virtual server object successfully created.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 201 indicates that the API did not complete successfully.

Chapter 18: Virtual server resource

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/virtual-servers -d '{"name":"TestEVS",
"clusterNodeId":1, "ipAddress":"172.17.37.63", "netmask":"255.255.255.0",
"ethernetLinkAggregation":"ag1"}' -X POST
```

Response example

```
HTTP/1.1 201 OK
{
   "virtualServer" : {
      "UUID" : "b912ecdc-77df-11d0-958c-49e1bb864b23",
      "virtualServerId" : 7,
      "ipAddresses" : [ "172.17.37.63" ],
      "isEnabled" : true,
      "name" : "TestEVS",
      "objectId" : "373a3a3a4f49445f24232140255f56",
      "status" : "ONLINE",
      "type" : "File services",
       }
    "uri" : "https://172.17.11.11:8444/v7/storage/virtual-servers/
373a3a3a4f49445f24232140255f56"
}
```

Rename a virtual server

Renames a virtual server. A unique value renames the virtual server.

HTTP request syntax (URI)

```
PATCH <base_URI>/v7/storage/virtual-servers/{id}
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string/number	Specifies either virtual server object ID or HNAS storage virtual server ID.
name	BODY	Y	string	New name for the virtual server.

Return codes

Code	Data	Description
204	No Data	Virtual server was successfully renamed.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example: Using the virtual server object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/virtual-servers/313a3a3a4f49445f24232140255f56
-d '{"name":"testing1"}' -X PATCH
```

Response example: Using the virtual server object ID

```
HTTP/1.1 204 No Content
```

Request example: Using the HNAS storage virtual server ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/virtual-servers/8 -d '{"name":"testing1"}' -X
PATCH
```

Response example: Using the HNAS storage virtual server ID

```
HTTP/1.1 204 No Content
```

Enable a virtual server

Enables a virtual server. A unique value identifies a virtual server to be enabled. The NFS/CIFS exports on the virtual server may be used after the virtual server is enabled.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/virtual-servers/{id}/enable
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string/number	Specifies either the virtual server object ID or the HNAS storage virtual server ID.

Return codes

Code	Data	Description
204	No Data	Virtual server successfully enabled.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example: Using the virtual server object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/virtual-servers/
333a3a3a4f49445f24232140255f56/enable -X POST
```

Response example: Using the virtual server object ID

```
HTTP/1.1 204 No Content
```

Request example: Using the HNAS storage virtual server ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/virtual-servers/15/enable -X POST
```

Response example: Using the HNAS storage virtual server ID

```
HTTP/1.1 204 No Content
```

Chapter 18: Virtual server resource

Disable a virtual server

Disables a virtual server. A unique value identifies a virtual server to be disabled. The NFS/CIFS exports on a virtual server will stop functioning after the virtual server is disabled.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/virtual-servers/{id}/disable
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string/number	Specifies either the virtual server object ID or the HNAS storage virtual server ID.

Return codes

Code	Data	Description
204	No Data	Virtual server successfully disabled.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example: Using the virtual server object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/virtual-servers/333a3a3a4f49445f24232140255f56/
disable -X POST
```

Response example: Using the virtual server object ID

```
HTTP/1.1 204 No Content
```

Request example: Using the HNAS storage virtual server ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/virtual-servers/15/disable -X POST
```

Response example: Using the HNAS storage virtual server ID

```
HTTP/1.1 204 No Content
```

Delete a virtual server

Deletes a virtual server. A unique value identifies a virtual server to be deleted. You cannot delete an enabled virtual server.

HTTP request syntax (URI)

```
DELETE <base_URI>/v7/storage/virtual-servers/{id}
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string/number	Specifies either virtual server object ID or HNAS storage virtual server ID

Return codes

Code	Data	Description
204	No Data	Virtual server successfully deleted.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Chapter 18: Virtual server resource

Request example: Using the virtual server object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/virtual-servers/333a3a3a4f49445f24232140255f56
-X DELETE
```

Response example: Using the virtual server object ID

```
HTTP/1.1 204 No Content
```

Request example: Using the HNAS storage virtual server ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/virtual-servers/15 -X DELETE
```

Response example: Using the HNAS storage virtual server ID

```
HTTP/1.1 204 No Content
```

Get the node associated with a virtual server

Retrieves the cluster node on which a virtual server resides.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/virtual-servers/{id}/nodes
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string/number	Specifies either virtual server object ID or HNAS storage virtual server ID

Return codes

Code	Data	Description
200	Nodes	Node associated with the virtual server successfully retrieved.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.

Chapter 18: Virtual server resource

Code	Data	Description
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example: Using the virtual server object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/virtual-servers/313a3a3a4f49445f24232140255f56/
nodes
```

Response example: Using the virtual server object ID

```
HTTP/1.1 200 OK
{
    "nodes" : [
        {
            "UUID" : "328511ce-680f-11d1-9001-040400070206",
            "firmwareVersion" : "13.0.4323.05",
            "ipAddresses" : [ "10.0.0.20", "192.168.48.1" ],
            "model" : "G600",
            "name" : "G600-440726-1",
            "nodeId" : 1,
            "objectId" : "313a3a3a4f49445f24232140255f56",
            "serial": "440726",
            "status": "ONLINE"
        }
    ]
}
```

Request example: Using the HNAS storage virtual server ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/virtual-servers/8/nodes
```

Response example: Using the HNAS storage virtual server ID

```
HTTP/1.1 200 OK
{
    "nodes" : [
        {
            "UUID" : "328511ce-680f-11d1-9001-040400070206",
            "firmwareVersion" : "13.0.4323.05",
            "ipAddresses" : [ "10.0.0.20", "192.168.48.1" ],
            "model" : "G600",
```

Chapter 18: Virtual server resource

```
        "name" : "G600-440726-1",
        "nodeId" : 1,
        "objectId" : "313a3a3a4f49445f24232140255f56",
        "serial": "440726",
        "status": "ONLINE"
      }
   ]
}
```

Get IP addresses of a virtual server

Gets all IP addresses of a virtual server.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/virtual-servers/{id}/ip-addresses
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string/number	Specifies either the virtual server object ID or the HNAS storage virtual server ID.

Return codes

Code	Data	Description
200	ipAddresses	List of IP addresses.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example: Using the virtual server object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/virtual-servers/
313a3a3a4f49445f24232140255f56/ip-addresses
```

Response example: Using the virtual server object ID

```
HTTP/1.1 200 OK
{
    "ipAddresses" : [ "172.17.239.142", "192.0.2.2" ]
}
```

Request example: Using the HNAS storage virtual server ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/virtual-servers/8/ip-addresses
```

Response example: Using the HNAS storage virtual server ID

```
HTTP/1.1 200 OK
{
    "ipAddresses" : [ "172.27.5.13", "fdca:f995:220a:480:1:1:22:2" ]
}
```

Add an IP address to a virtual server

Adds an IP address to a virtual server.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/virtual-servers/{id}/ip-addresses
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string/number	Specifies either the virtual server object ID or the HNAS storage virtual server ID.
ipAddress	BODY	Y	string	New IPv4 or IPv6 address.
port	BODY	Y	string	Name of the port.

Chapter 18: Virtual server resource

Name	Type	Required	Values	Description
mask	BODY	Y	string	Netmask must be of the form x.x.x.x for an IPv4 address or specify a prefix length for an IPv6 address.

Return codes

Code	Data	Description
204	No Data	New IP address successfully added.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example: Using the virtual server object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/virtual-servers/
333a3a3a4f49445f24232140255f56/ip-addresses -X POST -d '{ "ipAddress":"203.2.43.2",
"port":"ag1", "mask": "255.255.255.0"}'
```

Response example: Using the virtual server object ID

```
HTTP/1.1 204 No Content
```

Request example: Using the HNAS storage virtual server ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/virtual-servers/15/ip-addresses -X POST -d
'{ "ipAddress":"203.2.43.2", "port":"ag1", "mask": "255.255.255.0"}'
```

Response example: Using the HNAS storage virtual server ID

```
HTTP/1.1 204 No Content
```

Chapter 18: Virtual server resource

Delete an IP address from a virtual server

Deletes an IP address from a virtual server.

HTTP request syntax (URI)

```
DELETE <base_URI>/v7/storage/virtual-servers/{id}/ip-addresses/{ipAddress}
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string/number	Specifies either the virtual server object ID or the HNAS storage virtual server ID.
ipAddress	URI_PARAM	Y	string	IP address to delete.

Return codes

Code	Data	Description
204	No Data	IP address successfully deleted.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example: Using the virtual server object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/virtual-servers/
333a3a3a4f49445f24232140255f56/ip-addresses/245.23.42.23 -X DELETE
```

Response example: Using the virtual server object ID

```
HTTP/1.1 204 No Content
```

Chapter 18: Virtual server resource

Request example: Using the HNAS storage virtual server ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/virtual-servers/15/ip-addresses/245.23.42.23 -
X DELETE
```

Response example: Using the HNAS storage virtual server ID

```
HTTP/1.1 204 No Content
```

Get iSCSI domain

Retrieves the iSCSI domain associated with a virtual server.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/virtual-servers/{id}/iscsi-domain
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string/number	Specifies either virtual server object ID or HNAS storage virtual server ID

Return codes

Code	Data	Description
200	iSCSIDomain	The iSCSI domain name was retrieved successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Chapter 18: Virtual server resource

Request example: Using the virtual server object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/virtual-servers/333a3a3a4f49445f24232140255f56/
iscsi-domain
```

Response example: Using the virtual server object ID

```
HTTP/1.1 200 OK
{
    "iSCSIDomain" : "xyz-iscsiDomain"
}
```

Request example: Using the HNAS storage virtual server ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/virtual-servers/15/iscsi-domain
```

Response example: Using the HNAS storage virtual server ID

```
HTTP/1.1 200 OK
{
    "iSCSIDomain" : "xyzDO9923"
}
```

Set iSCSI domain

Sets the iSCSI domain associated with a virtual server.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/virtual-servers/{id}/iscsi-domain
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string/number	Specifies either the virtual server object ID or the HNAS storage virtual server ID.
iSCSIDomain	BODY	Y	string	iSCSI domain name to be set.

Return codes

Code	Data	Description
204	No Data	iSCSI domain name has been set successfully.
400	No Data	Missing or invalid request contents.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example: Using the virtual server object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/virtual-servers/
333a3a3a4f49445f24232140255f56/iscsi-domain -d '{"iSCSIDomain":"xyz- iscsiDomain"}'
-X POST
```

Response example: Using the virtual server object ID

```
HTTP/1.1 204 No Content
```

Request example: Using the HNAS storage virtual server ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/virtual-servers/15/iscsi-domain -d
'{"iSCSIDomain":"xyz-iscsiDomain"}' -X POST
```

Response example: Using the HNAS storage virtual server ID

```
HTTP/1.1 204 No Content
```

Get iSCSI initiators of a virtual server

Retrieves a list of iSCSI initiators associated with a virtual server.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/virtual-servers/{id}/initiators
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string/number	Specifies either the virtual server object ID or the HNAS storage virtual server ID.

Return codes

Code	Data	Description
200	initiators	List of initiators successfully retrieved.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example: Using the virtual server object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/virtual-servers/
313a3a3a4f49445f24232140255f56/initiators
```

Response example: Using the virtual server object ID

```
HTTP/1.1 200 OK
{
    "initiators" : [
        "iqn.1991-05.com.xyz:example",
        "iqn.1991-05.com.xyz:example1",
        "iqn.1994-05.com.xyz:f13ab9d5d37f"
    ]
}
```

Request example: Using the HNAS storage virtual server ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/virtual-servers/1/initiators
```

Chapter 18: Virtual server resource

Response example: Using the HNAS storage virtual server ID

```
HTTP/1.1 200 OK
{
   "initiators" : [
      "iqn.1991-05.com.xyz:example",
      "iqn.1991-05.com.xyz:example1",
      "iqn.1994-05.com.xyz:f13ab9d5d37f"
   ]
}
```

Get virus scan info of a virtual server

Retrieves virus scan info of a virtual server in the system.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/virtual-servers/{id}/virus-scan
```

Parameters

Name	Type	Required	Values	Description
id	URI_PARAM	Y	string/number	Specifies either the virtual server object ID or the HNAS storage virtual server ID.

Return codes

Code	Data	Description
200	virusScan	Virus scan info of a virtual server successfully retrieved.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example: Using the virtual server object ID

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.11.11:8444/v7/storage/virtual-servers/
313a3a3a4f49445f24232140255f56/virus-scan
```

Response example: Using the virtual server object ID

```
{
   "virusScan" : {
      "areAllFileTypesScanned" : true,
      "fileExtensions" : [
         "ACE",
         "ACM",
         "ACV",
         "ACX",
         "ADT",
         "APP",
         "ASD",
         "ASP",
         "ASX",
         "AVB"
      ],
      "isSuccessful" : true,
      "isVirusScanEnabled" : true,
      "mode" : "ICAP",
      "scanServers" : [
         {
            "enabled" : true,
            "hostName" : "172.27.250.222",
            "ipAddress" : "172.27.250.222",
            "port" : 1344,
            "scanStatus" : 0,
            "serviceName" : "AVSCANRESP"
         },
         {
            "enabled" : true,
            "hostName" : "172.27.250.224",
            "ipAddress" : "172.27.250.224",
            "port" : 1344,
            "scanStatus" : 0,
            "serviceName" : "AVSCANRESP"
         }
      ]
   }
}
```

Chapter 18: Virtual server resource

Chapter 19: Virtual volume resource

The virtual volume resource is a special directory that enables quota management. It generates various email notifications to subscribed receivers indicating an exceeded quota.

Virtual volume object model

The object model describing this resource contains the following objects.

virtualVolume

Attribute	JSON Type	Data Type	Description
objectId	string	string	Consolidated access ID to identify the virtual volume.
virtualVolumeId	number	uint64	Integer value that identifies the virtual volume.
name	string	string	Virtual volume name.
path	string	string	Virtual volume path.
virtualServerId	number	ushort	Virtual server ID where that volume resides.
filesystemId	string	string	File system ID from which to build the virtual volume.
permanentId	number	uint64	Virtual volume permanent ID.
totalVolumeCapacity	number	uint64	Total virtual volume capacity in bytes.
usageVolumeCapacity	number	uint64	Total usage of capacity in bytes.
fileCount	number	uint64	Current number of files on the virtual volume.
emails	array	array	List of emails associated with the virtual volume.

Get virtual volumes

Retrieves virtual volumes in a file system.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/virtual-volumes/{virtualServerId}/{filesystemId}
```

Parameters

Name	Type	Required	Values	Description
virtualServerId	URI_PARAM	Y	number	Virtual server identification number.
filesystemId	URI_PARAM	Y	string	Unique identity of the file system.

Return codes

Code	Data	Description
200	virtualVolume	An array of objects of virtualVolume successfully retrieved.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/virtual-
volumes/5/7547FEE51E4FEC080000000000000000
```

Response example

```
HTTP/1.1 200 OK
{
  "virtualVolumes": [
  {
```

Chapter 19: Virtual volume resource

```
    "emails" : [ "" ],
    "fileCount" : 1,
    "filesystemId" : "75412709E1CB9AAB0000000000000000",
    "objectId" :
"353a3a3a3735343132373039453143423941414230303030303030303030303030303030303a3a3a33",
    "name" : "xyz$name$with$dollar",
    "path" : "/xyz-test",
    "permanentId" : 8449077299320167083,
    "totalVolumeCapacity" : 19520290816,
    "usageVolumeCapacity" : 0,
    "virtualServerId" : 5,
    "virtualVolumeId" : 12
  },
  {
    "emails" : [ "" ],
    "fileCount" : 1,
    "filesystemId" : "75412709E1CB9AAB0000000000000000",
    "objectId" :
"353a3a3a3735343132373039453143423941414230303030303030303030303030303030303a3a3a35",
    "name" : "xyz*name*with*star",
    "path" : "/xyz-*path",
    "permanentId" : 8449077299320167083,
    "totalVolumeCapacity" : 19520290816,
    "usageVolumeCapacity" : 0,
    "virtualServerId" : 5,
    "virtualVolumeId" : 4
  }
  ]
}
```

Get a virtual volume

Retrieves a virtual volume by its ID.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/virtual-volumes/{virtualVolumeObjectId}
```

Parameters

Name	Type	Required	Values	Description
virtualVolumeObjectId	URI_PARAM	Y	string	ID of the virtual volume object.

Return codes

Code	Data	Description
200	virtualVolume	virtualVolume successfully retrieved.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/virtual-volumes/
353a3a3a37353431323730394531434239414142303030303030303030303030303030303a3 a3a3136
```

Response example

```
HTTP/1.1 200 OK
{
  "virtualVolume":{
   "emails" : [ "user1@xyz.com", "user2@xyz.com" ],
   "fileCount" : 1,
   "filesystemId" : "7547FEE51E4FEC080000000000000000",
      "objectId" :
"353a3a3a37353431323730394531434239414142303030303030303030303030303030303a3a3a3136",

   "name" : "xyz-doc-test-0426",
   "path" : "/xyzvirtualvolume/docTest",
   "permanentId" : 8451003485793545224,
   "totalVolumeCapacity" : 4982833152,
   "usageVolumeCapacity" : 0,
   "virtualServerId" : 5,
   "virtualVolumeId" : 12
  }
}
```

Create a virtual volume

Creates a virtual volume.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/virtual-volumes
```

Parameters

Name	Type	Required	Values	Description
virtualServerId	BODY	Y	number	Virtual server ID for newly created virtual volume.
filesystemId	BODY	Y	string	File system ID string to which the virtual volume belongs.
virtualVolumeName	BODY	Y	string	Virtual volume name.
fileSystemPath	BODY	Y	string	Absolute file system path to use for virtual volume. The path should be in UNIX format.
createPathIfNotExists	BODY	N	boolean	Boolean flag to create the path.
emails	BODY	N	array	An array of emails of users interested in receiving various notifications.

Return codes

Code	Data	Description
201	virtualVolume	Virtual volume object returned when created successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.

Code	Data	Description
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 201 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/virtual-volumes -d '{"virtualVolumeName":"xyz-
xyz-unit-test1121", "virtualServerId":5,
"filesystemId":"75412709E1CB9AAB0000000000000000","filesystemPath":"/
xyzVirtualVolumeTest/test-11-21", "createPathIfNotExists":true,"emails":
["user1@xyz.com","user2@xyz.com"]}' -X POST
```

Response example

```
HTTP/1.1 201 Created
{
   "virtualVolume" : {
      "emails" : [ "user1@xyz.com", "user2@xyz.com" ],
      "fileCount" : 1,
      "filesystemId" : "75412709E1CB9AAB0000000000000000",

"objectId" :"363a3a3a4141444444453237394546394332354630303030303030303030303030303030303
a3a3a31",
      "name" : "xyz-unit-test1121",
      "path" : "/xyzVirtualVolumeTest/test-11-21",
      "permanentId" : 8449077299320167083,
      "totalVolumeCapacity" : 19520290816,
      "usageVolumeCapacity" : 0,
      "virtualServerId" : 5,
         "virtualVolumeId" : 12
   },
"uri" : "https://172.17.57.75:8444/v7/storage/virtual-volumes/
363a3a3a4141444444453237394546394332354630303030303030303030303030303030303a3a3a31"
}
```

Update a virtual volume

Updates a virtual volume by its ID.

HTTP request syntax (URI)

```
PATCH <base_URI>/v7/storage/virtual-volumes/{virtualVolumeObjectId}
```

Parameters

Name	Type	Required	Values	Description
virtualVolumeObjectId	URI_PARM	Y	string	ID of the virtual volume object.
newVirtualVolumeName	BODY	Y	string	New virtual volume name.
Emails	BODY	N	array	Optional parameter to specify a list of emails associated with this virtual volume.

Return codes

Code	Data	Description
204	No Data	virtualVolume successfully updated.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/virtual-volumes/
353a3a3a37353431323730394531434239414142303030303030303030303030303030303a3a3a3136 -
d '{"newVirtualVolumeName":"xyz-doc-test-0426","emails": ["user1@xyz.com",
"user2@xyz.com","user3@xyz.com"]}' -X PATCH
```

Response example

```
HTTP/1.1 204 No Content
```

Chapter 19: Virtual volume resource

Delete a virtual volume

Deletes a virtual volume by its ID.

HTTP request syntax (URI)

```
DELETE <base_URI>/v7/storage/virtual-volumes/{virtualVolumeObjectId}
```

Parameters

Name	Type	Required	Values	Description
virtualVolumeObjectId	URI_PARAM	Y	string	ID of the virtual volume object

Return codes

Code	Data	Description
204	No Data	virtualVolume deleted successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/virtual-volumes/
353a3a3a37353431323730394531434239414142303030303030303030303030303030303a3a3a3136 -
X DELETE
```

Response example

```
HTTP/1.1 204 No Content
```

Chapter 20: Virtual volume quota resource

The virtual volume quota resource provides quota management for a virtual volume to which a specific quota is associated.

Virtual volume quota object model

The object model describing this resource contains the following objects.

virtualVolumeQuota

Attribute	JSON Type	Data Type	Description
objectId	string	string	ID of the virtual volume quota object.
virtualVolumeName	string	string	Name of the virtual volume quota object.
virtualVolumeId	number	uint64	ID of the virtual volume.
virtualServerId	number	uint	Virtual server ID of the quota object.
filesystemId	string	string	File system ID of the virtual volume.
quota	object	object	Object that describes the quota setup.

quota

Attribute	JSON Type	Data Type	Description
diskUsage	number	uint64	Actual usage of disk space of the quota in units of bytes.
diskUsageThreshold	object	object	Quota space size usage. The object type is quotaThresholdPercentages.
fileUsage	number	uint64	Actual number of files created under the quota.

Attribute	JSON Type	Data Type	Description
fileCountThreshold	object	object	Quota file number usage. The object type is quotaThresholdPercentages.
logEvent	boolean	boolean	Boolean flag indicating whether events should be logged.
targetDomain	string	string	Target domain. That is, domain of CIFS user or group.
targetName	string	string	Target name. That is, user or group name of quota..
targetType	string	string	Target type. Current supported values are: ▪ USER ▪ GROUP ▪ VIRTUAL_VOLUME

quotaThresholdPercentages

Attribute	JSON Type	Data Type	Description
limit	number	uint64	The upper limit of space size or file numbers.
isHard	boolean	boolean	Indicates whether the limit is a hard limit, that is, whether the limit will be enforced.
reset	number	int	Percent of the amount of space size or file numbers specified in the limit field at which a reset alert is sent. This value cannot exceed the value of the warning alert. A minimum value is required.
warning	number	int	Percent of the amount of space or number of the file count specified in the limit field at which a warning alert is sent. This value cannot exceed the value of the severe alert. A minimum value is required.

Attribute	JSON Type	Data Type	Description
severe	number	int	Percent of the amount of space or number of file count specified in the limit field at which a severe alert is sent. This value cannot be bigger than 100. A minimum value is required.

Get virtual volume quotas for a file system

Retrieves all quotas associated with all virtual volumes in a file system. This includes all quotas of all virtual volumes such as user/group quota of all virtual volumes and anonymous virtual volume quota of all virtual volumes. Note that if there are a lot of virtual volume quotas, this function will return a large amount of data.

> **Note:**
>
> Virtual volume user/group quota is different from file system user/group quota. Virtual volume user/group quota is limited to the quota of the virtual volumes.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/quotas/{virtualServerId}/filesystem/{filesystemId}
```

Parameters

Name	Type	Required	Values	Description
virtualServerId	URI_PARAM	Y	number	Virtual server identifier number.
filesystemId	URI_PARAM	Y	string	Unique identity of the file system.

Return codes

Code	Data	Description
200	virtualVolumeQuota	An array of virtualVolumeQuota objects is returned.
400	No Data	Missing or invalid request contents.

Code	Data	Description
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: Td5qNSpXX4.732uVwjjuN1Wgmxw7yJwL5nygQk79k6pbVg.wvMFqH2"
https://172.27.250.210:8444/v7/storage/quotas/2/filesystem/
AF0BFEDC1CF3BCC70000000000000000
```

Response example

```
HTTP/1.1 200 Ok
{
   "virtualVolumeQuotas" : [
      {
         "filesystemId" : "AF0BFEDC1CF3BCC70000000000000000",
         "objectId" :
"323a3a3a4146304246454443314346334243433730303030303030303030303030303030303a3a3a313a3a3
a303a3a3a4f49445f24232140255f56",
         "quota" : {
            "diskUsage" : 0,
            "diskUsageThreshold" : {
               "isHard" : false,
               "limit" : 0,
               "reset" : 5,
               "severe" : 85,
               "warning" : 75
            },
            "fileCountThreshold" : {
               "isHard" : false,
               "limit" : 0,
               "reset" : 5,
               "severe" : 85,
               "warning" : 75
            },
            "fileUsage" : 0,
            "logEvent" : false,
            "targetDomain" : "",
            "targetName" : "nfsuser",
            "targetType" : "USER"
         },
         "virtualServerId" : 2,
```

Chapter 20: Virtual volume quota resource

```
          "virtualVolumeId" : 1,
          "virtualVolumeName" : "vv1"
      },
      {
          "filesystemId" : "AF0BFEDC1CF3BCC70000000000000000",
          "objectId" :
"323a3a3a4146304246454443314346334243433773030303030303030303030303030303a3a3a313a3a3
a303a3a3a4f49445f24232140255f56",
          "quota" : {
            "diskUsage" : 0,
            "diskUsageThreshold" : {
               "isHard" : true,
               "limit" : 6442450944,
               "reset" : 5,
               "severe" : 85,
               "warning" : 75
            },
            "fileCountThreshold" : {
               "isHard" : false,
               "limit" : 0,
               "reset" : 5,
               "severe" : 85,
               "warning" : 75
            },
            "fileUsage" : 0,
            "logEvent" : false,
            "targetDomain" : "",
            "targetName" : "nfsgroup",
            "targetType" : "GROUP"
          },
          "virtualServerId" : 2,
          "virtualVolumeId" : 1,
          "virtualVolumeName" : "vv1"
      },
      {
          "filesystemId" : "AF0BFEDC1CF3BCC70000000000000000",
          "objectId" :
"323a3a3a4146304246454443314346334243433773030303030303030303030303030303a3a3a313a3a3
a303a3a3a4f49445f24232140255f56",
          "quota" : {
            "diskUsage" : 0,
            "diskUsageThreshold" : {
               "isHard" : true,
               "limit" : 24696061952,
               "reset" : 5,
               "severe" : 85,
               "warning" : 75
            },
            "fileCountThreshold" : {
               "isHard" : true,
               "limit" : 90000,
```

Chapter 20: Virtual volume quota resource

```
            "reset" : 5,
            "severe" : 85,
            "warning" : 75
         },
         "fileUsage" : 1,
         "logEvent" : false,
         "targetDomain" : "",
         "targetName" : "",
         "targetType" : "VIRTUAL_VOLUME"
      },
      "virtualServerId" : 2,
      "virtualVolumeId" : 1,
      "virtualVolumeName" : "vv1"
   },
   {
      "filesystemId" : "AF0BFEDC1CF3BCC70000000000000000",
      "objectId" :
"323a3a3a4146304242645444331434633424343373030303030303030303030303030303a3a3a323a3a3
a303a3a3a4f49445f24232140255f56",
      "quota" : {
         "diskUsage" : 0,
         "diskUsageThreshold" : {
            "isHard" : true,
            "limit" : 1234567890,
            "reset" : 65,
            "severe" : 95,
            "warning" : 75
         },
         "fileCountThreshold" : {
            "isHard" : false,
            "limit" : 1000,
            "reset" : 85,
            "severe" : 95,
            "warning" : 90
         },
         "fileUsage" : 0,
         "logEvent" : true,
         "targetDomain" : "",
         "targetName" : "xyz.com",
         "targetType" : "USER"
      },
      "virtualServerId" : 2,
      "virtualVolumeId" : 2,
      "virtualVolumeName" : "xyzvvol-test1"
   },
   {
      "filesystemId" : "AF0BFEDC1CF3BCC70000000000000000",
      "objectId" :
"323a3a3a4146304242645444331434633424343373030303030303030303030303030303a3a3a323a3a3
a303a3a3a4f49445f24232140255f56",
      "quota" : {
```

Chapter 20: Virtual volume quota resource

```
            "diskUsage" : 0,
            "diskUsageThreshold" : {
               "isHard" : true,
               "limit" : 1234567890,
               "reset" : 65,
               "severe" : 95,
               "warning" : 75
            },
            "fileCountThreshold" : {
               "isHard" : false,
               "limit" : 1000,
               "reset" : 85,
               "severe" : 95,
               "warning" : 90
            },
            "fileUsage" : 0,
            "logEvent" : true,
            "targetDomain" : "BUILTIN",
            "targetName" : "Current Owner",
            "targetType" : "USER"
         },
         "virtualServerId" : 2,
         "virtualVolumeId" : 2,
         "virtualVolumeName" : "xyzvvol-test1"
      },
      {
         "filesystemId" : "AF0BFEDC1CF3BCC70000000000000000",
         "objectId" :
"323a3a3a4146304246454443314346334243433730303030303030303030303030303030303a3a3a323a3a3
a303a3a3a4f49445f24232140255f56",
         "quota" : {
            "diskUsage" : 0,
            "diskUsageThreshold" : {
               "isHard" : true,
               "limit" : 1234567890,
               "reset" : 65,
               "severe" : 95,
               "warning" : 75
            },
            "fileCountThreshold" : {
               "isHard" : false,
               "limit" : 1000,
               "reset" : 85,
               "severe" : 95,
               "warning" : 90
            },
            "fileUsage" : 1,
            "logEvent" : true,
            "targetDomain" : "",
            "targetName" : "",
            "targetType" : "VIRTUAL_VOLUME"
```

Chapter 20: Virtual volume quota resource

```
        },
        "virtualServerId" : 2,
        "virtualVolumeId" : 2,
        "virtualVolumeName" : "xyzvvol-test1"
      }
    ]
}
```

Get a virtual volume quota

Retrieves a virtual volume quota by virtual volume ID.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/virtual-volumes/{vivolObjectId}/quotas
```

Parameters

Name	Type	Required	Values	Description
vivolObjectId	URI_PARAM	Y	string	Virtual volume object ID.

Return codes

Code	Data	Description
200	virtualVolumeQuota	virtualVolumeQuota object is returned.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/virtual-volumes/
353a3a3a37353431323730394531434239414142303030303030303030303030303030303a3a3a3136/
quotas
```

Response example

```
HTTP/1.1 200 Ok
{
  "virtaulVolumeQuota" :
  {
    "filesystemId" : "864553F647F8647D0000000000000000",
    "objectId" :
"333a3a3a38363435353346363437463836343744303030303030303030303030303030303a3a3a313a3a3
a303a3a3a4f49445f24232140255f56",
    "quota" :
    {
      "diskUsage" : 0,
      "diskUsageThreshold" :
      {
        "isHard" : false,
        "limit" : 2147483648,
        "reset" : 5,
        "severe" : 85,
        "warning" : 75
      },
      "fileCountThreshold" :
      {
        "isHard" : false,
        "limit" : 3,
        "reset" : 5,
        "severe" : 85,
        "warning" : 75
      },
      "fileUsage" : 0,
      "logEvent" : true,
      "targetDomain" : "",
      "targetName" : "Network",
      "targetType" : "VIRTUAL_VOLUME"
    },
    "virtualServerId" : 3,
    "virtualVolumeId" : 1,
  }
}
```

Get a virtual volume user quota

Retrieves a virtual volume user quota by virtual volume ID.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/virtual-volumes/{vivolObjectId}/quotas/user
```

Parameters

Name	Type	Required	Values	Description
vivolObjectId	URI_PARAM	Y	string	Virtual volume object ID.
userName	BODY	Y	string	User name. If the user is a Windows user, then use the format "DOMAIN\\username" or if the user is a UNIX user, use the format "username" .

Return codes

Code	Data	Description
200	quota	Virtual volume user quota object is returned.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/virtual-volumes/
323a3a3a373534364439364539414341444544373030303030303030303030303030303a3a3a313a3a
3a303a3a3a4f49445f24232140255f56/quotas/user -d '{"userName":"514"}' -X GET
```

Response example

```
HTTP/1.1 200 OK
{
  "quota": {
    "diskUsage": 0,
    "diskUsageThreshold": {
      "isHard": true,
      "limit": 0,
      "reset": 5,
      "severe": 0,
```

Chapter 20: Virtual volume quota resource

```
      "warning": 0
    },
    "fileCountThreshold": {
      "isHard": true,
      "limit": 0,
      "reset": 5,
      "severe": 0,
      "warning": 0
    },
    "fileUsage": 0,
    "logEvent": false,
    "targetDomain": "",
    "targetName": "514",
    "targetType": "USER"
  }
}
```

Get a virtual volume group quota

Retrieve a virtual volume group quota by virtual volume ID.

HTTP request syntax (URI)

```
GET <base_URI>/v7/storage/virtual-volumes/{vivolObjectId}/quotas/group
```

Parameters

Name	Type	Required	Values	Description
vivolObjectId	URI_PARAM	Y	string	Virtual volume object ID.
groupName	BODY	Y	string	Group name. If the group is a Windows group then use the format "DOMAIN\ \groupname" or if the group is a UNIX group, use the format "groupname".

Return codes

Code	Data	Description
200	quota	Virtual volume group quota object is returned.

Chapter 20: Virtual volume quota resource

Code	Data	Description
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 200 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/virtual-volumes/
323a3a3a37353436443936453941434144454437303030303030303030303030303a3a3a313a3a
3a303a3a3a4f49445f24232140255f56/quotas/group -d'{"groupName":"development2"}' -X
GET
```

Response example

```
HTTP/1.1 200 OK
{
  "quota": {
    "diskUsage": 0,
    "diskUsageThreshold": {
      "isHard": true,
      "limit": 0,
      "reset": 5,
      "severe": 0,
      "warning": 0
    },
    "fileCountThreshold": {
      "isHard": true,
      "limit": 0,
      "reset": 5,
      "severe": 0,
      "warning": 0
    },
    "fileUsage": 0,
    "logEvent": false,
    "targetDomain": "",
    "targetName": "development2",
    "targetType": "GROUP"
  }
}
```

Create a virtual volume quota

Creates a virtual volume quota for a virtual volume, by virtual volume ID. Only a single virtual volume quota can be associated with a virtual volume.

HTTP request syntax (URI)

```
POST <base_URI>/v7/storage/virtual-volumes/{vivolObjectId}/quotas
```

Parameters

Name	Type	Required	Values	Description
vivolObjectId	URI_PARAM	Y	string	ID of the virtual volume object to create the quota for.
diskUsageThreshold	BODY	Y	object	quotaThresholdPercentages object of disk space to be added to the virtual volume.
fileCountThreshold	BODY	Y	object	quotaThresholdPercentages object of file count to be added to the virtual volume.
logEvent	BODY	Y	boolean	Whether to log an event.

Return codes

Code	Data	Description
201	virtualVolumeQuota	virtualVolumeQuota object is returned.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 201 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/virtual-volumes/
353a3a3a373534313237303945314342394141423030303030303030303030303030303a3a3a3136/
quotas -d '{"diskUsageThreshold": {"limit":1234567890, "isHard":true,"reset":65,
"warning":75,"severe":95},"fileCountThreshold":{"limit":1000,"isHard":false,
"reset":85,"warning":90,"severe":95}, "logEvent":true}' -X POST
```

Response example

```
HTTP/1.1 201 Created
{
    "virtualVolumeQuota" : {
        "filesystemId" : "75412709E1CB9AAB0000000000000000",
        "objectId" :
"353a3a3a373534313237303945314342394141423030303030303030303030303030303a3a3a3136",
        "quota" : {
            "diskUsage" : 0,
            "diskUsageThreshold" : {
                "isHard" : true,
                "limit" : 1234567890,
                "reset" : 65,
                "severe" : 95,
                "warning" : 75
            },
            "fileCountThreshold" : {
                "isHard" : false,
                "limit" : 1000,
                "reset" : 85,
                "severe" : 95,
                "warning" : 90
            },
            "fileUsage" : 1,
            "logEvent" : true
        },
        "virtualServerId" : 5,
        "virtualVolumeId" : 16,
        "virtualVolumeName" : "xyz-unit-test2"
    },
    "uri" : "https://172.17.57.75:8444/v7/storage/quotas/
353a3a3a373534313237303945314342394141423030303030303030303030303030303a3a3a3136"
}
```

Modify a virtual volume quota

Modifies a virtual volume quota by virtual volume ID.

HTTP request syntax (URI)

```
PATCH <base_URI>/v7/storage/virtual-volumes/{vivolObjectId}/quotas
```

Parameters

Name	Type	Required	Values	Description
vivolObjectId	URI_PARAM	Y	string	Virtual volume ID.
diskUsageThreshold	BODY	Y	object	quotaThresholdPercentages object of the disk space to be added to the virtual volume.
fileCountThreshold	BODY	Y	object	quotaThresholdPercentages object of the file count to be added to the virtual volume.
logEvent	BODY	Y	boolean	Indicates whether to log an event.

Return codes

Code	Data	Description
204	No Data	Virtual volume quota successfully modified.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/virtual-volumes/
353a3a3a373534313237303945314342394141423030303030303030303030303030303a3a3a3136/
quotas -d '{"diskUsageThreshold": {"limit":1234567890,"isHard":true,"reset":65,
"warning":75,"severe":95},"fileCountThreshold":{"limit":1000,"isHard":false,
"reset":85,"warning":90,"severe":95}, "logEvent":true}' -X PATCH
```

Response example

```
HTTP/1.1 204 No Content
```

Delete a virtual volume quota

Deletes a virtual volume quota by virtual volume ID.

HTTP request syntax (URI)

```
DELETE <base_URI>/v7/storage/virtual-volumes/{vivolObjectId}/quotas
```

Parameters

Name	Type	Required	Values	Description
vivolObjectId	URI_PARAM	Y	string	ID of the virtual volume object.

Return codes

Code	Data	Description
204	No Data	Virtual volume quota deleted successfully.
400	No Data	Missing or invalid request contents.
404	Error Message	Requested resource not found.
500	Error message	Error associated with the storage system.
501	No Data	Server has not implemented the request operation on the resource.

Any HTTP status code other than 204 indicates that the API did not complete successfully.

Request example

```
curl -vk -H "X-Api-Key: zrxvSDAv9x.RIP4gkmKarG3beF.or.4Tc2im7oeqYN88C9XPGHxbXC"
https://172.17.57.75:8444/v7/storage/virtual-volumes/
353a3a3a37353431323730394531434239414142303030303030303030303030303a3a3a3136/
quotas -X DELETE
```

Response example

```
HTTP/1.1 204 No Content
```

Chapter 20: Virtual volume quota resource

Chapter 21: Troubleshooting

This chapter provides steps for troubleshooting.

File storage failure cases

Failure Case 1

Incorrect cluster IP address of the HNAS system provided in the HTTP Request header. To provide the correct IP address, see Acquiring the correct admin EVS IP address (on page 17). Also, you may have mistakenly used the SMU IP address in the HTTP request header. If so, change it to the Admin EVS IP address.

Failure Case 2

Incorrect credential for the HNAS storage system provided in the HTTP Request header. To validate the HNAS storage system credentials being used in the HTTP request, see Authenticating with the REST API server (on page 16). Also, you may have mistakenly used the SMU credentials in the HTTP request header. If so, change them to the HNAS cluster credentials.

Failure Case 3

Wrong REST Server API mode. If the REST server is running in legacy mode, some of the API calls mentioned in this document may not work, and others may return less information than is expected. Make sure that the output from the `rest-server-status` command indicates that the REST Server mode is **native**, and that the server is **running**. See "Enabling Native REST API" Section for more details.

Failure Case 4

REST server fails to start when switching modes. When switching between legacy and native modes, the REST server may sometimes fail to start, this is usually due to the legacy REST server(s) not shutting down instantaneously and holding onto the TCP port. The native REST server is not able to start as the TCP port is not available for it to bind to. Waiting for 30 seconds and retrying the start request again will normally fix this issue. This is more likely to happen on a cluster, as the REST server needs to be stopped on all nodes before it can be restarted in the new mode.

Hitachi Vantara