

IBM Spectrum Copy Data Management[™] 2.2.12.0

Storage Plugin REST API

Table of Contents

Table of Contents.....	1
Introduction.....	9
Overview.....	10
Conventions.....	11
JSON.....	11
Headers.....	11
Verbs.....	11
Status Codes.....	11
Pagination.....	12
Actions.....	12
Action Request Body.....	12
Application Programming Interface.....	13
API Root Endpoint.....	13
API Required Headers.....	13
Resources.....	13
Plugin Information.....	14
Endpoints.....	14
Verbs.....	14
GET /info.....	14
Description.....	14
Request.....	14
Body.....	14
Response.....	14
Body.....	14
Status Codes.....	15
Session.....	15
Endpoints.....	15
Verbs.....	15
POST /session.....	15
Description.....	15
Request.....	15
Body.....	15
Response.....	16
Body.....	16
Status Codes.....	16
Log.....	16
Endpoints.....	16
Verbs.....	16
GET /log.....	16
Description.....	16
Request.....	17
Body.....	17
Response.....	17
Status Codes.....	17
Storage.....	17
Endpoints.....	17
Verbs.....	17
GET /storage.....	17
Description.....	17
Request.....	17

Body	17
Headers	17
Example Request	18
Response.....	18
Body	18
Status Codes	18
Server.....	18
Endpoints.....	19
Verbs	19
GET /server	19
Description	19
Request.....	19
Body	19
Headers	19
Response.....	19
Body	19
Status Codes	20
GET /server/<server_id>	20
Description	20
Request.....	20
Body	20
Headers	20
Response.....	21
Body	21
Status Codes	22
GET /server/<server_id>/system.....	22
Description	22
Request.....	22
Body	22
Headers	22
Response.....	22
Body	22
Status Codes	23
Volume	23
Endpoints.....	23
Verbs	23
GET /server/<server_id>/volume.....	23
Description	23
Request.....	24
Body	24
Headers	24
Response.....	24
Body	24
Status Codes	25
GET /server/<server_id>/volume/<volume_id>	25
Description	25
Request.....	25
Body	25
Headers	25
Response.....	25
Body	25
Status Codes	26
DELETE /server/<server_id>/volume/<volume_id>	26
Description	26
Request.....	27

Body	27
Headers	27
Response.....	27
Body	27
Status Codes	27
GET /server/<server_id>/volume/<volume_id>/splitstatus	27
Description	27
Request.....	27
Body	27
Headers	27
Response.....	27
Body	27
Status Codes	28
Actions.....	28
POST /server/<server_id>/volume/<volume_id>?action=clone.....	28
Description	28
Request.....	28
Body	28
Headers	28
Response.....	28
Body	28
Status Codes	29
POST /server/<server_id>/volume/<volume_id>?action=mount	29
Description	29
Request.....	30
Body	30
Headers	30
Response.....	30
Body	30
Status Codes	30
POST /server/<server_id>/volume/<volume_id>?action=split	30
Description	30
Request.....	30
Body	30
Headers	30
Response.....	30
Body	30
Status Codes	31
POST /server/<server_id>/volume/<volume_id>?action=restore	31
Description	31
Request.....	31
Body	31
Headers	31
Response.....	31
Body	31
Status Codes	32
POST /server/<server_id>/volume/<volume_id>?action=restore-to-alternate	32
Description	32
Request.....	32
Body	32
Headers	33
Response.....	33
Body	33
Status Codes	34
POST /server/<server_id>/volume/<volume_id>?action=restore-to-new.....	34

Description	34
Request.....	34
Body	34
Headers	34
Response.....	34
Body	34
Status Codes	35
LUN	36
Endpoints.....	36
Verbs.....	36
GET /server/<server_id>/lun.....	36
Description	36
Request.....	36
Body	36
Headers	36
Response.....	36
Body	36
Status Codes	37
GET /server/<server_id>/lun/<lun_id>	37
Description	37
Request.....	37
Body	37
Headers	37
Response.....	37
Body	37
Status Codes	38
Actions.....	38
POST /server/<server_id>/lun/<lun_id>?action=online.....	38
Description	38
Request.....	38
Body	38
Headers	38
Response.....	38
Body	38
Status Codes	38
Snapshot.....	38
Endpoints.....	39
Verbs.....	39
GET /server/<server_id>/volume/<volume_id>/snapshot	39
Description	39
Request.....	39
Body	39
Headers	39
Response.....	39
Body	39
Status Codes	40
GET /server/<server_id>/volume/<volume_id>/snapshot/<snapshot_id>	40
Description	40
Request.....	40
Body	40
Headers	40
Response.....	40
Body	40
Status Codes	40
POST /server/<server_id>/volume/<volume_id>/snapshot	41

Description	41
Request.....	41
Body	41
Headers	41
Response.....	41
Body	41
Status Codes	42
DELETE /server/<server_id>/volume/<volume_id>/snapshot/<snapshot_id>	42
Description	42
Request.....	42
Body	42
Headers	42
Response.....	42
Body	42
Status Codes	42
Host.....	42
Endpoints.....	42
Verbs.....	43
GET /server/<server_id>/host	43
Description	43
Request.....	43
Body	43
Headers	43
Response.....	43
Body	43
Status Codes	44
GET /server/<server_id>/host/<host_id>	44
Description	44
Request.....	44
Body	44
Headers	44
Response.....	44
Body	44
Status Codes	45
POST /server/<server_id>/host.....	45
Description	45
Request.....	45
Body	45
Headers	45
Response.....	45
Body	45
Status Codes	46
GET /server/<server_id>/host/<host_id>/lun	46
Description	46
Request.....	46
Body	46
Headers	46
Response.....	46
Body	46
Status Codes	47
Actions.....	47
POST /server/<server_id>/host/<host_id>?action=map	47
Description	47
Request.....	47
Body	47

Headers	47
Response.....	47
Body	47
Status Codes	48
POST /server/<server_id>/host/<host_id>?action=unmap.....	48
Description	48
Request.....	48
Body	48
Headers	48
Response.....	48
Body	48
Status Codes	48
Host Group	49
Endpoints.....	49
Verbs.....	49
GET /server/<server_id>/hostgroup.....	49
Description	49
Request.....	49
Body	49
Headers	49
Response.....	49
Body	49
Status Codes	49
Actions.....	50
POST /server/<server_id>/hostgroup/<host_group_id>?action=map	50
Description	50
Request.....	50
Body	50
Headers	50
Response.....	50
Body	50
Status Codes	50
Network.....	51
Endpoints.....	51
Verbs.....	51
GET /server/<server_id>/network.....	51
Description	51
Request.....	51
Body	51
Headers	51
Response.....	51
Body	51
Status Codes	52
GET /server/<server_id>/network/<network_id>.....	52
Description	52
Request.....	52
Body	52
Headers	52
Response.....	52
Body	52
Status Codes	53
Pool.....	53
Endpoints.....	53
Verbs.....	54
GET /server/<server_id>/pool.....	54

Description	54
Request.....	54
Body	54
Headers	54
Response.....	54
Body	54
Status Codes	54
GET /server/<server_id>/pool/<pool_id>	54
Description	54
Request.....	54
Body	54
Headers	55
Response.....	55
Body	55
Status Codes	55
Consistency Group	55
Endpoints.....	55
Verbs	56
GET /server/<server_id>/consistencygroup	56
Description	56
Request.....	56
Body	56
Headers	56
Response.....	56
Body	56
Status Codes	56
GET /server/<server_id>/consistencygroup/<consistency_group_id>	57
Description	57
Request.....	57
Body	57
Headers	57
Response.....	57
Body	57
Status Codes	57
Actions.....	58
POST /server/<server_id>/consistencygroup/<consistencygroup_id>?action=add-volume	58
Description	58
Request.....	58
Body	58
Headers	58
Response.....	58
Body	58
Status Codes	58
POST /server/<server_id>/consistencygroup/<consistencygroup_id>?action=remove-volume	58
Description	58
Request.....	59
Body	59
Headers	59
Response.....	59
Body	59
Status Codes	59
POST /server/<server_id>/consistencygroup/<consistencygroup_id>?action=condense.....	59
Description	59
Request.....	59
Body	59

Headers	59
Response.....	60
Body	60
POST /server/<server_id>/consistencygroup/<consistencygroup_id>?action=take-snapshot.....	60
Description	60
Request.....	60
Body	60
Headers	60
Response.....	60
Body	60
Status Codes	61
Consistency Group Snapshots.....	61
Endpoints.....	61
Verbs	61
GET /server/<server_id>/consistencygroup/<consistency_group_id>/snapshot	61
Description	61
Request.....	61
Body	61
Headers	61
Response.....	61
Body	61
Status Codes	62

Introduction

IBM Spectrum Copy Data Management (CDM) handles many different types of storage for supporting backup and restore. To achieve this, it needs to issue a well-defined set of commands to the underlying storage to execute such actions as the creation of a volume or to take a snapshot. A set of commands is now defined in the form of RESTful (REST) Application Programming Interfaces (APIs) so that a specific implementation can be provided for each type of storage.

Overview

This document describes the architectural overview of the IBM Spectrum CDM storage provider/plugin REST API that should be supported to enable IBM Spectrum CDM to manage the storage.

The documentation in this guide provides information for the IBM Spectrum CDM storage provider/plugin API based on the REST framework. Storage providers should implement this REST API to allow the IBM Spectrum CDM products to connect to your storage in order to query information about storage objects and to perform basic operations by using the hypertext transfer protocol (HTTP), secure hypertext transfer protocol (HTTPS) and the principles of REST APIs.

This application programming interface uses REST architecture designed to work with web-based applications in a simplified way, by using four basic HTTP methods for applications to interact with: GET, POST, PUT, and DELETE.

Detailed in this guide are the methods and the required Uniform Resource Identifiers (URIs) necessary for a developer's application to complete the action listed. This Storage Plugin REST API Guide contains several interaction examples that demonstrate basic functions such as obtaining a session ID, executing a search, registering a storage provider, and snapshot and recovery life cycle actions.

Conventions

The following are conventions that cover data formats, URL formats, etc.

JSON

REST API requests and responses are in JSON format.

Headers

The following headers are required for every REST API operation:

Header	Description
x-storageplugin-sessionid	Required with a valid session ID value.
Content-Type	Required with the value set to <code>application/json</code> .
Accept	Required with the value set to <code>application/json</code> .

Verbs

IBM Spectrum CDM tries to adhere as closely as possible to standard HTTP and REST conventions in its use of HTTP verbs.

Verb	Description
GET	Used to retrieve a resource.
POST	Used to create a new resource.
PUT	Used to update an existing resource, including partial updates.
DELETE	Used to delete an existing resource.

Status Codes

IBM Spectrum CDM tries to adhere as closely as possible to standard HTTP and REST conventions in its use of HTTP status codes.

Status Code	Usage
200 OK	The request completed successfully.

201 Created	A new resource has been created successfully. The resource's URI is available from the response's <code>Location</code> header.
204 No Content	An update to an existing resource has been applied successfully.
400 Bad Request	The request was malformed. The response body will include an error providing further information.
404 Not Found	The requested resource did not exist.

Pagination

Many `GET` REST APIs support pagination using the following syntax:

```
https://<ip>:<port>/storageplugin/api/storage/xyz/volume?limit=25&start=50
```

Actions

REST API leverages HTTP verbs to perform resource-based operations. These verbs are typically mapped to CRUD operations: create, read, update, delete. Create maps to `POST`, Read maps to `GET`, Update maps to `PUT`, and Delete maps to `DELETE`. Missing from the standard REST API list of verbs is one that maps to the semantics of an action. IBM Spectrum CDM leverages the `POST` verb to perform actions. The `POST` verb along with a JSON request body manages the execution of various actions. The action is specified as a query parameter called `action` on the URI. The following is an example of performing an action:

```
https://<ip>:<port>/storageplugin/api/storage/xyz/volume/{volumeId}?action=createSnapshot
```

Action Request Body

```
{
  "name": "snapshot.1"
}
```

Application Programming Interface

API Root Endpoint

All storage plugin API endpoints use the HTTP or HTTPS prefix:

```
http://<ip>:<port>/storageplugin/api  
https://<ip>:<port>/storageplugin/api
```

API Required Headers

The following headers are required for every REST API operation:

Header	Description
x-storageplugin-sessionid	The session associated with a request. This session ID can be obtained from the <code>session</code> endpoint.
Content-Type	The content type of the body associated with a request. Currently, only <code>application/json</code> is supported.
Accept	The expected content type of the body associated with a response. Currently, only <code>application/json</code> is supported.

Resources

API endpoints are supported for the following resources:

- "Plugin Information" on page 14
- "Session" on page 15
- "Log" on page 16
- "Storage" on page 17
- "Server" on page 18
- "Volume" on page 23
- "LUN" on page 36 (Required for block storage.)
- "Snapshot" on page 38
- "Host" on page 42
- "Host Group" on page 49
- "Network" on page 51
- "Pool" on page 53
- "Consistency Group" on page 55

- "Consistency Group Snapshots" on page 61

Plugin Information

Information about the plugin itself can be obtained using the `info` endpoint. Information includes the storage type name, a unique plugin ID, the version, and a link to the logo in bitmap format. Each storage plugin manages a single type of storage. There can be many different instances (storage arrays/nodes) managed but they must all be of the same storage type. Authentication or session is not required.

Endpoints

- `http://<ip>:<port>/storageplugin/api/info`

Verbs

GET `/info`

Description

Obtains general information about the storage plugin.

Request

Body

None

Response

Body

```
{
  "name": "<name>",
  "id": "<id>",
  "version": "<version>",
  "imageUrl": "<image_url>"
  "iconUrl": "<icon_url>"
}
```

where:

- **name:** The display name of the plugin.
- **id:** The unique ID of the plugin. Only one plugin with this ID can be registered with IBM Spectrum CDM. (Note: This ID is case-sensitive.)
- **version:** The version of the plugin, such as "1.0.0".
- **image_url:** An RFC-2397 URL that can be used by IBM Spectrum CDM to display an image associated with the plugin. (Restriction: The image must be 48 x 48 pixels.)

- **icon_url:** An RFC-2397 URL that can be used by IBM Spectrum CDM to display an icon associated with the plugin. (Restriction: The icon must be 16 x 16 pixels.)

Status Codes

Status Code	Description
200 OK	The information was successfully retrieved.

Session

The `session` endpoint is used to authenticate to a specific storage array or node managed by the plugin and obtain a unique session ID for which to use for subsequent API calls. Please note that the target storage array or node is specified during the creation of a session by the IP address or hostname passed via the `POST` command. The storage plugin tracks active sessions since each session could be to a different storage array or node. REST is state-less but state can be simulated using sessions.

Endpoints

- `http://<ip>:<port>/storageplugin/api/session`

Verbs

POST /session

Description

Creates a new session.

Request

Body

```
{
  "hostname": "<hostname>",
  "ipAddress": "<ip>",
  "port": <port>
}
```

where:

- **hostname:** The hostname associated with the storage for which the session is to be created. This may be the same as the IP address.
- **ipAddress:** The IP address associated with the storage for which the session is to be created.

- **port:** The port number associated with the storage for which the session is to be created.

Response

Body

```
{
  "sessionId": "<session_id>",
  "storageId": "<storage_id>"
}
```

where:

- **session_id:** The ID associated with the created session.
- **storage_id:** The ID of the storage associated with the created session.

Status Codes

Status Code	Description
201 Created	The session was successfully created.
401 Unauthorized	The session could not be created due to an authorization failure, such as incorrect username or password.

Log

The log endpoint allows clients to download the logs associated with the plugin.

Endpoints

- `http://<ip>:<port>/storageplugin/api/log`

Verbs

GET /log

Description

Downloads the logs associated with the plugin.

The downloaded format of the logs depends on the value of the Accept header in the request. Currently, the following types are supported:

File Type	Accept Header Value
.zip	application/zip application/octet-stream

Request

Body

None

Response

The compressed logs in the format

Status Codes

Status Code	Description
201 Created	The session was successfully created.
401 Unauthorized	The session could not be created due to an authorization failure, such as incorrect username or password.

Storage

The `storage` endpoint is used to get information about a storage array or node.

Note: The storage plugin can manage multiple storage arrays or nodes of the same storage type. The specific storage array being queried or managed via this API is determined by the session when it is created. For more information, see the Session section.

Endpoints

- `http://<ip>:<port>/storageplugin/api/storage`

Verbs

GET `/storage`

Description

Retrieve the storage associated with the supplied session ID.

Request

Body

None

Headers

Header	Description
<code>x-storageplugin-sessionid</code>	The session ID associated with the request.

Example Request

Response

Body

```
{
  "name": "<name>",
  "id": "<id>",
  "hostName": "<hostname>",
  "ipAddress": "<ip>",
  "port": <port>,
  "pluginId": "<plugin_id>",
  "vendor": "<vendor>",
  "version": "<version>",
  "serverIds": [<server_ids>]
}
```

where:

- **name:** The name of the storage.
- **id:** The ID of the storage. This ID matches the `storage_id` found in the `session` endpoint.
- **hostName:** The hostname of the storage.
- **ip:** The IP address of the storage.
- **port:** The port number of the storage.
- **plugin_id:** The ID of this plugin.
- **vendor:** The vendor that supports this storage. For example, if the storage provider is ABC Storage, the vendor is ABC.
- **server_ids:** The IDs of the servers associated with this storage.

Status Codes

Status Code	Description
200 OK	The storage information was successfully retrieved.
415 Unsupported Media Type	The media type in the Accept request header is not supported.

Server

Use the `server` endpoint to obtain information about a storage virtual server. Some storage arrays or nodes such as NetApp vServers support this. If storage does not support virtual servers then it should be a list/collection of size "1" with the information of the storage array or node itself filled in as the one virtual server with the `server_id` field set to "0".

Endpoints

- `http://<ip>:<port>/storageplugin/api/server`
- `http://<ip>:<port>/storageplugin/api/server/<server_id>`
- `http://<ip>:<port>/storageplugin/api/server/<server_id>/system`

Verbs

GET `/server`

Description

Retrieves all existing servers corresponding to the storage associated with the supplied session ID.

Request

Body

None

Headers

Header	Description
<code>x-storageplugin-sessionid</code>	The session ID associated with the request.

Response

Body

```
{
  "servers": [
    {
      "id": "<id>",
      "name": "<name>",
      "supports": {
        "replication": <replication>,
        "volumeReplication": <volume_replication>,
        "consistencyGroupReplication": <consistency_group_replication>,
        "newSnapshotReplication": <new_snapshot_replication>,
        "existingSnapshotReplication": <existing_snapshot_replication>,
        "replicationToItself": <replication_to_itself>,
        "deleteFromSnapshotGroup": <delete_from_snapshot_group>,
        "persistentConsistencyGroup": <persistent_consistency_group>,
        "singleConsistencyGroup": <single_consistency_group>
      },
      "config": {
        "isDpDestination": <is_dp_destination>,
        "isNfsServiceEnabled": <is_nfs_service_enabled>,
        "isIscsiServiceEnabled": <is_iscsi_service_enabled>,
        "isIncrementalConsistencyGroup":
<is_incremental_consistency_group>
      },
      ...
    }
  ]
}
```

where:

- **name:** The name of the server.
- **id:** The ID of the server.
- **replication:** Does the server support replication?
- **volume_replication:** Does the server support volume replication?
- **consistency_group_replication:** Does the server support consistency group replication?
- **new_snapshot_replication:** Does the server support new snapshot replication?
- **existing_snapshot_replication:** Does the server support existing snapshot replication?
- **replication_to_itself:** Does the server support replication to itself?
- **delete_from_snapshot_group:** Does the server support delete from snapshot group?
- **persistent_consistency_group:** Does the server support persistent consistency groups?
- **single_consistency_group:** Does the server support single consistency groups?
- **is_dp_destination:** Is the server configured as a DP destination?
- **is_nfs_service_enabled:** Is the NFS service enabled on the server?
- **is_iscsi_service_enabled:** Is the iSCSI service enabled on the server?
- **is_incremental_consistency_group:** Is the server configured as an incremental consistency group?

Status Codes

Status Code	Description
200 OK	The list of existing server(s) was successfully retrieved.
401 Unauthorized	A session ID was not provided.

GET /server/<server_id>

Description

Retrieves existing server with the supplied **server_id** corresponding to the storage associated with the supplied session ID.

Request

Body

None

Headers

Header	Description
--------	-------------

x-storageplugin-sessionid

The session ID associated with the request.

Response

Body

```
{
  "id": "<id>",
  "name": "<name>",
  "supports": {
    "replication": "<replication>",
    "volumeReplication": "<volume_replication>",
    "consistencyGroupReplication": "<consistency_group_replication>",
    "newSnapshotReplication": "<new_snapshot_replication>",
    "existingSnapshotReplication": "<existing_snapshot_replication>",
    "replicationToItself": "<replication_to_itself>",
    "deleteFromSnapshotGroup": "<delete_from_snapshot_group>",
    "persistentConsistencyGroup": "<persistent_consistency_group>",
    "singleConsistencyGroup": "<single_consistency_group>"
  },
  "config": {
    "isDpDestination": "<is_dp_destination>",
    "isNfsServiceEnabled": "<is_nfs_service_enabled>",
    "isIscsiServiceEnabled": "<is_iscsi_service_enabled>",
    "isIncrementalConsistencyGroup":
"<is_incremental_consistency_group>"
  }
}
```

where:

- **name:** The name of the server.
- **id:** The ID of the server.
- **replication:** Does the server support replication?
- **volume_replication:** Does the server support volume replication?
- **consistency_group_replication:** Does the server support consistency group replication?
- **new_snapshot_replication:** Does the server support new snapshot replication?
- **existing_snapshot_replication:** Does the server support existing snapshot replication?
- **replication_to_itself:** Does the server support replication to itself?
- **delete_from_snapshot_group:** Does the server support delete from snapshot group?
- **persistent_consistency_group:** Does the server support persistent consistency groups?
- **single_consistency_group:** Does the server support single consistency groups?
- **is_dp_destination:** Is the server configured as a DP destination?

- **is_nfs_service_enabled:** Is the NFS service enabled on the server?
- **is_iscsi_service_enabled:** Is the iSCSI service enabled on the server?
- **is_incremental_consistency_group:** Is the server configured as an incremental consistency group?

Status Codes

Status Code	Description
200 OK	The list of existing server(s) was successfully retrieved.
401 Unauthorized	A session ID was not provided.
404 Not Found	The server associated with the supplied server_id does not exist.

GET /server/<server_id>/system

Description

Retrieves the system information associated with a server

Request

Body

None

Headers

Header	Description
x-storageplugin-sessionid	The session ID associated with the request

Response

Body

```
{
  "capacityInBytes": <capacity_in_bytes>,
  "systemSizeInBytes": <system_size_in_bytes>,
  "snapshotsSizeInBytes": <snapshots_size_in_bytes>,
  "volumesSizeInBytes": <volumes_size_in_bytes>,
  "dataReduction": <data_reduction>,
  "totalSizeInBytes": <total_size_in_bytes>,
  "sharedSpaceInBytes": <shared_space_in_bytes>,
  "thinProvisioning": <thin_provisioning>,
  "totalReduction": <total_reduction>,
  "provisionedSizeInBytes": <provisioned_size_in_bytes>,
  "parity": "<parity>",
  "serialNumber": "<serial_number>",
  "model": "<model>",
  "version": "<version>"
}
```

where:

- **capacity_in_bytes:** The capacity of the server, in bytes.

- **system_size_in_bytes**: The number of bytes used by the system.
- **snapshots_size_in_bytes**: The size of all snapshots, in bytes.
- **volumes_size_in_bytes**: The size of all volumes, in bytes.
- **data_reduction**: The data reduction ratio (a decimal value).
- **total_size_in_bytes**: Total size of the server, in bytes.
- **shared_space_in_bytes**: The shared space, in bytes.
- **thin_provisioning**: The thin provisioning ratio (a decimal value).
- **total_reduction**: The total reduction ratio (a decimal value).
- **provisioned_size_in_bytes**: The provisioned size of the server, in bytes.
- **parity**: The parity setting for the server.
- **serial_number**: The serial number of the server.
- **mode**: The model of the server.
- **version**: The version of the server.

Status Codes

Status Code	Description
200 OK	The server system information was successfully retrieved.
401 Unauthorized	A session ID was not provided.

Volume

The `/volume` endpoint provides storage volumes provision information for client hosts.

Endpoints

- `http://<ip>:<port>/storageplugin/api/server/<server_id>/volume`
- `http://<ip>:<port>/storageplugin/api/server/<server_id>/volume/<volume_id>`

Verbs

GET `/server/<server_id>/volume`

Description

Retrieves all existing volumes associated with the supplied **server_id** and session ID.

Request

Body

None

Headers

Header	Description
x-storageplugin-sessionid	The session ID associated with the request

Response

Body

```
{
  "volumes": [
    {
      "id": "<id>",
      "name": "<name>",
      "creationTime": "<creation_time>",
      "serverGroupKey": "<server_group_key>",
      "poolId": "<pool_id>",
      "volumeType": "<volume_type>",
      "stats": {
        "size": "<size>",
        "usedSize": "<used_size>",
      },
      "supports": {
        "revert": "<revert>",
        "split": "<split>",
        "autoGrowVadpRep": "<auto_grow_vadp_rep>"
      },
      "config": {
        "thinProvisioned": "<thin_provisioned>"
      }
    },
    ...
  ]
}
```

where:

- **name:** The name of the volume.
- **id:** The ID of the volume.
- **creation_time:** An epoch date representing the creation time of the volume.
- **server_group_key:** The key of the server group to which the volume belongs.
- **pool_id:** The ID of the pool to which the volume belongs.
- **volume_type:** The type of the volume, including vendor-specific classification, if applicable.
- **size:** The size, in bytes, of the volume.
- **used_size:** The number of bytes currently used (filled) by the volume.

- **revert:** Does the volume support revert?
- **split:** Does the volume support splitting? This value must be one of the following:
 - NONE
 - ANY_TIME
 - NOT_NEEDED
 - FROM_SNAPSHOT

If an unsupported value is provided, the value is assumed to be `NONE`.

- **auto_grow_vadp_rep:** Does the volume support auto-grow VADP rep?
- **thin_provisioned:** Is the volume configured as thin provisioned?

Status Codes

Status Code	Description
200 OK	The list of existing volumes was successfully retrieved.
401 Unauthorized	A session ID was not provided.

GET /server/<server_id>/volume/<volume_id>

Description

Retrieves existing volume associated with the supplied `volume_id`, `server_id`, and session ID.

Request

Body

None

Headers

Header	Description
x-storageplugin-sessionid	The session ID associated with the request.

Response

Body

```
{
  "id": "<id>",
  "name": "<name>",
  "creationTime": "<creation_time>",
  "serverGroupKey": "<server_group_key>",
  "poolId": "<pool_id>",
  "volumeType": "<volume_type>",
  "stats": {
    "size": "<size>",
    "usedSize": "<used_size>",
  },
  "supports": {
    "revert": "<revert>",
    "split": "<split>",
    "autoGrowVadpRep": "<auto_grow_vadp_rep>"
  }
}
```

```

    },
    "config": {
        "thinProvisioned": "<thin_provisioned>"
    }
}

```

where:

- **name:** The name of the volume.
- **id:** The ID of the volume.
- **creation_time:** An epoch date representing the creation time of the volume.
- **server_group_key:** The key of the server group to which the volume belongs.
- **pool_id:** The ID of the pool to which the volume belongs.
- **volume_type:** The type of the volume, including vendor-specific classification, if applicable.
- **size:** The size, in bytes, of the volume.
- **used_size:** The number of bytes currently used (filled) by the volume.
- **revert:** Does the volume support revert?
- **split:** Does the volume support splitting? This value must be one of the following:
 - NONE
 - ANY_TIME
 - NOT_NEEDED
 - FROM_SNAPSHOT

If an unsupported value is provided, the value is assumed to be `NONE`.

- **auto_grow_vadp_rep:** Does the volume support auto-grow VADP rep?
- **thin_provisioned:** Is the volume configured as thin provisioned?

Status Codes

Status Code	Description
200 OK	The existing volume was successfully retrieved.
401 Unauthorized	A session ID was not provided.
404 Not Found	The server associated with the supplied <code>volume_id</code> and <code>server_id</code> does not exist.

DELETE `/server/<server_id>/volume/<volume_id>`

Description

Deletes an existing volume.

Request**Body***None***Headers**

Header	Description
x-storageplugin-sessionid	The session ID associated with the request.

Response**Body***None***Status Codes**

Status Code	Description
200 OK	The existing volume was successfully deleted.
404 Not Found	The server associated with the supplied <code>volume_id</code> and <code>server_id</code> does not exist.

GET /server/<server_id>/volume/<volume_id>/splitstatus**Description**

Retrieves the status for the currently in-progress split task.

Request**Body***None***Headers**

Header	Description
x-storageplugin-sessionid	The session ID associated with the request.

Response**Body**

```
{
  "percentageComplete": <percentage_complete>,
  "complete": <is_complete>
}
```

where:

- **percentage_complete:** The completion percentage, between 0 and 100, inclusive, of the split task
- **is_complete:** A boolean flag denoting if the split task is complete or not

Status Codes

Status Code	Description
200 OK	The volume split task was successfully started.
401 Unauthorized	A session ID was not provided.

Actions

POST /server/<server_id>/volume/<volume_id>?action=clone

Description

Clones a volume.

Request

Body

```
{
  "snapshot": {
    "id": "<snapshot_id>",
    "name": "<snapshot_name>"
  },
  "name": "<name>",
  "serverGroupKeys": [<server_group_key>", ...],
  "instantSplit": <is_instant_split>
}
```

where:

- **snapshot_id**: The ID of the snapshot to clone.
- **snapshot_name**: The name of the snapshot to clone.
- **name**: Destination name of the clone.
- **server_group_key**: Server group key associated with the clone.
- **is_instant_split**: Is this clone an instant split?

Headers

Header	Description
x-storageplugin-sessionid	The session ID associated with the request.

Response

Body

```
{
  "id": "<id>",
  "name": "<name>",
  "creationTime": "<creation_time>",
  "serverGroupKey": "<server_group_key>",
  "poolId": "<pool_id>",
  "volumeType": "<volume_type>",
  "stats": {
    "size": "<size>",
```

```

        "usedSize": "<used_size>",
    },
    "supports": {
        "revert": "<revert>",
        "split": "<split>",
        "autoGrowVadpRep": "<auto_grow_vadp_rep>"
    },
    "config": {
        "thinProvisioned": "<thin_provisioned>"
    }
}

```

where:

- **name:** The name of the volume.
- **id:** The ID of the volume.
- **creation_time:** An epoch date representing the creation time of the volume.
- **server_group_key:** The key of the server group to which the volume belongs.
- **pool_id:** The ID of the pool to which the volume belongs.
- **volume_type:** The type of the volume, including vendor-specific classification, if applicable.
- **size:** The size, in bytes, of the volume.
- **used_size:** The number of bytes currently used (filled) by the volume.
- **revert:** Does the volume support revert?
- **split:** Does the volume support splitting? This value must be one of the following:
 - NONE
 - ANY_TIME
 - NOT_NEEDED
 - FROM_SNAPSHOT

If an unsupported value is provided, the value is assumed to be `NONE`.

- **auto_grow_vadp_rep:** Does the volume support auto-grow VADP rep?
- **thin_provisioned:** Is the volume configured as thin provisioned?

Status Codes

Status Code	Description
200 OK	The volume was successfully cloned.
401 Unauthorized	A session ID was not provided.

POST /server/<server_id>/volume/<volume_id>?action=mount

Description

Mounts a volume.

Request

Body

```
{
  "path": "<mount_path>"
}
```

where:

- **mount_path**: The path to which the volume should be mounted, if applicable

Headers

Header	Description
x-storageplugin-sessionid	The session ID associated with the request.

Response

Body

```
{
  "path": "<path>"
}
```

where:

- **path**: The path to which the volume was mounted, if applicable

Status Codes

Status Code	Description
200 OK	The volume was successfully mounted.
401 Unauthorized	A session ID was not provided.

POST /server/<server_id>/volume/<volume_id>?action=split

Description

Splits a volume.

Request

Body

None

Headers

Header	Description
x-storageplugin-sessionid	The session ID associated with the request.

Response

Body

```
{
  "percentageComplete": <percentage_complete>,
}
```

```
"complete": <is_complete>
}
```

where:

- **percentage_complete**: The completion percentage, between 0 and 100, inclusive, of the split task
- **is_complete**: A boolean flag denoting if the split task is complete or not

Status Codes

Status Code	Description
200 OK	The volume split task was successfully started.
401 Unauthorized	A session ID was not provided.

POST /server/<server_id>/volume/<volume_id>?action=restore

Description

Restores a volume from an existing snapshot.

Request

Body

```
{
  "snapshotId": "<snapshot_id>"
}
```

where:

- **snapshot_id**: The ID of the snapshot to restore.

Headers

Header	Description
x-storageplugin-sessionid	The session ID associated with the request.

Response

Body

```
{
  "id": "<id>",
  "name": "<name>",
  "creationTime": "<creation_time>",
  "serverGroupKey": "<server_group_key>",
  "poolId": "<pool_id>",
  "volumeType": "<volume_type>",
  "stats": {
    "size": "<size>",
    "usedSize": "<used_size>",
  },
  "supports": {
    "revert": "<revert>",
    "split": "<split>",
  }
}
```



```

    "autoGrowVadpRep": "<auto_grow_vadp_rep>"
  },
  "config": {
    "thinProvisioned": "<thin_provisioned>"
  }
}

```

where:

- **name:** The name of the volume.
- **id:** The ID of the volume.
- **creation_time:** An epoch date representing the creation time of the volume.
- **server_group_key:** The key of the server group to which the volume belongs.
- **pool_id:** The ID of the pool to which the volume belongs.
- **volume_type:** The type of the volume, including vendor-specific classification, if applicable.
- **size:** The size, in bytes, of the volume.
- **used_size:** The number of bytes currently used (filled) by the volume.
- **revert:** Does the volume support revert?
- **split:** Does the volume support splitting? This value must be one of the following:
 - NONE
 - ANY_TIME
 - NOT_NEEDED
 - FROM_SNAPSHOT

If an unsupported value is provided, the value is assumed to be `NONE`.

- **auto_grow_vadp_rep:** Does the volume support auto-grow VADP rep?
- **thin_provisioned:** Is the volume configured as thin provisioned?

Status Codes

Status Code	Description
200 OK	The volume was successfully restored.
401 Unauthorized	A session ID was not provided.

POST `/server/<server_id>/volume/<volume_id>?action=restore-to-alternate`

Description

Restores a snapshot to a different (alternate) volume.

Request

Body

```

{
  "sourceSnapshotId": "<source_snapshot_id>",

```

```
{
  "targetVolumeId": "<target_volume_id>"
}
```

where:

- **source_snapshot_id**: The ID of the snapshot to restore.
- **target_volume_id**: The ID of the volume to restore the snapshot to.

Headers

Header	Description
x-storageplugin-sessionid	The session ID associated with the request.

Response

Body

```
{
  "id": "<id>",
  "name": "<name>",
  "creationTime": "<creation_time>",
  "serverGroupKey": "<server_group_key>",
  "poolId": "<pool_id>",
  "volumeType": "<volume_type>",
  "stats": {
    "size": "<size>",
    "usedSize": "<used_size>",
  },
  "supports": {
    "revert": "<revert>",
    "split": "<split>",
    "autoGrowVadpRep": "<auto_grow_vadp_rep>"
  },
  "config": {
    "thinProvisioned": "<thin_provisioned>"
  }
}
```

where:

- **name**: The name of the volume.
- **id**: The ID of the volume.
- **creation_time**: An epoch date representing the creation time of the volume.
- **server_group_key**: The key of the server group to which the volume belongs.
- **pool_id**: The ID of the pool to which the volume belongs.
- **volume_type**: The type of the volume, including vendor-specific classification, if applicable.
- **size**: The size, in bytes, of the volume.
- **used_size**: The number of bytes currently used (filled) by the volume.
- **revert**: Does the volume support revert?

- **split:** Does the volume support splitting? This value must be one of the following:
 - NONE
 - ANY_TIME
 - NOT_NEEDED
 - FROM_SNAPSHOT

If an unsupported value is provided, the value is assumed to be `NONE`.

- **auto_grow_vadp_rep:** Does the volume support auto-grow VADP rep?
- **thin_provisioned:** Is the volume configured as thin provisioned?

Status Codes

Status Code	Description
200 OK	The volume was successfully restored.
401 Unauthorized	A session ID was not provided.

POST /server/<server_id>/volume/<volume_id>?action=restore-to-new

Description

Restores a snapshot to a new volume.

Request

Body

```
{
  "sourceSnapshotId": "<source_snapshot_id>",
  "targetVolumeName": "<target_volume_name>",
  "targetPoolIds": ["<target_pool_id>", ...]
}
```

where:

- **source_snapshot_id:** The ID of the snapshot to restore.
- **target_volume_name:** The name of the new volume to restore the snapshot to.
- **target_ool_id:** The ID of a pool associated with the newly created volume.

Headers

Header	Description
x-storageplugin-sessionid	The session ID associated with the request.

Response

Body

```
{
  "id": "<id>",
  "name": "<name>",
  "creationTime": "<creation_time>",
  "serverGroupKey": "<server_group_key>",
}
```

```

"poolId": "<pool_id>",
"volumeType": "<volume_type>",
"stats": {
  "size": "<size>",
  "usedSize": "<used_size>",
},
"supports": {
  "revert": "<revert>",
  "split": "<split>",
  "autoGrowVadpRep": "<auto_grow_vadp_rep>"
},
"config": {
  "thinProvisioned": "<thin_provisioned>"
}
}

```

where:

- **name:** The name of the volume.
- **id:** The ID of the volume.
- **creation_time:** An epoch date representing the creation time of the volume.
- **server_group_key:** The key of the server group to which the volume belongs.
- **pool_id:** The ID of the pool to which the volume belongs.
- **volume_type:** The type of the volume, including vendor-specific classification, if applicable.
- **size:** The size, in bytes, of the volume.
- **used_size:** The number of bytes currently used (filled) by the volume.
- **revert:** Does the volume support revert?
- **split:** Does the volume support splitting? This value must be one of the following:
 - NONE
 - ANY_TIME
 - NOT_NEEDED
 - FROM_SNAPSHOT

If an unsupported value is provided, the value is assumed to be `NONE`.

- **auto_grow_vadp_rep:** Does the volume support auto-grow VADP rep?
- **thin_provisioned:** Is the volume configured as thin provisioned?

Status Codes

Status Code	Description
200 OK	The volume was successfully restored.
401 Unauthorized	A session ID was not provided.

LUN

Storage LUNs provisioned for client hosts. Use the `/lun` endpoint to obtain information about storage LUNs.

Endpoints

- `http://<ip>:<port>/storageplugin/api/server/<server_id>/lun`
- `http://<ip>:<port>/storageplugin/api/server/<server_id>/lun/<lun_id>`

Verbs

GET `/server/<server_id>/lun`

Description

Retrieves all existing Logical Unit Numbers (LUNs) associated with the supplied `server_id` and session ID.

Request

Body

None

Headers

Header	Description
<code>x-storageplugin-sessionid</code>	The session ID associated with the request.

Response

Body

```
{
  "luns": [
    {
      "id": "<id>",
      "name": "<name>",
      "serialNumber": "<serial_number>",
      "naaId": "<naa_id>",
      "volumeId": "<volume_id>",
      "privateIscsiTarget": "<private_iscsi_target>",
      "volumeType": "<volume_type>"
    },
    ...
  ]
}
```

where:

- **id**: The ID of the LUN.
- **name**: The name of the LUN.

- **serial_number**: The serial number of the LUN.
- **naa_id**: The Network Address Authority (NAA) ID of the LUN.
- **volume_id**: The ID of the volume associated with the LUN.
- **private_iscsi_target**: The iSCSI target of the LUN, if applicable.
- **volume_type**: The type of the volume associated with the LUN, if applicable.

Status Codes

Status Code	Description
200 OK	The list of existing LUNs was successfully retrieved.
401 Unauthorized	A session ID was not provided.

GET /server/<server_id>/lun/<lun_id>

Description

Retrieves the Logical Unit Numbers (LUNs) associated with the supplied `lun_id`, `server_id` and session ID.

Request

Body

None

Headers

Header	Description
x-storageplugin-sessionid	The session ID associated with the request.

Response

Body

```
{
  "id": "<id>",
  "name": "<name>",
  "serialNumber": "<serial_number>",
  "naaId": "<naa_id>",
  "volumeId": "<volume_id>",
  "privateIscsiTarget": "<private_iscsi_target>",
  "volumeType": "<volume_type>"
}
```

where:

- **id**: The ID of the LUN.
- **name**: The name of the LUN.
- **serial_number**: The serial number of the LUN.
- **naa_id**: The Network Address Authority (NAA) ID of the LUN.
- **volume_id**: The ID of the volume associated with the LUN.

- **private_iscsi_target:** The iSCSI target of the LUN, if applicable.
- **volume_type:** The type of the volume associated with the LUN, if applicable.

Status Codes

Status Code	Description
200 OK	The existing LUN associated with the supplied <code>lun_id</code> was successfully retrieved.
401 Unauthorized	A session ID was not provided.
404 Not Found	A LUN associated with the supplied <code>lun_id</code> does not exist.

Actions

POST `/server/<server_id>/lun/<lun_id>?action=online`

Description

Turns a LUN online (i.e., makes a LUN ready for online use).

Request

Body

None

Headers

Header	Description
<code>x-storageplugin-sessionid</code>	The session ID associated with the request.

Response

Body

None

Status Codes

Status Code	Description
200 OK	The LUN was successfully online.
401 Unauthorized	A session ID was not provided.

Snapshot

Volume snapshots. Use the `/snapshot` endpoint to obtain information about volume snapshots.

Endpoints

- `http://<ip>:<port>/storageplugin/api/server/<server_id>/volume/<volume_id>/snapshot`
- `http://<ip>:<port>/storageplugin/api/server/<server_id>/volume/<volume_id>/snapshot/<snapshot_id>`

Verbs

GET `/server/<server_id>/volume/<volume_id>/snapshot`

Description

Retrieves the list of snapshot associated with the supplied `volume_id`, `server_id`, and session ID.

Request

Body

None

Headers

Header	Description
<code>x-storageplugin-sessionid</code>	The session ID associated with the request.

Response

Body

```
{
  "snapshots": [
    {
      "id": "<id>",
      "name": "<name>",
      "creationTime": <creation_time>,
      "consistencyGroupSnapshotId": "<consistency_group_snapshot_id>",
      "config": {
        "groundCloneRequired": <ground_clone_required>
      }
    },
    ...
  ]
}
```

where:

- **name:** The name of the snapshot.
- **id:** The ID of the snapshot.
- **creation_time:** An epoch date representing the creation time of the snapshot.
- **consistency_group_snapshot_id:** The ID of the consistency group snapshot mapped to this volume snapshot (if applicable).
- **ground_clone_required:** Is ground clone required?

Status Codes

Status Code	Description
200 OK	The list of existing snapshots was successfully retrieved.
401 Unauthorized	A session ID was not provided.

GET /server/<server_id>/volume/<volume_id>/snapshot/<snapshot_id>

Description

Retrieves existing snapshot associated with the supplied `snapshot_id`, `volume_id`, `server_id`, and session ID.

Request

Body

None

Headers

Header	Description
x-storageplugin-sessionid	The session ID associated with the request.

Response

Body

```
{
  "id": "<id>",
  "name": "<name>",
  "creationTime": <creation_time>,
  "consistencyGroupSnapshotId": "<consistency_group_snapshot_id>",
  "config": {
    "groundCloneRequired": <ground_clone_required>
  }
}
```

where:

- **name:** The name of the snapshot.
- **id:** The ID of the snapshot.
- **creation_time:** An epoch date representing the creation time of the snapshot.
- **consistency_group_snapshot_id:** The ID of the consistency group snapshot mapped to this volume snapshot (if applicable).
- **ground_clone_required:** Is ground clone required?

Status Codes

Status Code	Description
200 OK	The existing snapshot was successfully retrieved.
401	A session ID was not provided.

Unauthorized	
404 Not Found	The server associated with the supplied <code>snapshot_id</code> , <code>volume_id</code> and <code>server_id</code> does not exist.

POST `/server/<server_id>/volume/<volume_id>/snapshot`

Description

Creates a new snapshot associated with the supplied `volume_id`, `server_id`, and session ID.

Request

Body

```
{
  "name": "<name>",
  "poolId": "<poolId>",
  "replicate": <replicate>,
  "isAutoDelete": <isAutoDelete>
}
```

where:

- **name:** The name of the snapshot.
- **poolId:** The ID of the pool with which to associate the snapshot.
- **replicate:** A Boolean flag denoting if the snapshot is a replication snapshot.
- **isAutoDelete:** A Boolean flag denoting if the auto-delete feature is set for the snapshot.

Headers

Header	Description
<code>x-storageplugin-sessionid</code>	The session ID associated with the request.

Response

Body

```
{
  "id": "<id>",
  "name": "<name>",
  "creationTime": <creation_time>,
  "config": {
    "groundCloneRequired": <ground_clone_required>
  }
}
```

where:

- **name:** The name of the snapshot.
- **id:** The ID of the snapshot.
- **creation_time:** An epoch date representing the creation time of the snapshot.

- **ground_clone_required:** Is ground clone required?

Status Codes

Status Code	Description
201 Created	The supplied snapshot was created.
401 Unauthorized	A session ID was not provided.

DELETE /server/<server_id>/volume/<volume_id>/snapshot/<snapshot_id>

Description

Deletes an existing snapshot associated with the supplied `snapshot_id`, `volume_id`, `server_id`, and session ID.

Request

Body

None

Headers

Header	Description
x-storageplugin-sessionid	The session ID associated with the request.

Response

Body

None

Status Codes

Status Code	Description
204 No Content	The existing snapshot was successfully deleted.
401 Unauthorized	A session ID was not provided.
404 Not Found	The desired snapshot could be found.

Host

Storage client hosts. Using the `/host` endpoint, information can be obtained about storage client hosts.

Endpoints

- `http://<ip>:<port>/storageplugin/api/server/<server_id>/host`
- `http://<ip>:<port>/storageplugin/api/server/<server_id>/host/<host_id>`

Verbs

GET /server/<server_id>/host

Description

Retrieves the list of hosts associated with the supplied `server_id` and session ID.

Request

Body

None

Headers

Header	Description
x-storageplugin-sessionid	The session ID associated with the request.

Response

Body

```
{
  "hosts": [
    {
      "id": "<id>",
      "name": "<name>",
      "serverGroupKeys": ["<server_group_key>", ...],
      "mappedLunIds": ["<lun_id>", ...],
      "iqns": ["<iqn>", ...],
      "wwns": ["<wwn>", ...],
      "config": {
        "fc": <fc>,
        "iscsi": <iscsi>
      }
    },
    ...
  ]
}
```

where:

- **name:** The name of the host.
- **id:** The ID of the host.
- **server_group_key:** A server group key associated with the host.
- **lun_id:** The ID of a LUN associated with the host.
- **iqn:** An iSCSI Qualified Name (IQN) associated with the host.
- **wwn:** A World Wide Name (WWN) associated with the host.
- **fc:** Is the host a fiber channel host?
- **iscsi:** Is the host an iSCSI host?

Status Codes

Status Code	Description
200 OK	The list of existing hosts was successfully retrieved.
401 Unauthorized	A session ID was not provided

GET /server/<server_id>/host/<host_id>

Description

Retrieves the host associated with the supplied `host_id`, `server_id` and session ID.

Request

Body

None

Headers

Header	Description
x-storageplugin-sessionid	The session ID associated with the request.

Response

Body

```
{
  "id": "<id>",
  "name": "<name>",
  "serverGroupKeys": ["<server_group_key>", ...],
  "mappedLunIds": ["<lun_id>", ...],
  "iqns": ["<iqn>", ...],
  "wwns": ["<wwn>", ...],
  "config": {
    "fc": <fc>,
    "iscsi": <iscsi>
  }
}
```

where:

- **name**: The name of the host.
- **id**: The ID of the host.
- **server_group_key**: A server group key associated with the host.
- **lun_id**: The ID of a LUN associated with the host.
- **iqn**: An iSCSI Qualified Name (IQN) associated with the host.
- **wwn**: A World Wide Name (WWN) associated with the host.
- **fc**: Is the host a fiber channel host?
- **iscsi**: Is the host an iSCSI host?

Status Codes

Status Code	Description
200 OK	The existing host associated with the supplied <code>host_id</code> was successfully retrieved.
401 Unauthorized	A session ID was not provided.
404 Not Found	A host associated with the supplied <code>host_id</code> does not exist.

POST `/server/<server_id>/host`

Description

Creates a new host.

Request

Body

```
{
  "name": "<name>",
  "iqns": ["<iqn>", ...],
  "wwns": ["<wwn>", ...]
}
```

where:

- **name:** The name of the host.
- **iqn:** An iSCSI Qualified Name (IQN) associated with the host.
- **wwn:** A World Wide Name (WWN) associated with the host.

Headers

Header	Description
<code>x-storageplugin-sessionid</code>	The session ID associated with the request.

Response

Body

```
{
  "id": "<id>",
  "name": "<name>",
  "serverGroupKeys": ["<server_group_key>", ...],
  "mappedLunIds": ["<lun_id>", ...],
  "iqns": ["<iqn>", ...],
  "wwns": ["<wwn>", ...],
  "config": {
    "fc": <fc>,
    "iscsi": <iscsi>
  }
}
```

where:

- **name:** The name of the host.
- **id:** The ID of the host.
- **server_group_key:** A server group key associated with the host.
- **lun_id:** The ID of a LUN associated with the host.
- **iqn:** An iSCSI Qualified Name (IQN) associated with the host.
- **wwn:** A World Wide Name (WWN) associated with the host.
- **fc:** Is the host a fiber channel host?
- **iscsi:** Is the host an iSCSI host?

Status Codes

Status Code	Description
200 OK	The list of existing hosts was successfully retrieved.
401 Unauthorized	A session ID was not provided.

GET /server/<server_id>/host/<host_id>/lun

Description

Retrieves the all of the LUNs associated with a host.

Request

Body

None

Headers

Header	Description
x-storageplugin-sessionid	The session ID associated with the request.

Response

Body

```
{
  "name": "<name>",
  "volumeId": "<volume_id>",
  "hostId": "<host_id>",
  "hostGroupId": "<host_group_id>",
  "lunId": "<lun_id>",
  "lunNumber": <lun_number>
}
```

where:

- **name:** The name of the volume.
- **volume_id:** The ID of the volume.

- **host_id**: The ID of the host.
- **host_group_id**: The ID of the host group, if applicable.
- **lun_id**: The ID of the LUN.
- **lun_number**: The number of the LUN.

Status Codes

Status Code	Description
200 OK	The existing host associated with the supplied <code>host_id</code> was successfully retrieved.
401 Unauthorized	A session ID was not provided.
404 Not Found	A host associated with the supplied <code>host_id</code> does not exist.

Actions

POST /server/<server_id>/host/<host_id>?action=map

Description

Maps a LUN to a host.

Request

Body

```
{
  "volumeId": "<volume_id>",
  "lunId": "<lun_id>",
  "lunNumber": <lun_number>
}
```

where:

- **volume_id**: The ID of the volume to map.
- **lun_id**: The ID of the LUN to map.
- **lun_number**: The number of the LUN to map.

Headers

Header	Description
x-storageplugin-sessionid	The session ID associated with the request.

Response

Body

```
{
  "name": "<name>",
  "volumeId": "<volume_id>",
  "hostId": "<host_id>",
  "hostGroupId": "<host_group_id>",
}
```



```

    "lunId": "<lun_id>",
    "lunNumber": <lun_number>
  }

```

where:

- **name:** The name of the volume.
- **volume_id:** The ID of the volume.
- **host_id:** The ID of the host.
- **host_group_id:** The ID of the host group, if applicable.
- **lun_id:** The ID of the LUN.
- **lun_number:** The number of the LUN to map.

Status Codes

Status Code	Description
200 OK	The volume was successfully mapped.
401 Unauthorized	A session ID was not provided.

POST /server/<server_id>/host/<host_id>?action=unmap

Description

Maps a volume to a host.

Request

Body

```

{
    "lunId": "<lun_id>"
}

```

where:

- **volume_id:** The ID of the volume to unmap.

Headers

Header	Description
x-storageplugin-sessionid	The session ID associated with the request.

Response

Body

None

Status Codes

Status Code	Description
200 OK	The volume was successfully unmapped.
401 Unauthorized	A session ID was not provided.

Host Group

Storage client hosts. Using the `/host` endpoint, information can be obtained about storage client hosts.

Endpoints

- `http://<ip>:<port>/storageplugin/api/server/<server_id>/hostgroup`

Verbs

GET `/server/<server_id>/hostgroup`

Description

Retrieves the list of all host groups.

Request

Body

None

Headers

Header	Description
<code>x-storageplugin-sessionid</code>	The session ID associated with the request.

Response

Body

```
{
  "groups": [
    {
      "id": "<id>",
      "name": "<name>",
      "hostIds": ["<host_id>", ...]
    },
    ...
  ]
}
```

where:

- **id**: The ID of the host group.
- **name**: The name of the host group.
- **host_id**: The ID of a host associated with the host group.

Status Codes

Status Code	Description
-------------	-------------

200 OK	The list of existing host groups was successfully retrieved.
401 Unauthorized	A session ID was not provided

Actions

POST /server/<server_id>/hostgroup/<host_group_id>?action=map

Description

Maps a LUN to a host group.

Request

Body

```
{
  "volumeId": "<volume_id>",
  "lunId": "<lun_id>",
  "lunNumber": <lun_number>
}
```

where:

- **volume_id**: The ID of the volume to map.
- **lun_id**: The ID of the LUN to map.
- **lun_number**: The number of the LUN to map.

Headers

Header	Description
x-storageplugin-sessionid	The session ID associated with the request.

Response

Body

```
{
  "name": "<name>",
  "volumeId": "<volume_id>",
  "lunId": "<lun_id>",
  "lunNumber": <lun_number>
}
```

where:

- **name**: The name of the volume.
- **volume_id**: The ID of the volume.
- **lun_id**: The ID of the LUN.
- **lun_number**: The number of the LUN to map.

Status Codes

Status Code	Description
-------------	-------------

200 OK	The volume was successfully mapped.
401 Unauthorized	A session ID was not provided.

Network

Storage network interfaces. Used to retrieve network interfaces for FC, iSCSI, NFS and CIFS.

Endpoints

- `http://<ip>:<port>/storageplugin/api/server/<server_id>/network`
- `http://<ip>:<port>/storageplugin/api/server/<server_id>/network/<network_id>`

Verbs

GET `/server/<server_id>/network`

Description

Retrieves all existing networks associated with the supplied `server_id` and session ID.

Request

Body

None

Headers

Header	Description
<code>x-storageplugin-sessionid</code>	The session ID associated with the request.

Response

Body

```
{
  "networks": [
    {
      "id": "<id>",
      "name": "<name>",
      "ipAddress": "<ip_address>",
      "wwpn": "<wwpn>",
      "iqn": "<iqn>",
      "serverGroupKey": "<server_group_key>",
      "config": {
        "nfs": <nfs>,
        "cifs": <cifs>,
        "iscsi": <iscsi>,
        "fc": <fc>
      },
      ...
    },
    ...
  ]
}
```

```
} ]
```

where:

- **id**: The ID of the network.
- **name**: The name of the network.
- **ip_address**: The IP address of the network.
- **wwpn**: The World Wide Port Number (WWPN) associated with the network.
- **iqn**: The iSCSI Qualified Name (IQN) associated with the network.
- **server_group_key**: The server group key associated with the network.
- **nfs**: Is the network a Network File System (NFS) network?
- **cifs**: Is the network a Common Internet File System (CIFS) network?
- **iscsi**: Is the network an Internet Small Computer Systems Interface (iSCSI) network?
- **fc**: Is the network a Fiber Channel (FC) network?

Status Codes

Status Code	Description
200 OK	The list of existing networks was successfully retrieved.
401 Unauthorized	A session ID was not provided

GET /server/<server_id>/network/<network_id>

Description

Retrieves the network associated with the supplied `network_id`, `server_id` and session ID.

Request

Body

None

Headers

Header	Description
x-storageplugin-sessionid	The session ID associated with the request.

Response

Body

```
{
  "id": "<id>",
  "name": "<name>",
  "ipAddress": "<ip_address>",
  "wwpn": "<wwpn>",
  "iqn": "<iqn>",
  "serverGroupKey": "<server_group_key>",
  "config": {
```

```

    "nfs": <nfs>,
    "cifs": <cifs>,
    "iscsi": <iscsi>,
    "fc": <fc>
  }
}

```

where:

- **id**: The ID of the network.
- **name**: The name of the network.
- **ip_address**: The IP address of the network.
- **wwpn**: The World Wide Port Number (WWPN) associated with the network.
- **iqn**: The iSCSI Qualified Name (IQN) associated with the network.
- **server_group_key**: The server group key associated with the network.
- **nfs**: Is the network a Network File System (NFS) network?
- **cifs**: Is the network a Common Internet File System (CIFS) network?
- **iscsi**: Is the network an Internet Small Computer Systems Interface (iSCSI) network?
- **fc**: Is the network a Fiber Channel (FC) network?

Status Codes

Status Code	Description
200 OK	The existing network associated with the supplied <code>network_id</code> was successfully retrieved.
401 Unauthorized	A session ID was not provided.
404 Not Found	A network associated with the supplied <code>network_id</code> does not exist.

Pool

Storage pools.

Endpoints

- `http://<ip>:<port>/storageplugin/api/server/<server_id>/pool`
- `http://<ip>:<port>/storageplugin/api/server/<server_id>/pool/<pool_id>`

Verbs

GET /server/<server_id>/pool

Description

Retrieves all existing pools associated with the supplied `server_id` and session ID.

Request

Body

None

Headers

Header	Description
x-storageplugin-sessionid	The session ID associated with the request.

Response

Body

```
{
  "pools": [
    {
      "id": "<id>",
      "name": "<name>",
      "freeSpace": <free_space>
    },
    ...
  ]
}
```

where:

- **id**: The ID of the pool.
- **name**: The name of the pool.
- **free_space**: The free space left in the pool in bytes.

Status Codes

Status Code	Description
200 OK	The list of existing pools was successfully retrieved.
401 Unauthorized	A session ID was not provided

GET /server/<server_id>/pool/<pool_id>

Description

Retrieves the pool associated with the supplied `pool_id`, `server_id` and session ID.

Request

Body

None

Headers

Header	Description
x-storageplugin-sessionid	The session ID associated with the request.

Response

Body

```
{
  "id": "<id>",
  "name": "<name>",
  "freeSpace": <free_space>
}
```

where:

- **id:** The ID of the pool.
- **name:** The name of the pool.
- **free_space:** The free space left in the pool in bytes.

Status Codes

Status Code	Description
200 OK	The existing pool associated with the supplied <code>pool_id</code> was successfully retrieved.
401 Unauthorized	A session ID was not provided.
404 Not Found	A pool associated with the supplied <code>pool_id</code> does not exist.

Consistency Group

Consistency volume groups.

Endpoints

- `http://<ip>:<port>/storageplugin/api/server/<server_id>/consistencygroup`
- `http://<ip>:<port>/storageplugin/api/server/<server_id>/consistencygroup/<consistency_group_id>`

Verbs

GET /server/<server_id>/consistencygroup

Description

Retrieves all existing consistency groups associated with the supplied `server_id` and session ID.

Request

Body

None

Headers

Header	Description
x-storageplugin-sessionid	The session ID associated with the request.

Response

Body

```
{
  "consistencyGroups": [
    {
      "id": "<id>",
      "name": "<name>",
      "creationTime": <creation_time>,
      "volumeIds": [<volume_id>, ...],
      "config": {
        "incremental": <incremental>,
        "replicationTarget": <replication_target>
      },
      ...
    },
    ...
  ]
}
```

where:

- **id**: The ID of the consistency group.
- **name**: The name of the consistency group.
- **creation_time**: The epoch time when the consistency group was created.
- **volume_id**: The ID of the volume associated with the consistency group.
- **incremental**: A Boolean flag denoting if the consistency group is incremental.
- **replication_target**: A Boolean flag denoting if the consistency group is a replication target.

Status Codes

Status Code	Description
200 OK	The list of existing networks was successfully retrieved.
401 Unauthorized	A session ID was not provided.

GET /server/<server_id>/consistencygroup/<consistency_group_id>

Description

Retrieve the consistency group associated with the supplied consistency_group_id, server_id and session ID.

Request

Body

None

Headers

Header	Description
x-storageplugin-sessionid	The session ID associated with the request.

Response

Body

```
{
  "id": "<id>",
  "name": "<name>",
  "creationTime": <creation_time>,
  "volumeIds": [<volume_id>, ...],
  "config": {
    "incremental": <incremental>,
    "replicationTarget": <replication_target>
  }
}
```

where:

- **id**: The ID of the consistency group.
- **name**: The name of the consistency group.
- **creation_time**: The epoch time when the consistency group was created.
- **volume_id**: The ID of the volume associated with the consistency group.
- **incremental**: A Boolean flag denoting if the consistency group is incremental.
- **replication_target**: A Boolean flag denoting if the consistency group is a replication target.

Status Codes

Status Code	Description
200 OK	The list of existing networks was successfully retrieved.
401 Unauthorized	A session ID was not provided.
404 Not Found	The requested consistency group could not be found.

Actions

POST /server/<server_id>/consistencygroup/<consistencygroup_id>?action=add-volume

Description

Adds a volume to a consistency group.

Request

Body

```
{
  "snapshotName": "<snapshot_name>",
  "volumeId": "<volume_id>",
  "poolId": "<pool_id>",
  "isIncremental": <is_incremental>
}
```

where:

- **name:** The name of the snapshot.
- **volume_id:** The ID of the volume to add.
- **pool_id:** The ID of the pool with which to associate the volume.
- **is_incremental:** A Boolean flag denoting if the added volume is incremental.

Headers

Header	Description
x-storageplugin-sessionid	The session ID associated with the request.

Response

Body

None

Status Codes

Status Code	Description
200 OK	The volume was successfully added to the consistency group.
401 Unauthorized	A session ID was not provided.

POST

/server/<server_id>/consistencygroup/<consistencygroup_id>?action=remove-volume

Description

Deletes a set of volumes from a consistency group.

Request

Body

```
{
  "volumeIds": ["<volume_id>", ...]
}
```

where:

- **volume_id**: The ID of the volume to delete from the consistency group.

Headers

Header	Description
x-storageplugin-sessionid	The session ID associated with the request.

Response

Body

None

Status Codes

Status Code	Description
200 OK	The volume was successfully deleted from the consistency group.
401 Unauthorized	A session ID was not provided.

POST

/server/<server_id>/consistencygroup/<consistencygroup_id>?action=condense

Description

Condenses a consistency group.

Request

Body

```
{
  "ageInDays": <age_in_days>,
  "numberOfInstances": <number_of_instances>,
  "snapshotPrefixes": [<snapshot_prefix>, ...]
}
```

where:

- **age_in_days**: The age, in days, to condense.
- **number_of_instances**: The number of instances of condense.
- **snapshot_prefix**: The snapshot prefix used while condensing.

Headers

Header	Description
--------	-------------

x-storageplugin-sessionid	The session ID associated with the request.
---------------------------	---

Response

Body

None

Status Codes

Status Code	Description
200 OK	The condense successfully completed.
401 Unauthorized	A session ID was not provided.

POST /server/<server_id>/consistencygroup/<consistencygroup_id>?action=take-snapshot

Description

Takes a snapshot of the consistency group.

Request

Body

```
{
  "prefix": "<prefix>",
  "replicate": <replicate>,
  "isAutoDelete": <is_auto_delete>,
  "isIncremental": <is_incremental>
}
```

where:

- **prefix:** The prefix to use for the snapshot.
- **replicate:** A Boolean flag denoting if the snapshot is a replication.
- **is_auto_delete:** A Boolean flag denoting if the auto-delete feature is enabled for the snapshot.
- **is_incremental:** A Boolean flag denoting if the snapshot is incremental.

Headers

Header	Description
x-storageplugin-sessionid	The session ID associated with the request.

Response

Body

```
{
  "id": "<id>",
  "name": "<name>"
}
```

where:

- **id:** The ID of the created snapshot.
- **name:** The name of the created snapshot.

Status Codes

Status Code	Description
200 OK	The snapshot was successfully taken.
401 Unauthorized	A session ID was not provided.

Consistency Group Snapshots

Storage consistency group snapshots.

Endpoints

- `http://<ip>:<port>/storageplugin/api/server/<server_id>/consistencygroup/<consistency_group_id>/snapshot`

Verbs

GET `/server/<server_id>/consistencygroup/<consistency_group_id>/snapshot`

Description

Retrieves all existing snapshots associated with the supplied `consistency_group_id`, `server_id` and session ID.

Request

Body

None

Headers

Header	Description
<code>x-storageplugin-sessionid</code>	The session ID associated with the request.

Response

Body

```
{
  "snapshots": [
    {
      "id": "<id>",
      "name": "<name>"
    },
    ...
  ]
}
```

where:

- **id:** The ID of the network.
- **name:** The name of the network.
- **volume_snapshot_id:** The ID of the associated volume snapshot.

Status Codes

Status Code	Description
200 OK	The list of existing snapshots was successfully retrieved.
401 Unauthorized	A session ID was not provided