IBM Spectrum Copy Data Management™ 2.2.12.0

Storage Plugin REST API

ibm2212011/6/2020spradg

Table of Contents

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 `application/json`.
`Accept`	Required with the value set to `application/json`.

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 `Location` 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 `session` endpoint.
`Content-Type`	The content type of the body associated with a request. Currently, only `application/json` is supported.
`Accept`	The expected content type of the body associated with a response. Currently, only `application/json` is supported.

Resources

API endpoints are supported for the following resources:

- "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
`x-storageplugin-sessionid`	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
`x-storageplugin-sessionid`	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 `volume_id` and `server_id` 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 `volume_id` and `server_id` 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
`x-storageplugin-sessionid`	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 `lun_id` was successfully retrieved.
`401 Unauthorized`	A session ID was not provided.
`404 Not Found`	A LUN associated with the supplied `lun_id` 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
`x-storageplugin-sessionid`	The session ID associated with the request.

Response

Body

None

Status Codes

Status Code	Description
`200 OK`	The LUN was successfully onlined.
`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
`x-storageplugin-sessionid`	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 `snapshot_id`, `volume_id` and `server_id` 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
`x-storageplugin-sessionid`	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 `host_id` was successfully retrieved.
`401 Unauthorized`	A session ID was not provided.
`404 Not Found`	A host associated with the supplied `host_id` 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
`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 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 `host_id` was successfully retrieved.
`401 Unauthorized`	A session ID was not provided.
`404 Not Found`	A host associated with the supplied `host_id` 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
`x-storageplugin-sessionid`	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
`x-storageplugin-sessionid`	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 `network_id` was successfully retrieved.
`401 Unauthorized`	A session ID was not provided.
`404 Not Found`	A network associated with the supplied `network_id` 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 `pool_id` was successfully retrieved.
`401 Unauthorized`	A session ID was not provided.
`404 Not Found`	A pool associated with the supplied `pool_id` 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
`x-storageplugin-sessionid`	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