IBM DS8900F/DS8880
4.0

RESTful API Guide

SC27-9823-00

Note

Before using this information and the product it supports, read the information in "Notices" on page 33.

This edition applies to version 4, release 0 of the *RESTful API Guide* and to all subsequent releases and modifications until otherwise indicated in new editions.

The material in this document is an excerpt from the online IBM DS8900/DS8880 Documentation and is provided for convenience. This document should be used along with online IBM Documentation.

This edition replaces version 3, release 3, modification 1.

Contents

About this guide

This document includes specifications for RESTful API operations.

Operation types

The specifications for two types of RESTful API operations are covered in this guide:

Queries
Requests for properties information from the specified resource of a storage system.

Updates
Requests that create, delete, or update a resource on a specified storage system.

Storage systems

All of the RESTful API operations specified in this guide are supported on IBM DS8900F/DS8880 and, with some exceptions, DS8870.

Operations not supported on the DS8870 relate to ODD files, performance files, statesave files, and PE packages.

Summary of changes

IBM RESTful API version 4 introduces the following new features. Changed and new information for IBM RESTful API version 4 is indicated by a vertical bar (|) to the left of the change.

This table provides the technical changes and enhancements to the IBM RESTful API as of March 2021.

Function	Description
Removed HMC Port number	Port {hmc} replaces {hmc}:443 in URL definitions.
New note regarding the display of capacity values	Note: In this guide, capacity values "cap", "capalloc", "capavail", and "capraw" denoted with "# `bytes (FB)`" are displayed as a number of bytes for fixed block volumes (FB). Capacity values "cap_gb" and "cap_gib" will be in gigabytes and gibibytes. See note on page 6 for more information.
Removed Repositories (ESE/TSE)	Storage resource repositories like TSE are no longer supported and have been replaced with pools and volumes. See Table 1 on page 7 on Page 7 for more information.
Updated CKD capacity units for Pools and Volumes	Capacity units in CKD for pools and volumes updated to cylinders for Pools and cylinders, gigabytes, and gibibytes for Volumes. See Table 1 on page 7 on Page 7 for more information.

IBM RESTful API version 3, release 3, modification 2 introduced the following new features.
This table provides the technical changes and enhancements to the IBM RESTful API as of May 20, 2019.

Function	Description
Updated HMC port number	Port {hmc}:443 replaces {hmc}:8452 due to forthcoming port sunset
New GET for performance summary, etdata, and system data.	Queries files to determine performance (supported for DS8880, only)
New GET for managed arrays	Queries managed arrays
New GET for users	Queries system users
New GET for I/O enclosures	Queries I/O enclosures
New GET for encryption groups	Queries encryption groups
New POST | PUT | DELETE for logical subsystems	Creates, updates, and deletes logical subsystems
Updated GET | POST for FlashCopy	Queries and creates FlashCopy relationships
Updated GET for host ports	Queries host ports
Denoted measurement unit (bytes) for FB capacity values	Added #`bytes(FB)` to "cap", "capalloc", "capavail", and "capraw" values. (See Note on page 6.)
Added Note on page 6	Note: In this guide, capacity values "cap", "capalloc", "capavail", and "capraw" denoted with "# `bytes (FB)`" are displayed as a number of bytes for fixed block volumes (FB). However, for count key data (CKD) volume queries, capacity values can display as a number of bytes or cylinders.

Chapter 1. RESTful API

The RESTful API is an application on your storage system HMC for initiating simple storage operations through the Web.

The RESTful (Representational State Transfer) API is a platform-independent means by which to initiate create, read, update, and delete operations in the storage system and supporting storage devices. These operations are initiated with the HTTP commands: **POST**, **GET**, **PUT**, and **DELETE**.

The RESTful API is intended for development, testing, and debugging of the client management infrastructures in your storage system. You can use the RESTful API with a CURL command or through standard Web browsers. For instance, you can use the storage system with the RESTClient add-on.

Overview

DS8000 RESTful API supports create, read, update, and delete operations.

RESTful API uses the following operational methods:

HTTP POST
Mapped to create and update operations.

HTTP GET
Mapped to query (read) operations.

HTTP PUT
Mapped to update operations.

HTTP DELETE
Mapped to delete operations.

Resource types

- Single resource – A volume, a port, or a host.
- List of resources – All pools on a specific DS8000 system.

URL structure

The URL structure for an operation is `https://`*`{ip_address}`*`:`*`{port_number}`*`/api/v1/`*`{storage_object}`*`/...` where *{ip_address}* is the IP address of the DS8000 storage system HMC, *{port_number}* is the 3-4 digit port number of the HMC, and *{storage_object}* is the storage object that is targeted for the operation.

(Optional) : You can omit the port number and its preceding colon whereby the number will be inferred and reflected in response data. For instance, `https://`*`{ip_address}`*`/api/v1/`*`{storage_object}`*`/`.

Note : The requests and responses format is JSON. All of the examples in this document are formatted as JSON (JavaScript Object Notation). All POST and PUT requests must have the following HTTP header: `Content-Type:application/json`.

Data types

Boolean
Boolean attributes are provided as `True` or `False`. For example, mirrored attributes of volume resources.

Timestamp
Time stamps are displayed in milliseconds, since the UNIX epoch (1970-01-01 00:00:00).

Security

RESTful API is dependent upon Hypertext Transfer Protocol (HTTP) for security. Token creation is required for authentication.

The following HTTP features are used for RESTful API security:

HTTP or HTTPS
Provides an encrypted HTTP channel between the client and the server.

HTTP basic authentication
Carries client provided credentials to the server.

Creating a token

You must create (**POST**) a token before other RESTful API operations can be initiated.

About this task

The successful completion of this task fulfills all security requirements for the use of the RESTful API. The created token is used in the URL parameter for all subsequent operations that are initiated with the RESTful API.

Note : After the token is created, all subsequent requests must include the following in the header: `X-Auth-Token:`*`{token_name}`*

Procedure

1. Open the Firefox browser.
2. Open the RESTful client by selecting **Tools** > **RESTClient**.
3. In the **Method** field, select POST.
4. In the **URL** field, type `https://`*`{hmc_address}`*`:`*`{port_number}`*`/api/v1/tokens` where *{hmc_address}* is the IP address of the DS8000 storage system HMC, and *{port_number}* is a four-digit port number.
5. Create a custom header.
   a) Select **Headers** > **Custom Header**.
   b) In the **Name** field, enter `Content-Type`.
   c) In the **Value** field, enter `application/json`.
   d) Click **Okay**.
6. In the **Body** field, type `{ "request" : { "params" : { "username" : "admin" , "password" :` *`{user_password}`* `} } }` where *{user_password}* specifies the password.
7. Click **Send**.

Results

If successful, the **Response** window displays the results such as the following example (shown in a condensed format):

```
{ "server" : { "status" : "ok" , "code" : "200" , "message" : "Operation done
successfully." } , "token" : { "token" : "ddb1743a" , "expired_time" :
"2014-08-25T03:28:15-0700" , "max_idle_interval" : "1800000" } }
```

Chapter 2. Queries

The RESTful API can query one or more system resources.

You can run a query by submitting the **HTTP GET** command on a URL. The following examples show the format for single and multiple resource queries:

A single resource

This query requests the properties of the volume `vol2`.

```
/volumes/vol2
```

A list of resources that belong to another resource

This query requests all of the volumes that belong to the pool `mypool`.

```
/pools/mypool/volumes
```

Refer to the following information to query the various system resources.

Encryption groups

Display detailed properties of encryption groups

```
GET /v1/encryption_groups
GET /v1/encryption_groups/{id}
```

Events

Display a list of event records of a specified severity level and time period

```
GET /v1/events[?severity=info,warning,error&after=yyyy-MM-
ddTHH:mm:ssZ&before=yyyy-MM-ddTHH:mm:ssZ]
```

Files (DS8880, only)

Download performance summary, etdata, and system data

```
GET v1/files/performancesummary
GET v1/files/etdata
GET v1/files/sysdata
```

FlashCopy relationships

Display a list of FlashCopy relationships and the status information of each one

```
GET /v1/flashcopy Deprecated
GET /v1/volumes/{volume_id}/flashcopy Deprecated
GET /v1/cs/flashcopies?volume_id_from={volume_id}&volume_id_to={volume_id}
GET /v1/cs/flashcopies/{flashcopy_id}
```

Hosts

Display detailed properties of hosts

```
GET /v1/hosts
GET /v1/hosts/{host_name}
```

Host mappings

Display detailed properties of host mappings

```
GET /v1/hosts/{host_name}/mappings
GET /v1/hosts/{host_name}/mappings/{mapping_id}
```

Host ports

Display detailed properties of host ports

```
GET /v1/host_ports[?wwpns={wwpn},{wwpn},…{wwpn}&state="logged in,logged
out,unconfigured"]
```

```
GET /v1/host_ports/{wwpn}
```

Display detailed properties of the host ports under a specified host

```
GET /v1/hosts/{host_name}/host_ports
```

I/O enclosures

Display detailed properties of all I/O enclosures or a specified I/O enclosure

```
GET /v1/io_enclosures
```

```
GET /v1/io_enclosures/{id}
```

I/O ports

Display detailed properties of all I/O ports or a specified I/O port

```
GET /v1/ioports
```

```
GET /v1/ioports/{port_id}
```

Display detailed properties of host I/O ports

```
GET /v1/hosts/{host_name}/ioports
```

LSSs

Display a list of logical subsystems (LSSs) and status information

```
GET /v1/lss[?type=fb|ckd]
```

```
GET /v1/lss/{lss_id}
```

Managed arrays

Display detailed properties of managed arrays

```
GET /v1/marrays
```

```
GET /v1/marrays/{marray_id}
```

```
GET /v1/pools/{pool_id}/marrays
```

Nodes

Display detailed properties of all nodes or a specified node

```
GET /v1/nodes
```

```
GET /v1/nodes/{node_id}
```

ODD files (`DS8880, only`)

Display all ODD dump files that have been generated

The files are listed from latest to oldest. The query parameters **`offset`** and **`limit`** are used to extract a part of the results.

```
GET /v1/odd
```

Display latest ODD dump files offloading status

Note : If dump files have not been offloaded previously, the status is `failed`.

```
GET /v1/odd/offload
```

PE package (DS8880, only)

Display status of offloading PE package according to area

Note : If PE Package in some areas is never generated and offloaded before, the status is `offload_failed`.

```
GET /v1/pepackage/offload?area=[general | ui | ettraces | etlogs | panic |
prelongedIO]
```

Pools

Display detailed properties of all pools or a specified pool

```
GET /v1/pools
```

```
GET /v1/pools/{pool_id}
```

Remote Mirror and Copy

Display a list of Remote Mirror and Copy (formerly PPRC) volume relationships with status information

```
GET /v1/volumes/{volume_id}/pprc
```
Deprecated

```
GET /v1/pprc
```
Deprecated

Note : Use the following queries in place of the deprecated ones above.

Display a list of Remote Mirror and Copy (formerly PPRC) volume relationships for a storage image

```
GET /v1/cs/pprcs?volume_id_from={volume_id}&volume_id_to={volume_id}
```

```
GET /v1/cs/pprcs/{pprc_id}
```

Display a list of ESCON or Fibre Channel I/O ports that can be defined as Remote Mirror and Copy (formerly PPRC) paths

```
GET /v1/cs/physical_links?
source_lss_id={source_lss_id}&target_lss_id={target_lss_id}&target_system_ww
nn={target_system_wwnn}
```

Display a list of existing Remote Mirror and Copy path definitions

```
GET /v1/cs/pprcs/paths?
source_lss_id_from={source_lss_id}&source_lss_id_to={source_lss_id}
```

```
GET /v1/cs/pprcs/paths/{path_id}
```

Self upgrades

Upload the self upgrade pack or display the results (including time, status, and message) of a self upgrade

```
GET /v1/self_upgrade
```

Space-efficient storage

Display detailed properties report of the specified repository of a pool

```
GET /v1/pools/{pool_id}/tserep
```

```
GET /v1/pools/{pool_id}/eserep
```

Statesave files (DS8880, only)

Display all generated statesave dump files

The files are listed from latest to oldest. Use the query parameters **`"offset"`** and **`"limit"`** to extract part of the results.

```
GET /v1/statesave
```

Display status latest offloaded statesave dump files

Note: If dump files were not offloaded, the status is "`failed`".

```
GET /v1/statesave/offload
```

Systems

Display detailed properties of the storage systems

```
GET /v1/systems
```

Display detailed performance data

```
GET /v1/systems/{sn}/performance[?
after=2014-04-20T13:00:00+0800&before=2014-04-20T13:00:59+0800]
```

Users

Display detailed properties of user accounts

```
GET /v1/users
```

```
GET /v1/users/{user_name}
```

Volumes

Display detailed properties for an individual volume

```
GET /v1/volumes/{volume_id}
```

Display detailed properties for individual volumes under a logical configuration

```
GET /v1/lss/{lss_id}/volumes
```

Display detailed properties for individual volumes under specified host

```
GET /v1/hosts/{host_name}/volumes
```

Display detailed properties for individual volumes in specified pool

```
GET /v1/pools/{pool_id}/volumes
```

Query responses

RESTful API returns a response to a query with a structure that varies based on the system resource and the status of the operation.

RESTful API returns responses that are specific to the requested query.

GET response structure

The successful response displays data pertaining to the properties of the specified resource. If the query is unsuccessful, server information is displayed.

`counts`

Displays the response count and total count of objects

`data_count`

Holds the number of objects in this response data.

`total_count`

Holds the number of total objects.

`data`

Displays the queried resource (for example, "volumes") followed by its properties.

Note : See Table 1 on page 7 for the capacity units that pertain to the values of "cap", "capalloc", "capavail", and "capraw" by storage resource type. "cap_gb" and "cap_gib" will be in gigabytes and gibibytes.

Table 1. Capacity units for storage resource queries.

Storage resource	**Fixed block (FB)**	**Count key data (CKD)**
	Capacity unit	
Pools	bytes	cylinders
Volumes	bytes, gigabytes, and gibibytes	cylinders, gigabytes, and gibibytes
System	bytes	bytes

`server`
Displays the status of the requested query, a failure code, and a corresponding error message.

Example: Successful response to a volume query

```
GET /v1/volumes/0000
"counts":
{
 "data_counts": "",
 "total_counts": ""
},
"data"
{
 "volumes":
  [
   {
    "link" : { "rel" : "self", "href" : "https://{hmc}/api/v1/volumes/0000" },
    "id" : "0000",
    "name" : "mytest",
    "state" : "pinned" | "readonly" | "inaccessible" | "indeterminate_data_loss" |
    "rank_failed" | "rank_repairing" | "rank_repaired" | "global_inaccessible" |
    "global_lost" | "virtual_space_fault" | "nvs_inaccessible" | "normal" |
    "configuring" | "reconfiguring" | "migrating" | "deconfiguring" | "config_error" |
    "reconfig_error" | "migration_error" | "deconfig_error" | "merging" |
    "migration_paused" | "migration_cancelled" | "transposition_error" | "unknown"
    "cap" : "322122547200",                         # bytes (FB)
     "cap_gb": "300",
     "cap_gib": "322.1",
     "real_cap": "322122547200",
     "virtual_cap": "0";
    "stgtype" : "fb" | "ckd",
    "VOLSER" : "",
    "lss" : { "id" : "00",  "link" : { "rel" : "self", "href" : "https://{hmc}/api/v1/
lss/00" } },
    "allocmethod" : "legacy" | "rotatevols" | "rotateexts" | "managed" | "",
    "tp" : "none",
    "capalloc" : "134217728",                         # bytes (FB)
    "MTM" : "2107-900",
    "datatype" : "FB 512" | "FB 512T" | "FB 520P" | "FB 520U" | "FB 520PV" |
    "FB 520UV" | "3380" | "3390" | "3390-A" | "",
    "easytier" : "managed" | "none" | "unknown",
    "tieralloc" :
    [
    { "tier" : "ENT" , "allocated" : "34502" }
    ],
    "pool" : { "id" : "P2", "link" : { "rel" : "self", "href" : "https://{hmc}/api/v1/
pools/P2" }}
  }
 ]
}
```

Example: Unsuccessful response to an invalid query

The request `GET /myvolumes` returns the following response because the query is not valid.

```
GET /myvolumes

 {
 "server" :
  {
   "status" : "failed",
   "code" : "BE7A001D",
   "message" : "Failed to get bundle version."
  }
 }
```

Paging support

Paging restricts the number of records returned by the server. For instance, a volume query request can return thousands of volume records. To enhance your control of received data, two parameters are available.

`offset`

This parameter sets where the list of resources starts, according to the index number of the resource.

- If no offset is defined, the response contains the entire list of resources.
- If an offset is greater than the total resources, the response is empty.
- The index number of the first resource is `0`.

`limit`

This parameter sets the number of resources that the response returns.

- If no limit is defined, the response contains the entire list of resources.
- If a limit is less than the total resources, the list is limited by the limit value; otherwise, all resources are listed.

The returned list of resources is sorted by name. When used, the response is limited by the value of `limit`.

Querying for all volumes of a specified pool

`/v1/pools/P0/volumes`

Querying for the first 400 volumes of all systems

`/v1/pools/P0/volumes?limit=400`

Querying for 40 volumes, starting with the 11th volume

`/v1/pools/P0/volumes?offset=10&limit=40`

Querying for all volumes, starting with the 102nd volume

`/v1/pools/P0/volumes?offset=101`

Querying events

RESTful API queries for an event a little differently than for all other resources.

The RESTful API events query mechanism supports paging. In addition, the response can be filtered with the following parameters:

`severity`

Filters the response by the severity of the events. Available values are, by order of severity: `info` (informational), `warning`, and `error`.

If this parameter is not in use, the events are not filtered.

`after`

Returns events that took place after a specified timestamp.

`before`

Returns events that took place before a specified timestamp.

The timestamp syntax is `yyyy-MM-ddTHH:mm:ssZ`:

yyyy - year

4 digits.

MM - month

`01` to `12`.

dd - day
01 to 31.

T - time
Indicates that the time follows.

HH - hours
00 to 23. The default is 00.

mm - minutes
00 to 59. The default is 00.

ss - seconds
00 to 59. The default is 00.

Z - Zulu, or Coordinated Universal Timc (UTC) offset
-1200 to +1400.

Note : Since the plus symbol is a reserved keyword for HTTP URL, clients must replace + with %2B explicitly if the browser or other client tools cannot translate it automatically.

If the query is not filtered, the response is unrestricted.

Query

```
GET /v1/events[?
severity=info,warning,error&after=2014-04-20T13:00:00+0800&before=2014-04-20
T13:00:59+0800]
```

Response

```
“data”
{
 “events”:
  [
   {
    “id” : “SE...”
    “type” : “PoolCapacityChanged” ,
    “severity” : ”error”,
    “time” : “2014-04-20T13:00:23-0700”,
    “resource_id” : “”,
    “formatted_parameter” : [ ]
   }
  ]
}
```

Chapter 3. Requests

You can create, edit, and delete system resources through RESTful API requests.

All RESTful API requests have the following fields:

Action
This field determines the type of the requested action. Available values are POST (create), PUT (update), and DELETE.

Params
This field specifies the parameters that are required for the action (for example, the size of a volume that the system is required to create).

The general structure of a request, for example, is POST /v1/volumes. The following two examples show the requests for the deletion and updating of a specified volume.

Deletion
Deleting a volume named 0000:

```
DELETE /v1/volumes/0000
```

Update
Updating a volume named 0000:

```
PUT /v1/volumes/0000
```

Note : In this release, only a single update request in the body is supported.

The following information provides the content for requests on various system objects and includes their parameter options.

FlashCopy

Create

```
POST /v1/cs/flashcopies
 {
 "request":
  {
  "params" :
   {
   "volume_pairs" :
    [
     {
     "source_volume" : "0000",
     "target_volume" : "1000"
     },
        ...
    ],
     "options" :
     [
     "freeze_consistency",
     "inhibit_target_writes",
     "record_changes",
     "no_background_copy",
     "persistent",
     "allow_pprc_target_primary",
     "reverse_restore",
     "fast_reverse_restore",
     "permit_space_efficient_target",
     "fail_space_efficient_target_out_of_space"
     ]
    }
   }
  }
```

Delete

```
DELETE /v1/cs/flashcopies?ids={flashcopy_id},{flashcopy_id},...,{flashcopy_id}
                                          #max 32 flashcopy ids are supported
```

```
DELETE /v1/cs/flashcopies/{flashcopy_id}
```

Unfreeze

```
POST /v1/cs/flashcopies/unfreeze
 {
 "request":
  {
  "params" :
   {
   "lss_ids" :
    [
    "00",
    "01",
       ...
    ]
   }
  }
 }
```

Hosts

Create

```
POST /v1/hosts
{
"request" :
 {
 "params" :
   {
   "name" : "testHost_1",
   "hosttype" : "Win2003"
   }
  }
 }
```

Update

```
PUT /v1/hosts/{host_name}
{
"request":
 {
 "params" :
  {
  "ioports" : [portID1,portID2,...] | all     #"ioports" : [] means remove all
  }                                           ioports from a specific host
 }
}
```

Delete

```
POST /v1/hosts/{host_name}
```

Host mapping

Create

```
POST /v1/hosts/{host_name}/mappings
{
 "request" :
 {
 "params" :
  {
  "volumes" : [0001,0002],
  "mappings" :
   [
   {00:0001},
   {01:0002}
   ]
  }
 }
}
```

Delete

```
DELETE /v1/hosts/{host_name}/mappings/{mapping_id}
```

Host ports

Create

```
POST /v1/host_ports
{
"request" :
 {
 "params" :
  {
  "wwpn" : "50050763030813A2",
  "host" : "myhost1"
  }
 }
}
```

Update

```
PUT /v1/host_ports/{wwpn}
{
"request" :
 {
 "params" :
  {
  "host" : "myhost1"
  }
 }
}
```

Delete

```
DELETE /v1/host_ports/{wwpn}
```

Logical subsystem

Create

```
POST /v1/lss
{
"request" :
 {
 "params" :
  {
  "id" : "00",
  "type" : "ckd"
  "sub_system_identifier" : "FF00"
  "ckd_base_cu_type" : "3990-3" | "3990-tpf" | "3990-6" | "bs2000"
  }
 }
}
```

Update

```
PUT /v1/lss/{lss_id}
{
"request" :
 {
 "params" :
  {
  "sub_system_identifier" : "FF01",
  "pprc_consistency_group" : "enable" | "disable",
  "critical_mode" : "enable" | "disable",
  "extended_long_busy_time" : "60",
  "cc_session_timeout" : "300",
  "xrc_session_timeout" : "300",
  "ckd_base_cu_type" : "3990-3" | "3990-tpf" | "3990-6" | "bs2000"
  }
 }
}
```

Delete

```
DELETE /v1/lss/{lss_id}
```

ODD files (DS8880, only)

Create

```
POST /v1/odd/offload
{
"request" :
  {
  "params" :
   {
   "file_names" :
          [
            "cpssdump17_ODDdump.150625_141348.1.0.tar.gz" ,
            "cpssdump17_ODDdump.150625_141339.1.1.tar.gz"
       ]
   }
  }
}
```

PE package (DS8880, only)

Create

```
POST /v1/pepackage/offload
{
"request" :
  {
  "params" :
   {
   "area" : "general" | "ui" | "ettraces" | "etlogs" | "panic" | "prolongedIO" ,
   }
  }
}
```

Remote Mirror and Copy

Create

```
POST /v1/cs/pprcs
 {
  "request" :
   {
    "params" :
     {
      "volume_pairs" :
         [
           {
      "source_volume" : "0000" ,
      "source_system_id" : "2107-1300861" ,
      "target_volume" : "1000" ,
      "target_system_id" : "2107-75YZ881" ,
           } ,
           ...
           ] ,
       "type" :
       "global_copy" | "metro_mirror" ,
       "option" :
           [
       "permit_space_efficient_target" ,
       "initial_copy_full" ,
       "failover"
           ]
```

```
        }
     }
  }
```

Update

```
PUT /v1/cs/pprcs?ids={pprc_id}},{pprc_id},…,{pprc_id}
```

```
PUT /v1/cs/pprcs/{pprc_id}
```

```
{
 "request" :
 {
  "params" :
   {
    "options" :
     [
       "setup_preserved_mirror" ,
    "reset_preserved_mirror"
      ]
    }
  }
}
```

Delete

```
DELETE /v1/cs/pprcs?ids={pprc_id},{pprc_id},…,{pprc_id}
```

```
DELETE /v1/cs/pprcs/{pprc_id}
```

```
POST /v1/cs/pprcs/delete
```

```
{
 "request" :
  {
   "params" :
    {
     "volume_full_ids" :
      [
       {
        "volume_id" : "0000" ,
        "system_id" : "2107-1300861"
        } ,
       ]
        "options" :
       [
        "unconditional" ,
        "issue_source" ,
        "issue_target"
      ]
    }
  }
}
```

Pause

```
POST /v1/cs/pprcs/pause
{
 "request" :
  {
   “params” :
    {
     "pprc_ids" :
      [
       "2107-1300861-0000:2107-1300855-1000" ,
       "2107-1300861-0001:2107-1300855-1001" ,
      ]
    }
  }
}
```

Resume

```
POST /v1/cs/pprcs/resume
{
 "request" :
  {
   "params" :
    {
     "pprc_ids" :
     [
      "2107-1300861-0000:2107-1300855-1000" ,
      "2107-1300861-0001:2107-1300855-1001" , …
      ] ,
    "type" : "global_copy" | "metro_mirror" ,
       "options" :
      [
     "permit_space_efficient_target" ,
     "initial_copy_out_of_sync" ,
        "failback"
      ]
    }
  }
}
```

Freeze

```
POST /v1/cs/pprcs/freeze
{
 "request" :
  {
   "params" :
    {
     "lss_pairs" :
      [
        {
          "source_lss_id" : "00" ,
          "source_system_id" : "2107-1300861" ,
          "target_lss_id" : "10" ,
          "target_system_id" : "2107-1300855"
         },
      ]
    }
  }
}
```

Unfreeze

```
POST /v1/cs/pprcs/unfreeze
{
 "request" :
  {
   "params" :
    {
     "lss_pairs" :
      [
        {
          "source_lss_id" : "00" ,
          "source_system_id" : "2107-1300861" ,
          "target_lss_id" : "10" ,
          "target_system_id" : "2107-1300855"
         } ,
      ]
    }
  }
}
```

Remote Mirror and Copy paths

Create

```
POST /v1/cs/pprcs/paths
{
   "request" :
      {
     "params" :
        {
          "source_lss_id" : "00" ,
          "target_lss_id" : "00" ,
```

```
    "pprc_consistency_group": "enable" | "disable",
```

```
    "target_system_wwnn" : "5005076306FFD341" ,
          "port_pairs" :
             [
            {
              "source_port_id" : "I0003" ,
              "target_port_id" : "I0007"
                  }
                ...
             ]
          }
      }
  }
```

Delete

```
DELETE /v1/cs/pprcs/paths/{path_id}
```

Space-efficient storage

Update

```
PUT /v1/pools/{pool_id}/tserep|eserep
{
"request":
 {
 "params" :
  {
  "cap" : "10",                                      (optional)
  "captype" : "gib" | "bytes" | "cyl" | "mod1" ,      (optional)
  "threshold" : "70"                                  (optional)
  }
 }
}
```

Delete

```
DELETE /v1/pools/{pool_id}/tserep|eserep
```

Statesave files (DS8880, only)

Create

```
POST /v1/statesave/offload
{
"request" :
  {
  "params" :
   {
   "file_names" :
           [
             "cpssdump17.150625_141348.1.0.tar.gz" ,
             "cpssdump17.150625_141339.1.1.tar.gz"
           ]
       }
    }
 }
```

Tokens

Create

```
POST /v1/token
{
"request":
 {
```

```
 "params" :
  {
  "username" : "username"
  "password" : "password"
  "maxliveInterval" : "86400000"          (optional)
  "maxInactiveInterval" : ""              (optional)
  }
 }
}
```

Volumes

Create

```
POST /v1/volumes
{
"request" :
  {
  "params" :
   {
   "name" : "mytest" ,
   "namecol" : [ "mytest1", "mytest2" ],
   "quantity" : "64" ,
   "stgtype" : "fb" | "ckd" ,
   "pool" : "P2" ,
   "cap" : "12" ,
   "captype" : "gib" | "bytes" | "cyl" | "mod1" ,      (optional)
   "lss" : "00" ,
   "tp" : "none" | "ese" | "tse"                       (optional)
   }
  }
```

Update

```
PUT /v1/volumes/{volume_id}
{
"request" :
{
 "params" :
  {
  "cap" : "20",                                       (optional)
  "captype" : "gib" | "bytes" | "cyl" | "mod1" ,      (optional)
  "pool" : "P3",                                      (optional)
  }
 }
}
```

Delete

```
DELETE /v1/volumes/{volume_id}
```

```
DELETE /v1/volumes?ids={volume_id},{volume_id},...,{volume_id}
```

Update responses

RESTful API displays responses that denote the success or failure of update requests.

Examples

Request and response of a successful operation

```
DELETE /v1/volumes/0000
{
 “server”:
  {
  “status” : ”ok”,
  “code” : ””,
  "message" : “The operation completed successfully.”
  }
}
```

Request and response of an unsuccessful operation

```
DELETE /v1/volumes/0000
{
“server”:
 {
 “status” : ”failed”,
 “code” : ”BE7A0001”,
 "message" : “Resource 0000 does not exist.”
 }
}
```

Error handling

RESTful API responds to each request with an HTTP status code and a server status code.

HTTP status code
The HTTP status code is a quick indicator of whether the request succeeded. This status code is also helpful for automation scripts that do not parse error responses in their entirety.

Server status code
The server status code is a more detailed error information to assist you in troubleshooting.

The status codes are enumerated as detailed in the following tables.

Table 2. HTTP status codes

HTTP status code	HTTP message	Description
200	OK	The request succeeded. The response contains either a list of resources (in case of query) or is empty.
201	Created	The request succeeded. A new resource is created. The response contains the new URL for this resource.
202	Accepted	The request has been accepted, and under processing.
301	Moved Permanently	The resource has been moved to the new URL(s) permanently.
400	Bad request	The request contains invalid scope keywords, or invalid URL parameters. Examples of invalid URL parameters are: non numeric values, redundant parameters, etc.
401	Unauthorized	The user credentials that were provided are not valid.
404	Not found	The request addresses non existing resources.
405	Method Not Allowed	The specified HTTP Method for the URL does not supported.
500	Internal Server error	The request is valid, but some application level error occurred. See server status for more information.
503	Service Unavailable	The service is temporarily unavailable.

Table 3. Server status codes

HTTP status code	Server status code	Server message	Description
200	empty	`The operation completed successfully.`	The request succeeded. The response contains either a list of resources (in case of query) or empty.
201	empty	`The operation completed successfully.`	The request succeeded. A new resource is created. The response contains the new URL for this resource.
202	empty	The operation completed successfully.	The request succeeded to be submitted. But it is still under processing. A resource is created or updated. If it is a creation operation, the response contains the new URL for this resource.
301	-	-	The resource has been moved permanently. The HTTP header attribute, "Location", gives the new URL
400, 401, 500, 503	BEXXXXXX	A short message which describes the code.	The request failed. A message describes the problem. The codes from the API server begin with BE74. Other BE codes are from the DS8000 microcode level.
404	-	-	The request addresses nonexistent resources. The response only contains HTTP header. No response body is returned.
405	-	-	The HTTP method for the request addresses is not supported.

Request URLs

You can use URLs to submit resource requests through the RESTful API.

The following request URLs are defined:

Events

- `GET /v1/events[?severity=info,warning,error&after=yyyy-MM-ddTHH:mm:ssZ&before=yyyy-MM-ddTHH:mm:ssZ]` – Lists events from a specific system.

Files (`DS8880, only`)

- `GET v1/files/performancesummary` – Displays latest performance summary.
- `GET v1/files/etdata` – Displays latest EasyTier data.
- `GET v1/files/sysdata` – Displays latest system data.

FlashCopy

- `GET /v1/flashcopy` – Lists FlashCopy volumes.
- `GET /v1/volumes/{volume_id}/flashcopy` – Lists all FlashCopy copies of a specific volume.

Hosts

- `GET|POST /v1/hosts` – Lists all hosts or creates a host.
- `GET|PUT|DELETE /v1/hosts/{host_name}` – Lists, updates, or deletes a host.

Host mappings

- `GET|POST /v1/hosts/{host_name}/mappings` – Lists or adds volumes' mappings under a host.
- `GET|DELETE /v1/hosts/{host_name}/mappings/{mapping_id}` – Lists or deletes a specific volume mapping under a host.

Host ports

- `GET|POST /v1/host_ports` – Lists all host ports or creates a host port.
- `GET|PUT|DELETE /v1/host_ports/{wwpn}` – Lists, updates, or deletes a specific host port.
- `GET /v1/hosts/{host_name}/host_ports` – Lists all host ports of a specific host.

I/O enclosures

- `GET /v1/io_enclosures` – Lists all I/O enclosures.
- `GET /v1/io_enclosures/{ioenclosure_id}` – Lists a specific I/O enclosure.

I/O ports

- `GET /v1/ioports` – Lists all I/O ports.
- `GET /v1/ioports/{port_id}` – Lists a specific I/O port.
- `GET /v1/hosts/{host_name}/ioports` – Lists all I/O ports of a specific host.

LSS

- `GET /v1/lss[?type=fb|ckd]` – Displays all LSSs of type FB or CKD.
- `GET /v1/lss/{lss_id}` – Displays a specific LSS.

Managed arrays

- `GET /v1/marrays` – Displays all managed arrays.
- `GET /v1/marrays/{marray_id}` – Displays a specific managed array.
- `GET /v1/pools/{pool_id}/marrays` – Displays all managed arrays of a specific pool.

Nodes

- `GET /v1/nodes` – Displays all nodes.
- `GET /v1/nodes/{node_id}` – Displays a specific node.

ODD files (`DS8880, only`)

- `GET /v1/odd` – Displays generated ODD dump files on both nodes.
- `GET /v1/odd/offload` – Displays status of latest ODD dump files offloading.
- `POST /v1/odd/offload/` – Sends ODD dump files to IBM.

PE package (`DS8880, only`)

- `GET /v1/pepackage/offload` – Displays latest PE package offloading status.
- `POST /v1/pepackage/offload` – Generates latest PE package(s) automatically and sends to IBM.

Pools

- `GET /v1/pools` – Lists all pools.
- `GET /v1/pools/{pool_id}` – Lists a specific pool.

Remote Mirror and Copy (formerly PPRC)

- `GET /v1/volumes/{volume_id}/pprc` – Displays the PPRC properties of a specific volume.
- `GET /v1/pprc` – Lists all PPRC relationships.
- `GET /v1/cs/pprcs?volume_id_from={volume_id}&volume_id_to={volume_id}`
- `GET /v1/cs/pprcs/{pprc_id}`
- `GET /v1/cs/physical_links?source_lss_id={source_lss_id}&target_lss_id={target_lss_id}&target_system_wwnn={target_system_wwnn}`
- `GET /v1/cs/pprcs/paths?source_lss_id_from={source_lss_id}&source_lss_id_to={source_lss_id}`
- `GET /v1/cs/pprcs/paths/{path_id}`

Self upgrade

- `GET /v1/self_upgrade` – Displays information on last upgrade.
- `POST /v1/self_upgrade` – Upgrades API services.

Space-efficient

- `GET|PUT|DELETE /v1/pools/{pool_id}/tserep|eserep` – Lists, updates, or deletes a TSE or ESE repository for a specific pool.

Statesave files (`DS8880, only`)

- `GET /v1/statesave` – Displays all generated statesave dump files.
- `GET /v1/statesave/offload` – Displays status of latest statesave dump files offloading.
- `POST /v1/statesave/offload` – Sends statesave dump files to IBM.

Systems

- `GET /v1/systems` – Lists all system properties.
- `GET /v1/systems/{serial_number}/performance` – Lists performance from a specific system.

Users

- `GET /v1/users` – Lists all system users.
- `GET /v1/users/{user_name}` – Lists a specific system user.

Volumes

- `GET|PUT|DELETE /v1/volumes/`*`{volume_id}`* – Lists, updates, or deletes a specific volume.
- `POST /v1/volumes` – Adds a volume.
- `GET /v1/lss/`*`{lss_id}`*`/volumes` – Displays volumes of a specific LSS.
- `GET /v1/pools/`*`{pool_id}`*`/volumes` – Lists volumes of a specific pool.
- `GET /v1/hosts/`*`{host_name}`*`/volumes` – Lists all volumes of a specific host.

Resource definitions

RESTful API sends the settings of the specified resource in response to all submitted **GET** requests.

Encryption groups

```
"data"
 {
 "encryption_groups":
  [
   {
   "id" : "1",
   "link" :
   {
   "rel" : "self",
   "href" : "https://{hmc}/api/v1/
   encryption_groups/1"
   },
   "state" : "accessible" | "rekeying" | "inaccessible" | "unconfigured"
   }
  ]
 }
```

Events

```
"data"
 {
 "events":
  [
   {
   "id" : "SE..."
   "type" : "PoolCapacityChanged" ,
   "severity" : "error",
   "time" : "2014-04-20T13:00:23-0700",
   "resource_id" : "",
   "formatted_parameter" : [ ]
   }
  ]
 }
```

FlashCopy relationships

```
"data"
 {
 "flashcopies":
  [
   {
   "id" : "0000:1000",
   "link" :
   {
   "rel" : "self",
   "href" : "https://{hmc}/api/v1/flashcopies/0000:1000"
   },
   "source_volume":
   {
   "id":"0000",
   "link":{ rel="self", href="https://{hmc}/api/v1/volumes/0000"}
   },
   "target_volume":
    {
    "id":"1000",
    "link":{ rel="self", href="https://{hmc}/api/v1/volumes/1000"}
    },
    "out_of_sync_tracks" : "0",
    "state" : "invalid" | "valid" | "invalid" | "validation_required"
  | "volume_inaccessible" | "tgt_failed"
    }
   ]
  }
```

Hosts

```
"data"
 {
  "hosts":
   [
    {
     "link" : { "rel" : "self", "href" : "https://{hmc}/api/v1/hosts/
     test_host" },
     "name" : "test_host",
     "state" : "online" | "offline",
     "hosttype" : "VMware",
     "addrmode" : "SCSI mask" | "SCSI map" | "OS/400 mask" | ,
     "addrdiscovery" : "lunpolling" | "reportlun",
     "lbs" : "512" | "520",
     "ioports" : {"link" : { "rel" : "self", "href" : "https://{hmc}/
     api/v1/hosts/test_host/ioports" }},
     "host_ports" : { "link" : { "rel" : "self", "href" : "/https://{hmc}/
     api/v1/hosts/test_host/host_ports" }},
     "volumes" : { "link" : { "rel" : "self", "href" : "/https://{hmc}/
     api/v1/hosts/test_host/volumes" }}
    }
   ]
 }
```

Host mappings

```
"data"
 {
 "mappings":
  [
   {
   "link" : { "rel" : "self", "href" : "https://{hmc}/api/v1/
   hosts/test_host/mappings/00" },
   "lunid" : "00",
   "volume" :
   {
   "id" : "0005",
   "link" : { "rel" : "self", "href" : "/https://{hmc}/api/v1/
   volumes/0005" }
   }
  }
 ]
}
```

Host ports

```
"data"
 {
  "host_ports":
   [
    {
     "link" : { "rel" : "self", "href" : "https://{hmc}/api/v1/
     host_ports/50050763030813A2" },
     "wwpn" : "50050763030813A2",
     "state" : "logged in" | "logged out" | "unconfigured",
     "hosttype" : "VMware",
     "addrdiscovery" : "lunpolling" | "reportlun",
     "lbs" : "512" | "520",
     "host":
    {
     "name" : "myhost",
     "link" : { "rel" : "self", "href" : "https://{hmc}/api/v1/
     hosts/myhost" }                 }
    }
   ]
  }
```

I/O enclosures

```
"data"
  {
  "io_enclosures":
   [
    {
    "id" : "1",
    "link" :
    {
```

```
    "rel" : "self",
    "href" : "https://{hmc}/api/v1/io_enclosures/1"
    },
    "name" : "cpssebay01"
    "state" : "online" | "offline" | "fenced" | "quiescing" | "resuming"
    }
   ]
  }
```

I/O ports

```
"data"
 {
 "fc_ports":
  [
   {
   "link" : { "rel" : "self", "href" : "https://{hmc}/api/v1/
   ioports/I0030" },
   "id" : "I0030",
   "state" : "online" | "offline" | "fenced" | "quiescing" |
   "resuming" | "deconfigured",
   "protocol" : "FICON" | "FC-AL" | "SCSI-FCP" | "SCSI-FCP/FICON",
   "wwpn" : "50050763030313A2",
   "type" : "ESCON" | "Fibre Channel-LW" | "Fibre Channel-SW" |
   "Fibre Channel-LW 4KM",
   "speed" : "8 Gb/s",
   "loc" : "U1400.1B1.RJ34736-P1-C4-T0"
   }
  ]
 }
```

LSSs

```
"data"
 {
 "lss":
  [
   {
   "id" : "00",
   "type" : "fb" | "ckd",
   "group" : "0",
   "addrgrp" : "0",
```

```
   "sub_system_identifier" : "FF10",
   "ckd_base_cu_type" : "3990-3" | "3990-tpf" | "3990-6" | "bs2000"
   "pprc_consistency_group" : "enable" | "disable",
   "critical_mode" : "enable" | "disable",
   "extended_long_busy_time" : "60",
   "cc_session_timeout" : "300",
   "xrc_session_timeout" : "300",
```

```
   "configvols" : "16",
   "volumes" :
    {
     "link" : { "rel" : "self", "href" : "https://{hmc}/api/v1/
     lss/00/volumes" }
    }
   }
  ]
 }
```

Managed arrays

```
"data"
 {
 "marrays":
  [
   {
    "id" : "MA1"
    "link" : { "rel" : "self", "href" : "https://{hmc}/api/v1/
    marrays/MA1" },
    "disk_class" : "enterprise" | "nearline" | "reserved" | "sata"
    | "ssd" | "flash" | "unknown",
    "pool" :
    {
       "id" : P0,
       "link" : { "rel" : "self", "href" : "https://{hmc}/api/v1/
       pools/P0" }
    }
   }
  ]
 }
```

ODD files (DS8880, only)

```
"data"
 {
 "odd":
 [
  {
   "status" : "ongoing" | "failed" | "successful"
  }
 ]
}
```

PE package (DS8880, only)

```
"data"
 {
 "pepackage":
 [
  {
   "area" : "general" | "ui" | "ettraces" | "etlogs" | "panic" | "prolongedIO" ,
   "status" : "ongoing" | "failed" | "partially_successful" | "successful"
  }
 ]
}
```

Performance

```
"data"
 {
 "performance":
 [
  {
   "name" : "NI5_2",
   "IOPS" :
  {
  "read": "54285.37",
  "write": "17371.14",
  "total": "71656.52"
  },
  "responseTime" :
  {
  "read": "1",
  "write": "1.83",
  "average": "1.2"
  },
  "performancesampletime": "2015-04-01T01:31:34-0700"
  },                        …
 ]
}
```

Pools

```
"data"
 {
 "pools":
  [
   {
   "link" : { "rel" : "self", href : "https://{hmc}/api/v1/pools/P0" },
   "id" : "P0",
   "name" : "mypool"
   "node" : "0",
   "stgtype" : "fb" | "ckd",
   "cap" : "634715635712",                               # bytes (FB)
   "capalloc" : "0",                                     # bytes (FB)
   "capavail" : "634715635712",                          # bytes (FB)
   "overprovisioned" : "2.4",
   "easytier" :  "managed" | "none" | "unknown"
   "extent_size" : "1GiB" | "1113cyl" | "16MiB" | "21cyl" | ""
   "tieralloc" :
    [
     { "tier" : "ENT" , "cap" : "634715635712", "allocated" : "0",          # bytes (FB)
     "assigned" : "0" }
    ]
    "threshold" : "80",
    "eserep" : { "link" : {  "rel" : "self", "href" : "https://{hmc}/api/
    v1/pools/P0/eserep" } },
    "tserep" : {"link" : {  "rel" : "self", "href" : "https://{hmc}/api/
    v1/pools/P0/tserep" } },
    "volumes" : {"link" : { "rel" : "self", "href" : "https://{hmc}/api/
    v1/pools/P0/volumes"}}
    }
   ]
  }
```

Remote Mirror and Copy (formerly PPRC)

```
"data"
  {
   "pprcs":
    [
      {
        "id" : "2107-1300861_0000:2107-75YZ881_1000",
        "source_volume":
       {
        "id":"0000",
        "link":{rel="self", href="https://{hmc}/api/v1/volumes/0000"}
               # link will be empty if this query is triggered on target
          },
        "source_system":
          {
        "id" : "2107-75YZ881",
        "link":{rel="self", href="https://{hmc}/api/v1/systems/2107-75YZ881"}
               # same as above
          }
         "target_volume":
             {
         "id":"1000", "link":{}
               # link will be empty if this query is triggered on source
          },
         "target_system"
             {
         "id" : "2107-75YZ881", "link" : {}
               # same as above
          },
         "type" : "metro_mirror" | "globalcopy" | "unknown",
         "state" : "copy_pending" | "full_duplex" | "target_copy_pending" |
         "target_full_duplex" | "target_suspended" |
         "suspended" | "invalid" | "volume_inaccessible" | "validation_required"
     }
   ]
 }
```

Remote Mirror and Copy (formerly PPRC) paths

```
"data":
 {
  "paths":
   [
    {
      "id": "5005076306FFD341_00:5005076306FFD341_00" ,
      "link":
        {
         "rel": "self" ,
         "href": "https:/r8gui1h.tuc.stglabs.ibm.com/api/v1/cs/
                  pprcs/paths/5005076306FFD341_00:5005076306FFD341_00"
           },
           "source_lss_id": "00" ,
           "target_lss_id": "00" ,
           "target_system_wwnn": "5005076306FFD341",
           "pprc_consistency_group": "disabled" | "enabled",
                "port_pairs":
                [
                    {
                        "source_port_id": "I0003",
                        "target_port_id": "I0007",
                        "state": "success" | "failed" | "invalid"
                    }
                ]
            }
        ]
    }
```

Remote Mirror and Copy (formerly PPRC) physical link

```
"data":
   {
    "physical_links":
      [
        {
          "source_port_id": "I0003",
          "target_port_id": "I0007"
        }
     ]
  }
```

Remote Mirror and Copy (formerly PPRC) volumes

```
"data"
 {
 "pprcs":
  [
   {
   "id" : "2107-1300861_0000:2107-75YZ881_1000" ,
   "source_volume" :
   {
   "id":"0000",
   "link": # link will be empty if this query is triggered on secondary
     {
      "rel"="self",
      "href"="https://{hmc}/api/v1/volumes/0000"
     }
    },
     "source_volume":
    {
    "id":"1000",
    "link": # same as above
    {
      "rel"="self",
      "href"="https://{hmc}/api/v1/systems/2107-75YZ881"
     }
  }
    "target_volume":
    {
    "id":"1000",
    "link":{} # link will be empty if this query is triggered on primary
    },
    "target_system"
  {
    "id":"2107-75YZ881",
    "link":{} # same as above
    },
    "type" : "metromirror" | "globalcopy" | "unknown"
    "state" : "copy_pending" | "full_duplex" | "target_copy_pending" |
    "target_full_duplex" | "target_suspended" | "suspended" | "invalid" |
    "volume_inaccessible" | "validation_required"
    }
   ]
  }
```

Self upgrade

```
"data"
 {
 "upgrade":
  [
   {
   "last_upgrade_time" : "2015-01-20T15:07:25-0700"
   "last_upgrade_status" : "successful" | "failed" ,
   "last_upgrade_message" : ""
   }
  ]
 }
```

Space-efficient storage

```
"data"
 {
```

```
 "eserep | tserep":
  [
   {
   "link" : { "rel" : "self", "href" : "https://{hmc}/api/v1/
   pools/P0/tserep" },
   "cap" : "16911433728",                   # bytes (FB)
   "capalloc" : "0",                        # bytes (FB)
   "capavail" : "16911433728",              # bytes (FB)
   "overprovisioned" : "2.4",
   "threshold" : "85",
   pool :
    {
    "id" : P0,
    "link" : { "rel" : "self", "href" : "https://{hmc}/api/v1/
    pools/P0" }
    }
   }
  ]
 }
```

Statesave files (`DS8880, only`)

```
"data"
 {
 "statesave":
 [
  {
   "status" : "ongoing" | "failed" | "successful"
  }
 ]
}
```

Systems

```
"data"
 {
  "systems":
   [
    {
     "name" : "NI5_2",
     "state" : "online" | "fenced" | "forced_quiescing" | "offline" |
     "quiescing" | "quiesce_exception" | "resuming" | "discovery",
     "release" : "7.4",
     "bundle" : "87.50.34.0",
     "mtm" : "2421-961",
     "sn" : "75YZ881",
     "cap" : "105871",                        # bytes (FB)
     "capalloc" : "75005",                    # bytes (FB)
     "capavail" : "30866",                    # bytes (FB)
     "capraw" : "228420"                      # bytes (FB)
    }
   ]
  }
```

Users

```
"data"
 {
 "users":
  [
   {
    "name" : "admin"
    "state" : "locked" | "active" ,
    "group" :
     [
     "Administrator",
     "SecurityAdministrator",
     "OperatorPhysicalConf",
     "OperatorLogicalConf”,
     "OperatorCopyServices”,
     "OperatorServiceFunctions”,
     "Monitor”,
     "Auditor”,
     "Default”
     ]
    }
   ]
  }
```

Volumes

```
"data"
 {
  "volumes" :
   [
    {
     "link" : { "rel" : "self", "href" : "https://{hmc}/api/v1/
     volumes/0000" },
     "id" : "0000",
     "name" : "mytest",
     "state" : "pinned" | "readonly" | "inaccessible" |
     "indeterminate_data_loss" |
     "rank_failed" | "rank_repairing" | "rank_repaired" |
     "global_inaccessible" |
     "global_lost" | "virtual_space_fault" | "nvs_inaccessible" |
     "normal" | "configuring" | "reconfiguring" | "migrating" |
     "deconfiguring" | "config_error" | "reconfig_error" |
     "migration_error" | "deconfig_error" | "merging" |
     "migration_paused" | "migration_cancelled" |
     "transposition_error" | "unknown"
     "cap" : "322122547200",                         # bytes (FB)
  "cap_gb": "300",
     "cap_gib": "322.1",
     "real_cap": "322122547200",
     "virtual_cap": "0";
     "stgtype" : "fb" | "ckd",
     "VOLSER" : "",
     "lss" : { "id" : "00",  "link" : { "rel" : "self", "href" :
     "https://{hmc}/api/v1/lss/00" } },
     "allocmethod" : "legacy" | "rotatevols" | "rotateexts" | "managed" | "",
     "tp" : "none",
     "capalloc" : "134217728",                       # bytes (FB)
     "MTM" : "2107-900",
     "datatype" : "FB 512" | "FB 512T" | "FB 520P" | "FB 520U" |
     "FB 520PV" | "FB 520UV" | "3380" | "3390" | "3390-A" | "",
     "easytier" : "managed" | "none" | "unknown"
     "tieralloc" :
     [
      { "tier" : "ENT" , "allocated" : "34502" }
     ]
     "pool" : { "id" : "P2", "link" : { "rel" : "self", "href" :
     "https://{hmc}/api/v1/pools/P2" }}
   }
  ]
 }
```

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBMproducts. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

Laser Notice: U.S. FDA CDRH NOTICE if low power lasers are utilized, integrated, or offered with end product systems as applicable. Complies with 21 CFR 1040.10 and 1040.11 except for conformance with IEC 60825-1 Ed. 3., as described in Laser Notice No. 56, dated May 8, 2019.

Trademarks

IBM®, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Copyright and trademark information website.

Firefox is a registered trademark of the Mozilla Foundation.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Printed in USA

SC27-9823-00