Tegile IntelliFlash™ API Reference Guide

Version 1.2

Notice

Copyright

Tegile Systems, Inc. | 7999 Gateway Blvd. | Suite 120 | Newark, | CA | 94560 | 1-855-5-TEGILE (1-855-583-4453) | www.tegile.com

Contents

Preface

About this guide

The Tegile IntelliFlash™ API Reference Guide contains detailed information about using the Tegile IntelliFlash REST APIs.

Note: When referring to older versions of the software, this document uses the new name **IntelliFlash** in place of **Zebi**.

Audience

The IntelliFlash API Reference Guide is intended for developers and solution engineers who will be creating applications using the IntelliFlash API.

Tegile Documentation

The following table: *Tegile Documentation* lists the technical documentation available for Tegile storage arrays and expansion shelves.

Table 1: Tegile Documentation

Name	Description
Tegile T-Series Hardware Guide	Contains detailed descriptions, hardware specifications, and rack installation instructions.
Tegile IntelliFlash™ User Guide	Contains detailed instructions on how configure, use, and administer Tegile arrays.
Tegile IntelliFlash™ API Reference Guide	Contains detailed descriptions of the Tegile REST API.

Support

Tegile support services give you access to online, telephone, and onsite support. Tegile provides multiple levels of support through a Technical Support team and Field Engineers. For details on Tegile support offerings, contact your Tegile Account team.

Contacting Tegile Technical Support

Telephone Support

Tegile Technical Support provides 24-hour/365 days-a-year telephone support for all customers who have a support contract. You can reach Tegile Technical Support by phone at the following numbers:

- US Toll Free: **`1-855-483-4453`**
- UK Toll Free: **`0-808-234-2044`**
- Australia Toll Free: **`1-800-937-949`**

Email and Online Support

You can email Tegile Technical Support at *support@tegile.com*. You can access the Tegile Technical Support portal at *https://support.tegile.com*.

Documentation Comments

We welcome your inputs on Tegile documentation. To share your feedback, send an email to: *doc-comments@tegile.com*. Please include the document title and revision, and refer to specific pages, topics, and paragraphs whenever possible.

Revision

Date	Description
08/25/2015	The *createSnapshots*, *deleteSnapshots*, *deleteSnapshots* and *cloneSnapshot* APIs are deprecated and have been moved to *Appendix B* section. **Documentation-only changes**: Included the *listLunsById* API and the *LunStatus* object that were omitted; fixed an incorrect example in the *startReplication* API; and several improvements and minor fixes to the descriptions.
03/20/2015	Terminology changes for IntelliFlash v3.0.0.
11/21/2014	Minor improvements and fixes to the document.
09/19/2014	Updated for IntelliFlash API version 1.2. Method documentation reorganized. Python sample added to the *Sample Programs* chapter. New APIs included in the following chapters *User and Group*, *Dataset*, *Snapshot*, and *Replication*. New objects and enumerations included.
02/06/2014	Initial release for IntelliFlash API version 1.0

Chapter 1

Introduction to the IntelliFlash API

Topics:

The IntelliFlash API provides an interface to securely configure and provision storage on IntelliFlash using a programming or scripting language. It enables you to integrate Tegile arrays with third-party software. It also allows you to automate common and repetitive tasks, such as retrieving a list of volumes and their status, provisioning new volumes, and creating and managing snapshots.

Important: You can use version 1.2 of the IntelliFlash API with Tegile arrays that have IntelliFlash version 2.1.2.5 or higher, such as IntelliFlash version 2.1.3 or IntelliFlash version 2.1.3.5.

Key Features

The key features of the IntelliFlash API are:

- **REST (Representational State Transfer) API**: The API uses HTTP 1.1 request methods. Because HTTP is a well-known protocol and many scripting languages support it, it simplifies the task of building scripts and applications that use the API.
- **JSON (JavaScript Object Notation) data structures**: The API uses JSON as the data exchange format. All parameters in requests sent by the client must use JSON. Similarly, the responses sent by the server (including error responses) are JSON data structures.
- **HTTPS**: The API uses HTTPS to secure the communication between the client and the server.
- **Basic Authentication**: The API uses HTTP Basic authentication over Transport Layer Security (TLS)/Secure Sockets Layer (SSL). This allows only authorized personnel/programs to securely access the API.

Warning: Because the API requires the IntelliFlash Web UI administrator credentials for authentication, make sure to adequately secure the machine and the user account from which the client programs/scripts are run. Administrators must ensure that the credentials are not compromised by someone reading the script.

See Also

- For more information about REST, see this Wikipedia article on *Representational State Transfer*.
- For more information about JSON, see this Wikipedia article on *JSON*. For a quick review of JSON data structure, see *Appendix A: JSON Quick Reference* .
- For more information on HTTPS, see this Wikipedia article on *HTTPS*.

Scope of the API

The functionality exposed by the IntelliFlash API includes:

- Listing pools, projects, volumes, users, groups, luns, shares, snapshots, initiators, initiator groups, targets, target groups, and system properties.
- Creating users, groups, users and groups, volumes, snapshots, initiator groups, mapping for volumes, and iSCSI initiators.
- Cloning snapshots.
- Deleting users, groups snapshots, volumes, shares, mapping for volume, volume snapshot, share snapshot, and other datasets.
- Checking if an initiator group exists.
- Adding an initiator to an initiator group.
- Retrieving the initiator group for an initiator.
- Getting replication configuration list, status, and starting replication.

Error and Exception Handling in the API

In situations where a method does not succeed, the API will return one or more of the following responses:

- An HTTP status code that indicates an error:
  - A 4xx client error as described *on Wikipedia*.
  - A 5xx server error as described *on Wikipedia*.
- An integer that indicates an error (for example the values listed in the *COMMAND_STATUS*, *CLEANUP_STATUS*, and *SNAPSHOT_PROGRESS_STATUS* enumerations.)
- A JSON object that contains a enumeration field that indicates an error (as described in the previous point).

Best practices for accessing the controllers using the API

API calls are directed to the hostname or the IP that you use to invoke the APIs. For a Tegile array, your script, or program, can communicate with either controller. This could be an issue in the following circumstances:

- A failover occurs and the controller to which the API calls were directed becomes inaccessible.
- The array has pools on both controllers, and you are not sure which controller to use.

The following best practices ensure that the above situations do not occur and your API executes correctly.

Best Practice 1: Array with single pool

If your Tegile array has a single pool, you can ensure that the API calls work irrespective of the controller that the pool resides on by completing the following steps:

1. Create a floating IP address for the management interface groups (mgmt0) on both controllers.

   You can create a floating IP address from the **Settings** > **HA** page of the IntelliFlash Web UI. See the *Tegile IntelliFlash™ User Guide* for more information.

2. Use this floating IP address to invoke the API.

Best Practice 2: Array has two or more pools (one or more on each controller)

If your Tegile array has an active-active configuration (in which each controller has one or more pools), your script, or program, must be able to access both controllers.

To ensure that your script accesses the correct controller, follow this procedure in the script that invokes the API:

1. Use the management interface group (`mgmt0` or `ipmp0`) of each controller.
2. Query the pools to determine which pools are currently on the controller.
3. Execute additional API commands after confirming that the intended target pool is currently on the controller with which you are communicating. This ensures that your script is communicating with the pool that you are targeting.

Best Practice 3: Program that is invoking the APIs can access the data network of the array

If the program or system that is invoking the APIs has access to the data network of the array, it should use a floating IP address that is in the same resource group as the pool that it is targeting. This enables the program to work, regardless of the controller the pool is currently active on.

To use a floating IP address that is in the same resource group as the pool that the API is targeting, complete the following steps:

- From the **Settings** > **HA** page of the IntelliFlash Web UI, ensure you have a floating IP address in the same resource group as the pool.
- If not, you can create the required floating IP address from the same page (**Settings** > **HA** .) To create the required floating IP address, click **Add Floating IP** in the required resource group.

For more information, see the Tegile IntelliFlash™ User Guide.

Chapter 2

Common Workflows

Topics:

The following sections describe some of the common workflows that can be automated using the IntelliFlash API.

Creating a Volume and Exposing it

You can create a volume and expose it using the IntelliFlash API.

Prerequisites

To accomplish this, you must first ensure that the Tegile array controller contains the following:

- A pool
- A project in that pool
- An FC or iSCSI target
- An FC or iSCSI target group
- A mapping between the target and the target group

After ensuring that the prerequisites listed above are met, use the IntelliFlash API to:

1. Create a volume using the *createVolume* method.
2. Create an iSCSI initiator using the *createIscsiInitiator* method.
3. Create an initiator group using the *createInitiatorGroup* method.

   Note: You do not need to create FC initiators as they automatically get added because of NPIV.

4. Associate the initiator with the initiator group using the *addInitiatorToInitiatorGroup* method.
5. Map the volume to a target group and an initiator group using the *createMappingForVolume* method.

Backing up a Volume

You can back up an existing IntelliFlash volume using the IntelliFlash API.

To accomplish this, use the IntelliFlash API to complete the following steps:

1. Create a volume snapshot using the *createVolumeSnapshot* method.
2. Create a clone from the volume snapshot using the *cloneVolumeSnapshot* method.
3. Create a volume from the clone using the *createVolume* method.
4. Clean-up the volume snapshot and clones using the *deleteVolumeSnapshot* method.

Chapter 3

Using the API Method Reference

Topics:

- *Using the API Examples*
- *curl syntax*

The documentation of each API method includes descriptions of the method and its parameters and return type. It also include examples with sample responses.

Using the API Examples

The documentation for each API method includes examples.

The examples use the **curl** command for the requests. The documentation for most API methods includes two types of examples:

- Working examples with sample responses.
- Erroneous examples with error responses. These examples are erroneous because they use incorrect data for a particular context. The purpose of the erroneous examples is only to illustrate some of the responses that a Tegile array controller will return if it receives incorrect data.

Before using the examples in your scripts and programs, ensure that you make the following changes:

- Use the authentication token returned by your Tegile array controller instead of the dummy token (AUTH_TOKEN) given in the examples. The authentication token must be encoded as a Base64 string to use the REST API. For example, you can use the following Linux command (that is part of the Linux coreutils package) to convert your credentials to Base64.

```
# echo 'username:password' | base64
```

- Use data that is relevant to your environment and requirements instead of the dummy data given in the examples.
- Use the IP address of your Tegile array controller instead of the dummy IP address given in the examples.

curl syntax

The examples use the **curl** command to represent the HTTP requests.

The **curl** examples include the HTTP headers, the JSON data sent in the request, and the endpoint of the API.

Tegile storage arrays use a self-signed certificate. This may prevent the **curl** command from working. As a workaround, you can use the **-k** parameter with the **curl** command to ignore the self-signed certificate.

Chapter 4

Sample Programs

Topics:

The following sample programs illustrate how to access the IntelliFlash API using Perl and Python.

Note: The IntelliFlash API uses basic authentication over HTTPS. If you are using self-signed certificates on the Tegile array, the program that invokes the IntelliFlash APIs must include instructions to trust the SSL certificate.

Sample Perl Script

The following Perl script illustrates how to authenticate, accept (trust) the self-signed certificate, and invoke the *listPools* API.

```
use REST::Client;
use JSON;
use Data::Dumper;
use MIME::Base64;

# next line is for ignoring the certificate if it is self-
signed.
$ENV{PERL_LWP_SSL_VERIFY_HOSTNAME}=0;

$username = 'admin';
$password = 't';
my $host= "https://198.51.100.10";
my $url = "/zebi/api/v1/listPools";
my $json_data = "";

# Below line is for basic authentication
my $headers = { Accept => 'application/json',
      Authorization => 'Basic '. encode_base64($username . ':' .
 $password)};

my $client = REST::Client->new();
$client->setHost($host);
$client->setTimeout(60);

# For API Call
$client->POST($url,($json_data,$headers));

print Dumper ($client->responseContent());
```

Note: You must replace the IP (198.51.100.10) with the IP of your Tegile array controller.

Sample Python Program

The following Python program illustrates how to authenticate, accept (trust) the self-signed certificate, and invoke the following APIs:

- *listPools*
- *listProjects*
- *listVolumes*
- *createVolumeSnapshot*
- *cloneVolumeSnapshot*
- *getReplicationConfigList*
- *startReplication*

- *getReplicationStatus*

```
import httplib2;
import base64;
import json;

h = httplib2.Http();
auth = base64.encodestring('admin' + ":" + "t");

url = "https://198.51.100.10/zebi/api/v1/listPools";
method = "GET";
headerMap = {'content-type':'application/json',
 'Authorization' : 'Basic ' + auth};

resp, content = h.request(url, method, headers=headerMap);

poolArray = json.loads(content);

# List projects inside a the pool
url = "https://198.51.100.10/zebi/api/v1/listProjects";
method = "POST";

poolName = poolArray[0]["name"];

#Prepare Parameter Array
paramArray = [];
paramArray.append(poolName);
paramArray.append(True);
paramJSONData = json.dumps(paramArray);

resp, content = h.request(url, method, paramJSONData,
 headers=headerMap);

projectArray = json.loads(content);

print resp.status;
print content;

#List Volumes inside a project

url = "https://198.51.100.10/zebi/api/v1/listVolumes";
method = "POST";

projectName = projectArray[0]["name"];
paramArray = [];
paramArray.append(poolName);
paramArray.append(projectName);
paramArray.append(True);
paramJSONData = json.dumps(paramArray);
resp, content = h.request(url, method, paramJSONData,
 headers=headerMap);

volumeArray = json.loads(content);

#Create a volume snapshot
url = "https://198.51.100.10/zebi/api/v1/createVolumeSnapshot";
method = "POST";
snapName = "API-SNAP";

firstVolume = volumeArray[0];
```

```
paramArray = [];
paramArray.append(firstVolume);
paramArray.append(snapName);
paramArray.append(False);
paramJSONData = json.dumps(paramArray);
print paramJSONData;
resp, content = h.request(url, method, paramJSONData,
 headers=headerMap);

print resp.status;
print content;

url = "https://198.51.100.10/zebi/api/v1/cloneVolumeSnapshot";
method = "POST";

snapshotPath = firstVolume["datasetPath"] + "@" + "Manual-V-" +
 snapName;
volumeCloneName = "API-Clone";

paramArray = [];
paramArray.append(snapshotPath);
paramArray.append(volumeCloneName);
paramArray.append(False);
paramArray.append(True);
paramJSONData = json.dumps(paramArray);
print paramJSONData;
resp, content = h.request(url, method, paramJSONData,
 headers=headerMap);
print resp.status;
print content;

#Replication configs
url = "https://198.51.100.10/zebi/api/v1/
getReplicationConfigList";
method = "POST";
projectName = "vdi";

paramArray = [];
paramArray.append(poolName);
paramArray.append(projectName);
paramJSONData = json.dumps(paramArray);
print paramJSONData;
resp, content = h.request(url, method, paramJSONData,
 headers=headerMap);

replicationConfigArray = json.loads(content);
firstReplicationConfig = replicationConfigArray[0];

#Trigger replication
url = "https://198.51.100.10/zebi/api/v1/startReplication";
method = "POST";
projectName = "vdi";

paramArray = [];
paramArray.append(firstReplicationConfig);
paramJSONData = json.dumps(paramArray);
print paramJSONData;
resp, content = h.request(url, method, paramJSONData,
 headers=headerMap);

print resp.status;
```

```
print content;

url = "https://198.51.100.10/zebi/api/v1/getReplicationStatus";
method = "POST";
projectName = "vdi";

paramArray = [];
paramArray.append(firstReplicationConfig);
paramJSONData = json.dumps(paramArray);
print paramJSONData;
resp, content = h.request(url, method, paramJSONData,
 headers=headerMap);

print resp.status;
print content;
```

Note: You must replace the IP (198.51.100.10) with the IP of your Tegile array.

Chapter 5

User and Group Methods

Topics:

listUsers

Lists all the local users with their user Id, group name, and group Id. This is an HTTP GET method.

Related APIs

listGroups, *createUser*, *createUser*, *createUserAndGroup*, *deleteUser*.

Parameters

None

Returns

Returns an array of JSON objects. Each object has the user name, user id, group name, and group id of a local user.

Example

Request (curl)

```
curl -X GET -H "Authorization:Basic Auth_TOKEN" \
  -H Content-Type:application/json \
 https://198.51.100.10/zebi/api/v1/listUsers -k
```

Response

The above request returns the HTTP status code 200 (OK) and the following data:

```
  [
    {
      userName:    "username"
      groupName:    "groupname"
      userId:    104
      groupId:    104
    }
  ]
```

listGroups

Lists all the local groups and the users included in each group. This is an HTTP GET method.

Related APIs

createGroup, *createGroup*, *deleteGroup*, *createUserAndGroup*, *listUsers*.

Parameters

None

Returns

Returns an array of JSON objects. Each object has the group name, group id, and users of a group. The user list itself is a JSON array containing the names of users in a group.

Example

Request (curl)

```
curl -X GET -H "Authorization:Basic Auth_TOKEN" \
  -H Content-Type:application/json \
 https://198.51.100.10/zebi/api/v1/listGroups -k
```

Response

The above request returns the HTTP status code 200 (OK) and the following data:

```
[
  {
    groupName:   "TestGroup",
    groupId:   104,
    userList: [
                "TestUser"
              ]
  }
]
```

createUser

Creates a user with given username, user Id, group name, and password.

Related APIs

listUsers, *listGroups*, *createUser*, *createUserAndGroup*, *deleteUser*.

Parameters

userName

Username of the new user. The characters /, \\, !, @, #, $, %, ^, *, (,), :, ;, \, are not allowed in the username. The empty and "guest" strings and the null value are also not allowed in the username.

uid

User Id of the new user.

groupName

Name of the group in which the new user will be included. The characters /, \ \, !, @, #, $, %, ^, *, (,), :, ;, \, are not allowed in the groupname. The empty and "guest" strings and the null value are also not allowed in the groupname.

password

Password of the new user. The '/' and space charatcers and the empty and null strings are not allowed in password.

Returns

Returns an integer: the number 0 if the request succeeds.

Examples

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
  -H Content-Type:application/json \
 -d '["UserName", 123, "GroupName", "newpwd"]' \
 https://198.51.100.10/zebi/api/v1/createUser -k
```

Response

The above request returns the HTTP status code 200 (OK) and the following data:

```
  0
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
  -H Content-Type:application/json \
 -d '["UserName", 123, "IncorrectGroupName", "newpwd"]' \
 https://198.51.100.10/zebi/api/v1/createUser -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
  {
    message:   "Group IncorrectGroupName does not exist."
    extendedData: { }
```

```
    details:   ""
    code:   "EZEBI_GENERAL"
  }
```

createUser

Creates a user with given username, group name, and password. The user ID is generated by the IntelliFlash OS.

Related APIs

listUsers, *listGroups*, *createUser*, *createUserAndGroup*, *deleteUser*.

Parameters

userName

Username of the user that will be created. The characters /, \\, !, @, #, $, %, ^, *, (,), :, ;, \, are not allowed in the username. The empty and "guest" strings and the null value are also not allowed in the username.

groupName

Name of the group in which the new user will be included. The characters /, \ \, !, @, #, $, %, ^, *, (,), :, ;, \, are not allowed in the groupname. The empty and "guest" strings and the null value are also not allowed in the groupname.

password

Password of the new user. The '/' and space charatcers and the empty and null strings are not allowed in password.

Returns

Returns an integer: the number 0 if the request succeeds.

Examples

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["NewUserName", "ExistingGroupName", "NewUserPwd"]' \
 https://198.51.100.10/zebi/api/v1/createUser -k
```

Response

The above request returns the HTTP status code 200 (OK) and the following data:

```
0
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["NewUserName", "NotAGroupName", "NewUserPwd"]' \
 https://198.51.100.10/zebi/api/v1/createUser -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
  message:   "Group TechPub does not exist."
  extendedData: { }
  details:   ""
  code:   "EZEBI_GENERAL"
}
```

createGroup

Creates a user group with the specified group name and group ID.

Related APIs

listGroups, *createGroup*, *createUserAndGroup*, *deleteGroup*, *listUsers*, *deleteUser*.

Parameters

groupName

Name of the group. The characters /, \\, !, @, #, $, %, ^, *, (,), :, ;, \, are not allowed in the groupname. The empty and "guest" strings and the null value are also not allowed in the groupname.

gid

Group ID of the group. The group id should be a number less than 99999999.

Returns

Returns an integer: the number 0 if the request succeeds.

Examples

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
  -H Content-Type:application/json \
 -d '["NewGroup", 1234]' \
 https://198.51.100.10/zebi/api/v1/createGroup -k
```

Response

The above request returns the HTTP status code 200 (OK) and the following data:

```
  0
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
  -H Content-Type:application/json \
 -d '["NewGroup", 1234]' \
 https://198.51.100.10/zebi/api/v1/createGroup -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
  {
    message:   "The specified group Id already belongs to
another group."
    extendedData: { }
    details:   ""
    code:   "EZEBI_GENERAL"
  }
```

createGroup

Creates a user group with the specified group name. The group ID is generated by the system.

Related APIs

listGroups, *createGroup*, *createUserAndGroup*, *deleteGroup*, *listUsers*, *deleteUser*.

Parameters

groupName

Name of the group. The characters /, \\, !, @, #, $, %, ^, *, (,), :, ;, \, are not allowed in the groupname. The empty and "guest" strings and the null value are also not allowed in the groupname.

Returns

Returns an integer: the number 0 if the request succeeds.

Examples

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
  -H Content-Type:application/json \
 -d '["AnotherNewGroup"]' \
 https://198.51.100.10/zebi/api/v1/createGroup -k
```

Response

The above request returns the HTTP status code 200 (OK) and the following data:

```
  0
```

Erroneous Request (curl)

```
  curl -X POST -H "Authorization:Basic Auth_TOKEN" \
    -H Content-Type:application/json \
    -d '[""]' https://198.51.100.10/zebi/api/v1/createGroup -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
  {
    message:   "Invalid Group Name."
    extendedData: { }
    details:   ""
    code:   "EZEBI_GENERAL"
  }
```

createUserAndGroup

Creates user and group with auto generated user ID and group ID. The group will be created first and then the user. The user will be associated with the group. If the group with given group name already exists, the user will be created and associated with the existing group.

Related APIs

listUsers, *listGroups*, *createGroup*, *createGroup*, *deleteGroup*, *createUser*, *createUser*, *deleteUser*.

Parameters

userName

Username of the new user. The characters /, \\, !, @, #, $, %, ^, *, (,), :, ;, \, are not allowed in the username. The empty and "guest" strings and the null value are also not allowed in the username.

password

Password of the new user.

groupName

Name of the group in which the new user will be included. The characters /, \ \, !, @, #, $, %, ^, *, (,), :, ;, \, are not allowed in the groupname. The empty and "guest" strings and the null value are also not allowed in the groupname.

Returns

Returns an integer: the number 0 if the request succeeds.

Examples

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
  -H Content-Type:application/json \
 -d '["testUser2", "testpwd2", "testGroup2"]' \
 https://198.51.100.10/zebi/api/v1/createUserAndGroup -k
```

Response

The above request returns the HTTP status code 200 (OK) and the following data:

```
  0
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
  -H Content-Type:application/json \
 -d '["testUser2", "testpwd2", "testGroup2"]' \
 https://198.51.100.10/zebi/api/v1/createUserAndGroup -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
  {
    message:   "UX: /usr/sbin/useradd: ERROR: testUser2 is
 already in use. Choose another.\ 9"
    extendedData: { }
    details:   ""
    code:   "EZEBI_GENERAL"
  }
```

changeUserPassword

Sets a new password for the specified user (a "Local User" that was created on the Tegile array.)

Related APIs

listUsers

Parameters

userName

Name of the user.

password

Password of the new user. The '/' and space charatcers and the empty and null strings are not allowed in password.

Returns

Returns an integer: the number 0 if the request succeeds.

Examples

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '["UserName", "Password"]' \
https://198.51.100.10/zebi/api/v1/changeUserPassword -k
```

Response

The above request returns the HTTP status code 200 (OK) and the following data:

```
0
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
  -H Content-Type:application/json \
  -d '["IncorrectUserName", "Password"]' \
  https://198.51.100.10/zebi/api/v1/changeUserPassword -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
  message: "User IncorrectUserName does not exist."
  extendedData: { }
  details: ""
  code: "EZEBI_GENERAL"
}
```

deleteUser

Deletes the specified user (a "Local User" that was created on the Tegile array.)

Related APIs

listUsers, *listGroups*, *deleteGroup*.

Parameters

userName

Name of the user. The characters /, \\, !, @, #, $, %, ^, *, (,), :, ;, \, are not allowed in the username. The empty and "guest" strings and the null value are also not allowed in the username.

Returns

Returns an integer: the number 0 if the request succeeds.

Examples

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
  -H Content-Type:application/json \
 -d '["testUser2"]' \
 https://198.51.100.10/zebi/api/v1/deleteUser -k
```

Response

The above request returns the HTTP status code 200 (OK) and the following data:

```
 0
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
  -H Content-Type:application/json \
 -d '["testUser2"]' \
 https://198.51.100.10/zebi/api/v1/deleteUser -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
  {
    message:   "User testUser2 does not exist."
    extendedData: { }
    details:   ""
    code:   "EZEBI_GENERAL"
  }
```

deleteGroup

Deletes the specified user group (a "Local Group" that was created on the Tegile array). If the group contains existing users, all the users would not be part of this group.

Warning: If you do not require the users in this group, it is recommended to delete the users before deleting the group.

Related APIs

listGroups, *listUsers*, *deleteUser*, *createUserAndGroup*.

Parameters

groupName

Name of the group that has to be deleted. The characters /, \\, !, @, #, $, %, ^, *, (,), :, ;, \, are not allowed in the groupname. The empty and "guest" strings and the null value are also not allowed in the groupname.

Returns

Returns an integer: the number 0 if the request succeeds.

Examples

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
  -H Content-Type:application/json \
 -d '["NewGroup"]' \
 https://198.51.100.10/zebi/api/v1/deleteGroup -k
```

Response

The above request returns the HTTP status code 200 (OK) and the following data:

```
  0
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
  -H Content-Type:application/json \
  -d '["NewGroup"]' \
  https://198.51.100.10/zebi/api/v1/deleteGroup -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
  {
    message:   "Group NewGroup does not exist."
    extendedData: { }
    details:   ""
```

```
    code:   "EZEBI_GENERAL"
  }
```

Chapter 6

SAN Methods

Topics:

listISCSIInitiatorGroups

Lists all the iSCSI initiator groups available on a Tegile array. This is an HTTP GET method.

Related APIs

listISCSITargetGroups, *initiatorGroupExists*, *addInitiatorToInitiatorGroup*, *createMappingForVolume*.

Parameters

None

Returns

Returns an array of JSON strings. Each string has the names of all iSCSI Initiator groups on the Tegile array.

Examples

Request (curl):

```
curl -X GET -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
https://198.51.100.10/zebi/api/v1/listISCSIInitiatorGroups \
-k
```

Response

```
[
  "iqn.2012-11.com.tegile.iscsi:testinigroup-group"
]
```

listISCSITargetGroups

Lists all the iSCSI target groups available on a Tegile array. This is an HTTP GET method.

Related APIs

listISCSIInitiatorGroups, *createMappingForVolume*.

Parameters

None

Returns

Returns an array of JSON strings. Each string has the names of all iSCSI target groups on the Tegile array.

Example

Request (curl):

```
curl -X GET -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 https://198.51.100.10/zebi/api/v1/listISCSITargetGroups -k
```

Response

```
[
 "default-p1aut-iscsi-target-group","testtargetgroup"
]
```

listFCInitiatorGroups

Lists the names of all Fibre Channel initiator groups created on a Tegile array. This is an HTTP GET method.

Related APIs

listFCTargetGroups, *listInitiatorsInInitiatorGroup*, *createMappingForVolume*.

Parameters

None

Returns

Returns an array of JSON strings. Each string has the names of all Fibre Channel Initiator groups on the Tegile array.

Examples

Request (curl):

```
curl -X GET -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 https://198.51.100.10/zebi/api/v1/listFCInitiatorGroups  -k
```

Response

```
[
```

```
 "newFCInitiatorGrp"
]
```

listFCTargetGroups

Lists all Fibre Channel Target groups available on a Tegile array. This is an HTTP GET method.

Related APIs

listFCInitiatorGroups, *createMappingForVolume*.

Parameters

None

Returns

Returns an array of JSON strings. Each string has the names of all Fibre Channel Target groups on the Tegile array. If array does not have any FC card, an empty array is returned.

Example

Request (curl):

```
curl -X GET -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 https://198.51.100.10/zebi/api/v1/listFCTargetGroups -k
```

Response

```
[
 "default-fc-target-group"
]
```

initiatorGroupExists

Checks if an initiator group exists on the Tegile array.

Related APIs

listInitiatorsInInitiatorGroup, *addInitiatorToInitiatorGroup*

Parameters

initiatorGroupName

A string: the name of the initiator group.

Returns

Returns a boolean value: **`true`** if the group exists, and **`false`** if it does not.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["iqn.2012-11.com.tegile.iscsi:testinigroup-group"]' \
 https://198.51.100.10/zebi/api/v1/initiatorGroupExists -k
```

Response

If the initiator group exists, the above request returns the HTTP status code 200 (OK) and the following data:

```
 true
```

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json -d '[2012]' \
 https://198.51.100.10/zebi/api/v1/initiatorGroupExists -k
```

Response

If the initiator group does not exist, the above request returns the HTTP status code 200 (OK) and the following data:

```
  false
```

createInitiatorGroup

Creates an initiator group on a Tegile array.

Related APIs

createIscsiInitiator, *listInitiatorsInInitiatorGroup*, *addInitiatorToInitiatorGroup*

Parameters

initiatorGroupName

A string: name of the new initiator group. The characters ,, /,\\, !, ?, @, <, >, #, $, ',%, ^,*,(,), ~,+, =, },|, {, [,], ;, \', \", _, & are not allowed in initiatorgroupname. The empty and space characters and the null values are not allowed in initiatorgroupname.

Returns

Returns an integer:

- (0) indicates that the request succeeded
- (1) indicates that the request was not attempted
- (2) indicates that the request failed

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '[ "APIInitiatorGroup" ]' \
 https://198.51.100.10/zebi/api/v1/createInitiatorGroup -k
```

Response

The above request returns the HTTP status code 200 (OK) and the following data:

```
  0
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '[ "API_InitiatorGroup" ]' \
 https://198.51.100.10/zebi/api/v1/createInitiatorGroup -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
  "message":"Initiator group name cannot have special
 characters
              like *,#,/,\\,!,@,~,(,),[,],{,},=,%...",
  "extendedData":{},
  "details":"",
  "code":"EZEBI_GENERAL"
```

```
}
```

listInitiatorsInInitiatorGroup

Lists all initiators belonging to the specified initiator group.

Related APIs

initiatorGroupExists, *getInitiatorGroup*, *listTargetsInTargetGroup*.

Parameters

initiatorGroupName

A string: name of an iSCSI or a Fibre Channel initiator group.

Returns

Returns an array of JSON strings. Each string has the names of all the initiators in the specified initiator group.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["APIInitiatorGroup"]' \
https://198.51.100.10/zebi/api/v1/\
listInitiatorsInInitiatorGroup -k
```

Response

```
[
 "iqn.2012-11.com.tegile.iscsi:testinigroup"
]
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["iqn.2012-11.com.tegile.iscsi:testinigroup-grp"]' \
 https://198.51.100.10/zebi/api/v1/
listInitiatorsInInitiatorGroup -k
```

Error Response

If the initiator group is not found, the above request returns the HTTP status code 200 (OK) and with no data.

listTargetsInTargetGroup

Lists all targets associated with the target group.

Related APIs

listInitiatorsInInitiatorGroup, *createMappingForVolume*.

Parameters

targetGroupName

A string: name of an iSCSI or a Fibre Channel target group.

Returns

Returns an array of JSON strings. Each string has the names of targets in the specified target group.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["default-test-iscsi-target-group"]' \

https://198.51.100.10/zebi/api/v1/listTargetsInTargetGroup \
-k
```

Response

```
["iqn.2012-02.com.tegile:zebi2-test"]
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["default-test1-iscsi-target-group"]' \

https://198.51.100.10/zebi/api/v1/listTargetsInTargetGroup \
-k
```

Error Response

If the target group is not found, the above request returns the HTTP status code 200 (OK) and with no data.

createIscsiInitiator

Creates an iSCSI initiator object on the Tegile array.

Related APIs

initiatorGroupExists, *addInitiatorToInitiatorGroup*

Parameters

iscsiInitiator

A JSON object of type *IscsiInitiator_V1_0* that contains the name of the initiator and optional CHAP information.

Returns

Returns an integer:

- (0) indicates that the request succeeded
- (1) indicates that the request was not attempted
- (2) indicates that the request failed

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["APIInitiator"]' \
 https://198.51.100.10/zebi/api/v1/createIscsiInitiator -k
```

Response

The above request returns the HTTP status code 200 (OK) and the following data:

```
 0
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["API_Initiator_2"]' \
```

```
 https://198.51.100.10/zebi/api/v1/createIscsiInitiator -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
  "message":"Initiator group name cannot have special characters
             like *,#,/,\\,!,@,~,(,),[,],{,},=,%...",
  "extendedData":{},
  "details":"",
  "code":"EZEBI_GENERAL"
}
```

getInitiatorGroup

Gets the name of the initiator group to which the initiator belongs.

Related APIs

listInitiatorsInInitiatorGroup, *createIscsiInitiator*, *addInitiatorToInitiatorGroup*

Parameters

initiatorName

A string: name of an initiator.

Returns

Returns an array of JSON strings. Each string has the name of the initiator group associated with the given initiator.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["iqn.2012-11.com.tegile.iscsi:api_initiator_1"]' \
 https://198.51.100.10/zebi/api/v1/getInitiatorGroup -k
```

Response

The above request returns the HTTP status code 200 (OK) and the following data:

```
 iqn.2012-11.com.tegile.iscsi:testinigroup-group
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["iqn.2012-11.com.tegile.iscsi:api_initiator_2"]' \
 https://198.51.100.10/zebi/api/v1/getInitiatorGroup -k
```

Error Response

If the initiator name is not found, the request returns the HTTP status code 200 (OK) and with no data.

addInitiatorToInitiatorGroup

Associates an initiator with an initiator group. If the initiator group is not present then this method attempts to create it. If the initiator does not exist, then the method fails.

Related APIs

initiatorGroupExists, *addInitiatorToInitiatorGroup*

Parameters

initiatorName

A string: the name of an initiator.

initiatorGroupName

A string: the name of an initiator group.

Returns

Returns an integer:

- (0) indicates that the request succeeded
- (1) indicates that the request was not attempted
- (2) indicates that the request failed

Examples

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '["iqn.2012-11.com.tegile.iscsi:api_initiator_1", \
```

```
  "iqn.2012-11.com.tegile.iscsi:testinigroup-group"]' \
https://198.51.100.10/zebi/api/v1/addInitiatorToInitiatorGroup -
k
```

Response

The above request returns the HTTP status code 200 (OK) and the following data:

```
 0
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '["iqn.2012-11.com.tegile.iscsi:api_initiator_2", \
 "iqn.2012-11.com.tegile.iscsi:testinigroup-group2"]' \
https://198.51.100.10/zebi/api/v1/addInitiatorToInitiatorGroup -
k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
  "message": "Failed to add initiator group member
 iqn.2012-11.com.tegile.iscsi:api_initiator_2 to
 iqn.2012-11.com.tegile.iscsi:testinigroup-group2 :
 STMF_ERROR_EXISTS",
  "extendedData": {},
  "details": "",
  "code": "EZEBI_GENERAL"
}
```

createMappingForVolume

Maps a volume to an initiator group and a target group.

Related APIs

deleteMappingFromVolume, *initiatorGroupExists*, *listISCSIInitiatorGroups*, *listISCSITargetGroups*, *createVolume*.

Parameters

datasetPath

A string: the dataset path of the volume. The dataset path has the format: `PoolName`/Local/`ProjectName`/`VolumeName`. You can get the datasetPath from the **`listVolumes`** API. For more information, see *listVolumes* and *Volume_V1_0.datasetPath*.

initiatorGroupName

A string: the name of the initiator group to which the volume must be mapped.

targetGroupName

A string: the name of the target group to which the volume must be mapped.

lunNumber

An integer: used as the LUN number for the newly defined LUN. To assign a LUN number automatically (default), use the value -1.

Returns

Returns an integer:

- (0) indicates that the request succeeded
- (1) indicates that the request was not attempted
- (2) indicates that the request failed

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '[ "pool1/Local/TechPubs/TechPubsLUN", \
 "api_InitiatorGroup", \
 "iqn.2014-11.com.tegile.iscsi:testtargetgroup-group", -1]' \
https://198.51.100.10/zebi/api/v1/createMappingForVolume -k
```

Response

```
 0
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '[ "pool1/Local/TechPubs/TechPubsLUN2", \
"api_InitiatorGroup", \
```

```
"iqn.2014-11.com.tegile.iscsi:testtargetgroup-group", -1]' \
https://198.51.100.10/zebi/api/v1/createMappingForVolume -k
```

Error Response

If the initiator group is not found, the above request returns the HTTP status code 200 (OK) and with no data.

deleteMappingFromVolume

Deletes the view (mapping) between the given volume, initiator group, and target group.

Related APIs

createMappingForVolume, *initiatorGroupExists*, *listISCSIInitiatorGroups*, *listISCSITargetGroups*, *listVolumes* on page 50.

Parameters

datasetPath

A string: the dataset path for the volume. The dataset path has the format: `PoolName`/Local/`ProjectName`/`VolumeName`. You can get the datasetPath from the **`listVolumes`** API. For more information, see *listVolumes* and *Volume_V1_0.datasetPath*.

initiatorGroupName

A string: the name of an initiator group.

targetGroupName

A string: the name of a target group.

Returns

Returns an integer:

- (0) indicates that the request succeeded
- (1) indicates that the request was not attempted
- (2) indicates that the request failed

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["pool1/Local/TechPubs/TechPubsLUN", \
  "api_InitiatorGroup", \
```

```
  "iqn.2014-11.com.tegile.iscsi:testtargetgroup-group"]' \
https://198.51.100.10/zebi/api/v1/deleteMappingFromVolume -k
```

Response

```
 0
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["pool1/Local/TechPubs2/TechPubsLUN", \
  "api_InitiatorGroup", \
  "iqn.2014-11.com.tegile.iscsi:testtargetgroup-group"]' \
https://198.51.100.10/zebi/api/v1/deleteMappingFromVolume -k
```

Error Response

```
HTTP Status Code: 500
{
 "message": "Unable to open pool1/Local/TechPubs2 : dataset does
 not exist",
 "extendedData": {
 "EX_CAUSE_CODE_NAME": "EZFS_NOENT",
 "EX_CAUSE_MESSAGE": "Unable to open pool1/Local/TechPubs2 :
 dataset does not exist",
 "EX_CAUSE_CODE_NUMBER": "2009"
 },
 "details": "Unable to open pool1/Local/TechPubs2 : dataset does
 not exist",
 "code": "EZEBI_RESOURCE_NOT_FOUND"
}
```

Chapter 7

Dataset Methods

Topics:

listPools

Lists all the pools on the Tegile array controller to which the API client (program or system) is connected. The controller to which the API client is connected may vary if the API client is using a floating IP address. This is an HTTP GET method.

Related APIs

listProjects, *listVolumes*, *listLunsByID*, *listShares*.

Parameters

None

Returns

Returns a JSON array of *Pool_V1_0* objects that contains details of all the pools on the controller.

Examples

Request (curl):

```
curl -X GET -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 https://198.51.100.10/zebi/api/v1/listPools -k
```

Response

```
[
  {
    "name": "pool1",
    "availableSize": 1960590146048,
    "totalSize": 1961726312448
  }
]
```

listProjects

Lists all the local or replicated projects in a pool.

Related APIs

listPools, *listVolumes*, *listLunsById*, *listShares*.

Parameters

poolName

A string: the name of the pool for which projects need to be listed.

local

A boolean: a **`true`** returns the local projects only; a **`false`** returns the replicated projects only.

Returns

Returns a JSON array of *Project_V1_0* objects that contains details of all the local or replicated projects in the specified pool.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["pool1",true]' \
 https://198.51.100.10/zebi/api/v1/listProjects -k
```

Response

```
[
  {
    "poolName": "pool1",
    "name": "TechPubs",
    "local": true
  }
]
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json -d '["pool11",true]' \
 https://198.51.100.10/zebi/api/v1/listProjects -k
```

Error Response

```
HTTP Status Code: 500
{
  "message": "Unable to open pool11/Local : dataset does not
 exist",
  "extendedData": {
        "EX_CAUSE_CODE_NAME": "EZFS_NOENT",
        "EX_CAUSE_MESSAGE": "Unable to open pool11/Local :
 dataset does not exist",
        "EX_CAUSE_CODE_NUMBER": "2009"
        },
  "details": "Unable to open pool11/Local : dataset does not
 exist",
```

```
  "code": "EZEBI_RESOURCE_NOT_FOUND"
}
```

listVolumes

Lists all the local or replicated volumes within a Project.

Related APIs

listPools, *listProjects*, *listLunsById*,*listShares*, *createVolume*.

Parameters

poolName

A string: the name of the pool that contains the project specified by the **`projectName`** parameter.

projectName

A string: the name of the project for which volumes need to be listed.

local

A boolean: a **`true`** returns the local volumes only; a **`false`** returns the replicated volumes only.

Returns

Returns a JSON array of *Volume_V1_0* objects that contains details of all the local or replicated volumes within the requested project.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["pool1","TechPubs", true]' \
 https://198.51.100.10/zebi/api/v1/listVolumes -k
```

Response

```
[
  {
    "poolName": "pool1",
    "projectName": "TechPubs",
    "name": "TechPubsLUN",
    "luId": "600144F012190100000052D92EC20155",
    "volSize": 1073741824,
```

```
    "blockSize": "32KB",
    "thinProvision": false,
    "protocol": "iSCSI",
    "datasetPath": "pool1/Local/TechPubs/TechPubsLUN",
    "local": true
  }
]
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["pool0","TechPubs", true]' \
 https://198.51.100.10/zebi/api/v1/listVolumes -k
```

Error Response

```
HTTP Status Code: 500
{
  "message": "Unable to open pool0/Local/TechPubs : dataset does
 not exist",
  "extendedData": {
        "EX_CAUSE_CODE_NAME": "EZFS_NOENT",
        "EX_CAUSE_MESSAGE": "Unable to open pool0/Local/
TechPubs : dataset does not exist",
        "EX_CAUSE_CODE_NUMBER": "2009"
        },
  "details": "Unable to open pool0/Local/TechPubs : dataset does
 not exist",
  "code": "EZEBI_RESOURCE_NOT_FOUND"
}
```

listLunsById

Lists the LUNs for the requested LUN IDs (GUIDs).

Related APIs

listPools, *listProjects*, *listShares*, *createVolume*, *listVolumes*.

Parameters

lunid

An array of LUN GUIDs, such as:

```
["600144F050488F00000055CB17BF001F",
 "600144F050488F00000055CB17D60020"]
```

Returns

Returns a JSON array of *LunStatus* objects that contains details of all the LUNs requested.

Example

Request (curl):

```
curl -X POST -H Authorization:'Basic YWRtaW46dA==' \
-H Content-Type:application/json \
-d '[["600144F050488F00000055CB17BF001F",
 "600144F050488F00000055CB17D60020"]]' \
https://198.51.100.10/zebi/api/v1/listLunsById -k
```

Response


```
[
{
"commandStatus": 0,
"commandException": null,
"guid": "600144F050488F00000055CB17BF001F",
"alias": "/dev/zvol/rdsk/pool-a/Local/test/lun1",
"dataFile": "/dev/zvol/rdsk/pool-a/Local/test/lun1",
"metaFile": null,
"vendorId": null,
"productId": null,
"mgmtURL": "",
"serialNumber": null,
"viewCount": 1,
"size": "",
"blockSize": null,
"writeProtect": false,
"writeCacheDisable": false,
"operationalStatus": 2,
"accessState": 0,
"datasetPath": "pool-a/Local/test/lun1"
},
{
"commandStatus": 0,
"commandException": null,
"guid": "600144F050488F00000055CB17D60020",
"alias": "/dev/zvol/rdsk/pool-a/Local/test/lun3",
"dataFile": "/dev/zvol/rdsk/pool-a/Local/test/lun3",
"metaFile": null,
"vendorId": null,
"productId": null,
"mgmtURL": "",
"serialNumber": null,
"viewCount": 1,
"size": "",
"blockSize": null,
"writeProtect": false,
"writeCacheDisable": false,
"operationalStatus": 2,
"accessState": 0,
"datasetPath": "pool-a/Local/test/lun3"
}
```

```
]
```

listShares

Lists all the local or replicated shares in a project.

Related APIs

listPools, *listProjects*, *listVolumes*, *listLunsById*, *createShare*, *createShare*.

Parameters

poolName

A string: the name of the pool that contains the project specified by the projectName parameter.

projectName

A string: the name of the project for which shares need to be listed.

local

A boolean: a **`true`** returns the local shares only; a **`false`** returns the replicated shares only.

Returns

Returns a JSON array of *Share_V1_0* objects that contains details of all the local or replicated shares in the specified pool and project.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["p1aut","project2", true]' \
 https://198.51.100.10/zebi/api/v1/listShares -k
```

Response

```
[
  {
    "poolName": "p1aut",
    "projectName": "project2",
    "name": "default_share",
    "availableSize": 9275971622400,
    "totalSize": 9275971769856,
    "datasetPath": "p1aut/Local/project2/default_share",
```

```
    "mountpoint": /export/p1aut/project2/default_share,
    "local": true
  }
]
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["pool0","TechPubs", true]' \
 https://198.51.100.10/zebi/api/v1/listShares -k
```

Error Response

```
HTTP Status Code: 500
{
  "message": "Unable to open pool0/Local/TechPubs : dataset does
 not exist",
  "extendedData": {
      "EX_CAUSE_CODE_NAME": "EZFS_NOENT",
      "EX_CAUSE_MESSAGE": "Unable to open pool0/Local/TechPubs :
 dataset does not exist",
      "EX_CAUSE_CODE_NUMBER": "2009"
      },
  "details": "Unable to open pool0/Local/TechPubs : dataset does
 not exist",
  "code": "EZEBI_RESOURCE_NOT_FOUND"
}
```

createShare

Creates a share with the specified share options and share permissions.

Related APIs

listShares, *createShare*, *deleteShare*.

Parameters

poolName

A string : the name of the pool in which the share is created.

projectName

A string: the name of the project in which the share is created. The characters ,, /,\\, !, ?, @, <, >, #, $, ',%, ^,*,(,), ~,+, =, },|, :, {, [,], ;, \', \", & are not allowed in projectname. The empty and space characters and the null values are not allowed in projectname.

shareName

A string: the share name.

shareOptions

A *ShareOptions* object that specifies the mount point, block size, quota, and reservation. This parameter is optional. If some of the settings included in this parameter are not specified, the defaults are as follows:

- If Block Size is null or is an empty string ("") then the block size of the new share is set to 32KB and the override record (Block) size flag is set to false.
- If the Mount Point is not specified or is an empty string ("") then the default mountpoint is used and the override mountpoint flag is set to false.
- If the Quota and Reservation are not specified or is set to "-1" then no quota or reservation is applied to the new share.

sharePermissions

A *SharePermissions* object that contains ACLs and defines how those ACLs are applied to the new share.

Returns

Returns an integer: the number 0, if the request succeeds.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["poolName", "projectName", "newShareName",  \
    {                                             \
        "blockSize": "64KB",                      \
        "quota": -1,                              \
        "reservation": -1                         \
    },                                            \
    [{                                            \
        "sharePermissionEnum": "GROUP",           \
        "sharePermissionMode": "ALLOW",           \
        "groupList": [{                           \
            "groupName": "newGroupName",          \
            "groupId": 104                        \
        }]                                        \
    }]                                            \
]                                                 \
 https://198.51.100.10/zebi/api/v1/createShare -k
```

Response

HTTP Status Code: 200

```
 0
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["BadPoolName", "projectName", "newShare2",  \
    {                                             \
        "blockSize": "64KB",                      \
        "quota": -1,                              \
        "reservation": -1                         \
    },                                            \
    [{                                            \
        "sharePermissionEnum": "GROUP",           \
        "sharePermissionMode": "ALLOW",           \
        "groupList": [{                           \
            "groupName": "newGroupName",          \
            "groupId": 104                        \
        }]                                        \
    }]                                            \
]                                                 \
 https://198.51.100.10/zebi/api/v1/createShare -k
```

Error Response

HTTP Status Code: 400

```
{
      "message":   "Error while saving: shareName.
         Reason: Unable to open BadPoolName/Local/projectName:
         dataset does not exist",
      "extendedData": { },
      "details":   "",
      "code":   "EZEBI_GENERAL"
}
```

createShare

Creates a share with the default share properties (A block size of 32 KB; no quota; no reservation).

Related APIs

listShares, *createShare*, *deleteShare*.

Parameters

poolName

A string: the name of the pool in which the share is created.

projectName

A string: the name of the project in which the share is created. The characters ,, /,\\, !, ?, @, <, >, #, $, ',%, ^,*,(,), ~,+, =, },|, :, {, [,], ;, \', \", & are not allowed in projectname. The empty and space characters and the null values are not allowed in projectname.

shareName

A string: the share name.

sharePermissions

A *SharePermissions* object that defines permissions for the new share using ACLs.

Returns

An integer: The number 0 if the request succeeds.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["poolName", "projectName", "shareName",      \
      [                                            \
        {"sharePermissionEnum": "GROUP", "sharePermissionMode":
 "ALLOW",     \
           "groupList": [                          \
            {"groupName": "newAPIGroup", "groupId": 104}
             \
            ]                                      \
        }                                          \
      ]                                            \
 https://198.51.100.10/zebi/api/v1/createShare -k
```

Response

HTTP Status Code: 200

```
 0
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["BadPoolName", "projectName", "shareName",      \
      [                                            \
        {"sharePermissionEnum": "GROUP", "sharePermissionMode":
 "ALLOW",     \
```

```
            "groupList": [                                  \
             {"groupName": "newAPIGroup", "groupId": 104}
              \
             ]                                              \
          }                                                 \
       ]                                                    \
 https://198.51.100.10/zebi/api/v1/createShare -k
```

Error Response

HTTP Status Code: 400

```
{
      "message":   "Error while saving: shareName.
         Reason: Unable to open BadPoolName/Local/projectName:
         dataset does not exist",
      "extendedData": { },
      "details":   "",
      "code":   "EZEBI_GENERAL"
}
```

createVolume

Creates a volume with the specified settings.

Related APIs

initiatorGroupExists, *addInitiatorToInitiatorGroup*

Parameters

volume

A JSON object of type *Volume_V1_0* that contains the parameters required to create the volume.

inheritSANViewSettingsFromProject

A boolean value: indicates whether to copy the view settings related to the intended protocol (iSCSI or FC) from the project. The default views created on the project are copied over if this parameter is true. If this parameter is false, then the volume is created with no views attached to it.

Returns

Returns an integer:

- (0) indicates that the request succeeded
- (1) indicates that the request was not attempted
- (2) indicates that the request failed

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '[       \
      {      \
        "blockSize":"32KB",                          \
        "datasetPath":"pool1/Local/TechPubs",     \
        "local":true,                                \
        "name":"api_createVolume_name",           \
        "poolName":"pool1",                          \
        "projectName":"TechPubs",                    \
        "protocol":"iSCSI",                          \
        "thinProvision":"true",                      \
        "volSize":"3276800000"},                     \
      true]' \
 https://198.51.100.10/zebi/api/v1/createVolume -k
```

Response

```
0
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '[         \
        {      \
          "blockSize":"32KB",                                  \
          "datasetPath":"pool1/Local/TechPubsBadPath",     \
          "local":true,                                        \
          "name":"api_createVolume_name",                   \
          "poolName":"pool1",                                  \
          "projectName":"TechPubs",                            \
          "protocol":"iSCSI",                                  \
          "thinProvision":"true",                              \
          "volSize":"3276800000"},                             \
    true]'                                                     \
 https://198.51.100.10/zebi/api/v1/createVolume -k
```

Error Response

```
HTTP Status Code: 400
{
  "message": "A volume/share with the same name pool1/Local/
TechPubs/api_createVolume_name already exists.",
  "extendedData": {},
  "details": "",
  "code": "EZEBI_GENERAL"
```

```
}
```

deleteDataset

Deletes the specified dataset.

Caution: If the **`recursive`** parameter is set to **`true`** all dependent objects are deleted. For example, if the **`datasetPath`** points to a project all shares and LUNs in the project, and their snapshots and clones are deleted.

Related APIs

listVolumes.

Parameters

datasetPath

A string: the path to the dataset. The dataset path has the format: `PoolName`/Local/`ProjectName`/`VolumeName`.

recursive

A boolean value: indicates whether the dependents (for example clones of the dataset) of this dataset should be removed (if true) before trying to delete the dataset or not (if false). Deletion might fail if the dataset has dependents.

errorIfNotExist

A boolean value: indicates whether to raise (if true) an exception if the path specified by datasetPath does not exist.

Returns

Returns no data.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
  -H Content-Type:application/json \
  -d '["pool1/Local/TechPubs/TechPubsTest",true, true]' \
  https://198.51.100.10/zebi/api/v1/deleteDataset -k
```

Response

On success, the above request returns the HTTP status code 200 (OK) and with no data.

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
  -H Content-Type:application/ \
  -d '["pool1/Local/TechPubs/api_createVolume_name", \
    false, false]' \
  https://198.51.100.10/zebi/api/v1/deleteDataset -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
  "message": "Unable to delete pool1/Local/TechPubs/
api_createVolume_name",
  "extendedData": {
      "EX_CAUSE_CODE_NAME": "EZFS_BUSY",
      "EX_CAUSE_MESSAGE": "dataset is busy",
      "EX_CAUSE_CODE_NUMBER": "2007"
      },
  "details": "dataset is busy",
  "code": "EZEBI_GENERAL"
}
```

deleteShare

Deletes the specified share, and optionally, any dependents of the share.

Caution: If the **`recursive`** parameter is set to **`true`**, all dependent objects (snapshots and clones of the given share) are also deleted.

Related APIs

listShares, *createShare*, *createShare*.

Parameters

datasetPath

A string: the path which uniquely identifies the share. The dataset path has the format: `PoolName`/Local/`ProjectName`/`ShareName`.

recursive

A boolean: a **`true`** specifies that dependents of the share should be deleted before deleting the share or not (**`false`**)

errorIfNotExist

A boolean value: that specifies if an exception is raised (if true) if the given dataset path does not exist or not (if false).

Returns

Returns no data.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["Pool1/Local/Project1/Share1", false, false]' \
 https://198.51.100.10/zebi/api/v1/deleteShare -k
```

Response

On success, the above request returns the HTTP status code 200 (OK) and no data.

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["test/Local/KKKK/NoSuchShare", false, false]' \
 https://198.51.100.10/zebi/api/v1/deleteShare -k
```

Error Response

The above request returns the HTTP status code 500 (Internal Server Error) and the following message:

```
{
"message":"Unable to open test/Local/KKKK : dataset does not
 exist",
"extendedData":
    {"EX_CAUSE_CODE_NAME":"EZFS_NOENT",
      "EX_CAUSE_MESSAGE":"Unable to open test/Local/KKKK :
 dataset does not exist",
      "EX_CAUSE_CODE_NUMBER":"2009"},
"details":"Unable to open test/Local/KKKK : dataset does not
 exist",
"code":"EZEBI_RESOURCE_NOT_FOUND"
}
```

deleteVolume

Deletes the specified volume, and optionally, any dependents of the volume.

> **Caution:** If the **`recursive`** parameter is set to **`true`**, all dependent objects (snapshots and clones of the given volume) are also deleted.

Related APIs

listVolumes,*createVolume*.

Parameters

datasetPath

A string: the path which uniquely identifies the volume on the Tegile array controller. The dataset path has the format: `PoolName`/Local/`ProjectName`/`VolumeName`. You can get the datasetPath from the **`listVolumes`** API. For more information, see *listVolumes* and *Volume_V1_0.datasetPath*.

recursive

A boolean: indicates whether the dependents (for example, clones of the dataset) of the dataset should be removed (if true) before trying to delete the dataset. This API fails if you try to delete a volume that has dependents and the recursive parameter is set to **`false`**.

errorIfNotExist

A boolean value: indicates whether to raise (if true) an exception if the path specified by the *datasetPath* parameter does not exist.

Returns

Returns no data.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["pool1/Local/TechPubs/api_createVolume_name_2", \
    true,true]' \
 https://198.51.100.10/zebi/api/v1/deleteVolume -k
```

Response

The above request returns the HTTP status code 200 (OK) and with no data.

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["pool1/Local/TechPubs/api_createVolume_", \
   true,true]' \
 https://198.51.100.10/zebi/api/v1/deleteVolume -k
```

Error Response

The above request returns the HTTP status code 500 (Internal Server Error) and the following message:

```
{
  "message": "Unable to delete pool1/Local/TechPubs/
api_createVolume_ because it does not exist",
  "extendedData": {},
  "details": "",
  "code": "EZEBI_RESOURCE_NOT_FOUND"
}
```

Chapter 8

Snapshot Methods

Topics:

listSnapshots

Lists names of snapshots (from the specified dataset) that match with the given regex pattern.

Parameters

datasetPath

A string that contains the dataset path. The dataset path has the format: `PoolName`/Local/`ProjectName`/`VolumeName`. You can get the datasetPath from the **`listVolumes`** API. For more information, see *listVolumes* and *Volume_V1_0.datasetPath*.

snapshotPattern

A string that contains a regex pattern for matching snapshot names. Use an empty string to list all snapshots.

Returns

A JSON array of strings that contains names of snapshots (from the specified dataset) that match with the given regex pattern.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["pool1/Local/TechPubs/TechPubsLUN",".*"]' \
 https://198.51.100.10/zebi/api/v1/listSnapshots -k
```

Response

The above request returns the HTTP status code 200 (OK) and the following data:

```
[
    "Auto-LF-Day-011714-21:15",
    "Auto-LF-Day-011814-21:15",
    "Auto-LF-Day-011914-21:15",
    "Auto-LF-Week-011914-21:30",
    "Auto-LF-Day-012014-21:15",
    "Auto-LF-Day-012114-21:15"
]
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H Content-Type:application/json \
 -d '["pool1/Local/TechPubs/TechPubs",""]' \
 https://198.51.100.10/zebi/api/v1/listSnapshots -k
```

Error Response

```
HTTP Status Code: 500
{
  "message": "Unable to open pool1/Local/TechPubs/TechPubs :
 dataset does not exist",
  "extendedData": {
      "EX_CAUSE_CODE_NAME": "EZFS_NOENT",
      "EX_CAUSE_MESSAGE": "Unable to open pool1/Local/TechPubs/
TechPubs : dataset does not exist",
      "EX_CAUSE_CODE_NUMBER": "2009"
       },
  "details": "Unable to open pool1/Local/TechPubs/TechPubs :
 dataset does not exist",
  "code": "EZEBI_RESOURCE_NOT_FOUND"
}
```

createProjectSnapshot

Recursively creates snapshots of the specified project and the datasets within the project. The string "Manual-P-" is prefixed to the names of the snapshots created.

Related APIs

getProjectSnapshotCreationStatus, *listSnapshots*, *createVolumeSnapshot*, *createShareSnapshot*, , *deleteProjectSnapshot*.

Parameters

project

A *Project_V1_2* object that specifies the project for which the snapshots are created.

snapshotName

Name for the new snapshots that are created. The characters ,, /,\\, !, ?, @, <, >, #, $, ',%, ^,*,(,), ~,+, =, },|, :, {, [,], ;, \', \", & are not allowed in snapshotName. The empty and space characters and the null values are not allowed in snapshotName.

quiesce

A boolean that specifies whether the snapshot is quiesced or not.

Returns

No Data.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H "Content-Type:application/json" \
 -d '[{"name": "sProj", "local": true, \
  "poolName": "pool1"}, "NewTPSS9", false]' \
 https://198.51.100.10/zebi/api/v1/createProjectSnapshot -k
```

Response

The above request returns the HTTP status code 200 (OK) and with no data.

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H "Content-Type:application/json" \
 -d '[{"name": "sProj", "local": true, \
  "poolName": "NotExistantPool"}, "NewTPSS9", false]' \
 https://198.51.100.10/zebi/api/v1/createProjectSnapshot -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
 message:   "Unable to open NotExistantPool/Local/sProj: dataset
 does not exist."
 extendedData: { }
 details:   ""
 code:   "EZEBI_GENERAL"
}
```

createVolumeSnapshot

Recursively creates snapshot of the specified volume. The string "Manual-V-" is prefixed to the names of the snapshots created.

Related APIs

getProjectSnapshotCreationStatus, *listSnapshots*, *createProjectSnapshot*, *createShareSnapshot*, *deleteProjectSnapshot*.

Parameters

volume

A *Volume_V1_0* object for which snapshot needs to be created.

snapshotName

Name for the new snapshots that are created. The characters ,, /,\\, !, ?, @, <, >, #, $, ',%, ^,*,(,), ~,+, =, },|, :, {, [,], ;, \', \", & are not allowed in snapshotName. The empty and space characters and the null values are not allowed in snapshotName.

quiesce

A boolean value that specifies whether the snapshots are quiesced or not.

Returns

No Data.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN"
 -H "Content-Type:application/json" \
 -d '[{"poolName": "pool1", \
  "projectName": "vProj", "name": "vol2", \
  "luId": "600144F0A6308900000053BD51250002", \
  "volSize": 161061273600, "blockSize": "32KB", \
  "thinProvision": false, "protocol": "FC", \
  "datasetPath": "pool1/Local/vProj/vol2", \
  "local": true}, "vProj_S3", false]' \
 https://198.51.100.10/zebi/api/v1/createVolumeSnapshot -k
```

Response

The above request returns the HTTP status code 200 (OK) and with no data.

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN"
 -H "Content-Type:application/json" \
 -d '[{"poolName": "NotaPool", \
       "projectName": "vProj", "name": "vol2", \
       "luId": "600144F0A6308900000053BD51250002", \
       "volSize": 161061273600, "blockSize": "32KB", \
       "thinProvision": false, "protocol": "FC", \
       "datasetPath": "NotaPool/Local/vProj/vol2", \
       "local": true}, "vProj_S3", false]' \
  https://198.51.100.10/zebi/api/v1/createVolumeSnapshot -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
 message:   "Unable to open NotaPool/Local/vProj/vol2: \
             dataset does not exist."
 extendedData: { }
 details:   ""
 code:   "EZEBI_GENERAL"
}
```

createShareSnapshot

Recursively creates snapshot of the specified share. The string "Manual-S-" is prefixed to names of the snapshots created.

Related APIs

getShareSnapshotCreationStatus, *listSnapshots*, *deleteShareSnapshot*, *cloneShareSnapshot*.

Parameters

share

The *Share_V1_0* object that specifies the share for which the snapshots are created.

snapshotName

Name for the new snapshots that are created. The characters ,, /,\\, !, ?, @, <, >, #, $, ',%, ^,*,(,), ~,+, =, },|, :, {, [,], ;, \', \", & are not allowed in snapshotName. The empty and space characters and the null values are not allowed in snapshotName.

quiesce

A boolean value that specifies whether the snapshots are quiesced or not.

Returns

No Data.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
```

```
-d '[{"poolName": "pool1", "projectName": "sProj",\
"name": "TP_Check-newclone", "availableSize": 0, \
"totalSize": 7794361020176, \
"datasetPath": "pool1/Local/sProj/TP_Check-newclone", \
"mountpoint": null, "local": true }, \
"NewShareSnapShot", false]' \
https://198.51.100.10/zebi/api/v1/createShareSnapshot  -k
```

Response

The above request returns the HTTP status code 200 (OK) and with no data.

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '[{"poolName": "pool1", "projectName": "sProj", \
"name": "TP_Check-newclone", "availableSize": 0, \
"totalSize": 7794361020176, \
"datasetPath": "pool1/Local/sProj/TP_Check-newclone", \
"mountpoint": null, "local": true }, \
"NewShareSnapShot", false]' \
https://198.51.100.10/zebi/api/v1/createShareSnapshot -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
{ message: "Unable to open NotAPool/Local/sProj/TP_s1:
    dataset does not exist."
  extendedData: { }
  details:   ""
  code:   "EZEBI_GENERAL"}
```

getProjectSnapshotCreationStatus

Gets the status of a project snapshot creation request.

Related APIs

createProjectSnapshot.

Parameters

dataSetPath

Dataset path of the project. The dataset path has the format: `PoolName`/Local/`ProjectName`. You can get the datasetPath from the **`listProjects`** API. For more information, see *listProjects*.

snapshotName

Name of the project snapshot for which status is required. You must use the name that you specified while invoking the *createProjectSnapshot* API, because this API prefixes the string "Manual-P-" to the name before getting the status.

Returns

A JSON object of type *SnapshotProgressStatus*.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H "Content-Type:application/json" \
 -d '["pool1/Local/sProj", "Manual-P-NewTPSS111"]' \
https://198.51.100.10/zebi/api/v1/\
getProjectSnapshotCreationStatus -k
```

Response

The above request returns the HTTP status code 200 (OK) and the following data:

```
[
  {
    SnapshotProgressStatus: 0
  }
]
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H "Content-Type:application/json" \
 -d '["NotAPool/Local/sProj", ""]' \
 https://198.51.100.10/zebi/api/v1/\
getProjectSnapshotCreationStatus -k
```

Error Response

The above request returns the HTTP status code 200 (OK) and the following data:

```
{
  snapshotProgressStatus:   2
}
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H "Content-Type:application/json" \
 -d '["NotAPool/Local/sProj", "Manual-P-NewTPSS111"]' \
 https://198.51.100.10/zebi/api/v1/\
getProjectSnapshotCreationStatus -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
  message:   "Unable to open pool1/Local/sProj2: dataset does
 not exist."
  extendedData: { }
  details:   ""
  code:   "EZEBI_GENERAL"
}
```

getVolumeSnapshotCreationStatus

Gets the status of a volume snapshot creation request.

Related APIs

createVolumeSnapshot.

Parameters

dataSetPath

Dataset path of the volume. The dataset path has the format: `PoolName`/Local/`ProjectName`/`VolumeName`. You can get the datasetPath from the **`listVolumes`** API. For more information, see *listVolumes* and *Volume_V1_0.datasetPath*.

snapshotName

Name of the volume snapshot for which status is required. You must use the name that you specified while invoking the *createVolumeSnapshot* API, because this API prefixes the string "Manual-V-" to the name before getting the status.

Returns

A JSON object of type *SnapshotProgressStatus*.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H "Content-Type:application/json" \
 -d '["pool1/Local/vProj/vol2", \
"vProj_S3"]' \
https://198.51.100.10/zebi/api/v1/\
getVolumeSnapshotCreationStatus -k
```

Response

The above request returns the HTTP status code 200 (OK) and the following data.

```
[
  {
    SnapshotProgressStatus: 0
  }
]
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '["pool1/Local/vProj/vol2", \
"vProj_S"]' \
https://198.51.100.10/zebi/api/v1/\
getVolumeSnapshotCreationStatus -k
```

Error Response

The above request returns the HTTP status code 200 (OK) and the following data:

```
{snapshotProgressStatus:   2}
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '["NoPool/Local/vProj/vol2", \
"vProj_S3"]' \
https://198.51.100.10/zebi/api/v1/\
getVolumeSnapshotCreationStatus -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
  message:   "Unable to open NoPool/Local/vProj/vol2: dataset
 does not exist."
  extendedData: { }
  details:   ""
  code:   "EZEBI_GENERAL"
}
```

getShareSnapshotCreationStatus

Gets the status of a share snapshot creation request.

Related APIs

createShareSnapshot.

Parameters

dataSetPath

Dataset path of the share. The dataset path has the format: `PoolName`/Local/`ProjectName`/`ShareName`. You can get the datasetPath from the **`listShares`** API. For more information, see *listShares* and *Share_V1_0.datasetPath*.

snapshotName

Name of the share snapshot for which status is required. You must use the name that you specified while invoking the *createShareSnapshot* API, because this API prefixes the string "Manual-S-" to the name before getting the status.

Returns

A JSON object of type *SnapshotProgressStatus*.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H "Content-Type:application/json" \
 -d '["pool1/Local/sProj/TP_Check-newclone", \
"NewShareSnapShot"]' \
 https://198.51.100.10/zebi/api/v1/\
getShareSnapshotCreationStatus -k
```

Response

The above request returns the HTTP status code 200 (OK) and the following data:

```
[
  {
    SnapshotProgressStatus: 0
  }
]
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H "Content-Type:application/json" \
 -d '["pool1/Local/sProj/TP_Check", \
"NotASnapShot"]' \
 https://198.51.100.10/zebi/api/v1/\
getShareSnapshotCreationStatus -k
```

Error Response

The above request returns the HTTP status code 200 (OK) and the following data:

```
[
  {
    SnapshotProgressStatus: 2
  }
]
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H "Content-Type:application/json" \
 -d '["NotAPool/Local/sProj/TP_Check", \
"NotASnapShot"]'  \
 https://198.51.100.10/zebi/api/v1/\
getShareSnapshotCreationStatus -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
  message:   "Unable to open NotAPool/Local/sProj/TP_Check:
 dataset does not exist."
  extendedData: { }
  details:   ""
```

```
  code:    "EZEBI_GENERAL"
}
```

cloneProjectSnapshot

Clones the specified project-level snapshot. This creates new datasets at the share and volume levels for all shares and volumes that have a snapshot with the specified name.

Related APIs

getProjectCloneStatus, *createProjectSnapshot*, *deleteProjectSnapshot*.

Parameters

snapshotPath

Path of the project-level snapshot that has to be cloned. The snapshot path has the format: `PoolName`/Local/`ProjectName`@`SnapshotName`. You can get the snapshotPath from the **`listSnapshots`** API. For more information, see *listSnapshots*.

cloneName

A string that is used to create names of the new datasets. The clone name is appended to the resultant share and volume names. The characters ,, /, \\, !, ?, @, <, >, #, $, ',%, ^,*,(,), ~,+, =, }, |, :, {, [,], ;, \', \", & are not allowed in clonename. The empty and space characters and the null values are not allowed in clonename.

inheritProjectSettings

A boolean value that indicates whether the new dataset will inherit project settings.

Returns

No Data.

Examples

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H "Content-Type:application/json" \
 -d '["pool1/Local/sProj@Manual-P-NewTPSS", \
 "mkclone", false]' \
 https://198.51.100.10/zebi/api/v1/cloneProjectSnapshot -k
```

Response

The above request returns the HTTP status code 200 (OK) and with no data.

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H "Content-Type:application/json" \
 -d '["pool1/Local/sProj@NewTPSS", \
  "mkclone2", false]' \
 https://198.51.100.10/zebi/api/v1/cloneProjectSnapshot -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
  message:   "Unable to open pool1/Local/sProj@NewTPSS: dataset
 does not exist."
  extendedData: { }
  details:   ""
  code:   "EZEBI_GENERAL"
}
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H "Content-Type:application/json" \
 -d '["pool1/Local/sProj@Manual-P-NewTPSS", \
 "", false]' \
 https://198.51.100.10/zebi/api/v1/cloneProjectSnapshot -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
  message:   "Clone name is not valid."
  extendedData: { }
  details:   ""
  code:   "EZEBI_GENERAL"
}
```

cloneVolumeSnapshot

Clones the specified snapshot of a volume.

Related APIs

createVolumeSnapshot, *deleteVolumeSnapshot*.

Parameters

snapshotPath

The snapshot path of the volume dataset to be cloned. The snapshot path has the format: `PoolName`/Local/`ProjectName`/`VolumeName`@`SnapshotName`. You can get the snapshotPath from the **`listSnapshots`** API. For more information, see *listSnapshots*.

cloneName

A string that is used to create the name of the new dataset. The clone name is appended to the resultant volume name. The characters ,, /,\\, !, ?, @, <, >, #, $, ',%, ^,*,(,), ~,+, =, }, |, :, {, [,], ;, \', \", & are not allowed in clonename. The empty and space characters and the null values are not allowed in clonename.

inheritViewsFromVolume

A boolean value that indicates whether the new dataset will inherit views from the volume.

inheritViewsFromProject

A boolean value that indicates whether the new dataset will inherit views from the project.

Returns

No Data.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '["pool1/Local/vProj/vol2\
@Manual-V-vProj_S3", "mkclone2", false, false]' \
https://198.51.100.10/zebi/api/v1/cloneVolumeSnapshot -k
```

Response

The above request returns the HTTP status code 200 (OK) and with no data.

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H "Content-Type:application/json" \
 -d '["pool1/Local/vProj/vol2\
@vProj_S3","mkclone2", false]' \
https://198.51.100.10/zebi/api/v1/cloneVolumeSnapshot -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
  message:   "No such method found in v1. Please check if method
 name and arguments are correct."
  extendedData: {
              EX_CAUSE_MESSAGE:   "No such method found in v1.
                 Please check if method name and arguments are
 correct."
               }
  details:   "No such method found in v1.
             Please check if method name and arguments are
 correct."
  code:   "EZEBI_GENERAL"
}
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '["pool1/Local/vProj/vol2@vProj","mkclone2", false]' \
https://198.51.100.10/zebi/api/v1/cloneVolumeSnapshot -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
  message:   "Unknown error cloning volume."
  extendedData: { }
  details:   ""
  code:   "EZEBI_GENERAL"
}
```

cloneShareSnapshot

Clones the specified share-level snapshot.

Related APIs

createShareSnapshot, *deleteShareSnapshot*.

Parameters

snapshotPath

Path of the share-level snapshot that has to be cloned. The snapshot path has the format: `PoolName`/Local/`ProjectName`/`ShareName`@`SnapshotName`. You can get the snapshotPath from the **`listSnapshots`** API. For more information, see *listSnapshots*.

cloneName

A string that is used to create the name of the new dataset. The clone name is appended to the resultant share name. The characters ,, /,\\, !, ?, @, <, >, #, $, ', %, ^,*,(,), ~,+, =, }, |, :, {, [,], ;, \', \", & are not allowed in clonename. The empty and space characters and the null values are not allowed in clonename.

inheritShareSettings

A boolean value that indicates whether the new dataset will inherit the share settings.

Returns

No Data.

Examples

Request (curl):

```
  curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '["pool1/Local/sProj/TP_Check@Manual-P-NewTPSS",\
"mk32", false]' \
https://198.51.100.10/zebi/api/v1/cloneShareSnapshot -k
```

Response

The above request returns the HTTP status code 200 (OK) and with no data.

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '["pool1/Local/sProj/TP_Check/Manual-P-NewTPSS", \
"mk11", false]'
https://198.51.100.10/zebi/api/v1/cloneShareSnapshot -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
  message:   "Error while cloning: pool1/Local/sProj/
TP_Check@null.
              Reason: Unable to open pool1/Local/sProj/mk11 :
 dataset does not exist"
  extendedData: { }
  details:   ""
  code:   "EZEBI_GENERAL"
}
```

getProjectCloneStatus

Gets the status of a clone request on the specified project snapshot.

Related APIs

cloneProjectSnapshot.

Parameters

snapshotPath

Path to a project snapshot. The snapshot path has the format: `PoolName`/Local/`ProjectName`@`SnapshotName`. You can get the snapshotPath from the **`listSnapshots`** API. For more information, see *listSnapshots*.

cloneName

Name of the new dataset.

Returns

A JSON object of type *ProjectCloneProgressStatus_v1_2*.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
```

```
-d '["pool1/Local/sProj@Manual-P-NewTPSS", \
"mkclone"]' \
https://198.51.100.10/zebi/api/v1/getProjectCloneStatus -k
```

Response

The above request returns the HTTP status code 200 (OK) and the following data:

```
{
  failedSubProjects:   1
  totalSubProjects:   6
  projectCloneState:   3
}
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '["pool1/Local/sProj@NewTPSS","mkclone"]' \
https://198.51.100.10/zebi/api/v1/getProjectCloneStatus -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
  message:   "Unable to open pool1/Local/sProj@NewTPSS: dataset
 does not exist."
  extendedData: { }
  details:   ""
  code:   "EZEBI_GENERAL"
}
```

deleteProjectSnapshot

Deletes the specified project snapshot.

Caution: If the **`recursive`** parameter is set to **`true`**, all dependent objects (snapshots and clones of the specified project snapshot) are also deleted.

Related APIs

createProjectSnapshot.

Parameters

projectSnapshotPath

Dataset path of the project snapshot. The dataset path of a project snapshot has the following format: `PoolName`/Local/`ProjectName`@`SnapshotName`. You can get the snapshotPath from the **`listSnapshots`** API. For more information, see *listSnapshots*.

recursive

A boolean value that specifies whether dependants of the snapshot are deleted before the snapshot is deleted.

Returns

A JSON object of type *SnapShotDeletionStatus* that contains information about the snapshot deletion status.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H "Content-Type:application/json" \
 -d '["pool1/Local/sProj@Manual-P-NewTPSS2", \
  true]' \
 https://198.51.100.10/zebi/api/v1/deleteProjectSnapshot  -k
```

Response

The above request returns the HTTP status code 200 (OK) and the following data:

```
{
snapshotDeletionStatus:   0
  deletedList: [
        "pool1/Local/sProj@Manual-P-NewTPSS9"
        "pool1/Local/sProj/TP_Check@Manual-P-NewTPSS9"
        "pool1/Local/sProj/TP_Check-newclone@Manual-P-NewTPSS9"
        "pool1/Local/sProj/TP_NFS_Share@Manual-P-NewTPSS9"
        "pool1/Local/sProj/TP_NFS_Share-newclone@Manual-P-
NewTPSS9"
        "pool1/Local/sProj/manus-pc-backup@Manual-P-NewTPSS9"
        "pool1/Local/sProj/newShareClone@Manual-P-NewTPSS9"
    ]
    failedToDeleteList: [ ]
}
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H "Content-Type:application/json" \
 -d '["pool1/Local/sProj@NewTPSS2", true]' \
```

```
 https://198.51.100.10/zebi/api/v1/deleteProjectSnapshot -k
```

Error Response

The above request returns the HTTP status code 200 (OK) and the following data:

```
{
snapshotDeletionStatus:   2
deletedList: [ ]
failedToDeleteList: ["pool1/Local/sProj@NewTPSS2"]
}
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
 -H "Content-Type:application/json" \
 -d '["pool1/Local/sProj/Manual-P-NewTPSS2", \
   true]' \
 https://198.51.100.10/zebi/api/v1/deleteProjectSnapshot -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
  message:   "Unable to open NotAPool/Local/sProj: dataset does
 not exist."
  extendedData: { }
  details:   ""
  code:   "EZEBI_GENERAL"
}
```

deleteVolumeSnapshot

Deletes the specified volume snapshot.

Caution: If the **`recursive`** parameter is set to **`true`**, all dependent objects (snapshots and clones of the specified volume snapshot) are also deleted.

Related APIs

createVolumeSnapshot

Parameters

volumeSnapshotPath

Dataset path of the volume snapshot. The dataset path of a volume snapshot has the following format: `PoolName`/Local/`ProjectName`/`VolumeName`@`SnapshotName`. You can get the snapshotPath from the **`listSnapshots`** API. For more information, see *listSnapshots*.

recursive

A boolean value that specifies whether dependents of the snapshot are deleted before deleting the snapshot.

Returns

A JSON object of type *SnapShotDeletionStatus*.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '[{"pool1/Local/proj1/vol1@Manual-V-snap1", false}]' \
https://198.51.100.10/zebi/api/v1/deleteVolumeSnapshot -k
```

Response

The above request returns the HTTP status code 200 (OK) and the following data:

```
{
 snapshotDeletionStatus:   0
 deletedList: ["pool1/Local/proj1/vol1@Manual-V-snap1"]
 failedToDeleteList: [ ]
}
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json"  \
-d '[{"pool1/Local/NoProj/vol1@Manual-V-snap1", false}]' \
https://198.51.100.10/zebi/api/v1/deleteVolumeSnapshot -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
```

```
  message:   "Unable to open pool1/Local/NoProj/vol1@Manual-V-
snap1: dataset does not exist."
  extendedData: { }
  details:   ""
  code:   "EZEBI_GENERAL"
}
```

deleteShareSnapshot

Deletes the specified share snapshot.

Caution: If the **recursive** parameter is set to **true**, all dependent objects (snapshots and clones of the specified share snapshot) are also deleted.

Related APIs

createShareSnapshot.

Parameters

shareSnapshotPath

Dataset path of the share snapshot. The dataset path of a share snapshot has the following format: `PoolName`/Local/`ProjectName`/`ShareName`@`SnapshotName`. You can get the snapshotPath from the **listSnapshots** API. For more information, see *listSnapshots*.

recursive

A boolean value that specifies whether dependents of the snapshot are deleted before deleting the snapshot.

Returns

A JSON object of type *SnapShotDeletionStatus*.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '["pool1/Local/proj1/share1@Manual-P-snap1", false]' \
https://198.51.100.10/zebi/api/v1/deleteShareSnapshot -k
```

Response

```
{
snapshotDeletionStatus:   0
deletedList: ["pool1/Local/proj1/share1@Manual-P-snap1"]
failedToDeleteList: [ ]
}
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '["pool1/Local/proj1/NoSuchShare@Manual-P-snap1", false]' \
https://198.51.100.10/zebi/api/v1/deleteShareSnapshot -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
  snapshotDeletionStatus:   2
  deletedList: [ ]
  failedToDeleteList: [
    "pool1/Local/proj1/NoSuchShare@Manual-P-snap1"
   ]
}
```

Chapter 9

Replication Methods

Topics:

getReplicationConfigList

Lists all the replication configurations for the specified project.

Related APIs

getReplicationStatus.

Parameters

poolName

Name of a pool.

projectName

Name of a project within the specified pool.

Returns

A JSON object of type *ReplicationConfig_V1_2*.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '["pool1","p1"]' \
https://198.51.100.10/zebi/api/v1/getReplicationConfigList -k
```

Response

The above request returns the HTTP status code 200 (OK) and the following data:

```
[
  {
    id:   1
    projectName:   "p1"
    projectGuid:   "f5553354-2a91-4533-8e98-1cd52b1da3d6"
    poolName:   "pool1"
    baseDataSetName:   "pool1/Local/p1"
    scopeOption:   0
    remoteHost:   "198.51.100.11"
    lastSnapshotName:   ""
    remotePoolName:   "san-pool"
    remoteProjectName:   "p1"
    remoteBaseDataSetName:   "san-pool/Replica/p1"
  }
]
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '["NotAPool","Failover_LUN"]' \
https://198.51.100.10/zebi/api/v1/getReplicationConfigList -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
  message:   null
  extendedData: {
              EX_CAUSE_MESSAGE:   null
              }
  details:   null
  code:   "EZEBI_GENERAL"
}
```

getReplicationStatus

Gets the replication status for the specified replication configuration.

Related APIs

getReplicationConfigList, *startReplication*.

Parameters

replicationConfig

An object of type *ReplicationConfig_V1_2* that contains the replication configuration. You can get the list of replication configurations from the **`getReplicationConfigList`** API. For more information, see *getReplicationConfigList*.

Returns

A JSON object of type *ReplicationStatus_v1_2*.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json"  -d '[    \
{         \
```

```
"id": 1,  \
"projectName": "p1",    \
"projectGuid": "f5553354-2a91-4533-8e98-1cd52b1da3d6",    \
"poolName": "pool1",    \
"baseDataSetName": "pool1/Local/p1",    \
"scopeOption": 0,    \
"remoteHost": "198.51.100.11",    \
"lastSnapshotName": "",            \
"remotePoolName": "san-pool",      \
"remoteProjectName": "p1",          \
"remoteBaseDataSetName": "san-pool/Replica/p1"  \
}    \
]'   \
https://198.51.100.10/zebi/api/v1/getReplicationStatus -k
```

Response

The above request returns the HTTP status code 200 (OK) and the following data:

```
{
currentStatus:   1
startTimestamp:   1410165951163
completeTimestamp:   1410165951163
updateTimestamp:   1410165951120
dataSent:   0
sendSpeed:   0
taskSize:   0
completedTask:   0
}
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
 -d '[ \
{ \
"id": 1, \
"projectName": "p2", \
"projectGuid": "f5553354-2a91-4533-8e98-1cd52b1da3d6", \
"poolName": "pool-22", \
"baseDataSetName": "pool1/Local/p1", \
"scopeOption": 0, \
"remoteHost": "10.7.1.16", \
"lastSnapshotName": "", \
"remotePoolName": "san-pool", \
"remoteProjectName": "p1", \
"remoteBaseDataSetName": "san-pool/Replica/p1" \
} \
]' https://198.51.100.10/zebi/api/v1/getReplicationStatus -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
message:   "Pool pool-22 is not mounted."
extendedData: { }
details:   ""
code:   "EZEBI_GENERAL"
}
```

startReplication

Starts a replication for the specified replication configuration.

Related APIs

getReplicationConfigList, *getReplicationStatus*.

Parameters

replicationConfig

An object of type *ReplicationConfig_V1_2* that contains the replication configuration.

Returns

No Data.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '[    \
  {    \
 "projectName": "rep_project",    \
 "remoteProjectName": "replica_project",    \
 "remoteBaseDataSetName": "testpool/Replica/replica_project",
 \
 "poolName": "pool1",    \
 "lastSnapshotName": "",    \
 "scopeOption": 1,    \
 "remoteHost": "198.51.100.20",    \
 "baseDataSetName": "p1aut-1/Local/rep_project",    \
 "id": 1,    \
 "projectGuid": "9d6b46ce-05dd-4df1-9ca9-4924bfeb9473",    \
 "remotePoolName": "testpool"    \
  }    \
]'       \
```

```
https://198.51.100.10/zebi/api/v1/startReplication -k
```

Response

The above request returns the HTTP status code 200 (OK) and with no data.

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '[    \
  {    \
 "projectName": "rep_project",    \
 "remoteProjectName": "replica_project",    \
 "remoteBaseDataSetName": "testpool/Replica/replica_project",
 \
 "poolName": "pool-22",    \
 "lastSnapshotName": "",    \
 "scopeOption": 1,    \
 "remoteHost": "198.51.100.20",    \
 "baseDataSetName": "p1aut-1/Local/rep_project",    \
 "id": 1,    \
 "projectGuid": "9d6b46ce-05dd-4df1-9ca9-4924bfeb9473",    \
 "remotePoolName": "testpool"    \
  }    \
]'       \
https://198.51.100.10/zebi/api/v1/startReplication -k
```

Error Response

The above request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
message:   "Pool pool-22 is not mounted."
extendedData: { }
details:   ""
code:   "EZEBI_GENERAL"
}
```

Chapter 10

System Method

Topics:

listSystemProperties

Lists values of the requested system properties for a Tegile array.

Parameters

properties

An array of strings where each string is a predefined string literal indicating a system property. The enumeration *ZEBI_SYSTEM_PROPERTY* defines the string literals that can be requested.

Returns

A JSON array of strings that contains values of the requested system properties. The error "EZEBI_RESOURCE_NOT_FOUND" is returned if a requested system property is not available.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '[["ZEBI_API_VERSION","ZEBI_APPLIANCE_VERSION"]]' \
https://198.51.100.10/zebi/api/v1/listSystemProperties -k
```

Response

```
[    "1.1",    "A1"]
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json  \
-d '[["ZEBI_API_VERSIONS"]]' \
https://198.51.100.10/zebi/api/v1/listSystemProperties -k
```

Error Response

```
[
"EZEBI_RESOURCE_NOT_FOUND"
]
```

Chapter 11

Objects

Topics:

The objects used by the IntelliFlash API are described in the following topics.

DatasetStatus

Field	Type	Description
cleanupException	String	Contains details of the exception, if an exception occurs.
cleanupStatus	Integer return value defined in *CLEANUP_STATUS*	See *CLEANUP_STATUS*.
commandException	String	Contains details of the exception, if an exception occurs.
commandStatus	Integer return value defined in *COMMAND_STATUS*	See *COMMAND_STATUS*.
datasetPath	String	A string that contains the dataset path. A dataset path should have the format `PoolName`/Local/`ProjectName`/`VolumeName` for volumes and `PoolName`/ Local/`ProjectName`/`ShareName` for shares.
overwriteException	String	Contains details of the exception, if an exception occurs.
overwriteStatus	Integer return value defined in *OVERWRITE_STATUS*	See *OVERWRITE_STATUS*.

IscsiInitiator_V1_0

Field	Type	Description
chapSecret	String	Optional chap secret if the initiator uses CHAP for authentication.
chapUserName	String	Optional chap username if the initiator uses CHAP for authentication.

Field	Type	Description
initiatorName	String	Standard initiator names can have either of these two formats: • **`iqn.yyyy-mm.[reverse-domain-name]`** • **`eui.02004567A425678D`** (EUI-64 identifier - 16 ASCII-encoded hexadecimal digits) The characters ,, /,\\, !, ?, @, <, >, #, $, ',%, ^,*, (,), ~,+, =, },|, {, [,], ;, \', \", _, & are not allowed in initiatorgroupname. The empty and space characters and the null values are not allowed in initiatorgroupname.

Pool_V1_0

Field	Type	Description
availableSize	long	The available size of the pool in bytes.
name	String	Name of the storage pool.
totalSize	long	The total size of the pool in bytes.

Project_V1_0

Field	Type	Description
local	Boolean	Indicates whether the project belongs to the current Tegile array controller.
name	String	Name of the project.
poolName	String	The pool in which the project exists.

Share_V1_0

Field	Type	Description
availableSize	Long	The available size of the share in bytes.
datasetPath	String	This field is a string that uniquely identifies the share on a Tegile array. A dataset path should have the format: `PoolName`/Local/`ProjectName`/`ShareName`.
local	Boolean	This boolean identifies whether the share belongs to a local project or a replicated project.

Field	Type	Description
mountpoint	String	This string exposes the mountpoint of the share on a Tegile array.
name	String	Name of the share.
poolName	String	The pool that contains this share.
projectName	String	The project that contains this share.
totalSize	Long	The total size of the share in bytes.

Volume_V1_0

Field	Type	Description
blockSize	String	The block size of the volume.
datasetPath	String	This field is a string that uniquely identifies the volume on a Tegile array. A dataset path should have the format: `PoolName`/Local/`ProjectName`/`VolumeName`. You can get the datasetPath from the **`listVolumes`** API. For more information, see *listVolumes*. The datasetPath is not required for *createVolume* API.
local	Boolean	This boolean identifies whether the volume belongs to a local project or a replicated project. The local boolean is not required for *createVolume* API.
luId	String	The unique identifier for the lun. The luId is not required for *createVolume* API.
name	String	Name of the volume.
poolName	String	The pool that contains this volume.
projectName	String	The project that contains this volume.
protocol	String	This is the protocol on which the volume will be exposed. The valid values are iSCSI, FC, and Unknown.
thinProvision	Boolean	Indicates whether this volume is thin provisioned or thick provisioned.
volSize	Long	The size of the volume in bytes.

Project_V1_2

Field	Type	Description
local	boolean	Indicates whether the project belongs to the current Tegile array controller.
name	String	Name of the project.
poolName	String	The pool in which the project exists.

ShareOptions

Field	Type	Description
blockSize	String	Block size of the share. Valid values are 4KB, 8KB, 16KB, 32KB, 64KB, or 128KB.
mountPoint	String	Mount point of the share
quota	Long	Maximum amount of storage space (in bytes) the share can use. If set to "-1", no quota limit is set on the share.
reservation	Long	Amount of storage space (in bytes) reserved for the share. If set to "-1", no storage space is reserved for the share.

SharePermissions

Field	Type	Description
groupList	Array of *LocalGroup_V1_2* objects	A JSON array of *LocalGroup_V1_2* object. You can use the response of the *listGroups* method for this paramater. This will be used if the *sharePermissionEnum* parameter (*Permission_type_enum*) is set to GROUP.
sharePermissionEnum	Integer return value defined in *Permission_type_enum*	User ACL permission type. Valid values are defined by the *Permission_type_enum* enumeration.
sharePermissionMode	Integer return value defined in *Mode_enum*	User ACL mode. Valid values are defined by the *Mode_enum* enumeration.

Field	Type	Description
userList	Array of *LocalUser_V1_2* objects	A JSON array of *LocalUser_V1_2* object. You can use the response of the *listUsers* method for this parameter. This will be used if the *sharePermissionEnum* parameter (*Permission_type_enum*) is set to USER.

ReplicationConfig_V1_2

Field	Type	Description
baseDataSetName	String	Base Dataset name
id	Long	Replication config id
lastSnapshotName	String	Last snapshot name
poolName	String	Pool name
projectGuid	String	Project Guid
projectName	String	Project name
remoteBaseDataSetName	String	Remote dataset name
remoteHost	String	Target(Remote) host
remotePoolName	String	Remote pool name
remoteProjectName	String	Remote project name
scopeOption	Integer return value defined in *Replication_Scope_Option*	Scope option

ReplicationStatus_v1_2

Field	Type	Description
completedTask	int	Number of tasks completed
completeTimestamp	Date	Time stamp indicating when replication completed.
currentStatus	Integer return value defined in *State*	Current status of replication
dataSent	long	Total data sent
sendSpeed	long	Replication data send speed

Field	Type	Description
startTimestamp	Date	Time stamp indicating when replication started.
taskSize	int	Total task size
updateTimestamp	Date	Time stamp indicating when replication was last updated.

SnapshotProgressStatus

Field	Type	Description
SnapshotProgressStatus	Integer	An integer from the enumeration *SNAPSHOT_PROGRESS_STATUS* that indicates the snapshot progress status.

ProjectCloneProgressStatus_v1_2

Field	Type	Description
failedSubProjects	integer	Number of sub projects (shares and volumes) for which clone snapshot has failed.
projectCloneState	Integer return value defined in *CLONE_PROGRESS_STATUS*	Clone progress state (in progress, success, partial, or failure).
totalSubProjects	integer	Total number of sub projects for the given project.

SnapShotDeletionStatus

Field	Type	Description
deletedList	List	Deleted snapshots list
failedToDeleteList	List	Not deleted snapshots list
snapshotDeletionStatus	Integer return value defined in *SNAPSHOT_DELETION_STATUS*	Snapshot deletion status (success, partial, or failure)

deleteProjectSnapshot, *deleteVolumeSnapshot*, *deleteShareSnapshot*, .

LocalUser_V1_2

Field	Type	Description
groupId	int	Group ID of the group
groupName	String	Name of the group
userId	int	User ID of the user
userName	String	Name of the user

LocalGroup_V1_2

Field	Type	Description
groupId	int	Group ID of the group
groupName	String	Name of the group
userList	List	List of users associated to the group

LunStatus

Field	Description
accessState	An integer indicating whether the LUN is: • active (0) • active to standby (1) • standby (2) • standby to active (3)
alias	The LUN alias, if specified.
blockSize	The block size of the LUN.
commandException	Contains details of the exception, if an exception occurs.
commandStatus	See *COMMAND_STATUS*.
dataFile	The data file path for the LUN.
datasetPath	A string that contains the path to the dataset. The dataset path should have the format `PoolName`/Local/`ProjectName`/`VolumeName` for LUNs.
guid	The lunId of the LUN.
metaFile	The meta file path of the LUN.

Field	Description
mgmtURL	The management URL of the LUN.
operationalStatus	Operational status of the LU. • Stmf (SCSI target mode framework) logical unit offline (0) • Stmf logical unit offlining (1) • Stmf logical unit online (2) • Stmf logical unit onlining (3) • Stmf logical unit unregistered (4)
productId	Field not used.
serialNumber	Field not used.
size	The size of the LUN.
vendorId	Field not used.
viewCount	The number of mappings defined for the LUN.
writeCacheDisable	A boolean that indicates if the data write cache is disabled.
writeProtect	A boolean that indicates if write protect is enabled or disabled. • Read-Only (True) • Write (False)

Chapter 12

Enumerations

Topics:

The enumerations used by the IntelliFlash API are described in the following topics.

ZEBI_SYSTEM_PROPERTY

Value	Description
ZEBI_APPLIANCE_MODEL	Indicates the Tegile array controller storage array model, such as HA2100, HA2400, HA2800, T3100, or T3200.
ZEBI_APPLIANCE_VERSION	The Tegile array version, such as A1 and A2.
ZEBI_GUI_VERSION	The IntelliFlash Web UI version, such as 2.1.3.5 (150617)-399.
ZEBI_SUPPORTED_TDPS_API_VERSIONS	The Tegile Data Protection Service (TDPS) versions supported by this version of the IntelliFlash API.
ZEBI_API_MINOR_VERSION	Indicates the minor version of the IntelliFlash API.
ZEBI_API_VERSION	Indicates the full version of the IntelliFlash API.

COMMAND_STATUS

Status	Returned Value	Description
COMMAND_SUCCEED	0	Indicates that command (request) succeeded.
COMMAND_NOT_ATTEMPTED	1	Indicates that command (request) not attempted.
COMMAND_FAILED	2	Indicates that command (request) failed.

CLEANUP_STATUS

Status	Returned Value	Description
CLEANUP_NONE	0	Indicates cleanup none.
CLEANUP_NEEDED	1	Indicates that cleanup is needed.
CLEANUP_DONE	2	Indicates that cleanup is completed.
CLEANUP_FAILED	3	Indicates that cleanup has failed.

OVERWRITE_STATUS

Status	Returned Value	Description
OVERWRITE_NONE	0	Indicates overwrite none.
OVERWRITE_DONE	1	Indicates overwrite completed.
OVERWRITE_FAILED	2	Indicates overwrite failed.

Replication_Scope_Option

Specifies the condition that determines which datasets in a project will be replicated when you start replication on the project.

Related APIs, Objects, and Enumerations

getReplicationConfigList, *getReplicationStatus*, *startReplication*, *ReplicationConfig_V1_2*.

Value	Description
FULL	All datasets in the project will be replicated
INCLUDE	All selected datasets will be replicated
EXCLUDE	All selected datasets will not be replicated

State

The `State` enumeration indicates the state of a replication request.

States of a replication task

An uninterrupted and successful replication task goes through the start, sending, completing, and completed states. If the task fails due to an error or if you abort a running task, you can restart it at a later time.

Related APIs, Objects, and Enumerations

getReplicationConfigList, *getReplicationStatus*, *startReplication*, *ReplicationConfig_V1_2*, *ReplicationStatus_v1_2*.

Status	Returned Value	Description
UNKNOWN	0	Indicates that the replication task exited due to an unknown error.
START	1	Indicates that the replication task has started.

Status	Returned Value	Description
RESTART	2	Indicates that an interrupted replication task has restarted.
SENDING	3	Indicates that the system is sending replication data.
COMPLETING	4	Indicates that data transfer for replication is complete, and the replication task is finishing.
COMPLETED	5	Indicates that the replication task is complete.
ERROR	6	Indicates that the replication task exited with an error due to system, network, or other issues.
ABORTING	7	Indicates that the initial state (before **`ABORTED`**) of an aborted replication. You cannot restart the replication task if it is aborting.
ABORTED	8	Indicates that you have aborted the replication. If you abort a replication, the system rolls back to the previous replication snapshot completely.
ABANDONING	9	Indicates that the system is **`ABANDONING`** a running replication task. The system abandons a replication if you manually switchover the pool or if the pool goes offline for any reason.

Mode_enum

Indicates the mode for ACLs supplied using the SharePermissions object to the createShare methods.

Value	Description
ALLOW	Indicates that permission should be granted to the specified set of users.
DENY	Indicates that permission should be denied to the specified set of users.

Related APIs, Objects, and Enumerations

createShare, *createShare*, *SharePermissions*.

Permission_type_enum

Indicates the scope of ACLs supplied using the SharePermissions object to the createShare methods.

Related APIs, Objects, and Enumerations

createShare, *createShare*, *SharePermissions*.

Value	Description
EVERYONE	Indicates that the supplied ACL is for everyone.
USER	Indicates that the supplied ACL is for the specified user.
GROUP	Indicates that the supplied ACL is for the specified group.

SNAPSHOT_PROGRESS_STATUS

Status	Returned Value	Description
SUCCESS	0	The snapshot request completed successfully.
INPROGRESS	1	The snapshot request is in progress.
ERROR	2	The snapshot request failed due to an error.

SNAPSHOT_DELETION_STATUS

Indicates the status of a snapshot deletion request.

Related APIs, Objects, and Enumerations

deleteProjectSnapshot, *deleteVolumeSnapshot*, *deleteShareSnapshot*.

Status	Returned Value	Description
SUCCESS	0	Indicates that the snapshot deletion succeeded.
PARTIAL	1	This value is applicable only to the *deleteProjectSnapshot* API. It indicates that only some of the snapshots could be deleted (other snapshots that were selected for deletion could not be deleted.)
FAILURE	2	Indicates that the snapshot deletion has failed.

CLONE_PROGRESS_STATUS

Indicates the status of a clone project snapshot request.

A value from this enumeration is returned by the *getProjectCloneStatus* API to indicate the status of a project clone request.

Related APIs, Objects, and Enumerations

getProjectCloneStatus, *cloneProjectSnapshot*, *ProjectCloneProgressStatus_v1_2*.

Status	Returned Value	Description
INPROGRESS	0	Indicates that the *cloneProjectSnapshot* request is in progress.
SUCCESS	1	Indicates that all of the projects snapshots are cloned successfully.
PARTIAL	2	Indicates that some of the projects snapshots are cloned successfully
FAILURE	3	Indicates that none of the projects snapshots are cloned.

Chapter 13

IntelliFlash API Error Codes

Topics:

Error Codes

The IntelliFlash API uses the following error codes:

Error Code	Description
EZEBI_GENERAL	Indicates a general error.
EZEBI_INVALID_ARGUMENT	Indicates invalid arguments.
EZEBI_PERMISSION_DENIED	Indicates that permission is denied.
EZEBI_NOMEMORY	Indicates that no memory is left.
EZEBI_NOSPACE	Indicates that no space is left on device.
EZEBI_RESOURCE_SUSPENDED	Indicates that resource operation is suspended.
EZEBI_RESOURCE_BUSY	Indicates that resource is busy.
EZEBI_RESOURCE_INUSE	Indicates that resource required is being used by others.
EZEBI_RESOURCE_EXIST	Indicates that target already exists.
EZEBI_RESOURCE_CORRUPTED	Indicates that resource is corrupted.
EZEBI_RESOURCE_NOT_FOUND	Indicates that resource is not found.
EZEBI_REQUEST_EXIST	Indicates that request is in progress already.
EZEBI_REQUEST_INTERRUPTED	Indicates that request is interrupted.
EZEBI_REQUEST_TIMEOUT	Indicates that request is timed out.
EZEBI_HOST_UNREACHABLE	Indicates that host is unreachable.
EZEBI_HOST_UNKNOWN	Indicates that host is unknown.

Appendix A

Appendix A

Topics:

JSON Quick Reference

This quick reference includes some JSON examples for users who are not familiar with the JSON syntax.

All JSON data sent in HTTP requests must be enclosed within square brackets ([ ]). For example, to send a single string, use the following:

```
["pool1"]
```

Note: As JSON ignores whitespace, such as newlines, tabs, and spaces, you can also send the following:

```
[
    "pool1"
]
```

Boolean

```
true
```

```
false
```

Integers

```
213
```

String

```
"pool1"
```

Array of strings

```
[
    "string1","string2","string3"
]
```

Objects

```
{
    "lunNumber":-1,
    "name":"testVol",
    "local":true,
}
```

Array of objects

```
[
```

```
    {“lunNumber”:-1,“name”:”testVol”,“local”:true},
    {“lunNumber”:-1,“name”:”testVol”,“local”:true}
]
```

Mixed

```
    “DatasetPath”,
    [{“lunNumber”:-1,“name”:”testVol”,“local”:true}],
    true
```

Appendix B

Appendix B

Topics:

- *Deprecated APIs*

Deprecated APIs

The following APIs have been deprecated. Instead of the deprecated APIs use one of the alternate APIs suggested in the note.

createSnapshots

Creates a snapshot for the specified dataset using the specified snapshot name. It can also delete previously-created snapshots, if an error occurs.

Note: This API is deprecated. Use the *createVolumeSnapshot*, *createProjectSnapshot*, or *createShareSnapshot* APIs instead.

Related APIs

getProjectSnapshotCreationStatus, *getVolumeSnapshotCreationStatus*, *getShareSnapshotCreationStatus*, *listSnapshots*, *cloneSnapshot* .

Parameters

snapshotPaths

An array of strings that specify the dataset path (including the name of the snapshot to be created.) This string should have the format: `datasetPath@SnapshotName`. The `datasetPath` should identify a share or a volume.

override

A boolean value that indicates whether a snapshot is regenerated (if true) if a snapshot with the same name already exists.

cleanupOnError

A boolean value that indicates whether to clean up (if true) previously-created snapshots if an error happens.

Returns

A JSON array of *DatasetStatus* objects that contain the dataset path of the newly created snapshot and the results of the operation.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '[["pool1/Local/TechPubs/TechPubsTest@api_SnapShotName_1"], \
```

```
false, false]' \
https://198.51.100.10/zebi/api/v1/createSnapshots -k
```

Response

```
[
 {
   "datasetPath": "pool1/Local/TechPubs/
TechPubsTest@api_SnapShotName_1",
   "overwriteStatus": 0,
   "overwriteException": null,
   "commandStatus": 0,
   "commandException": null,
   "cleanupStatus": 0,
   "cleanupException": null
  }
]
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '[["pool1/Local/TechPubs/TechPubs@api_SnapShotName_4"], \
false, false]' \
https://198.51.100.10/zebi/api/v1/createSnapshots -k
```

Error Response

```
   [
    {
        "datasetPath": "pool1/Local/TechPubs/
TechPubs@api_SnapShotName_4",
        "overwriteStatus": 0,
        "overwriteException": null,
        "commandStatus": 2,
        "commandException": {
            "code": "EZEBI_RESOURCE_NOT_FOUND",
            "details": "Unable to open pool1/Local/TechPubs/
TechPubs : dataset does not exist",
            "extendedData": {
                "EX_CAUSE_CODE_NAME": "EZFS_NOENT",
                "EX_CAUSE_MESSAGE": "Unable to open pool1/Local/
TechPubs/TechPubs : dataset does not exist",
                "EX_CAUSE_CODE_NUMBER": "2009"
            },
            "message": "Unable to open pool1/Local/TechPubs/
TechPubs : dataset does not exist"
        },
        "cleanupStatus": 0,
        "cleanupException": null
    }
]
```

deleteSnapshots

Deletes the specified snapshots and optionally, all dependent snapshots.

Note: This API is deprecated. Use the *deleteShareSnapshot* or *deleteVolumeSnapshot* APIs instead.

Caution: If the **`recursive`** parameter is set to **`true`**, all dependent objects (snapshots and clones of the specified snapshot) are also deleted.

Related APIs

createSnapshots

Parameters

snapshotPaths

An array of strings that contains paths to snapshots to be deleted. This snapshot paths should have the format: `datasetPath@SnapshotName`. The datasetPath should be a path to a valid share or volume.

recursive

A boolean value that indicates whether to remove the dependents (if true) of this snapshot before trying to delete it.

errorIfNotExist

A boolean that indicates whether to raise an exception (if true) if any of the given snapshot path does not exist.

Returns

A JSON array of *DatasetStatus* objects that contain the dataset path of the deleted snapshot and results of the operation.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '[["pool1/Local/TechPubs/TechPubsTest@api_SnapShotName_1"], \
false, true]' \
https://198.51.100.10/zebi/api/v1/deleteSnapshots -k
```

Response

```
[
{
```

```
"datasetPath": "pool1/Local/TechPubs/
TechPubsTest@api_SnapShotName_1",
"overwriteStatus": 0,
"overwriteException": null,
"commandStatus": 0,
"commandException": null,
"cleanupStatus": 0,
"cleanupException": null
}
]
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '[["pool1/Local/TechPubs/TechPubs@api_SnapShotName_1"], \
false, true]' \
https://198.51.100.10/zebi/api/v1/deleteSnapshots -k
```

Error Response

```
[
[
{
"datasetPath": "pool1/Local/TechPubs/
TechPubs@api_SnapShotName_1",
"overwriteStatus": 0,
"overwriteException": null,
"commandStatus": 2,
"commandException": {
"code": "EZEBI_RESOURCE_NOT_FOUND",
"details": "Unable to open pool1/Local/TechPubs/
TechPubs@api_SnapShotName_1 : dataset does not exist",
"extendedData": {
"EX_CAUSE_CODE_NAME": "EZFS_NOENT",
"EX_CAUSE_MESSAGE": "Unable to open pool1/Local/TechPubs/
TechPubs@api_SnapShotName_1 : dataset does not exist",
"EX_CAUSE_CODE_NUMBER": "2009"
},
"message": "Dataset pool1/Local/TechPubs/
TechPubs@api_SnapShotName_1 does not exists"
},
"cleanupStatus": 0,
"cleanupException": null
}
]
]
```

deleteSnapshots

Deletes snapshots (and optionally all dependent snapshots in the specified path) whose names match with the given pattern.

Note: This API is deprecated. Use the *deleteShareSnapshot* or *deleteVolumeSnapshot* APIs instead.

Caution: If the **`recursive`** parameter is set to **`true`**, all dependent objects (snapshots and clones of the specified snapshot) are also deleted.

Related APIs

createSnapshots

Parameters

datasetPath

A string that contains the dataset path of the snapshot. The dataset path has the format: `PoolName`/Local/`ProjectName`/`VolumeName`.

snapshotPattern

A regular expression (regex) for matching snapshot names. Use an empty string to delete all snapshots in the given path.

recursive

A boolean value that indicates whether to remove (if true) dependents of the matching snapshots before deleting the snapshots themselves.

errorIfNotExist

A boolean value that indicates whether to raise an exception (if true) if the path specified by datasetPath does not exist.

Returns

A JSON array of *DatasetStatus* objects that contain the dataset path of the deleted snapshot and results of the operation.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '["pool1/Local/TechPubs/TechPubsTest", "api*", \
false, true]' \
https://198.51.100.10/zebi/api/v1/deleteSnapshots -k
```

Response

```
[
{
"datasetPath": "api_SnapShotName_1",
"overwriteStatus": 0,
"overwriteException": null,
"commandStatus": 2,
"commandException": {
"code": "EZEBI_RESOURCE_NOT_FOUND",
"details": "Unable to open api_SnapShotName_1 : dataset does not
 exist",
"extendedData": {
 "EX_CAUSE_CODE_NAME": "EZFS_NOENT",
 "EX_CAUSE_MESSAGE": "Unable to open api_SnapShotName_1 :
 dataset does not exist",
 "EX_CAUSE_CODE_NUMBER": "2009"
 },
"message": "Dataset api_SnapShotName_1 does not exists"
 },
"cleanupStatus": 0,
"cleanupException": null
},
{
"datasetPath": "Auto-LF-Day-012114-21:15",
"overwriteStatus": 0,
"overwriteException": null,
"commandStatus": 2,
"commandException": {
"code": "EZEBI_RESOURCE_NOT_FOUND",
"details": "Unable to open Auto-LF-Day-012114-21:15 : dataset
 does not exist",
"extendedData": {
"EX_CAUSE_CODE_NAME": "EZFS_NOENT",
"EX_CAUSE_MESSAGE": "Unable to open Auto-LF-Day-012114-21:15 :
 dataset does not exist",
"EX_CAUSE_CODE_NUMBER": "2009"
},
"message": "Dataset Auto-LF-Day-012114-21:15 does not exists"
},
"cleanupStatus": 0,
"cleanupException": null
}
]
```

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '["pool1/Local/TechPubs/TechPubs", "api*", \
false, true]' \
https://198.51.100.10/zebi/api/v1/deleteSnapshots -k
```

Error Response

```
{
"message": "Unable to open pool1/Local/TechPubs/TechPubs :
 dataset does not exist",
"extendedData": {
"EX_CAUSE_CODE_NAME": "EZFS_NOENT",
"EX_CAUSE_MESSAGE": "Unable to open pool1/Local/TechPubs/
TechPubs : dataset does not exist",
"EX_CAUSE_CODE_NUMBER": "2009"
},
"details": "Unable to open pool1/Local/TechPubs/TechPubs :
 dataset does not exist",
"code": "EZEBI_RESOURCE_NOT_FOUND"
}
```


cloneSnapshot

Clones a snapshot to a new dataset.

Note: This API is deprecated. Use the *cloneProjectSnapshot*, *cloneShareSnapshot*, or *cloneVolumeSnapshot* APIs instead.

Related APIs

listSnapshots, *createSnapshots*.

Parameters

snapshotPath

A string that identifies the path for the snapshot that needs to be cloned. The snapshot path has the format: *datasetPath@snapshotName*. The `datasetPath` must be a valid path to a share or a volume.

cloneName

A string that contains the name of the new dataset.

clonesSettings

A boolean value that indicates whether to clone the settings that are required to share the new dataset.

readOnly

A boolean value that indicates whether to make the new dataset a read-only clone.

promotesIt

A boolean value that indicates whether to promote the new dataset so that it is possible to remove to old dataset.

Returns

If the dataset is a volume, the LUN ID (GUID) of the new volume is returned. If the dataset is a share, nothing is returned.

Examples

Request (curl):

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '["pool1/Local/TechPubs/TechPubsTest@api_SnapShotName_1", \
"api_CloneName_1", false, false, false]' \
https://198.51.100.10/zebi/api/v1/cloneSnapshot -k
```

Response

The above request returns the HTTP status code 200 (OK) and no data.

Erroneous Request 1 (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json  \
-d '["pool1/Local/TechPubs/TechPubsTest@api_SnapShotName-Bad", \
"api_CloneName_1", false, false, false]' \
https://198.51.100.10/zebi/api/v1/cloneSnapshot -k
```

Error Response

```
HTTP Status Code: 400
{
  "message": "An entity with the same name already exists.",
  "extendedData": {},
  "details": "",
  "code": "EZEBI_GENERAL"
}
```

Erroneous Request 2 (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '["pool1/Local/TechPubs/TechPubs@api_SnapShotName", \
"api_CloneName_1", false, false, false]' \
https://198.51.100.10/zebi/api/v1/cloneSnapshot -k
```

Error Response

```
{
  "message": "Unable to open pool1/Local/TechPubs/TechPubs :
 dataset does not exist",
  "extendedData": {
```

```
        "EX_CAUSE_CODE_NAME": "EZFS_NOENT",
        "EX_CAUSE_MESSAGE": "Unable to open pool1/Local/
TechPubs/TechPubs : dataset does not exist",
        "EX_CAUSE_CODE_NUMBER": "2009"
        },
  "details": "Unable to open pool1/Local/TechPubs/TechPubs :
 dataset does not exist",
  "code": "EZEBI_RESOURCE_NOT_FOUND"
}
```