

Tegile IntelliFlash™ API Reference Guide

Version 2.0

Notice

Copyright

2017 © Western Digital Corporation or its affiliates, all rights reserved.

Notice: No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or stored in a database or retrieval system for any purpose, without the express written permission of Tegile Systems, Inc. (hereinafter referred to as “Tegile”).

Tegile reserves the right to make changes to this document at any time without notice and assumes no responsibility for its use. Tegile products and services can be ordered only under the terms and conditions of Tegile's applicable agreements. All of the features described in this document may not be currently available. Refer to the latest product announcement or contact your local Tegile sales office for information on feature and product availability. This document includes the latest information available at the time of publication.

Tegile, IntelliFlash, IntelliFlash HD, IntelliCare, IntelliShell, Zebi, and MASS are trademarks of Tegile Systems, Inc. in the United States and other countries.

All other trademarks, service marks, and company names in this document are properties of their respective owners.

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.....	vii
Audience.....	vii
Tegile Documentation.....	vii
Support.....	viii
Contacting Tegile Technical Support.....	viii
Documentation Comments.....	viii
Revision.....	viii
 Chapter 1: Introduction to the IntelliFlash API.....	 1
Key Features.....	2
Scope of the API.....	2
Unified APIs.....	2
Error and Exception Handling in the API.....	3
Error Codes.....	3
Using the API Examples.....	4
curl Syntax.....	5
Creating a Volume and Exposing It.....	5
Backing Up a Volume.....	6
 Chapter 2: Sample Programs.....	 7
Sample Perl Script.....	8
Sample Python Program.....	8
Sample PowerShell Program.....	11
 Chapter 3: User and Group Methods.....	 13
listUsers.....	14
listGroups.....	15
createUser.....	16
createUser.....	17
createGroup.....	19
createGroup.....	20
createUserAndGroup.....	21
changeUserPassword.....	23
deleteUser.....	24
deleteGroup.....	25
 Chapter 4: SAN Methods.....	 27
listISCSIInitiatorGroups.....	28
listISCSITargetGroups.....	28

listFCInitiatorGroups.....	29
listFCTargetGroups.....	30
initiatorGroupExists.....	30
createInitiatorGroup.....	31
listInitiatorsInInitiatorGroup.....	33
listTargetsInTargetGroup.....	34
createScsiInitiator.....	35
getInitiatorGroup.....	36
addInitiatorToInitiatorGroup.....	37
createMappingForVolume.....	38
deleteMappingFromVolume.....	40

Chapter 5: Dataset Methods..... 43

listPools.....	44
listProjects.....	44
listVolumes.....	46
listLunsById.....	47
listShares.....	49
createShare.....	51
createShare.....	53
createVolume.....	55
deleteDataset.....	57
deleteShare.....	58
deleteVolume.....	60

Chapter 6: Snapshot Methods..... 63

listSnapshots.....	64
createProjectSnapshot.....	65
createVolumeSnapshot.....	66
createShareSnapshot.....	68
getProjectSnapshotCreationStatus.....	69
getVolumeSnapshotCreationStatus.....	71
getShareSnapshotCreationStatus.....	73
cloneProjectSnapshot.....	75
cloneVolumeSnapshot.....	77
cloneVolumeSnapshot.....	78
cloneShareSnapshot.....	80
getProjectCloneStatus.....	81
deleteProjectSnapshot.....	83
deleteVolumeSnapshot.....	85
deleteShareSnapshot.....	86
rollBackToProjectSnapshot.....	88
rollBackToVolumeSnapshot.....	89
rollBackToShareSnapshot.....	90

Chapter 7: Replication Methods..... 93

getReplicationConfigList.....	94
getReplicationStatus.....	95
startReplication.....	97
Chapter 8: System Methods.....	99
listSystemProperties.....	100
Chapter 9: SNMP Methods.....	101
recreateSNMPTables.....	102
resyncSNMPTables.....	102
Chapter 10: Objects.....	105
DatasetStatus.....	106
IscsiInitiator_V1_0.....	106
Pool_V1_0.....	107
Project_V1_0.....	107
Share_V1_0.....	107
Volume_V1_0.....	108
Project_V1_2.....	109
ShareOptions.....	109
SharePermissions.....	109
ReplicationConfig_V1_2.....	110
ReplicationStatus_v1_2.....	110
SnapshotProgressStatus.....	111
ProjectCloneProgressStatus_v1_2.....	111
SnapshotDeletionStatus.....	111
LocalUser_V1_2.....	111
LocalGroup_V1_2.....	112
LunStatus.....	112
Chapter 11: Enumerations.....	115
ZEBI_SYSTEM_PROPERTY.....	116
COMMAND_STATUS.....	116
CLEANUP_STATUS.....	116
OVERWRITE_STATUS.....	117
Replication_Scope_Option.....	117
State.....	117
Mode_enum.....	118
Permission_type_enum.....	119
SNAPSHOT_PROGRESS_STATUS.....	119
SNAPSHOT_DELETION_STATUS.....	119
CLONE_PROGRESS_STATUS.....	120
Appendix A: Appendix A.....	121
JSON Quick Reference.....	122

Appendix B: Appendix B..... 125

- Deprecated APIs..... 126
 - createSnapshots..... 126
 - deleteSnapshots..... 128
 - deleteSnapshots..... 130
 - cloneSnapshot.....132

Preface

About this Guide

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



Note: To reference older versions of the software, this document uses the term IntelliFlash instead of Zebi, except where noted.

Audience

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

Tegile Documentation

The following table, [Tegile Documentation](#), lists the technical documentation types available for Tegile arrays and expansion shelves.

Table 1: Tegile Documentation

Name	Description
Tegile T4000 Series and IntelliFlash HD Hardware Guide	Contains detailed descriptions, hardware specifications, and rack installation instructions for Tegile storage arrays and expansion shelves.
Tegile IntelliFlash User Guide	Contains detailed instructions on how to configure, use, and administer Tegile arrays.
Tegile IntelliFlash API Reference Guide	Contains detailed description of the Tegile REST API.
Tegile IntelliFlash Configuration Wizard Guide	Contains instructions on how to initially configure a Tegile array using the Configuration Wizard.
Tegile IntelliFlash Release Notes	Provides details on enhancements, fixed issues, and known issues for a release.

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
10/30/2017	Updated for IntelliFlash version 3.5.4.1. Document changes include: <ul style="list-style-type: none">• Chapter 1: Introduction to the IntelliFlash API [updated; Supersedes previous Chapter 2: Common Workflows, Chapter 3: Using the API Method Reference, and Chapter 13: IntelliFlash API Error Codes.]• Chapter 9: SNMP Methods [new]

Date	Description
07/31/2017	Miscellaneous changes: Updated examples, descriptions, and tables throughout the document.
07/08/2016	Updated for IntelliFlash version 3.5.0.1. Three new APIs added: rollBackToProjectSnapshot , rollBackToVolumeSnapshot , and rollBackToShareSnapshot . Enhanced listSystemProperties API and updated parameters for ZEBI_SYSTEM_PROPERTY and Using the API Examples .
01/26/2015	<p>Initial release for IntelliFlash REST APIs version 2.0. The changes in this version are the following:</p> <ul style="list-style-type: none"> • Added cloneVolumeSnapshot API. • Removed Best practices for accessing the controllers using the API section. • Added Unified APIs section. • Changed the path <code>../v1/...</code> to <code>../v2/..</code> for all the APIs.

Chapter 1

Introduction to the IntelliFlash API

Topics:

- [Key Features](#)
- [Scope of the API](#)
- [Unified APIs](#)
- [Error and Exception Handling in the API](#)
- [Error Codes](#)
- [Using the API Examples](#)
- [curl Syntax](#)
- [Creating a Volume and Exposing It](#)
- [Backing Up a Volume](#)

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: Use version 2.0 of the IntelliFlash REST APIs with Tegile arrays that are running IntelliFlash version 3.5.0 and higher. It is recommended not to use Version 1.2 of the API with 3.5.0 or later.

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.

Scope of the API

The IntelliFlash REST APIs enable you to do the following tasks:

- List pools, projects, volumes, users, groups, LUNs, shares, snapshots, initiators, initiator groups, targets, target groups, and system properties.
- Create users, groups, volumes, snapshots, initiator groups, mapping for volumes, and iSCSI initiators.
- Clone snapshots.
- Roll back project, share, and volume snapshots.
- Delete users, groups, snapshots, volumes, shares, mapping for volume, volume snapshot, share snapshot, and other datasets.
- Check whether an initiator group exists.
- Add an initiator to an initiator group.
- Retrieve the initiator group for an initiator.
- Obtain replication configuration list, status, and to start replication.

Unified APIs

IntelliFlash REST APIs run on and obtains results from both controllers in a Tegile array. For example, the **listProjects** API returns projects belonging to a pool, irrespective of the controller on which the pool currently resides (Controller-A or Controller-B).



Note:

The IntelliFlash REST APIs have been unified in version 2.0 and higher. Previous versions of the IntelliFlash REST APIs—version 1.0 and version 1.2—could access only one controller in an API request.



Caution: Tegile cautions against using Version 1.2 of the IntelliFlash REST APIs with IntelliFlash version 3.5.0 or higher. You must use version 2.0 of the IntelliFlash REST APIs instead.

Invoking the Unified API with the array management IP address

You are required to use the array management IP address instead of the controller management IP addresses for the unified API to work correctly.

URL changes across the API versions

APIs in version 2.x include "/v2/" in the API endpoint. For example:

`https://<ArrayManagementHostNameOrIPAddress>/zebi/api/v2/<APIname>`

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. Possible status codes include:
 - 400 (bad request)
 - 404 (not found).
 - 500 (internal server error).
- 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 an enumeration field that indicates an error.

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.

Error Code	Description
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.

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 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 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 -n 'username:password' | base64
```

- Use data that is relevant to your environment and requirements instead of the dummy data given in the examples.
- Use the Array Management IP address 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 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 warnings/errors generated due to the self-signed certificate.

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 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. If the FC fabric configurations are correct, the initiators automatically log in to the target ports.

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. Mount the newly cloned volume and take a backup.
4. Clean up the cloned volume using the [deleteVolume](#) method.
5. Clean up the snapshot from which the clone was created using the [deleteVolumeSnapshot](#) method.

Chapter 2

Sample Programs

Topics:

- [Sample Perl Script](#)
- [Sample Python Program](#)
- [Sample PowerShell Program](#)

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/v2/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 address (198.51.100.10) with your Array Management IP address.

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

url = "https://198.51.100.10/zebi/api/v2/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/v2/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/v2/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/v2/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/v2/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/v2/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/v2/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/v2/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 address (198.51.100.10) with the IP address of your Tegile array.

Sample PowerShell Program

The following PowerShell program illustrates how to authenticate, accept (trust) the self-signed certificate, and invoke the [createShare](#) API.

```

$bytes = [System.Text.Encoding]::UTF8.GetBytes("admin:s")
$token = [System.Convert]::ToBase64String($bytes)

$headers = @{"Authorization"="Basic $token"; "Content-Type"="application/
json"}
$url = "https://198.51.100.10/zebi/api/v2/createShare"
$method = "POST"
[System.Net.ServicePointManager]::ServerCertificateValidationCallback =
{ $TRUE }
[System.Net.ServicePointManager]::SecurityProtocol =
[System.Net.SecurityProtocolType]::Tls12;

$shareOptions = @{}
$shareOptions.add("blockSize", "64KB")
$shareOptions.add("quota", -1)
$shareOptions.add("reservation", -1)

#This group should already exist on the array.
#A better approach is to obtain the group list using the listGroups API and
then
#use the needed group, instead of hard-coding like below
$group = @{}
$group.add("groupName", "group01")
$group.add("groupId", 104)

$groupList = @($group)

$sharePermission = @{}
$sharePermission.add("groupList", $groupList)
$sharePermission.add("sharePermissionEnum", 2) #2 is group permission
$sharePermission.add("sharePermissionMode", 0) #0 is "Allow"

$sharePermissionArray = @($sharePermission)

$parameters = "pool-b","test-project","APIShare",$shareOptions,
$sharePermissionArray
$jsonString = ConvertTo-Json -Compress -Depth 4 $parameters

```

```
$jsonString
```

```
Invoke-RestMethod -Method $method -Headers $headers -Uri $url -Body  
$jsonString
```



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

Chapter 3

User and Group Methods

Topics:

- [*listUsers*](#)
- [*listGroups*](#)
- [*createUser*](#)
- [*createUser*](#)
- [*createGroup*](#)
- [*createGroup*](#)
- [*createUserAndGroup*](#)
- [*changeUserPassword*](#)
- [*deleteUser*](#)
- [*deleteGroup*](#)

The following sections describe User and Group API methods, parameters and return types. They also include examples with sample responses.

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/v2/listUsers -k
```

Response

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

```
[
{
  "userName":"nfsuser1",
  "groupName":"nfsgrp1",
  "userId":104,"groupId":104
},
{
  "userName":"nfsuser5",
  "groupName":"nfsgrp5",
  "userId":105,
  "groupId":105
},
{
  "userName":"nfsuser4",
  "groupName":"nfsgrp4",
  "userId":106,
  "groupId":106},
{
  "userName":"nfsuser3",
  "groupName":"nfsgrp3",
  "userId":107,
  "groupId":107
},
{"userName":"nfsuser2",
```



```
"groupName":"nfsgrp2",
"userId":108,
"groupId":108
}
]
```

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/v2/listGroups -k
```

Response

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

```
[
{
"groupName":"nfsgrp1",
"groupId":104,
"userList":["nfsuser1"]
},
{
"groupName":"nfsgrp4",
"groupId":106,
"userList":["nfsuser4"]
},
{
"groupName":"nfsgrp5",
"groupId":105,
"userList":["nfsuser5"]
},
{
```

```

"groupName": "nfsgrp2",
"groupId": 108,
"userList": ["nfsuser2"]
},
{
"groupName": "nfsgrp3",
"groupId": 107,
"userList": ["nfsuser3"]
}
]

```

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 characters and the empty and null strings are not allowed in password.

Returns

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

Examples

Example 1

Request (curl)

```

curl -X POST -H 'Content-Type: application/json' -d '{
  "userName": "nfsuser2",
  "uid": 108,
  "groupName": "nfsgrp2",
  "password": "nfsuser2"
}' http://localhost:8080/api/createUser

```

```
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/v2/createUser -k
```

Response

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

```
0
```

Example 2

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/v2/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 system.

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 characters and the empty and null strings are not allowed in password.

Returns

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

Examples

Example 1

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/v2/createUser -k
```

Response

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

```
0
```

Example 2

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/v2/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

Example 1

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/v2/createGroup -k
```

Response

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

```
0
```

Example 2

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/v2/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

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '["AnotherNewGroup"]' \
https://198.51.100.10/zebi/api/v2/createGroup -k
```

Response

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

```
0
```

Example 2

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '[""]' https://198.51.100.10/zebi/api/v2/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

Example 1

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/v2/createUserAndGroup -k
```

Response

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

```
0
```

Example 2

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/v2/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 characters and the empty and null strings are not allowed in password.

Returns

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

Examples

Example 1

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/v2/changeUserPassword -k
```

Response

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

```
0
```

Example 2

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/v2/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.)



Warning: The delete operation is not reversible.

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

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '["testUser2"]' \
https://198.51.100.10/zebi/api/v2/deleteUser -k
```

Response

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

```
0
```

Example 2

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/v2/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:

- The delete operation is not reversible.
- 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

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '{"NewGroup"}' \
https://198.51.100.10/zebi/api/v2/deleteGroup -k
```

Response

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

```
0
```

Example 2

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/v2/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 4

SAN Methods

Topics:

- [*listISCSIInitiatorGroups*](#)
- [*listISCSITargetGroups*](#)
- [*listFCInitiatorGroups*](#)
- [*listFCTargetGroups*](#)
- [*initiatorGroupExists*](#)
- [*createInitiatorGroup*](#)
- [*listInitiatorsInInitiatorGroup*](#)
- [*listTargetsInTargetGroup*](#)
- [*createScsiInitiator*](#)
- [*getInitiatorGroup*](#)
- [*addInitiatorToInitiatorGroup*](#)
- [*createMappingForVolume*](#)
- [*deleteMappingFromVolume*](#)

The following sections describe SAN methods, parameters and return types. They also include examples with sample responses.

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.

Example

Request (curl)

```
curl -X GET -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
https://198.51.100.10/zebi/api/v2/listISCSIInitiatorGroups \
-k
```

Response

```
[
  "inigrp1",
  "testinigroup"
]
```

listISCSITargetGroups

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

Related APIs

[listISCSIInitiatorGroups](#), [createMappingForVolume](#).

Parameters

None

Returns

Returns an array of JSON strings. Each string returned is a name of an iSCSI target group found within the list of all iSCSI target groups found on the array.

Example

Request (curl)

```
curl -X GET -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
https://198.51.100.10/zebi/api/v2/listISCSITargetGroups -k
```

Response

```
[
  "default-plaut-iscsi-target-group",
  "tgtgrp1", "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 in this list is a group name within the complete list of Fibre Channel Initiator group names found on the Tegile array.

Example

Request (curl)

```
curl -X GET -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '[]' https://198.51.100.10/zebi/api/v2/listFCInitiatorGroups -k
```

Response

```
[
  "fcinigroup",
  "fcinigroup1"
]
```

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 returned is the name of one Fibre Channel (FC) target group within the list of all FC target groups on the array. If the 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/v2/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

Example 1

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/v2/initiatorGroupExists -k
```

Response

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

```
true
```

Example 2

Erroneous Request

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json -d '[2012]' \
https://198.51.100.10/zebi/api/v2/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

[createScsiInitiator](#), [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, where:

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

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '[ "APIInitiatorGroup" ]' \
https://198.51.100.10/zebi/api/v2/createInitiatorGroup -k
```

Response

The above request returns the HTTP status code 200 (OK) and the integer 0 indicating success.

Example 2

Erroneous Request

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '[ "API InitiatorGroup" ]' \
https://198.51.100.10/zebi/api/v2/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 returned has a name of an initiator in the specified initiator group.

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '["APIInitiatorGroup"]' \
https://198.51.100.10/zebi/api/v2/listInitiatorsInInitiatorGroup -k
```

Response

```
[
  "iqn.1998-01.com.vmware:esx99",
  "iqn.1998-01.com.vmware:esx98"
]
```

Example 2

Erroneous Request

```
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/v2/listInitiatorsInInitiatorGroup -k
```

Error Response

If the initiator group is not found, the above request returns the HTTP status code 200 (OK) and 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 returned is a name of a target in the specified target group.

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '["iscsitarget61"]' \

https://198.51.100.10/zebi/api/v2/listTargetsInTargetGroup \
-k
```

Response

```
[
  "iqn.2012-02.com.tegile:iscsitarget61",
  "iqn.2012-02.com.tegile:test",
  "iqn.2012-02.com.tegile:test1"
]
```

Example 2

Erroneous Request

```
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/v2/listTargetsInTargetGroup \
-k
```

Error Response

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

createIscsiInitiator

Creates an iSCSI initiator object on the Tegile array. If the initiator name already exists, then the method fails.

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, where

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

Examples

Example 1

Request (curl)

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

Response

The above request returns the HTTP status code 200 (OK) and an integer 0 indicating success.

Example 2

Erroneous Request

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

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

Error Response

In the above request, the initiator name is empty. So the request returns the HTTP status code 400 (bad request) and the following message:

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "",
  "message": "Initiator name is not valid.",
  "extendedData": {}
}
```

getInitiatorGroup

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

Related APIs

[listInitiatorsInInitiatorGroup](#), [createIscsiInitiator](#), [addInitiatorToInitiatorGroup](#)

Parameters

initiatorName

The name of the initiator. This is a string.

Returns

Returns a JSON string. The string has the name of the initiator group associated with the given initiator.

Examples

Example 1

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/v2/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
```

Example 2

Erroneous Request

```
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/v2/getInitiatorGroup -k
```

Error Response

If the initiator name is not found, the request returns the HTTP status code 200 (OK) and 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

The name of an initiator.

initiatorGroupName

A string: the name of an initiator group.

Returns

Returns an integer, where:

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

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '["iqn.2012-11.com.tegile.iscsi:Initiator1", "iscsi-TestGroup"]' \
```

```
https://198.51.100.10/zebi/api/v2/addInitiatorToInitiatorGroup -k
```

Response

The above request returns the HTTP status code 200 (OK) and an integer 0 indicating success.

Example 2

Erroneous Request

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '["iqn.2012-11.com.tegile.iscsi:Initiator2", "iscsi-TestGroup2"]'\
https://198.51.100.10/zebi/api/v2/addInitiatorToInitiatorGroup -k
```

Error Response

In the above request, the initiator already exists in the group. So the 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:Initiator2
             to iscsi-TestGroup2 : 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

The dataset path of the volume. This is a string. 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](#).

initiatorGroupName

The name of the initiator group to which the volume must be mapped. This is a string.

targetGroupName

The name of the target group to which the volume must be mapped. This is a string.

lunNumber

The LUN number for the newly defined LUN. To assign a LUN number automatically (default), use the value -1. This is an integer.

Returns

Returns an integer, where

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

Examples

Example 1

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/v2/createMappingForVolume -k
```

Response

```
0
```

Example 2

Erroneous Request

```
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/v2/createMappingForVolume -k
```

Error Response

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

deleteMappingFromVolume

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



Note: The delete operation deletes the mapping. But you can add the original mapping again.

Related APIs

[createMappingForVolume](#), [initiatorGroupExists](#), [listISCSIInitiatorGroups](#), [listISCSITargetGroups](#), [listVolumes](#).

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](#).

initiatorGroupName

A string. The name of an initiator group.

targetGroupName

A string. The name of a target group.

Returns

Returns an integer, where

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

Examples

Example 1

Request (curl)

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

```
-d '["pool1/Local/TechPubs/TechPubsLUN", \
  "iscsi-initiatorGroup", \
  "iscsi-TargetGroup"]' \
https://198.51.100.10/zebi/api/v2/deleteMappingFromVolume -k
```

Response

The above request returns an integer 0, which indicates success.

Example 2

Erroneous Request

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '["pool1/Local/TechPubs2/TechPubsLUN", \
  "iscsi-initiatorGroup", \
  "iscsi-TargetGroup"]' \
https://198.51.100.10/zebi/api/v2/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 5

Dataset Methods

Topics:

- [*listPools*](#)
- [*listProjects*](#)
- [*listVolumes*](#)
- [*listLunsById*](#)
- [*listShares*](#)
- [*createShare*](#)
- [*createShare*](#)
- [*createVolume*](#)
- [*deleteDataset*](#)
- [*deleteShare*](#)
- [*deleteVolume*](#)

The following sections describe Dataset methods, parameters and return types. They also include examples with sample responses.

listPools

Lists all the pools on the array. 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.

Example

Request (curl)

```
curl -X GET -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json -d '[]'\
https://198.51.100.10/zebi/api/v2/listPools -k
```

Response

```
[
{
  "name":"pool-a",
  "availableSize":3931776248832,
  "totalSize":3931908341760
},
{
  "name":"pool-b",
  "availableSize":1965925029376,
  "totalSize":1965954170880
}
```

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

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '["plaut", true]' \
https://198.51.100.10/zebi/api/v2/listProjects -k
```

Response

```
[
{
  "poolName":"plaut",
  "name":"CIFS_TEST",
  "local":true
},
{
  "poolName":"plaut",
  "name":"new_proj",
  "local":true
}
]
```

Example 2

Erroneous Request

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json -d '["pool11",true]' \
https://198.51.100.10/zebi/api/v2/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

Example 1

Request (curl)

```

curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '["plaut","project1", true]' \
https://198.51.100.10/zebi/api/v2/listVolumes -k

```

Response

```

[
{

```



```

    "poolName": "plaut",
    "projectName": "project1",
    "name": "iscsilun_0",
    "luId": "600144F0DE8CCA000000561C554A0006",
    "volSize": 1073741824,
    "blockSize": "4KB",
    "thinProvision": false,
    "protocol": "iSCSI",
    "datasetPath": "plaut/Local/project2/iscsilun_0",
    "local": true
  }
}

```

Example 2

Erroneous Request

```

curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '["pool0","TechPubs", true]' \
https://198.51.100.10/zebi/api/v2/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 LUN details for the specified LUN IDs.

Related APIs

[listPools](#), [listProjects](#), [listVolumes](#), [listShares](#), [createVolume](#).

Parameters

lunIds

An array of strings where each string contains the logical unit ID of a LUN on the array. For example, "600144F0FA2A820000004FF35C280003".

Returns

A JSON array of [LunStatus](#) objects that contain details of the requested LUNs.

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d
'[[{"600144F0B4510D00000005631F7DB0001","600144F0B4510D00000005631F7E80002"}]]' \
https://198.51.100.10/zebi/api/v2/listLunsById -k
```

Response

```
[
{
  "viewCount" : 1,
  "operationalStatus" : 2,
  "metaFile" : null,
  "commandStatus" : 0,
  "size" : "1073741824",
  "vendorId" : null,
  "writeCacheDisable" : false,
  "dataFile" : "/dev/zvol/rdisk/pool-a/Local/smb_nfs/llun1",
  "guid" : "600144F0B4510D00000005631F7DB0001",
  "accessState" : 0,
  "commandException" : null,
  "blockSize" : null,
  "productId" : null,
  "serialNumber" : null,
  "writeProtect" : false,
  "alias" : "/dev/zvol/rdisk/pool-a/Local/smb_nfs/llun1",
  "mgmtURL" : "",
  "datasetPath" : "pool-a/Local/smb_nfs/llun1"
},
{
  "viewCount" : 1,
  "operationalStatus" : 2,
  "metaFile" : null,
  "commandStatus" : 0,
  "size" : "",
  "vendorId" : null,
  "writeCacheDisable" : false,
  "dataFile" : "/dev/zvol/rdisk/pool-a/Local/smb_nfs/lun2",
  "guid" : "600144F0B4510D00000005631F7E80002",
  "accessState" : 0,
  "commandException" : null,
  "blockSize" : null,
  "productId" : null,
  "serialNumber" : null,
  "writeProtect" : false,
  "alias" : "/dev/zvol/rdisk/pool-a/Local/smb_nfs/lun2",
  "mgmtURL" : "",
  "datasetPath" : "pool-a/Local/smb_nfs/lun2"
}
]
```

Example 2

Erroneous Request

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '[[{"600144F0121901000000052D92EC20165"}]]' \
https://198.51.100.10/zebi/api/v2/listLunsById -k
```

Error Response

```
HTTP Status Code: 200
[
  {
    "commandStatus": 2,
    "commandException": {
      "code": "EZEBI_RESOURCE_NOT_FOUND",
      "details": "",
      "extendedData": {},
      "message": "Lun 600144F0121901000000052D92EC20165 doesn't exist"
    },
    "guid": null,
    "alias": null,
    "dataFile": null,
    "metaFile": null,
    "vendorId": null,
    "productId": null,
    "mgmtURL": null,
    "serialNumber": null,
    "viewCount": 0,
    "size": null,
    "blockSize": null,
    "writeProtect": false,
    "writeCacheDisable": false,
    "operationalStatus": 0,
    "accessState": 0,
    "datasetPath": null
  }
]
```

listShares

Lists all the local and 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**Example 1****Request (curl)**

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '["plaut","project2", true]' \
https://198.51.100.10/zebi/api/v2/listShares -k
```

Response

```
[
  {
    "poolName": "plaut",
    "projectName": "project2",
    "name": "default_share",
    "availableSize": 9275971622400,
    "totalSize": 9275971769856,
    "datasetPath": "plaut/Local/project2/default_share",
    "mountpoint": "/export/plaut/project2/default_share",
    "local": true
  }
]
```

Example 2**Erroneous Request**

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '["pool0","TechPubs", true]' \
https://198.51.100.10/zebi/api/v2/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.



Important:

In an SMB3 enabled environment, if a project has both NFS and SMB sharing enabled, creating share using **createShare** API is not supported. To enable share creation, turn off any one of these protocols on the project.

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

An array of the [SharePermissions](#) object that defines permissions for the new share using ACLs.

Returns

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

Examples

Example 1

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": 2,
    "sharePermissionMode": 0,
    "groupList": [{
      "groupName": "newGroupName",
      "groupId": 104
    }]
  }]
}'
https://198.51.100.10/zebi/api/v2/createShare -k
```

Response

HTTP Status Code: 200

```
0
```

Example 2

Erroneous Request

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '{"BadPoolName", "projectName", "newShare2", \
  {
    "blockSize": "64KB",
    "quota": -1,
    "reservation": -1
  },
  [{
    "sharePermissionEnum": 2,
    "sharePermissionMode": 0,
    "groupList": [{
      "groupName": "newGroupName",
      "groupId": 104
    }]
  }]
}'
https://198.51.100.10/zebi/api/v2/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).



Important:

In an SMB3 enabled environment, if a project has both NFS and SMB sharing enabled, creating share using **createShare** API is not supported. To enable share creation, turn off any one of these protocols on the project.

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

An array of the [SharePermissions](#) object that defines permissions for the new share using ACLs.

Returns

An integer: The number 0 if the request succeeds.

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '{"poolName": "projectName", "shareName", \
  [ \
    {"sharePermissionEnum": 2, "sharePermissionMode": 0, \
      "groupList": [ \
        {"groupName": "newAPIGroup", "groupId": 104} \
      ] \
    } \
  ] \
}' \
https://198.51.100.10/zebi/api/v2/createShare -k
```

Response

HTTP Status Code: 200

```
0
```

Example 2

Erroneous Request


```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '{"BadPoolName", "projectName", "shareName", \
  [ \
    {"sharePermissionEnum": 2, "sharePermissionMode": 0, \
      "groupList": [ \
        {"groupName": "newAPIGroup", "groupId": 104} \
      ] \
    } \
  ] \
}' \
https://198.51.100.10/zebi/api/v2/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

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. This is a boolean value.

Returns

Returns an integer, where:

- 0 indicates that the request succeeded.

- 1 indicates that the request was not attempted.
- 2 indicates that the request failed.

Examples

Example 1

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/v2/createVolume -k
```

Response

```
0
```

Example 2

Erroneous Request

```
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/v2/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.



Warning: The delete operation is not reversible.

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

Example 1

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/v2/deleteDataset -k
```

Response

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

Example 2

Erroneous Request

```
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/v2/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.



Warning: The delete operation is not reversible.

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`. You can obtain the `datasetPath` from the `listShares` API. For more information, see [listShares](#) and [Share_V1_0](#).

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

Example 1

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/v2/deleteShare -k
```

Response

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

Example 2

Erroneous Request

```
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/v2/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.



Warning: The delete operation is not reversible.

Related APIs

[listVolumes](#), [createVolume](#).

Parameters

datasetPath

A string: the path which uniquely identifies the volume on the Tegile array. 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](#).

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**.

errorIfExists

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

Returns

Returns no data.

Examples

Example 1

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/v2/deleteVolume -k
```

Response

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

Example 2**Erroneous Request**

```
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/v2/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 6

Snapshot Methods

Topics:

- [*listSnapshots*](#)
- [*createProjectSnapshot*](#)
- [*createVolumeSnapshot*](#)
- [*createShareSnapshot*](#)
- [*getProjectSnapshotCreationStatus*](#)
- [*getVolumeSnapshotCreationStatus*](#)
- [*getShareSnapshotCreationStatus*](#)
- [*cloneProjectSnapshot*](#)
- [*cloneVolumeSnapshot*](#)
- [*cloneVolumeSnapshot*](#)
- [*cloneShareSnapshot*](#)
- [*getProjectCloneStatus*](#)
- [*deleteProjectSnapshot*](#)
- [*deleteVolumeSnapshot*](#)
- [*deleteShareSnapshot*](#)
- [*rollBackToProjectSnapshot*](#)
- [*rollBackToVolumeSnapshot*](#)
- [*rollBackToShareSnapshot*](#)

The following sections describe Snapshot methods, parameters and return types. They also include examples with sample responses.

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](#).

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

Example 1

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/v2/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"
]
```

Example 2

Erroneous Request

```
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/v2/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

Example 1

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/v2/createProjectSnapshot -k
```

Response

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

Example 2

Erroneous Request

```
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/v2/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

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN"
-H "Content-Type:application/json" \
-d '[{"poolName": "pool1", \
  "projectName": "vProj", "name": "vol2", \
  "luId": "600144F0A63089000000053BD51250002", \
  "volSize": 161061273600, "blockSize": "32KB", \
  "thinProvision": false, "protocol": "FC", \
  "datasetPath": "pool1/Local/vProj/vol2", \
  "local": true}, {"poolName": "vProj_S3", false}]' \
https://198.51.100.10/zebi/api/v2/createVolumeSnapshot -k
```

Response

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

Example 2

Erroneous Request

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

```
"projectName": "vProj", "name": "vol2", \
"luId": "600144F0A63089000000053BD51250002", \
"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/v2/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

Example 1

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/v2/createShareSnapshot -k
```

Response

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

Example 2

Erroneous Request

```
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/v2/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_sl:
  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

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '["pool1/Local/sProj", "NewTPSS111"]' \
https://198.51.100.10/zebi/api/v2/\
getProjectSnapshotCreationStatus -k
```

Response

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

```
[
  {
    snapshotProgressStatus: 0
  }
]
```

Example 2

Erroneous Request

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '["NotAPool/Local/sProj", ""]' \
https://198.51.100.10/zebi/api/v2/\
```



```
getProjectSnapshotCreationStatus -k
```

Error Response

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

```
{
  snapshotProgressStatus: 2
}
```

Example 3

Erroneous Request

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '["NotAPool/Local/sProj", "NewTPSS111"]' \
https://198.51.100.10/zebi/api/v2/\
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](#).

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

Example 1

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/v2/\
getVolumeSnapshotCreationStatus -k
```

Response

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

```
[
  {
    snapshotProgressStatus: 0
  }
]
```

Example 2

Erroneous Request

```
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/v2/\
getVolumeSnapshotCreationStatus -k
```

Error Response

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

```
{snapshotProgressStatus: 2}
```

Example 3

Erroneous Request

```
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/v2/\
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](#).

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

Example 1

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/v2/\
getShareSnapshotCreationStatus -k
```

Response

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

```
[
  {
    snapshotProgressStatus: 0
  }
]
```

Example 2

Erroneous Request

```
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/v2/\
getShareSnapshotCreationStatus -k
```

Error Response

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

```
[
  {
    snapshotProgressStatus: 2
  }
]
```

Example 3

```
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/v2/\
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

Example 1

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/v2/cloneProjectSnapshot -k
```

Response

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

Example 2

Erroneous Request

```
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/v2/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"
}
```

Example 3

Erroneous Request

```
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/v2/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

[cloneVolumeSnapshot](#), [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.

protocol

A boolean value that indicates protocol to be set for the clone. Valid values are **true** for iSCSI and **false** for FC.

Returns

No Data.

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '["pool1/Local/vProj/volume1@Manual-Vsnap5",
    "vol_clone_new_iscsi_5", true, true, true]' \
https://198.51.100.10/zebi/api/v2/cloneVolumeSnapshot -k
```

Response

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

cloneVolumeSnapshot

Clones the specified snapshot of a volume.

Related APIs

[cloneVolumeSnapshot](#), [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

Example 1

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/v2/cloneVolumeSnapshot -k
```

Response

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

Example 2

Erroneous Request

```
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/v2/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 v2. Please check if method name and
arguments are correct."
  extendedData: {
    EX_CAUSE_MESSAGE: "No such method found in v2.
Please check if method name and arguments are correct."
  }
  details: "No such method found in v2.
Please check if method name and arguments are correct."
  code: "EZEBI_GENERAL"
}
```

Example 3

Erroneous Request

```
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/v2/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

Example 1

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/v2/cloneShareSnapshot -k
```

Response

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

Example 2

Erroneous Request

```
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/v2/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

Example 1

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/v2/getProjectCloneStatus -k
```

Response

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

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

Example 2

Erroneous Request

```
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/v2/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.



Warning: The delete operation is not reversible.

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 dependents 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

Example 1

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/v2/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: [ ]
}
```

Example 2

Erroneous Request

```
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/v2/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"]
}
```

Example 3

Erroneous Request

```
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/v2/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.



Warning: The delete operation is not reversible.

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

Example 1

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/v2/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: [ ]
}
```

Example 2

Erroneous Request

```
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/v2/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.



Warning: The delete operation is not reversible.

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

Example 1

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/v2/deleteShareSnapshot -k
```

Response

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

Example 2

Erroneous Request

```
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/v2/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"
  ]
}
```

rollBackToProjectSnapshot

Reverts the project state to the point-in-time state when the snapshot was taken.



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

Related APIs

[createProjectSnapshot](#), [listSnapshots](#), [deleteProjectSnapshot](#)

Parameters

snapshotPath

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

deleteDependents

A boolean value: indicates whether to delete the snapshot dependents.

If the **deleteDependents** is set to **false** and rollback is invoked, the method throws an error if there are existing dependents for the snapshot.

Returns

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

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '["pool-2-mirror/Local/proj-test@Manual-P-test",true]' \
https://198.51.100.10/zebi/api/v2/rollBackToProjectSnapshot -k
```

Response

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

```
0
```

Example 2

Erroneous Request

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '["pool-2-mirror/Local/proj-test1@Manual-P-test",true]' \
https://198.51.100.10/zebi/api/v2/rollBackToProjectSnapshot -k
```

Error Response

```
{ "message": "Snapshot
  pool-2-mirror/Local/proj-test1@Manual-P-test does not
  exist.", "extendedData": {}, "details": "", "code": "EZEBI_GENERAL" }
```

rollBackToVolumeSnapshot

Reverts the volume state to the point-in-time state when the snapshot was taken.



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

Related APIs

[createVolumeSnapshot](#), [listSnapshots](#), [deleteVolumeSnapshot](#)

Parameters

snapshotPath

Path of the volume-level snapshot that has to be rolled back. 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](#).

deleteDependents

A boolean value: indicates whether to delete the snapshot dependents.

If the **deleteDependents** is set to **false** and rollback is invoked, the method throws an error if there are existing dependents for the snapshot.

Returns

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

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '["pool-2-mirror/Local/proj-test/lun_test@Manual-V-test",true]' \
https://198.51.100.10/zebi/api/v2/rollBackToVolumeSnapshot -k
```

Response

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

```
0
```

Example 2

Erroneous Request

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '["pool-2-mirror/Local/proj-test/lun_test_dummy@Manual-V-test",true]' \
https://198.51.100.10/zebi/api/v2/rollBackToVolumeSnapshot -k
```

Error Response

```
{ "message": "Snapshot
pool-2-mirror/Local/proj-test/lun_test_dummy@Manual-V-test does not
exist.", "extendedData": {}, "details": "", "code": "EZEBI_GENERAL" }
```

rollBackToShareSnapshot

Reverts the share state to the point-in-time state when the snapshot was taken.



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

Related APIs

createShareSnapshot, listSnapshots, deleteShareSnapshot

Parameters

snapshotPath

Path of the share-level snapshot that has to be rolled back. 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](#).

deleteDependents

A boolean value: indicates whether to delete the snapshot dependents.

If the **deleteDependents** is set to **false** and rollback is invoked, the method throws an error if there are existing dependents for the snapshot.

Returns

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

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '["pool-2-mirror/Local/proj-test/share1@Manual-S-test",true]' \
https://198.51.100.10/zebi/api/v2/rollBackToShareSnapshot -k
```

Response

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

```
0
```

Example 2

Erroneous Request

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '["pool-2-mirror/Local/proj-test/share_test@Manual-S-test",true]' \
https://198.51.100.10/zebi/api/v2/rollBackToShareSnapshot -k
```

Error Response

```
{ "message": "Snapshot
pool-2-mirror/Local/proj-test/share_test@Manual-S-test does not
exist.", "extendedData": {}, "details": "", "code": "EZEBI_GENERAL" }
```

Chapter 7

Replication Methods

Topics:

- [*getReplicationConfigList*](#)
- [*getReplicationStatus*](#)
- [*startReplication*](#)

The following sections describe Replication methods, parameters and return types. They also include examples with sample responses.

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

Example 1

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/v2/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"
  }
]
```


Example 2

Erroneous Request

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '["NotAPool","Failover_LUN"]' \
https://198.51.100.10/zebi/api/v2/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

Example 1

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/v2/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
}
```

Example 2

Erroneous Request

```
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/v2/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

Example 1

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": "plaut-1/Local/rep_project",
    "id": 1,
```

```
"projectGuid": "9d6b46ce-05dd-4df1-9ca9-4924bfeb9473",    \
"remotePoolName": "testpool"    \
}    \
]'    \
https://198.51.100.10/zebi/api/v2/startReplication -k
```

Response

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

Example 2

Erroneous Request

```
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": "plaut-1/Local/rep_project",    \
    "id": 1,    \
    "projectGuid": "9d6b46ce-05dd-4df1-9ca9-4924bfeb9473",    \
    "remotePoolName": "testpool"    \
  }
]'    \
https://198.51.100.10/zebi/api/v2/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 8

System Methods

Topics:

- [*listSystemProperties*](#)

The following sections describe the System methods, parameters and return types. They also include examples with sample responses.

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

Example 1

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/v2/listSystemProperties -k
```

Response

```
[  "1.2",    "A1"]
```

Example 2

Erroneous Request

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H Content-Type:application/json \
-d '["ZEBI_API_VERSIONS"]' \
https://198.51.100.10/zebi/api/v2/listSystemProperties -k
```

Error Response

```
[
  "EZEBI_RESOURCE_NOT_FOUND"
]
```

Chapter 9

SNMP Methods

Topics:

- [*recreateSNMPTables*](#)
- [*resyncSNMPTables*](#)

The following sections describe SNMP methods, parameters and return types. They also include examples with sample responses.

recreateSNMPTables

Recreates the SNMP entries. After deletion or creation of objects, the SNMP table entries might have gaps in the indices. Use this API to re-index the table entries.

Related APIs

[resyncSNMPTables](#)

Parameters

None

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown

EZEBI_GENERAL

This exception is thrown if the operation failed or SNMP is not enabled.

Examples

Example 1

Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    ]' \
  https://198.51.100.10/zebi/api/v2/recreateSNMPTables -k
```

Response:

The above request returns the HTTP status code 200 (OK) and 0 indicating success. This initiates the SNMP table recreation in the background in asynchronous mode.

resyncSNMPTables

Resyncs the SNMP entries between the two controllers of the array. SNMP queries can be sent to the management IP address or the individual controller IP address. If querying the

individual controllers for the same OID (SNMP Object Index) fetches different response, use this API to resync the table entries between the controllers.

Related APIs

[recreateSNMPTables](#)

Parameters

None

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown

EZEBI_GENERAL

This exception is thrown if the operation failed, or SNMP is not enabled.

Examples

Example 1

Request (curl)

```
curl -X POST \  
  -H 'authorization: Basic Auth TOKEN \  
  -H 'cache-control: no-cache' \  
  -H 'content-type: application/json' \  
  -d '[  
  
    ]' \  
  https://198.51.100.10/zebi/api/v2/resyncSNMPTables -k
```

Response:

The above request returns the HTTP status code 200 (OK) and 0 indicating success. This initiates the SNMP table resync between controllers in the background in asynchronous mode.

Chapter 10

Objects

Topics:

- [*DatasetStatus*](#)
- [*IscsiInitiator_V1_0*](#)
- [*Pool_V1_0*](#)
- [*Project_V1_0*](#)
- [*Share_V1_0*](#)
- [*Volume_V1_0*](#)
- [*Project_V1_2*](#)
- [*ShareOptions*](#)
- [*SharePermissions*](#)
- [*ReplicationConfig_V1_2*](#)
- [*ReplicationStatus_v1_2*](#)
- [*SnapshotProgressStatus*](#)
- [*ProjectCloneProgressStatus_v1_2*](#)
- [*SnapShotDeletionStatus*](#)
- [*LocalUser_V1_2*](#)
- [*LocalGroup_V1_2*](#)
- [*LunStatus*](#)

The following sections describe the objects used by the IntelliFlash API.

DatasetStatus

Field	Type	Description
cleanupException	String	Contains details of the exception, if an exception occurs.
cleanupStatus	Integer	An integer return value as defined in 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	<p>Standard initiator names can have either of these two formats:</p> <ul style="list-style-type: none"> iqn.yyyy-mm.[reverse-domain-name] eui.02004567A425678D (EUI-64 identifier - 16 ASCII-encoded hexadecimal digits) <p>The characters <code>.,/,\,!,?,@,<,>,#,\$,%,&^,*,(,~,+,,{ ,;,\',\","_&</code> are not allowed in initiatorgroupname. The empty and space characters and the null values are not allowed in initiatorgroupname.</p>

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 array.
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: <code>PoolName/Local/ProjectName/VolumeName</code> . 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.
luld	String	The unique identifier for the lun. The luld 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 array.
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 parameter. 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)

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	Type	Description
accessState	Integer	An integer indicating whether the LUN is: <ul style="list-style-type: none"> • active (0) • active to standby (1) • standby (2) • standby to active (3)
alias	String	The LUN alias, if specified.
blockSize	String	The block size of the LUN.
commandException	String	Contains details of the exception, if an exception occurs.
commandStatus	Integer	An integer return value as defined in COMMAND_STATUS .
dataFile	String	The data file path for the LUN.
datasetPath	String	A string that contains the path to the dataset. The dataset path should have the format <code>PoolName/Local/ProjectName/VolumeName</code> for LUNs.
guid	String	The lunId of the LUN.
metaFile	String	The meta file path of the LUN.
mgmtURL	String	The management URL of the LUN.

Field	Type	Description
operationalStatus	Integer	Operational status of the LU. <ul style="list-style-type: none"> • 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	String	Field not used.
serialNumber	String	Field not used.
size	String	The size of the LUN.
vendorId	String	Field not used.
viewCount	Integer	The number of mappings defined for the LUN.
writeCacheDisable	Boolean	A boolean that indicates if the data write cache is disabled.
writeProtect	Boolean	A boolean that indicates if write protect is enabled or disabled. <ul style="list-style-type: none"> • Read-Only (True) • Write (False)

Chapter 11

Enumerations

Topics:

- *ZEBI_SYSTEM_PROPERTY*
- *COMMAND_STATUS*
- *CLEANUP_STATUS*
- *OVERWRITE_STATUS*
- *Replication_Scope_Option*
- *State*
- *Mode_enum*
- *Permission_type_enum*
- *SNAPSHOT_PROGRESS_STATUS*
- *SNAPSHOT_DELETION_STATUS*
- *CLONE_PROGRESS_STATUS*

The following sections describe the enumerations used by the IntelliFlash API.

ZEBI_SYSTEM_PROPERTY

Value	Description
ZEBI_APPLIANCE_MODEL	Indicates the array model.
ZEBI_APPLIANCE_VERSION	Indicates the array version.
ZEBI_GUI_VERSION	The IntelliFlash Web UI version.
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.
INTELLIFLASH_ARRAY_GUID	Indicates the GUID of the IntelliFlash array.
INTELLIFLASH_ARRAY_FQDN	Indicates the FQDN of the IntelliFlash array.

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 is not needed.
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 is not required.
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.

Value	Returned Value	Description
FULL	0	All datasets in the project will be replicated
INCLUDE	1	All selected datasets will be replicated
EXCLUDE	2	All selected datasets will not be replicated

Related APIs, Objects, and Enumerations

[getReplicationConfigList](#), [getReplicationStatus](#), [startReplication](#), [ReplicationConfig_V1_2](#).

State

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

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.
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.

Status	Returned Value	Description
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.

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](#).

Mode_enum

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

Value	Returned Value	Description
ALLOW	0	Indicates that permission should be granted to the specified set of users.
DENY	1	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.

Value	Returned Value	Description
EVERYONE	0	Indicates that the supplied ACL is for everyone.
USER	1	Indicates that the supplied ACL is for the specified user.
GROUP	2	Indicates that the supplied ACL is for the specified group.

Related APIs, Objects, and Enumerations

[*createShare*](#), [*createShare*](#), [*SharePermissions*](#).

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.

Status	Returned Value	Description
SUCCESS	0	Indicates that the snapshot deletion succeeded.
PARTIAL	1	This value is applicable only to the <i>deleteProjectSnapshot</i> 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.

Related APIs, Objects, and Enumerations

[deleteProjectSnapshot](#), [deleteVolumeSnapshot](#), [deleteShareSnapshot](#).

CLONE_PROGRESS_STATUS

Indicates the status of a clone project snapshot request.

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

Related APIs, Objects, and Enumerations

A value from this enumeration is returned by the [getProjectCloneStatus](#) API to indicate the status of a project clone request.

[getProjectCloneStatus](#), [cloneProjectSnapshot](#), [ProjectCloneProgressStatus_v1_2](#).

Appendix A

Appendix A

Topics:

- [JSON Quick Reference](#)

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.

createSnapshots

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

Alternate APIs

The createSnapshots 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/v2/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/v2/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.



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

Alternate APIs

The deleteSnapshots API is deprecated. Use the [deleteShareSnapshot](#) or [deleteVolumeSnapshot](#) APIs instead.

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/v2/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/v2/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.



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

Alternate APIs

The deleteSnapshots API is deprecated. Use the [deleteShareSnapshot](#) or [deleteVolumeSnapshot](#) APIs instead.

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/v2/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/v2/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.

Alternate APIs

The cloneSnapshot 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/v2/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/v2/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/v2/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"
}
```