



IntelliFlash API Reference

Version 2.3

Notice

Legal disclaimer

Western Digital Technologies, Inc. or its affiliates' (collectively "Western Digital") general policy does not recommend the use of its products in life support applications wherein a failure or malfunction of the product may directly threaten life or injury.

Accordingly, in any use of Western Digital products in life support systems or other applications where failure could cause damage, injury, or loss of life, the products should only be incorporated in systems designed with appropriate redundancy, fault tolerant or back-up features. The user of Western Digital products in life support or other such applications assumes all risk of such use and agrees to indemnify, defend, and hold harmless Western Digital or its affiliates against all damages.

This document and related material are for information use only and are subject to change without prior notice. Western Digital assumes no responsibility for any errors that may appear in this document or related material, nor for any damages or claims resulting from the furnishing, performance, or use of this document or related material. Absent a written agreement signed by Western Digital or its authorized representative to the contrary, Western Digital explicitly disclaims any express and implied warranties and indemnities of any kind that may, or could, be associated with this document and related material, and any user of this document or related material agrees to such disclaimer as a precondition to receipt and usage hereof.

Each user of this document or any product referred to herein expressly waives all guaranties and warranties of any kind associated with this document any related materials or such product, whether expressed or implied, including without limitation, any implied warranty of merchantability or fitness for a particular purpose or non-infringement. Each user of this document or any product referred to herein also expressly agrees Western Digital shall not be liable for any incidental, punitive, indirect, special, or consequential damages, including without limitation physical injury or death, property damage, lost data, loss of profits or costs of procurement of substitute goods, technology, or services, arising out of or related to this document, any related

materials or any product referred to herein, regardless of whether such damages are based on tort, warranty, contract, or any other legal theory, even if advised of the possibility of such damages.

This document and its contents, including diagrams, schematics, methodology, work product, and intellectual property rights described in, associated with, or implied by this document, are the sole and exclusive property of Western Digital. No intellectual property license, express or implied, is granted by Western Digital associated with the document recipient's receipt, access and/or use of this document or the products referred to herein; Western Digital retains all rights hereto.

Contact information

Address

5601 Great Oaks Parkway, San Jose, California 95119

Phone

U.S. (Toll-Free): 800.801.4618

International: +1 408.717.6000

© 2019 Western Digital Corporation or its affiliates. All Rights Reserved.

Western Digital, the Western Digital logo, IntelliCare, IntelliFlash, the IntelliFlash logo, IntelliShell, and SnapWRX are registered trademarks or trademarks of Western Digital Corporation or its affiliates in the US and/or other countries. VMware, VMware ESXi, VMware vSphere, and VMware vCenter are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions. All other marks are the property of their respective owners. Product specifications subject to change without notice. Pictures shown may vary from actual products. Not all products are available in all regions of the world.

Contents

| | |
|---|---------------|
| Chapter 1: Introduction to the IntelliFlash Rest APIs..... | 1 |
| API Version History..... | 2 |
| Key Features..... | 8 |
| Scope of the API..... | 9 |
| Unified APIs..... | 10 |
| Error and Exception Handling in the API..... | 10 |
| Error Codes..... | 10 |
| curl Syntax..... | 11 |
| Using the API Examples..... | 11 |
| Creating a Volume and Exposing It..... | 12 |
| Backing Up a Volume..... | 13 |
| Chapter 2: Sample Programs..... | 15 |
| Sample Perl Script..... | 16 |
| Sample Python Program..... | 16 |
| Sample PowerShell Program..... | 19 |
| Chapter 3: User and Group Methods..... | 21 |
| changeUserPassword..... | 22 |
| createGroup..... | 23 |
| createGroup..... | 24 |
| createUser..... | 26 |
| createUser..... | 27 |
| createUserAndGroup..... | 29 |
| deleteGroup..... | 30 |
| deleteUser..... | 32 |
| listGroups..... | 33 |
| listUsers..... | 34 |
| Chapter 4: SAN Methods..... | 37 |
| addInitiatorToInitiatorGroup..... | 38 |
| createFCInitiator..... | 39 |
| createInitiatorGroup..... | 41 |
| createScsiInitiator..... | 42 |
| createISCSITarget..... | 44 |
| createISCSITargetForGroup..... | 47 |
| createMappingForVolume..... | 49 |
| createMappingForVolume..... | 50 |
| createTargetGroup..... | 53 |
| deleteInitiatorGroup..... | 55 |

| | |
|-------------------------------------|----|
| deleteISCSIInitiator..... | 56 |
| deleteISCSITarget..... | 58 |
| deleteMappingFromVolume..... | 60 |
| deleteTargetGroup..... | 61 |
| getInitiatorGroup..... | 63 |
| getProjectDefaultFcITView..... | 64 |
| getProjectDefaultIscsiITView..... | 66 |
| getVolumeITView..... | 69 |
| initiatorGroupExists..... | 71 |
| listFCInitiators..... | 72 |
| listFCInitiatorGroups..... | 74 |
| listFCTargets..... | 75 |
| listFCTargetGroups..... | 77 |
| listInitiatorGroups..... | 78 |
| listISCSIInitiatorGroups..... | 79 |
| listISCSIInitiators..... | 80 |
| listInitiatorsInInitiatorGroup..... | 82 |
| listISCSITargetGroups..... | 83 |
| listISCSITargets..... | 84 |
| listTargetGroups..... | 86 |
| listTargetsInTargetGroup..... | 87 |
| modifyISCSITargetAlias..... | 88 |
| moveTargetToTargetGroup..... | 90 |
| moveInitiatorToInitiatorGroup..... | 93 |

Chapter 5: Dataset Methods..... 97

| | |
|-------------------------------|-----|
| abortCopy..... | 99 |
| checkPoolIntegrity..... | 100 |
| copyDataset..... | 102 |
| createProject..... | 105 |
| createMappingForProject..... | 107 |
| createShare..... | 111 |
| createShare..... | 113 |
| createVolume..... | 115 |
| deleteDataset..... | 117 |
| deleteDataset..... | 118 |
| deleteMappingFromProject..... | 121 |
| deletePool..... | 123 |
| deleteProject..... | 125 |
| deleteShare..... | 126 |
| deleteShare..... | 128 |
| deleteVolume..... | 130 |
| deleteVolume..... | 131 |
| exportPool..... | 133 |
| getCopyStatus..... | 135 |
| getDatasetSpaceInfo..... | 137 |
| getFloatingIPList..... | 138 |
| getPoolSpaceInfo..... | 141 |

| | |
|---------------------------------|-----|
| getProject..... | 142 |
| getProjectProperty..... | 145 |
| getShare..... | 146 |
| getShareProperty..... | 149 |
| getVolume..... | 151 |
| getVolumeProperty..... | 154 |
| importPool..... | 155 |
| inheritPropertyFromProject..... | 157 |
| isProjectExposedOverNFS..... | 159 |
| isShareExposedOverNFS..... | 161 |
| listAllCopyOperations..... | 162 |
| listLunsById..... | 164 |
| listPools..... | 166 |
| listProjects..... | 167 |
| listRunningCopyOperations..... | 168 |
| listShares..... | 170 |
| listSharesByMountPoints..... | 171 |
| listVolumes..... | 173 |
| modifyProjectProperties..... | 175 |
| modifyShareProperties..... | 177 |
| modifyVolumeProperties..... | 179 |
| resetPoolError..... | 181 |
| setNFSSharingOnProject..... | 182 |
| setNFSSharingOnShare..... | 185 |

Chapter 6: Snapshot Methods.....189

| | |
|--|-----|
| cloneProjectSnapshot..... | 190 |
| cloneReplicaProjectSnapshot..... | 192 |
| cloneReplicaSubProjectSnapshot..... | 194 |
| cloneShareSnapshot..... | 196 |
| cloneVolumeSnapshot..... | 197 |
| cloneVolumeSnapshot..... | 199 |
| createProjectSnapshot..... | 201 |
| createShareSnapshot..... | 202 |
| createSnapshotSchedule..... | 204 |
| createVolumeSnapshot..... | 207 |
| deleteProjectSnapshot..... | 209 |
| deleteShareSnapshot..... | 211 |
| deleteSnapshotSchedule..... | 213 |
| deleteSnapshotSchedules..... | 215 |
| deleteVolumeSnapshot..... | 217 |
| getProjectCloneStatus..... | 219 |
| getProjectSnapshotCreationStatus..... | 220 |
| getShareSnapshotCreationStatus..... | 222 |
| getSnapshotSchedule..... | 224 |
| getVolumeSnapshotCreationStatus..... | 227 |
| inheritSnapshotSettingsFromProject..... | 228 |
| isSnapshotSchedulesInheritedFromProject..... | 230 |

| | |
|---|-----|
| listDependenciesAndSnapshotCountOnDelete..... | 231 |
| listDependenciesAndSnapshotCountOnRollback..... | 233 |
| listSnapshots..... | 235 |
| rollBackToProjectSnapshot..... | 237 |
| rollBackToShareSnapshot..... | 238 |
| rollBackToVolumeSnapshot..... | 239 |

Chapter 7: Replication Methods..... 241

| | |
|-------------------------------|-----|
| getReplicationConfigList..... | 242 |
| getReplicationStatus..... | 243 |
| startReplication..... | 245 |

Chapter 8: System Methods..... 247

| | |
|------------------------------|-----|
| getDiskCount..... | 248 |
| getDisks..... | 248 |
| getDisksByChassis..... | 252 |
| getSMBConfig..... | 254 |
| getUpgradeHistory..... | 255 |
| identifyDisk..... | 256 |
| identifyDiskByIndex..... | 258 |
| isProjectExposedOverSMB..... | 260 |
| isShareExposedOverSMB..... | 261 |
| listSystemProperties..... | 263 |
| setSMBConfig..... | 264 |
| setSMBSharingOnProject..... | 266 |
| setSMBSharingOnShare..... | 269 |

Chapter 9: Network ACL Methods..... 273

| | |
|---|-----|
| addNFSNetworkACLOnProject..... | 274 |
| addNFSNetworkACLOnShare..... | 277 |
| addSMBNetworkACLOnProject..... | 280 |
| addSMBNetworkACLOnShare..... | 284 |
| getNFSNetworkACLsOnProject..... | 287 |
| getNFSNetworkACLsOnShare..... | 289 |
| getSMBNetworkACLsOnProject..... | 291 |
| getSMBNetworkACLsOnShare..... | 293 |
| inheritNetworkACLsettingsFromProject..... | 295 |
| removeAllNFSNetworkACLsOnProject..... | 297 |
| removeAllNFSNetworkACLsOnShare..... | 299 |
| removeAllSMBNetworkACLsOnProject..... | 300 |
| removeAllSMBNetworkACLsOnShare..... | 302 |
| removeNFSNetworkACLOnProject..... | 303 |
| removeNFSNetworkACLOnShare..... | 306 |
| removeSMBNetworkACLOnProject..... | 308 |
| removeSMBNetworkACLOnShare..... | 310 |
| setNFSNetworkACLsOnProject..... | 312 |

| | |
|---------------------------------|-----|
| setNFSNetworkACLsOnShare..... | 315 |
| setSMBNetworkACLsOnProject..... | 319 |
| setSMBNetworkACLsOnShare..... | 323 |

Chapter 10: SNMP Methods.....327

| | |
|--------------------------------|-----|
| addSNMPTrapListener..... | 328 |
| disableSNMPService..... | 330 |
| enableSNMPService..... | 331 |
| getSNMPSettings..... | 333 |
| isSNMPServiceEnabled..... | 334 |
| modifySNMPCommunityString..... | 335 |
| recreateSNMPTables..... | 337 |
| removeSNMPTrapListener..... | 338 |
| resyncSNMPTables..... | 340 |

Chapter 11: Analytics Methods.....343

| | |
|---|-----|
| getOneMinuteSystemAnalyticsHistory..... | 344 |
| getOneMinuteDataAnalyticsHistory..... | 358 |

Chapter 12: Notification Methods.....375

| | |
|-------------------------------------|-----|
| getRecentNotifications..... | 376 |
| getRecentCriticalNotifications..... | 378 |

Chapter 13: Objects..... 383

| | |
|-------------------------------|-----|
| ArrayUpgrade_V2_1..... | 385 |
| CopyDestination_V2_1..... | 385 |
| CopySource_V2_1..... | 385 |
| CopyStatus_V2_1..... | 386 |
| DataAnalyticsResult_V2_3..... | 387 |
| DatasetProperty_V2_1..... | 387 |
| DatasetSpaceInfo_V2_1..... | 387 |
| DatasetStatus..... | 388 |
| Disk_V2_1..... | 389 |
| FCInitiator_V2_1..... | 389 |
| FCTarget_V2_1..... | 389 |
| FloatingIP_V2_2..... | 390 |
| InitiatorGroup_V2_1..... | 390 |
| IsctlInitiator_V1_0..... | 390 |
| ISCSIInitiator_V2_1..... | 391 |
| ISCSITarget_V2_1..... | 392 |
| ISCSITargetCreate_V2_1..... | 393 |
| ITView_V2_1..... | 394 |
| LocalGroup_V1_2..... | 394 |
| LocalUser_V1_2..... | 394 |
| LunStatus..... | 395 |
| NetworkACL_V2_1..... | 396 |

| | |
|--------------------------------------|------------|
| Notification_V2_3..... | 396 |
| Pool_V1_0..... | 397 |
| PoolSpaceInfo_V2_1..... | 397 |
| Project_V1_0..... | 398 |
| Project_V1_2..... | 398 |
| Project_V2_1..... | 398 |
| ProjectCloneProgressStatus_v1_2..... | 402 |
| ReplicationConfig_V1_2..... | 402 |
| ReplicationStatus_v1_2..... | 402 |
| Schedule_V2_1..... | 403 |
| Share_V1_0..... | 405 |
| Share_V2_1..... | 405 |
| ShareOptions..... | 408 |
| SharePermissions..... | 409 |
| SMBConfig_V2_2..... | 409 |
| SnapShotDeletionStatus..... | 410 |
| SnapshotProgressStatus..... | 410 |
| SnapshotSchedule_V2_1..... | 410 |
| SNMP_Setting_V2_1..... | 410 |
| SystemAnalyticsResult_V2_3..... | 411 |
| TargetGroup_V2_2..... | 411 |
| Volume_V1_0..... | 411 |
| Volume_V2_1..... | 412 |
| UserACL (Read Only) v2.1..... | 414 |
| Chapter 14: Enumerations..... | 415 |
| CLEANUP_STATUS..... | 416 |
| CLONE_PROGRESS_STATUS..... | 416 |
| COMMAND_STATUS..... | 416 |
| Mode_enum..... | 417 |
| OVERWRITE_STATUS..... | 417 |
| Permission_type_enum..... | 417 |
| Replication_Scope_Option..... | 418 |
| SNAPSHOT_DELETION_STATUS..... | 418 |
| SNAPSHOT_PROGRESS_STATUS..... | 419 |
| State..... | 419 |
| ZEBI_SYSTEM_PROPERTY..... | 420 |
| Appendix A: Appendix A..... | 421 |
| JSON Quick Reference..... | 422 |
| Appendix B: Appendix B..... | 425 |
| Deprecated APIs..... | 426 |
| createSnapshots..... | 426 |
| deleteSnapshots..... | 428 |
| deleteSnapshots..... | 430 |

cloneSnapshot.....432

Preface

Audience

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

The IntelliFlash *API Reference* is intended for developers and solution engineers creating applications using the IntelliFlash REST APIs.

IntelliFlash Documentation

The following table lists the technical documentation types available for the IntelliFlash arrays.

Table 1: IntelliFlash Product Documentation

| Name | Description |
|---|--|
| IntelliFlash N-Series All-Flash Array Hardware Guide | Contains detailed descriptions, hardware specifications, and rack installation instructions for the N-Series NVMe storage arrays. |
| IntelliFlash T-Series and HD-Series Arrays Hardware Guide | Contains detailed descriptions, hardware specifications, and rack installation instructions for the T-Series and HD-Series storage arrays. |
| IntelliFlash User Guide | Contains detailed instructions on how to configure, use, and administer IntelliFlash arrays. |
| IntelliFlash API Reference | Contains detailed information about the IntelliFlash REST APIs. |
| IntelliFlash Release Notes | Provides details on enhancements, fixed issues, and known issues for a release. |

Document Revision History

| Date | Description |
|------------|---|
| 11/29/2018 | <p>Update for the new APIs included in IntelliFlash REST APIs version 2.3. IntelliFlash 3.9.x.x uses version 2.3 of the APIs.</p> <p>The following new APIs are added:</p> <ul style="list-style-type: none">• getOneMinuteDataAnalyticsHistory• getOneMinuteSystemAnalyticsHistory• getRecentNotifications• getRecentCriticalNotifications |
| 07/16/2018 | <p>Update for the new APIs included in IntelliFlash REST APIs version 2.2. IntelliFlash 3.7.1.0 uses version 2.2 of the APIs.</p> <p>Document changes include:</p> <ul style="list-style-type: none">• Chapter 1: Introduction to the IntelliFlash API [updated the API Version History topic with new and updated APIs introduced in version 2.2].• Chapter 5: Dataset Methods [added new APIs introduced in version 2.2: moveInitiatorToInitiatorGroup, listTargetGroups, getSMBConfig, setSMBConfig, and getFloatingIPList. Updated moveTargetToTargetGroup, getDisks identifyDisk, and identifyDiskByIndex].• Chapter 11: Objects [updated Share v2_1. Added TargetGroup_V2_2]. |
| 11/17/2017 | <p>Document changes include:</p> <ul style="list-style-type: none">• Chapter 1: Introduction to the IntelliFlash API [updated; includes a new topic, API Version History.]• Chapter 11: Objects [updated Project_V2_1]• Throughout guide: Added "First Available Version" section in all the API method descriptions. |

| Date | Description |
|------------|--|
| 10/10/2017 | <p>Update for the new APIs included in IntelliFlash REST API version 2.1. IntelliFlash 3.7.0.x uses version 2.1 of the IntelliFlash APIs.</p> <p>Document changes include:</p> <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 4: SAN Methods [updated] • Chapter 5: Dataset Methods [updated] • Chapter 6: Snapshot Methods [updated] • Chapter 8: System Methods [updated] • Chapter 9: Network ACL Methods [new] • Chapter 10: SNMP Methods [new] • Chapter 11: Objects [updated] |
| 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 in the "Using the API Examples" chapter. |
| 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 ../v1/... to ../v2/.. for all the APIs. |

Chapter 1

Introduction to the IntelliFlash Rest APIs

Topics:

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

The IntelliFlash Rest APIs provide an interface to securely configure and provision storage on IntelliFlash using a programming or scripting language. It enables you to integrate IntelliFlash 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.

API Version History

| APIs | Introduced in API Version | First Available IntelliFlash Version |
|--|-----------------------------------|---|
| getOneMinuteDataAnalyticsHistory getOneMinuteSystemAnalyticsHistory getRecentNotifications getRecentCriticalNotifications | 2.3 | IntelliFlash 3.9.0.0 |
| moveInitiatorToInitiatorGroup listTargetGroups getSMBConfig setSMBConfig getFloatingIPList | 2.2 | IntelliFlash 3.7.1.0 |
| moveTargetToTargetGroup getDisks identifyDisk identifyDiskByIndex Share v2_1 object | 2.1 (introduced) 2.2 (updated) | IntelliFlash 3.7.0.x (introduced) IntelliFlash 3.7.1.0 (updated) |
| abortCopy addNFSNetworkACLOnProject addNFSNetworkACLOnShare addSMBNetworkACLOnProject addSMBNetworkACLOnShare addSNMPTrapListener checkPoolIntegrity cloneReplicaProjectSnapshot cloneReplicaSubProjectSnapshot copyDataset | 2.1 | IntelliFlash 3.7.0.x |

| APIs | Introduced in API Version | First Available IntelliFlash Version |
|--|---------------------------|--------------------------------------|
| <i>createFCInitiator</i> <i>createISCSITarget</i> <i>createISCSITargetForGroup</i> <i>createMappingForVolume</i> <i>createProject</i> <i>createMappingForProject</i> <i>createSnapshotSchedule</i> <i>createTargetGroup</i> | 2.1 | IntelliFlash 3.7.0.x |
| <i>deleteDataset</i> <i>deleteInitiatorGroup</i> <i>deleteISCSIInitiator</i> <i>deleteISCSITarget</i> <i>deleteMappingFromProject</i> <i>deletePool</i> <i>deleteProject</i> <i>deleteShare</i> <i>deleteSnapshotSchedule</i> <i>deleteSnapshotSchedules</i> <i>deleteTargetGroup</i> <i>deleteVolume</i> <i>disableSNMPService</i> <i>enableSNMPService</i> <i>exportPool</i> | 2.1 | IntelliFlash 3.7.0.x |

| APIs | Introduced in API Version | First Available IntelliFlash Version |
|--|------------------------------|---|
| <i>getCopyStatus</i> <i>getDatasetSpaceInfo</i> <i>getDiskCount</i> <i>getDisksByChassis</i> <i>getNFSNetworkACLsOnProject</i> <i>getNFSNetworkACLsOnShare</i> <i>getPoolSpaceInfo</i> <i>getProject</i> <i>getProjectDefaultFclTVView</i> <i>getProjectDefaultIscsiTVView</i> <i>getProjectProperty</i> <i>getShare</i> <i>getShareProperty</i> <i>getSMBNetworkACLsOnProject</i> <i>getSMBNetworkACLsOnShare</i> <i>getSnapshotSchedule</i> <i>getSNMPSettings</i> <i>getUpgradeHistory</i> <i>getVolume</i> <i>getVolumeITView</i> <i>getVolumeProperty</i> | 2.1 | IntelliFlash 3.7.0.x |

| APIs | Introduced in API Version | First Available IntelliFlash Version |
|--|---------------------------|--------------------------------------|
| <i>importPool</i> <i>inheritNetworkACLsettingsFromProject</i> <i>inheritPropertyFromProject</i> <i>inheritSnapshotSettingsFromProject</i> <i>isProjectExposedOverNFS</i> <i>isProjectExposedOverSMB</i> <i>isShareExposedOverNFS</i> <i>isShareExposedOverSMB</i> <i>isSnapshotSchedulesInheritedFromProject</i> <i>isSNMPServiceEnabled</i> | 2.1 | IntelliFlash 3.7.0.x |
| <i>listAllCopyOperations</i> <i>listDependenciesAndSnapshotCountOnDelete</i> <i>listDependenciesAndSnapshotCountOnRollback</i> <i>listFCInitiators</i> <i>listFCTargets</i> <i>listInitiatorGroups</i> <i>listISCSIInitiators</i> <i>listISCSITargets</i> <i>listRunningCopyOperations</i> <i>listSharesByMountPoints</i> <i>modifyISCSITargetAlias</i> <i>modifyProjectProperties</i> <i>modifyShareProperties</i> <i>modifySNMPCommunityString</i> <i>modifyVolumeProperties</i> | 2.1 | IntelliFlash 3.7.0.x |

| APIs | Introduced in API Version | First Available IntelliFlash Version |
|--|---------------------------|--|
| <i>recreateSnmpTables</i> <i>removeAllNFSNetworkACLsOnProject</i> <i>removeAllNFSNetworkACLsOnShare</i> <i>removeAllSMBNetworkACLsOnProject</i> <i>removeAllSMBNetworkACLsOnShare</i> <i>removeNFSNetworkACLOnProject</i> <i>removeNFSNetworkACLOnShare</i> <i>removeSMBNetworkACLOnProject</i> <i>removeSMBNetworkACLOnShare</i> <i>removeSNMPTrapListener</i> | 2.1 | IntelliFlash 3.7.0.x |
| <i>resetPoolError</i> <i>resyncSnmpTables</i> <i>setNFSNetworkACLsOnProject</i> <i>setNFSNetworkACLsOnShare</i> <i>setNFSSharingOnProject</i> <i>setNFSSharingOnShare</i> <i>setSMBNetworkACLsOnProject</i> <i>setSMBNetworkACLsOnShare</i> <i>setSMBSharingOnProject</i> <i>setSMBSharingOnShare</i> | 2.1 | IntelliFlash 3.7.0.x |
| <i>recreateSNMPTables</i> <i>resyncSNMPTables</i> | 2.1 | IntelliFlash 3.7.0.x IntelliFlash 3.5.4.0 |
| <i>rollBackToProjectSnapshot</i> <i>rollBackToShareSnapshot</i> <i>rollBackToVolumeSnapshot</i> | 2.0 | IntelliFlash 3.5.0.1 |
| <i>cloneVolumeSnapshot</i> | 2.0 | IntelliFlash 3.5.0.0 |

| APIs | Introduced in API Version | First Available IntelliFlash Version |
|--|---------------------------|--------------------------------------|
| <i>changeUserPassword</i> <i>cloneProjectSnapshot</i> <i>cloneShareSnapshot</i> <i>cloneVolumeSnapshot</i> <i>cloneVolumeSnapshot</i> <i>createGroup</i> <i>createProjectSnapshot</i> <i>createShareSnapshot</i> <i>createUser</i> <i>createUserAndGroup</i> <i>createVolumeSnapshot</i> | 1.2 | IntelliFlash 2.1.2.1 |
| <i>deleteGroup</i> <i>deleteProjectSnapshot</i> <i>deleteShareSnapshot</i> <i>deleteUser</i> <i>deleteVolumeSnapshot</i> <i>getProjectCloneStatus</i> <i>getProjectSnapshotCreationStatus</i> <i>getReplicationConfigList</i> <i>getReplicationStatus</i> <i>getShareSnapshotCreationStatus</i> <i>getVolumeSnapshotCreationStatus</i> <i>listGroups</i> <i>listUsers</i> <i>startReplication</i> | 1.2 | IntelliFlash 2.1.2.1 |

| APIs | Introduced in API Version | First Available IntelliFlash Version |
|---|---------------------------|--------------------------------------|
| <i>addInitiatorToInitiatorGroup</i> <i>cloneSnapshot</i> <i>createInitiatorGroup</i> <i>createIscsiInitiator</i> <i>createMappingForVolume</i> <i>createSnapshots</i> <i>createVolume</i> <i>deleteDataset</i> <i>deleteMappingFromVolume</i> <i>deleteSnapshots</i> <i>deleteVolume</i> <i>getInitiatorGroup</i> <i>initiatorGroupExists</i> | 1.0 | IntelliFlash 2.1.0.0 |
| <i>listFCInitiatorGroups</i> <i>listFCTargetGroups</i> <i>listInitiatorsInInitiatorGroup</i> <i>listISCSIInitiatorGroups</i> <i>listISCSITargetGroups</i> <i>listLunsById</i> <i>listPools</i> <i>listProjects</i> <i>listShares</i> <i>listSnapshots</i> <i>listSystemProperties</i> <i>listTargetsInTargetGroup</i> <i>listVolumes</i> | 1.0 | IntelliFlash 2.1.0.0 |

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.
- View history of IntelliFlash installations and upgrades to the array.
- Delete, export, import, or expand pools
- Create or delete projects
- View, modify, or manage projects and dataset properties
- Add, set, or modify ACLs on projects and shares
- Create or move target groups
- Clone replica datasets
- Obtain IT views on SAN volumes and projects
- View space info for pools and datasets
- Create or delete mappings for projects and volumes
- View or identify disks
- View, create, inherit, or delete snapshot schedules
- View, create, or delete SAN targets, target groups, and initiators

- Enable or disable SNMP services
- Add or remove SNMP trap listeners
- Copy datasets, list or abort any ongoing copy operations
- Promote clones before deleting a share or LUN

Unified APIs

IntelliFlash REST APIs run on and obtain results from both controllers in an IntelliFlash 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 are unified starting from API version 2.0. Previous versions of the IntelliFlash REST APIs—version 1.0 and version 1.2—could access only one controller in an API request.

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. |

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

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.

IntelliFlash 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.

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 an IntelliFlash 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 IntelliFlash 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.

Creating a Volume and Exposing It

Prerequisites

You can create a volume and expose it using the IntelliFlash API. To accomplish this, you must first ensure that the IntelliFlash 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 IntelliFlash 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 IntelliFlash 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 IntelliFlash array.

Chapter 3

User and Group Methods

Topics:

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

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

changeUserPassword

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

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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"
}
```

createGroup

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

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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.

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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"
}
```

createUser

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

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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 "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.

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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"
```

```
}
```

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.

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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"
}
```

deleteGroup

Deletes the specified user group (a "Local Group" that was created on the IntelliFlash 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.

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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"
}
```

deleteUser

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



Warning: The delete operation is not reversible.

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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"
}
```

listGroups

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

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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"]
}
]
```

listUsers

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

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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  
  }  
]
```

Chapter 4

SAN Methods

Topics:

- [*addInitiatorToInitiatorGroup*](#)
- [*createFCInitiator*](#)
- [*createInitiatorGroup*](#)
- [*createIscsiInitiator*](#)
- [*createISCSITarget*](#)
- [*createISCSITargetForGroup*](#)
- [*createMappingForVolume*](#)
- [*createMappingForVolume*](#)
- [*createTargetGroup*](#)
- [*deleteInitiatorGroup*](#)
- [*deleteISCSInitiator*](#)
- [*deleteISCSITarget*](#)
- [*deleteMappingFromVolume*](#)
- [*deleteTargetGroup*](#)
- [*getInitiatorGroup*](#)
- [*getProjectDefaultFcITView*](#)
- [*getProjectDefaultIscsiITView*](#)
- [*getVolumeITView*](#)
- [*initiatorGroupExists*](#)
- [*listFCInitiators*](#)
- [*listFCInitiatorGroups*](#)
- [*listFCTargets*](#)
- [*listFCTargetGroups*](#)
- [*listInitiatorGroups*](#)
- [*listISCSInitiatorGroups*](#)
- [*listISCSInitiators*](#)
- [*listInitiatorsInInitiatorGroup*](#)
- [*listISCSITargetGroups*](#)
- [*listISCSITargets*](#)
- [*listTargetGroups*](#)
- [*listTargetsInTargetGroup*](#)
- [*modifyISCSITargetAlias*](#)
- [*moveTargetToTargetGroup*](#)
- [*moveInitiatorToInitiatorGroup*](#)

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

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.

First Available Version

API v1.0, IntelliFlash 2.1.0.0

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"
}
```

createFCInitiator

Creates the Fibre Channel (FC) Initiator on the IntelliFlash Array.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[listFCInitiators](#), [deleteISCSIInitiator](#), [listInitiatorGroups](#), [createISCSIInitiator](#)

Parameters

initiatorName

The name of the fibre channel initiator you want to create. This must be in the format `wwn.<16 hex digits>`. For example, `wwn.5001438001FF7742`.

initiatorGroupName

The name of the fibre channel initiator group to add the initiator as a member.

Returns

Returns an integer status where:

- 0 indicates that the request succeeded.

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

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified initiator group is not found.

EZEBI_RESOURCE_EXIST

This exception is thrown if the specified initiator already exists.

EZEBI_INVALID_ARGUMENT

This exception is thrown for the following conditions:

- If the parameters are invalid (bad format or blank)
- If the initiator group is over 512 characters long
- If the initiator group has invalid characters such as *,#,/,\,!,@,~,(),[],{,},=, and %.
- If the initiator name is not in the format of wwn.<16 alpha-numeric-hex characters>

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["wwn.5001438001ffbbbb", "fc-igroup"]' \
https://198.51.100.10/zebi/api/v2/createFCInitiator -k
```

Response

The above request returns the HTTP status code 200 (OK) and an integer status of 0 to indicate that the initiator was created in the specified initiator group.

Example 2

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
```



```
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["wwn.5001438001ffbfff", "fc-igroup"]' \
https://198.51.100.10/zebi/api/v2/createFCInitiator -k
```

Error Response

In this example, the fibre channel initiator already exists. So the request returns the HTTP status code 400 (bad request) and the following response:

```
{"code":"EZEBI_RESOURCE_EXIST","details":"","message":"The Fibre Channel initiator [wwn.5001438001ffbfff] already exists, but must not exist for this operation [create initiator].","extendedData":{}}
```

createInitiatorGroup

Creates an initiator group on an IntelliFlash array.

First Available Version

API v1.0, IntelliFlash 2.1.0.0

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"
}
```

createIscsiInitiator

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

First Available Version

API v1.0, IntelliFlash 2.1.0.0

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":"iqn.2012-10.com.acme:test1",
      "chapUserName":"","chapSecret":""}]' \
https://198.51.100.10/zebi/api/v2/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":"","chapUserName":"","chapSecret":""}]' \
https://198.51.100.10/zebi/api/v2/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": {}
}
```

createISCSITarget

Creates an iSCSI target with the specified target group, chap authentication, alias, and network bindings.

This API allows the user to specify all the attributes of the target in the JSON *ISCSITargetCreate_V2_1* parameter object passed.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

createTargetGroup, *moveTargetToTargetGroup*, *createISCSITargetForGroup*, *deleteISCSITarget*, *modifyISCSITargetAlias*, *listISCSITargets*

Parameters

iscsiTarget

A JSON object of type *ISCSITargetCreate_V2_1* that contains the attributes of the iSCSI target to create.

The JSON fields of the iSCSI target object parameter are

- targetSuffixName
- targetAlias
- targetGroupName
- targetAuthenticationMode
- targetChapName
- targetChapSecret
- targetNetworkBinding

The target suffix name is the user specified portion after the iqn target name colon character. For example, iqn.2012-02.com.tegile:myTargetSuffix.

Specify only a target suffix and not the full target name because the iSCSI target prefix is pre-defined for all targets on the array. The alias is the user friendly alternate name of the target.

The authentication mode is a value of 'none', 'chap', or 'mutual'. The chapName and chapSecret must be set depending on whether the authentication mode is set. Only if 'mutual' authentication is set, the chapName and chapSecret must be specified. The network binding (IP:PORT) can be any of the bindings

associated with the target group name's pool. The default target group cannot be modified or used here.

Returns

Returns an integer status where:

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

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified iSCSI target group does not exist.

EZEBI_RESOURCE_EXIST

This exception is thrown if the specified target already exists.

EZEBI_INVALID_ARGUMENT

This exception is thrown for the following conditions:

- If the parameters are invalid (bad format or blank)
- If the default group is being modified

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '[{
  "targetSuffixName": "test",
  "targetAlias": "testAlias",
  "targetGroupName": "iscsi-tgroup",
  "targetAuthenticationMode": "none",
  "targetChapName": "",
  "targetChapSecret": "",
  "targetNetworkBinding": [
    "10.68.97.211:3260"
  ] } ]' \
https://198.51.100.10/zebi/api/v2/createISCSITarget -k
```

Response

The above request returns the HTTP status code 200 (OK) and an integer status of 0 to indicate that the target "iqn.2012-02.com.tegile:test" was created in the specified target group "iscsi-tgroup".

Example 2

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '[{
  "targetSuffixName": "test3",
  "targetAlias": "testAlias",
  "targetGroupName": "iscsi-tgroup",
  "targetAuthenticationMode": "invalidauth",
  "targetChapName": "",
  "targetChapSecret": "",
  "targetNetworkBinding": [
    "10.68.97.211:3260"
  ] } ]' \
https://198.51.100.10/zebi/api/v2/createISCSITarget -k
```

Error Response

In this example, the chap authentication is an invalid type value. So the request returns the HTTP status code 400 (bad request) and the following response:

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "createISCSITarget.arg0.targetAuthenticationMode value
'invalidauth':
  Chap authentication should be one of the values:none, chap, or mutual
  createISCSITarget.arg0 value

'com.tegile.skywalk.api.v2.IPublicAPI_V2_1$ISCSITargetCreate_V2_1@45b06d12':
  Invalid CHAP authentication passed[invalidauth],
  supported values are: none, chap, or mutual.",
  "message":
  "Chap authentication should be one of the values:none, chap, or mutual
  Invalid CHAP authentication passed[invalidauth],
  supported values are: none, chap, or mutual.",
  "extendedData": {
    "EX_CAUSE_MESSAGE": null
  }
}
```

createISCSITargetForGroup

Creates an iSCSI target for the specified target group.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[createTargetGroup](#), [moveTargetToTargetGroup](#), [createISCSITarget](#)

Parameters

targetNameSuffix

The iSCSI target name suffix.

This is the portion of the iSCSI iqname that comes after the colon. This is the user specified part of the target name.

targetAlias

The iSCSI alias user friendly name used to refer to the target.

This is typically the same as the target suffix.

targetGroupName

Existing iSCSI target group that the new target will be a member of.

New targets created here cannot be added to the default target group. So the name here cannot be the default target group.

Returns

Returns an integer status where:

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

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified target group does not exist.

EZEBI_RESOURCE_EXIST

This exception is thrown if the specified target already exists.

EZEBI_INVALID_ARGUMENT

This exception is thrown for the following conditions:

- If the parameters are invalid (bad format or blank)
- If the target group has invalid characters such as *,#,/,\,!,@,~,(),[,]{,},=, and %.
- If the target group is the default target group
- If the target group is over 512 characters long

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["myTargetSuffix", "myTargetAlias","myTargetGroup" ]' \
https://198.51.100.10/zebi/api/v2/createISCSITargetForGroup -k
```

Response

The above request returns the HTTP status code 200 (OK) and an integer status of 0 to indicate that the target "iqn.2012-02.com.tegile:myTargetSuffix" was created in the specified target group "myTargetGroup".

Example 2

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["myTargetSuffix", "myTargetAlias","group-does-not-exist-yet" ]' \
https://198.51.100.10/zebi/api/v2/createISCSITargetForGroup -k
```

Error Response

In this example, the target group does not exist. So the request returns the HTTP status code 400 (bad request) and the following response:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
```



```

"details": "Failed to find target group [group-does-not-exist-yet].",
"message": "Failed to setup iSCSI target for creation on target group:
group-does-not-exist-yet",
"extendedData":
{
  "EX_CAUSE_MESSAGE":
  "Failed to find target group [group-does-not-exist-yet].",
  "EX_CAUSE_CODE_NAME": "EZEBI_RESOURCE_NOT_FOUND"
}
}

```

createMappingForVolume

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

First Available Version

API v1.0, IntelliFlash 2.1.0.0

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.

createMappingForVolume

Creates a mapping for a volume, with the option for it to be read only.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[createMappingForProject](#), [deleteMappingFromProject](#)

Parameters**datasetPath**

Path of the LUN. The format is <poolName>/Local/<projectName>/<lunName>.

This operation is not allowed for Replica datasets.

initiatorGroupName

Name of the initiator group in the existing mapping.

targetGroupName

Name of the target group in the existing mapping.

LUNNumber

LUN number to use. Pass a value of -1 if you want the system to assign an available LUN number.

readOnly

Whether mapping is read-only. The values are True and False.

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown**EZEBI_RESOURCE_NOT_FOUND**

This exception is thrown if the specified project, initiator group, or target group cannot be found.

EZEBI_INVALID_ARGUMENT

This exception is thrown for any of the following reasons:

- The path specified belongs to a replica dataset.
- The volume protocol does not match the protocol of the SAN groups.
- The mapping does not exist.
- The mapping for this target group already exists (when trying to make All mapping).
- The mapping with 'All' initiator groups already exists on this target group for this LUN.

Examples**Example 1**

Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Local/demoProject/demoLUN", "All", "demoiSCSITargetGroup", 1, true
]' \
https://198.51.100.10/zebi/api/v2/createMappingForVolume -k
```

Response

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

Example 2

Erroneous Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Local/demoProject/demoLUN", "All", "default-pool-a-iscsi-target-
group", 1, true
]' \
https://198.51.100.10/zebi/api/v2/createMappingForVolume -k
```

Error Response

In this example, the request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "",
  "message": "View already exists with different read only value: true",
  "extendedData": {}
}
```

Example 3

Erroneous Request

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Local/demoProject/demoLUN", "demoFCInitiatorGroup",
    "demoiSCSITargetGroup", 1, true
```

```
}', \
https://198.51.100.10/zebi/api/v2/createMappingForVolume -k
```

Error Response

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

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "",
  "message": "Cannot create mapping: volume protocol does not match
protocol
of specified initiator group.",
  "extendedData": {}
}
```

createTargetGroup

Creates a target group with the specified target group name in the resource group containing the named pool.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[moveTargetToTargetGroup](#), [listFCTargetGroups](#), [listISCSITargetGroups](#)

Parameters

targetGroupName

Name of the new target group.

poolName

Specifies the pool resource group to place the target group.

The poolName applies to iSCSI target groups only. For FC target groups, specify an empty string or null.

Returns

Returns an integer status where:

- 0 indicates that the request succeeded.
- 1 indicates that the request failed.

Exceptions Thrown**EZEBI_RESOURCE_NOT_FOUND**

This exception is thrown if the specified pool cannot be found.

EZEBI_INVALID_ARGUMENT

This exception is thrown if an invalid character is detected in either the targetGroupName or poolName.

Examples**Example 1****Request (curl)**

```
curl -X POST \
  -H 'authorization: Basic AUTH_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '["aCoolTargetGroup", "pool63a"]' \
  https://198.51.100.10/zebi/api/v2/createTargetGroup -k
```

Response

The above request returns the HTTP status code 200 (OK) and 0 indicating a successful request to create an iSCSI target group.

Example 2**Erroneous Request (curl)**

```
curl -X POST \
  -H 'authorization: Basic AUTH_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '["aBad#TargetGroup", "pool63a"]' \
  https://198.51.100.10/zebi/api/v2/createTargetGroup -k
```

Error Response

The above request contains invalid characters. So the request returns HTTP status code 400 (bad request) and the following JSON exception:

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "createTargetGroup.arg0 value
'aBad#TargetGroup': Target group name cannot have
special characters like *,#,/,\,!,@,~,,(,),[,],{,},=,%...",
  "message":
  "Target group name cannot have special characters like
*,#,/,\,!,@,~,,(,),[,],{,},=,%..."
```

```

    "extendedData": {
      "EX_CAUSE_MESSAGE": null
    }
  }
}

```

deleteInitiatorGroup

Deletes the initiator group specified from the IntelliFlash array.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[listInitiatorGroups](#), [deleteISCSIInitiator](#), [createInitiatorGroup](#)

Parameters

initiatorGroupName

The name of the initiator group you want to delete.

Returns

Returns an integer status where:

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

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified initiator group is not found.

EZEBI_INVALID_ARGUMENT

This exception is thrown for the following conditions:

- If the parameters are invalid (bad format or blank)
- If the initiator group is over 512 characters long

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["myInitiatorGroupToDelete"]' \
https://198.51.100.10/zebi/api/v2/deleteInitiatorGroup -k
```

Response

The above request returns the HTTP status code 200 (OK) and an integer status of 0 to indicate that the initiator group 'myInitiatorGroupToDelete' was deleted.

Example 2

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["myGroupToDeleteDoesNotExist"]' \
https://198.51.100.10/zebi/api/v2/deleteInitiatorGroup -k
```

Error Response

In this example, the initiator group does not exist. So the request returns the HTTP status code 404 (not found) and the following response:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "Failed to delete initiator group myGroupToDeleteDoesNotExist :
    STMF_ERROR_NOT_FOUND",
  "message": "Error detected deleting initiator group
    [myGroupToDeleteDoesNotExist].",
  "extendedData": {
    "EX_CAUSE_MESSAGE": "Failed to delete initiator group myGroupToDeleteDoesNotExist :
      STMF_ERROR_NOT_FOUND",
    "EX_CAUSE_CODE_NAME": "com.tegile.solaris.lib.stmf.StmfException:
      Failed to delete initiator group myGroupToDeleteDoesNotExist :
      STMF_ERROR_NOT_FOUND",
    "EX_CAUSE_CODE_NUMBER": "32770"
  }
}
```

deleteISCSIInitiator

Deletes an iSCSI initiator with the specified initiator name.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[deleteISCSIInitiator](#), [listISCSIInitiatorGroups](#), [listISCSITargets](#), [createISCSIInitiator](#)

Parameters

initiatorName

The name of the initiator you want to delete. For example, iqn.1991-05.com.microsoft:test.tegile.com. The 'initiatorName' is returned by listISCSIInitiators API.

Values returned

Returns an integer status where:

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

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified iSCSI target does not exist.

EZEBI_INVALID_ARGUMENT

This exception is thrown for the following conditions:

- If the parameters are invalid (bad format or blank)
- If the default group is being modified

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '["iqn.1991-05.com.microsoft:test.tegile.com"]' \
https://198.51.100.10/zebi/api/v2/deleteISCSIInitiator -k
```

Response

The above request returns the HTTP status code 200 (OK) and an integer status of 0 to indicate that the target "iqn.2012-02.com.tegile:test" was deleted.

Example 2

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["iqn.1991-05.com.microsoft:doesnotexist.tegile.com"]' \
https://10.68.97.100/zebi/api/v2/deleteISCSIInitiator -k
```

Error Response

In this example, the initiator does not exist. So the request returns the HTTP status code 404 (not found) and the following response:

```
{ "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "iSCSI Initiator was not found.",
  "message": "Failed to find iSCSI Initiator to delete
[iqn.1991-05.com.microsoft:doesnotexist.tegile.com].",
  "extendedData": { "EX_CAUSE_MESSAGE": "iSCSI Initiator was not found.",
    "EX_CAUSE_CODE_NAME": "EZEBI_RESOURCE_NOT_FOUND" } }
```

deleteISCSITarget

Deletes an iSCSI target with the specified target name.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[createTargetGroup](#), [moveTargetToTargetGroup](#), [createISCSITargetForGroup](#), [createISCSITarget](#), [modifyISCSITargetAlias](#), [listISCSITargets](#)

Parameters

targetName

The name of the target you want to delete. For example, iqn.2012-02.com.tegile:myTargetSuffix.

The targetName is returned by listISCSITargets API. Targets in default target group cannot be deleted.

Returns

Returns an integer status where:

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

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified iSCSI target does not exist.

EZEBI_INVALID_ARGUMENT

This exception is thrown for the following conditions:

- If the parameters are invalid (bad format or blank)
- If the default group is being modified

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '["iqn.2012-02.com.tegile:test"]' \
https://198.51.100.10/zebi/api/v2/deleteISCSITarget -k
```

Response

The above request returns the HTTP status code 200 (OK) and an integer status of 0 to indicate that the target "iqn.2012-02.com.tegile:test" was deleted.

Example 2

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["iqn.2012-02.com.tegile:doesnotexist"]' \
https://10.68.97.100/zebi/api/v2/deleteISCSITarget -k
```

Error Response

In this example, the target does not exist. So the request returns the HTTP status code 404 (not found) and the following response:

```
{ "code": "EZEBI_RESOURCE_NOT_FOUND", "details": "",
  "message": "Failed to find iSCSI
  Target[iqn.2012-02.com.tegile:doesnotexist].
  The target must exist for this operation to succeed.",
  "extendedData": {} }
```

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.

First Available Version

API v1.0, IntelliFlash 2.1.0.0

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"
}
```

deleteTargetGroup

Deletes the target group specified from the IntelliFlash array.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[createTargetGroup](#), [moveTargetToTargetGroup](#), [createISCSITarget](#)

Parameters

targetGroupName

The name of the target group you want to delete.

Returns

Returns an integer status where:

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

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified target group is not found.

EZEBI_INVALID_ARGUMENT

This exception is thrown for the following conditions:

- If the parameters are invalid (bad format or blank)
- If the target group is over 512 characters long
- If the target group is the default target group

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["myTargetGroupToDelete"]' \
https://198.51.100.10/zebi/api/v2/deleteTargetGroup -k
```

Response:

The above request returns the HTTP status code 200 (OK) and an integer status of 0 to indicate that the target group 'myTargetGroupToDelete' was deleted.

Example 2

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["myTargetGroupToDeleteDoesNotExist"]' \
https://198.51.100.10/zebi/api/v2/deleteTargetGroup -k
```

Error Response

In this example, the target group does not exist. So the request returns the HTTP status code 404 (not found) and the following response:

```
{"code":"EZEBI_RESOURCE_NOT_FOUND","details":"","message":"Failed to find target group [myTargetGroupToDeleteDoesNotExist].","extendedData":{}}
```

getInitiatorGroup

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

First Available Version

API v1.0, IntelliFlash 2.1.0.0

Related APIs

[listInitiatorsInInitiatorGroup](#), [createScsiInitiator](#), [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.

getProjectDefaultFclTVView

Lists all of the existing default Fibre Channel views given a project's dataset path.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[getVolumeITView](#), [getProjectDefaultIscsiITView](#)

Parameters

datasetPath

A string representing the project dataset path.

A project dataset must start with a letter followed by a series of alpha-numeric. It can contain only the following characters: 'A-Z', 'a-z', '0-9', '_', ':', '-', '.', and '/'. The dataset, however, should not end with '/'.

Returns

A list of [ITView_V2_1](#) objects.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

If the specified project cannot be found.

EZEBI_INVALID_ARGUMENT

If an invalid character is detected in the project dataset path.

Examples

Example 1

Request (curl)

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

Response

The above request returns the HTTP status code 200 (OK) and a list of the [ITView_V2_1](#) objects:

```
[
  {
    "hostGroupName": "newfc",
    "targetGroupName": "default-fc-target-group",
    "lunNbr": -1,
    "readOnly": false
  }
]
```

Example 2

Erroneous Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN' \
```

```
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '["pool63b/Local/nonExistentTestProject"]' \
https://198.51.100.10/zebi/api/v2/getProjectDefaultFcITView -k
```

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "Cannot find the specified project 'pool63b/Local/nonExistentTestProject'.",
  "extendedData": {}
}
```

Example 3

Erroneous Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '["pool63b/Lo+cal/testProject"]' \
https://198.51.100.10/zebi/api/v2/getProjectDefaultFcITView -k
```

Error Response

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

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "getProjectDefaultFcITView.arg0
value 'pool63b/Lo+cal/testProject':
Dataset path is not in the expected format that starts with a letter
followed by alpha-numeric and may contain only the following:
'A-Z', 'a-z', '0-9', '_', '.', '-', ':', and '/' but not end with '/'.",
  "message": "Dataset path is not in the expected format that starts
with a letter followed by alpha-numeric and may contain only the
following:
'A-Z', 'a-z', '0-9', '_', '.', '-', ':', and '/' but not end with '/'.",
  "extendedData": {
    "EX_CAUSE_MESSAGE": null
  }
}
```

getProjectDefaultIscsiITView

Returns all the existing default iSCSI views for the specified project dataset path.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[getVolumeITView](#), [getProjectDefaultFcITView](#)

Parameters

datasetPath

A string representing the project dataset path.

A project dataset must start with a letter followed by a series of alphanumeric. It can contain only the following characters: 'A-Z', 'a-z', '0-9', '_', ':', '-', '.', and '/'. The dataset, however, should not end with '/'.

Returns

A list of [ITView_V2_1](#) objects.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified project cannot be found.

EZEBI_INVALID_ARGUMENT

This exception is thrown if an invalid character is detected in the project dataset path.

Examples

Example 1

Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '{"pool-test/Local/proj_snap"}' \
https://198.51.100.10/zebi/api/v2/getProjectDefaultIscsiITView -k
```

Response

The above request returns the HTTP status code 200 (OK) and a list of [ITView_V2_1](#) objects if found.

```
{
  "hostGroupName": "All",
```

```

    "targetGroupName": "default-pool-test-iscsi-target-group",
    "lunNbr": -1,
    "readOnly": false
  }
]

```

Example 2

Erroneous Request (curl)

```

curl -X POST \
-H 'authorization: Basic Auth TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '["pool63b/Local/nonExistentTestProject"]' \
https://198.51.100.10/zebi/api/v2/getProjectDefaultIscsiITView -k

```

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```

{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "Cannot find the specified project
'pool63b/Local/nonExistentTestProject'.",
  "extendedData": {}
}

```

Example 3

Erroneous Request (curl)

```

curl -X POST \
-H 'authorization: Basic AUTH_TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '["pool63b/Lo+cal/testProject"]' \
https://198.51.100.10/zebi/api/v2/getProjectDefaultIscsiITView -k

```

Error Response

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

```

{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "getProjectDefaultIscsiITView.arg0
value 'pool63b/Lo+cal/testProject':
Dataset path is not in the expected format that starts with a letter
followed by alpha-numeric and may contain only the following:
'A-Z', 'a-z', '0-9', '_', '.', '-', ':', and '/' but not end with '/'.",

```

```

    "message": "Dataset path is not in the expected format that starts
    with a letter followed by alpha-numeric and may contain only the
    following:
    'A-Z', 'a-z', '0-9', '_', '.', '-', ':', and '/' but not end with '/'.",
    "extendedData": {
        "EX_CAUSE_MESSAGE": null
    }
}

```

getVolumeITView

Lists all the existing default iSCSI views for the specified volume dataset path.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[getProjectDefaultIscsiITView](#), [getProjectDefaultFcITView](#)

Parameters

datasetPath

A string representing the volume dataset path.

A volume dataset must start with a letter followed by a series of alpha-numeric characters. It can contain only the following characters: 'A-Z', 'a-z', '0-9', '_', '.', '-', ':', and '/'. The dataset, however, should not end with '/'.

Returns

A list of [ITView_V2_1](#) objects.

Exceptions Thrown

EZEBI_INVALID_ARGUMENT

This exception is thrown if an invalid character is detected in the volume dataset path.

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified volume doesn't exist.

Examples

Example 1

Request (curl)

```
curl -X POST \
-H 'authorization: Basic AUTH_TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '["pool63b/Local/testProject/LUN1"]' \
https://198.51.100.10/zebi/api/v2/getVolumeITView -k
```

Response

The above request returns the HTTP status code 200 (OK) and a list of [ITView_V2_1](#) objects.

```
[
  {
    "hostGroupName": "All",
    "targetGroupName": "iqn.2012-02.com.tegile:repz61-pool63b-testproject-
group",
    "lunNbr": 0,
    "readOnly": false
  }
]
```

Example 2

Erroneous Request (curl)

```
curl -X POST \
-H 'authorization: Basic AUTH_TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '["pool63b/Local/testProject/nonExistentLun"]' \
https://198.51.100.10/zebi/api/v2/getVolumeITView -k
```

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "Cannot find the specified volume
'pool63b/Local/testProject/nonExistentLun'.",
  "extendedData": {}
}
```

Example 3

Erroneous Request (curl)

```
curl -X POST \
-H 'authorization: Basic AUTH_TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '["pool63b/Lo+cal/testProject/nonExistentLun"]' \
```

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

Error Response

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

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "getVolumeITView.arg0 value
'pool63b/lo+cal/testProject/nonExistentLun':
Dataset path is not in the expected format that starts with a
letter followed by alpha-numeric and may contain only the following:
'A-Z', 'a-z', '0-9', '_', '.', '-', ':', and '/' but not end with '/'.",
  "message": "Dataset path is not in the expected format that starts with
a letter followed by alpha-numeric and may contain only the following:
'A-Z', 'a-z', '0-9', '_', '.', '-', ':', and '/' but not end with '/'.",
  "extendedData": {
    "EX_CAUSE_MESSAGE": null
  }
}
```

initiatorGroupExists

Checks if an initiator group exists on the IntelliFlash array.

First Available Version

API v1.0, IntelliFlash 2.1.0.0

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
```

listFCInitiators

Lists the FC initiators on the IntelliFlash array.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[createFCInitiator](#), [listFCInitiatorGroups](#), [listInitiatorsInInitiatorGroup](#), [listTargetsInTargetGroup](#)

Parameters

initiatorNamePattern

The name of the FC initiator you want to retrieve, or a regular expression pattern to retrieve more than one FC initiator name.

For example:

- "wwn.21000024FF279210" lists a specific initiator.
- ".*" lists all Fibre Channel initiators.

- "wwn.123.*" lists FC initiators starting with wwn.123.
- ".*89112" lists initiators ending with 89112.

Returns

A JSON array of the [FCInitiator_V2_1](#) objects that contain the details of the iSCSI initiators that currently exist.

Exceptions Thrown

EZEBI_INVALID_ARGUMENT

This exception is thrown if the parameters are invalid (bad format, blank, or an invalid regular expression).

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '[".*"]' \
https://198.51.100.10/zebi/api/v2/listFCInitiators -k
```

Response:

The above request returns the HTTP status code 200 (OK) and an array of JSON [FCInitiator_V2_1](#) objects for all the FC initiators found on the array. For example:

```
[
  {
    "initiatorName": "wwn.5001438001FF7742",
    "initiatorGroupName": "fc-igroup"
  },
  {
    "initiatorName": "wwn.5001438001FFBBBA",
    "initiatorGroupName": "fc-igroup"
  }
]
```

Example 2

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["wwn.5001438001FCCCCC"]' \
https://198.51.100.10/zebi/api/v2/listFCInitiators -k
```

Error Response

In this example, the initiator pattern requested does not exist. So the request returns HTTP status code 200 (OK) and an empty array of JSON [FCInitiator_V2_1](#) objects.

Example 3

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '[""]' \
https://198.51.100.10/zebi/api/v2/listIFCInitiators -k
```

Error Response

The above request returns the HTTP status code 400 (bad request) and the following response:

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "listFCInitiators.arg0 value '':
Pattern name is not valid, an invalid null, empty or blank pattern name
was detected.",
  "message": "Pattern name is not valid,
an invalid null, empty or blank pattern name was detected.",
  "extendedData": {
    "EX_CAUSE_MESSAGE": null
  }
}
```

listFCInitiatorGroups

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

First Available Version

API v1.0, IntelliFlash 2.1.0.0

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 IntelliFlash 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"
]
```

listFCTargets

Lists the Fibre Channel (FC) targets on the IntelliFlash array.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[listTargetsInTargetGroup](#), [createTargetGroup](#), [moveTargetToTargetGroup](#)

Parameters

targetNamePattern

The name of the FC target you want to retrieve, or a regular expression pattern to retrieve more than one FC target name.

For example:

- "wwn.1234567890123456" lists a specific FC target.

- `".*"` retrieves all the FC targets.
- `"wwn.210.*"` retrieves targets starting with `wwn.210`.
- `".*89112"` retrieves targets ending with `89112`.

Returns

A JSON array of the [FCTarget_V2_1](#) objects that contains the details of the available FC targets.

Exceptions Thrown

EZEBI_INVALID_ARGUMENT

This exception is thrown if the parameters are invalid (bad format, blank, or an invalid regular expression).

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '[".*"]' \
https://198.51.100.10/zebi/api/v2/listFCTargets -k
```

Response

The above request returns the HTTP status code 200 (OK) and an array of JSON [FCTarget_V2_1](#) objects for all the FC targets found on the array. For example:

```
[
  {
    "targetName": "wwn.21000042BB326F3C",
    "targetStatus": "online",
    "targetNode": "controller-a",
    "targetGroupName": "default-fc-target-group",
    "targetSpeed": "4Gb",
    "targetPortType": "HBA"
  },
  {
    "targetName": "wwn.31C5A0BE4F7D7463",
    "targetStatus": "offline",
    "targetNode": "controller-b",
    "targetGroupName": "",
    "targetSpeed": "not established",
    "targetPortType": "NPIV"
  }
]
```

```
    },
  ]
}
```

Example 2

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["wwn.5001438001FCCCCC"]' \
https://198.51.100.10/zebi/api/v2/listFCTargets -k
```

Error Response

In this example, the target pattern requested does not exist. So the request returns HTTP status code 200 (OK) and an empty array of the JSON *FCTarget_V2_1* objects.

Example 3

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '[""]' \
https://198.51.100.10/zebi/api/v2/listIFCTargets -k
```

Error Response

The above request returns the HTTP status code 400 (bad request) and the following response:

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "listFCTargets.arg0 value '':
Pattern name is not valid, an invalid null,
empty or blank pattern name was detected.",
  "message": "Pattern name is not valid, an invalid null,
empty or blank pattern name was detected.",
  "extendedData": {
    "EX_CAUSE_MESSAGE": null
  }
}
```

listFCTargetGroups

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

First Available Version

API v1.0, IntelliFlash 2.1.1.1

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"
]
```

listInitiatorGroups

Lists all initiator groups available on the IntelliFlash array.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[deleteInitiatorGroup](#)

Parameters

None

Returns

A list of [InitiatorGroup_V2_1](#) objects.

Example**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/listInitiatorGroups -k
```

Response

The above request returns the HTTP status code 200 (OK) and a list of [InitiatorGroup_V2_1](#) objects if found.

```
[
  {
    "initiatorGroupName": "esx124group",
    "intendedProtocol": "iSCSI"
  },
  {
    "initiatorGroupName": "esx125group",
    "intendedProtocol": "iSCSI"
  }
]
```

listISCSIInitiatorGroups

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

First Available Version

API v1.0, IntelliFlash 2.1.0.0

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 IntelliFlash 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"
]
```

listISCSIInitiators

Lists the iSCSI initiator details on the IntelliFlash array.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[deletescsilinitiator](#), [listISCSIInitiatorGroups](#), [listInitiatorsInInitiatorGroup](#), [listISCSITargets](#), [createscsilinitiator](#)

Parameters

initiatorNamePattern

The name of the iSCSI initiator you want to retrieve, or a regular expression pattern to retrieve more than one iSCSI initiators.

For example, "iqn.1991-05.com.microsoft:myHost" lists a specific initiator, ".*" retrieves all the initiators, "iqn.1991-05.com.microsoft:.*" retrieves initiators with that prefix, ".*myHost.*" retrieves initiators that contain 'myHost' in the name.

Returns

A JSON array of the [ISCSIInitiator_V2_1](#) objects that contain the details of the iSCSI initiators that currently exist.

Exceptions Thrown

EZEBI_INVALID_ARGUMENT

This exception is thrown if the parameters are invalid (bad format, blank, or an invalid regular expression).

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '[".*"]' \
https://10.68.89.10/zebi/api/v2/listISCSIInitiators -k
```

Response:

The above request returns the HTTP status code 200 (OK) and an array of JSON [ISCSIInitiator_V2_1](#) objects for all the ISCSI initiators found on the array. For example:

```
[
  {
    "initiatorName": "eui.1543553633737272",
    "initiatorGroupName": "",
    "chapUserName": null,
    "chapSecret": null
  },
  {
    "initiatorName": "iqn.1991-05.com.microsoft:a.tegile.com",
    "initiatorGroupName": "iscsi-igroup1",
    "chapUserName": "myChapName",
    "chapSecret": null
  }
]
```

Example 2

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["iqn.2012-02.com.tegile:does-not-exist"]' \
https://10.68.89.10/zebi/api/v2/listISCSIInitiators -k
```

Error Response

In this example, the initiator pattern requested does not exist. So the request returns HTTP status code 200 (OK) and an empty array of JSON [iSCSIInitiator_V2_1](#) objects.

Example 3

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN"
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["*() (&^%$"]' \
https://10.68.89.10/zebi/api/v2/listISCSIInitiators -k
```

Error Response

The above request returns the HTTP status code 400 (bad request) and the following response:

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "listISCSIInitiators.arg0 value `*() (&^%$`:
Invalid regular expression argument was detected.
Please check the regular expression syntax.",
  "message": "Invalid regular expression argument was detected.
Please check the regular expression syntax.",
  "extendedData": {
    "EX_CAUSE_MESSAGE": null
  }
}
```

listInitiatorsInInitiatorGroup

Lists all initiators belonging to the specified initiator group.

First Available Version

API v1.0, IntelliFlash 2.1.0.0

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.

listISCSITargetGroups

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

First Available Version

API v1.0, IntelliFlash 2.1.0.0

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"
]
```

listISCSITargets

Lists the iSCSI target details on the IntelliFlash array.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[createTargetGroup](#), [moveTargetToTargetGroup](#), [createISCSITarget](#), [createISCSITargetForGroup](#), [deleteISCSITarget](#)

Parameters**targetNamePattern**

The name of the target you want to retrieve, or a target name regular expression pattern to retrieve a list of targets.

For example, "iqn.2012-12.com.acme:atarget" lists a specific target, ".*" retrieves all the iSCSI targets, "iqn.2012-12.com.tegile:.*" retrieves targets with the prefix iqn.2012-12.com.tegile, ".*pool-a.*" retrieves targets with 'pool-a' in the name.

Returns

A JSON array of [ISCSITarget_V2_1](#) objects that contain the details of the iSCSI targets.

Exceptions Thrown

EZEBI_INVALID_ARGUMENT

This exception is thrown if the parameters are invalid (bad format, blank, or an invalid regular expression).

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '[".*"]' \
https://10.68.89.10/zebi/api/v2/listISCSITargets -k
```

Response:

The above request returns the HTTP status code 200 (OK) and a JSON array of [ISCSITarget_V2_1](#) objects for all the iSCSI targets found on the array. For example:

```
[ {
  "targetSuffixName": "test",
  "targetAlias": "iscsi-alias",
  "targetGroupName": "iscsi-tgroup",
  "targetAuthenticationMode": "none",
  "targetChapName": "",
  "targetChapSecret": null,
  "targetNetworkBinding": [
    "10.68.97.211:3260"
  ],
  "targetName": "iqn.2012-02.com.tegile:test"
},
{
  "targetSuffixName": "pool-demo",
  "targetAlias": "pool-demo",
  "targetGroupName": "default-pool-demo-iscsi-target-group",
  "targetAuthenticationMode": "none",
  "targetChapName": "",
  "targetChapSecret": null,
  "targetNetworkBinding": [
    "10.68.97.212:3260"
  ],
  "targetName": "iqn.2012-02.com.tegile:pool-demo"
} ]
```

Example 2

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["ign.2012-02.com.tegile:no-exist"]' \
https://10.68.89.10/zebi/api/v2/listISCSITargets -k
```

Error Response

In this example, the target pattern requested does not exist. So the request returns HTTP status code 200 (OK) and an empty array of JSON ISCSITarget_V2_1 objects for ISCSI targets found on the array.

listTargetGroups

Lists all target groups available on the IntelliFlash array.

If no targets are mapped to a target group, the intendedProtocol is shown as "Unknown."

First Available Version

API v2.2, IntelliFlash 3.7.1.0

Related APIs

[createTargetGroup](#), [listISCSITargetGroups](#), [listFCTargetGroups](#)

Parameters

None

Returns

A list of [TargetGroup_V2_2](#) objects.

Example

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/listTargetGroups -k
```

Response

The above request returns the HTTP status code 200 (OK) and a list of [TargetGroup_V2_2](#) objects:

```
[
  {
    "targetGroupName": "default-fc-target-group",
    "intendedProtocol": "FC"
  },
  {
    "targetGroupName": "default-pool-a-iscsi-target-group",
    "intendedProtocol": "iSCSI"
  },
  {
    "targetGroupName": "default-pool-a-virtual-fc-target-group",
    "intendedProtocol": "FC"
  },
  {
    "targetGroupName": "iqn.2012-02.com.tegile:trayambah.a-pool-a-proj1-group",
    "intendedProtocol": "Unknown"
  }
]
```

listTargetsInTargetGroup

Lists all targets associated with the target group.

First Available Version

API v1.0, IntelliFlash 2.1.0.0

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.

modifyISCSITargetAlias

Modifies the iSCSI target alias (or user friendly name) of the target to newly specified name.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

Parameters

targetName

The name of the target you want to modify. For example, "iqn.2012-02.com.tegile:myTargetSuffix". The targetName is returned by listISCSITargets API. The targets in default target group cannot be modified.

targetAlias

The new target alias name.

Returns

Returns an integer status where:

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

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified iSCSI target does not exist.

EZEBI_INVALID_ARGUMENT

This exception is thrown for the following conditions:

- If the parameters are invalid (bad format or blank)
- If the default target is being modified

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["iqn.2012-02.com.tegile:test","rename-this-alias"]' \
https://10.68.89.10/zebi/api/v2/modifyISCSITargetAlias -k
```

Response

The above request returns the HTTP status code 200 (OK) and an integer status of 0 to indicate that the user friendly alias of the target "iqn.2012-02.com.tegile:test" has been renamed to "rename-this-alias".

Example 2

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["iqn.2012-02.com.tegile:does-not-exist","new-alias"]' \
https://10.68.89.10/zebi/api/v2/modifyISCSITargetAlias -k
```

Error Response

In this example, the target does not exist. So the request returns the HTTP status code 404 (not found) and the following response:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "Failed to find iSCSI Target[iqn.2012-02.com.tegile:does-not-exist].",
  "message": "The target must exist for this operation to succeed.",
  "extendedData": {
    "EX_CAUSE_MESSAGE": "Failed to find iSCSI Target[iqn.2012-02.com.tegile:does-not-exist]. The target must exist for this operation to succeed.",
    "EX_CAUSE_CODE_NAME": "EZEBI_RESOURCE_NOT_FOUND"
  }
}
```

moveTargetToTargetGroup

Moves an iSCSI or Fibre Channel target to an existing target group.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Updated in API v2.2.

Related APIs

[createTargetGroup](#), [listFCTargetGroups](#), [listISCSITargetGroups](#), [listTargetGroups](#), [createISCSITarget](#), [listTargetsInTargetGroup](#)

Parameters

targetName

Name of the target you want to move. This can be an iSCSI or Fibre Channel target name.

For iSCSI target name, it must be in the format, **iqn.yyyy-mm.**

[reverse-domain-name]:unique-user-suffix. For example,
iqn.2012-02.com.tegile:myTargetPool1Suffix.

For Fibre Channel target, it must be in the format, **wwn.[16-hexadecimal-numbers]**. For example, **wwn.21000024FF199A22**.

destinationTargetGroupName

Name of the destination target group. This must be an existing target group.

The protocol (iSCSI or FC) of the target being moved must match the protocol of the destination target group, or the target group must be empty. The iSCSI targets must match the resource group of the destination target resource group. The target cannot be moved to or from the default target group. If the target is already in a target group, the API removes the target from that target group.

force

Boolean value indicating to bypass in-use target check.



Note: This parameter is included in API v2.2.

If the value is true, the API moves the target even if it is in an active project or LUN mapping. If the value is false, API throws an exception when the target is being used in a project or LUN mapping.

Returns

Returns an integer status where:

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

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the target or target group does not exist.

EZEBI_INVALID_ARGUMENT

This exception is thrown for the following conditions:

- If an invalid character is detected in either the targetName or the destinationTargetGroupName.
- If the target group has a different protocol or belongs to a different resource group than the target.
- If an attempt is made to move a target to or from the default target groups.

EZEBI_GENERAL

This exception is thrown if an internal error is detected.

EZEBI_RESOURCE_BUSY

This exception is thrown if the target is already in a group that is in an active LUN or project mapping.

Examples

Exampe 1

Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '["iqn.2012-02.com.tegile:testtarget1", "aCoolTargetGroup", false]' \
  https://198.51.100.10/zebi/api/v2/moveTargetToTargetGroup -k
```

Response

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

Example 2

Erroneous Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '["iqn.2012-02.com.tegile:testtarget1", "aNonExistentTargetGroup",
false]' \
  https://198.51.100.10/zebi/api/v2/moveTargetToTargetGroup -k
```

Erroneous Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "Cannot find target group: aNonExistentTargetGroup",
  "extendedData": {}
}
```

Example 3

Erroneous Request (curl)

```
curl -X POST \
```

```
-H 'authorization: Basic AUTH TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '["iqn.2012-02.com.tegile:testtarget1", "emptyTargetGroup", false]' \
https://198.51.100.10/zebi/api/v2/moveTargetToTargetGroup -k
```

Erroneous Response

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

```
{
  "code": "EZEBI_RESOURCE_BUSY",
  "details": "Unable to remove the target safely
from its current active Target Group. Please remove 'is-tgroup-active'
from its existing project and LUN mapping(s) and try again.
Can't remove the Target because the group [is-tgroup-active]
which contains the target is being used by 1 lun(s)",
  "message": "Failed to move target [iqn.2012-02.com.tegile:target1]
to the target group [emptyTargetGroup]",
  "extendedData": {
    "EX_CAUSE_MESSAGE": "Unable to remove the Target safely from its
current active Target Group. Please remove 'is-tgroup-active'
from its existing project and LUN mapping(s) and try again.
Can't remove the Target because the group [is-tgroup-active]
which contains the target is being used by 1 lun(s)",
    "EX_CAUSE_CODE_NAME": "EZEBI_RESOURCE_BUSY"
  }
}
```

moveInitiatorToInitiatorGroup

Moves an iSCSI or Fibre Channel initiator to an existing initiator group.

First Available Version

API v2.2, IntelliFlash 3.7.1.0

Related APIs

[listISCSIInitiators](#), [listInitiatorsInInitiatorGroup](#), [listISCSIInitiatorGroups](#), [listInitiatorGroups](#), [createIscsiInitiator](#), [addInitiatorToInitiatorGroup](#), [createFCInitiator](#), [createInitiatorGroup](#)

Parameters

initiatorName

Name of the initiator you want to move.

This can be an iSCSI or Fibre Channel initiator in any of the following formats:

- `iqn.yyyy-mm.[reverse-domain-name]:unique-user-suffix`

- `eui.[16-hexadecimal-digits]`
- `wwn.[16-hexadecimal-numbers]`

For example, `iqn.2017-12.com.tegile.dev:test`,
`eui.1543553633737279` or `wwn.5001438002211672`.

destinationInitiatorGroupName

Name of the destination initiator group. This must be an existing initiator group.

The protocol (iSCSI or FC) of the initiator being moved must match the protocol of the destination initiator group, or the initiator group must be empty. If the initiator is already in an initiator group, the API removes the initiator from that initiator group.

force

Boolean value indicating to bypass in-use initiator check.

If the value is true, the API moves the initiator even if it is in an active project or LUN mapping. If the value is false, API throws an exception when the initiator is being used in a project or LUN mapping.

Returns

Returns an integer status where:

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

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the initiator or initiator group does not exist.

EZEBI_INVALID_ARGUMENT

This exception is thrown for the following conditions:

- If an invalid character is detected in either the `initiatorName` or the `destinationInitiatorGroupName` parameters.
- If the initiator group has a different protocol than the initiator.

EZEBI_GENERAL

This exception is thrown if an internal error is detected.

EZEBI_RESOURCE_BUSY

This exception is thrown if the initiator is already in a group that is in an active LUN or project mapping.

Examples

Exampe 1

Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '["iqn.2012-02.com.tegile:testInitiator1", "emptyInitiatorGroup",
false]' \
https://198.51.100.10/zebi/api/v2/moveInitiatorToInitiatorGroup -k
```

Response

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

Example 2

Erroneous Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '["iqn.2012-02.com.tegile:testInitiator1", "NonExistentInitiatorGroup",
false]' \
https://198.51.100.10/zebi/api/v2/moveInitiatorToInitiatorGroup -k
```

Erroneous Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI RESOURCE NOT FOUND",
  "details": "Please specify an existing iSCSI or Fibre Channel initiator
group.",
  "message": "Initiator group [nonExistentInitiatorGroup] to move
the initiator into was not found.",
  "extendedData": {}
}
```

Example 3

Erroneous Request (curl)

```
curl -X POST \
```

```
-H 'authorization: Basic YOUR_TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '["iqn.2017-10.com.tegile.dev:initiator1", "empty-is-igroup", false]' \
https://YOUR_IP/zebi/api/v2/moveInitiatorToInitiatorGroup -k
```

Erroneous Response

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

```
{
  "code": "EZEBI_RESOURCE_BUSY",
  "details": "Unable to remove initiator safely
from its current active initiator group. Please remove inuse-is-igroup
from its existing project and LUN mapping(s) and try again.
Can't remove the Initiator because the group [inuse-is-igroup]
which contains the initiator is being used by 1 Lun(s)",
  "message": "Failed to move initiator
[iqn.2017-10.com.tegile.dev:initiator1]
to the initiator group [empty-is-igroup]",
  "extendedData":
  {
    "EX_CAUSE_MESSAGE":
    "Unable to remove initiator safely from
its current active initiator group. Please remove
inuse-is-igroup from its existing project and LUN mapping(s) and try
again.
Can't remove the Initiator because the group [inuse-is-igroup]
which contains the initiator is being used by 1 Lun(s)",
    "EX_CAUSE_CODE_NAME": "EZEBI_RESOURCE_BUSY"
  }
}
```

Chapter 5

Dataset Methods

Topics:

- *abortCopy*
- *checkPoolIntegrity*
- *copyDataset*
- *createProject*
- *createMappingForProject*
- *createShare*
- *createShare*
- *createVolume*
- *deleteDataset*
- *deleteDataset*
- *deleteMappingFromProject*
- *deletePool*
- *deleteProject*
- *deleteShare*
- *deleteShare*
- *deleteVolume*
- *deleteVolume*
- *exportPool*
- *getCopyStatus*
- *getDatasetSpaceInfo*
- *getFloatingIPList*
- *getPoolSpaceInfo*
- *getProject*
- *getProjectProperty*
- *getShare*
- *getShareProperty*
- *getVolume*
- *getVolumeProperty*
- *importPool*
- *inheritPropertyFromProject*
- *isProjectExposedOverNFS*
- *isShareExposedOverNFS*
- *listAllCopyOperations*
- *listLunsById*
- *listPools*
- *listProjects*
- *listRunningCopyOperations*

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

- [*listShares*](#)
- [*listSharesByMountPoints*](#)
- [*listVolumes*](#)
- [*modifyProjectProperties*](#)
- [*modifyShareProperties*](#)
- [*modifyVolumeProperties*](#)
- [*resetPoolError*](#)
- [*setNFSSharingOnProject*](#)
- [*setNFSSharingOnShare*](#)

abortCopy

Aborts a running copy operation.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[copyDataset](#), [getCopyStatus](#), [listRunningCopyOperations](#), [listAllCopyOperations](#)

Parameters

taskGUID

The GUID of the copy operation task to be aborted.

Returns

Returns the HTTP status code 200 (OK) and return code 0.

Exceptions Thrown

EZEBI_GENERAL

This exception is thrown if the task GUID is invalid for the array.

Examples

Example 1

Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '["europa:84944c8a-8bc1-4705-809e-144363797a57"]' \
  https://198.51.100.10/zebi/api/v2/abortCopy -k
```

Response

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

Example 2

Erroneous Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '["europa1:84944c8a-8bc1-4705-809e-144363797a57"]'
https://198.51.100.10/zebi/api/v2/abortCopy -k
```

Error Response

The above request returns HTTP status code 400 with the following message:

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "",
  "message": "Cannot find copy task
'europa1:84944c8a-8bc1-4705-809e-144363797a57'
in pool: europa1.",
  "extendedData": {}
}
```

checkPoolIntegrity

Validates the integrity of the disks associated with a pool.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[importPool](#), [exportPool](#), [resetPoolError](#)

Parameters

poolName

Name of the pool where you want to clear the errors.

start

An integer which specifies whether to start (1) or stop (0) the pool integrity check.

Returns

Returns an integer status where:

- 0 indicates that the request succeeded.
- 1 indicates that the request failed.

Exceptions Thrown**EZEBI_RESOURCE_NOT_FOUND**

This exception is thrown if the pool does not exist.

EZEBI_INVALID_ARGUMENT

This exception is thrown if an invalid character is detected in the pool name or a start value is not 0 or 1.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples**Example 1****Request (curl)**

```
curl -X POST \
  -H 'authorization: Basic AUTH_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '["pool6a", 1]' \
  https://198.51.100.10/zebi/api/v2/checkPoolIntegrity -k
```

Response

The above request returns the HTTP status code 200 (OK) and 0 as integer indicating a successful operation.

Example 2**Erroneous Request (curl)**

```
curl -X POST \
  -H 'authorization: Basic AUTH_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '["nonExistentPool6a", 1]' \
  https://198.51.100.10/zebi/api/v2/checkPoolIntegrity -k
```

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "Pool not found.",
  "message": "The pool [nonExistentPool6a] was not found.",
  "extendedData": {
    "EX_CAUSE_MESSAGE": "Pool not found.",
    "EX_CAUSE_CODE_NAME": "EZEBI_RESOURCE_NOT_FOUND"
  }
}
```

```
}
```

Example 3

Erroneous Request (curl)

```
curl -X POST \
  -H 'authorization: Basic AUTH_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '["pool63a", 0]' \
  https://198.51.100.10/zebi/api/v2/checkPoolIntegrity -k
```

Error Response

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

```
{
  "code": "EZEBI_GENERAL",
  "details": "",
  "message": "Failed to check pool [pool63a] integrity.
  This could be due to an invalid scan pool state transition request",
  "extendedData": {}
}
```

copyDataset

Starts a copy operation.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[getCopyStatus](#), [listRunningCopyOperations](#), [listAllCopyOperations](#), [abortCopy](#)

Parameters

[CopySource_V2_1](#)

[CopyDestination_V2_1](#)

Returns

A task GUID for the copy operation.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified source dataset is not available.

EZEBI_INVALID_ARGUMENT

This exception is thrown for the following conditions:

- If an invalid character is present in the CopySource_V2_1 or the CopyDestination_V2_1 field.
- If the specified source dataset is not available.
- If the dataset names are invalid.
- If the host is invalid.
- If the start and end subproject suffix numbers are invalid.

Examples

Example 1

Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[{"poolName": "europa",
"projectName": "images",
"subProjectName": "template"
},
{
"hostName": "10.68.132.120",
"poolName": "napa",
"projectName": "remotecopy",
"subProjectNamePrefix": "prod",
"subProjectNameNumberStart": 1,
"subProjectNameNumberEnd": 3,
"subProjectNameWildcard": ""
}
]' \
https://198.51.100.10/zebi/api/v2/copyDataset -k
```

Response

The above request returns the HTTP status code 200 (OK) and a string. The returned string is the copy operation's GUID:

```
'84944c8a-8bc1-4705-809e-144363797a57'
```

Example 2

Erroneous Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '{"poolName": "europal",
    "projectName": "images",
    "subProjectName": "template"
  },
  {
    "hostName": "10.68.132.120",
    "poolName": "napa",
    "projectName": "remotecopy",
    "subProjectNamePrefix": "prod",
    "subProjectNameNumberStart": 1,
    "subProjectNameNumberEnd": 3,
    "subProjectNameWildcard": ""
  }
]' \
https://198.51.100.10/zebi/api/v2/copyDataset -k
```

Error Response

The above request returns HTTP status code 400 with the following message:

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "",
  "message": "Unable to find project images in europal",
  "extendedData": {}
}
```

Example 3

Erroneous Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '{"poolName": "europal@",
    "projectName": "images",
    "subProjectName": "template"
  },
  {
    "hostName": "10.68.132.120",
    "poolName": "napa",
    "projectName": "remotecopy",
    "subProjectNamePrefix": "prod",
    "subProjectNameNumberStart": 1,
    "subProjectNameNumberEnd": 3,
    "subProjectNameWildcard": ""
  }
]' \
https://198.51.100.10/zebi/api/v2/copyDataset -k
```


Error Response

The above request returns the HTTP status code 400 and the following message:

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "copyDataset.arg0.poolName value 'europal@':
Pool name is not in the expected format
that starts with a letter followed by alpha-numeric,_,-....",
  "message": "Pool name is not in the expected format
that starts with a letter followed by alpha-numeric,_,-....",
  "extendedData": {
    "EX_CAUSE_MESSAGE": null
  }
}
```

createProject

Creates a generic project with given parameters.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[getProject](#), [modifyProjectProperties](#), [getProjectProperty](#)

Parameters

Project

A [Project_V2_1](#) object for which project needs to be created.

If the parameters are not specified, default values are used. “poolName”, “projectName” are mandatory. You can configure other general properties in [Project_V2_1](#) that are not marked as “read only”.

Returns

Returns an integer status where:

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

Exceptions Thrown

EZEBI_INVALID_ARGUMENT

If the parameters are invalid (when the key-value pair is not supported by [Project_V2_1](#)).

Examples

Example 1

Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  {
    "projectName":"general-project",
    "poolName":"pool-a",
    "quotaInByte":42949672960,
    "intendedProtocolList":["iSCSI", "NFS"],
    "compression":"gzip-9",
    "recordSize":"128Kb",
    "dedup":"on"
  }
]'
https://198.51.100.10/zebi/api/v2/createProject -k
```

Response

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

Example 2

Erroneous Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  {
    "projectName":"general-project",
    "poolName":"pool-a",
    "quotaInByte":42949672960,
    "intendedProtocolList":["iSCSI", "NFS"],
    "compression":"really-bad-compression-algorithm",
    "recordSize":"128Kb",
    "dedup":"on"
  }
]'
https://198.51.100.10/zebi/api/v2/createProject -k
```

Error Response

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

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "createProject.arg0 value
'com.tegile.skywalk.api.v2.IPublicAPI_V2_1$Project_V2_1@1778db80':
Invalid Compression Algorithm, disable compression by passing in 'Off'
or enable compression by using one of the following: Lzjb, Gzip-2, Gzip,
Gzip-9, Lz4.",
  "message": "Invalid Compression Algorithm, disable compression by passing
in 'Off'
or enable compression by using one of the following: Lzjb, Gzip-2, Gzip,
Gzip-9, Lz4.",
  "extendedData": {
    "EX_CAUSE_MESSAGE": null
  }
}
```

createMappingForProject

Creates a project-level default mapping.

First Available Version

API v2.1, IntelliJFlash 3.7.0.x

Related APIs

[deleteMappingFromProject](#), [createMappingForVolume](#)

Parameters

datasetPath

Path of the project. The format is <poolName/Local/<projectName>.

This operation is not allowed for replica project datasets.

initiatorGroupName

Name of the initiator group.

targetGroupName

Name of the target group.

readOnly

Whether mapping is read-only. The values are True and False.

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success or if the desired view already exists.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified project, initiator group, or target group cannot be found.

EZEBI_INVALID_ARGUMENT

This exception is thrown for any of the following reasons:

- The path specified belongs to a replica dataset.
- The target group and initiator group have different protocols.
- The target group and initiator group are either iSCSI or FC protocols (unknown).
- The view already exists with the wrong read-only type.
- The mapping for this target group already exists (when trying to make All mapping).
- The mapping with 'All' initiator groups already exists on this target group.

Examples

Example 1

Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/demoProject", "All", "default-pool-a-iscsi-target-group",
    true
  ]' \
  https://198.51.100.10/zebi/api/v2/createMappingForProject -k
```

Response

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

Example 2

Erroneous Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/demoProject", "demoiSCSIInitiatorGroup",
    "demoiSCSITargetGroup", true
  ]' \
  https://198.51.100.10/zebi/api/v2/createMappingForProject -k
```

```
}' \
https://198.51.100.10/zebi/api/v2/createMappingForProject -k
```

Error Response

In this example, the request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
  "code":"EZEBI_INVALID_ARGUMENT",
  "details":"",
  "message":"Specified initiator and target groups are not the same
protocol.
Initiator protocol: Unknown. Target protocol: iSCSI",
  "extendedData":{}
}
```

Example 3

Erroneous Request

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Replica/replicaProject", "All", "default-pool-a-iscsi-target-
group", true
]' \
https://198.51.100.10/zebi/api/v2/createMappingForProject -k
```

Error Response

In this example, the request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
  "code":"EZEBI_INVALID_ARGUMENT",
  "details":"createMappingForProject.arg0 value 'pool-a/Replica/
replicaProject':
Local dataset path expected. For example, valid formats are 'pool-name/
Local/project-name'
or 'pool-name/Local/project-name/share-or-lun-name'.",
  "message":"Local dataset path expected. For example, valid formats are
'pool-name/Local/project-name' or 'pool-name/Local/project-name/share-
or-lun-name'.",
  "extendedData":{"EX_CAUSE_MESSAGE":null}
}
```

Example 4

Erroneous Request

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Replica/replicaProject", "All", "default-pool-a-iscsi-target-
    group", true
  ]' \
  https://198.51.100.10/zebi/api/v2/createMappingForProject -k
```

Error Response

In this example, the request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "createMappingForProject.arg0 value 'pool-a/Replica/
replicaProject':
    Local dataset path expected.
    For example, valid formats are 'pool-name/Local/project-name'
  or
    'pool-name/Local/project-name/share-or-lun-name'.",
  "message": "Local dataset path expected.
    For example, valid formats are 'pool-name/Local/project-name'
  or
    'pool-name/Local/project-name/share-or-lun-name'.",
  "extendedData": {"EX_CAUSE_MESSAGE": null}
}
```

Example 5

Erroneous Request

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/demoProject", "demoFCInitiatorGroup", "demoiSCSITargetGroup",
    true
  ]' \
  https://198.51.100.10/zebi/api/v2/createMappingForProject -k
```

Error Response

In this example, the request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
```

```

    "code": "EZEBI_INVALID_ARGUMENT",
    "details": "",
    "message": "Specified initiator and target groups are not the same
protocol.          Initiator protocol: FC. Target protocol: iSCSI",
    "extendedData": {}
}

```

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.

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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.

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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.

First Available Version

API v1.0, IntelliFlash 2.1.0.0

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.

First Available Version

API v1.0, IntelliFlash 2.1.0.0

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"
}
```

deleteDataset

Promotes dependents as specified, and deletes the dataset whose path is supplied as an argument.

Dependents are deleted as well if promote is false.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[deleteShare](#), [deleteVolume](#)

Parameters**datasetPath**

This string uniquely identifies the dataset on the IntelliFlash array.

recursive

Defines whether to remove the dependants (clones for example) and LUN of the dataset before trying to delete it. Deletion might fail if there are dependants or a LUN exists on a dataset.

errorIfNotExist

Defines whether to raise an exception if the datasetPath to delete doesn't exist.

promote

Defines whether to promote dependents.

Returns

COMMAND_STATUS.COMMAND_SUCCEED(0) on success.

Exceptions Thrown**EZEBI_RESOURCE_NOT_FOUND**

This error is thrown if the specified dataset doesn't exist.

EZEBI_RESOURCE_INUSE

This error is thrown for the following conditions:

- If replication is currently running
- If replication configuration exists and the project to delete is local
- If deleting replication configuration fails

EZEBI_RESOURCE_SUSPENDED

This error is thrown if the pool is suspended.

EZEBI_INVALID_ARGUMENT

This error is thrown for the following conditions:

- If the dataset is empty
- If the dataset length is greater than the maximum length
- If the dataset contains invalid characters

EZEBI_GENERAL

This error is thrown if deletion failed.

Examples

Example 1

Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Local/my-project/my-dataset", false, true, true
]'
https://198.51.100.10/zebi/api/v2/deleteDataset -k
```

Response

The above request returns the HTTP status code 200 (OK) and 0 as integer indicating a successful operation.

Example 2

Erroneous Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Local/my-project/non-exist-dataset", false, true, true
]' https://198.51.100.10/zebi/api/v2/deleteDataset -k
```

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "No such dataset exists.",
  "extendedData": {}
}
```


deleteMappingFromProject

Deletes an existing project mapping.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[createMappingForProject](#), [createMappingForVolume](#)

Parameters

datasetPath

Path of the project. The format is <poolName/Local/<projectName>.

initiatorGroupName

Name of the initiator group in the existing mapping.

targetGroupName

Name of the target group in the existing mapping.

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified project, initiator group, or target group cannot be found.

EZEBI_INVALID_ARGUMENT

This exception is thrown for any of the following reasons:

- The target group and initiator group have different protocols.
- The target group and initiator group are either iSCSI or FC protocols (unknown).
- The view or mapping does not exist.

Examples

Example 1

Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/demoProject", "demoiSCSIInitiatorGroup",
    "demoiSCSITargetGroup"
  ]' \
  https://198.51.100.10/zebi/api/v2/deleteMappingFromProject -k
```

Response

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

Example 2

Erroneous Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/demoProject",
    "demoiSCSIInitiatorGroup",
    "demoiSCSITargetGroup"
  ]' \
  https://198.51.100.10/zebi/api/v2/deleteMappingFromProject -k
```

Error Response

In this example, the request returns the HTTP status code 400 (bad request) and the following message:

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "",
  "message": "Did not find the view to delete.
Project=pool-a/Local/demoProject.
Initiator group=demoiSCSIInitiatorGroup .Target
group=demoiSCSITargetGroup",
  "extendedData": {}
}
```

Example 3

Erroneous Request

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN' \
  -H 'cache-control: no-cache' \
```

```
-H 'content-type: application/json' \
-d '["pool-a/Local/demoProject",
    "demoiSCSIInitiatorGroup",
    "UNKNOWNiSCSITargetGroup"
]' \
https://198.51.100.10/zebi/api/v2/deleteMappingFromProject -k
```

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "Specified target group 'UNKNOWNiSCSITargetGroup' does not exist.",
  "extendedData": {}
}
```

deletePool

This API does not delete a pool, but moves the specified pool to the exported state. When a pool is exported, all shares, LUNs, and the pool itself are unmounted and hosts cannot access data from the pool.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[importPool](#), [exportPool](#), [resetPoolError](#), [checkPoolIntegrity](#)

Parameters

poolName

Name of the pool to delete.

force

A boolean, which specifies whether to force the deletion of the pool with its dependents.

Returns

Returns an integer status where:

- 0 indicates that the request succeeded.
- 1 indicates that the request failed.

Exceptions Thrown**EZEBI_RESOURCE_NOT_FOUND**

If the pool does not exist.

EZEBI_INVALID_ARGUMENT

If an invalid character is detected in the pool name.

Examples**Example 1****Request (curl)**

```
curl -X POST \
  -H 'authorization: Basic AUTH_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '{"pool6a", false}' \
  https://198.51.100.10/zebi/api/v2/deletePool -k
```

Response

The above request returns the HTTP status code 200 (OK) and 0 as integer indicating a successful operation.

Example 2**Erroneous Request (curl)**

```
curl -X POST \
  -H 'authorization: Basic AUTH_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '{"nonExistentPool6a", false}' \
  https://198.51.100.10/zebi/api/v2/deletePool -k
```

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "Pool not found.",
  "message": "The pool [nonExistentPool63a] was not found.",
  "extendedData": {
    "EX_CAUSE_MESSAGE": "Pool not found.",
    "EX_CAUSE_CODE_NAME": "EZEBI_RESOURCE_NOT_FOUND"
  }
}
```

deleteProject

Deletes a project

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[createProject](#), [listProjects](#)

Parameters

projectDatasetpath

The path for the project dataset.

Returns

Returns an integer status where:

- 0 indicates that the request succeeded.
- 1 indicates that the request failed.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified project doesn't exist.

EZEBI_INVALID_ARGUMENT

This exception is thrown if an invalid character is detected in the project dataset path.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST \
  -H 'authorization: Basic AUTH_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '["pool63a/Local/testProject"]' \
  https://198.51.100.10/zebi/api/v2/deleteProject -k
```

Response

The above request returns the HTTP status code 200 (OK) and 0 as integer indicating a successful operation.

Erroneous Request (curl)

```
curl -X POST \
  -H 'authorization: Basic AUTH_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '["pool63a/Local/nonExistentProject"]' \
  https://198.51.100.10/zebi/api/v2/deleteProject -k
```

Example 2

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "Cannot find the specified project 'pool63a/Local/nonExistentProject'.",
  "extendedData": {}
}
```

deleteShare

Promotes dependents as specified, and then deletes the share whose dataset path is supplied as an argument.

Dependents are deleted as well if promote is false.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[deleteDataset](#)

Parameters

datasetPath

This string identifies the share on the IntelliFlash array.

recursive

Whether to remove the dependents of the dataset before deleting it.

errorIfNotExist

Whether to raise an exception if the datasetPath to delete doesn't exist.

promote

Whether to remove the dependents of the dataset before deleting it.

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown**EZEBI_RESOURCE_NOT_FOUND**

This exception is thrown if a dataset is not found.

EZEBI_RESOURCE_INUSE

This exception is thrown if the replication is currently running.

EZEBI_RESOURCE_SUSPENDED

This exception is thrown if the pool is suspended.

EZEBI_GENERAL

This exception is thrown if deletion failed.

EZEBI_INVALID_ARGUMENT

This exception is thrown for the following conditions:

- If the dataset is empty
- If the length is greater than the maximum length
- If it contains invalid characters

Examples**Example 1****Request (curl):**

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/my-project/my-share", false, true, true
  ]' https://198.51.100.10/zebi/api/v2/deleteShare -k
```

Response

The above request returns the HTTP status code 200 (OK) and 0 as integer indicating a successful operation.

Example 2

Erroneous Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/my-project/non-exist-share", false, true, true
  ]' https://198.51.100.10/zebi/api/v2/deleteShare -k
```

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "No such share exists.",
  "extendedData": {}
}
```

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.

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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**)

errorIfExists

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.

First Available Version

API v1.0, IntelliFlash 2.1.0.0

Related APIs

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

Parameters

datasetPath

A string: the path which uniquely identifies the volume on the IntelliFlash 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**.

errorIfNotExist

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"
}
```

deleteVolume

Promotes the specified dependent and then deletes the volume whose dataset path is supplied as an argument. Dependents are deleted as well if promote is false.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[deleteDataset](#)

Parameters

datasetpath

The string that uniquely identifies the volume on the IntelliFlash array.

recursive

Whether to remove the dependants (for example, clones) and lun of the dataset before trying to delete it. Deletion might fail if there are dependants or lun existing on the dataset.

errorIfNotExist

Whether to raise an exception if the datasetPath to delete doesn't exist.

promote

Whether to promote clone before deletion

Returns

COMMAND_STATUS.COMMAND_SUCCEED(0) on success

Exceptions Thrown**EZEBI_RESOURCE_NOT_FOUND**

This exception is thrown if the dataset is not found.

EZEBI_RESOURCE_INUSE

This exception is thrown if replication is currently running.

EZEBI_RESOURCE_SUSPENDED

This exception is thrown if pool is suspended.

EZEBI_GENERAL

This exception is thrown if deletion failed.

EZEBI_INVALID_ARGUMENT

This exception is thrown for the following conditions:

- If the dataset is empty
- If the length is greater than the maximum length
- If it contains invalid characters

Examples**Example 1****Request (curl):**

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Local/my-project/my-volume", false, true, true
]' https://198.51.100.10/zebi/api/v2/deleteVolume -k
```

Response

The above request returns the HTTP status code 200 (OK) and 0 as integer indicating a successful operation.

Example 2

Erroneous Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/my-project/non-exist-volume", false, true, true
  ]' https://198.51.100.10/zebi/api/v2/deleteVolume -k
```

Error Response

The above request returns the HTTP status code 404 and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "No such volume exists.",
  "extendedData": {}
}
```

exportPool

Exports a pool.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[deletePool](#), [importPool](#), [resetPoolError](#), [checkPoolIntegrity](#)

Parameters

poolName

Name of the pool to export.

Returns

Returns an integer status where:

- 0 indicates that the request succeeded.

- 1 indicates that the request failed.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

If the pool does not exist.

EZEBI_INVALID_ARGUMENT

If an invalid character is detected in the poolName.

EZEBI_RESOURCE_EXIST

If an active pool with the same name exists already.

EZEBI_OPERATION_NOT_ALLOWED

If pool is in a state where export cannot be done.

EZEBI_GENERAL

Examples

Example 1

Request (curl):

```
curl -X POST \
  -H 'authorization: Basic AUTH_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '["pool6a"]' \
  https://198.51.100.10/zebi/api/v2/exportPool -k
```

Response

The above request returns the HTTP status code 200 (OK) and 0 as integer indicating a successful operation.

Example 2

Erroneous Request

```
curl -X POST \
  -H 'authorization: Basic AUTH_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '["nonExistentPool6a", false]' \
  https://198.51.100.10/zebi/api/v2/exportPool -k
```

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "Pool requested for export was not found",
  "message": "Failed to export the pool [nonExistentPool6a].",
  "extendedData": {
    "EX_CAUSE_MESSAGE": "Pool requested for export was not found",
    "EX_CAUSE_CODE_NAME": "EZEBI_RESOURCE_NOT_FOUND"
  }
}
```

getCopyStatus

Returns the status of a copy operation task.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[copyDataset](#), [listRunningCopyOperations](#), [listAllCopyOperations](#), [abortCopy](#)

Parameters

taskGUID

The GUID of the copy operation task.

Returns

Returns the [CopyDestination_V2_1](#) object.

Exceptions Thrown

EZEBI_GENERAL

This exception is thrown if the task GUID is invalid.

Examples

Example 1

Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '["europa:84944c8a-8bc1-4705-809e-144363797a57"]' \
```

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

Response

The above request returns the HTTP status code 200 (OK) and the [CopyDestination_V2_1](#) object.

```
{
  "numberOfCopies": 3,
  "completedCopies": 3,
  "status": "COMPLETED",
  "allDatasets": [
    "napa/Local/remotecopy/prod1",
    "napa/Local/remotecopy/prod2",
    "napa/Local/remotecopy/prod3"
  ],
  "completedDatasets": [
    "napa/Local/remotecopy/prod1",
    "napa/Local/remotecopy/prod2",
    "napa/Local/remotecopy/prod3"
  ],
  "pendingDatasets": [],
  "percentComplete": 100,
  "errorCode": null,
  "startTime": "2017-07-10T11:00:16-07:00",
  "endTime": "2017-07-10T11:01:03-07:00"
}
```

Example 2

Erroneous Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '["euro:84944c8a-8bc1-4705-809e-144363797a57"]' \
  https://198.51.100.10/zebi/api/v2/getCopyStatus -k
```

Error Response

The above request returns HTTP status code 400 with the following message:

```
{
  "code": "EZEBI_GENERAL",
  "details": "",
  "message": "Pool euro is not mounted.",
  "extendedData": {}
}
```


getDatasetSpaceInfo

Returns a [DatasetSpaceInfo_V2_1](#) object if the specified dataset exists.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[getPoolSpaceInfo](#)

Parameters

Dataset-path

A string specifying path to the dataset (share or volume).

Returns

A [DatasetSpaceInfo_V2_1](#) object.

Exceptions Thrown

EZEBI_INVALID_ARGUMENT

This error is thrown if the parameters are invalid (if dataset name contains special characters).

EZEBI_GENERAL

This error is thrown if the operation failed with internal reasons.

EZEBI_RESOURCE_NOT_FOUND

This error is thrown if the specified dataset doesn't exist.

Examples

Example 1

Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  {
    "pool-a/Local/my-project/my-dataset"
  }
]'
https://198.51.100.10/zebi/api/v2/getDatasetSpaceInfo -k
```

Response

The above request returns the HTTP status code 200 (OK) and a [DatasetSpaceInfo_V2_1](#) object.

```
{
  "originalUsedByDataAndSnapshot": 30208,
  "usedByDataAndSnapshot": 30208,
  "compressionSavingsPercentage": 0,
  "available": 1202104,
  "usedByData": 13824,
  "usedBySnapshot": 16384,
  "usedByReservation": 0,
  "quota": 1232312,
  "volSize": 0
}
```

Example 2

Erroneous Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    {
      "pool-a/Local/my-project/non-exist"
    }
  ]'
https://198.51.100.10/zebi/api/v2/getDatasetSpaceInfo -k
```

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "Cannot find dataset: pool-a/Local/my-project/non-exist",
  "extendedData": {}
}
```

getFloatingIPList

Lists floating IP addresses associated with a storage pool.

First Available Version

API v2.2, IntelliFlash 3.7.1.0

Parameters

poolName

The name of the target pool.

For empty string "", the API lists all the floating IP addresses.

Returns

Returns a [FloatingIP_V2_2](#) object.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified pool does not exist.

EZEBI_GENERAL

This exception is thrown if an internal error is detected.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the pool name is invalid.

Examples

Example 1

Request (curl)

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

Response

```
[
  {
    "resourceGroupName": "defaultResourceGroup1",
    "description": "data_10g_0@ha-controller-b,data_10g_0@ha-controller-a",
    "failoverMode": "Immediately",
    "ipAddress": "198.51.10.20",
    "netmask": "255.255.0.0",
    "poolName": "pool-a"
  }
]
```

Example 2

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/getFloatingIPList -k
```

Response

As the request was sent with an empty string, the API returns all the floating IP addresses.

```
[
  {
    "resourceGroupName": "defaultResourceGroup1",
    "description": "data_10g_0@ha-controller-b,data_10g_0@ha-controller-a",
    "failoverMode": "Immediately",
    "ipAddress": "198.30.10.20",
    "netmask": "255.255.0.0",
    "poolName": "pool-a"
  },
  {
    "resourceGroupName": "defaultResourceGroup1",
    "description": "data_10g_0@ha-controller-b,data_10g_0@ha-controller-a",
    "failoverMode": "Immediately",
    "ipAddress": "198.168.100.50",
    "netmask": "255.255.0.0",
    "poolName": "pool-b"
  },
]
```

Example 3

Erroneous Request (curl)

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

Error Response

```
[
  {
    "code": "EZEBI_INVALID_ARGUMENT",
    "details": "getFloatingIPList.arg0.value 'null':
               Pool name is not valid, null, or empty name detected.",
    "message": "Pool name is not valid, null, or empty name detected.",
    "extendedData":
      {
        "EX_CAUSE_MESSAGE": null
      }
  }
]
```

getPoolSpaceInfo

Returns a [PoolSpaceInfo_V2_1](#) object if it exists.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[getDatasetSpaceInfo](#)

Parameters

Pool-name

A string specifying pool name.

Returns

A [PoolSpaceInfo_V2_1](#) object.

Exceptions Thrown

EZEBI_INVALID_ARGUMENT

This error is thrown if the parameters are invalid (pool name contains special characters).

EZEBI_GENERAL

This error is thrown if the operation failed with internal reasons.

EZEBI_RESOURCE_NOT_FOUND

This error is thrown if the specified pool doesn't exist.

Examples

Example 1

Request (curl)

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

Response

The above request returns the HTTP status code 200 (OK) and a [PoolSpaceInfo_V2_1](#) object.

```
{
  "totalPoolSize": 1024173568,
  "originalUsedByDataAndSnapshot": 899392896,
  "usedByDataAndSnapshot": 248960000,
  "compressionSavingsPercentage": 72.32,
  "dedupeSavingsPercentage": 0,
  "totalSavingsPercentage": 72.32,
  "usedByAll": 252706816,
  "available": 771466752,
  "usedByData": 248025600,
  "usedBySnapshot": 934400,
  "usedByReservation": 0,
  "totalMetaSize": 0,
  "usedMeta": 0
}
```

Example 2

Erroneous Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  {
    "pool-non-exist"
  }
]' https://198.51.100.10/zebi/api/v2/getPoolSpaceInfo -k
```

Error Response

The above request returns the HTTP status code 404 (Not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "Cannot find the pool: pool-non-exist",
  "extendedData": {}
}
```

getProject

Returns a [Project_V2_1](#) object if the project exists.

Related APIs

[createProject](#), [modifyProjectProperties](#), [getProjectProperty](#)

Parameters**pool-name**

A string specifying the name of the pool

Project-name

A string specifying the name of the project.

Is-local

A Boolean value indicating whether the project is local or replica.

project

The name of the project.

Returns

A [Project_V2_1](#) object.

Exceptions thrown**EZEBI_INVALID_ARGUMENT**

This exception is thrown if the parameters are invalid (project or pool name contains special characters).

EZEBI_GENERAL

This exception is thrown if operation failed with internal reasons.

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if specified project doesn't exist.

Examples**Example 1****Request (curl)**

```
curl -X POST \
  -H 'authorization: Basic YOUR_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a", "my-project", true
  ]' https://YOUR_IP/zebi/api/v2/getProject -k
```

Response

The above request returns the HTTP status code 200 (OK) and a [Project_V2_1](#) object.

```
{
  "poolName": "pool-a",
```

```

    "projectName": "my-project ",
    "localDataset": true,
    "purpose": "generic",
    "mountPoint": "/export/my-project ",
    "compression": "lz4",
    "compressedLog": "off",
    "intendedProtocolList": [
        "NFS",
        "SMB",
        "FC",
        "iSCSI"
    ],
    "quotaInByte": 0,
    "quotaEnabled": false,
    "dedup": "on",
    "copies": "1",
    "primaryCache": "all",
    "secondaryCache": "all",
    "readonly": "off",
    "logbias": "latency",
    "aclInherit": "off",
    "aclMode": "passthrough",
    "krbStatus": false,
    "defaultVolumeSizeInByte": 1073741824,
    "defaultVolumeBlockSize": "4KB",
    "defaultThinProvisioning": true,
    "sync": "standard",
    "zfsDataSetName": "pool-a/Local/my-project ",
    "recordSize": "16KB",
    "quota": 0,
    "quotaMetric": "GB",
    "defaultVolumeSize": 1,
    "defaultVolumeSizeUnit": "GB"
}

```

Example 2

Erroneous Request (curl)

```

curl -X POST \
-H 'authorization: Basic YOUR_TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
    "pool-a", "non-exist", true
]' https://YOUR_IP/zebi/api/v2/getProject -k

```

Error Response

The above request returns the HTTP status code 404 (Not found) and the following message:

```

{
    "code": "EZEBI_RESOURCE_NOT_FOUND",
    "details": "",
    "message": "Cannot find the specified project 'pool-a/Local/non-exist'.",
    "extendedData": {}
}

```



```
}
```

getProjectProperty

Returns a [DatasetProperty_V2_1](#) object if the project exists.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[createProject](#), [modifyProjectProperties](#), [getProject](#)

Parameters

Dataset-path

A string specifying the project path.

Property-name

A string specifying the property to query.

Returns

A [DatasetProperty_V2_1](#) object that contains the query key and value.

Exceptions Thrown

EZEBI_INVALID_ARGUMENT

This exception is thrown if the parameters are invalid (project or property name contains invalid characters).

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified project doesn't exist.

Examples

Example 1

Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Local/my-project", "dedup"
]'
```

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

Response

The above request returns the HTTP status code 200 (OK) and a [DatasetProperty_V2_1](#) object.

```
{
  "propertyKey": "dedup",
  "propertyValue": "on"
}
```

Example 2

Erroneous Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/my-project", "non-exist-property"
  ]'
https://198.51.100.10/zebi/api/v2/getProjectProperty -k
```

Error Response

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

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "",
  "message": "Can't get property:non-exist-property.
Please supply a valid project property.
Check API documents for more details",
  "extendedData": {}
}
```

getShare

Returns a [Share_V2_1](#) object if it exists.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[createShare](#), [modifyShareProperties](#), [getShareProperty](#)

Parameters**Share-path**

A string specifying path to share.

Returns

A [Share_V2_1](#) object.

Exceptions Thrown**EZEBI_INVALID_ARGUMENT**

This exception is thrown if the parameters are invalid (project or share name contains special characters).

EZEBI_GENERAL

This exception is thrown if the operation failed with internal reasons.

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified share doesn't exist.

Examples**Example 1****Request (curl)**

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  {
    "pool-a/Local/my-project/my-share"
  }
]' https://198.51.100.10/zebi/api/v2/getShare -k
```

Response

The above request returns the HTTP status code 200 (OK) and a [Share_V2_1](#) object.

```
{
  "name": "my-share",
  "poolName": "pool-a",
  "projectName": "my-project ",
  "purpose": "custom",
  "guid": "07ee6f26-f701-4e98-8db6-778736e72df8",
  "compression": "lz4",
  "overrideCompression": false,
  "localDataset": true,
  "reservationInByte": 43,
  "reservation": 0.00004100799560546875,
  "reservationMetric": "MB",
```

```

"reservationEnabled": true,
"dedup": "on",
"overrideDedup": false,
"copies": "1",
"overrideCopies": false,
"primaryCache": "metadata",
"overridePrimaryCache": true,
"secondaryCache": "all",
"overrideSecondaryCache": false,
"readonly": "off",
"overrideReadonly": false,
"logbias": "throughput",
"overrideLogbias": true,
"sync": "standard",
"overrideSync": false,
"overrideProjectSnapshotSettings": false,
"zfsDataSetName": "pool-a/Local/my-project/my-share ",
"compressedLog": "off",
"overrideCompressedLog": true,
"overrideMountPoint": true,
"quotaInByte": 1232312,
"quota": 1.1752243041992188,
"quotaMetric": "MB",
"quotaEnabled": true,
"mountPoint": "/export/my-project/my-share",
"availableSize": 1202104,
"totalSize": 1232312,
"overrideSharenfs": true,
"overrideSharesmb": true,
"krbStatus": false,
"cifsDisplayName": "my-share",
"guestStatus": false,
"aclInherit": "off",
"overrideAclInherit": false,
"recordSize": "16KB",
"overrideRecordSize": false,
"atime": "on",
"nbmand": "off",
"aclList": [
  {
    "id": 0,
    "controllerId": null,
    "aclType": "Group",
    "aclUser": null,
    "aclGroup": "dasd",
    "aclValDisplay": "rwxpdDaARWcCos",
    "aclVal": 2032127,
    "aclMode": "Allow",
    "aclInheritanceFlag": "Default",
    "includeSubShares": false,
    "userId": 0,
    "groupId": 104
  },
  {
    "id": 0,
    "controllerId": null,
    "aclType": "User",
    "aclUser": "user1",
    "aclGroup": null,
    "aclValDisplay": "rwxpdDaARWcCos",
    "aclVal": 2032127,

```

```

        "aclMode": "Allow",
        "aclInheritanceFlag": "Default",
        "includeSubShares": false,
        "userId": 104,
        "groupId": 0
    },
    "containerName": "Local"
}

```

Example 2

Erroneous Request (curl)

```

curl -X POST \
-H 'authorization: Basic Auth_TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
{
  "pool-a/Local/my-project/non-exist"
}
]' https://198.51.100.10/zebi/api/v2/getShare -k

```

Error Response

The above request returns the HTTP status code 500 (internal server error) and the following message:

```

{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "Unable to open pool-a/Local/my-project/non-exist: dataset does not exist",
  "message": "Unable to open pool-a/Local/my-project/non-exist: dataset does not exist",
  "extendedData": {
    "EX_CAUSE_MESSAGE": "Unable to open pool-a/Local/my-project/non-exist: dataset does not exist",
    "EX_CAUSE_CODE_NAME": "EZFS_NOENT",
    "EX_CAUSE_CODE_NUMBER": "2009"
  }
}

```

getShareProperty

Returns a [DatasetProperty_V2_1](#) if it exists.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[createVolume](#), [modifyVolumeProperties](#), [getVolumeProperty](#), [getProjectProperty](#)

Parameters

Share-path

A string specifying the path to share.

Returns

A [DatasetProperty_V2_1](#) contains the query key and value.

Exceptions Thrown

EZEBI_INVALID_ARGUMENT

This error is thrown if the parameters are invalid (project or share name contains special characters)

EZEBI_GENERAL

This error is thrown if the operation failed with internal reasons.

EZEBI_RESOURCE_NOT_FOUND

This error is thrown if the specified share doesn't exist.

Examples

Example 1

Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  {
    "pool-a/Local/my-project/my-share",
    "quotaInByte"
  }
]' https://198.51.100.10/zebi/api/v2/getShareProperty -k
```

Response

The above request returns the HTTP status code 200 (OK) and a [DatasetProperty_V2_1](#) object.

```
{
  "propertyKey": "quotaInByte",
  "propertyValue": 0
}
Erroneous Request (curl)
curl -X POST \
-H 'authorization: Basic Auth TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  {
    "
```

```
    "pool-a/Local/my-project/my-share",
    "non-exist-property"
  }
] ' https://198.51.100.10/zebi/api/v2/getShareProperty -k
```

Example 2

Erroneous Request

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  {
    "pool-a/Local/my- project/my-share",
    "non-exist-property"
  }
]'
https://198.51.100.10/zebi/api/v2/getShareProperty -k
```

Error Response

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

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "",
  "message": "Can't get property:non-exist-property is not a valid
field!.
Please supply a valid property. Check API documents for more details",
  "extendedData": {}
}
```

getVolume

Returns a [Volume_V2_1](#) object if it exists.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[createVolume](#), [modifyVolumeProperties](#), [getVolumeProperty](#)

Parameters

Volume-path

A string specifying the path to volume.

Returns

A [Volume_V2_1](#) object.

Exceptions Thrown

EZEBI_INVALID_ARGUMENT

This exception is thrown if the parameters are invalid (project or volume name contains special characters).

EZEBI_GENERAL

This exception is thrown if the operation failed with internal reasons.

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified volume doesn't exist.

Examples

Example 1

Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    {
      "pool-a/Local/my-project/my-volume"
    }
  ]' https://198.51.100.10/zebi/api/v2/getVolume -k
```

Response

The above request returns the HTTP status code 200 (OK) and a [Volume_V2_1](#) object.

```
{
  "name": "my-volume",
  "poolName": "pool-a",
  "projectName": "my-project",
  "purpose": "custom",
  "guid": "007ef1b2-cae6-41ff-9523-de0d81249819",
  "compression": "lz4",
  "overrideCompression": false,
  "localDataset": true,
  "reservationInByte": 0,
  "reservation": 0,
  "reservationMetric": null,
  "reservationEnabled": false,
  "dedup": "off",
  "overrideDedup": false,
  "copies": "1",
  "overrideCopies": false,
  "primaryCache": "all",
  "overridePrimaryCache": false,
```



```

    "secondaryCache": "all",
    "overrideSecondaryCache": false,
    "readonly": "off",
    "overrideReadonly": false,
    "logbias": "latency",
    "overrideLogbias": false,
    "sync": "standard",
    "overrideSync": false,
    "overrideProjectSnapshotSettings": false,
    "zfsDataSetName": "pool-a/Local/my-project/ my-volume ",
    "compressedLog": "off",
    "overrideCompressedLog": false,
    "volSize": 33285996544,
    "luId": "600144F087EC75370000594058180013",
    "usedSize": 48733696,
    "thinProvisioning": true,
    "blockSize": "32KB",
    "writeBackCache": "enable",
    "overrideViews": false,
    "protocol": "iSCSI",
    "containerName": "Local"
}

```

Example 2

Erroneous Request (curl)

```

curl -X POST \
-H 'authorization: Basic Auth TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  {
    "pool-a/Local/my-project/non-exist"
  }
]' https://198.51.100.10/zebi/api/v2/getVolume -k

```

Error Response

The above request returns the HTTP status code 500 (internal server error) and the following message:

```

{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "Unable to open pool-a/Local/my-project/non-exist: dataset does not exist",
  "message": "Unable to open pool-a/Local/my-project/non-exist: dataset does not exist",
  "extendedData": {
    "EX_CAUSE_MESSAGE": "Unable to open pool-a/Local/my-project /non-exist: dataset does not exist",
    "EX_CAUSE_CODE_NAME": "EZFS_NOENT",
    "EX_CAUSE_CODE_NUMBER": "2009"
  }
}

```

getVolumeProperty

Returns a [DatasetProperty_V2_1](#) object if it exists.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[createVolume](#), [modifyVolumeProperties](#), [getShareProperty](#), [getProjectProperty](#)

Parameters

Volume-path

A string specifying path to volume.

Returns

A [DatasetProperty_V2_1](#) object that contains the query key and value.

Exceptions Thrown

EZEBI_INVALID_ARGUMENT

This exception is thrown if the parameters are invalid (project or volume name contains special characters).

EZEBI_GENERAL

This exception is thrown if the operation failed with internal reasons.

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified volume doesn't exist.

Examples

Example 1

Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/my-project/my-volume",
    "dedup"
  ]' https://198.51.100.10/zebi/api/v2/getVolumeProperty -k
```

Response

The above request returns the HTTP status code 200 (OK) and a [DatasetProperty_V2_1](#) object.

```
{
  "propertyKey": "dedup",
  "propertyValue": "off"
}
```

Example 2

Erroneous Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/my-project/my-volume",
    "non-exist-property"
  ]'
https://198.51.100.10/zebi/api/v2/getVolumeProperty -k
```

Error Response

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

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "",
  "message": "Can't get property:non-exist-property is not a valid field!. Please supply a valid property. Check API documents for more details",
  "extendedData": {}
}
```

importPool

Imports a pool.

First Available Version

API v2.1, IntelliJFlash 3.7.0.x

Related APIs

[deletePool](#), [exportPool](#), [resetPoolError](#), [checkPoolIntegrity](#)

Parameters

poolName

Name of the pool to import.

Returns

Returns an integer status where:

- 0 indicates that the request succeeded.
- 1 indicates that the request failed.

Exceptions Thrown**EZEBI_RESOURCE_NOT_FOUND**

This exception is thrown if the pool does not exist.

EZEBI_INVALID_ARGUMENT

This exception is thrown if an invalid character is detected in the poolName.

EZEBI_RESOURCE_EXIST

This exception is thrown if an active pool with the same name exists already.

EZEBI_OPERATION_NOT_ALLOWED

This exception is thrown if pool is in a state that doesn't permit import.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples**Example 1****Request (curl)**

```
curl -X POST \
  -H 'authorization: Basic AUTH_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '["pool6a"]' \
  https://198.51.100.10/zebi/api/v2/importPool -k
```

Response

The above request returns the HTTP status code 200 (OK) and 0 as integer indicating a successful operation.

Example 2**Erroneous Request (curl)**

```
curl -X POST \
  -H 'authorization: Basic AUTH_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
```

```
-d '["nonExistentPool6a", false]' \
https://198.51.100.10/zebi/api/v2/importPool -k
```

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "Pool requested for import was not found",
  "message": "Failed to import the pool [nonExistentPool6a].",
  "extendedData": {
    "EX_CAUSE_MESSAGE": "Pool requested for import was not found",
    "EX_CAUSE_CODE_NAME": "EZEBI_RESOURCE_NOT_FOUND"
  }
}
```

inheritPropertyFromProject

Reverts share or volume configuration value to parent configuration value.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[createProject](#), [modifyProjectProperties](#), [getProjectProperty](#), [modifyShareProperties](#),
[modifyVolumeProperties](#)

Parameters

Dataset-path

A string specifying path to the dataset (either share or volume).

Property-name

A string specifying the property to revert.

Valid property name includes Compression, Dedup, Copies, PrimaryCache, SecondaryCache, Readonly, Logbias, Sync, CompressedLog, ProjectSnapshotSettings, MountPoint, Sharenfs, Sharesmb, AclInherit, and Views.

Returns

Returns an integer status where:

- 0 indicates that the request succeeded.

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

Exceptions Thrown

EZEBI_INVALID_ARGUMENT

This exception is thrown if the parameters are invalid (project or share name contains special characters).

EZEBI_GENERAL

This exception is thrown if the operation failed with internal reasons.

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified dataset doesn't exist.

Examples

Example 1

Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  {
    "pool-a/Local/my-project/my-dataset",
    "Dedup"
  }
]'
https://198.51.100.10/zebi/api/v2/inheritPropertyFromProject -k
```

Response

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

Example 2

Erroneous Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  {
    "pool-a/Local/my-project/my-dataset",
    "non-supported-property"
  }
]'
https://198.51.100.10/zebi/api/v2/inheritPropertyFromProject -k
```

Error Response

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

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "inheritPropertyFromProject.arg1 value 'non-supported-property':
    Supported Inheritable Properties includes: Compression, Dedup, Copies,
    PrimaryCache,
    SecondaryCache, Readonly, Logbias, Sync, CompressedLog,
    ProjectSnapshotSettings,
    MountPoint, Sharenf, Sharesmb, AclInherit, Views.",
  "message": "Supported Inheritable Properties includes:
    Compression, Dedup, Copies, PrimaryCache, SecondaryCache, Readonly,
    Logbias, Sync,
    CompressedLog, ProjectSnapshotSettings, MountPoint, Sharenf, Sharesmb,
    AclInherit, Views.",
  "extendedData": {
    "EX_CAUSE_MESSAGE": null
  }
}
```

isProjectExposedOverNFS

Returns whether the NFS protocol is enabled for a project.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[setNFSSharingOnProject](#), [setNFSNetworkACLsOnProject](#), [addNFSNetworkACLOnProject](#),
[removeNFSNetworkACLOnProject](#), [removeAllNFSNetworkACLsOnProject](#),
[getNFSNetworkACLsOnProject](#)

Parameters

datasetPath

Path of the project. The format is <poolName>/Local/<projectName>.

This operation is not allowed for replica project datasets.

Returns

Returns True or False based on whether NFS protocol is enabled.

Exceptions Thrown**EZEBI_RESOURCE_NOT_FOUND**

This exception is thrown if the specified project cannot be found.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the path specified belongs to replica dataset.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples**Example 1****Request (curl)**

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

Response

The above request returns the HTTP status code 200 (OK) and returns a true or false value indicating whether NFS is enabled over the specified project.

Example 2**Erroneous Request (curl)**

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

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
```



```

    "code": "EZEBI_RESOURCE_NOT_FOUND",
    "details": "",
    "message": "Cannot find the specified project 'pool-a/Local/UNKNOWNProject'.",
    "extendedData": {}
}

```

isShareExposedOverNFS

Returns whether the NFS protocol is enabled for the share.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[setNFSSharingOnShare](#), [setNFSNetworkACLsOnShare](#), [addNFSNetworkACLOnShare](#),
[removeNFSNetworkACLOnShare](#), [removeAllNFSNetworkACLsOnShare](#), [getNFSNetworkACLsOnShare](#)

Parameters

datasetPath

Path of the share. The format is <poolName>/Local/<projectName>/<shareName>.

This operation is not allowed for replica share datasets.

Returns

Returns True or False based on whether the NFS protocol is enabled.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified share cannot be found.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the path specified belongs to replica dataset.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/Project/demoShare"
  ]' \
  https://198.51.100.10/zebi/api/v2/isShareExposedOverNFS -k
```

Response:

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

```
true
```

Example 2

Erroneous Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/Project/UNKNOWNShare"
  ]' \
  https://198.51.100.10/zebi/api/v2/isShareExposedOverNFS -k
```

Error Response

The above request returns the HTTP status code 500 (internal server error) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "Cannot find the specified share 'pool-a/Local/Project/UNKNOWNShare'.",
  "extendedData": {}
}
```

listAllCopyOperations

Lists all copy operations.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[copyDataset](#), [getCopyStatus](#), [listRunningCopyOperations](#), [abortCopy](#)

Parameters

None

Returns

Lists the GUIDs of all copy operations.

Exceptions Thrown

None

Examples**Example 1****Request (curl)**

```
curl -X GET \
  -H 'authorization: Basic Auth_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  https://198.51.100.10/zebi/api/v2/listAllCopyOperations -k
```

Response

The above request returns the HTTP status code 200 (OK) and a list of GUIDs.

```
[
  "europa:aeca7e25-a74e-4a81-860d-e1f3640f2ad5",
  "europa:2282afb2-97b2-4ff8-86e5-10e3368766a3",
  "europa:69bc5f81-242d-4908-9b5c-5a4029cadd7d",
  "europa:a621708c-9a09-44da-b4d8-733a9e4883b5",
  "europa:bb8602e3-1431-4507-b83e-63e7a4ae2bf5",
  "europa:0e7b6bc7-5337-4974-83db-6f9e179034ea",
  "europa:42a85c00-d067-42c9-9947-883e853d0862",
  "europa:12bebd45-ae3c-451d-98ea-befdd595568a",
  "europa:825ae7b1-f596-400d-89bc-a4f7466badc3",
  "europa:1e1bc102-d8cc-4f8e-88b0-a0cc41f224fc",
  "europa:69bc5f81-242d-4908-9b5c-5a4029cadd7d"
]
```

listLunsById

Lists LUN details for the specified LUN IDs.

First Available Version

API v1.0, IntelliFlash 2.1.0.0

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
  }
]

```

listPools

Lists all the pools on the array. This is an HTTP GET method.

First Available Version

API v1.0, IntelliFlash 2.1.0.0

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.

First Available Version

API v1.0, IntelliFlash 2.1.0.0

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"
}
```

listRunningCopyOperations

Lists all running copy operations.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[copyDataset](#), [getCopyStatus](#), [listAllCopyOperations](#), [abortCopy](#)

Parameters

None

Returns

Lists the GUIDs of all currently running operations. Copy operations that are completed or aborted are not listed.

Exceptions Thrown

None

Examples

Example 1

Request (curl)

```
curl -X GET \
  -H 'authorization: Basic Auth TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  https://198.51.100.10/zebi/api/v2/listRunningCopyOperations -k
```

Response

The above request returns the HTTP status code 200 (OK) and a list of GUIDs. For example:

```
[
  "europa:137b93ca-280f-4693-a4d8-a39496827d9e",
  "europa:69bc5f81-242d-4908-9b5c-5a4029cadd7d"
]
```

Example 2

Erroneous Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[{"poolName": "europa1",
"projectName": "images",
"subProjectName": "template"
},
{
"hostName": "10.68.132.120",
"poolName": "napa",
"projectName": "remotecopy",
"subProjectNamePrefix": "prod",
"subProjectNameNumberStart": 1,
"subProjectNameNumberEnd": 3,
"subProjectNameWildcard": ""
}
]' \
  https://198.51.100.10/zebi/api/v2/copyDataset -k
```

Error Response

The above request returns HTTP status code 400 with the following message:

```
{
```

```

    "code": "EZEBI_INVALID_ARGUMENT",
    "details": "",
    "message": "Unable to find project images in europa1",
    "extendedData": {}
  }

```

listShares

Lists all the local and replicated shares in a project.

First Available Version

API v1.0, IntelliFlash 2.1.0.0

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"
}
```

listSharesByMountPoints

Lists all shares available for given mountpoints or mountpoint patterns.

Sample valid patterns include:

- /export/project/share1, /export/project/*, /export/project?/Share[1-3]
- /export/project/{share1,newshare}, *Project*

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[listShares](#)

Parameters

mountPointPatterns

An array of of mount points patterns.

Returns

A list of key value pairs, which consists of a mount point and its associated [Share_V2_1](#) object.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specific resource (pool, project, or share) is not found.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the mountpoint pattern is invalid.

Examples

Example 1

Request (curl)

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

Response

The above request returns the HTTP status code 200 (OK) and a list of key-value pairs that consists of a mount point and its associated [Share_V2_1](#) object.

```
{
  "/export/*": [
    {
      "poolName": "pool63b",
      "projectName": "testProject",
      "name": "testShare",
      "availableSize": 1930781823488,
      "totalSize": 1945635603968,
      "datasetPath": "pool63b/Local/testProject/testShare",
      "mountpoint": "/export/testProject/testShare",
      "local": true
    }
  ]
}
```

```

    },
    {
      "poolName": "pool63a",
      "projectName": "doNotCreateMeAgain",
      "name": "firstShare",
      "availableSize": 1073693696,
      "totalSize": 1073741824,
      "datasetPath": "pool63a/Local/doNotCreateMeAgain/firstShare",
      "mountpoint": "/export/doNotCreateMeAgain",
      "local": true
    }
  ]
}

```

Example 2

Erroneous Request (curl)

```

curl -X POST \
  -H 'authorization: Basic AUTH_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[["export/*"]]' \
  https://198.51.100.10/zebi/api/v2/listSharesByMountPoints -k

```

Error Response

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

```

{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "",
  "message": "Mountpoint Pattern is not valid.  
It can begin with '/', '[', '{' or '*'.  
Example: /export/ExampleProject , *export/*.",
  "extendedData": {}
}

```

listVolumes

Lists all the local or replicated volumes within a project.

First Available Version

API v1.0, IntelliFlash 2.1.0.0

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"
}
```

modifyProjectProperties

Modifies project properties with given values.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[createProject](#), [getProject](#), [getProjectProperty](#)

Parameters

Project-path

A string specifying the path of the project.

Project

A [Project_V2_1](#) object as key-value collection of properties to change.

You can modify other properties in [Project_V2_1](#) that are not marked as “read only”.

Returns

Returns an integer status where:

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

Exceptions Thrown

EZEBI_INVALID_ARGUMENT

If the parameters are invalid (when an invalid key-value pair is not supported by [Project_V2_1](#)).

EZEBI_RESOURCE_NOT_FOUND

If the specified project doesn't exist.

Examples

Example 1

Request (curl):

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Local/my-project",
  {
    "quotaInByte":42949672960,
    "compression":"gzip-9",
    "recordSize":"128Kb",
    "dedup":"on"
  }
]' https://198.51.100.10/zebi/api/v2/modifyProjectProperties -k
```

Response

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

Example 2

Erroneous Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Local/my-project",
  {
    "quotaInByte":123,    // invalid
    "compression":"gzip-9",
    "recordSize":"128Kb",
    "dedup":"on"
  }
]' https://198.51.100.10/zebi/api/v2/modifyProjectProperties -k
```

Error Response

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

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "modifyProjectProperties.arg1 value
```



```
'com.tegile.skywalk.api.v2.IPublicAPI_V2_1$Project_V2_1@7049d5c7':
Please set quotaInByte, reservationInByte to configure quota/reservation
feature.
quotaInByte/reservationInByte value should be at least 1MB(1048576)
or set to 0 for no limit.",
"message": "Please set quotaInByte, reservationInByte to
configure quota/reservation feature. quotaInByte/reservationInByte
value should be
at least 1MB(1048576), or set to 0 for no limit.",
"extendedData": {
  "EX_CAUSE_MESSAGE": null
}
}
```

modifyShareProperties

Returns a [Share_V2_1](#) object if it exists.

You can configure other general properties in [Share_V2_1](#) that is not marked as "read only".

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[createShare](#), [getShare](#)

Parameters

Share-path

A string specifying the path to share.

Share

A [Share_V2_1](#) object as key-value collection of properties to change.

Returns

Returns an integer status where:

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

Exceptions Thrown

EZEBI_INVALID_ARGUMENT

This exception is thrown if the parameters are invalid (project or share name contains special characters).

EZEBI_GENERAL

This exception is thrown if the operation failed with internal reasons.

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified share doesn't exist.

Examples**Example 1****Request (curl)**

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Local/my-project/my-share",
  {
    "sync": "always",
    "quotaInByte": 123456789
  }
]'
https://198.51.100.10/zebi/api/v2/modifyShareProperties -k
```

Response

```
0
```

Example 2**Erroneous Request (curl)**

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Local/my-project/my-share",
  {
    "sync": "non-valid-value"
  }
]'
https://198.51.100.10/zebi/api/v2/modifyShareProperties -k
```

Error Response

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

```
{
```

```

"code": "EZEBI_INVALID_ARGUMENT",
"details": "modifyShareProperties.arg1 value 'com.tegile.skywalk.api.v2.
IPublicAPI_V2_1$Share_V2_1@1c8df8ce': Supported Data Sync Preference
includes: Standard, Always.",
"message": "Supported Data Sync Preference includes: Standard, Always.",
"extendedData":
{
    "EX_CAUSE_MESSAGE": null
}
}

```

modifyVolumeProperties

Modifies volume properties with given values.

You can configure other general properties in [Volume_V2_1](#) that is not marked as “read only”.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[createVolume](#), [getVolume](#)

Parameters

Volume-path

A string specifying the path of the volume.

Volume

A [Volume_V2_1](#) object as key-value collection of properties to change.

Returns

Returns an integer status where:

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

Exceptions Thrown

EZEBI_INVALID_ARGUMENT

This exception is thrown if the parameters are invalid (project or volume name contains special characters).

EZEBI_GENERAL

This exception is thrown if the operation failed with internal reasons.

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified volume doesn't exist.

Examples

Example 1

Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[pool-a/Local/genericProj/genericProjLUN_1,
  {
    "compression": "gzip-9",
    "recordSize": "128Kb",
    "dedup": "on"
  }
]'
https://198.51.100.10/zebi/api/v2/modifyVolumeProperties -k
```

Response

```
0
```

Example 2

Erroneous Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[pool-a/Local/genericProj/genericProjLUN_1,
  {
    "compression": "really-bad-compression-algorithm",
  }
]'
https://198.51.100.10/zebi/api/v2/modifyVolumeProperties -k
```

Error Response

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

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "createProject.arg0 value
'com.tegile.skywalk.api.v2.IPublicAPI_V2_1$Project_V2_1@1778db80':
Invalid Compression Algorithm, disable compression by passing in
```

```

    'Off' or enable compression by using one of the following:
    Lzjb, Gzip-2, Gzip, Gzip-9, Lz4.",
    "message": "Invalid Compression Algorithm, disable compression by
    passing in 'Off' or enable
    compression by using one of the following: Lzjb, Gzip-2, Gzip, Gzip-9,
    Lz4.",
    "extendedData": {"EX_CAUSE_MESSAGE": null}
}

```

resetPoolError

Clears errors associated with a pool.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[deletePool](#), [exportPool](#), [importPool](#), [checkPoolIntegrity](#)

Parameters

poolName

Name of the pool where you want to clear the errors.

Returns

Returns an integer status where:

- 0 indicates that the request succeeded.
- 1 indicates that the request failed.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the pool does not exist.

EZEBI_INVALID_ARGUMENT

This exception is thrown if an invalid character is detected in the poolName.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST \
  -H 'authorization: Basic AUTH_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '{"pool63a"}' \
  https://198.51.100.10/zebi/api/v2/resetPoolError -k
```

Response

The above request returns the HTTP status code 200 (OK) and 0 as integer indicating a successful operation.

Example 2

Erroneous Request (curl)

```
curl -X POST \
  -H 'authorization: Basic AUTH_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '{"nonExistentPool6a"}' \
  https://198.51.100.10/zebi/api/v2/resetPoolError -k
```

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "Pool not found.",
  "message": "The pool [nonExistentPool6a] was not found.",
  "extendedData": {
    "EX_CAUSE_MESSAGE": "Pool not found.",
    "EX_CAUSE_CODE_NAME": "EZEBI_RESOURCE_NOT_FOUND"
  }
}
```

setNFSSharingOnProject

Enables or disables NFS protocol for project. If you disable NFS protocol for a project, any existing network ACLs on the project are removed as well.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[setNFSNetworkACLsOnProject](#), [addNFSNetworkACLOnProject](#), [removeNFSNetworkACLOnProject](#), [removeAllNFSNetworkACLsOnProject](#), [isProjectExposedOverNFS](#), [getNFSNetworkACLsOnProject](#)

Parameters

datasetPath

Path of the project. The format is <poolName>/Local/<projectName>.

This operation is not allowed for replica project datasets.

turnOn

Enables or disables NFS protocol on the project.

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified project is not found.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the path specified belongs to a replica dataset.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/demoProject", true
  ]' \
  https://198.51.100.10/zebi/api/v2/setNFSSharingOnProject -k
```

Response

The above request returns the HTTP status code 200 (OK) and 0 indicating success. This enables NFS sharing on the specified project.

Example 2

Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/demoProject", false
  ]' \
  https://198.51.100.10/zebi/api/v2/setNFSSharingOnProject -k
```

Response

The above request returns the HTTP status code 200 (OK) and 0 indicating success. This disables NFS sharing on the specified project.

Example 3

Erroneous Request

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/UNKNOWNProject", true
  ]' \
  https://198.51.100.10/zebi/api/v2/setNFSSharingOnProject -k
```

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "Cannot find the specified project 'pool-a/Local/UNKNOWNProject'.",
  "extendedData": {}
}
```

Example 4

Erroneous Request

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
```



```
-d '[
"pool-a/Replica/replicaProject", true
]' \
https://198.51.100.10/zebi/api/v2/setNFSSharingOnProject -k
```

Error Response

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

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "setNFSSharingOnProject.arg0 value 'pool-a/Replica/replicaProject':
Local dataset path expected. For example, valid formats are
'pool-name/Local/project-name' or 'pool-name/Local/project-name/share-or-lun-name'.",
  "message": "Local dataset path expected.
For example, valid formats are 'pool-name/Local/project-name' or
'pool-name/Local/project-name/share-or-lun-name'.",
  "extendedData": {"EX_CAUSE_MESSAGE": null}
}
```

setNFSSharingOnShare

Enables or disables NFS protocol for a share.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[setNFSNetworkACLsOnShare](#), [addNFSNetworkACLOnShare](#), [removeNFSNetworkACLOnShare](#), [removeAllNFSNetworkACLsOnShare](#), [isShareExposedOverNFS](#), [getNFSNetworkACLsOnShare](#)

Parameters

datasetPath

Path of the project. The format is <poolName>/Local/<projectName>/<shareName>.

This operation is not allowed for replica project datasets.

turnOn

Enables or disables NFS protocol on the share.

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown**EZEBI_RESOURCE_NOT_FOUND**

This exception is thrown if the specified share is not found.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the path specified belongs to a replica dataset.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples**Example 1****Request (curl)**

```
curl -X POST \
-H 'Authorization:Basic Auth_TOKEN' \
-H 'cache-control: no-cache'\
-H 'Content-Type:application/json' \
-d '["pool-a/Local/project/demoShare", true]'
https://198.51.100.10/zebi/api/v2/setNFSSharingOnShare -k
```

Response

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

Example 2**Erroneous Request (curl)**

```
curl -X POST \
-H 'Authorization:Basic Auth_TOKEN' \
-H 'cache-control: no-cache'\
-H 'Content-Type:application/json' \
-d '["pool-a/Local/project/UNKNOWNShare", true]'
https://198.51.100.10/zebi/api/v2/setNFSSharingOnShare -k
```

Error Response

The above request returns the HTTP status code 500 (internal server error) and the following response:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "Unable to open pool-a/Local/project/UNKNOWNShare :
dataset does not exist",
  "message": "Unable to open pool-a/Local/project/UNKNOWNShare :
dataset does not exist",
  "extendedData": {
    "EX_CAUSE_MESSAGE": "Unable to open pool-a/Local/project/UNKNOWNShare :
dataset does not exist",
    "EX_CAUSE_CODE_NAME": "EZFS_NOENT",
```

```
    "EX_CAUSE_CODE_NUMBER": "2009"  }
}
```

Example 3

Erroneous Request

```
curl -X POST \
-H 'Authorization:Basic Auth_TOKEN' \
-H 'cache-control: no-cache'\
-H 'Content-Type:application/json' \
-d '["pool-a/Replica/replicaProj/replicaShare", true]'
https://198.51.100.10/zebi/api/v2/setNFSSharingOnShare -k
```

Error Response

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

```
{
  "code":"EZEBI_INVALID_ARGUMENT",
  "details":"setNFSSharingOnShare.arg0 value
    'pool-a/Replica/replicaProj/replicaShare': Local dataset path
    expected.
    For example, valid formats are 'pool-name/Local/project-name' or
    'pool-name/Local/project-name/share-or-lun-name'.",
  "message":"Local dataset path expected.
    For example, valid formats are 'pool-name/Local/project-name'
    or
    'pool-name/Local/project-name/share-or-lun-name'.",
  "extendedData":{
    "EX_CAUSE_MESSAGE":null
  }
}
```

Chapter 6

Snapshot Methods

Topics:

- [*cloneProjectSnapshot*](#)
- [*cloneReplicaProjectSnapshot*](#)
- [*cloneReplicaSubProjectSnapshot*](#)
- [*cloneShareSnapshot*](#)
- [*cloneVolumeSnapshot*](#)
- [*cloneVolumeSnapshot*](#)
- [*createProjectSnapshot*](#)
- [*createShareSnapshot*](#)
- [*createSnapshotSchedule*](#)
- [*createVolumeSnapshot*](#)
- [*deleteProjectSnapshot*](#)
- [*deleteShareSnapshot*](#)
- [*deleteSnapshotSchedule*](#)
- [*deleteSnapshotSchedules*](#)
- [*deleteVolumeSnapshot*](#)
- [*getProjectCloneStatus*](#)
- [*getProjectSnapshotCreationStatus*](#)
- [*getShareSnapshotCreationStatus*](#)
- [*getSnapshotSchedule*](#)
- [*getVolumeSnapshotCreationStatus*](#)
- [*inheritSnapshotSettingsFromProject*](#)
- [*isSnapshotSchedulesInheritedFromProject*](#)
- [*listDependenciesAndSnapshotCountOnDelete*](#)
- [*listDependenciesAndSnapshotCountOnRollback*](#)
- [*listSnapshots*](#)
- [*rollBackToProjectSnapshot*](#)
- [*rollBackToShareSnapshot*](#)
- [*rollBackToVolumeSnapshot*](#)

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

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.

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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"
}
```

cloneReplicaProjectSnapshot

Clones a replica project by its specified snapshot.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[*cloneReplicaSubProjectSnapshot*](#)

Parameters

snapshotPath

Snapshot path of the project to clone from. The path can be `pool/container/project@snapshot` or `pool/project@snapshot`.

cloneName

The name of the clone project.

readOnly

Read only clone.

inheritSettings

Inherit project settings.

keepLuGuid

Keep LUN GUID

Returns

COMMAND_STATUS.COMMAND_SUCCEED(0) on success.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specific project is not found.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the snapshot path is invalid.

EZEBI_RESOURCE_INUSE

This exception is thrown if the target project name or target project mount point already exists.

EZEBI_REQUEST_INTERRUPTED

This exception is thrown if the clone failed.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKENY' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Replica/my-project@my-snapshot", "my-clone", false, true, true
  ]' https://198.51.100.10/zebi/api/v2/cloneReplicaProjectSnapshot -k
```

Response

The above request returns the HTTP status code 200 (OK) and 0 as an integer indicating a successful operation.

Example 2

Erroneous Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Replica/my-project@non-exist-snapshot", "my-clone", false, true,
    true
  ]'
https://198.51.100.10/zebi/api/v2/cloneReplicaProjectSnapshot -k
```

Error Response

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

```
{
  "code": "EZEBI_GENERAL",
  "details": "Failed to clone from pool-a/Replica/my-project@non-exist-
snapshot to
pool-a/Local/myclone.
Result message: Failure: cannot open 'pool-a/Replica/my-project@non-
exist-snapshot':
dataset does not exist",
  "message": "Failed to clone Project 'my-project' when cloning
'pool-a/Replica/my-project@non-exist-snapshot'.",
  "extendedData": {"EX_CAUSE_MESSAGE": "Failed to clone from
pool-a/Replica/my-project@non-exist-snapshot to pool-a/Local/myclone.
```

```
Result message: Failure: cannot open 'pool-a/Replica/my-project@non-
exist-snapshot':
  dataset does not exist", "EX_CAUSE_CODE_NAME":"EZEBI_GENERAL"}
}
```

cloneReplicaSubProjectSnapshot

Clones a replica sub-project by the specified snapshot.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[cloneReplicaProjectSnapshot](#)

Parameters

snapshotPath

Snapshot path of the project to clone from. The path could be `pool/container/project@snapshot` or `pool/project@snapshot`.

cloneName

The name of the clone project.

readOnly

Read-only clone.

inheritSettings

Inherit project settings.

keepLuGuid

Keep LUN GUID

Returns

COMMAND_STATUS.COMMAND_SUCCEED(0) on success.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specific project is not found.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the snapshot path is invalid.

EZEBI_RESOURCE_INUSE

This exception is thrown if the target project name or target project mount point already exists.

EZEBI_REQUEST_INTERRUPTED

This exception is thrown if the clone failed.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples**Example 1****Request (curl)**

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Replica/my-project/my-dataset@my-snapshot", "my-clone", false,
    true, true
  ]'
https://198.51.100.10/zebi/api/v2/cloneReplicaSubProjectSnapshot -k
```

Response

The above request returns the HTTP status code 200 (OK) and 0 as integer indicating a successful operation.

Example 2**Erroneous Request (curl)**

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Replica/my-project/my-dataset@non-exist-snapshot",
    "my-clone", false, true, true
  ]'
https://198.51.100.10/zebi/api/v2/cloneReplicaSubProjectSnapshot -k
```

Error Response

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

```
{
  "code": "EZEBI_REQUEST_INTERRUPTED",
```

```

    "details": "Failed to clone from pool-a/Replica/my-project/my-
dataset@non-exist-snapshot
    to pool-a/Local/myclone/my-dataset. Result message: Failure: cannot
    open
    'pool-a/Replica/my-project/my-dataset@non-exist-snapshot': dataset does
    not exist",
    "message": "Failed to clone share 'pool-a/Replica/my-project/my-
dataset@non-exist-snapshot'.",
    "extendedData": {"EX_CAUSE_MESSAGE": "Failed to clone from
    pool-a/Replica/my-project/my-dataset@non-exist-snapshot to pool-a/
    Local/myclone/my-dataset.
    Result message: Failure: cannot open 'pool-a/Replica/my-project/my-
    dataset@non-exist-snapshot':
    dataset does not exist", "EX_CAUSE_CODE_NAME": "EZEBI_GENERAL"}
}

```

cloneShareSnapshot

Clones the specified share-level snapshot.

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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"
}
```

cloneVolumeSnapshot

Clones the specified snapshot of a volume.

First Available Version

API v2.0, IntelliFlash 3.5.0.0

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

Returns the HTTP status code 200 (OK) and 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, false, 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.

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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"
}

```

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.

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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"
}
```

createShareSnapshot

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

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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" }
```

createSnapshotSchedule

Creates a snapshot schedule for the specified dataset.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[getSnapshotSchedule](#), [deleteSnapshotSchedules](#), [deleteSnapshotSchedule](#)

Parameters

schedule

Snapshot schedule definition.

This operation is not allowed for replica project datasets.

Returns

Returns a schedule ID.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified dataset does not exist.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the schedule is invalid.

Examples

Example 1

Request (curl)

```
curl -X POST \
```

```
-H 'authorization: Basic Auth TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
{
  "scheduleId":0,
  "retentionPeriod": 6,
  "scheduleIntervalType": "Week",
  "repeatInterval": 1,
  "startDate": "2017-08-09",
  "startTime": "10:00",
  "endTime": "",
  "daysOfWeek": "2,3,4,5,6",
  "dayOfMonth": 0,
  "weekdayOfMonth": "",
  "datasetPath": "pool-a/Local/demoProject",
  "quiesce": "Off"
}
]' \
https://198.51.100.10/zebi/api/v2/createSnapshotSchedule -k
```

Response:

The above request returns the HTTP status code 200 (OK) and a new schedule ID. For example, 44.

Example 2

Erroneous Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
{
  "scheduleId":0,
  "retentionPeriod": 6,
  "scheduleIntervalType": "Hour",
  "repeatInterval": 1,
  "startDate": "2017-08-10",
  "startTime": "10:00",
  "endTime": "",
  "daysOfWeek": "",
  "dayOfMonth": 0,
  "weekdayOfMonth": "",
  "datasetPath": "pool-a/Local/demoProject",
  "quiesce": "Off"
}
]' \
https://198.51.100.10/zebi/api/v2/createSnapshotSchedule -k
```

Response

In this example, the request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "createSnapshotSchedule.arg0
value
'com.tegile.skywalk.api.v2.IPublicAPI_V2_1$SnapshotSchedule_V2_1@10ed209':
Invalid end time. End time should be in 24HR HH:MM format.",
  "message": "Invalid end time. End time should be in 24HR HH:MM format.",
  "extendedData": {"EX_CAUSE_MESSAGE": null}
}
```

Example 3

Erroneous Request

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
{
  "scheduleId": 0,
  "retentionPeriod": 6,
  "scheduleIntervalType": "Monthly",
  "repeatInterval": 1,
  "startDate": "2017-08-10",
  "startTime": "10:00",
  "endTime": "20:00",
  "daysOfWeek": "",
  "dayOfMonth": 0,
  "weekdayOfMonth": "",
  "datasetPath": "pool-a/Local/demoProject",
  "quiesce": "Off"
}
]' \
https://198.51.100.10/zebi/api/v2/createSnapshotSchedule -k
```

Error Response

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

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "createSnapshotSchedule.arg0 value
'com.tegile.skywalk.api.v2.IPublicAPI_V2_1$SnapshotSchedule_V2_1@5b6c021e':
Invalid schedule interval type.
Possible values are: Minute, Hour, Day, Week, Month.",
  "message": "Invalid schedule interval type.
Possible values are: Minute, Hour, Day, Week, Month.",
}
```

```
    "extendedData":{"EX_CAUSE_MESSAGE":null}
}
```

Example 4

Erroneous Request

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
{
  "scheduleId":0,
  "retentionPeriod": 6,
  "scheduleIntervalType": "Hour",
  "repeatInterval": 1,
  "startDate": "2017-08-10",
  "startTime": "10:00",
  "endTime": "20:00",
  "daysOfWeek": "",
  "dayOfMonth": 0,
  "weekdayOfMonth": "",
  "datasetPath": "pool-a/Local/UNKNOWNProject",
  "quiesce": "Off"
}
]' \
https://198.51.100.10/zebi/api/v2/createSnapshotSchedule -k
```

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code":"EZEBI_RESOURCE_NOT_FOUND",
  "details":"",
  "message":"Dataset does not exist.",
  "extendedData":{}
}
```

createVolumeSnapshot

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

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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"
}
```

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.

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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"
}
```

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.

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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"
  ]
}
```

deleteSnapshotSchedule

Deletes the snapshot schedule for the dataset path, specified by the snapshot ID.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[createSnapshotSchedule](#), [getSnapshotSchedule](#), [deleteSnapshotSchedules](#)

Parameters

datasetPath

Full path of the dataset or project.

This operation is not allowed for replica project datasets.

scheduleId

ID of the snapshot schedule.

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified dataset does not exist.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the path specified belongs to a replica dataset or if the dataset path is invalid.

Examples

Example 1

Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/demoProject", 41
  ]' \
  https://198.51.100.10/zebi/api/v2/deleteSnapshotSchedule -k
```

Response

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

Example 2

Erroneous Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/demoProject", 100
  ]' \
  https://198.51.100.10/zebi/api/v2/deleteSnapshotSchedule -k
```

Error Response

In this example, the request returns the HTTP status code 400 (bad request) and the following message:

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "",
  "message": "Specified schedule id does not exist.",
  "extendedData": {}
}
```

Example 3

Erroneous Request

```
curl -X POST \
```

```
-H 'authorization: Basic Auth TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
"pool-a/Replica/replicaProject", 10
]' \
https://198.51.100.10/zebi/api/v2/deleteSnapshotSchedule -k
```

Error Response

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

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "deleteSnapshotSchedule.arg0 value 'pool-a/Replica/replicaProject':
Local dataset path expected. For example, valid formats are
'pool-name/Local/project-name' or 'pool-name/Local/project-name/share-or-lun-name'.",
  "message": "Local dataset path expected. For example,
valid formats are 'pool-name/Local/project-name' or
'pool-name/Local/project-name/share-or-lun-name'.",
  "extendedData": {"EX_CAUSE_MESSAGE": null}
}
```

deleteSnapshotSchedules

Deletes all snapshot schedules for the specified dataset.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[createSnapshotSchedule](#), [getSnapshotSchedule](#), [deleteSnapshotSchedule](#)

Parameters

datasetPath

Full path of the dataset or project.

This operation is not allowed for replica project datasets.

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown**EZEBI_RESOURCE_NOT_FOUND**

This exception is thrown if the specified dataset does not exist.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the path specified belongs to a replica dataset or if the dataset path is invalid.

Examples**Example 1****Request (curl)**

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

Response

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

Example 2**Request (curl)**

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

Response

In this example, the request returns the HTTP status code 404 (Not Found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
}
```



```
{
  "message": "Dataset does not exist.",
  "extendedData": {}
}
```

Example 3

Request

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

Response

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

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "deleteSnapshotSchedule.arg0 value 'pool-a/Replica/replicaProject': Local dataset path expected. For example, valid formats are 'pool-name/Local/project-name' or 'pool-name/Local/project-name/share-or-lun-name'.",
  "message": "Local dataset path expected. For example, valid formats are 'pool-name/Local/project-name' or 'pool-name/Local/project-name/share-or-lun-name'.",
  "extendedData": {"EX_CAUSE_MESSAGE": null}
}
```

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.

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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"
}
```

getProjectCloneStatus

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

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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"
}
```

getProjectSnapshotCreationStatus

Gets the status of a project snapshot creation request.

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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"
}
```

getShareSnapshotCreationStatus

Gets the status of a share snapshot creation request.

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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"
}
```

getSnapshotSchedule

Returns snapshot schedules for a dataset.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[createSnapshotSchedule](#), [deleteSnapshotSchedules](#), [deleteSnapshotSchedule](#)

Parameters

datasetPath

Full path of the dataset.

This operation is not allowed for replica project datasets.

Returns

Returns a list of snapshot schedules for the specified dataset path.

Exceptions Thrown**EZEBI_RESOURCE_NOT_FOUND**

This exception is thrown if the specified dataset does not exist.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the path specified belongs to a replica dataset.

Examples**Example 1****Request (curl)**

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

Response:

The above request returns the HTTP status code 200 (OK) and a schedule object as shown below.

```
[
  {
    "scheduleId": 42,
    "retentionPeriod": 6,
    "scheduleIntervalType": "Week",
    "repeatInterval": 1,
    "startDate": "2017-08-09",
    "startTime": "00:00",
    "endTime": "00:00",
    "daysOfWeek": "2,3,4,5,6",
    "dayOfMonth": 0,
    "weekdayOfMonth": null,
    "datasetPath": "pool-a/Local/demoProject",
    "quiesce": "Off"
  }
]
```

Example 2**Erroneous Request (curl)**

```
curl -X POST \
```

```
-H 'authorization: Basic Auth TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
"pool-a/Local/UNKNOWNProject"
]' \
https://198.51.100.10/zebi/api/v2/getSnapshotSchedule -k
```

Response

The above request returns the HTTP status code 404 (Not Found) and the following message:

```
{
  "code":"EZEBI_RESOURCE_NOT_FOUND",
  "details":"",
  "message":"Dataset does not exist.",
  "extendedData":{}
}
```

Example 3

Erroneous Request (curl)

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

Response

In this example, the request returns the HTTP status code 400 (Bad Request) and the following message:

```
{
  "code":"EZEBI_INVALID_ARGUMENT",
  "details":"getSnapshotSchedule.arg0 value
'pool-a/Replica/replicaProject': Local dataset path expected.
For example, valid formats are 'pool-name/Local/project-name' or
'pool-name/Local/project-name/share-or-lun-name'.",
  "message":"Local dataset path expected.
For example, valid formats are 'pool-name/Local/project-name' or
'pool-name/Local/project-name/share-or-lun-name'.",
  "extendedData":{"EX_CAUSE_MESSAGE":null}
}
```

getVolumeSnapshotCreationStatus

Gets the status of a volume snapshot creation request.

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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"
}
```

inheritSnapshotSettingsFromProject

Inherits project snapshot settings for local dataset.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[*isSnapshotSchedulesInheritedFromProject*](#)

Parameters**datasetPath**

This string uniquely identifies the dataset on the IntelliFlash array.

Returns

COMMAND_STATUS.COMMAND_SUCCEED(0) on success.

Exceptions Thrown**EZEBI_RESOURCE_NOT_FOUND**

This error is thrown if the specified dataset doesn't exist.

EZEBI_INVALID_ARGUMENT

This error is thrown if the path specified belongs to a replica dataset.

Examples**Example 1****Request (curl)**

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/my-project/my-dataset"
  ]'
https://198.51.100.10/zebi/api/v2/inheritSnapshotSettingsFromProject -k
```

Response

The above request returns the HTTP status code 200 (OK) and 0 as integer indicating a successful operation.

Example 2**Erroneous Request (curl)**

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN' \
```

```
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Local/my-project/non-exist-dataset"
]'
https://198.51.100.10/zebi/api/v2/inheritSnapshotSettingsFromProject -k
```

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "Invalid dataset path. ",
  "extendedData": {}
}
```

isSnapshotSchedulesInheritedFromProject

Checks whether the dataset is currently inheriting project snapshot settings.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[inheritSnapshotSettingsFromProject](#)

Parameters

datasetPath

This string uniquely identifies the dataset on the IntelliFlash array.

Returns

TRUE, if the snapshot schedule is inherited from project

FALSE, if the snapshot schedule is overridden by the project rule

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This error is thrown if the specified dataset doesn't exist.

Examples

Example 1

Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/my-project/my-dataset"
  ]'
https://198.51.100.10/zebi/api/v2/isSnapshotSchedulesInheritedFromProject -k
```

Response

The above request returns the HTTP status code 200 (OK) and a boolean (true or false).

Example 2

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/my-project/non-exist-dataset"
  ]'
https://198.51.100.10/zebi/api/v2/isSnapshotSchedulesInheritedFromProject -k
```

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "Invalid dataset path. ",
  "extendedData": {}
}
```

listDependenciesAndSnapshotCountOnDelete

Lists all the dependents affected by the delete operation.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[listDependenciesAndSnapshotCountOnRollback](#)

Parameters**path**

The project path, share or volume path, or snapshot path to delete.

Returns

A *Dependencies* object.

Exceptions Thrown**EZEBI_RESOURCE_NOT_FOUND**

This error is thrown if the specified project path, share or volume path, or snapshot path is not found.

EZEBI_INVALID_ARGUMENT

This error is thrown if the specified path is invalid.

Examples**Example 1****Request (curl)**

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-01/Local/genericProject"
  ]'
https://198.51.100.10/zebi/api/v2/listDependenciesAndSnapshotCountOnDelete -k
```

Response

```
{
  "deleteableSnapCount": {
    "pool-01/Local/genericProject/clone2": 2,
    "pool-01/Local/genericProject/genericLun-genericProject-clone": 38,
    "pool-01/Local/genericProject/clone3": 1,
    "pool-01/Local/genericProject/genericLun": 4351,
    "pool-01/Local/genericProject/genericLun-genericProjectClone": 2,
    "pool-01/Local/genericProject": 3,
    "pool-01/Local/genericProject/genericLun-genericProject-clone-genericProjectClone": 1
  },
  "deleteableClonesWithSnapCount": {
    "pool-01/Local/genericProject/clone2": 2,
    "pool-01/Local/genericProject/clone3": 1,
    "pool-01/Local/genericProject/genericLun-genericProjectClone": 2,
    "pool-01/Local/genericProject/genericLun": 4351,
    "pool-01/Local/genericProject/clone2-rollbackclone": 0,
  }
}
```



```

    "pool-01/Local/genericProject/
    genericLun-genericProjectClone-rollbackclone": 0,
    "pool-01/Local/genericProject/
    genericLun-genericProject-clone-genericProjectClone": 1,
    "pool-01/Local/genericProject/clone3-rollbackclone": 0,
    "pool-01/Local/genericProject/genericLun-rollbackclone": 0,
    "pool-01/Local/genericProject/
    genericLun-genericProject-clone-genericProjectClone-rollbackclone":
0,
    "pool-01/Local/genericProject/
    genericLun-genericProject-clone-rollbackclone": 0
  }
}

```

Example 2

Erroneous Request (curl)

```

curl -X POST \
  -H 'authorization: Basic Auth TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-01/Local/non-exist-project"
  ]'
https://198.51.100.10/zebi/api/v2/listDependenciesAndSnapshotCountOnDelete -
k

```

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```

{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "Dataset does not exist. ",
  "extendedData": {}
}

```

listDependenciesAndSnapshotCountOnRollback

Lists all the dependents affected by the rollback operation.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[listDependenciesAndSnapshotCountOnDelete](#)

Parameters**snapshotPath**

The project path, share or volume path, or snapshot path to roll back to.

Returns

A *Dependencies* object.

Exceptions Thrown**EZEBI_RESOURCE_NOT_FOUND**

This error is thrown if the specified snapshot is not found.

EZEBI_INVALID_ARGUMENT

This error is thrown if the specified snapshot path is invalid.

Examples**Example 1****Request (curl)**

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-01/Local/genericProject@Manual-P-ss"
  ]'
https://198.51.100.10/zebi/api/v2/listDependenciesAndSnapshotCountOnRollback
-k
```

Response

```
{
  "deleteableSnapCount": {
    "pool-01/Local/genericProject/genericLun-genericProject-clone": 1,
    "pool-01/Local/genericProject/genericLun": 1,
    "pool-01/Local/genericProject": 1
  },
  "deleteableClonesWithSnapCount": {
    "pool-01/Local/genericProject/genericLun-rollbackclone": 0,
    "pool-01/Local/genericProject/genericLun-genericProject-clone-rollbackclone": 0
  }
}
```

Example 2

Erroneous Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    ""pool-01/Local/genericProject@non-exist-snapshot""
  ]'
https://198.51.100.10198.51.100.10/zebi/api/v2/
listDependenciesAndSnapshotCountOnRollback -k
```

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "Dataset does not exist. ",
  "extendedData": {}
}
```

listSnapshots

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

First Available Version

API v1.0, IntelliFlash 2.1.0.0

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"
}
```

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.

First Available Version

API v2.0, IntelliFlash 3.5.0.1

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" }
```

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.

First Available Version

API v2.0, IntelliFlash 3.5.0.1

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" }
```

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.

First Available Version

API v2.0, IntelliFlash 3.5.0.1

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"}
```

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.

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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

Lists the replication status for the specified replication configuration.

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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.

First Available Version

API v1.2, IntelliFlash 2.1.2.1

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:

- [*getDiskCount*](#)
- [*getDisks*](#)
- [*getDisksByChassis*](#)
- [*getSMBConfig*](#)
- [*getUpgradeHistory*](#)
- [*identifyDisk*](#)
- [*identifyDiskByIndex*](#)
- [*isProjectExposedOverSMB*](#)
- [*isShareExposedOverSMB*](#)
- [*listSystemProperties*](#)
- [*setSMBConfig*](#)
- [*setSMBSharingOnProject*](#)
- [*setSMBSharingOnShare*](#)

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

getDiskCount

Returns the count of disks connected to the IntelliFlash array.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[getDisksByChassis](#), [getDisks](#), [identifyDisk](#), [identifyDiskByIndex](#)

Parameters

None

Returns

An integer count of all the disks found on the IntelliFlash array.

Exceptions Thrown

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST \  
-H "Authorization:Basic Auth_TOKEN" \  
-H "Content-Type:application/json" -d '[]' \  
-H 'cache-control: no-cache' \  
https://10.68.97.100/zebi/api/v2/getDiskCount -k
```

Response

The above request returns the HTTP status code 200 (OK) and a numeric count of the disks detected. For example, 27.

getDisks

Returns the details of disks connected to the IntelliFlash Array.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[getDisksByChassis](#), [getDiskCount](#), [identifyDisk](#), [identifyDiskByIndex](#)

Parameters**diskAliasPattern**

The disk name (alias) to obtain detailed JSON disk objects. This is the same disk alias name shown in IntelliFlash Web UI for each disk.

This parameter can be a complete specific disk name or partial regular expression to match several disks.

For example, you can specify `c2t5000C50040CF0707d0` to obtain a specific disk, `".*"` to obtain all disks, `c2t5000C50040.*` to obtain disks starting with `c2t5000C50040`, `c2t5000.*` to obtain disks starting with `c2t5000`, `.*5000C50040CF07.*` to obtain disks that contain `5000C50040CF07` in the name.

Returns

A JSON array of the [Disk_V2_1](#) objects that contain the details of the requested disks.

Exceptions Thrown**EZEBI_INVALID_ARGUMENT**

This exception is thrown if the parameters are invalid (for example, bad format or blank), or if the requested wild card regular expression is in an invalid format.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples**Example 1****Request (curl)**

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '[".*"]' \
https://198.51.100.10/zebi/api/v2/getDisks -k
```

Response

The above request returns the HTTP status code 200 (OK) and a list of all disks on that IntelliFlash array. For example:

```
[
  {
    "diskChassisIndex": 0,
    "diskBayIndex": 0,
    "diskChassisName": "[N5800]",
    "diskSize": "1.46 TiB",
    "poolName": "r2",
    "diskType": "NVMe",
    "diskAlias": "c4t000CCA0B01331E00d0",
    "deviceId": "w000cca0b01331e00"
  },
  {
    "diskChassisIndex": 0,
    "diskBayIndex": 1,
    "diskChassisName": "[N5800]",
    "diskSize": "1.46 TiB",
    "poolName": "",
    "diskType": "NVMe",
    "diskAlias": "c5t000CCA0B01335F00d0",
    "deviceId": "w000cca0b01335f00"
  },
  {
    "diskChassisIndex": 0,
    "diskBayIndex": 2,
    "diskChassisName": "[N5800]",
    "diskSize": "1.46 TiB",
    "poolName": "",
    "diskType": "NVMe",
    "diskAlias": "c6t000CCA0B01335A80d0",
    "deviceId": "w000cca0b01335a80"
  }
]
```

```
{
  "diskChassisIndex": 0,
  "diskBayIndex": 1,
  "diskChassisName": "IS1201-0022[HA2100]",
  "diskSize": "1.82 TB",
  "poolName": "ZebiSystem",
  "diskType": "HDD",
  "diskAlias": "c2t5000C50040CF04E7d0",
  "deviceId": "n5000c50040cf04e7"
},
{
  "diskChassisIndex": 0,
  "diskBayIndex": 2,
  "diskChassisName": "IS1201-0022[HA2100]",
  "diskSize": "1.82 TB",
  "poolName": "vmppool",
  "diskType": "HDD",
  "diskAlias": "c2t5000C50040CF0707d0",
  "deviceId": "n5000c50040cf0707"
```

```
    "deviceId": "n5000c50040cf0707"
  },
```

Example 2

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["c2t5000C50040CF0F97d0" ]' \
https://198.51.100.10/zebi/api/v2/getDisks -k
```

Error Response

In this example, the disk "c2t5000C50040CF0F97d0" does not exist. So the request returns the HTTP status code 200 (OK) and the following response of an empty array of disks:

```
[]
```

Example 3

Request (curl)

```
curl -X POST -H "Authorization:Basic AUTH_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '[""]' \
https://198.51.100.10/zebi/api/v2/getDisks -k
```

Response

In this example, the disk name was empty. So the request returns the HTTP status code 400 (Bad Request) and the following response:

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "getDisks.arg0 value '': Pattern name is not valid,
    an invalid null, empty or blank pattern name was detected.",
  "message": "Pattern name is not valid, an invalid null,
    empty or blank pattern name was detected.",
  "extendedData":
    {
      "EX_CAUSE_MESSAGE": null
    }
}
```

getDisksByChassis

Lists the disks connected to the specified disk chassis. The chassis are the IntelliFlash array itself (head) and the JBOD chassis connected to the array.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[getDisks](#), [getDiskCount](#), [identifyDisk](#), [identifyDiskByIndex](#)

Parameters

chassisIndex

The chassis index on the array that contains the disk bays to enumerate. Often the array has only one disk chassis, which would be chassis index 0. But an array with 3 chassis might typically have chassis index 0, 1, and 2, depending how they are connected. The chassis index can be retrieved by calling `getDisks` API first in the returned JSON array of [Disk_V2_1](#) objects.

Returns

A JSON array of the [Disk_V2_1](#) objects that contain the details of the disks found on the requested chassis.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the requested disk chassis index does not exist, or is out of range.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '[0]' \
```

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

Response

The above request returns the HTTP status code 200 (OK) and a JSON array of [Disk_V2_1](#) objects on the first chassis. For example:

```
{
  {
    "diskChassisIndex": 0,
    "diskBayIndex": 1,
    "diskChassisName": "IS1201-0022[HA2100]",
    "diskSize": "1.82 TB",
    "poolName": "ZebiSystem",
    "diskType": "HDD",
    "diskAlias": "c2t5000C50040CF04E7d0",
    "deviceId": "n5000c50040cf04e7"
  },
  {
    "diskChassisIndex": 0,
    "diskBayIndex": 2,
    "diskChassisName": "IS1201-0022[HA2100]",
    "diskSize": "1.82 TB",
    "poolName": "vmppool",
    "diskType": "HDD",
    "diskAlias": "c2t5000C50040CF0707d0",
    "deviceId": "n5000c50040cf0707"
  }
}
```

Example 2

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '[500]' \
https://198.51.100.10/zebi/api/v2/getDisksByChassis -k
```

Error Response

As the chassis index 500 does not exist, the request returns the HTTP status code 404 (not found) and the following exception:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "Failed to find the disk chassis requested (500).",
  "extendedData": {}
}
```

getSMBConfig

Lists all the SMB configuration values.

First Available Version

API v2.2, IntelliFlash 3.7.1.0

Related APIs

[setSMBConfig](#)

Parameters

None

Returns

Returns an [SMBConfig_V2_2](#) object that contains the SMB configuration details.

Exceptions Thrown

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

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

Response

```
{  
  "pdc": null,  
  "subsharesFeatureEnabled": false,  
  "smbProtocolMode": "SMB3",  
  "restrictAnonymous": true,  
  "restrictGuest": true  
}
```

getUpgradeHistory

Returns the history of IntelliFlash software installs and upgrades for the array.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[listSystemProperties](#)

Parameters

N/A

Returns

A JSON array of the [ArrayUpgrade_V2_1](#) objects that contain the upgrades done on the IntelliFlash array.

Exceptions Thrown

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '[]' \
https://198.51.100.10/zebi/api/v2/getUpgradeHistory -k
```

Response:

The above request returns the HTTP status code 200 (OK) and the [ArrayUpgrade_V2_1](#) objects that contain the upgrades done on the IntelliFlash array. For example:

```
[
  {
    "version": "3.7.0.0.170619",
    "timeInstalledOnNodeA": "Tue Jun 20 14:19:39 PDT 2017",
    "timeInstalledOnNodeB": "Tue Jun 20 13:43:25 PDT 2017"
```

```

    },
    {
      "version": "3.7.0.0.170425",
      "timeInstalledOnNodeA": "Wed Apr 26 22:07:17 PDT 2017",
      "timeInstalledOnNodeB": "Wed Apr 26 21:31:57 PDT 2017"
    },
    {
      "version": "3.6.0.0.170215",
      "timeInstalledOnNodeA": "Wed Feb 15 13:09:10 PST 2017",
      "timeInstalledOnNodeB": "Wed Feb 15 16:28:39 PST 2017"
    }
  ]

```

identifyDisk

Identifies a disk connected to the IntelliFlash array by flashing the bay light of the disk.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[getDisks](#), [getDisksByChassis](#), [getDiskCount](#), [identifyDiskByIndex](#)

Parameters

diskAlias

The name of the disk, which you want to identify. For example, c2t5000C50040CF0707d0 could be the disk name.

This is the same disk alias name shown in the IntelliFlash Web UI for each disk.

Returns

Returns an integer status, where:

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

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified disk was not found.

EZEBI_GENERAL

This exception is thrown if the operation failed.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the disk name parameter is invalid (bad format or blank).

Examples

Example 1

Request (curl)

```
curl -X POST \
-H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["c2t5000C500409B5663d0",true]' \
https://198.51.100.10/zebi/api/v2/identifyDisk -k
```

Response

The above request returns the HTTP status code 200 (OK) and 0. This indicates that the blinking disk light for the disk "c2t5000C500409B5663d0" was successfully turned on.

Example 2

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["XYZ",true]' \
https://198.51.100.10/zebi/api/v2/identifyDisk -k
```

Error Response

In this example, 'xyz' is not a valid disk name. The disk name must be in the **c[Controller#]t[Target#]d[Disk#]** format. For example, **c2t5000C50040CF0F97d0**. So the request returns the HTTP status code 400 (bad request) and the following response:

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "identifyDisk.arg0 value 'xyz': The disk alias name should
be
  in the format of c[Controller#]t[Target#]d[Disk#] (e.g.
c2t5000C50040CF0F97d0).
  Disk names can be found in the UI disk hardware page, or by enumerating
all disks.",
  "message": "The disk alias name should in
the format of c[Controller#]t[Target#]d[Disk#] (e.g.
c2t5000C50040CF0F97d0).
  Disk names can be found in the UI disk hardware page, or by enumerating
all disks.",
```

```

    "extendedData":
    {
        "EX_CAUSE_MESSAGE": null
    }
}

```

identifyDiskByIndex

Identifies a disk connected to the IntelliFlash Web UI array by flashing the bay light of the disk.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[getDisks](#), [getDisksByChassis](#), [getDiskCount](#), [identifyDisk](#)

Parameters

diskBayIndex

The disk bay index number of the disk you want to identify.

This is the same number displayed in the IntelliFlash Web UI and same disk index number returned in the disk objects by the `getDisks` API

chassisIndex

The disk chassis index number of the disk you want to identify. This is the same disk chassis index number returned in the disk objects by the `getDisks` API and displayed in the IntelliFlash Web UI.

blinkLED

JSON boolean to indicate whether to turn on or turn off the disk bay light used to identify the disk.

Returns

Returns an integer status where:

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

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the chassis index or disk bay index is out of range or not found.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '[2,2,true]' \
https://198.51.100.10/zebi/api/v2/identifyDiskByIndex -k
```

Response

The above request returns the HTTP status code 200 (OK) and 0. This indicates that the blinking disk light was successfully turned on.

Example 2

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-d '[2,10000,true]' \
https://198.51.100.10/zebi/api/v2/identifyDiskByIndex -k
```

Error Response

In this example, 10000 is not a valid existing disk chassis index . So the request returns the HTTP status code 400 (bad request) and the following response:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "Invalid chassis index of 10000 was passed,
the maximum chassis index currently available is 2.",
  "extendedData": {}
}
```

isProjectExposedOverSMB

Returns whether the SMB protocol is enabled for a project.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[setSMBSharingOnProject](#), [setSMBNetworkACLsOnProject](#), [addSMBNetworkACLOnProject](#), [removeSMBNetworkACLOnProject](#), [removeAllSMBNetworkACLsOnProject](#), [getSMBNetworkACLsOnProject](#)

Parameters

datasetPath

Path of the project. The format is <poolName>/Local/<projectName>.

This operation is not allowed for replica project datasets.

Returns

Returns True or False based on whether SMB protocol is enabled.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified project cannot be found.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the path specified belongs to replica dataset.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST \  
  -H 'authorization: Basic Auth_TOKEN \  
  -H 'cache-control: no-cache' \  
  -H 'content-type: application/json' \  
  -d '[  
    "pool-a/Local/demoProject"
```

```
    ]' \
https://198.51.100.10/zebi/api/v2/isProjectExposedOverSMB -k
```

Response:

The above request returns the HTTP status code 200 (OK) and returns a true or false value indicating whether SMB is enabled over the specified project.

Example 2

Erroneous Request (curl)

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

Error Response:

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "Cannot find the specified project 'pool-a/Local/UNKNOWNProject'.",
  "extendedData": {}
}
```

isShareExposedOverSMB

Returns whether the SMB protocol is enabled for a share.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[setSMBSharingOnShare](#), [setSMBNetworkACLsOnShare](#), [addSMBNetworkACLOnShare](#), [removeSMBNetworkACLOnShare](#), [removeAllSMBNetworkACLsOnShare](#), [getSMBNetworkACLsOnShare](#)

Parameters

datasetPath

Path of the share. The format is <poolName>/Local/<projectName>/<shareName>.

Returns

Returns True or False based on whether the SMB protocol is enabled.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified share cannot be found.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the path specified belongs to replica dataset.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST \  
  -H 'authorization: Basic Auth_TOKEN \  
  -H 'cache-control: no-cache' \  
  -H 'content-type: application/json' \  
  -d '[  
    "pool-a/Local/Project/demoShare"  
  ]' \  
  https://198.51.100.10/zebi/api/v2/isShareExposedOverSMB -k
```

Response

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

```
true
```

Example 2

Erroneous Request (curl)

```
curl -X POST \  

```

```
-H 'authorization: Basic Auth_TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Local/Project/UNKNOWNShare"
]' \
https://198.51.100.10/zebi/api/v2/isShareExposedOverSMB -k
```

Error Response

The above request returns the HTTP status code 500 (internal server error) and the following response:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "Cannot find the specified share 'pool-a/Local/Project/UNKNOWNShare'.",
  "extendedData": {}
}
```

listSystemProperties

Lists values of the requested system properties for an IntelliFlash array.

First Available Version

API v1.0, IntelliFlash 2.1.0.0

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"
]
```

setSMBConfig

Defines SMB configuration values.

First Available Version

API v2.2, IntelliFlash 3.7.1.0

Related APIs

[getSMBConfig](#)

Parameters

An [SMBConfig_V2_2](#) object that contains the SMB configuration details.

Returns

Returns an integer status where:

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

Exceptions Thrown**EZEBI_INVALID_ARGUMENT**

This exception is thrown if the path specified belongs to a replica dataset.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples**Example 1****Request (curl)**

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[{"smbProtocolMode": "SMB3"}]' \
https://198.51.100.10/zebi/api/v2/setSMBConfig -k
```

Response

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

Example 2**Erroneous Request (curl)**

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
{"smbProtocolMode": "SMB3"}
]' \
https://198.51.100.10/zebi/api/v2/setSMBConfig -k
```

Error Response

```
{
  "code": "EZEBI_GENERAL",
  "details": "",
  "message": "The following shares have NFS and SMB enabled and hence
cannot move to SMB3:
pool-a/Local/proj1/share-1, pool-a/Local/proj1/share-2."
  "extendedData": []
}
```

Example 3

Erroneous Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
{"smbProtocolMode": "SMB1"}
]' \
https://198.51.100.10/zebi/api/v2/setSMBConfig -k
```

Error Response

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "setSMBConfig.arg0.smbProtocolMode value 'SMB1':
Incorrect input: SMB protocol can only be CIFS or SMB3.",
  "message": "Incorrect input: SMB protocol can only be CIFS or SMB3.",
  "extendedData": {
    "EX_CAUSE_MESSAGE": null
  }
}
```

setSMBSharingOnProject

Enables or disables SMB protocol for project. If you disable SMB protocol for a project, any existing network ACLs on the project are removed as well.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[setSMBNetworkACLsOnProject](#), [addSMBNetworkACLOnProject](#), [removeSMBNetworkACLOnProject](#), [removeAllSMBNetworkACLsOnProject](#), [isProjectExposedOverSMB](#), [getSMBNetworkACLsOnProject](#)

Parameters

datasetPath

Path of the project. The format is <poolName>/Local/<projectName>.

This operation is not allowed for replica project datasets.

turnOn

Enables or disables SMB protocol on the project.

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown**EZEBI_RESOURCE_NOT_FOUND**

This exception is thrown if the specified project is not found.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the path specified belongs to a replica dataset.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples**Example 1****Request (curl)**

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/demoProject", true
  ]' \
  https://198.51.100.10/zebi/api/v2/setSMBSharingOnProject -k
```

Response

The above request returns the HTTP status code 200 (OK) and 0 indicating success. This enables SMB sharing on the specified project.

Example 2**Request (curl)**

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/demoProject", false
  ]' \
  https://198.51.100.10/zebi/api/v2/setSMBSharingOnProject -k
```

Response

The above request returns the HTTP status code 200 (OK) and 0 indicating success. This disables CMB sharing on the specified project.

Example 3

Erroneous Request

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/UNKNOWNProject", true
  ]' \
  https://198.51.100.10/zebi/api/v2/setSMBSharingOnProject -k
```

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "Cannot find the specified project 'pool-a/Local/UNKNOWNProject'.",
  "extendedData": {}
}
```

Example 4

Erroneous Request

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Replica/replicaProject", true
  ]' \
  https://198.51.100.10/zebi/api/v2/setSMBSharingOnProject -k
```

Error Response

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

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
```

```

    "details": "setSMBSharingOnProject.arg0 value 'pool-a/Replica/
replicaProject':
    Local dataset path expected. For example, valid formats are
    'pool-name/Local/project-name' or 'pool-name/Local/project-name/share-
or-lun-name'.",
    "message": "Local dataset path expected.
For example, valid formats are 'pool-name/Local/project-name' or
'pool-name/Local/project-name/share-or-lun-name'.",
    "extendedData": {"EX_CAUSE_MESSAGE": null}
}

```

setSMBSharingOnShare

Enables or disables SMB protocol for share. If the dataset contains any network ACLs, they are removed as well.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[setSMBNetworkACLsOnShare](#), [addSMBNetworkACLOnShare](#), [removeSMBNetworkACLOnShare](#), [removeAllSMBNetworkACLsOnShare](#), [isShareExposedOverSMB](#), [getSMBNetworkACLsOnShare](#)

Parameters

datasetPath

Path of the project. The format is <poolName>/Local/<projectName>/<shareName>.

This operation is not allowed for replica project datasets.

turnOn

Enables SMB protocol on the project.

displayName

Display name of the share.

enableGuestMode

Enables guests mode for the share.

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified share is not found.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the path specified belongs to a replica dataset.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST \
  -H 'Authorization:Basic Auth_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'Content-Type:application/json' \
  -d '["pool-a/Local/project/demoShare", true, "test", true]'
https://198.51.100.10/zebi/api/v2/setSMBSharingOnShare -k
```

Response

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

Example 2

Request (curl)

```
curl -X POST \
  -H 'Authorization:Basic Auth_TOKEN' \
  -H 'cache-control: no-cache' \
  -H 'Content-Type:application/json' \
  -d '["pool-a/Local/project/UNKNOWNShare", true, "test", true]'
https://198.51.100.10/zebi/api/v2/setSMBSharingOnShare -k
```

Response

The above request returns the HTTP status code 500 (internal server error) and the following response:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "Unable to open pool-a/Local/project/UNKNOWNShare :
dataset does not exist",
  "message": "Unable to open pool-a/Local/project/UNKNOWNShare :
dataset does not exist",
  "extendedData": {
    "EX_CAUSE_MESSAGE": "Unable to open pool-a/Local/project/UNKNOWNShare :
dataset does not exist",
    "EX_CAUSE_CODE_NAME": "EZFS_NOENT",
```

```
    "EX_CAUSE_CODE_NUMBER": "2009" }
}
```

Example 3

Erroneous Request

```
curl -X POST \
-H 'Authorization:Basic Auth_TOKEN' \
-H 'cache-control: no-cache'\
-H 'Content-Type:application/json' \
-d '["pool-a/Replica/replicaProj/replicaShare", true, "test", true]'
https://198.51.100.10/zebi/api/v2/setSMBSharingOnShare -k
```

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{ "code": "EZEBI_INVALID_ARGUMENT",
  "details": "setSMBSharingOnShare.arg0 value
'pool-a/Replica/replicaProj/replicaShare': Local dataset path expected.
For example, valid formats are 'pool-name/Local/project-name' or
'pool-name/Local/project-name/share-or-lun-name'.",
  "message": "Local dataset path expected.
For example, valid formats are 'pool-name/Local/project-name' or
'pool-name/Local/project-name/share-or-lun-name'.",
  "extendedData": { "EX_CAUSE_MESSAGE": null } }
```

Chapter 9

Network ACL Methods

Topics:

- [*addNFSNetworkACLOnProject*](#)
- [*addNFSNetworkACLOnShare*](#)
- [*addSMBNetworkACLOnProject*](#)
- [*addSMBNetworkACLOnShare*](#)
- [*getNFSNetworkACLsOnProject*](#)
- [*getNFSNetworkACLsOnShare*](#)
- [*getSMBNetworkACLsOnProject*](#)
- [*getSMBNetworkACLsOnShare*](#)
- [*inheritNetworkACLsettingsFromProject*](#)
- [*removeAllNFSNetworkACLsOnProject*](#)
- [*removeAllNFSNetworkACLsOnShare*](#)
- [*removeAllSMBNetworkACLsOnProject*](#)
- [*removeAllSMBNetworkACLsOnShare*](#)
- [*removeNFSNetworkACLOnProject*](#)
- [*removeNFSNetworkACLOnShare*](#)
- [*removeSMBNetworkACLOnProject*](#)
- [*removeSMBNetworkACLOnShare*](#)
- [*setNFSNetworkACLsOnProject*](#)
- [*setNFSNetworkACLsOnShare*](#)
- [*setSMBNetworkACLsOnProject*](#)
- [*setSMBNetworkACLsOnShare*](#)

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

addNFSNetworkACLOnProject

Adds network ACL to the NFS project.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[setNFSSharingOnProject](#), [setNFSNetworkACLsOnProject](#), [removeNFSNetworkACLOnProject](#), [removeAllNFSNetworkACLsOnProject](#), [isProjectExposedOverNFS](#), [getNFSNetworkACLsOnProject](#)

Parameters

datasetPath

Path of the project. The format is <poolName>/Local/<projectName>.

This operation is not allowed for replica project datasets.

type

Type of network ACL host, whether IP or FQDN.

host

The host to use for providing access.

accessMode

Access mode. 'rw' for Read-Write access, 'ro' for Read-Only access.

isRoot

Whether ACL is root access.

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified project cannot be found.

EZEBI_INVALID_ARGUMENT

This exception is thrown for the following conditions:

- If an incorrect IP address or access mode is specified.
- If the specified project does not support the protocol.
- If the path specified belongs to a replica dataset.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples**Example 1****Request (curl)**

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
"pool-a/Local/demoProject", "FQDN", "www.example.com", "rw", true
]' \
https://198.51.100.10/zebi/api/v2/addNFSNetworkACLOnProject -k
```

Response:

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

Example 2**Request (curl)**

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
"pool-a/Local/demoProject", "IP", "198.51.100.255", "rw", true
]' \
https://198.51.100.10/zebi/api/v2/addNFSNetworkACLOnProject -k
```

Response

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

Example 3**Erroneous Request (curl)**

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
"pool-a/Local/demoProject", "IP", "198.51.100.255", "rw", true
]' \
https://198.51.100.10/zebi/api/v2/addNFSNetworkACLOnProject -k
```

Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code":"EZEBI_RESOURCE_NOT_FOUND",
  "details":"NFS sharing is not enabled on the specified dataset.
  Please enable NFS sharing on the dataset and then try the operation
  again.",
  "message":"NFS sharing is not enabled for the specified project
  'pool-a/Local/demoProject'.",
  "extendedData":{}
}
```

Example 4

Erroneous Request

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
"pool-a/Local/UNKNOWNProject", "IP", "198.51.100.255", "rw", true
]' \
https://198.51.100.10/zebi/api/v2/addNFSNetworkACLOnProject -k
```

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code":"EZEBI_RESOURCE_NOT_FOUND",
  "details":"",
  "message":"Cannot find the specified project 'pool-a/Local/
UNKNOWNProject'.",
  "extendedData":{}
}
```

Example 5

Erroneous Request

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
```

```
"pool-a/Local/demoProject", "IP", "198.51.100.255", "read-write", true
] ' \
https://198.51.100.10/zebi/api/v2/addNFSNetworkACLOnProject -k
```

Error Response

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

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "addNFSNetworkACLOnProject.arg3
value 'read-write': Network ACL can only be \"rw\" (Read-Write) or \"ro
\" (Read-Only).",
  "message": "Network ACL can only be \"rw\" (Read-Write) or \"ro\" (Read-
Only).",
  "extendedData": {"EX_CAUSE_MESSAGE": null}
}
```

addNFSNetworkACLOnShare

Adds network ACL to the NFS share.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[setNFSSharingOnShare](#), [setNFSNetworkACLsOnShare](#), [removeNFSNetworkACLOnShare](#),
[removeAllNFSNetworkACLsOnShare](#), [isShareExposedOverNFS](#), [getNFSNetworkACLsOnShare](#)

Parameters

datasetPath

Path of the share. The format is <poolName>/Local/<projectName>/<shareName>.

This operation is not allowed for replica share datasets.

type

Type of network ACL host, whether IP or FQDN.

host

The host to use for providing access.

accessMode

Access mode. 'rw' for Read-Write access, 'ro' for Read-Only access.

isRoot

Whether ACL is root access.

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown**EZEBI_RESOURCE_NOT_FOUND**

This exception is thrown if the specified share cannot be found.

EZEBI_INVALID_ARGUMENT

This exception is thrown for the following conditions:

- If an incorrect IP address or access mode is specified.
- If the specified share does not support the protocol.
- If the path specified belongs to a replica dataset.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples**Example 1****Request (curl)**

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/Project/demoShare", "FQDN", "www.example.com", "rw", true
  ]' \
  https://198.51.100.10/zebi/api/v2/addNFSNetworkACLOnShare -k
```

Response

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

Example 2**Request (curl)**

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN \
  -H 'cache-control: no-cache' \
```

```
-H 'content-type: application/json' \
-d '[
"pool-a/Local/Project/demoShare", "IP", "198.51.100.255", "rw", true
]' \
https://198.51.100.10/zebi/api/v2/addNFSNetworkACLOnShare -k
```

Response

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

Example 3

Erroneous Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[pool-a/Local/Project/demoShare", "IP", "198.51.100.255", "rw", true]' \
https://198.51.100.10/zebi/api/v2/addNFSNetworkACLOnShare -k
```

Response

The above request returns the HTTP status code 500 (internal server error) and the following message:

```
{
  "code":"EZEBI_RESOURCE_NOT_FOUND",
  "details":"NFS sharing is not enabled on the specified dataset.
Please enable NFS sharing on the dataset and then try the operation
again.",
  "message":"NFS sharing is not enabled for the specified share
'pool-a/Local/Project/demoShare'.",
  "extendedData":{}
}
```

Example 4

Erroneous Request

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
"pool-a/Local/Project/UNKNOWNShare", "IP", "198.51.100.255", "rw", true
]' \
https://198.51.100.10/zebi/api/v2/addNFSNetworkACLOnShare -k
```

Error Response

The above request returns the HTTP status code 500 (internal server error) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "Cannot find the specified share 'pool-a/Local/Project/UNKNOWNShare'.",
  "extendedData": {}
}
```

Example 5

Erroneous Request

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Local/Project/demoShare", "IP", "198.51.100.255", "read-write",
  true
]' \
https://198.51.100.10/zebi/api/v2/addNFSNetworkACLOnShare -k
```

Error Response

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

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "addNFSNetworkACLOnShare.arg3
value 'read-write': Network ACL can only be \"rw\" (Read-Write) or \"ro\" (Read-Only).",
  "message": "Network ACL can only be \"rw\" (Read-Write) or \"ro\" (Read-Only).",
  "extendedData": {"EX_CAUSE_MESSAGE": null}
}
```

addSMBNetworkACLOnProject

Adds network ACL to the SMB project.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

setSMBSharingOnProject, setSMBNetworkACLsOnProject, removeSMBNetworkACLOnProject, removeAllSMBNetworkACLsOnProject, isProjectExposedOverSMB, getSMBNetworkACLsOnProject

Parameters

datasetPath

Path of the project. The format is <poolName>/Local/<projectName>.

This operation is not allowed for replica project datasets.

type

Type of network ACL host, whether IP or FQDN.

host

The host to use for providing access.

accessMode

Access mode. 'rw' for Read-Write access, 'ro' for Read-Only access.

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified project cannot be found.

EZEBI_INVALID_ARGUMENT

This exception is thrown for the following conditions:

- If an incorrect IP address or access mode is specified.
- If the specified project does not support the protocol.
- If the path specified belongs to a replica dataset.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN \
```

```
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
"pool-a/Local/demoProject", "FQDN", "www.example.com", "rw", true
]' \
https://198.51.100.10/zebi/api/v2/addSMBNetworkACLOnProject -k
```

Response

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

Example 2

Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
"pool-a/Local/demoProject", "IP", "198.51.100.255", "rw", true
]' \
https://198.51.100.10/zebi/api/v2/addSMBNetworkACLOnProject -k
```

Response

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

Example 3

Erroneous Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
"pool-a/Local/demoProject", "IP", "198.51.100.255", "rw", true
]' \
https://198.51.100.10/zebi/api/v2/addSMBNetworkACLOnProject -k
```

Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "SMB sharing is not enabled on the specified dataset.
Please enable SMB sharing on the dataset and then try the operation
again.",
  "message": "SMB sharing is not enabled for the specified project
```

```

    'pool-a/Local/demoProject'.",
    "extendedData":{}
}

```

Example 4

Erroneous Request

```

curl -X POST \
-H 'authorization: Basic Auth TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
"pool-a/Local/UNKNOWNProject", "IP", "198.51.100.255", "rw", true
]' \
https://198.51.100.10/zebi/api/v2/addSMBNetworkACLOnProject -k

```

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```

{
  "code":"EZEBI_RESOURCE_NOT_FOUND",
  "details":"",
  "message":"Cannot find the specified project 'pool-a/Local/UNKNOWNProject'.",
  "extendedData":{}
}

```

Example 5

Erroneous Request

```

curl -X POST \
-H 'authorization: Basic Auth TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
"pool-a/Local/demoProject", "IP", "198.51.100.255", "read-write", true
]' \
https://198.51.100.10/zebi/api/v2/addSMBNetworkACLOnProject -k

```

Error Response

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

```

{
  "code":"EZEBI_INVALID_ARGUMENT",
  "details":"addSMBNetworkACLOnProject.arg3

```

```

    value 'read-write': Network ACL can only be \"rw\" (Read-Write) or \"ro
\" (Read-Only).\",
    \"message\": \"Network ACL can only be \"rw\" (Read-Write) or \"ro\" (Read-
Only).\",
    \"extendedData\": {\"EX_CAUSE_MESSAGE\": null}
}

```

addSMBNetworkACLOnShare

Adds a network ACL to the SMB share.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[setSMBSharingOnShare](#), [setSMBNetworkACLsOnShare](#), [removeSMBNetworkACLOnShare](#), [removeAllSMBNetworkACLsOnShare](#), [isShareExposedOverSMB](#), [getSMBNetworkACLsOnShare](#)

Parameters

datasetPath

Path of the share. The format is <poolName>/Local/<projectName>/<shareName>.

This operation is not allowed for replica share datasets.

type

Type of network ACL host, whether IP or FQDN.

host

The host to use for providing access.

accessMode

Access mode. 'rw' for Read-Write access, 'ro' for Read-Only access.

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified share cannot be found.

EZEBI_INVALID_ARGUMENT

This exception is thrown for the following conditions:

- If an incorrect IP address or access mode is specified.
- If the specified share does not support the protocol.
- If the path specified belongs to a replica dataset.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Local/Project/demoShare", "FQDN", "www.example.com", "rw", true
]' \
https://198.51.100.10/zebi/api/v2/addSMBNetworkACLOnShare -k
```

Response

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

Example 2

Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Local/Project/demoShare", "IP", "198.51.100.255", "rw", true
]' \
https://198.51.100.10/zebi/api/v2/addSMBNetworkACLOnShare -k
```

Response

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

Example 3

Erroneous Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN \
```

```
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Local/Project/demoShare", "IP", "198.51.100.255", "rw", true
]' \
https://198.51.100.10/zebi/api/v2/addSMBNetworkACLOnShare -k
```

Response

The above request returns the HTTP status code 500 (internal server error) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "SMB sharing is not enabled on the specified dataset.
  Please enable SMB sharing on the dataset and then try the operation
  again.",
  "message": "SMB sharing is not enabled for the specified project
  'pool-a/Local/Project/demoShare'.",
  "extendedData": {}
}
```

Example 4

Erroneous Request

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Local/Project/UNKNOWNShare", "IP", "198.51.100.255", "rw", true
]' \
https://198.51.100.10/zebi/api/v2/addSMBNetworkACLOnShare -k
```

Error Response

The above request returns the HTTP status code 500 (internal server error) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "Cannot find the specified project 'pool-a/Local/Project/
  UNKNOWNShare'.",
  "extendedData": {}
}
```

Example 5

Erroneous Request

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/Project/demoShare", "IP", "198.51.100.255", "read-write",
    true
  ]' \
  https://198.51.100.10/zebi/api/v2/addSMBNetworkACLOnShare -k
```

Error Response

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

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "addSMBNetworkACLOnShare.arg3
    value 'read-write': Network ACL can only be \"rw\" (Read-Write) or \"ro
    \" (Read-Only).",
  "message": "Network ACL can only be \"rw\" (Read-Write) or \"ro\" (Read-
    Only).",
  "extendedData": {"EX_CAUSE_MESSAGE": null}
}
```

getNFSNetworkACLsOnProject

Returns all the network ACLs of the NFS project.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[setNFSSharingOnProject](#), [setNFSNetworkACLsOnProject](#), [addNFSNetworkACLOnProject](#),
[removeNFSNetworkACLOnProject](#), [removeAllNFSNetworkACLsOnProject](#), [isProjectExposedOverNFS](#)

Parameters

datasetPath

Path of the project. The format is <poolName>/Local/<projectName>.

Returns

List of network ACLs objects of type NetworkACL_V2_1.

Exceptions Thrown**EZEBI_RESOURCE_NOT_FOUND**

This exception is thrown if the specified project cannot be found.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples**Example 1****Request (curl)**

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

Response

The above request returns the HTTP status code 200 (OK) and returns the list of NFS network ACL objects.

```
[
  {
    "hostType": "IP",
    "host": "198.51.100.255",
    "accessMode": "ro",
    "rootAccessForNFS": true
  },
  {
    "hostType": "FQDN",
    "host": "www.example.com",
    "accessMode": "rw",
    "rootAccessForNFS": true
  }
]
```

Example 2**Erroneous Request (curl)**

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
```



```
-d '[
"pool-a/Local/UNKNOWNProject"
]' \
https://198.51.100.10/zebi/api/v2/getNFSNetworkACLsOnProject -k
```

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "Cannot find the specified project 'pool-a/Local/UNKNOWNProject'.",
  "extendedData": {}
}
```

getNFSNetworkACLsOnShare

Returns all the network ACLs of the NFS share.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[setNFSSharingOnShare](#), [setNFSNetworkACLsOnShare](#), [addNFSNetworkACLOnShare](#), [removeNFSNetworkACLOnShare](#), [removeAllNFSNetworkACLsOnShare](#), [isShareExposedOverNFS](#)

Parameters

datasetPath

Path of the share. The format is <poolName>/Local/<projectName>/<shareName>.

Returns

List of network ACLs objects of type NetworkACL_V2_1.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified share cannot be found.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/Project/demoShare"
  ]' \
  https://198.51.100.10/zebi/api/v2/getNFSNetworkACLsOnShare -k
```

Response

The above request returns the HTTP status code 200 (OK) and returns the list of NFS network ACL objects.

```
[
  {
    "hostType": "IP",
    "host": "198.51.100.255",
    "accessMode": "ro",
    "rootAccessForNFS": true
  },
  {
    "hostType": "FQDN",
    "host": "www.example.com",
    "accessMode": "rw",
    "rootAccessForNFS": true
  }
]
```

Example 2

Erroneous Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/Project/UNKNOWNShare"
  ]' \
  https://198.51.100.10/zebi/api/v2/getNFSNetworkACLsOnShare -k
```

Error Response

The above request returns the HTTP status code 500 (internal server error) and the following response:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "Cannot find the specified share 'pool-a/Local/Project/UNKNOWNShare'.",
  "extendedData": {}
}
```

getSMBNetworkACLsOnProject

Returns all the network ACLs of the SMB project.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[setSMBSharingOnProject](#), [setSMBNetworkACLsOnProject](#), [addSMBNetworkACLOnProject](#), [removeSMBNetworkACLOnProject](#), [removeAllSMBNetworkACLsOnProject](#), [isProjectExposedOverSMB](#)

Parameters

datasetPath

Path of the project. The format is <poolName>/Local/<projectName>.

Returns

List of network ACL objects of type [NetworkACL_V2_1](#).

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified project cannot be found.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

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

Response:

The above request returns the HTTP status code 200 (OK) and returns the list of SMB network ACL objects.

```
[
  {
    "hostType": "IP",
    "host": "198.51.100.255",
    "accessMode": "ro",
    "rootAccessForNFS": true
  },
  {
    "hostType": "FQDN",
    "host": "www.example.com",
    "accessMode": "rw",
    "rootAccessForNFS": true
  }
]
```

Example 2

Erroneous Request (curl)

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

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
```

```

    "details": "",
    "message": "Cannot find the specified project 'pool-a/Local/UNKNOWNProject'.",
    "extendedData": {}
  }

```

getSMBNetworkACLsOnShare

Returns all the network ACLs of the SMB share.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[setSMBSharingOnShare](#), [setSMBNetworkACLsOnShare](#), [addSMBNetworkACLOnShare](#), [removeSMBNetworkACLOnShare](#), [removeAllSMBNetworkACLsOnShare](#), [isShareExposedOverSMB](#)

Parameters

datasetPath

Path of the share. The format is <poolName>/Local/<projectName>/<shareName>.

Returns

List of network ACL objects of the [NetworkACL_V2_1](#) type.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified share cannot be found.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```

curl -X POST \
  -H 'authorization: Basic Auth_TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[

```

```
"pool-a/Local/Project/demoShare"
]' \
https://198.51.100.10/zebi/api/v2/getSMBNetworkACLsOnShare -k
```

Response

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

```
[
  {
    "hostType": "IP",
    "host": "198.51.100.255",
    "accessMode": "ro",
    "rootAccessForNFS": true
  },
  {
    "hostType": "FQDN",
    "host": "www.example.com",
    "accessMode": "rw",
    "rootAccessForNFS": true
  }
]
```

Example 2

Erroneous Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
"pool-a/Local/Project/UNKNOWNShare"
]' \
https://198.51.100.10/zebi/api/v2/getSMBNetworkACLsOnShare -k
```

Error Response

The above request returns the HTTP status code 500 (internal server error) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "SMB sharing is not enabled on the specified dataset.
Please enable SMB sharing on the dataset and then try the
operation again.",
  "message": "SMB sharing is not enabled for the specified share
'pool-a/Local/Project/UNKNOWNShare'.",
  "extendedData": {}
}
```

inheritNetworkACLsettingsFromProject

Enables or disables NFS protocol for a share.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

setNFSNetworkACLsOnShare, addNFSNetworkACLOnShare, removeNFSNetworkACLOnShare, removeAllNFSNetworkACLsOnShare, isShareExposedOverNFS, getNFSNetworkACLsOnShare, setNFSNetworkACLsOnShare, addNFSNetworkACLOnShare, removeNFSNetworkACLOnShare, removeAllNFSNetworkACLsOnShare, isShareExposedOverNFS, getNFSNetworkACLsOnShare

Parameters

datasetPath

Path of the project. The format is <poolName>/Local/<projectName>/<shareName>.

This operation is not allowed for replica share datasets.

Values returned

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified share cannot be found.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the path specified belongs to a replica dataset.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST \
  -H 'Authorization:Basic Auth_TOKEN' \
  -H 'cache-control: no-cache' \
```

```
-H 'Content-Type:application/json' \
-d '["pool-a/Local/project/demoShare"]' \
  https://198.51.100.10/zebi/api/v2/inheritNetworkACLsettingsFromProject
-k
```

Response

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

Example 2

Erroneous Request (curl)

```
curl -X POST \
-H 'Authorization:Basic Auth_TOKEN' \
-H 'cache-control: no-cache'\
-H 'Content-Type:application/json' \
-d '["pool-a/Local/project/UNKNOWNShare"]' \
  https://198.51.100.10/zebi/api/v2/inheritNetworkACLsettingsFromProject
-k
```

Response

The above request returns the HTTP status code 500 (internal server error) and the following response:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "Unable to open pool-a/Local/project/UNKNOWNShare :
             dataset does not exist",
  "message": "Unable to open pool-a/Local/project/UNKNOWNShare :
             dataset does not exist",
  "extendedData": {
    "EX_CAUSE_MESSAGE": "Unable to open pool-a/Local/project/UNKNOWNShare :
                       dataset does not exist",
    "EX_CAUSE_CODE_NAME": "EZFS_NOENT",
    "EX_CAUSE_CODE_NUMBER": "2009" }
}
```

Example 3

Erroneous Request

```
curl -X POST \
-H 'Authorization:Basic Auth_TOKEN' \
-H 'cache-control: no-cache'\
-H 'Content-Type:application/json' \
-d '["pool-a/Replica/replicaProj/replicaShare"]' \
  https://198.51.100.10/zebi/api/v2/inheritNetworkACLsettingsFromProject
-k
```


Error Response

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

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "inheritNetworkACLsettingsFromProject.arg0 value
    'pool-a/Replica/replicaProj/replicaShare': Local dataset path
    expected.
    For example, valid formats are 'pool-name/Local/project-name'
    or
    'pool-name/Local/project-name/share-or-lun-name'.",
  "message": "Local dataset path expected.
    For example, valid formats are 'pool-name/Local/project-name' or
    'pool-name/Local/project-name/share-or-lun-name'.",
  "extendedData": {
    "EX_CAUSE_MESSAGE": null
  }
}
```

removeAllNFSNetworkACLsOnProject

Removes all network ACLs from an NFS project.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[setNFSSharingOnProject](#), [setNFSNetworkACLsOnProject](#), [addNFSNetworkACLOnProject](#),
[removeNFSNetworkACLOnProject](#), [isProjectExposedOverNFS](#), [getNFSNetworkACLsOnProject](#)

Parameters

datasetPath

Path of the project. The format is <poolName>/Local/<projectName>.

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified project cannot be found.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the specified project does not support the protocol.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

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

Response

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

Example 2

Erroneous Request (curl)

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

Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "Cannot find the specified project 'pool-a/Local/UNKNOWNProject'.",
  "extendedData": {}
}
```

removeAllNFSNetworkACLsOnShare

Removes all network ACLs from an NFS share.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[setNFSSharingOnShare](#), [setNFSNetworkACLsOnShare](#), [addNFSNetworkACLOnShare](#), [removeNFSNetworkACLOnShare](#), [isShareExposedOverNFS](#), [getNFSNetworkACLsOnShare](#)

Parameters

datasetPath

Path of the share. The format is <poolName>/Local/<projectName>/<shareName>.

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified share cannot be found.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the specified share does not support the protocol.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '["pool-a/Local/Project/demoShare"]'
https://198.51.100.10/zebi/api/v2/removeAllNFSNetworkACLsOnShare -k
```

Response

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

Example 2

Erroneous Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '["pool-a/Local/Project/UNKNOWNShare"]'
https://198.51.100.10/zebi/api/v2/removeAllNFSNetworkACLsOnShare -k
```

Response

The above request returns the HTTP status code 500 (internal server error) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "NFS sharing is not enabled on the specified dataset.
  Please enable NFS sharing on the dataset and then try the operation
  again.",
  "message": "NFS sharing is not enabled for the specified share
  'pool-a/Local/project/UNKNOWNShare'.", "extendedData": {}}
```

removeAllSMBNetworkACLsOnProject

Removes all network ACLs from an SMB project.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[setSMBSharingOnProject](#), [setSMBNetworkACLsOnProject](#), [addSMBNetworkACLOnProject](#), [removeSMBNetworkACLOnProject](#), [isProjectExposedOverSMB](#), [getSMBNetworkACLsOnProject](#)

Parameters

datasetPath

Path of the project. The format is <poolName>/Local/<projectName>.

This operation is not allowed for replica project datasets.

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown**EZEBI_RESOURCE_NOT_FOUND**

This exception is thrown if the specified project cannot be found.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the specified project does not support the protocol.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples**Example 1****Request (curl)**

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

Response

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

Example 2**Erroneous Request (curl)**

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

Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "Cannot find the specified project 'pool-a/Local/UNKNOWNProject'.",
  "extendedData": {}
}
```

removeAllSMBNetworkACLsOnShare

Removes all network ACLs from an SMB share.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[setSMBSharingOnShare](#), [setSMBNetworkACLsOnShare](#), [addSMBNetworkACLOnShare](#), [removeSMBNetworkACLOnShare](#), [isShareExposedOverSMB](#), [getSMBNetworkACLsOnShare](#)

Parameters

datasetPath

Path of the share. The format is <poolName>/Local/<projectName>/<shareName>.

This operation is not allowed for replica share datasets.

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified share cannot be found.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the specified share does not support the protocol.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '["pool-a/Local/Project/demoShare"]' \
  https://198.51.100.10/zebi/api/v2/removeAllSMBNetworkACLsOnShare -k
```

Response

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

Example 2

Erroneous Request

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '["pool-a/Local/Project/UNKNOWNShare"]' \
  https://198.51.100.10/zebi/api/v2/removeAllSMBNetworkACLsOnShare -k
```

Error Response

The above request returns the HTTP status code 500 (internal server error) and the following message:

```
{"code":"EZEBI_RESOURCE_NOT_FOUND",
 "details":"SMB sharing is not enabled on the specified dataset.
 Please enable SMB sharing on the dataset and then try the operation
 again.",
 "message":"SMB sharing is not enabled for the specified share
 'pool-a/Local/project/UNKNOWNShare'.", "extendedData":{}}
```

removeNFSNetworkACLOnProject

Removes network ACL from an NFS project.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[setNFSSharingOnProject](#), [setNFSNetworkACLsOnProject](#), [addNFSNetworkACLOnProject](#), [removeAllNFSNetworkACLsOnProject](#), [isProjectExposedOverNFS](#), [getNFSNetworkACLsOnProject](#)

Parameters

datasetPath

Path of the project. The format is <poolName>/Local/<projectName>.

This operation is not allowed for replica project datasets.

type

Type of network ACL host, whether IP or FQDN.

host

The host to use for providing access.

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified project cannot be found, or if the specified ACL does not exist.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the path specified belongs to a replica dataset.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Local/demoProject", "IP", "198.51.100.255"
]' \
https://198.51.100.10/zebi/api/v2/removeNFSNetworkACLOnProject -k
```


Response

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

Example 2

Erroneous Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
"pool-a/Local/UNKNOWNProject", "IP", "198.51.100.255"
]' \
https://198.51.100.10/zebi/api/v2/removeNFSNetworkACLOnProject -k
```

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code":"EZEBI_RESOURCE_NOT_FOUND",
  "details":"",
  "message":"Cannot find the specified project 'pool-a/Local/
UNKNOWNProject'.",
  "extendedData":{}
}
```

Example 3

Erroneous Request

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
"pool-a/Local/demoProject", "FQDN", "www.example.com"
]' \
https://198.51.100.10/zebi/api/v2/removeNFSNetworkACLOnProject -k
```

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code":"EZEBI_RESOURCE_NOT_FOUND",
  "details":"",
```

```

    "message": "The specified ACL 'www.example.com' does not belong to the
dataset.",
    "extendedData": {}
}

```

removeNFSNetworkACLOnShare

Removes network ACL from an NFS share.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[setNFSSharingOnShare](#), [setNFSNetworkACLsOnShare](#), [addNFSNetworkACLOnShare](#),
[removeAllNFSNetworkACLsOnShare](#), [isShareExposedOverNFS](#), [getNFSNetworkACLsOnShare](#)

Parameters

datasetPath

Path of the share. The format is <poolName>/Local/<projectName>/<shareName>.

This operation is not allowed for replica share datasets.

type

Type of network ACL host, whether IP or FQDN.

host

The host to use for providing access.

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified share cannot be found, or if the specified ACL does not exist.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the path specified belongs to a replica dataset.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
"pool-a/Local/Project/demoShare", "IP", "198.51.100.255"
]' \
https://198.51.100.10/zebi/api/v2/removeNFSNetworkACLOnShare -k
```

Response

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

Example 2

Erroneous Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
"pool-a/Local/Project/demoShare", "IP", "198.51.100.255"
]' \
https://198.51.100.10/zebi/api/v2/removeNFSNetworkACLOnShare -k
```

Response

The above request returns the HTTP status code 500 (internal server error) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "NFS sharing is not enabled on the specified dataset.
  Please enable NFS sharing on the dataset and then try the operation
  again.",
  "message": "NFS sharing is not enabled for the specified share
  'pool-a/Local/project/demoShare'.", "extendedData": {}}
```

removeSMBNetworkACLOnProject

Removes network ACL from an SMB project.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[setSMBSharingOnProject](#), [setSMBNetworkACLsOnProject](#), [addSMBNetworkACLOnProject](#), [removeAllSMBNetworkACLsOnProject](#), [isProjectExposedOverSMB](#), [getSMBNetworkACLsOnProject](#)

Parameters

datasetPath

Path of the project. The format is <poolName>/Local/<projectName>.

This operation is not allowed for replica project datasets.

type

Type of network ACL host, whether IP or FQDN.

host

The host to use for providing access.

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified project cannot be found, or if the specified ACL does not exist.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the path specified belongs to a replica dataset.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/demoProject", "IP", "198.51.100.255"
  ]' \
  https://198.51.100.10/zebi/api/v2/removeSMBNetworkACLOnProject -k
```

Response

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

Example 2

Erroneous Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/UNKNOWNProject", "IP", "198.51.100.255"
  ]' \
  https://198.51.100.10/zebi/api/v2/removeSMBNetworkACLOnProject -k
```

Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "Cannot find the specified project 'pool-a/Local/UNKNOWNProject'.",
  "extendedData": {}
}
```

Example 3

Erroneous Request

```
curl -X POST \
  -H 'authorization: Basic Auth_TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    "pool-a/Local/demoProject", "IP", "198.51.100.255"
```

```
] ' \
https://198.51.100.10/zebi/api/v2/removeSMBNetworkACLOnProject -k
```

Error Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "The specified ACL '@198.51.100.255' does not belong to the dataset.",
  "extendedData": {}
}
```

removeSMBNetworkACLOnShare

Removes network ACL from an SMB share.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[setSMBSharingOnShare](#), [setSMBNetworkACLsOnShare](#), [addSMBNetworkACLOnShare](#), [removeAllSMBNetworkACLsOnShare](#), [isShareExposedOverSMB](#), [getSMBNetworkACLsOnShare](#)

Parameters

datasetPath

Path of the share. The format is <poolName>/Local/<projectName>/<shareName>.

This operation is not allowed for replica share datasets.

type

Type of network ACL host, whether IP or FQDN.

host

The host to use for providing access.

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown**EZEBI_RESOURCE_NOT_FOUND**

This exception is thrown if the specified share cannot be found, or if the specified ACL does not exist.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the share specified belongs to a replica dataset.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples**Example 1****Request (curl)**

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
"pool-a/Local/Project/demoShare", "IP", "198.51.100.255"
]' \
https://198.51.100.10/zebi/api/v2/removeSMBNetworkACLOnShare -k
```

Response

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

Example 2**Erroneous Request**

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
"pool-a/Local/Project/demoShare", "IP", "198.51.100.255"
]' \
https://198.51.100.10/zebi/api/v2/removeSMBNetworkACLOnShare -k
```

Error Response

The above request returns the HTTP status code 500 (internal server error) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "SMB sharing is not enabled on the specified dataset. Please enable SMB sharing on the dataset and then try the operation again.",
  "message": "SMB sharing is not enabled for the specified share 'pool-a/Local/project/demoShare'.",
  "extendedData": {}
}
```

setNFSNetworkACLsOnProject

Set the network ACLs on the NFS project. If the dataset contains any existing network ACLs, they are replaced with the new ones.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[setNFSSharingOnProject](#), [addNFSNetworkACLOnProject](#), [removeNFSNetworkACLOnProject](#), [removeAllNFSNetworkACLsOnProject](#), [isProjectExposedOverNFS](#), [getNFSNetworkACLsOnProject](#)

Parameters

datasetPath

Path of the project. The format is <poolName>/Local/<projectName>.

This operation is not allowed for replica project datasets.

networkACLs

Array of Network ACLs. Each object in the array is of the NetworkACL_V2_1 type.

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified project cannot be found.

EZEBI_INVALID_ARGUMENT

This exception is thrown for the following conditions:

- If an incorrect IP address or access mode is specified.
- If the specified project does not support the protocol.
- If the path specified belongs to a replica dataset.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples**Example 1****Request (curl)**

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Local/demoProject", [{"hostType": "IP", "host": "198.51.100.255",
    "accessMode": "ro", "rootAccessForNFS": true},
    {"hostType": "FQDN", "host": "www.example.com", "accessMode": "rw",
    "rootAccessForNFS": true}]
]' \
https://198.51.100.10/zebi/api/v2/setNFSNetworkACLsOnProject -k
```

Response

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

Example 2**Request (curl)**

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Local/demoProject",
    [{"hostType": "IP", "host": "198.51.100.255", "accessMode": "ro",
    "rootAccessForNFS": true}]
]' \
https://198.51.100.10/zebi/api/v2/setNFSNetworkACLsOnProject -k
```

Response

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

Example 3**Erroneous Request (curl)**

```
curl -X POST \
```

```
-H 'authorization: Basic Auth TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
"pool-a/Local/demoProject",
  [{"hostType": "IP", "host": "198.51.100.255", "accessMode": "ro",
    "rootAccessForNFS": true}, {"hostType": "FQDN",
"host": "www.example.com",
    "accessMode": "rw", "rootAccessForNFS": true}]
]' \
https://198.51.100.10/zebi/api/v2/setNFSNetworkACLsOnProject -k
```

Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "NFS sharing is not enabled on the specified dataset.
Please enable NFS sharing on the dataset and then try the operation
again.",
  "message": "NFS sharing is not enabled for the specified project
'pool-a/Local/demoProject'.",
  "extendedData": {}
}
```

Example 4

Erroneous Request

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
"pool-a/Local/demoProject",
  [{"hostType": "IP", "host": "198.51.100.255", "accessMode": "ro",
    "rootAccessForNFS": true}, {"hostType": "FQDN",
"host": "www.example.com",
    "accessMode": "rw", "rootAccessForNFS": true}]
]' \
https://198.51.100.10/zebi/api/v2/setNFSNetworkACLsOnProject -k
```

Error Response

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

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
```

```

"details": "setNFSNetworkACLsOnProject.arg1[0] value
'com.example.skywalk.api.v2.IPublicAPI_V2_1$NetworkACL_V2_1@6cb39d3b':
Internet Protocol Network Address is in the wrong format.
Expecting a address in the NNN.NNN.NNN.NNN format, where each
address whole number is no greater than 255.",
"message": "Internet Protocol Network Address is in the wrong format.
Expecting a address in the NNN.NNN.NNN.NNN format, where each address
whole number is no greater than 255.",
"extendedData": {"EX_CAUSE_MESSAGE": null}
}

```

Example 5

Erroneous Request

```

curl -X POST \
-H 'authorization: Basic Auth TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
"pool-a/Replica/replicaProject", [{"hostType": "IP",
"host": "198.51.100.255",
"accessMode": "ro", "rootAccessForNFS": true},
{"hostType": "FQDN", "host": "www.example.com",
"accessMode": "rw", "rootAccessForNFS": true}]
]' \
https://198.51.100.10/zebi/api/v2/setNFSNetworkACLsOnProject -k

```

Error Response

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

```

{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "setNFSNetworkACLsOnProject.arg0 value 'pool-a/Replica/
replicaProject':
Local dataset path expected. For example, valid formats are
'pool-name/Local/project-name' or
'pool-name/Local/project-name/share-or-lun-name'.",
  "message": "Local dataset path expected.
For example, valid formats are 'pool-name/Local/project-name' or
'pool-name/Local/project-name/share-or-lun-name'.",
  "extendedData": {"EX_CAUSE_MESSAGE": null}
}

```

setNFSNetworkACLsOnShare

Set the network ACLs on the NFS share. If the dataset contains any existing network ACLs, they are replaced with the new ones.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[setNFSSharingOnShare](#), [addNFSNetworkACLOnShare](#), [removeNFSNetworkACLOnShare](#), [removeAllNFSNetworkACLsOnShare](#), [isShareExposedOverNFS](#), [getNFSNetworkACLsOnShare](#)

Parameters**datasetPath**

Path of the share. The format is <poolName>/Local/<projectName>/<shareName>.

This operation is not allowed for replica project datasets.

networkACLs

Array of Network ACLs. Each object in the array is of the NetworkACL_V2_1 type.

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown**EZEBI_RESOURCE_NOT_FOUND**

This exception is thrown if the specified share cannot be found.

EZEBI_INVALID_ARGUMENT

This exception is thrown for the following conditions:

- If an incorrect IP address or access mode is specified.
- If the specified share does not support the protocol.
- If the path specified belongs to a replica dataset.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples**Example 1****Request (curl)**

```
curl -X POST \
```

```
-H 'Authorization:Basic Auth_TOKEN' \
-H 'cache-control: no-cache'\
-H 'Content-Type:application/json' \
-d '["pool-a/Local/project/demoShare",
  [{"hostType": "IP", "host":"198.51.100.255",
    "accessMode":"ro", "rootAccessForNFS":true},
   {"hostType": "FQDN", "host":"www.example.com", "accessMode":"rw",
    "rootAccessForNFS":true}]]'
https://198.51.100.10/zebi/api/v2/setNFSNetworkACLsOnShare -k
```

Response

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

Example 2

Erroneous Request (curl)

```
curl -X POST \
-H 'Authorization:Basic Auth_TOKEN' \
-H 'cache-control: no-cache'\
-H 'Content-Type:application/json' \
-d '["pool-a/Local/project/demoShare",
  [{"hostType": "IP", "host":"198.51.100.255", "accessMode":"ro",
    "rootAccessForNFS":true},
   {"hostType": "FQDN", "host":"www.example.com", "accessMode":"rw",
    "rootAccessForNFS":true}]]'
https://198.51.100.10/zebi/api/v2/setNFSNetworkACLsOnShare -k
```

Error Response

The above request returns the HTTP status code 500 (internal server error) and the following message:

```
{"code":"EZEBI_RESOURCE_NOT_FOUND","details":
  "NFS sharing is not enabled on the specified dataset.
  Please enable NFS sharing on the dataset and then try the operation again.
", "message":"NFS sharing is not enabled for the specified share
'pool-a/Local/project/demoShare'.", "extendedData":{}}
```

Example 3

Erroneous Request

```
curl -X POST \
-H 'Authorization:Basic Auth_TOKEN' \
-H 'cache-control: no-cache'\
-H 'Content-Type:application/json' \
-d '["pool-a/Local/project/demoShare",
  [{"hostType": "IP", "host":"198.51.100.255", "accessMode":"ro",
    "rootAccessForNFS":true},
   {"hostType": "FQDN", "host":"www.example.com",
    "accessMode":"rw", "rootAccessForNFS":true}]]'
```

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

Error Response

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

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "setNFSNetworkACLsOnShare.arg1[0] value
'com.example.skywalk.api.v2.IPublicAPI_V2_1$NetworkACL_V2_1@6cb39d3b':
Internet Protocol Network Address is in the wrong format.
Expecting a address in the NNN.NNN.NNN.NNN format, where each
address whole number is no greater than 255.",
  "message": "Internet Protocol Network Address is in the wrong format.
Expecting a address in the NNN.NNN.NNN.NNN format, where each address
whole number is no greater than 255.",
  "extendedData": {"EX_CAUSE_MESSAGE": null}
}
```

Example 4

Erroneous Request

```
curl -X POST \
-H 'Authorization:Basic Auth_TOKEN' \
-H 'cache-control: no-cache' \
-H 'Content-Type:application/json' \
-d '{"pool-a/Replica/replicaProj/replicaShare",
  [{"hostType": "IP", "host": "198.51.100.255", "accessMode": "ro",
    "rootAccessForNFS": true},
  {"hostType": "FQDN", "host": "www.example.com", "accessMode": "rw",
    "rootAccessForNFS": true}]]'
https://198.51.100.10/zebi/api/v2/setSMBNetworkACLsOnShare -k
```

Error Response

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

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "setNFSNetworkACLsOnShare.arg0 value 'pool-a/Replica/
replicaProject':
Local dataset path expected. For example, valid formats are
'pool-name/Local/project-name' or
'pool-name/Local/project-name/share-or-lun-name'.",
  "message": "Local dataset path expected.
For example, valid formats are 'pool-name/Local/project-name' or
'pool-name/Local/project-name/share-or-lun-name'.",
  "extendedData": {"EX_CAUSE_MESSAGE": null}
}
```

Example 5

Erroneous Request

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Replica/replicaProject",
  [{"hostType": "IP", "host": "10.10.10.10",
    "accessMode": "ro", "rootAccessForNFS": true},
   {"hostType": "FQDN", "host": "www.example.com",
    "accessMode": "rw", "rootAccessForNFS": true}]
]' \
https://198.51.100.10/zebi/api/v2/setSMBNetworkACLsOnProject -k
```

Error Response

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

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "setSMBNetworkACLsOnProject.arg0 value
    'pool-a/Replica/replicaProject': Local dataset path expected.
    For example, valid formats are 'pool-name/Local/project-name'
  or
    'pool-name/Local/project-name/share-or-lun-name'.",
  "message": "Local dataset path expected.
    For example, valid formats are 'pool-name/Local/project-name'
  or
    'pool-name/Local/project-name/share-or-lun-name'.",
  "extendedData": {"EX_CAUSE_MESSAGE": null}
}
```

setSMBNetworkACLsOnProject

Sets the network ACLs on the SMB project.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[setSMBSharingOnProject](#), [addSMBNetworkACLOnProject](#), [removeSMBNetworkACLOnProject](#), [removeAllSMBNetworkACLsOnProject](#), [isProjectExposedOverSMB](#), [getSMBNetworkACLsOnProject](#)

Parameters**datasetPath**

Path of the project. The format is <poolName>/Local/<projectName>.

This operation is not allowed for replica project datasets.

networkACLs

Array of Network ACLs. Each object in the array is of type NetworkACL_V2_1.

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown**EZEBI_RESOURCE_NOT_FOUND**

This exception is thrown if the specified project cannot be found.

EZEBI_INVALID_ARGUMENT

This exception is thrown for the following conditions:

- If an incorrect IP address or access mode is specified.
- If the specified project does not support the protocol.
- If the path specified belongs to a replica dataset.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples**Example 1****Request (curl)**

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Local/demoProject", [{"hostType": "IP", "host": "198.51.100.255",
    "accessMode": "ro", "rootAccessForNFS": true},
    {"hostType": "FQDN", "host": "www.example.com", "accessMode": "rw",
    "rootAccessForNFS": true}]
]' \
https://198.51.100.10/zebi/api/v2/setSMBNetworkACLsOnProject -k
```

Response

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

Example 2

Erroneous Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Local/demoProject",
    [{"hostType": "IP", "host": "198.51.100.255", "accessMode": "ro",
      "rootAccessForNFS": true}, {"hostType": "FQDN",
"host": "www.example.com",
      "accessMode": "rw", "rootAccessForNFS": true}]
]' \
https://198.51.100.10/zebi/api/v2/setSMBNetworkACLsOnProject -k
```

Response

The above request returns the HTTP status code 404 (not found) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "SMB sharing is not enabled on the specified dataset.
Please enable SMB sharing on the dataset and then try the operation
again.",
  "message": "SMB sharing is not enabled for the specified project
'pool-a/Local/demoProject'.",
  "extendedData": {}
}
```

Example 3

Erroneous Request

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Local/demoProject",
    [{"hostType": "IP", "host": "198.51.100.255", "accessMode": "ro",
      "rootAccessForNFS": true}, {"hostType": "FQDN",
"host": "www.example.com",
      "accessMode": "rw", "rootAccessForNFS": true}]
]' \
https://198.51.100.10/zebi/api/v2/setSMBNetworkACLsOnProject -k
```

Error Response

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

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "setSMBNetworkACLsOnProject.arg1[0] value
'com.example.skywalk.api.v2.IPublicAPI_V2_1$NetworkACL_V2_1@6cb39d3b':
Internet Protocol Network Address is in the wrong format.
Expecting a address in the NNN.NNN.NNN.NNN format, where each
address whole number is no greater than 255.",
  "message": "Internet Protocol Network Address is in the wrong format.
Expecting a address in the NNN.NNN.NNN.NNN format, where each address
whole number is no greater than 255.",
  "extendedData": {"EX_CAUSE_MESSAGE": null}
}
```

Example 4

Erroneous Request

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
"pool-a/Replica/replicaProject", [{"hostType": "IP",
"host": "198.51.100.255",
  "accessMode": "ro", "rootAccessForNFS": true},
{"hostType": "FQDN", "host": "www.example.com",
  "accessMode": "rw", "rootAccessForNFS": true}]
]' \
https://198.51.100.10/zebi/api/v2/setSMBNetworkACLsOnProject -k
```

Error Response

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

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "setSMBNetworkACLsOnProject.arg0 value 'pool-a/Replica/
replicaProject':
Local dataset path expected. For example, valid formats are
'pool-name/Local/project-name' or
'pool-name/Local/project-name/share-or-lun-name'.",
  "message": "Local dataset path expected.
For example, valid formats are 'pool-name/Local/project-name' or
'pool-name/Local/project-name/share-or-lun-name'.",
  "extendedData": {"EX_CAUSE_MESSAGE": null}
}
```

setSMBNetworkACLsOnShare

Sets the network ACLs on the SMB share.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[setSMBSharingOnShare](#), [addSMBNetworkACLOnShare](#), [removeSMBNetworkACLOnShare](#), [removeAllSMBNetworkACLsOnShare](#), [isShareExposedOverSMB](#), [getSMBNetworkACLsOnShare](#)

Parameters

datasetPath

Path of the share. The format is <poolName>/Local/<projectName>/<shareName>.

This operation is not allowed for replica share datasets.

networkACLs

Array of Network ACLs. Each object in the array is of type NetworkACL_V2_1.

Returns

COMMAND_STATUS.COMMAND_SUCCEED (0) on success.

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the specified share cannot be found.

EZEBI_INVALID_ARGUMENT

This exception is thrown for the following conditions:

- If an incorrect IP address or access mode is specified.
- If the specified share does not support the protocol.
- If the path specified belongs to a replica dataset.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Local/Project/DemoShare", [{"hostType": "IP",
    "host": "198.51.100.255",
      "accessMode": "ro", "rootAccessForNFS": true},
    {"hostType": "FQDN", "host": "www.example.com", "accessMode": "rw",
      "rootAccessForNFS": true}]
]' \
https://198.51.100.10/zebi/api/v2/setSMBNetworkACLsOnShare -k
```

Response

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

Example 2

Erroneous Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Local/Project/demoShare",
    [{"hostType": "IP", "host": "198.51.100.255", "accessMode": "ro",
      "rootAccessForNFS": true}, {"hostType": "FQDN",
    "host": "www.example.com",
      "accessMode": "rw", "rootAccessForNFS": true}]
]' \
https://198.51.100.10/zebi/api/v2/setSMBNetworkACLsOnShare -k
```

Response

The above request returns the HTTP status code 500 (internal server error) and the following message:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "SMB sharing is not enabled on the specified dataset.
  Please enable SMB sharing on the dataset and then try the operation
  again.",
  "message": "SMB sharing is not enabled for the specified share
  'pool-a/Local/Project/demoShare'."}
```

```
    "extendedData": {}
  }
}
```

Example 3

Erroneous Request

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Local/Project/demoShare",
    [{"hostType": "IP", "host": "198.51.100.255", "accessMode": "ro",
      "rootAccessForNFS": true}, {"hostType": "FQDN",
"host": "www.example.com",
      "accessMode": "rw", "rootAccessForNFS": true}]
]' \
https://198.51.100.10/zebi/api/v2/setSMBNetworkACLsOnShare -k
```

Error Response

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

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "setSMBNetworkACLsOnShare.arg1[0] value
'com.example.skywalk.api.v2.IPublicAPI_V2_1$NetworkACL_V2_1@6cb39d3b':
Internet Protocol Network Address is in the wrong format.
Expecting a address in the NNN.NNN.NNN.NNN format, where each
address whole number is no greater than 255.",
  "message": "Internet Protocol Network Address is in the wrong format.
Expecting a address in the NNN.NNN.NNN.NNN format, where each address
whole number is no greater than 255.",
  "extendedData": {"EX_CAUSE_MESSAGE": null}
}
```

Example 4

Erroneous Request

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  "pool-a/Replica/replicaProject/replicaShare",
    [{"hostType": "IP", "host": "198.51.100.255",
      "accessMode": "ro", "rootAccessForNFS": true},
```

```

    {"hostType": "FQDN", "host":"www.example.com",
     "accessMode":"rw", "rootAccessForNFS":true}]
  ]' \
https://198.51.100.10/zebi/api/v2/setSMBNetworkACLsOnShare -k

```

Error Response

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

```

{
  "code":"EZEBI_INVALID_ARGUMENT",
  "details":"setSMBNetworkACLsOnShare.arg0 value
'pool-a/Replica/replicaProject/replicaShare':
Local dataset path expected. For example, valid formats are
'pool-name/Local/project-name' or
'pool-name/Local/project-name/share-or-lun-name'.",
  "message":"Local dataset path expected.
For example, valid formats are 'pool-name/Local/project-name' or
'pool-name/Local/project-name/share-or-lun-name'.",
  "extendedData":{"EX_CAUSE_MESSAGE":null}
}

```

Chapter 10

SNMP Methods

Topics:

- [*addSNMPTrapListener*](#)
- [*disableSNMPService*](#)
- [*enableSNMPService*](#)
- [*getSNMPSettings*](#)
- [*isSNMPServiceEnabled*](#)
- [*modifySNMPCommunityString*](#)
- [*recreateSNMPTables*](#)
- [*removeSNMPTrapListener*](#)
- [*resyncSNMPTables*](#)

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

addSNMPTrapListener

Adds an SNMP trap listener address that receives the SNMP event notifications.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[enableSNMPService](#), [disableSNMPService](#), [isSNMPServiceEnabled](#), [removeSNMPTrapListener](#), [getSNMPSettings](#), [modifySNMPCommunityString](#)

Parameters

trapListenerIp

IP address of the Trap listener.

Only up to 10 listeners are allowed. Must be a valid IPv4 address. The IP address cannot be any of the reserved addresses. For example, addresses such as 127.0.0.1, 0.0.0.0 and 255.255.255.255 are not allowed.

trapListenerPort

Port number of the Trap listener. The trap port must be in the 1-65535 range.

Returns

Returns an integer status where:

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

Exceptions Thrown

EZEBI_RESOURCE_EXIST

This exception is thrown if the trap listener you added already exists.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the IPv4 address of the listener is invalid (incorrectly blank, invalid IPv4 format, the trap port not in the 1-65535 range, or the SNMP service not enabled).

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["10.68.50.10",162]' \
https://198.51.100.10/zebi/api/v2/addSNMPTrapListener -k
```

Response

The above request returns the HTTP status code 200 (OK) and 0, indicating that it successfully added the SNMP trap listener.

Example 2

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["10.68.50.10",162]' \
https://198.51.100.10/zebi/api/v2/addSNMPTrapListener -k
```

Response

In this example, the requested SNMP trap listener already exists. So the request returns the HTTP status code 400 (bad request) and the following response:

```
{
  "code": "EZEBI_RESOURCE_EXIST",
  "details": "",
  "message":
    "Trap listener 10.68.50.10:162 already exists in SNMP settings and
    cannot be added",
  "extendedData": {}
}
```

Example 3

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["10.68.500.500",162]' \
https://198.51.100.10/zebi/api/v2/addSNMPTrapListener -k
```

Response

In this example, the SNMP trap was requested for an invalid IP address. So the request returns the HTTP status code 400 (bad request) and the following response:

```

{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "addSNMPTrapListener.arg0 value '10.68.500.500':  
Internet Protocol Network Address is in the wrong format.  
Expecting a address in the NNN.NNN.NNN.NNN format,  
where each address whole number is no greater than 255.",
  "message": "Internet Protocol Network Address is in the wrong format.  
Expecting a address in the NNN.NNN.NNN.NNN format,  
where each address whole number is no greater than 255.",
  "extendedData": {
    "EX_CAUSE_MESSAGE": null
  }
}

```

disableSNMPService

Disables the SNMP service on the IntelliFlash array.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[enableSNMPService](#), [isSNMPServiceEnabled](#), [addSNMPTrapListener](#), [removeSNMPTrapListener](#), [getSNMPSettings](#), [modifySNMPCommunityString](#)

Parameters

None

Returns

Returns an integer status where:

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

Exceptions Thrown

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '[]' \
https://198.51.100.10/zebi/api/v2/disableSNMPService -k
```

Response

The above request returns the HTTP status code 200 (OK) and 0, indicating that the API successfully disabled the service.

enableSNMPService

Enables the SNMP service on the IntelliFlash array and sets the SNMP community string.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[disableSNMPService](#), [isSNMPServiceEnabled](#), [addSNMPTrapListener](#), [removeSNMPTrapListener](#), [getSNMPSettings](#), [modifySNMPCommunityString](#)

Parameters

communityString

Community string to be used on the enabled SNMP service. The SNMP community string cannot be longer than 64 characters. It cannot have special characters such as *, #, /, \, !, @, ~, (,), [,], {, }, =, and %.

Returns

Returns an integer status where:

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

Exceptions Thrown

EZEBI_INVALID_ARGUMENT

This exception is thrown if the parameter is invalid. The SNMP community string is invalid when it is incorrectly blank or beyond 64 characters long, or has non-alphanumeric characters other than, dash, underscore or period.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["public"]' \
https://198.51.100.10/zebi/api/v2/enableSNMPService -k
```

Response

The above request returns the HTTP status code 200 (OK) and 0, indicating that the API successfully enabled the service.

Example 2

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN"
-H "Content-Type:application/json" \
-d '["public#!@~"]' \
https://198.51.100.10/zebi/api/v2/enableSNMPService -k
```

Response

In this example, the requested SNMP community string format was invalid. So the request returns the HTTP status code 400 (bad request) and the following response:

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "enableSNMPService.arg0 value 'public#!@~':
SNMP Community String cannot be longer than 64 characters
and it cannot have special characters
like *,#,/,\,!,@,~,(),[],{,},=,%... (default='public')",
  "message": "SNMP Community String cannot be longer than 64 characters
and it cannot have special characters
like *,#,/,\,!,@,~,(),[],{,},=,%... (default='public')",
  "extendedData": {
    "EX_CAUSE_MESSAGE": null
  }
}
```

```
}
```

getSNMPSettings

Returns the SNMP settings of the current SNMP service.

This includes the trap listeners and the SNMP community string in the JSON object [SNMP_Setting_V2_1](#).

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[enableSNMPService](#), [disableSNMPService](#), [isSNMPServiceEnabled](#), [addSNMPTrapListener](#), [removeSNMPTrapListener](#), [modifySNMPCommunityString](#)

Parameters

None

Returns

Returns the JSON object [SNMP_Setting_V2_1](#), which contains an array of the SNMP trap listener ip:port values, and the SNMP community string.

Exceptions Thrown

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '[]' \
https://198.51.100.10/zebi/api/v2/getSNMPSettings -k
```

Response

The above request returns the HTTP status code 200 (OK) and a JSON object [SNMP_Setting_V2_1](#). For example:

```
{
  "communityString": "public3",
  "trapListeners": [
    "10.68.97.43:162",
    "10.68.97.44:162",
    "10.68.97.45:162",
    "10.68.97.46:162",
    "10.68.50.10:10000"
  ]
}
```

Example 2

Erroneous 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/getSNMPSettings -k
```

Response

In this example, the requested SNMP service is not running. So the request returns the HTTP status code 400 (bad request) and the following response:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "SNMP Service must be enabled for this operation. Please enable SNMP and try again.",
  "extendedData": {}
}
```

isSNMPServiceEnabled

Checks whether the SNMP Service is enabled.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[enableSNMPService](#), [disableSNMPService](#), [addSNMPTrapListener](#), [removeSNMPTrapListener](#), [getSNMPSettings](#), [modifySNMPCommunityString](#)

Parameters

None

Returns

Returns a JSON boolean status where:

- False indicates that the service is not enabled.
- True indicates that the service is enabled.

Exceptions Thrown

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '[]' \
https://198.51.100.10/zebi/api/v2/isSNMPServiceEnabled -k
```

Response

The above request returns the HTTP status code 200 (OK) and JSON boolean result (true = SNMP enabled), indicating that the service is enabled.

modifySNMPCommunityString

Modifies the SNMP community string to be a new value.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[enableSNMPService](#), [disableSNMPService](#), [isSNMPServiceEnabled](#), [addSNMPTrapListener](#), [removeSNMPTrapListener](#), [getSNMPSettings](#)

Parameters

communityString

Community string to be used on the enabled SNMP service. The SNMP community string cannot be longer than 64 characters. It cannot have special characters such as *, #, /, \, !, @, ~, (,), [,], {, }, =, and %.

Returns

Returns an integer status where:

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

Exceptions Thrown

EZEBI_INVALID_ARGUMENT

This exception is thrown if the community string parameter is invalid. The SNMP community string is invalid when it is incorrectly blank or beyond 64 characters long, or has non-alphanumeric characters other than dash, underscore or period.

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["public2"]' \
https://198.51.100.10/zebi/api/v2/modifySNMPCommunityString -k
```

Response

The above request returns the HTTP status code 200 (OK) and 0, indicating that it successfully modified the SNMP community string to “public2”.

Example 2

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["public#!@~"]' \
https://198.51.100.10/zebi/api/v2/modifySNMPCommunityString -k
```

Response

In this example, the requested SNMP community string format was invalid. So the request returns the HTTP status code 400 (bad request) and the following response:

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "enableSNMPService.arg0 value `public#!@~`:
SNMP Community String cannot be longer than 64 characters
and it cannot have special characters
like *, #, /, \\, !, @, ~, (, ), [, ], {, }, =, %... (default='public')",
  "message": "SNMP Community String cannot be longer than 64 characters
and it cannot have special characters
like *, #, /, \\, !, @, ~, (, ), [, ], {, }, =, %... (default='public')",
  "extendedData": {
    "EX_CAUSE_MESSAGE": null
  }
}
```

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.

First Available Version

API v2.1, IntelliFlash 3.5.4.0/3.7.0.x

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.

removeSNMPTrapListener

Removes an SNMP trap listener address that would have received SNMP event notifications.

First Available Version

API v2.1, IntelliFlash 3.7.0.x

Related APIs

[enableSNMPService](#), [disableSNMPService](#), [isSNMPServiceEnabled](#), [addSNMPTrapListener](#), [getSNMPSettings](#), [modifySNMPCommunityString](#)

Parameters

trapListenerIp

IP address of the Trap listener.

Only up to 10 listeners are allowed. Must be a valid IPv4 address. The IP address cannot be any of the reserved addresses. For example, addresses such as 127.0.0.1, 0.0.0.0 and 255.255.255.255 are not allowed.

trapListenerPort

Port number of the Trap listener. The trap port must be in the 1-65535 range.

Returns

Returns an integer status where:

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

Exceptions Thrown

EZEBI_RESOURCE_NOT_FOUND

This exception is thrown if the trap listener does not exist.

EZEBI_INVALID_ARGUMENT

This exception is thrown if the IPv4 address of the listener is invalid (incorrectly blank, invalid IPv4 format, the trap port not in the 1-65535 range, or the SNMP service not enabled).

EZEBI_GENERAL

This exception is thrown if the operation failed.

Examples

Example 1

Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["10.68.50.10",162]' \
https://198.51.100.10/zebi/api/v2/removeSNMPTrapListener -k
```

Response

The above request returns the HTTP status code 200 (OK) and 0, indicating that it successfully removed the SNMP trap listener.

Example 2

Erroneous Request (curl)

```
curl -X POST -H "Authorization:Basic Auth_TOKEN" \
-H "Content-Type:application/json" \
-H 'cache-control: no-cache' \
-d '["10.68.50.28",162]' \
```

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

Response

In this example, the requested SNMP trap listener does not exist. So the request returns the HTTP status code 400 (bad request) and the following response:

```
{
  "code": "EZEBI_RESOURCE_NOT_FOUND",
  "details": "",
  "message": "Failed to find SNMP trap listener to remove",
  "extendedData": {}
}
```

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.

First Available Version

API v2.1, IntelliFlash 3.5.4.0/3.7.0.x

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 11

Analytics Methods

Topics:

- [getOneMinuteSystemAnalyticsHistory](#)
- [getOneMinuteDataAnalyticsHistory](#)

The following sections describe analytics methods, parameters, return types, and examples.

getOneMinuteSystemAnalyticsHistory

Returns the previous minute of analytics history for the specified system analytics.

The available system analytics are as follows:

- **CPU:** Provides the CPU usage data for each controller.

The following values are returned:

- Controller-[AB]/User: Percent CPU time spent in user space.
- Controller-[AB]/System: Percent CPU time spent in kernel.
- Controller-[AB]/Total_Used: Total percent CPU time used.
- Controller-[AB]/Interrupts: Average interrupts fired per second.
- Controller-[AB]/System_Calls: Average system calls per second.

- **Cache Hits:** Provides information on filesystem cache usage for each controller.

The following values are returned:

- Controller-[AB]/SSD_Reads: Percentage of reads that hit SSD cache.
- Controller-[AB]/RAM_Reads: Percentage of reads that hit RAM cache.
- Controller-[AB]/Cache_Reads: Percentage of reads that hit either RAM or SSD cache.

- **Pool Performance:** Provides average disk IO statistics for each disk type in each pool.

The values are returned in the following format:

```
[pool-name] / [disk-type] / [statistic]
```

Disk-types have the following values:

- Data: Disks used for data-only. HDDs in hybrid pools.
- Cache: SSDs designated as read/write cache.
- Meta: SSDs designated for metadata only.
- Iflash: SSD disk type housing metadata and cache.

The following statistics are returned for each disk:

- Read_MBps
- Write_Mbps
- Read_Ops
- Write_Ops
- Read_Latency
- Write_Latency

MBps and Ops values are per-second averages. Latency values are per-operation averages in milliseconds.

- **Network:** Provides network IO statistics for each interface, interface group, and controller.

For interfaces, the values are returned in the following format:

```
Controller-[AB]/I/[name]/[statistic]
```

For interface groups, the values are returned in the following format:

```
Controller-[AB]/IG/[name]/[statistic]
```

For controllers, the values are returned in the following format:

```
Controller-[AB]/Total/[statistic]
```

The following statistics are returned for each interface, interface group, or controller:

- Read_Mbps
- Transmit_Mbps

The network statistics are provided in Mbps, while all the other data statistics are provided in MBps.

First Available Version

API v2.3, IntelliFlash 3.9.0.0

Related APIs

[*getOneMinuteDataAnalyticsHistory*](#)

Parameters

analyticsTypes

An array of requested analytics types to include. The allowed values are NETWORK, POOL_PERFORMANCE, CPU, and CACHE_HITS.

Returns

Returns arrays of [*SystemAnalyticsResult_V2_3*](#) objects.

Exceptions Thrown

EZEBI_GENERAL

This exception is thrown when an internal error is detected.

EZEBI_INVALID_ARGUMENT

This exception is thrown when a non-existent system analytic is requested.

Examples

Example 1

Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    ["NETWORK", "POOL_PERFORMANCE", "CPU", "CACHE_HITS"]
  ]' \
  https://198.51.100.10/zebi/api/v2/getOneMinuteSystemAnalyticsHistory -k
```

Response:

```
[
  {
    "systemAnalyticsType": "POOL_PERFORMANCE",
    "timestamps": [
      1527187950000,
      1527187955000,
      1527187960000,
      1527187965000,
      1527187970000,
      1527187975000,
      1527187980000,
      1527187985000,
      1527187990000,
      1527187995000,
      1527188000000,
      1527188005000
    ],
    "datapoints": {
      "pool-a/Iflash/Read_Latency": [
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
      ],
      "pool-a/Data/Write_MBps": [
        6.11,
        13.75,
        13.41,
        9.21,
        15.65,
```

```

        4.11,
        10.86,
        16.16,
        4.8,
        15.11,
        13.55,
        11.09
    ],
    "pool-a/Iflash/Write_Latency": [
        1.74,
        0.61,
        0.4,
        0.38,
        1.51,
        0.6,
        0.51,
        0.65,
        0.32,
        0.52,
        0.44,
        0.52
    ],
    "pool-a/Iflash/Read_MBps": [
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
    ],
    "pool-a/Data/Write_Latency": [
        8.07,
        102.73,
        111.88,
        86.99,
        45.92,
        33.55,
        74.72,
        98.97,
        55.25,
        92.84,
        115.95,
        86.53
    ],
    "pool-a/Data/Read_Ops": [
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
    ],

```

```

        0,
        0
    ],
    "pool-a/Data/Read_MBps": [
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
    ],
    "pool-a/Iflash/Write_Ops": [
        653.25,
        563,
        588.5,
        600.5,
        669.5,
        614.75,
        594,
        606.5,
        605.5,
        589.5,
        580.25,
        604.5
    ],
    "pool-a/Data/Read_Latency": [
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
    ],
    "pool-a/Data/Write_Ops": [
        11,
        13.5,
        14,
        10,
        22.5,
        5.5,
        12,
        17.5,
        5.5,
        16.5,
        13,
        12.5
    ],
    "pool-a/Iflash/Read_Ops": [
        0,

```

```

        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
    ],
    "pool-a/Iflash/Write_MBps": [
        18.64,
        19.22,
        20.19,
        20.54,
        19.29,
        20.38,
        19.98,
        20.16,
        20.7,
        19.73,
        19.78,
        20.48
    ]
},
"averages": {
    "pool-a/Iflash/Read_Latency": 0,
    "pool-a/Data/Write_MBps": 11.15,
    "pool-a/Iflash/Write_Latency": 0.68,
    "pool-a/Iflash/Read_MBps": 0,
    "pool-a/Data/Write_Latency": 76.12,
    "pool-a/Data/Read_Ops": 0,
    "pool-a/Data/Read_MBps": 0,
    "pool-a/Iflash/Write_Ops": 605.81,
    "pool-a/Data/Read_Latency": 0,
    "pool-a/Data/Write_Ops": 12.79,
    "pool-a/Iflash/Read_Ops": 0,
    "pool-a/Iflash/Write_MBps": 19.92
}
},
{
    "systemAnalyticsType": "NETWORK",
    "timestamps": [
        1527187950000,
        1527187955000,
        1527187960000,
        1527187965000,
        1527187970000,
        1527187975000,
        1527187980000,
        1527187985000,
        1527187990000,
        1527187995000,
        1527188000000,
        1527188005000
    ],
    "datapoints": {
        "Controller-B/I/igb0/Receive_Mbps": [
            8,

```

```

      8,
      8,
      7,
      8,
      8,
      8,
      8,
      8,
      8,
      8,
      8,
      8
    ],
    "Controller-B/IG/mgmt0/Transmit_Mbps": [
      292,
      299,
      303,
      279,
      302,
      296,
      302,
      302,
      302,
      294,
      303,
      294
    ],
    "Controller-A/Total/Transmit_Mbps": [
      7,
      8,
      8,
      8,
      7,
      8,
      8,
      8,
      8,
      8,
      8,
      8,
      8
    ],
    "Controller-A/I/igb0/Transmit_Mbps": [
      7,
      8,
      8,
      8,
      7,
      8,
      8,
      8,
      8,
      8,
      8,
      8,
      8
    ],
    "Controller-A/IG/mgmt0/Transmit_Mbps": [
      7,
      8,
      8,
      8,
      7,
      8,

```

```

        8,
        8,
        8,
        8,
        8,
        8
    ],
    "Controller-B/Total/Transmit_Mbps": [
        292,
        299,
        303,
        279,
        302,
        296,
        302,
        302,
        302,
        294,
        303,
        294
    ],
    "Controller-A/I/igb0/Receive_Mbps": [
        269,
        289,
        302,
        307,
        277,
        303,
        297,
        299,
        309,
        294,
        296,
        305
    ],
    "Controller-B/I/igb0/Transmit_Mbps": [
        292,
        299,
        303,
        279,
        302,
        296,
        302,
        302,
        302,
        294,
        303,
        294
    ],
    "Controller-A/IG/mgmt0/Receive_Mbps": [
        269,
        289,
        302,
        307,
        277,
        303,
        297,
        299,
        309,
        294,
        296,

```

```

        305
    ],
    "Controller-B/IG/mgmt0/Receive_Mbps": [
        8,
        8,
        8,
        7,
        8,
        8,
        8,
        8,
        8,
        8,
        8,
        8
    ],
    "Controller-A/Total/Receive_Mbps": [
        269,
        289,
        302,
        307,
        277,
        303,
        297,
        299,
        309,
        294,
        296,
        305
    ],
    "Controller-B/Total/Receive_Mbps": [
        8,
        8,
        8,
        7,
        8,
        8,
        8,
        8,
        8,
        8,
        8,
        8
    ],
    ],
    },
    "averages": {
        "Controller-B/I/igb0/Receive_Mbps": 7.92,
        "Controller-B/IG/mgmt0/Transmit_Mbps": 297.33,
        "Controller-A/Total/Transmit_Mbps": 7.83,
        "Controller-A/I/igb0/Transmit_Mbps": 7.83,
        "Controller-A/IG/mgmt0/Transmit_Mbps": 7.83,
        "Controller-B/Total/Transmit_Mbps": 297.33,
        "Controller-A/I/igb0/Receive_Mbps": 295.58,
        "Controller-B/I/igb0/Transmit_Mbps": 297.33,
        "Controller-A/IG/mgmt0/Receive_Mbps": 295.58,
        "Controller-B/IG/mgmt0/Receive_Mbps": 7.92,
        "Controller-A/Total/Receive_Mbps": 295.58,
        "Controller-B/Total/Receive_Mbps": 7.92
    }
},
{

```



```

"systemAnalyticsType": "CACHE_HITS",
"timestamps": [
  1527187950000,
  1527187955000,
  1527187960000,
  1527187965000,
  1527187970000,
  1527187975000,
  1527187980000,
  1527187985000,
  1527187990000,
  1527187995000,
  1527188000000,
  1527188005000
],
"datapoints": {
  "Controller-A/SSD_Reads": [
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0
  ],
  "Controller-B/SSD_Reads": [
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0
  ],
  "Controller-A/Cache_Reads": [
    100,
    100,
    100,
    100,
    100,
    100,
    100,
    100,
    100,
    100,
    100,
    100,
    100
  ],
  "Controller-B/RAM_Reads": [
    100,
    100,

```

```

        100,
        100,
        100,
        100,
        100,
        100,
        100,
        100,
        100,
        100,
        100
    ],
    "Controller-A/RAM_Reads": [
        100,
        100,
        100,
        100,
        100,
        100,
        100,
        100,
        100,
        100,
        100,
        100,
        100,
        100,
        100
    ],
    "Controller-B/Cache_Reads": [
        100,
        100,
        100,
        100,
        100,
        100,
        100,
        100,
        100,
        100,
        100,
        100,
        100,
        100,
        100
    ]
},
"averages": {
    "Controller-A/SSD_Reads": 0,
    "Controller-B/SSD_Reads": 0,
    "Controller-A/Cache_Reads": 100,
    "Controller-B/RAM_Reads": 100,
    "Controller-A/RAM_Reads": 100,
    "Controller-B/Cache_Reads": 100
}
},
{
    "systemAnalyticsType": "CPU",
    "timestamps": [
        1527187950000,
        1527187955000,
        1527187960000,
        1527187965000,
        1527187970000,
        1527187975000,
        1527187980000,
        1527187985000,
        1527187990000,
        1527187995000
    ]
}

```

```

1527187995000,
1527188000000,
1527188005000
],
"datapoints": {
  "Controller-A/Total_Used": [
    7,
    6,
    5,
    5,
    6,
    5,
    6,
    6,
    5,
    8,
    5,
    5
  ],
  "Controller-A/System": [
    5,
    6,
    5,
    5,
    6,
    5,
    6,
    6,
    5,
    7,
    5,
    5
  ],
  "Controller-B/Interrupts": [
    18073,
    2883,
    2698,
    17917,
    5275,
    2707,
    18019,
    2719,
    2788,
    18173,
    5145,
    2818
  ],
  "Controller-A/User": [
    2,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    1,
    0,
    0
  ]
},
],

```

```

    "Controller-B/System_Calls": [
        28848,
        30049,
        27126,
        28281,
        31055,
        33048,
        29474,
        30022,
        30586,
        30349,
        28219,
        28531
    ],
    "Controller-A/Interrupts": [
        3712,
        21457,
        2734,
        2772,
        18501,
        3448,
        3827,
        22768,
        2719,
        3590,
        9798,
        8864
    ],
    "Controller-A/System_Calls": [
        41815,
        44313,
        44414,
        45189,
        44876,
        42671,
        44464,
        45133,
        45545,
        48780,
        42323,
        44076
    ],
    "Controller-B/User": [
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
    ],
    "Controller-B/System": [
        7,
        8,
        7,
        7,

```

```

        9,
        9,
        7,
        8,
        7,
        7,
        7,
        7
    ],
    "Controller-B/Total_Used": [
        7,
        8,
        7,
        7,
        9,
        9,
        7,
        8,
        7,
        7,
        7,
        7
    ]
},
"averages": {
    "Controller-A/Total_Used": 5.75,
    "Controller-A/System": 5.5,
    "Controller-B/Interrupts": 8267.92,
    "Controller-A/User": 0.25,
    "Controller-B/System_Calls": 29632.33,
    "Controller-A/Interrupts": 8682.5,
    "Controller-A/System_Calls": 44466.58,
    "Controller-B/User": 0,
    "Controller-B/System": 7.5,
    "Controller-B/Total_Used": 7.5
}
}
]

```

Example 2

Erroneous Request (curl)

```

curl -X POST \
  -H 'authorization: Basic Auth_TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    ["NONE"]
  ]' \
  https://198.51.100.10/zebi/api/v2/getOneMinuteSystemAnalyticsHistory -k

```

Error Response:

```
{
```

```

    "code": "EZEBI_INVALID_ARGUMENT",
    "details": "",
    "message": "Unknown system analytics type: 'NONE'",
    "extendedData": {}
  }

```

getOneMinuteDataAnalyticsHistory

Returns the previous minute of analytics history for the specified data entities.

The API returns the following values for each given data entity:

- Read_MBps
- Write_MBps
- Total_MBps
- Read_Ops
- Write_Ops
- Total_Ops
- Read_Latency
- Write_Latency
- Average_Latency

MBps and Ops values are per-second averages. Latency values are per-operation averages in milliseconds.

First Available Version

API v2.3, IntelliFlash 3.9.0.0

Related APIs

[getOneMinuteSystemAnalyticsHistory](#)

Parameters

datasets

A dataset can be a pool, project, share, or LUN. The path for each dataset should be in the following format:

```
[pool[/project[/dataset]]]
```

VMs

Provide the VM name in any of the following formats:

```
[vm-name]
```

```
[pool-name]/[vm-name]

[esx-host]/[vm-name]

[pool-name]/[esx-host]/[vm-name]
```

If multiple VMs match a given name, then all the matching VMs are included.

protocols

A protocol name such as NFS, SMB, iSCSI, or FC.

Returns

Returns arrays of [DataAnalyticsResult_V2_3](#) objects.

Exceptions Thrown

EZEBI_GENERAL

This exception is thrown when an internal error is detected.

EZEBI_INVALID_ARGUMENT

This exception is thrown when a non-existent protocol is requested.

Examples

Example 1

Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth_TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  ["pool-a", "pool-a/nfs2", "pool-a/nfs2/sh1"],
    null,
    ["NFS", "iSCSI"]
]' \
https://198.51.100.10/zebi/api/v2/getOneMinuteDataAnalyticsHistory -k
```

Response:

```
[
  {
    "entityType": "PROJECT",
    "entityName": "pool-a/nfs2",
    "timestamps": [
      1527188375000,
      1527188380000,
```

```

1527188385000,
1527188390000,
1527188395000,
1527188400000,
1527188405000,
1527188410000,
1527188415000,
1527188420000,
1527188425000,
1527188430000
],
"datapoints": {
  "Read_Ops": [
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0
  ],
  "Write_Ops": [
    1143,
    1153,
    1124,
    1122,
    1033,
    1057,
    1136,
    1151,
    1163,
    1156,
    1141,
    1145
  ],
  "Read_MBps": [
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0
  ],
  "Write_Latency": [
    0.52,
    0.57,
    0.51,
    0.54,
    1.88,
    1.34,

```



```

        0.52,
        0.59,
        0.5,
        0.51,
        0.52,
        0.53
    ],
    "Total_MBps": [
        35.74,
        36.03,
        35.15,
        35.09,
        32.3,
        33.04,
        35.51,
        35.99,
        36.35,
        36.14,
        35.66,
        35.8
    ],
    "Read_Latency": [
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
    ],
    "Total_Ops": [
        1143,
        1153,
        1124,
        1122,
        1033,
        1057,
        1136,
        1151,
        1163,
        1156,
        1141,
        1145
    ],
    "Write_MBps": [
        35.74,
        36.03,
        35.15,
        35.09,
        32.3,
        33.04,
        35.51,
        35.99,
        36.35,
        36.14,
        35.66,

```

[illegible]

```

    "Write_Ops": [
        1143,
        1153,
        1124,
        1122,
        1033,
        1057,
        1136,
        1151,
        1163,
        1156,
        1141,
        1145
    ],
    "Read_MBps": [
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
    ],
    "Write_Latency": [
        0.52,
        0.57,
        0.51,
        0.54,
        1.88,
        1.34,
        0.52,
        0.59,
        0.5,
        0.51,
        0.52,
        0.53
    ],
    "Total_MBps": [
        35.74,
        36.03,
        35.15,
        35.09,
        32.3,
        33.04,
        35.51,
        35.99,
        36.35,
        36.14,
        35.66,
        35.8
    ],
    "Read_Latency": [
        0,
        0,
        0,
        0,

```

```

        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
    ],
    "Total_Ops": [
        1143,
        1153,
        1124,
        1122,
        1033,
        1057,
        1136,
        1151,
        1163,
        1156,
        1141,
        1145
    ],
    "Write_MBps": [
        35.74,
        36.03,
        35.15,
        35.09,
        32.3,
        33.04,
        35.51,
        35.99,
        36.35,
        36.14,
        35.66,
        35.8
    ],
    "Average_Latency": [
        0.52,
        0.57,
        0.51,
        0.54,
        1.88,
        1.34,
        0.52,
        0.59,
        0.5,
        0.51,
        0.52,
        0.53
    ]
},
"averages": {
    "Read_Ops": 0,
    "Write_Ops": 1127,
    "Read_MBps": 0,
    "Write_Latency": 0.71,
    "Total_MBps": 35.23,
    "Read_Latency": 0,
    "Total_Ops": 1127,
    "Write_MBps": 35.23,

```



```

    0
  ],
  "Write_Latency": [
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0
  ],
  "Total_MBps": [
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0
  ],
  "Read_Latency": [
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0
  ],
  "Total_Ops": [
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0
  ],
  "Write_MBps": [
    0,
    0,

```

```

        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
    ],
    "Average_Latency": [
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
    ]
},
"averages": {
    "Read_Ops": 0,
    "Write_Ops": 0,
    "Read_MBps": 0,
    "Write_Latency": 0,
    "Total_MBps": 0,
    "Read_Latency": 0,
    "Total_Ops": 0,
    "Write_MBps": 0,
    "Average_Latency": 0
}
},
{
    "entityType": "POOL",
    "entityName": "pool-a",
    "timestamps": [
        1527188375000,
        1527188380000,
        1527188385000,
        1527188390000,
        1527188395000,
        1527188400000,
        1527188405000,
        1527188410000,
        1527188415000,
        1527188420000,
        1527188425000,
        1527188430000
    ],
    "datapoints": {
        "Read_Ops": [
            0,
            0,
            0,
            0,

```

```

        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
    ],
    "Write_Ops": [
        1143,
        1153,
        1124,
        1122,
        1033,
        1057,
        1136,
        1151,
        1163,
        1156,
        1141,
        1145
    ],
    "Read_MBps": [
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
    ],
    "Write_Latency": [
        0.52,
        0.57,
        0.51,
        0.54,
        1.88,
        1.34,
        0.52,
        0.59,
        0.5,
        0.51,
        0.52,
        0.53
    ],
    "Total_MBps": [
        35.74,
        36.03,
        35.15,
        35.09,
        32.3,
        33.04,
        35.51,
        35.99,
        36.35,

```



```

        36.14,
        35.66,
        35.8
    ],
    "Read_Latency": [
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
    ],
    "Total_Ops": [
        1143,
        1153,
        1124,
        1122,
        1033,
        1057,
        1136,
        1151,
        1163,
        1156,
        1141,
        1145
    ],
    "Write_MBps": [
        35.74,
        36.03,
        35.15,
        35.09,
        32.3,
        33.04,
        35.51,
        35.99,
        36.35,
        36.14,
        35.66,
        35.8
    ],
    "Average_Latency": [
        0.52,
        0.57,
        0.51,
        0.54,
        1.88,
        1.34,
        0.52,
        0.59,
        0.5,
        0.51,
        0.52,
        0.53
    ]
},

```

```

    "averages": {
      "Read_Ops": 0,
      "Write_Ops": 1127,
      "Read_MBps": 0,
      "Write_Latency": 0.71,
      "Total_MBps": 35.23,
      "Read_Latency": 0,
      "Total_Ops": 1127,
      "Write_MBps": 35.23,
      "Average_Latency": 0.71
    }
  },
  {
    "entityType": "PROTOCOL",
    "entityName": "NFS",
    "timestamps": [
      1527188375000,
      1527188380000,
      1527188385000,
      1527188390000,
      1527188395000,
      1527188400000,
      1527188405000,
      1527188410000,
      1527188415000,
      1527188420000,
      1527188425000,
      1527188430000
    ],
    "datapoints": {
      "Read_Ops": [
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
      ],
      "Write_Ops": [
        1143,
        1153,
        1124,
        1122,
        1033,
        1057,
        1136,
        1151,
        1163,
        1156,
        1141,
        1145
      ],
      "Read_MBps": [
        0,
        0,

```

```

0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0
],
"Write_Latency": [
0.52,
0.57,
0.51,
0.54,
1.88,
1.34,
0.52,
0.59,
0.5,
0.51,
0.52,
0.53
],
"Total_MBps": [
35.74,
36.03,
35.15,
35.09,
32.3,
33.04,
35.51,
35.99,
36.35,
36.14,
35.66,
35.8
],
"Read_Latency": [
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0
],
"Total_Ops": [
1143,
1153,
1124,
1122,
1033,
1057,
1136,

```

```

        1151,
        1163,
        1156,
        1141,
        1145
    ],
    "Write_MBps": [
        35.74,
        36.03,
        35.15,
        35.09,
        32.3,
        33.04,
        35.51,
        35.99,
        36.35,
        36.14,
        35.66,
        35.8
    ],
    "Average_Latency": [
        0.52,
        0.57,
        0.51,
        0.54,
        1.88,
        1.34,
        0.52,
        0.59,
        0.5,
        0.51,
        0.52,
        0.53
    ]
},
"averages": {
    "Read_Ops": 0,
    "Write_Ops": 1127,
    "Read_MBps": 0,
    "Write_Latency": 0.71,
    "Total_MBps": 35.23,
    "Read_Latency": 0,
    "Total_Ops": 1127,
    "Write_MBps": 35.23,
    "Average_Latency": 0.71
}
}
]

```

Example 2

Erroneous Request (curl)

```

curl -X POST \
  -H 'authorization: Basic Auth_TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    null,

```

```
    null,  
    ["VFS"]  
  ]' \  
https://198.51.100.10/zebi/api/v2/getOneMinuteDataAnalyticsHistory -k
```

Error Response:

```
{  
  "code": "EZEBI_INVALID_ARGUMENT",  
  "details": "",  
  "message": "Unknown protocol requested: 'VFS'",  
  "extendedData": {}  
}
```

Chapter 12

Notification Methods

Topics:

- [*getRecentNotifications*](#)
- [*getRecentCriticalNotifications*](#)

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

getRecentNotifications

Returns the most recent events within the specified duration that exceed the specified priority and match the event code filter.

First Available Version

API v2.3, IntelliFlash 3.9.0.0

Related APIs

[getRecentCriticalNotifications](#)

Parameters

durationInMinutes

The duration in minutes.

lowestPriority

Minimum priority. Accepted values are LOW, MEDIUM, HIGH, and CRITICAL. The values are case sensitive.

eventCodeFilter

Free-text filter for event code. For example, SNP and POL. A null filter value returns all results.

Returns

Returns an array of [Notification_V2_3](#) objects.

Exceptions Thrown

EZEBI_INVALID_ARGUMENT

This exception is thrown if the priority is not a known value, or if the duration is zero or negative.

Examples

Example 1

Request (curl)

```
curl -X POST \  
  -H 'authorization: Basic Auth_TOKEN \  
  -H 'cache-control: no-cache' \  
  -H 'content-type: application/json' \  
  -d '['
```



```

8400,
"CRITICAL",
""
]' \
https://198.51.100.10/zebi/api/v2/getRecentNotifications -k

```

Response:

```

[
  {
    "timestamp": 1536728451968,
    "eventCode": "DSK2208W22011",
    "priority": "Critical",
    "description": "JBOD Information mismatch across ha controllers for
jbod with chasis serial: 1123581321345589.",
    "errorMessage": null,
    "details": {
      "User": "SYSTEM",
      "Message": "Controller-A: [Chasis Id = TS1406-0090, Model = ]",
      "Controller-B": "[Chasis Id = ST1403-0036, Model = ]"
    }
  },
  {
    "timestamp": 1536728417815,
    "eventCode": "DSK2208W22011",
    "priority": "Critical",
    "description": "JBOD Information mismatch across ha controllers for
jbod with chasis serial: 1123581321345589.",
    "errorMessage": null,
    "details": {
      "User": "SYSTEM",
      "Message": "Controller-A: [Chasis Id = TS1406-0090, Model = ]",
      "Controller-B": "[Chasis Id = ST1403-0036, Model = ]"
    }
  },
  {
    "timestamp": 1536693680334,
    "eventCode": "CTR3600E36001",
    "priority": "Critical",
    "description": "Controller '2400-b' is down.",
    "errorMessage": null,
    "details": {
      "User": "SYSTEM",
      "Host 2": "2400-b",
      "Host 1": "2400-a"
    }
  },
  {
    "timestamp": 1536693568659,
    "eventCode": "CTR3600E36001",
    "priority": "Critical",
    "description": "Controller '2400-a' is down.",
    "errorMessage": null,
    "details": {
      "User": "SYSTEM",
      "Host 2": "2400-a",
      "Host 1": "2400-b"
    }
  }
]

```

```
}
]
```

Example 2

Erroneous Request (curl)

```
curl -X POST \
  -H 'authorization: Basic Auth TOKEN \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -d '[
    7400,
    "VERY HIGH",
    ""
  ]' \
  https://198.51.100.10/zebi/api/v2/getRecentNotifications -k
```

Error Response:

```
{
  "code": "EZEBI_INVALID_ARGUMENT",
  "details": "getRecentNotifications.arg1 value 'VERY HIGH':
    Invalid priority requested: 'VERY HIGH'",
  "message": "Invalid priority requested: 'VERY HIGH'",
  "extendedData": {
    "EX_CAUSE_MESSAGE": null
  }
}
```

getRecentCriticalNotifications

Returns the most recent critical events within the specified duration.

First Available Version

API v2.3, IntelliFlash 3.9.0.0

Related APIs

[getRecentNotifications](#)

Parameters

durationInMinutes

The duration in minutes.

Returns

Returns an array of [Notification_V2_3](#) objects.

Exceptions Thrown

EZEBI_INVALID_ARGUMENT

This exception is thrown if the duration is zero or negative.

Examples

Example 1

Request (curl)

```
curl -X POST \
-H 'authorization: Basic Auth TOKEN \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '[
  55000
]' \
https://198.51.100.10/zebi/api/v2/getRecentCriticalNotifications -k
```

Response:

```
[
  {
    "timestamp": 1532602657656,
    "eventCode": "DSK2204E23001",
    "priority": "Critical",
    "description": "Disk error occurred. A disk read error occurred.",
    "errorMessage": null,
    "details": {
      "VdevDevId": "id1,kdev@w000cca0b01a37e80/a",
      "Product Id": "N5200-D1",
      "User": "SYSTEM",
      "Message": "Disk error occurred. A disk read error occurred.",
      "Make": "HGST-HUSMR7619BDP3Y1",
      "Fault Disk Information": "Unavailable",
      "Chassis Id": "TG1806-1002",
      "VdevType": "disk",
      "Server Id": "NVMe-B",
      "PoolName": "pool-2",
      "VdevPath": "/dev/dsk/c8t000CCA0B01A37E80d0s0",
      "ParentType": "--",
      "RecordBy": "SystemMonitor"
    }
  },
  {
    "timestamp": 1532595977954,
    "eventCode": "DSK2204E23001",
    "priority": "Critical",
    "description": "Disk error occurred. A disk read error occurred.",
```

```

    "errorMessage": null,
    "details": {
      "VdevDevId": "idl,kdev@w000cca0b01a37e80/a",
      "Product Id": "N5200-D1",
      "User": "SYSTEM",
      "Message": "Disk error occurred. A disk read error occurred.",
      "Make": "HGST-HUSMR7619BDP3Y1",
      "Fault Disk Information": "Unavailable",
      "Chassis Id": "TG1806-1002",
      "VdevType": "disk",
      "Server Id": "NVMe-B",
      "PoolName": "pool-2",
      "VdevPath": "/dev/dsk/c8t000CCA0B01A37E80d0s0",
      "ParentType": "--",
      "RecordBy": "SystemMonitor"
    }
  },
  {
    "timestamp": 1532595438903,
    "eventCode": "DSK2204E23001",
    "priority": "Critical",
    "description": "Disk error occurred. A disk read error occurred.",
    "errorMessage": null,
    "details": {
      "VdevDevId": "idl,kdev@w000cca0b01a37e80/a",
      "Product Id": "N5200-D1",
      "User": "SYSTEM",
      "Message": "Disk error occurred. A disk read error occurred.",
      "Make": "HGST-HUSMR7619BDP3Y1",
      "Fault Disk Information": "Unavailable",
      "Chassis Id": "TG1806-1002",
      "VdevType": "disk",
      "Server Id": "NVMe-B",
      "PoolName": "pool-2",
      "VdevPath": "/dev/dsk/c8t000CCA0B01A37E80d0s0",
      "ParentType": "--",
      "RecordBy": "SystemMonitor"
    }
  },
  {
    "timestamp": 1532502366523,
    "eventCode": "DSK2204E23001",
    "priority": "Critical",
    "description": "Disk error occurred. A disk read error occurred.",
    "errorMessage": null,
    "details": {
      "VdevDevId": "idl,kdev@w000cca0b01a37e80/a",
      "Product Id": "N5200-D1",
      "User": "SYSTEM",
      "Message": "Disk error occurred. A disk read error occurred.",
      "Make": "HGST-HUSMR7619BDP3Y1",
      "Fault Disk Information": "Unavailable",
      "Chassis Id": "TG1806-1002",
      "VdevType": "disk",
      "Server Id": "NVMe-B",
      "PoolName": "pool-2",
      "VdevPath": "/dev/dsk/c8t000CCA0B01A37E80d0s0",
      "ParentType": "--",
      "RecordBy": "SystemMonitor"
    }
  }
}

```

```
]
```

Example 2

Erroneous Request (curl)

```
curl -X POST \  
-H 'authorization: Basic Auth_TOKEN \  
-H 'cache-control: no-cache' \  
-H 'content-type: application/json' \  
-d '[  
  0  
  ]' \  
https://198.51.100.10/zebi/api/v2/getRecentCriticalNotifications -k
```

Error Response:

```
{  
  "code": "EZEBI_INVALID_ARGUMENT",  
  "details": "",  
  "message": "Invalid duration: '0' minutes",  
  "extendedData": {}  
}
```

Chapter 13

Objects

Topics:

- [*ArrayUpgrade_V2_1*](#)
- [*CopyDestination_V2_1*](#)
- [*CopySource_V2_1*](#)
- [*CopyStatus_V2_1*](#)
- [*DataAnalyticsResult_V2_3*](#)
- [*DatasetProperty_V2_1*](#)
- [*DatasetSpaceInfo_V2_1*](#)
- [*DatasetStatus*](#)
- [*Disk_V2_1*](#)
- [*FCInitiator_V2_1*](#)
- [*FCTarget_V2_1*](#)
- [*FloatingIP_V2_2*](#)
- [*InitiatorGroup_V2_1*](#)
- [*IscsiInitiator_V1_0*](#)
- [*ISCSIInitiator_V2_1*](#)
- [*ISCSITarget_V2_1*](#)
- [*ISCSITargetCreate_V2_1*](#)
- [*ITView_V2_1*](#)
- [*LocalGroup_V1_2*](#)
- [*LocalUser_V1_2*](#)
- [*LunStatus*](#)
- [*NetworkACL_V2_1*](#)
- [*Notification_V2_3*](#)
- [*Pool_V1_0*](#)
- [*PoolSpaceInfo_V2_1*](#)
- [*Project_V1_0*](#)
- [*Project_V1_2*](#)
- [*Project_V2_1*](#)
- [*ProjectCloneProgressStatus_v1_2*](#)
- [*ReplicationConfig_V1_2*](#)
- [*ReplicationStatus_v1_2*](#)
- [*Schedule_V2_1*](#)
- [*Share_V1_0*](#)
- [*Share_V2_1*](#)
- [*ShareOptions*](#)
- [*SharePermissions*](#)
- [*SMBConfig_V2_2*](#)

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

- [SnapShotDeletionStatus](#)
- [SnapshotProgressStatus](#)
- [SnapshotSchedule_V2_1](#)
- [SNMP_Setting_V2_1](#)
- [SystemAnalyticsResult_V2_3](#)
- [TargetGroup_V2_2](#)
- [Volume_V1_0](#)
- [Volume_V2_1](#)
- [UserACL \(Read Only\) v2.1](#)

ArrayUpgrade_V2_1

This class represents the upgrade and install history of the array. Each object instance is one IntelliFlash OS install or upgrade.

| Field | Type | Description |
|----------------------|--------|---|
| Version | String | Version of the install or upgrade. |
| timeInstalledOnNodeA | String | Date and time the upgrade or install was done for node A. |
| timeInstalledOnNodeB | String | Date and time the upgrade or install was done for node B. |

CopyDestination_V2_1

Object defining destination datasets.

| Field | Description |
|---------------------------|--|
| hostName | Destination partner system name. Leave this field empty for same array. |
| poolName | Destination pool. |
| projectName | Destination project. |
| subProjectNamePrefix | Destination sub project name prefix. Leave this field empty for project copy. |
| subProjectNameNumberStart | Destination sub project name suffix start. The value should be a number. |
| subProjectNameNumberEnd | Destination sub project name suffix end. The value should be a number. |
| subProjectNameWildcard | Destination sub project name wildcard (bash style). |

CopySource_V2_1

Object defining the source dataset.

| Field | Description |
|----------------|--|
| poolName | Source dataset pool. |
| projectName | Source dataset project. |
| subProjectName | Source sub project name. Leave this field empty for project copy. |

CopyStatus_V2_1

Object representing the current status of the copy operation.

| Field | Description |
|-------------------|---|
| completedCopies | Completed copies for the task. |
| status | Current status of the task. It has the following values: <ul style="list-style-type: none"> • Unknown • Start • Connected • Sending • Local Replication • Completed • Error • Paused |
| allDatasets | All datasets created by the task. |
| completedDatasets | Datasets completed by the task. |
| pendingDatasets | Datasets yet to be created by the task. |
| percentComplete | Percentage of task completion. |
| errorCode | Error code of the task, if there is any error. |
| startTime | Start time of the task. |
| endTime | End time of the task. |

DataAnalyticsResult_V2_3

| Field | Type | Description |
|------------|---------------------------------|--|
| entityType | String | Data entity type. Possible values are VM, Dataset, Project, Pool, and Protocol. |
| entityName | String | Data entity name. For a dataset, pool, or a project, this is a path with the <code>[pool[/project[/dataset]]]</code> format. For a VM, this is a VM path with the <code>[pool]/[esx-host]/[vm-name]</code> format. For a protocol, this is a protocol name, such as NFS, SMB, iSCSI, or FC. |
| timestamps | Array of longs | Milliseconds since Unix epoch (midnight Jan 1 1970). |
| Datapoints | Map (String-->Array of numbers) | Datapoint list mapping. Each datapoint list has one element per timestamp. Possible values are Read_MBps, Write_MBps, Read_IOPs, Write_IOPs, Read_Latency, and Write_Latency. |
| Averages | Map (String-->Array of numbers) | Averages of the datapoint listings. |

DatasetProperty_V2_1

| Field | Description |
|---------------|-----------------------|
| propertyKey | Key of the property |
| propertyValue | Value of the property |

DatasetSpaceInfo_V2_1

| Field | Description |
|-------------------------------|---|
| originalUsedByDataAndSnapshot | Original used by dataset data and snapshot before compression (bytes). |
| usedByDataAndSnapshot | Amount used by the dataset data and snapshot after compression (bytes). |
| compressionSavingsPercentage | Amount saved from compression as a percentage. |

| Field | Description |
|-------------------|---|
| available | <p>For datasets with a quota, this represents the amount of space remaining in the quota.</p> <p>For shares and luns without a quota, this represents the amount of space remaining in the project's quota (if it exists).</p> <p>If a quota does not exist on a project, this value represents the amount of space (in bytes) remaining on the pool.</p> |
| usedByData | Amount used by this dataset's data after compression (bytes). |
| usedBySnapshot | Amount used by this dataset's snapshots after compression (bytes). |
| usedByReservation | Share and project field. Amount used by reservation (bytes). |
| quota | Share and project field. The user assigned quota for the dataset (bytes). The quota is 0 if it doesn't exist. |
| volSize | Volume only field. The size of the volume (bytes). |

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 | An integer return value as defined in COMMAND_STATUS . |
| datasetPath | String | A string that contains the dataset path. A dataset path should have the format <code>PoolName/Local/ProjectName/VolumeName</code> for volumes and <code>PoolName/Local/ProjectName/ShareName</code> for shares. |
| overwriteException | String | Contains details of the exception, if an exception occurs. |

| Field | Type | Description |
|-----------------|---------|---|
| overwriteStatus | Integer | An integer value as defined in OVERWRITE_STATUS . |

Disk_V2_1

| Field | Type | Description |
|------------------|---------|---|
| diskChassisIndex | Integer | Index of the chassis to enumerate, as listed by getDisks API. |
| diskBayIndex | Integer | Disk bay index relative to each disk chassis (as listed by the Web UI, and is zero relative). |
| diskChassisName | String | Name of the disk chassis as given by <ul style="list-style-type: none"> • Web UI • serial number [chassis name] |
| diskSize | String | Disk size with units (for example, 2TB). |
| poolName | String | Pool that is using the disk currently. |
| diskType | String | Disk type (for example, HDD, RW_META_SSD, or NVMe). |
| diskAlias | String | Disk name used to identify the disk. For example, "c2t5000CCA013067A90d0". |
| deviceId | String | Disk device id that is unique across array controllers. |

FCInitiator_V2_1

This class represents the initiator for the FC protocol.

| Field | Type | Description |
|--------------------|--------|--|
| initiatorName | String | Initiator name (for example, "wwn.5001438001FFAAAA"). |
| initiatorGroupName | String | Initiator group name. |

FCTarget_V2_1

This class represents the target for the FC protocol.

| Field | Type | Description |
|-----------------|--------|---|
| targetName | String | Target name (for example, "wwn.21000024FF236C15") |
| targetStatus | String | Target status (for example, "online" or "offline"). |
| targetNode | String | Node (or IntelliFlash controller) where the target is active. |
| targetGroupName | String | Target group name. |
| targetSpeed | String | Target speed (for example, 4GB, 8GB, or "not established"). |
| targetPortType | String | Whether the target port type is HBA or NPIV. |

FloatingIP_V2_2

| Field | Type | Description |
|-------------------|--------|--|
| resourceGroupName | String | The name of the resource group. |
| failoverMode | String | Lists the condition for a failover. The failover condition can be Immediately, Never, or Wait till all IP addresses fail. |
| ipAddress | String | The floating IP addresses. |
| netmask | String | The netmask of the floating IP address. |
| poolName | String | Pools associated with this IP address. |

InitiatorGroup_V2_1

| Field | Type | Description |
|--------------------|--------|--|
| initiatorGroupName | String | Initiator group name as listed in the Web UI. |
| intendedProtocol | String | The protocol the initiator group is configured to use (e.g. iSCSI, FC, or Unknown) |

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> |

ISCSIInitiator_V2_1

This class represents the iSCSI initiator attributes.

| Field | Type | Description |
|--------------------|--------|---|
| initiatorName | String | <p>Initiator Name (iqn or eui iscsi name)</p> <p>This must be in the iqn.yyyy.mmm.[reverse-domain-name]:unique-name or eui.16-hexadecimal-digits and no more than 255 characters.</p> |
| initiatorGroupName | String | <p>Initiator or host group name.</p> <p>This cannot have special characters such as <code>' , * , # , / , \ , ! , @ , ~ , (,) , [,] , { , } , = ,</code> and <code>%</code>. The string must be between 1 to 512 characters long.</p> |
| chapUserName | String | <p>Optional chap user name if the initiator uses chap for authentication.</p> <p>If specified this chap user name can't be blank or cannot have special characters such as <code>' , \ , / , ! , # , \$, % , ^ , & , * , (,) , ; , ,</code> and <code>@</code>.</p> |
| chapSecret | String | <p>Optional chap secret if the initiator uses CHAP authentication. This is only used on write-modify-create methods. The string is always null or empty for read methods.</p> <p>The chap secret must between 12 and 16 characters if specified, and cannot have special characters such as such as <code>' , \ , / , ! , # , \$, % , ^ , & , * , (,) , ; , ,</code> and <code>@</code>.</p> |

ISCSITarget_V2_1

This class represents the target for the iSCSI protocol and is returned when enumerating iSCSI targets.

| Field | Type | Description |
|--------------------------|-----------|--|
| targetName | String | This is the full iSCSI target name enumerated in the format: <code>iqn.yyyy-mm.[reverse-domain-name]:user-specified-suffix.</code> |
| targetSuffixName | String | Target suffix or user-specified portion of the iSCSI target name after the iqn colon. For example, <code>iqn.2012-12.com.tegile:targetSuffixName.</code> |
| targetAlias | String | Target Alias, as alternate user friendly name of target. This is often the same as the target name suffix. |
| targetGroupName | String | Target group name. |
| targetAuthenticationMode | String | Target authentication type. The values must be 'none', 'chap', or 'mutual' string names. 'none' indicates no user or password required, 'chap' indicates one way authentication (no user or password required), 'mutual' is two way authentication (provide chap user and secret). |
| targetChapName | String | User name for the chap authentication. Only required if 'mutual' chap authentication is chosen. |
| targetChapSecret | String | Password for the chap authentication. Only required if 'mutual' chap authentication is chosen. |
| targetNetworkBinding | String[] | Network binding string (ip:port) indicating the network ports to bind the target with. |

ISCSITargetCreate_V2_1

This class represents the target for iSCSI protocol and can be used to create an iSCSI target.

| Field | Type | Description |
|--------------------------|--------|--|
| targetSuffixName | String | <p>Target suffix. This is just the user specified portion of the iSCSI target name after the iqn colon.</p> <p>For example, <code>iqn.2012-12.com.tegile:targetSuffixName.</code></p> <p>This cannot have special characters such as ' ', *, #, /, \, !, @, ~, (,), [,], {, }, =, and %. The string must be between 1 to 255 characters long.</p> |
| targetAlias | String | Target Alias, as alternate user friendly name of target (often same as target name suffix). |
| targetGroupName | String | <p>Target group name</p> <p>This cannot have special characters such as *, #, /, \, !, @, ~, (,), [,], {, }, =, and %. The string must be 1 to 512 characters long.</p> |
| targetAuthenticationMode | String | <p>Target authentication type, which must be 'none', 'chap', or 'mutual' string names.</p> <p>'none' indicates no user or password required, 'chap' indicates one way authentication (no user or password required), and 'mutual' is two way authentication (provide chap user and secret).</p> |
| targetChapName | String | <p>Optional user name for the chap authentication.</p> <p>Only required if 'mutual' chap authentication was chosen. If specified this chap user name can't be blank or cannot have special characters such as ' ', \, /, !, #, \$, %, ^, &, *, (,), :, ;, and @.</p> |

| Field | Type | Description |
|----------------------|-----------|--|
| targetChapSecret | String | Optional password for the chap authentication. Only required if 'mutual' chap authentication is chosen. This is only set on write or modify operations. For read operations, this string is null. The chap secret must be between 12 and 16 characters if specified, and cannot have special characters such as ' ', \, /, !, #, \$, %, ^, &, *, (,), :, ;, and @. |
| targetNetworkBinding | String[] | Network Binding (in ip:port format) string indicating the network ports to bind target with. |

ITView_V2_1

| Field | Type | Description |
|-----------------|---------|---|
| hostGroupName | String | Name of the host group |
| targetGroupName | String | Name of the target group |
| lunNbr | Integer | LUN number associated with the mapping. Populated with the default lun number -1 for project default mappings |
| readOnly | Boolean | Flag that indicates whether the view is read only. True for read only, false for read-write. |

LocalGroup_V1_2

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

LocalUser_V1_2


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

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. |
| 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. |

| Field | Type | Description |
|-------------------|---------|--|
| 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 | <p>A boolean that indicates if write protect is enabled or disabled.</p> <p>The values are as follows:</p> <ul style="list-style-type: none"> • Read-Only (True) • Write (False) |

NetworkACL_V2_1

| Field | Type | Description |
|------------------|--------|---|
| hostType | String | Type of network ACL host, IP address or FQDN. |
| host | String | Host to which access is provided. |
| accessMode | String | Access mode: 'ro' for read-only access or 'rw' for read-write access. |
| rootAccessForNFS | String | <p>Flag that indicates whether the NFS ACL has root access.</p> <p> Note: This is applicable only for NFS ACLs.</p> |

Notification_V2_3

| Field | Type | Description |
|--------------|--------|---|
| timestamp | Long | Milliseconds since Unix epoch (midnight Jan 1 1970). |
| eventCode | String | Event-unique code. |
| priority | String | Priority. Possible values are LOW, MEDIUM, HIGH, or CRITICAL. |
| description | String | Event description or subject. |
| errorMessage | String | Error message (if applicable). |

| Field | Type | Description |
|---------|------------------------|--|
| details | Map of key-value pairs | Key value mapping of event-specific details. |

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. |

PoolSpaceInfo_V2_1

| Field | Description |
|-------------------------------|--|
| totalPoolSize | Total size of the pool (bytes). |
| originalUsedByDataAndSnapshot | Original used by data and snapshot before compression and deduplication (bytes). |
| usedByDataAndSnapshot | Amount used by data and snapshot after compression and deduplication (bytes). |
| compressionSavingsPercentage | Amount saved from compression as a percentage. |
| dedupeSavingsPercentage | Amount saved from deduplication as a percentage. |
| totalSavingsPercentage | Amount saved from compression and deduplication as a percentage. |
| usedByAll | Amount used for data and reservation (bytes). |
| available | Amount of available space left on the pool (bytes). |
| usedByData | Amount used by data after compression and deduplication (bytes). |
| usedByReservation | Amount used by reservation (bytes). |
| usedBySnapshot | Amount used by snapshot after compression and deduplication (bytes). |
| totalMetaSize | Total metadata size (bytes). |
| usedMeta | Used metadata (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. |

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. |

Project_V2_1

| Field | Description |
|--------------|--|
| poolName | Name of the pool. This field is mandatory while creating a project. The field becomes read only after creation. |
| projectName | Name of the project. This field is mandatory while creating a project. The field becomes read only after creation. |
| localDataset | Indicates whether it is a local project or a replica from a remote system. True indicates local, while False indicates replica. This field is read only. |
| purpose | Purpose of the project. Project created by createProject API has a “generic” purpose. This field is read only. |

| Field | Description |
|----------------------|--|
| mountPoint | Mount point of the project on file system. This field is read only. |
| compression | Compression algorithm applied on the project. Supported values: <ul style="list-style-type: none"> • off • lzjb • gzip-2 • gzip • gzip-9 • lz4 (default) |
| compressedLog | Log compression algorithm for the project. Supported values: <ul style="list-style-type: none"> • lz4 • off (default) |
| intendedProtocolList | Adds protocol features to the project. This field is a list of string values, and is read only. The default value is "NFS, SMB, FC, iSCSI". The value can also be a combination of any of the following protocols: <ul style="list-style-type: none"> • FC • iSCSI • NFS • SMB <div>  Note: After you create a project, you cannot dynamically modify this property. </div> |
| quotaInByte | Enables quota on the project if intended for NFS use. The value should be greater than or equal to 1 MB (1048576 bytes). The default value is 0 (no limit). |
| quotaEnabled | Checks whether quota is enabled on the project. This field is read only. |
| quota | Quota number. This field is read only. |
| quotaMetric | Quota metric unit. This field is read only. |

| Field | Description |
|----------------|--|
| dedup | Turns on or off deduplication on the project. Supported values: <ul style="list-style-type: none"> • on (default) • off |
| copies | Specifies number of the project copies. Supported values: <ul style="list-style-type: none"> • 1 (default) • 2 • 3 |
| primaryCache | Specifies primary cache type. Supported values: <ul style="list-style-type: none"> • All (default) • None • Metadata |
| secondaryCache | Specifies secondary cache type. Supported values: <ul style="list-style-type: none"> • All (default) • None • Metadata |
| readonly | Specifies whether project is read only. Supported values: <ul style="list-style-type: none"> • On • Off |
| logbias | Specifies log bias type. Supported values: <ul style="list-style-type: none"> • Latency (default) • Throughput |
| aclInherit | Turns on or off ACL inheritance on the project. Supported values: <ul style="list-style-type: none"> • On (default) • Off |

| Field | Description |
|-------------------------|--|
| aclMode | Checks the ACL mode on the project. This field is read only. |
| krbStatus | Checks the krb status if SMB is enabled. This field is read only. |
| defaultVolumeSizeInByte | Default volume size. The value should be greater than or equal to 1 MB (1048576 bytes). The default value is 1 GB. |
| defaultVolumeSize | Volume size number. This field is read only. |
| defaultVolumeSizeUnit | Volume size unit. This field is read only. |
| defaultVolumeBlockSize | Sets default volume block size. Supported values: <ul style="list-style-type: none"> • 4 KB • 8 KB • 16 KB • 32 KB (default) • 64 KB • 128 KB |
| defaultThinProvisioning | Sets default thin provisioning for volume. The value is either True or False. |
| sync | Checks synchronization mode. This field is read only. |
| zfsDataSetName | Returns ZFS dataset path name. This field is read only. |
| recordSize | Default share block size. Supported values: <ul style="list-style-type: none"> • 4 KB • 8 KB • 16 KB • 32 KB (default) • 64 KB • 128 KB |

ProjectCloneProgressStatus_v1_2

| Field | Type | Description |
|-------------------|---------|--|
| failedSubProjects | Integer | Number of sub projects (shares and volumes) for which clone snapshot has failed. |
| projectCloneState | Integer | The integer return value is defined in CLONE_PROGRESS_STATUS . |
| totalSubProjects | Integer | Total number of sub projects for the given project. |

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 | The integer return value is defined in Replication_Scope_Option . |

ReplicationStatus_v1_2

| Field | Type | Description |
|-------------------|---------|---|
| completedTask | Integer | Number of tasks completed. |
| completeTimestamp | Date | Time stamp indicating when replication was completed. |

| Field | Type | Description |
|-----------------|---------|---|
| currentStatus | Integer | Current status of replication. The integer return value is defined in State |
| dataSent | Long | Total data sent. |
| sendSpeed | Long | Replication data send speed. |
| startTimestamp | Date | Time stamp indicating when replication started. |
| taskSize | Integer | Total task size. |
| updateTimestamp | Date | Time stamp indicating when replication was last updated. |

Schedule_V2_1

| Field | Type | Description | | | | | | | | | | | | |
|----------------------|---------|---|------------------------|-----------------------|---------|---------|--------|-------|-------|-------|--------|--------|---------|--------|
| scheduleId | Integer | Schedule ID. Read-Only field. | | | | | | | | | | | | |
| retentionPeriod | Integer | Maximum period to retain snapshots. | | | | | | | | | | | | |
| | | <table><tr><th>Schedule Interval Type</th><th>Retention Period Unit</th></tr><tr><td>Minutes</td><td>Days</td></tr><tr><td>Hourly</td><td>Days</td></tr><tr><td>Daily</td><td>Weeks</td></tr><tr><td>Weekly</td><td>Months</td></tr><tr><td>Monthly</td><td>Years</td></tr></table> | Schedule Interval Type | Retention Period Unit | Minutes | Days | Hourly | Days | Daily | Weeks | Weekly | Months | Monthly | Years |
| | | Schedule Interval Type | Retention Period Unit | | | | | | | | | | | |
| | | Minutes | Days | | | | | | | | | | | |
| | | Hourly | Days | | | | | | | | | | | |
| | | Daily | Weeks | | | | | | | | | | | |
| | | Weekly | Months | | | | | | | | | | | |
| | | Monthly | Years | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| scheduleIntervalType | String | Type of schedule interval. The possible values are minutes, hours, days, weeks, or months. | | | | | | | | | | | | |
| repeatInterval | Integer | How often the schedule should recur. | | | | | | | | | | | | |
| | | <table><tr><th>Schedule Interval Type</th><th>Retention Interval</th></tr><tr><td>Minutes</td><td>Minutes</td></tr><tr><td>Hourly</td><td>Hours</td></tr><tr><td>Daily</td><td>Days</td></tr><tr><td>Weekly</td><td>Weeks</td></tr><tr><td>Monthly</td><td>Months</td></tr></table> | Schedule Interval Type | Retention Interval | Minutes | Minutes | Hourly | Hours | Daily | Days | Weekly | Weeks | Monthly | Months |
| | | Schedule Interval Type | Retention Interval | | | | | | | | | | | |
| | | Minutes | Minutes | | | | | | | | | | | |
| | | Hourly | Hours | | | | | | | | | | | |
| | | Daily | Days | | | | | | | | | | | |
| | | Weekly | Weeks | | | | | | | | | | | |
| | | Monthly | Months | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |

| Field | Type | Description |
|------------|--------|--|
| startDate | String | Start date for scheduling a snapshot in future. Should be specified in YYYY-MM-DD format. Start date can be either current date or a date in future. |
| startTime | String | Start time for the schedule. This should be in 24HR HH:MM format. The date cannot be a past date. |
| endTime | String | End time for the schedule. This should be in 24HR HH:MM format. This is applicable only for Minutes and Hourly Schedule interval types. |
| daysOfWeek | String | Days of the Week. This is for the Weekly Schedule interval types. Should be specified in comma-separated numbers and each number specifying day of the week. For example: Sunday:1, Monday:2, Tuesday:3 and so on. For Weekdays Only: 2,3,4,5,6 For Weekend Only: 1,7 For Mon-Wednesday only: 2,4 |
| dayOfMonth | String | Day of the month. This is for the monthly schedule interval types and the value indicates the date of each month when the schedule should be run. If the specified date does not exist in the s month, the job is not run and is scheduled to next month, depending on the repeat interval specified. For example, not each month has 31 days, so the job is skipped for those months. Either this field or weekdayOfMonth can be specified, but not both. |

| Field | Type | Description |
|----------------|--------|--|
| weekdayOfMonth | String | <p>Weekday of the month. This should be in W:D format. This is for the monthly schedule interval types.</p> <p>Examples:</p> <p>Sunday:1, Monday:2, Tuesday:3 and so on.</p> <p>2nd Monday of the Month : 2:2</p> <p>4th Thursday of the Month : 4:5</p> <p>This field could also contain the value "LAST" to indicate that the job has to run on the last day of the month.</p> <p>Either this field or the dayOfWeek can be specified, but not both.</p> |

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 an IntelliFlash 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. |
| mountpoint | String | This string exposes the mountpoint of the share on an IntelliFlash 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. |

Share_V2_1

| Field | Description |
|-------------|---|
| name | Name of the dataset. This field is read only. |
| poolName | Name of the parent storage pool. This field is read only. |
| projectName | Name of the project. This field is read only. |

| Field | Description |
|------------------------|---|
| purpose | Purpose of the dataset. This field is read only. |
| guid | Global unique identifier of the dataset. This field is read only. |
| compression | <p>Compression algorithm that applies on the dataset.</p> <p>Supported values:</p> <ul style="list-style-type: none"> • off • lzjb • gzip-2 • gzip • gzip-9 • lz4 |
| overrideCompression | Checks whether compression algorithm overrides parent project. This field is read only. |
| localDataset | Indicates whether the dataset is local or replica. This field is read only. |
| reservationInByte | Enables reservation on the dataset. This value should be greater than or equal to 1 MB (1048576 bytes) or set to 0 (no limit). |
| reservationEnabled | Checks whether reservation is enabled on the dataset. This field is read only. |
| reservation | Reservation number. This field is read only. |
| reservationMetric | Reservation metric unit. This field is read only. |
| dedup | Turns on or off deduplication on the dataset. Supported values are on and off. |
| overrideDedup | Check whether deduplication overrides parent project. This field is read only. |
| primaryCache | Specifies primary cache type. Supported values are All, None, and Metadata. |
| overridePrimaryCache | Checks whether primary cache overrides parent project. This field is read only. |
| secondaryCache | Specifies secondary cache type. Supported values are All, None, and Metadata. |
| overrideSecondaryCache | Checks whether secondary cache overrides parent project. |
| readonly | Specifies whether the dataset is read only. Supported values are On and Off |

| Field | Description |
|----------------------------------|--|
| overrideReadOnly | Checks whether read only overrides parent project. This field is read only. |
| logbias | Specifies log bias type. Supported values are Latency and Throughput |
| overrideLogbias | Checks whether log bias overrides parent project. This field is read only. |
| sync | Checks the synchronization mode. This field is read only. |
| overrideSync | Checks whether sync mode overrides parent project. This field is read only. |
| overrideProjectSnapshot Settings | Checks whether snapshot settings override parent project. This field is read only. |
| zfsDataSetName | Returns ZFS dataset path name. This field is read only. |
| compressedLog | Log compression algorithm for the dataset. Supported values are lz4 and off. |
| overrideCompressedLog | Checks whether log compression algorithm overrides parent project. This field is read only. |
| containerName | Checks current dataset container name. This field is read only. |
| mountPoint | Checks current mount point. This field is read only. |
| overrideMountPoint | Check whether mount point overrides default. This field is read only. |
| quotaInByte | Enables quota on the share if intended for NFS use. This value should be greater than or equal to 1 MB (1048576 bytes) or set to 0 (no limit). |
| quotaEnabled | Checks whether quota is enabled on the share. This field is read only. |
| quota | Quota number. This field is read only. |
| quotaMetric | Quota metric unit. This field is read only. |
| availableSize | Checks current available share size. This field is read only. |
| totalSize | Checks current total share size. This field is read only. |
| overrideSharenfs | Checks whether NFS sharing overrides project setting. This field is read only. |
| overrideSharesmb | Checks whether SMB sharing overrides project setting. This field is read only. |

| Field | Description |
|--------------------|--|
| krbStatus | Checks current Kerberos status of the share if SMB is enabled. This field is read only. |
| cifsDisplayName | Checks CIFS display name. This field is read only. |
| guestStatus | Checks guest status of the share. This field is read only. |
| aclInherit | Turns on or off ACL inheritance on the project. Supported values are On and Off. |
| overrideAclInherit | Check whether aclInherit overrides parent project setting. This field is read only. |
| recordSize | <p>Default share block size.</p> <p>Supported values:</p> <ul style="list-style-type: none"> • 4 KB • 8 KB • 16 KB • 32 KB • 64 KB • 128 KB <p>This field can be modified.</p> |
| overrideRecordSize | Check whether share block size overrides parent project. This field is read only. |
| atime | Checks access time. This field can be modified. |
| nbmand | Checks non-blocking mandatory locks. This field can be modified. |
| aclList | A list of UserACL objects. This field checks current ACLs applied to the share. This field is read only. |

ShareOptions

| Field | Type | Description |
|------------|--------|---|
| blockSize | String | Block size of the share. Valid values are 4 KiB, 8 KiB, 16 KiB, 32 KiB, 64 KiB, or 128 KiB. |
| 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. |

| Field | Type | Description |
|-------------|------|--|
| 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 | 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 | User ACL permission type. Valid values are defined by the Permission_type_enum enumeration. |
| sharePermissionMode | Integer | User ACL mode. Valid values are defined by the Mode_enum enumeration. |
| userList | Array | 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. |

SMBConfig_V2_2

| Field | Type | Description |
|-------------------------|---------|--|
| pdc | String | Primary domain controller. |
| subsharesFeatureEnabled | Boolean | Determines whether subshare can be created. Set the field to true or false. |
| smbProtocolMode | String | The protocol mode, whether CIFS or SMB 3.0. |
| restrictAnonymous | Boolean | Determines whether to restrict anonymous SMB user. Set the field to true or false. |

| Field | Type | Description |
|---------------|---------|--|
| restrictGuest | Boolean | Determines whether to restrict SMB guest user. Set the field to true or false. |

SnapshotDeletionStatus

| Field | Type | Description |
|------------------------|---------|---|
| deletedList | List | Deleted snapshots list |
| failedToDeleteList | List | Not deleted snapshots list |
| snapshotDeletionStatus | Integer | The integer return value is defined in SNAPSHOT_DELETION_STATUS |

SnapshotProgressStatus

| Field | Type | Description |
|------------------------|---------|---|
| snapshotProgressStatus | Integer | An integer from the enumeration SNAPSHOT_PROGRESS_STATUS that indicates the snapshot progress status. |

SnapshotSchedule_V2_1

| Field | Type | Description |
|-------------|--------|---|
| datasetPath | String | Dataset path. It can be a project, LUN or a share. This should not be a replica dataset. |
| quiesce | String | Quiesce on or off. |

SNMP_Setting_V2_1

This class represents the SNMP service settings (community string and trap listeners).

| Field | Type | Description |
|-----------------|--------|---|
| communityString | String | SNMP Community String. This must be 1 to 64 alpha-numeric characters, or should be * _ , . (default='public') |
| trapListeners | String | List of Trap listeners strings in IP:Port_Number format. |

SystemAnalyticsResult_V2_3

| Field | Type | Description |
|---------------------|---------------------------------|---|
| systemAnalyticsType | String | System analytics type. Possible values are CPU, CACHE_HITS, POOL_PERFORMANCE, and NETWORK. |
| timestamps | Array of longs | Milliseconds since Unix epoch (midnight Jan 1 1970). |
| Datapoints | Map (String-->Array of numbers) | Datapoint list mapping. Each datapoint list has one element per timestamp. |
| Averages | Map (String-->Array of numbers) | Averages of the datapoint listings. |

TargetGroup_V2_2

| Field | Type | Description |
|------------------|--------|--|
| targetGroupName | String | Target group name as listed in the Web UI. |
| intendedProtocol | String | The protocol (iSCSI, FC, or Unknown) that the target group is configured to use. Unknown is listed when there are no targets mapped to the target group. |

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 an IntelliFlash array. A dataset path should have the format: PoolName/Local/ProjectName/VolumeName. You can get the datasetPath from the listVolumes API. For more information, see listVolumes . The datasetPath is not required for createVolume API. |
| local | Boolean | This boolean identifies whether the volume belongs to a local project or a replicated project. The local boolean is not required for createVolume API. |

| Field | Type | Description |
|---------------|---------|--|
| 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. |

Volume_V2_1

| Field | Description |
|---------------------|---|
| name | Name of the dataset. This field is read only. |
| poolName | Name of the parent storage pool. This field is read only. |
| projectName | Name of the project. This field is read only. |
| purpose | Purpose of the project. This field is read only. |
| guid | Global unique identifier of the dataset. This field is read only. |
| compression | Compression algorithm applied on the dataset. Supported values are off, lzjb, gzip-2, gzip, gzip-9, and lz4. |
| overrideCompression | Checks whether compression algorithm overrides parent project. This field is read only. |
| localDataset | Indicates whether the dataset is local or replica. This field is read only. |
| reservationInByte | Enables reservation on the dataset. The value should be greater than 1 MB (1048576 bytes) or set to 0 (no limit). |
| reservationEnabled | Checks whether reservation is enabled on the dataset. This field is read only. |
| reservation | Reservation number. This field is read only. |
| reservationMetric | Reservation metric unit. This field is read only. |
| dedup | Turns on or off deduplication on the dataset. Supported values are on and off. |

| Field | Description |
|---------------------------------|---|
| overrideDedup | Checks whether deduplication overrides parent project. This field is read only. |
| primaryCache | Specifies primary cache type. Supported values are All, None, and Metadata. |
| overridePrimaryCache | Checks whether primary cache overrides parent project. This field is read only. |
| secondaryCache | Specifies secondary cache type. Supported values are All, None, and Metadata. |
| overrideSecondaryCache | Checks whether secondary cache overrides parent project. This field is read only. |
| readonly | Specifies whether it is read only. Supported values are on and off. |
| overrideReadonly | Checks whether read only overrides parent project. This field is read only. |
| logbias | Specify log bias type. Supported values are Latency and Throughput. |
| overrideLogbias | Checks whether log bias overrides parent project. This field is read only. |
| sync | Checks synchronization mode. This field is read only. |
| overrideSync | Checks whether sync mode overrides parent project. This field is read only. |
| overrideProjectSnapshotSettings | Checks whether snapshot settings override parent project. This field is read only. |
| zfsDataSetName | Returns ZFS dataset path name. This field is read only. |
| compressedLog | Log compression algorithm for the dataset. Supported values are lz4 and off. |
| overrideCompressedLog | Checks whether log compression algorithm overrides parent project. This field is read only. |
| containerName | Checks current dataset container name. This field is read only. |
| volSize | Volume size in bytes. This field is read only. |
| luid | Unique identifier for the lun. This field is read only. |
| usedSize | Used Volume size in bytes. This field is read only. |
| thinProvisioning | Indicates whether this volume is thin provisioned. This field is read only. |

| Field | Description |
|----------------|--|
| blockSize | Sets volume block size. Supported values: <ul style="list-style-type: none"> • 4 KB • 8 KB • 16 KB • 32 KB • 64 KB • 128 KB |
| writeBackCache | Enables or disables disk write back cache. Supported values are Enable and Disable. |
| overrideViews | Checks whether initiator-target view overrides parent project. This field is read only. |
| Protocol | Checks current protocol for the volume. This field is read only. |

UserACL (Read Only) v2.1

| Field | Description |
|--------------------|---|
| id | Trivia |
| controllerId | Trivia |
| aclType | Group/User/Everyone |
| aclUser | If type is User, show user name |
| aclGroup | If type is Group, show group name |
| aclValDisplay | String of ACL value. For example, rwxpdDaARWcCos. |
| aclVal | Integer value of aclValDisplay. |
| aclMode | Allow or deny |
| aclInheritanceFlag | Inheritance type. Default/Files/Directories/Both |
| includeSubShares | True/False |

Chapter 14

Enumerations

Topics:

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

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

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. |

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](#).

COMMAND_STATUS

| Status | Returned Value | Description |
|-----------------------|----------------|---|
| COMMAND_SUCCEED | 0 | Indicates that command (request) succeeded. |
| COMMAND_NOT_ATTEMPTED | 1 | Indicates that command (request) not attempted. |

| Status | Returned Value | Description |
|----------------|----------------|--|
| COMMAND_FAILED | 2 | Indicates that command (request) failed. |

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](#).

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. |

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. |

| Value | Returned Value | Description |
|-------|----------------|---|
| GROUP | 2 | Indicates that the supplied ACL is for the specified group. |

Related APIs, Objects, and Enumerations

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

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](#).

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 deleteProjectSnapshot API. It indicates that only some of the snapshots could be deleted (other snapshots that were selected for deletion could not be deleted.) |
| FAILURE | 2 | Indicates that the snapshot deletion has failed. |

Related APIs, Objects, and Enumerations

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

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. |

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. |
| 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](#).

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 Services (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. |

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.

errorIfExists

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.

errorIfExists

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"  
}
```