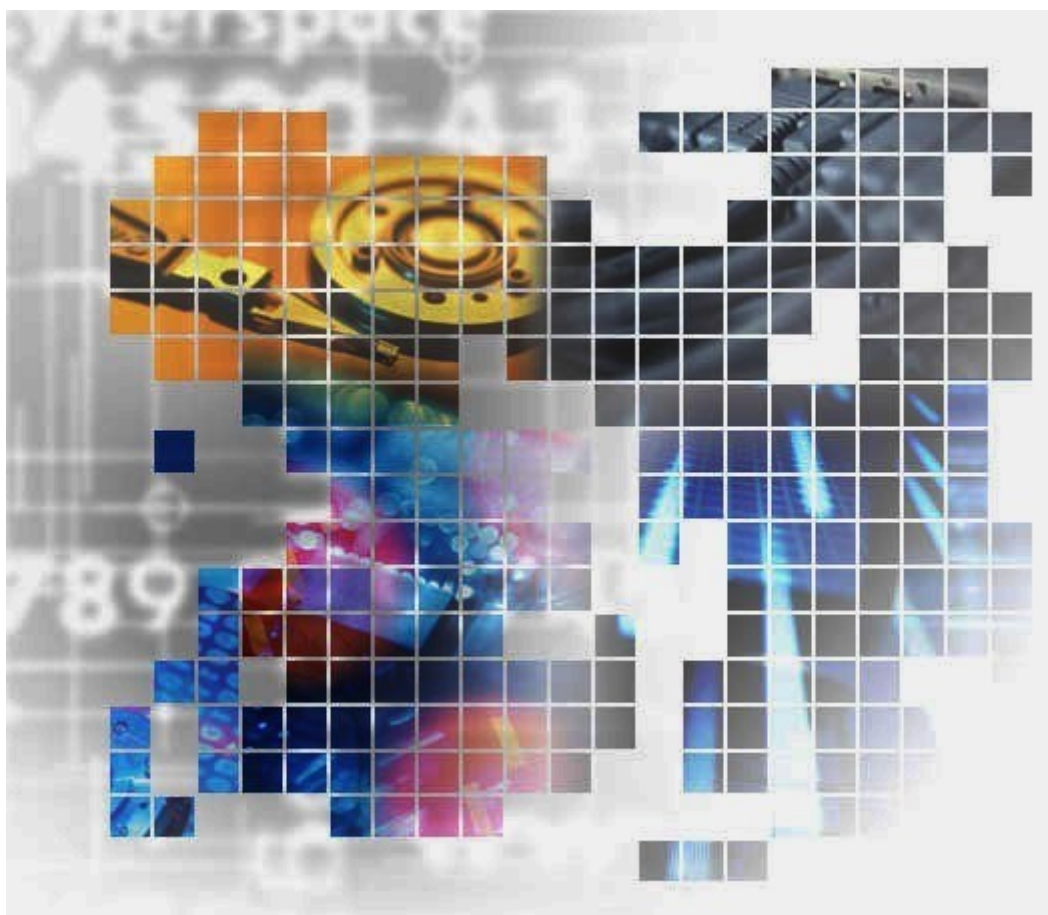


**iStorage V110/V310/V310F**

# **HA Storage Manager Embedded REST API リファレンスガイド**



---

## 著作権

©NEC Corporation 2024,2025

## 免責事項

このマニュアルの内容の一部または全部を無断で複製することはできません。

このマニュアルの内容については、将来予告なしに変更することがあります。

このマニュアルに基づいてソフトウェアを操作した結果、たとえ当該ソフトウェアがインストールされているお客様所有のコンピュータに何らかの障害が発生しても、当社は一切責任を負いかねますので、あらかじめご了承ください。

このマニュアルの当該ソフトウェアご購入後のサポートサービスに関する詳細は、弊社営業担当、お買い求めいただいた販売店または保守サービス会社にお問い合わせください。

## 商標類

Linux は、Linus Torvalds 氏の米国およびその他の国における登録商標です。

Windows は、マイクロソフト企業グループの商標です。

その他記載の会社名、製品名などは、それぞれの会社の商標もしくは登録商標です。

## 輸出時の注意

本製品を輸出される場合には、外国為替及び外国貿易法の規制並びに米国輸出管理規則など外国の輸出関連法規をご確認の上、必要な手続きをお取りください。

なお、不明な場合は、弊社担当営業にお問い合わせください。

## 発行

2025 年 1 月

---

# 目次

<b>第 1 章 REST API の概要 .....</b>	<b>1</b>
1.1 REST API のシステム構成 .....	1
1.2 SSL 通信を利用する場合の要件 .....	2
1.3 API の実行ユーザのロールとアクセス権限 .....	2
1.4 管理対象のリソースの指定 (URL) .....	3
1.5 オブジェクト ID の指定方法 .....	9
1.6 サポートする HTTP メソッド .....	9
1.7 ユーザ認証 .....	10
1.8 リクエストヘッダ .....	12
1.9 レスポンスヘッダ .....	13
1.10 HTTP ステータスコード .....	13
1.11 リクエストおよびレスポンスのフォーマット .....	15
1.12 クエリパラメータ .....	15
1.13 データ型 .....	16
1.14 出力形式 .....	17
1.15 データオブジェクト .....	17
1.16 commandStatus オブジェクト .....	19
1.17 エラーオブジェクト .....	21
<b>第 2 章 REST API で共通の操作 .....</b>	<b>24</b>
2.1 非同期処理の API のステータス情報を取得する .....	24
2.2 非同期処理の API のステータス情報を削除する .....	27
<b>第 3 章 ストレージシステム全体の操作 .....</b>	<b>29</b>
3.1 ストレージシステムの情報を取得する .....	29
3.2 スナップショットとプールの稼働状態の情報を取得する .....	35
<b>第 4 章 管理リソースの準備とボリューム割り当ての操作 .....</b>	<b>38</b>
4.1 ドライブの管理 .....	38
4.1.1 ドライブの情報を取得する .....	38
4.1.2 特定のドライブの情報を取得する .....	41
4.1.3 ドライブをスペアドライブに設定する .....	43
4.1.4 スペアドライブの設定を解除する .....	45

---

4.2	プールの管理 .....	47
4.2.1	プールの情報を取得する .....	50
4.2.2	特定のプールの情報を取得する .....	55
4.2.3	プールを作成する .....	58
4.2.4	プールの設定を変更する .....	62
4.2.5	プールを削除する .....	63
4.2.6	プールにドライブを追加する .....	65
4.3	ポートの管理 .....	68
4.3.1	ポートの情報を取得する .....	68
4.3.2	特定のポートの情報を取得する .....	75
4.3.3	ポートの設定を変更する .....	77
4.4	サーバの管理 .....	82
4.4.1	サーバの情報を取得する .....	83
4.4.2	特定のサーバの情報を取得する .....	86
4.4.3	サーバを登録する .....	89
4.4.4	サーバの設定を変更する .....	92
4.4.5	サーバを削除する .....	94
4.4.6	サーバにホストグループ (iSCSI ターゲット) を追加する .....	96
4.4.7	ホストグループ名 (iSCSI ターゲット名) をサーバのニックネームと同期する .....	99
4.4.8	HBA の WWN (iSCSI ネーム) の情報を取得する .....	100
4.4.9	特定のサーバの HBA の WWN (iSCSI ネーム) の情報を取得する .....	103
4.4.10	HBA の WWN (iSCSI ネーム) の情報を追加する .....	105
4.4.11	サーバから HBA の WWN (iSCSI ネーム) の情報を削除する .....	107
4.4.12	サーバのパス情報の一覧を取得する .....	109
4.4.13	サーバと特定のポート間のパス情報を取得する .....	111
4.4.14	サーバにパス情報を追加する .....	113
4.4.15	サーバからパス情報を削除する .....	115
4.4.16	iSCSI ターゲットの情報を取得する .....	117
4.4.17	特定の iSCSI ターゲットの情報を取得する .....	119
4.4.18	iSCSI ターゲットの設定を変更する .....	120
4.5	ボリュームの管理 .....	122
4.5.1	ボリュームの情報を取得する .....	124
4.5.2	特定のボリュームの情報を取得する .....	128
4.5.3	ボリュームを作成する .....	130
4.5.4	ボリュームの設定を変更する .....	133
4.5.5	ボリュームの容量を拡張する .....	136
4.5.6	ボリュームを削除する .....	137

---

---

4.5.7 特定のボリュームの QoS の設定情報を取得する .....	138
4.5.8 ボリュームの QoS の設定を変更する .....	142
4.5.9 ボリュームとサーバの接続情報を取得する .....	147
4.5.10 ボリュームと特定のサーバの接続情報を取得する .....	149
4.5.11 ボリュームとサーバを接続する .....	151
4.5.12 ボリュームと特定のサーバの接続を解除する .....	153
<b>第 5 章 スナップショットの操作.....</b>	<b>155</b>
5.1 スナップショットの流れ .....	155
5.2 スナップショットグループ名の一覧を取得する .....	156
5.3 特定のスナップショットグループの情報を取得する .....	158
5.4 スナップショットの情報を取得する .....	160
5.5 特定のスナップショットの情報を取得する .....	163
5.6 スナップショットを作成する .....	165
5.7 スナップショットをマッピングする .....	168
5.8 スナップショットをリストアする .....	170
5.9 スナップショットグループを削除する .....	171
5.10 スナップショットを削除する .....	173
<b>第 6 章 Active Mirror を利用するための環境構築.....</b>	<b>175</b>
6.1 Active Mirror を利用するための環境構築の流れ .....	175
6.2 外部ボリュームの情報を取得する .....	176
6.3 特定の外部ボリュームの情報を取得する .....	181
6.4 外部ボリュームを作成する .....	183
6.5 外部ボリュームのニックネームを変更する .....	186
6.6 外部ボリュームグループの情報を取得する .....	187
6.7 特定の外部ボリュームグループの情報を取得する .....	190
6.8 外部パスグループの情報を取得する .....	192
6.9 特定の外部パスグループの情報を取得する .....	195
6.10 Quorum ディスクの情報を取得する .....	196
6.11 特定の Quorum ディスクの情報を取得する .....	199
6.12 Quorum ディスクの情報を登録する .....	200
6.13 Quorum ディスクの情報を削除する .....	202
<b>付録 A. 制限事項および注意事項.....</b>	<b>205</b>

---

---

A.1 制限事項および注意事項 .....	205
<b>付録 B. このマニュアルの参考情報 .....</b>	<b>206</b>
B.1 このマニュアルで使用している略語 .....	206
B.2 KB（キロバイト）などの単位表記について .....	206
<b>索引 .....</b>	<b>207</b>

---

# はじめに

このマニュアルは、ストレージシステムの管理や運用をする HA Storage Manager Embedded の REST API の使い方について説明したものです。

REST API は、ストレージシステムの情報取得や構成変更を行うための、REST (Representational State Transfer) の原則に従った Web API を提供します。

## 対象ストレージシステム

このマニュアルでは、次に示すストレージシステムに対応する製品（プログラムプロダクト）を対象として記述しています。

- iStorage V110
- iStorage V310
- iStorage V310F

このマニュアルでは、これらのストレージシステムを単に「ストレージシステム」と称することがあります。

## マニュアルの参照と適合ファームウェアバージョン

このマニュアルは、次の DKCMAIN ファームウェアのバージョンに適合しています。

- 「A3-03-01-XX」

## 対象読者

このマニュアルは、次の方を対象読者として記述しています。

- ストレージシステムの運用に関する知識がある方
- REST API を利用したプログラムを作成するスキルがある方

## サポート

REST API の導入時および運用時のお問い合わせ先は、次のとおりです。

- PP サポートサービス

---

## マニュアルで使用する記号について

このマニュアルでは、製品を安全にご使用いただくための注意書きを、次のとおり記載しています。

### 注意

データの消失・破壊のおそれや、データの整合性がなくなるおそれがある場合などの注意を示します。

### 重要

重要情報や追加情報を示します。

### メモ

解説、補足説明、付加情報などを示します。

### ヒント

より効率的にストレージシステムを利用するのに役立つ情報を示します。

## 「容量削減機能が有効なボリューム」について

このマニュアルで「容量削減機能が有効なボリューム」と記載されている場合、特に断りのない限り、データ削減共有ボリュームおよび dedupe and compression により容量削減機能を有効に設定した仮想ボリュームのことを示します。



# 第1章

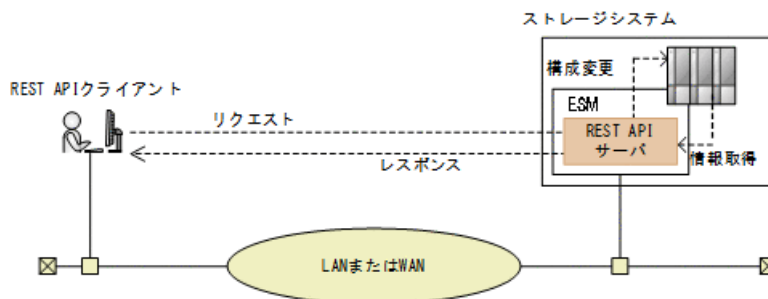
## REST API の概要

この章では、REST API を利用するための基本的なシステム構成や、管理対象のリソースの指定方法、API を実行する前に必要な情報、および API を実行したあとに参照する情報など、各 API で共通する情報について説明します。

### 1.1 REST API のシステム構成

REST API を使用するための基本的なシステム構成について説明します。

REST API の基本的なシステム構成を次に示します。



#### ストレージシステム

REST API での情報取得や構成変更の対象となるストレージシステムです。

#### ESM (Embedded Storage Manager)

ストレージシステムの基本的な管理機能を持つコンピュータです。外部からストレージシステムを管理する場合には、ESM と通信します。

コントローラ 1 (CTL01) とコントローラ 2 (CTL02) の各コントローラに存在しますが、REST API クライアントとは、これらを代表する IP アドレス (サービス IP) を使用して通信します。

#### REST API サーバ

REST API クライアントから REST API のリクエストを受け付け、ストレージシステムに命令を発行し、実行結果を REST API クライアントに返すサーバとしての役割を担うコンポーネントです。

ストレージシステムの ESM 上に配置されます。

#### REST API クライアント

REST API サーバへリクエストを発行するクライアントです。REST API を利用したソフトウェアまたはスクリプトが該当します。

---

## メモ

このマニュアルに記載された API のほかに、より詳細な情報を取得したり、構成変更する操作に対応した API も実行できます。操作方法については、マニュアル『REST API リファレンスガイド』を参照してください。

---

## 1.2 SSL 通信を利用する場合の要件

REST API で SSL 通信をする場合の要件について説明します。

REST API では、REST API クライアントと REST API サーバ間で SSL 通信が利用できます。

REST API クライアントと REST API サーバ間の SSL 通信には、ESM にインストールされている HTTPS 用のサーバ証明書が使用されます。デフォルトではこのサーバ証明書は自己署名証明書であるため、クライアントプログラムによっては、通信がエラーになる場合があります。この問題を解決するためには、次のどちらかの対応が必要です。

- ストレージシステムの証明書を VeriSign 社などの認証局に信頼されたサーバ証明書に変更する。

認証局の署名済み証明書の入手方法と ESM での証明書の更新方法については、マニュアル『システム管理者ガイド』を参照してください。

- クライアントプログラムでエラーを回避するように作成する。

クライアントプログラムでエラーを回避するための方法は、プログラム言語によって異なります。ここでは Python の例を説明します。

Python では、Requests ライブラリを使用している場合、リクエスト発行時に `verify=False` を指定することでサーバ証明書の検証処理をスキップできます。

---

## メモ

REST API クライアントと REST API サーバ間の SSL 通信に使用できる TLS バージョンと暗号方式 (Cipher Suite) については、マニュアル『システム管理者ガイド』を参照してください。

---

## 1.3 API の実行ユーザのロールとアクセス権限

REST API を使用してストレージシステムのリソースを操作するには、API を実行するユーザがすべてのリソースに対してアクセス権限を持っている必要があります。

また、操作対象のリソースに対して適切なロール (実行権限) が必要です。各 API を実行するために必要なロールについては、各 API の説明を参照してください。

REST API を使用する前に、必要なロールを持つユーザグループにユーザを登録してください。

## 1.4 管理対象のリソースの指定（URL）

REST API では、管理対象のリソースを URL の形式で指定します。

管理対象のリソースを特定するには、次の形式の URL を指定します。

**<プロトコル>://<ホスト名>:<ポート番号>/<アプリケーション>/<バージョン>/<ドメイン>**

- プロトコル

https または http を指定します。

ストレージシステムの設定で有効なプロトコルが指定できます。

セキュリティのため、https を使用することをお勧めします。

- ホスト名

ESM の IP アドレス（サービス IP）または名前解決のできるホスト名を指定します。

REST API クライアントは、ESM の IP アドレス（サービス IP）を使用してストレージシステムと通信します。サービス IP 以外の IP アドレスを使用している場合、IP アドレスの通信先がスタンバイ状態だとセッションが使用できなくなり、REST API が実行できません。この場合、出力されるメッセージに従って通信できる IP アドレスを指定するか、サービス IP を指定してください。

- ポート番号

接続に使用するポート番号を指定します。

デフォルトのポート番号は、443（SSL 通信用）および 80（非 SSL 通信用）です。デフォルトのポート番号で通信する場合は、ポート番号の指定を省略できます。

- アプリケーション

ConfigurationManager/simple を指定します。

### メモ

このドキュメントでは、「<プロトコル>://<ホスト名>:<ポート番号>/ConfigurationManager」をベース URL と表記します。

- バージョン

REST API のバージョンを指定します。現在指定できる値は v1 だけです。

- ドメイン

ドメインを指定します。現在指定できる値は objects だけです。

サポートする URL の形式を次に示します。

objects の配下には、操作対象のオブジェクト種別を指定します。

- command-status

- drives
- external-path-groups
- external-parity-groups
- external-volumes
- health-status
- pools
- ports
- quorum-disks
- servers
- snapshot-groups
- snapshots
- storage
- volumes
- volume-server-connections

各オブジェクトタイプでサポートする URL の形式の一覧を次に示します。

## オブジェクトタイプ `command-status` のリソース一覧

URL の形式（ベース URL を除く）	HTTP メソッド	処理方式	操作の内容
/simple/v1/objects/command-status/<オブジェクト ID>	GET	同期	非同期処理の API のステータス情報を取得する
	DELETE	同期	非同期処理の API のステータス情報を削除する

## オブジェクトタイプ `drives` のリソース一覧

URL の形式（ベース URL を除く）	HTTP メソッド	処理方式	操作の内容
/simple/v1/objects/drives	GET	同期	ドライブの情報を取得する
/simple/v1/objects/drives/actions/set-spare/invoke	POST	非同期	ドライブをスペアドライブに設定する
/simple/v1/objects/drives/actions/release-spare/invoke	POST	非同期	スペアドライブの設定を解除する
/simple/v1/objects/drives/<オブジェクト ID>	GET	同期	特定のドライブの情報を取得する

## オブジェクトタイプ external-path-groups のリソース一覧

URL の形式 (ベース URL を除く)	HTTP メソッド	処理方式	操作の内容
/simple/v1/objects/external-path-groups	GET	同期	外部パスグループの情報を取得する
/simple/v1/objects/external-path-groups/<オブジェクト ID>	GET	同期	特定の外部パスグループの情報を取得する

## オブジェクトタイプ external-parity-groups のリソース一覧

URL の形式 (ベース URL を除く)	HTTP メソッド	処理方式	操作の内容
/simple/v1/objects/external-parity-groups	GET	同期	外部ボリュームグループの情報を取得する
/simple/v1/objects/external-parity-groups/<オブジェクト ID>	GET	同期	特定の外部ボリュームグループの情報を取得する

## オブジェクトタイプ external-volumes のリソース一覧

URL の形式 (ベース URL を除く)	HTTP メソッド	処理方式	操作の内容
/simple/v1/objects/external-volumes	GET	同期	外部ボリュームの情報を取得する
	POST	非同期	外部ボリュームを作成する
/simple/v1/objects/external-volumes/<オブジェクト ID>	GET	同期	特定の外部ボリュームの情報を取得する
	PATCH	同期	外部ボリュームのニックネームを変更する

## オブジェクトタイプ health-status のリソース一覧

URL の形式 (ベース URL を除く)	HTTP メソッド	処理方式	操作の内容
/simple/v1/objects/health-status	GET	同期	ストレージリソースの稼働状態の情報を取得する

## オブジェクトタイプ pools のリソース一覧

URL の形式 (ベース URL を除く)	HTTP メソッド	処理方式	操作の内容
/simple/v1/objects/pools	GET	同期	プールの情報を取得する
	POST	非同期	プールを作成する

URL の形式（ベース URL を除く）	HTTP メソッド	処理方式	操作の内容
/simple/v1/objects/pools/<オブジェクト ID>	GET	同期	特定のプールの情報を取得する
	PATCH	同期	プールの設定を変更する
	DELETE	非同期	プールを削除する
/simple/v1/objects/pools/<オブジェクト ID>/actions/expand/invoke	POST	非同期	プールにドライブを追加する

## オブジェクトタイプ ports のリソース一覧

URL の形式（ベース URL を除く）	HTTP メソッド	処理方式	操作の内容
/simple/v1/objects/ports	GET	同期	ポートの情報を取得する
/simple/v1/objects/ports/<オブジェクト ID>	GET	同期	特定のポートの情報を取得する
	PATCH	同期	ポートの設定を変更する

## オブジェクトタイプ quorum-disks のリソース一覧

URL の形式（ベース URL を除く）	HTTP メソッド	処理方式	操作の内容
/simple/v1/objects/quorum-disks	GET	同期	Quorum ディスクの情報を取得する
	POST	非同期	Quorum ディスクの情報を登録する
/simple/v1/objects/quorum-disks/<オブジェクト ID>	GET	同期	特定の Quorum ディスクの情報を取得する
	DELETE	非同期	Quorum ディスクの情報を削除する

## オブジェクトタイプ servers のリソース一覧

URL の形式（ベース URL を除く）	HTTP メソッド	処理方式	操作の内容
/simple/v1/objects/servers	GET	同期	サーバの情報を取得する
	POST	非同期	サーバを登録する
/simple/v1/objects/servers/<オブジェクト ID>	GET	同期	特定のサーバの情報を取得する
	PATCH	非同期	サーバの設定を変更する
	DELETE	非同期	サーバの情報を削除する
/simple/v1/objects/servers/<オブジェクト ID>/actions/add-host-groups/invoke	POST	非同期	サーバにホストグループ (iSCSI ターゲット) を追加する
/simple/v1/objects/servers/<オブジェクト ID>/actions/sync-host-group-names/invoke	POST	非同期	ホストグループ名 (iSCSI ターゲット名) をサーバのニックネームと同期する

URL の形式（ベース URL を除く）	HTTP メソッド	処理方式	操作の内容
/simple/v1/objects/servers/<オブジェクト ID>/hbas	GET	同期	HBA の WWN (iSCSI ネーム) の情報を取得する
	POST	非同期	HBA の WWN (iSCSI ネーム) の情報を追加する
/simple/v1/objects/servers/<オブジェクト ID>/hbas/<オブジェクト ID>	GET	同期	特定のサーバの HBA の WWN (iSCSI ネーム) の情報を取得する
	DELETE	非同期	特定のサーバから HBA の WWN (iSCSI ネーム) の情報を削除する
/simple/v1/objects/servers/<オブジェクト ID>/paths	GET	同期	サーバのパス情報を取得する
	POST	非同期	サーバにパス情報を追加する
/simple/v1/objects/servers/<オブジェクト ID>/paths/<オブジェクト ID>	GET	同期	特定のサーバのパス情報を取得する
	DELETE	非同期	サーバからパス情報を削除する
/simple/v1/objects/servers/<オブジェクト ID>/target-iscsi-ports	GET	同期	iSCSI ターゲットの情報を取得する
/simple/v1/objects/servers/<オブジェクト ID>/target-iscsi-ports/<オブジェクト ID>	GET	同期	特定の iSCSI ターゲットの情報を取得する
	PATCH	非同期	iSCSI ターゲットの設定を変更する

## オブジェクトタイプ snapshot-groups のリソース一覧

URL の形式（ベース URL を除く）	HTTP メソッド	処理方式	操作の内容
/simple/v1/objects/snapshot-groups	GET	同期	スナップショットグループ名の一覧を取得する
/simple/v1/objects/snapshot-groups/<オブジェクト ID>	GET	同期	特定のスナップショットグループの情報を取得する
	DELETE	非同期	スナップショットグループを削除する

## オブジェクトタイプ snapshots のリソース一覧

URL の形式（ベース URL を除く）	HTTP メソッド	処理方式	操作の内容
/simple/v1/objects/snapshots	GET	同期	スナップショットの情報を取得する
	POST	非同期	スナップショットを作成する
/simple/v1/objects/snapshots/<オブジェクト ID>	GET	同期	特定のスナップショットの情報を取得する
	DELETE	非同期	スナップショットを削除する
/simple/v1/objects/snapshots/<オブジェクト ID>/actions/map/invoke	POST	非同期	スナップショットをマッピングする

URL の形式（ベース URL を除く）	HTTP メソッド	処理方式	操作の内容
/simple/v1/objects/snapshots/ ＜オブジェクト ID＞/actions/rest ore/invoke	POST	非同期	スナップショットをリストアする

## オブジェクトタイプ storage のリソース一覧

URL の形式（ベース URL を除く）	HTTP メソッド	処理方式	操作の内容
/simple/v1/objects/storage	GET	同期	ストレージシステムの情報を取得する

## オブジェクトタイプ volumes のリソース一覧

URL の形式（ベース URL を除く）	HTTP メソッド	処理方式	操作の内容
/simple/v1/objects/volumes	GET	同期	ボリュームの情報を取得する
	POST	非同期	ボリュームを作成する
/simple/v1/objects/volumes/＜ オブジェクト ID＞	GET	同期	特定のボリュームの情報を取得する
	PATCH	・ 同期 ・ 非同期	ボリュームの設定を変更する
	DELETE	非同期	ボリュームを削除する
/simple/v1/objects/volumes/＜ オブジェクト ID＞/actions/expand /invoke	POST	非同期	ボリュームの容量を拡張する
/simple/v1/objects/volumes/＜ オブジェクト ID＞/qos-setting	GET	同期	特定のボリュームの QoS の設定を取得する
	PATCH	非同期	ボリュームの QoS の設定を編集する

## オブジェクトタイプ volume-server-connections のリソース一覧

URL の形式（ベース URL を除く）	HTTP メソッド	処理方式	操作の内容
/simple/v1/objects/volume-ser ver-connections	GET	同期	ボリュームとサーバの接続情報を取得する
	POST	非同期	ボリュームとサーバを接続する
/simple/v1/objects/volume-ser ver-connections/＜オブジェクト ID＞	GET	同期	ボリュームと特定のサーバの接続情報を取得する
	DELETE	非同期	ボリュームと特定のサーバの接続を解除する

REST API の処理方式には、次に示す 2 つの処理方式があります。



- 同期処理

処理の実行結果がレスポンスとして返ります。

- 非同期処理

処理を受け付けたことを表す HTTP ステータスコード (202) とともにオブジェクトの情報がレスポンスとして返ります。

## メモ

複数のリクエストを同時に実行できるのは 16 個までです。

## 関連リンク

[サポートする HTTP メソッド \(9 ページ\)](#)

[HTTP ステータスコード \(13 ページ\)](#)

## 1.5 オブジェクト ID の指定方法

オブジェクト ID は、リソースを一意に識別するための ID です。

URL で特定のリソースを指定する場合に使用します。オブジェクト ID を指定するには、GET 操作を実行して、実行結果からオブジェクト ID を取得します。

**例：ボリュームのオブジェクト ID が「100」の場合**

```
volumes/100
```

## メモ

GET 操作でオブジェクト ID を取得した属性値は、RFC3986 に従って REST API サーバがエンコード済みの値を返却する場合があります。GET 操作で取得したオブジェクト ID を別の操作のリクエストに使用する場合は、オブジェクト ID をデコードしないでそのまま使用してください。

## 1.6 サポートする HTTP メソッド

リソースに対する操作を HTTP メソッドで指定します。

REST API では、次に示す HTTP メソッドをサポートしています。

HTTP メソッド	説明
GET	オブジェクトの情報を取得する。またはオブジェクトのリストを取得する。 例：プールの一覧を取得する。 複数のオブジェクトの情報を取得する場合、取得される情報はオブジェクト ID（例えば、id（プール ID やボリューム ID など））でソートされていません。クエリパラメータを指定して取得する情報をフィルタリングするか、取得するオブジェクト ID が特定できている場合は、特定のオブジェクト情報を取得する API を実行して取得してください。

HTTP メソッド	説明
POST	オブジェクトを新規に作成、追加、拡張などの操作をする。 例：プールを作成する。
PATCH	オブジェクトの属性や状態を部分的に変更する。 例：プールのしきい値を変更する。
DELETE	オブジェクトを削除する。 例：プールを削除する。

オブジェクトによって、使用できるメソッドが異なります。詳細は、各 API の説明を参照してください。

### —— 関連リンク ——

[管理対象のリソースの指定 \(URL\) \(3 ページ\)](#)

## 1.7 ユーザ認証

ストレージシステムに対する操作を実行する場合、ユーザ認証が必要です。ユーザ認証を行うには、Authorization ヘッダを指定します。

### ユーザ ID とパスワードによる認証

ユーザ ID とパスワードによる認証は、セッション生成時にのみ使用します。

Authorization ヘッダに、次の形式で認証情報を指定します。

```
Authorization: Basic <認証情報>
```

#### 認証情報

ユーザ ID とパスワードをコロン (:) でつないだ文字列を Base64 でエンコードした文字列を指定します。ストレージシステムのリソースを操作できるユーザアカウントのユーザ ID とパスワードを使用してください。

REST API では、ユーザ ID とパスワードに次の文字を使用できます。

項目	文字数	使用できる文字
ユーザ ID	1～63 文字	<ul style="list-style-type: none"> <li>半角英数字</li> <li>次の半角記号 ! # \$ % &amp; ' * + - . / = ? @ ^ _ ` {   } ~</li> </ul>
パスワード	6～63 文字	<ul style="list-style-type: none"> <li>半角英数字</li> <li>スペースを除くキー入力可能な ASCII 記号 ! " # \$ % &amp; ' ( ) * + , - . / : ; &lt; = &gt; ? @ [ \ ] ^ _ ` {   } ~</li> </ul>

## メモ

使用できる文字数、および文字列はパスワードポリシーによって異なります。詳細は『システム管理者ガイド』のパスワードポリシーに関する説明を参照してください。

ユーザ ID が sample-user、パスワードが sample-password の場合の **Authorization** ヘッダの例を次に示します。

```
Authorization: Basic c2FtcGx1LXVzZXI6c2FtcGx1LXBhc3N3b3Jk
```

## セッションによる認証

**Authorization** ヘッダに、次の形式でセッションのトークンを指定します。

```
Authorization: Session <トークン>
```

### トークン

トークンは、セッションを生成すると返却される認証情報です。この情報を基に、リクエストが認証済みユーザから発行されたかどうかを判定します。

**Authorization** ヘッダの指定例：

```
Authorization : Session b74777a3f9f04ea8bd8f09847fac48d3
```

## セッションの生成と使用

REST API では、セッションベースのユーザ認証を行います。REST API クライアントが REST API サーバにアクセスして操作を開始する際には、必ず最初にセッションを生成します。セッション生成のリクエストでは、ストレージシステムにアクセスするためのユーザ ID とパスワードによる認証を行います。セッション生成後は、セッションの情報を **Authorization** ヘッダに指定し、セッションの情報に基づいて認証を行います。セッションを操作する API やセッションの管理方法については、マニュアル『REST API リファレンスガイド』を参照してください。

リクエストライン：

```
POST <ベース URL>/v1/objects/sessions
```

レスポンスメッセージ：

```
{
  "token": "b74777a3f9f04ea8bd8f09847fac48d3",
  "sessionId": 3
}
```

---

**関連リンク**

---

[リクエストヘッダ \(12 ページ\)](#)

---

## 1.8 リクエストヘッダ

リクエストヘッダは、REST API クライアントから REST API サーバへの要求メッセージです。

REST API でサポートするリクエストヘッダについて説明します。

ヘッダ	指定区分	説明
Accept	任意	レスポンスのメディアタイプを指定するヘッダです。 指定できる値：*/* (json) デフォルト値：*/* (json)
Accept-Language	任意	クライアントが受け付けるメッセージの言語を指定するヘッダです。 日本語の言語を設定する場合は ja を指定します。日本語以外の言語を指定した場合、またはこのヘッダを指定しないときの言語は、英語 (en) となります。
Content-Type	任意	リクエストボディのメディアタイプを指定するヘッダです。 指定できる値：application/json デフォルト値：application/json
Content-Length	任意	リクエストボディのサイズを指定するヘッダです。 リクエストボディを指定する場合に Content-Length ヘッダを指定できます。クライアントソフトウェアの仕様によっては、自動的に付与されます。 指定できる値：バイト単位で指定します。 デフォルト値：なし
Authorization	必須	認証情報を指定するヘッダです。 次のどちらかの形式で指定します。 <ul style="list-style-type: none"><li>セッション生成時 Basic &lt;認証情報&gt; 認証情報はユーザ ID とパスワードを Base64 でエンコードした文字列を指定してください。ストレージシステムのリソースを操作できるユーザアカウントのユーザ ID とパスワードを使用してください。</li><li>セッション生成時以外 Session &lt;トークン&gt; セッション生成時に取得したトークンを指定してください。</li></ul>

上記以外のヘッダが指定された場合、そのヘッダは無視されます。

---

**関連リンク**

---

[ユーザ認証 \(10 ページ\)](#)

---

## 1.9 レスポンスヘッダ

レスポンスヘッダは、REST API サーバから REST API クライアントへの応答メッセージです。

REST API サーバが返すレスポンスヘッダについて説明します。

ヘッダ	説明
Content-Type	レスポンスデータのメディアタイプを示します。 デフォルト：application/json
Content-Length	レスポンスデータのサイズを示します。 レスポンスデータのサイズが大きい場合、このヘッダは返却されずに、データを分割して転送することを示す「Transfer-Encoding: chunked」が返却されます。
Transfer-Encoding	レスポンスデータの転送時のエンコード形式を示します。 サイズが大きいレスポンスデータを分割して転送する場合に、「chunked」が返却されます。
Retry-After	リクエストを再実行するまでの推奨待機時間（秒）を示します。 HTTP ステータスコード 503 が返却される場合に、このヘッダが設定されることがあります。
WWW-Authenticate	HTTP ステータスコード 401 が返却される場合に、認証が必要であることを示します。 <b>デフォルト：</b> <ul style="list-style-type: none"> <li>ユーザ ID とパスワードによる認証 Basic realm="Block storage"</li> <li>セッションによる認証 Session realm="Block storage"</li> </ul>

## 1.10 HTTP ステータスコード

REST API は、処理結果を示すために次に示す標準的な HTTP のステータスコードを使用します。

HTTP ステータスコード	説明
200	OK リクエストが正しく処理されました。
202	Accepted 非同期処理のリクエストの受け付けが完了したことを示します。
400	Bad Request リクエストヘッダ、クエリパラメータ、オブジェクト ID、またはリクエストボディが不正であることを示します。
401	Unauthorized リクエストヘッダに Authorization ヘッダが指定されていない、または Authorization ヘッダに指定された情報での認証に失敗したことを示します。
403	Forbidden 操作を実行するために必要な権限がないことを示します。

HTTP ステータスコード	説明
404	<p>Not Found</p> <p>URL で指定したリソースが見つからない、または指定した URL にリソースがないことを示します。</p>
406	<p>Not acceptable</p> <p>Accept ヘッダに、サポートしていないメディアタイプが指定されたことを示します。</p>
415	<p>Unsupported Media Type</p> <p>指定されたメディアタイプはサポートされていないことを示します。</p>
500	<p>Internal Server Error</p> <p>サーバで内部エラーが発生したことを示します。最も重要なエラーが代表として返却されます。</p> <p>複数のリソースを操作する API を実行した場合、一部だけ情報が取得できている、または操作が一部だけ完了しているときは、次に示す属性が出力されます。</p> <p>partialSuccessResponses</p> <p>出力される内容は操作した HTTP メソッドの種類によって異なります。</p> <ul style="list-style-type: none"> <li>• GET メソッドを操作した場合 取得できている情報（データオブジェクト）が出力されます。</li> <li>• POST メソッドを操作した場合 実行結果（statusResource）が出力されます。</li> </ul>
503	<p>Service Unavailable</p> <p>サービスが、ビジュー状態またはメンテナンスなどで一時的に使用できないことを示します。最も重要なエラーが代表として返却されます。</p> <p>レスポンスヘッダに「Retry-After」が返却された場合、返却された値（推奨待機時間（秒））を目安に再度リクエストを実行してください。</p> <p>複数のリソースを操作する API を実行した場合、一部だけ情報が取得できている、または操作が一部だけ完了しているときは、次に示す属性が出力されます。</p> <p>partialSuccessResponses</p> <p>出力される内容は操作した HTTP メソッドの種類によって異なります。</p> <ul style="list-style-type: none"> <li>• GET メソッドを操作した場合 取得できている情報（データオブジェクト）が出力されます。</li> <li>• POST または PATCH メソッドを操作した場合 実行結果（statusResource）が出力されます。</li> </ul>
504	<p>Gateway Timeout</p> <p>装置から時間内に応答がないことを示します。最も重要なエラーが代表として返却されます。</p> <p>複数のリソースを操作する API を実行した場合、一部だけ情報が取得できている、または操作が一部だけ完了しているときは、次に示す属性が出力されます。</p> <p>partialSuccessResponses</p> <p>出力される内容は操作した HTTP メソッドの種類によって異なります。</p> <ul style="list-style-type: none"> <li>• GET メソッドを操作した場合 取得できている情報（データオブジェクト）が出力されます。</li> <li>• POST または PATCH メソッドを操作した場合 実行結果（statusResource）が出力されます。</li> </ul>

---

**—— 関連リンク ——**

---

[データオブジェクト \(17 ページ\)](#)[エラーオブジェクト \(21 ページ\)](#)

---

## 1.11 リクエストおよびレスポンスのフォーマット

リソースの作成や追加 (POST)、変更時の属性値の指定 (PATCH)、またはリソースの情報取得 (GET) の結果には、JSON のフォーマットを使用します。

サポートする文字コードは UTF-8 です。

### リクエストの形式

- string 型の属性に空文字を指定した場合は、その属性の値は空になります。
- string 型以外の属性に空文字を指定した場合は、その属性は指定していないものと見なされます。

### レスポンスの形式

- API の処理が成功した場合、レスポンスは JSON 形式で返ります。
- 処理が失敗した場合、エラーの内容によっては JSON 形式ではなく HTML 形式でレスポンスが返る場合があります。

プログラム中で HTTP ステータスコードを基にエラー処理を行う場合には、レスポンスヘッダの Content-Type の値をチェックしてください。

---

### メモ

本マニュアルの JSON 形式の例では便宜的に改行を入れて記載していますが、実際のレスポンスでは改行が入らないことがあります。

---

## 1.12 クエリパラメータ

GET メソッドでオブジェクトを取得する際に、クエリパラメータを指定することで特定の条件で実行結果をフィルタリングできます。

クエリパラメータは、URL の末尾に次の形式で指定します。

```
?<パラメータ>=<値>
```

複数のパラメータを指定する場合、&記号でつないで指定します。複数のパラメータを指定する場合の例を次に示します。

?<パラメータ>=<値>&<パラメータ>=<値>...

パラメータは大文字と小文字が区別されます。各 API で指定できるパラメータ以外を指定した場合、無効なパラメータは指定されなかったものとみなし、有効なパラメータだけで実行結果がフィルタリングされます。

## 1.13 データ型

REST API で指定できるデータの型について説明します。

REST API がサポートするデータ型と対応する JSON のデータ型を次に示します。

データ型	JSON のデータ型	説明
boolean	boolean	true または false を表す型。 例: true
int	number	32 ビットの符号付き整数を表す型。 例: 100
long	number	64 ビットの符号付き整数を表す型。 例: 1048576
string	string	任意の文字列を表す型。 例: "SNB5B-R1R9NC"
ISO8601string	string	ISO 8601 拡張形式 (YYYY-MM-DDThh:mm:ssZ) で時刻を表す型。 指定できるタイムゾーンは UTC だけです。 例: "2017-09-30T09:27:35Z"

上記のデータ型以外に、JSON 形式の次のデータ型を使用します。

### object 型

属性と値をコロン (:) でつないだ文字列を {} で囲む形式です。属性と値のペアが複数ある場合は、コンマで区切ります。

### array 型

複数の値をコンマで区切った文字列を [] で囲む形式です。

---

#### 関連リンク

[管理対象のリソースの指定 \(URL\) \(3 ページ\)](#)

[HTTP ステータスコード \(13 ページ\)](#)

[データオブジェクト \(17 ページ\)](#)

---



## 1.14 出力形式

API を発行すると、API の処理方式、API の処理種別、実行結果に応じてレスポンスが返ります。

リクエストの処理が成功した場合のレスポンスの出力形式について次に示します。

API の処理方式	API の処理種別	実行結果のステータスコード	出力形式
同期処理	GET (単一のオブジェクトの取得)	200	各 API のレスポンスメッセージの説明を参照
	GET (複数のオブジェクトの取得)	200	データオブジェクト
	上記以外	200	各 API のレスポンスメッセージの説明を参照
非同期処理	すべて	202	commandStatus オブジェクト

リクエストの処理が失敗した場合は、レスポンスとしてエラーオブジェクトが返ります。

### —— 関連リンク ——

[管理対象のリソースの指定 \(URL\) \(3 ページ\)](#)

[HTTP ステータスコード \(13 ページ\)](#)

## 1.15 データオブジェクト

データオブジェクトは、オブジェクトのリストを返すためのオブジェクトです。

データオブジェクトのスキーマを次に示します。

属性	データ型	説明
data	array	オブジェクトのリスト

データオブジェクトと合わせて次の属性が出力されます。

属性	データ型	説明
count	int	オブジェクトの情報の格納数
totalCount	int	ストレージシステム内のデータ総数 この属性は、次に示す API を実行した場合に出力されます。 <ul style="list-style-type: none"> <li>• ボリュームの情報を取得する</li> <li>• ボリュームとサーバの接続情報を取得する</li> <li>• 外部ボリュームの情報を取得する</li> <li>• 外部ボリュームグループの情報を取得する</li> <li>• 外部パスグループの情報を取得する</li> <li>• スナップショットの情報を取得する</li> </ul>
hasNext	boolean	取得できていない情報があるかどうか出力されます。 <ul style="list-style-type: none"> <li>• true : 取得できていない情報がある</li> </ul>

属性	データ型	説明
		<ul style="list-style-type: none"> <li>• <code>false</code> : すべての情報が取得できている</li> </ul> <p>取得できていない情報がある場合、クエリパラメータを指定して取得する情報をフィルタリングするか、取得できている情報のうち、最大の ID（例えば、ボリューム ID やスナップショットの ID など）以降を複数回に分けて API を実行し取得してください。</p> <p>この属性は、次に示す API を実行した場合に出力されます。</p> <ul style="list-style-type: none"> <li>• ボリュームの情報を取得する</li> <li>• ボリュームとサーバの接続情報を取得する</li> <li>• 外部ボリュームの情報を取得する</li> <li>• 外部ボリュームグループの情報を取得する</li> <li>• 外部パスグループの情報を取得する</li> <li>• スナップショットの情報を取得する</li> </ul>

data オブジェクトの例を次に示します。

```
{
  "data": [
    {
      "id": 100,
      "nickname": "JH-26216_DP",
      "poolId": 63,
      "poolName": "NASOS",
      "totalCapacity": 1024,
      "freeCapacity": 982,
      "numberOfConnectingServers": 2,
      "numberOfSnapshots": 2
    },
    {
      "id": 101,
      "nickname": "JH-26216_DP",
      "poolId": 63,
      "poolName": "NASOS",
      "totalCapacity": 1024,
      "freeCapacity": 1024,
      "numberOfConnectingServers": 2,
      "numberOfSnapshots": 2
    }
  ],
  "count": 2,
  "totalCount": 2,
  "hasNext": false
}
```

---

#### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)  
[出力形式 \(17 ページ\)](#)

---

## 1.16 commandStatus オブジェクト

commandStatus オブジェクトは、非同期処理の API を発行したときに返却される API のステータス情報のオブジェクトです。

commandStatus オブジェクトのスキーマを次に示します。

属性	データ型	説明
id	int	非同期処理の API 実行時の問い合わせ用 ID
progress	string	API の進捗状況 次の値が返ります。 <ul style="list-style-type: none"> <li>unexecuted : API が未実行であることを示す</li> <li>processing : API が実行中であることを示す</li> <li>completed : API が実行完了したことを示す</li> </ul>
status	string	API の実行結果の状態 次の値が返ります。 <ul style="list-style-type: none"> <li>normal : API が成功した状態を示す</li> <li>error : API が失敗した状態を示す</li> </ul> この属性は、API の実行が完了した場合に出力されます。
affectedResources	string[]	操作対象のリソースにアクセスするための URL の一覧 1 つの API で複数のリソースを操作する場合は、すべての操作対象のリソースの URL が返ります。API が途中で失敗した場合は、処理が完了したことを確認できたリソースの URL だけが返ります。 リソースの削除操作が成功した場合、削除対象のリソースの URL が返ります。この URL にアクセスすると 404 エラーとなり、正常に削除されたことが確認できます。 この属性は、API の実行が完了した場合に出力されます。
errorResource	string	エラーが発生した URL この属性は、API の実行が失敗した場合に出力されます。
errorCode	object	ストレージシステムのエラーコード ストレージシステムでエラーが発生して、次に示すエラーコードがある場合にだけ値が返ります。 <ul style="list-style-type: none"> <li>SSB1 コード</li> <li>SSB2 コード</li> </ul> ストレージシステムのエラーコードは、ストレージシステムの保守に必要になります。 この属性は、API の実行が失敗した場合に出力されます。
errorMessage	string	エラーメッセージの内容 この属性は、API の実行が失敗した場合に出力されます。
operationDetails	object[]	操作対象のリソースの詳細情報 リソース種別に示すリソースに対して操作している場合に出力されます。 <ul style="list-style-type: none"> <li>operationType (string) リソースに対する操作種別 <ul style="list-style-type: none"> <li>CREATE : リソースが作成されたことを示す</li> <li>UPDATE : リソースが変更されたことを示す</li> <li>DELETE : リソースが削除されたことを示す</li> </ul> </li> </ul>

属性	データ型	説明
		<ul style="list-style-type: none"> <li>resourceType (string) 作成、変更、または削除が発生したリソースのリソース種別 <ul style="list-style-type: none"> <li>CommandStatus: 非同期処理の API のステータス情報に関するリソースであることを示す</li> <li>Pool: プールに関するリソースであることを示す</li> <li>Port: ポートに関するリソースであることを示す</li> <li>Server: サーバに関するリソースであることを示す</li> <li>Snapshot: スナップショットに関するリソースであることを示す</li> <li>Volume: ボリュームに関するリソースであることを示す</li> <li>VolumeServerConnection: ボリュームとサーバの接続に関するリソースであることを示す</li> <li>ExternalVolume: 外部ボリュームに関するリソースであることを示す</li> <li>QuorumDisk: Quorum ディスクに関するリソースであることを示す</li> </ul> </li> <li>resourceId (string) 作成、変更、または削除が発生したリソースのリソース ID</li> </ul>

## メモ

- オブジェクトの情報は最大で 65280 件保持されます。65280 件を超えた API のステータス情報は、非同期処理の API のステータス情報を削除する API を実行してオブジェクトの情報を削除してください。
- セッションがログアウトされた場合もオブジェクトの情報は削除されます。

API の実行を開始した場合の commandStatus オブジェクト例：

```
{
  "id": 1,
  "progress": "processing"
}
```

API が成功して完了した場合の commandStatus オブジェクトの例：

```
{
  "id": 2,
  "progress": "completed",
  "status": "normal",
  "affectedResources": [
    "/ConfigurationManager/simple/v1/objects/volumes/100"
  ],
  "operationDetails": [
    {
      "operationType": "CREATE",
      "resourceType": "Volume",
      "resourceId": "100"
    }
  ]
}
```

API が失敗して完了した場合の `commandStatus` オブジェクトの例：

```
{
  "id": 3,
  "progress": "completed",
  "status": "error",
  "errorResource": "/ConfigurationManager/simple/v1/objects/pools/63",
  "errorCode": {
    "SSB1": "2e11",
    "SSB2": "001f"
  },
  "errorMessage": "Snapshot or volume exists. Operation could not be completed.",
  "operationDetails": [
    {
      "operationType": "DELETE",
      "resourceType": "Pool",
      "resourceId": "63"
    }
  ]
}
```

---

#### 関連リンク

[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

[非同期処理の API のステータス情報を削除する \(27 ページ\)](#)

---

## 1.17 エラーオブジェクト

リクエストに対してエラーが発生した場合、HTTP ステータスコードとともに、エラーオブジェクトが返却されます。

エラーオブジェクトのスキーマを次の表に示します。

属性	データ型	説明
<code>errorSource</code>	string	エラーが発生した URL
<code>messageId</code>	string	メッセージ ID
<code>message</code>	string	エラーメッセージの内容
<code>cause</code>	string	エラーの要因
<code>solution</code>	string	エラーの対処
<code>errorCode</code>	string	ストレージシステムのエラーコード ストレージシステムでエラーが発生して、次に示すエラーコードがある場合にだけ値が返ります。 <ul style="list-style-type: none"> <li>• SSB1 コード</li> <li>• SSB2 コード</li> </ul> ストレージシステムのエラーコードは、ストレージシステムの保守に必要になります。

ストレージシステムでエラーが発生した場合のエラーオブジェクトの例：

```
{
  "errorSource": "/ConfigurationManager/simple/v1/objects/volumes/100/actions/expand/invoke",
  "messageId": "KART70000-E",
  "message": "The specified volume is being used by another program product
, or format in progress. Operation could not be completed. Wait for a while
and try again. Confirm the volume is not in use by other program product."
,
  "errorCode": {
    "SSB1": "B96B",
    "SSB2": "AF2E"
  }
}
```

エラーが発生しているが、一部だけ情報が取得できている、または操作が一部だけ完了している場合のエラーオブジェクトのスキーマを次の表に示します。

属性	データ型	説明
errorResponses	object[]	<p>エラーオブジェクトの詳細情報</p> <ul style="list-style-type: none"> <li>• errorSource (string) エラーが発生した URL</li> <li>• messageId (string) メッセージ ID</li> <li>• message (string) エラーメッセージの内容</li> <li>• cause (string) エラーの要因</li> <li>• solution (string) エラーの対処</li> <li>• errorCode (object) ストレージシステムのエラーコード ストレージシステムでエラーが発生して、次に示すエラーコードがある場合にだけ値が返ります。 <ul style="list-style-type: none"> <li>- SSB1 コード</li> <li>- SSB2 コード</li> </ul> ストレージシステムのエラーコードは、ストレージシステムの保守に必要になります。</li> </ul>
partialSuccessResponses	object[]	<p>エラーが発生しているが、一部だけ情報が取得できている、または操作が一部だけ完了している場合に値が返ります。</p> <p>すべて失敗している場合、空の配列が返ることがあります。</p>

ストレージシステムでエラーが発生したが、操作が一部だけ完了している場合のエラーオブジェクトの例：

```
{
  "errorResponses": [
    {
      "errorSource": "/ConfigurationManager/simple/v1/objects/snapshots",
      "messageId": "KART70000-E",
      "message": "The specified volume does not exist. Check the parameter
and try again. If this problem occurs repeatedly, contact customer support.
",

```

```
    "errorCode": {
      "SSB1": "2E20",
      "SSB2": "0000"
    }
  ],
  "partialSuccessResponses": [
    {
      "statusResource": "/ConfigurationManager/simple/v1/objects/command-status/3"
    }
  ]
}
```

---

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

---

## 第2章

# REST API で共通の操作

この章では、非同期処理の API のステータス情報取得やステータス情報の削除など、REST API で共通の操作について説明します。

## 2.1 非同期処理の API のステータス情報を取得する

非同期処理の API のステータス情報を取得します。リクエストの `Authorization` ヘッダには、非同期処理の API の実行時に使用したセッションのトークンを指定してください。

### 実行権限

ストレージ管理者（参照）

### リクエストライン

```
GET <ベース URL>/simple/v1/objects/command-status/<オブジェクト ID>
```

### リクエストメッセージ

#### オブジェクト ID

非同期処理の API を実行したときに返却された `statusResource` の URL の最後に出力されている数値を指定します。

返却された `statusResource` の例：

```
"statusResource": "/ConfigurationManager/simple/v1/objects/command-status/3"
```

属性	型	説明
id	int	(必須) 非同期処理の API の実行結果問い合わせ用 ID

#### クエリパラメータ

なし。

#### ボディ

なし。



## レスポンスメッセージ

### ボディ

ボリュームを作成する API を実行した場合の例：

```
{
  "id": 2,
  "progress": "completed",
  "status": "normal",
  "affectedResources": [
    "/ConfigurationManager/simple/v1/objects/volumes/100"
  ],
  "operationDetails": [
    {
      "operationType": "CREATE",
      "resourceType": "Volume",
      "resourceId": "100"
    }
  ]
}
```

プールを削除する API を実行して失敗した場合の例：

```
{
  "id": 3,
  "progress": "completed",
  "status": "error",
  "errorResource": "/ConfigurationManager/simple/v1/objects/pools/63",
  "errorCode": {
    "SSB1": "2e11",
    "SSB2": "001f"
  },
  "errorMessage": "Snapshot or volume exists. Operation could not be completed.",
  "operationDetails": [
    {
      "operationType": "DELETE",
      "resourceType": "Pool",
      "resourceId": "63"
    }
  ]
}
```

属性	型	説明
id	int	非同期処理の API 実行時の問い合わせ用 ID
progress	string	API の進捗状況 次の値が返ります。 <ul style="list-style-type: none"> <li>unexecuted：API が未実行であることを示す</li> <li>processing：API が実行中であることを示す</li> <li>completed：API が実行完了したことを示す</li> </ul>
status	string	API の実行結果の状態 <ul style="list-style-type: none"> <li>normal：API が成功した状態を示す</li> <li>error：API が失敗した状態を示す</li> </ul>

属性	型	説明
		この属性は、API の実行が完了した場合に出力されます。
affectedResources	string[]	<p>操作対象のリソースにアクセスするための URL の一覧 1 つの API で複数のリソースを操作した場合は、すべての操作対象のリソースの URL が返ります。API が途中で失敗した場合は、処理が完了したことを確認できたリソースの URL だけが返ります。</p> <p>リソースの削除操作が成功した場合、削除対象のリソースの URL が返ります。この URL にアクセスすると 404 エラーとなり、正常に削除されたことが確認できます。この属性は、API の実行が成功した場合に出力されます。</p>
errorResource	string	<p>エラーが発生した URL この属性は、API の実行が失敗した場合に出力されます。</p>
errorCode	object	<p>ストレージシステムのエラーコード ストレージシステムでエラーが発生して、次に示すエラーコードがある場合にだけ値が返ります。</p> <ul style="list-style-type: none"> <li>• SSB1 コード</li> <li>• SSB2 コード</li> </ul> <p>ストレージシステムのエラーコードは、ストレージシステムの保守に必要になります。 この属性は、API の実行が失敗した場合に出力されます。</p>
errorMessage	string	<p>エラーメッセージの内容 この属性は、API の実行が失敗した場合に出力されます。</p>
operationDetails	object[]	<p>操作対象のリソースの詳細情報 リソース種別に示すリソースに対して操作している場合に出力されます。</p> <ul style="list-style-type: none"> <li>• operationType (string) リソースに対する操作種別 <ul style="list-style-type: none"> <li>- CREATE : リソースが作成されたことを示す</li> <li>- UPDATE : リソースが変更されたことを示す</li> <li>- DELETE : リソースが削除されたことを示す</li> </ul> </li> <li>• resourceType (string) 作成、変更、または削除が発生したリソースのリソース種別 <ul style="list-style-type: none"> <li>- CommandStatus : 非同期処理の API のステータス情報に関するリソースであることを示す</li> <li>- Pool : プールに関するリソースであることを示す</li> <li>- Port : ポートに関するリソースであることを示す</li> <li>- Server : サーバに関するリソースであることを示す</li> <li>- Snapshot : スナップショットに関するリソースであることを示す</li> <li>- Volume : ボリュームに関するリソースであることを示す</li> <li>- VolumeServerConnection : ボリュームとサーバの接続に関するリソースであることを示す</li> <li>- ExternalVolume : 外部ボリュームに関するリソースであることを示す</li> </ul> </li> </ul>

属性	型	説明
		<ul style="list-style-type: none"> <li>- QuorumDisk: Quorum ディスクに関するリソースであることを示す</li> <li>• resourceId (string) 作成、変更、または削除が発生したリソースのリソース ID</li> </ul>

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192
.0.2.100/ConfigurationManager/simple/v1/objects/command-status/3
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[commandStatus オブジェクト \(19 ページ\)](#)

## 2.2 非同期処理の API のステータス情報を削除する

非同期処理の API のステータス情報を削除します。

### 実行権限

ストレージ管理者 (参照)

### リクエストライン

```
DELETE <ベース URL>/simple/v1/objects/command-status/<オブジェクト ID>
```

### リクエストメッセージ

#### オブジェクト ID

非同期処理の API を実行したときに返却された statusResource の URL の最後に出力されている数値を指定します。

返却された statusResource の例：

```
"statusResource": "/ConfigurationManager/simple/v1/objects/command-status/3"
```

属性	型	説明
id	int	(必須) 非同期処理の API の実行結果問い合わせ用 ID

### クエリパラメータ

なし。

### ボディ

なし。

## レスポンスメッセージ

### ボディ

属性	型	説明
affectedResources	string[]	削除した API のステータス情報を参照するための URL の一覧
operationDetails	object[]	ステータス情報を削除したリソースの詳細情報 詳細については、commandStatus オブジェクトの属性 operationDetails の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H "Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X DELETE https://192.0.2.100/ConfigurationManager/simple/v1/objects/command-status/3
```

---

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[commandStatus オブジェクト \(19 ページ\)](#)

---

# 第3章

## ストレージシステム全体の操作

この章では、ストレージシステムのさまざまな情報や、プールおよびスナップショットの稼働状況の情報などストレージシステム全体の操作について説明します。

### 3.1 ストレージシステムの情報を取得する

ストレージシステムに関する情報を取得します。

#### 実行権限

ストレージ管理者（参照）

#### リクエストライン

GET <ベース URL>/simple/v1/objects/storage

#### リクエストメッセージ

#### オブジェクト ID

なし。

#### クエリパラメータ

パラメータ	型	フィルタ条件
withEstimatedConfigurableCapacities	boolean	(任意) 設定可能な推定容量の情報を取得するかどうか • true : 取得する • false : 取得しない クエリパラメータを指定しない場合、false が指定されたと見なされます。

#### ボディ

なし。

#### レスポンスメッセージ

#### ボディ

```
{  
  "modelName": "iStorage V310",
```

```

"serial": "800001",
"nickname": "StorageSystem",
"numberOfTotalVolumes": 593,
"numberOfFreeDrives": 0,
"numberOfTotalServers": 2,
"totalPhysicalCapacity": 3605769,
"totalPoolCapacity": 3601542,
"totalPoolPhysicalCapacity": 3601542,
"usedPoolCapacity": 25536,
"freePoolCapacity": 3576006,
"totalPoolCapacityWithTiPool": 3601542,
"totalPoolPhysicalCapacityWithTiPool": 3601542,
"usedPoolCapacityWithTiPool": 25536,
"freePoolCapacityWithTiPool": 3576006,
"savingEffects": {
  "efficiencyDataReduction": 100,
  "preCapacityDataReduction": 25536,
  "postCapacityDataReduction": 25536,
  "efficiencyFmdSaving": -1,
  "preCapacityFmdSaving": 0,
  "postCapacityFmdSaving": 0,
  "isTotalEfficiencySupport": true,
  "totalEfficiencyStatus": "Valid",
  "dataReductionWithoutSystemDataStatus": "Valid",
  "softwareSavingWithoutSystemDataStatus": "NoTargetData",
  "totalEfficiency": 170,
  "dataReductionWithoutSystemData": 100,
  "preCapacityDataReductionWithoutSystemData": 24780,
  "postCapacityDataReductionWithoutSystemData": 24780,
  "softwareSavingWithoutSystemData": -1,
  "calculationStartTime": "2023-12-07T00:59:46Z",
  "calculationEndTime": "2023-12-07T00:59:58Z"
},
"esmVersion": "A3-00-00/11",
"esmOsVersion": "A3-01-04/20",
"dkcMicroVersion": "A3-03-01-10/01",
"warningLedStatus": "OFF",
"esmStatus": "ACTIVE",
"ipAddressIpv4Service": "10.196.191.200",
"ipAddressIpv4Ctl1": "10.196.191.229",
"ipAddressIpv4Ctl2": "10.196.191.230",
"ipAddressIpv6Service": "-",
"ipAddressIpv6Ctl1": "-",
"ipAddressIpv6Ctl2": "-"
}

```

属性	型	説明
modelName	string	ストレージシステムのモデル名
serial	string	ストレージシステムのシリアル番号
nickname	string	ストレージシステムのニックネーム
numberOfTotalVolumes	int	作成済みのボリューム数
numberOfFreeDrives	int	空きドライブ数
numberOfTotalServers	int	HA Storage Manager Embedded で管理しているサーバ数
totalPhysicalCapacity	long	ストレージシステム内のデータドライブに格納できるデータ容量 (MiB)

属性	型	説明
totalPoolCapacity	long	HA Storage Manager Embedded で管理しているプールの総有効容量 (MiB) 拡張中のプールがある場合、属性 totalPoolCapacity と totalPoolPhysicalCapacity の値が異なります。 プールの拡張が完了すると、これらの属性の値は同じになります。
totalPoolPhysicalCapacity	long	HA Storage Manager Embedded で管理しているプールの総容量 (MiB) 拡張中のプールがある場合、拡張後の総容量が出力されます。
usedPoolCapacity	long	HA Storage Manager Embedded で管理しているプールの総使用量 (MiB)
freePoolCapacity	long	HA Storage Manager Embedded で管理しているプールの総空き容量 (MiB)
totalPoolCapacityWithTiPool	long	ストレージシステム内のプールの総有効容量 (MiB) 拡張中のプールがある場合、属性 totalPoolCapacityWithTiPool と totalPoolPhysicalCapacityWithTiPool の値が異なります。 プールの拡張が完了すると、これらの属性の値は同じになります。
totalPoolPhysicalCapacityWithTiPool	long	ストレージシステム内のプールの総容量 (MiB) 拡張中のプールがある場合、拡張後の総容量が出力されます。
usedPoolCapacityWithTiPool	long	ストレージシステム内のプールの総使用量 (MiB)
freePoolCapacityWithTiPool	long	ストレージシステム内のプールの総空き容量 (MiB)
savingEffects	object	ストレージシステム全体に関するさまざまな削減効果の比率や容量の情報が出力されます。 <ul style="list-style-type: none"> <li>• efficiencyDataReduction (int) ※1 容量削減機能 (重複排除および圧縮機能) によって削減された HA Storage Manager Embedded で管理しているすべてのデータの削減効果の比率 削減後の値を 100 として計算した削減前の値が出力されます。 (例) 105 (削減前) : 100 (削減後) の場合、「"efficiencyDataReduction": 105」と出力されます。 -1 が出力された場合、その値は無効な情報です。</li> <li>• preCapacityDataReduction (long) ※1 容量削減機能 (重複排除および圧縮機能) によって削減される前の HA Storage Manager Embedded で管理しているすべてのデータの容量 (MiB)</li> <li>• postCapacityDataReduction (long) ※1 容量削減機能 (重複排除および圧縮機能) によって削減された後の HA Storage Manager Embedded で管理しているすべてのデータの容量 (MiB)</li> <li>• efficiencyFmdSaving (int) 無効な値を示す -1 が出力されます。</li> <li>• preCapacityFmdSaving (long)</li> </ul>

属性	型	説明
		<p>属性 <code>efficiencyFmdSaving</code> に無効な値-1 が出力されている場合、この属性には 0 が出力されます。</p> <ul style="list-style-type: none"> <li>• <code>postCapacityFmdSaving</code> (long) 属性 <code>efficiencyFmdSaving</code> に無効な値-1 が出力されている場合、この属性には 0 が出力されます。</li> <li>• <code>isTotalEfficiencySupport</code> (boolean) ストレージシステムの利用効率を上げるための機能（容量削減機能（重複排除および圧縮機能）、スナップショットによるバックアップ、Dynamic Provisioning による容量仮想化）による容量消費の節減効果（以降、「合計効果」と表記します）がサポートされているかどうか出力されます。  <ul style="list-style-type: none"> <li>- true : サポート</li> <li>- false : 未サポート</li> </ul> </li> <li>• <code>totalEfficiencyStatus</code> (string) ストレージシステム全体の合計効果の状態  <ul style="list-style-type: none"> <li>- Valid : 有効</li> <li>- NotSupported : 未サポート</li> <li>- CalculationInProgress : 集計中</li> <li>- NoTargetData : 集計対象のデータがない</li> <li>- Unknown : 不明</li> </ul> </li> <li>• <code>dataReductionWithoutSystemDataStatus</code> (string) 容量削減機能（重複排除および圧縮機能）によって削減されたストレージシステム全体の削減効果の状態  <ul style="list-style-type: none"> <li>- Valid : 有効</li> <li>- NotSupported : 未サポート</li> <li>- CalculationInProgress : 集計中</li> <li>- NoTargetData : 集計対象のデータがない</li> <li>- Unknown : 不明</li> </ul> </li> <li>• <code>softwareSavingWithoutSystemDataStatus</code> (string) 容量削減機能（重複排除および圧縮機能）による、容量削減機能が有効なボリュームに対するストレージシステム全体の削減効果の状態  <ul style="list-style-type: none"> <li>- Valid : 有効</li> <li>- NotSupported : 未サポート</li> <li>- CalculationInProgress : 集計中</li> <li>- NoTargetData : 集計対象のデータがない</li> <li>- Unknown : 不明</li> </ul> </li> <li>• <code>totalEfficiency</code> (long) ※2 ストレージシステム全体の合計効果の比率 削減後の値を 100 として計算した削減前の値が出力されます。            (例) 529652 (削減前) : 100 (削減後) の場合、「"totalEfficiency": 529652」と出力されます。            プールからボリュームを作成したあと、データが書き込まれる前は、最大値 9223372036854775807 が出力されます。         </li> </ul>



属性	型	説明
		<p>計算を開始した日時、および計算を終了した日時は、属性 calculationStartTime および calculationEndTime で確認できます。</p> <p>次の場合、無効値を示す-1 が出力されます。</p> <ul style="list-style-type: none"> <li>- 属性 isTotalEfficiencySupport が false</li> <li>- 属性 totalEfficiencyStatus が valid 以外</li> </ul> <p>• dataReductionWithoutSystemData (long) ※2</p> <p>容量削減機能（重複排除および圧縮機能）によって削減されたストレージシステム全体の削減効果の比率</p> <p>削減後の値を 100 として計算した削減前の値が出力されます。</p> <p>(例) 138 (削減前) : 100 (削減後) の場合、 「"dataReductionWithoutSystemData": 138」 と出力されます。</p> <p>計算を開始した日時、および計算を終了した日時は、属性 calculationStartTime および calculationEndTime で確認できます。</p> <p>次の場合、無効値を示す-1 が出力されます。</p> <ul style="list-style-type: none"> <li>- 属性 isTotalEfficiencySupport が false</li> <li>- 属性 dataReductionWithoutSystemDataStatus が valid 以外</li> </ul> <p>• softwareSavingWithoutSystemData (long) ※2</p> <p>容量削減機能（重複排除および圧縮機能）による、容量削減機能が有効なボリュームに対するストレージシステム全体の削減効果の比率</p> <p>削減後の値を 100 として計算した削減前の値が出力されます。</p> <p>(例) 138 (削減前) : 100 (削減後) の場合、 「"softwareSavingWithoutSystemData": 138」 と出力されます。</p> <p>次の場合、無効値を示す-1 が出力されます。</p> <ul style="list-style-type: none"> <li>- 属性 isTotalEfficiencySupport が false</li> <li>- 属性 softwareSavingWithoutSystemDataStatus が valid 以外</li> </ul> <p>• preCapacityDataReductionWithoutSystemData (long) ※2</p> <p>容量削減機能（重複排除および圧縮機能）によって削減される前のストレージシステム全体の容量 (MiB)</p> <p>属性 isTotalEfficiencySupport が false の場合、無効値を示す-1 が出力されます。</p> <p>• postCapacityDataReductionWithoutSystemData (long) ※2</p> <p>容量削減機能（重複排除および圧縮機能）によって削減された後のストレージシステム全体の容量 (MiB)</p> <p>属性 isTotalEfficiencySupport が false の場合、無効値を示す-1 が出力されます。</p> <p>• calculationStartTime (ISO8601string)</p> <p>計算を開始した日時 (UTC) が YYYY-MM-DDThh:mm:ssZ 形式で出力されます。</p> <p>• calculationEndTime (ISO8601string)</p>

属性	型	説明
		計算を終了した日時 (UTC) が <i>YYYY-MM-DDThh:mm:ssZ</i> 形式で出力されます。
gumVersion	string	ESM のバージョン
esmOsVersion	string	ESM の OS のバージョン
dkcMicroVersion	string	ストレージシステムのマイクロコードのバージョン
warningLedStatus	string	ストレージシステムの障害状態 <ul style="list-style-type: none"> <li>• OFF : 通常</li> <li>• ON : 障害が発生している</li> <li>• BLINK : 確認できていない SIM がある</li> <li>• Unknown : 不明</li> </ul>
esmStatus	string	ESM の状態 <ul style="list-style-type: none"> <li>• ACTIVE : 稼働中</li> <li>• STANDBY : 待機中</li> <li>• ERROR : 障害発生中</li> <li>• STARTING : 起動中</li> <li>• Unknown : 不明</li> </ul>
ipAddressIpv4Service	string	ESM の IPv4 の IP アドレス (サービス IP)
ipAddressIpv4Ctl1	string	コントローラ 1 (CTL01) の IPv4 の IP アドレス
ipAddressIpv4Ctl2	string	コントローラ 2 (CTL02) の IPv4 の IP アドレス
ipAddressIpv6Service	string	ESM の IPv6 の IP アドレス (サービス IP)
ipAddressIpv6Ctl1	string	コントローラ 1 (CTL01) の IPv6 の IP アドレス
ipAddressIpv6Ctl2	string	コントローラ 2 (CTL02) の IPv6 の IP アドレス

**注※1**

ユーザデータのほかにストレージシステムが生成するメタデータやガベージデータなどを含んだ値で計算します。そのため、削減後の値の方が大きくなる場合があります。

**注※2**

ストレージシステムが生成するメタデータやガベージデータなどは含まれません。

クエリパラメータ `withEstimatedConfigurableCapacities` に `true` を指定した場合、ストレージシステムの情報を取得する場合の属性に加えて次に示す属性が取得されます。

属性	型	説明
estimatedConfigurablePoolCapacity	long	プールに設定できる推定容量(MiB) -1 が出力された場合、その値は無効な情報です。
estimatedConfigurableVolumeCapacity	long	ボリュームに設定できる推定容量(MiB) -1 が出力された場合、その値は無効な情報です。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H "Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192.0.2.100/ConfigurationManager/simple/v1/objects/storage
```

---

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

---

## 3.2 スナップショットとプールの稼働状態の情報を取得する

プールおよびスナップショットの状態の情報を取得して、ストレージリソースの稼働状態を確認します。

## 実行権限

ストレージ管理者 (参照)

## リクエストライン

```
GET <ベース URL>/simple/v1/objects/health-status
```

## リクエストメッセージ

### オブジェクト ID

なし。

### クエリパラメータ

なし。

### ボディ

なし。

## レスポンスメッセージ

### ボディ

```
{
  "poolStatus":{
    "summary":"PartiallyBlocked",
    "abnormalItems":[
      {
        "status":"PartiallyBlocked",
        "poolIds":[
          0
        ]
      },
      {
        "status":"ExceededThreshold",
        "poolIds":[
          0,77,39,26
        ]
      }
    ]
  },
  "snapshotStatus":{
    "summary":"Normal"
  }
}
```

属性	型	説明
poolStatus	object	<p>プールの状態</p> <ul style="list-style-type: none"> <li>summary (string) <ul style="list-style-type: none"> <li>Normal: すべてのプールが正常</li> <li>ExceededThreshold: プールの使用率がしきい値を超えているプールが1つでもある</li> <li>PartiallyBlocked: プールを構成するボリュームが1つでも閉塞している</li> <li>Error: 満杯でエラー状態のプールが1つでもある</li> </ul> </li> <li>abnormalItems (object[]) <p>正常でないプールの情報が出力されます。</p> <ul style="list-style-type: none"> <li>status (string) <ul style="list-style-type: none"> <li>PartiallyBlocked: プールを構成するボリュームが一部閉塞している</li> <li>ExceededThreshold: プールの使用率がしきい値を超えている</li> <li>Error: プールが満杯でエラー状態</li> </ul> </li> <li>poolIds (int[]) <p>プール ID</p> </li> </ul> </li> </ul>
snapshotStatus	object	<p>スナップショットの状態</p> <ul style="list-style-type: none"> <li>summary (string) <ul style="list-style-type: none"> <li>Normal: すべてのスナップショットが正常</li> <li>Error: 障害が発生しているスナップショットが1つでもある</li> </ul> </li> <li>abnormalItems (object[])</li> </ul>

属性	型	説明
		正常でないスナップショットの情報が出力されます。 <ul style="list-style-type: none"><li>- status (string)</li><li>- Error: スナップショットに障害が発生している</li><li>- masterVolumeIds (int[]) スナップショットの作成元ボリューム ID</li></ul>

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H "Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192.0.2.100/ConfigurationManager/simple/v1/objects/health-status
```

---

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

---

## 第4章

# 管理リソースの準備とボリューム割り当ての操作

この章では、ドライブを指定したプールの作成、ボリュームの作成、使用率の高いボリュームの容量拡張やプールにドライブを追加する容量拡張などを REST API で行う操作について説明します。HA Storage Manager Embedded でサーバを管理する場合（以降はサーバと表記）は、REST API を使用して、サーバの情報の管理やサーバとポート間のパス情報の管理、サーバへのボリュームの割り当てなどの操作もできます。

## 4.1 ドライブの管理

REST API を使用してドライブに関する情報を一覧で取得したり、特定のドライブの情報を取得します。

### 4.1.1 ドライブの情報を取得する

ドライブに関する情報を取得します。ドライブの条件を指定して、絞り込んだ情報を取得することもできます。

#### 実行権限

ストレージ管理者（参照）

#### リクエストライン

```
GET <ベース URL>/simple/v1/objects/drives
```

#### リクエストメッセージ

##### オブジェクト ID

なし。

##### クエリパラメータ

パラメータ	型	フィルタ条件
poolId	int	(任意) ドライブが属するプール ID poolName と同時に指定しないでください。
poolName	string	(任意) ドライブが属するプール名

パラメータ	型	フィルタ条件
		指定した値の一部がプール名と一致する場合もドライブの情報が取得されます。 poolId と同時に指定しないでください。poolId を指定している場合、このパラメータは無視されます。
status	string	(任意) 取得するドライブの状態として、次の値を指定します。 <ul style="list-style-type: none"> <li>• Normal : 正常</li> <li>• Warning : エラー部位がある状態</li> <li>• Copying : コピー中</li> <li>• CopyIncomplete : コピー不完全</li> <li>• Reserved : スペアドライブが使用できない状態</li> <li>• Failed : 障害によるエラー状態</li> <li>• Blocked : 保守によるエラー状態</li> </ul>
useOfTheDrive	string	(任意) 取得するドライブの用途として、次の値を指定します。 <ul style="list-style-type: none"> <li>• data : データドライブ</li> <li>• spare : スペアドライブ</li> <li>• free : 未使用のドライブ</li> </ul>

## ボディ

なし。

## レスポンスメッセージ

## ボディ

ドライブの用途としてデータドライブ（クエリパラメータ useOfTheDrive に data を指定）の情報を取得した出力例を次に示します。

```

{
  "data": [
    {
      "location": "0-0",
      "status": "Normal",
      "driveType": "SSD",
      "driveInterface": "NVMe",
      "driveRpm": "NUMBER_0",
      "driveCapacity": 1900,
      "displayDriveCapacity": "1.9 TB",
      "typeCode": "SNB5B-R1R9NC",
      "poolIds": [
        0
      ],
      "useOfTheDrive": "data"
    },
    {
      "location": "0-1",
      "status": "Normal",
      "driveType": "SSD",

```

```

        "driveInterface": "NVMe",
        "driveRpm": "NUMBER_0",
        "driveCapacity": 1900,
        "displayDriveCapacity": "1.9 TB",
        "typeCode": "SNB5B-R1R9NC",
        "poolIds": [
            0
        ],
        "useOfTheDrive": "data"
    },
    {
        "location": "0-2",
        "status": "Normal",
        "driveType": "SSD",
        "driveInterface": "NVMe",
        "driveRpm": "NUMBER_0",
        "driveCapacity": 1900,
        "displayDriveCapacity": "1.9 TB",
        "typeCode": "SNB5B-R1R9NC",
        "poolIds": [
            0
        ],
        "useOfTheDrive": "data"
    },
    ...
    ...
],
"count": 10
}

```

属性	型	説明
location	string	ドライブの位置
status	string	ドライブの状態 <ul style="list-style-type: none"> <li>• Normal : 正常</li> <li>• Warning : エラー部位がある状態</li> <li>• Copying : コピー中</li> <li>• CopyIncomplete : コピー不完全</li> <li>• Reserved : スペアドライブが使用できない状態</li> <li>• Failed : 障害によるエラー状態</li> <li>• Blocked : 保守によるエラー状態</li> <li>• Unknown : 状態不明</li> </ul>
driveType	string	ドライブタイプ <ul style="list-style-type: none"> <li>• SSD</li> <li>• HDD (iStorage V110/V310 でのみ表示されます)</li> </ul>
driveInterface	string	ドライブインタフェース <ul style="list-style-type: none"> <li>• NVMe</li> <li>• SAS (iStorage V110/V310 でのみ表示されます)</li> </ul>
driveRpm	string	ドライブの回転数 (rpm) <ul style="list-style-type: none"> <li>• NUMBER_0</li> <li>• NUMBER_7200</li> <li>• NUMBER_10000</li> <li>• NUMBER_15000</li> </ul>



属性	型	説明
		<ul style="list-style-type: none"> <li>Unknown</li> </ul>
driveCapacity	int	ドライブの容量 (GB)
displayDriveCapacity	string	ドライブの容量が単位付き (GB または TB) で出力されます。
typeCode	string	ドライブタイプコード
poolIds	int[]	ドライブが属するプール ID のリスト
useOfTheDrive	string	ドライブの用途 <ul style="list-style-type: none"> <li>data : データドライブ</li> <li>spare : スペアドライブ</li> <li>free : 未使用のドライブ</li> </ul>

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192
.0.2.100/ConfigurationManager/simple/v1/objects/drives?useOfTheDrive=free
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[データオブジェクト \(17 ページ\)](#)

## 4.1.2 特定のドライブの情報を取得する

ドライブの位置を指定して、特定のドライブの情報を取得します。

## 実行権限

ストレージ管理者 (参照)

## リクエストライン

```
GET <ベース URL>/simple/v1/objects/drives/<オブジェクト ID>
```

## リクエストメッセージ

### オブジェクト ID

ドライブの情報取得で取得した `location` の値を指定します。

属性	型	説明
<code>location</code>	<code>string</code>	(必須) ドライブの位置

### クエリパラメータ

なし。

### ボディ

なし。

## レスポンスメッセージ

### ボディ

特定のドライブ (`location: 0-0`) の情報を取得した出力例を次に示します。

```
{
  "location": "0-0",
  "status": "Normal",
  "driveType": "SSD",
  "driveInterface": "NVMe",
  "driveRpm": "NUMBER_0",
  "driveCapacity": 1900,
  "displayDriveCapacity": "1.9 TB",
  "typeCode": "SNB5B-R1R9NC",
  "poolIds": [
    0
  ],
  "useOfTheDrive": "data"
}
```

取得する属性については、ドライブの情報取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192
.0.2.100/ConfigurationManager/simple/v1/objects/drives/0-0
```

---

**関連リンク**

---

[HTTP ステータスコード \(13 ページ\)](#)[ドライブの情報を取得する \(38 ページ\)](#)

---

### 4.1.3 ドライブをスペアドライブに設定する

ドライブの数を指定して、スペアドライブに設定します。

Advanced Dynamic Provisioning (以降「ADP」と表記します) を有効にした場合は、スペアドライブの設定をする必要がありません。

## 実行権限

ストレージ管理者 (プロビジョニング)

## リクエストライン

```
POST <ベース URL>/simple/v1/objects/drives/actions/set-spare/invoke
```

## リクエストメッセージ

### オブジェクト ID

なし。

### クエリパラメータ

なし。

### ボディ

ドライブタイプコードが SNB5B-R1R9NC のドライブを 4 台スペアドライブに設定する例を次に示します。

```
{
  "additionalSpareDrives": [
    {
      "driveTypeCode": "SNB5B-R1R9NC",
      "driveCount": 4
    }
  ]
}
```

属性	型	説明
additionalSpareDrives	object[]	スペアドライブに設定するドライブの情報 • (必須) driveTypeCode (string)

属性	型	説明
		ドライブタイプコードを 12 文字で指定します。※ 例：SNB5B-R1R9NC • (必須) driveCount (int) ドライブの数を 1～1440 までの整数で指定します。

**注※**

設定するドライブの条件（ドライブタイプ、ドライブインタフェース、ドライブの回転数、ドライブの容量）が同じでも、ドライブタイプコードが異なる場合があります。どのドライブタイプコードを指定してもドライブの条件に一致するドライブがスペアドライブに設定されます。設定されたドライブの情報は、ドライブの情報を取得する API で確認できます。

## レスポンスメッセージ

**ボディ**

属性	型	説明
statusResource	string	設定したスペアドライブの実行結果の問い合わせに使用する URL

**メモ**

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-bin
ary @./InputParameters.json https://192.0.2.100/ConfigurationManager/simple
/v1/objects/drives/actions/set-spare/invoke
```

**関連リンク**

[HTTP ステータスコード \(13 ページ\)](#)

[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

[ドライブの情報を取得する \(38 ページ\)](#)

## 4.1.4 スペアドライブの設定を解除する

ドライブの数を指定して、スペアドライブの設定を解除します。

### 実行権限

ストレージ管理者（プロビジョニング）

### リクエストライン

```
POST <ベース URL>/simple/v1/objects/drives/actions/release-spare/invoke
```

### リクエストメッセージ

#### オブジェクト ID

なし。

#### クエリパラメータ

なし。

#### ボディ

ドライブタイプコードが SNB5B-R1R9NC のスペアドライブを 4 台解除する例を次に示します。

```
{
  "releasedSpareDrives": [
    {
      "driveTypeCode": "SNB5B-R1R9NC",
      "driveCount": 4
    }
  ]
}
```

属性	型	説明
releasedSpareDrives	object[]	スペアドライブの設定を解除するドライブの情報 <ul style="list-style-type: none"><li>（必須）driveTypeCode (string) ドライブタイプコードを 12 文字で指定します。※ 例：SNB5B-R1R9NC</li><li>（必須）driveCount (int) スペアドライブの数を 1～1440 までの整数で指定します。</li></ul>

**注※**

解除するドライブの条件（ドライブタイプ、ドライブインタフェース、ドライブの回転数、ドライブの容量）が同じでも、ドライブタイプコードが異なる場合があります。どのドライブタイプコードを指定してもドライブの条件に一致するスペアドライブが解除されます。解除されたドライブの情報は、ドライブの情報を取得する API で確認できます。

## レスポンスメッセージ

**ボディ**

属性	型	説明
statusResource	string	スペアドライブの設定を解除したドライブの実行結果の問い合わせに使用する URL

**メモ**

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H "Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/simple/v1/objects/drives/actions/release-spare/invoke
```

**関連リンク**

[HTTP ステータスコード \(13 ページ\)](#)

[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

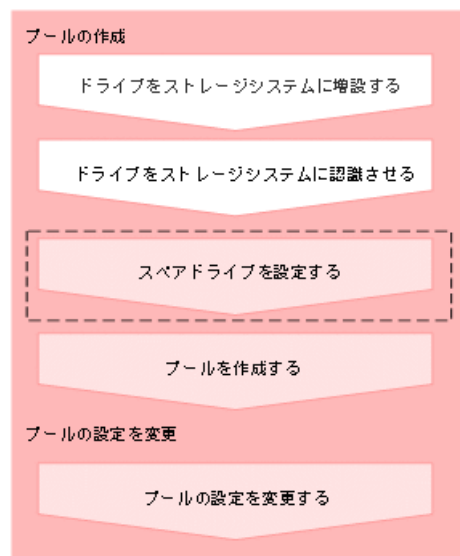
[ドライブの情報を取得する \(38 ページ\)](#)

## 4.2 プールの管理

REST API を使用してプールに関する情報を取得したり、特定のプールに関する詳細な情報を取得します。ドライブを指定してプールを作成したり、プールの使用率が高くなった場合に、プールにドライブを追加することでプールの容量を拡張する操作について説明します。

### プールを作成する流れ

プールを作成するには、あらかじめストレージシステムにドライブを認識させておく必要があります。プールを作成したあとに設定を変更することもできます。



(凡例)

- : REST APIで操作する手順
- : REST API以外で操作する手順
- : 必要に応じて実施する操作

- ドライブをストレージシステムに増設する
- ドライブをストレージシステムに認識させる

Maintenance Utility を使用してドライブを認識させます。

操作方法については、Maintenance Utility のヘルプを参照してください。

- ドライブをスベアドライブに設定する

ドライブの数を指定して、スベアドライブに設定します。

- プールを作成する

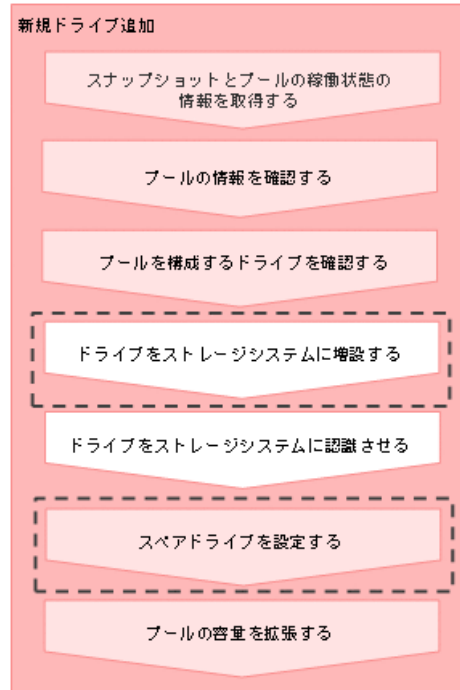
プール名とドライブ情報を指定して、プールを作成します。

- プールの設定を変更する




プール ID を指定して、プールの設定（プール名やしきい値など）を変更します。

## プールの容量を拡張する流れ

プールの容量を拡張するには、新規にドライブを増設します。ドライブの情報は、プールの情報を取得する API で確認できます。複数のプールで運用している場合は、未使用のプールを削除してドライブを流用する方法もあります。



(凡例)

-  : REST API で操作する手順
-  : REST API 以外で操作する手順
-  : 必要に応じて実施する操作

- スナップショットとプールの稼働状態の情報を取得する  
プールおよびスナップショットの状態の情報を取得して、ストレージリソースの稼働状態を確認します。
- プールの情報を取得する  
プールに関する情報を取得します。プールの条件を指定して、絞り込んだ情報を取得することもできます。
- 特定のプールの情報を取得する  
プール ID を指定して、特定のプールの情報を取得します。
- ドライブの情報を取得する  
ドライブに関する情報を取得します。ドライブの条件を指定して、絞り込んだ情報を取得することもできます。
- 特定のドライブの情報を取得する  
ドライブの位置を指定して、特定のドライブの情報を取得します。



- ドライブをストレージシステムに増設する
- ドライブをストレージシステムに認識させる

Maintenance Utility を使用してドライブを認識させます。

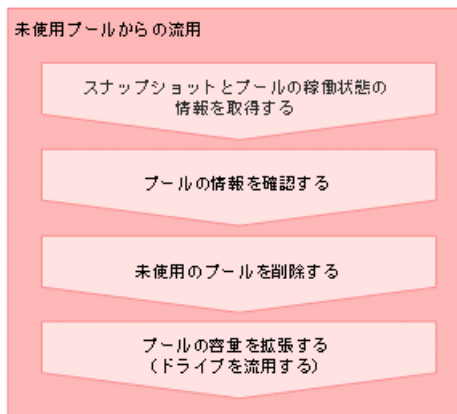
操作方法については、Maintenance Utility のヘルプを参照してください。

- ドライブをスペアドライブに設定する

ドライブの数を指定して、スペアドライブに設定します。

- プールにドライブを追加する

プール ID を指定して、プールにドライブを追加することでプールの容量を拡張します。



(凡例)

 : REST APIで操作する手順

- スナップショットとプールの稼働状態の情報を取得する

プールおよびスナップショットの状態の情報を取得して、ストレージリソースの稼働状態を確認します。

- プールの情報を取得する

プールに関する情報を取得します。プールの条件を指定して、絞り込んだ情報を取得することもできます。

- 特定のプールの情報を取得する

プール ID を指定して、特定のプールの情報を取得します。

- プールを削除する

プール ID を指定して、プールを削除します。

- プールにドライブを追加する

プール ID を指定して、プールにドライブを追加することでプールの容量を拡張します。

## 4.2.1 プールの情報を取得する

プールに関する情報を取得します。プールの条件を指定して、絞り込んだ情報を取得することもできます。

### 実行権限

ストレージ管理者（参照）

### リクエストライン

```
GET <ベース URL>/simple/v1/objects/pools
```

### リクエストメッセージ

#### オブジェクト ID

なし。

#### クエリパラメータ

パラメータ	型	フィルタ条件
name	string	(任意) プール名 指定した値の一部がプール名と一致する場合もプールの情報が取得されます。
status	string	(任意) 取得するプールの状態として、次の値を指定します。 <ul style="list-style-type: none"> <li>• Normal : 正常</li> <li>• ExceededThreshold : プールの使用率がしきい値を超えている</li> <li>• Error : プールが満杯でエラー状態</li> </ul>
configStatus	string	(任意) 取得するプールを構成するボリュームの状態として、PartiallyBlocked（プールを構成するボリュームが一部閉塞している）を指定します。

#### ボディ

なし。

### レスポンスメッセージ

#### ボディ

```
{
  "data": [
    {
      "id": 0,
```

```

    "name": "test_pool1",
    "status": "Normal",
    "encryption": "DISABLED",
    "totalCapacity": 3601542,
    "effectiveCapacity": 3601542,
    "usedCapacity": 210,
    "freeCapacity": 3601332,
    "capacityManage": {
        "usedCapacityRate": 1,
        "thresholdWarning": 70,
        "thresholdDepletion": 80
    },
    "savingEffects": {
        "efficiencyDataReduction": -1,
        "efficiencyFmdSaving": -1,
        "preCapacityFmdSaving": 0,
        "postCapacityFmdSaving": 0,
        "isTotalEfficiencySupport": true,
        "totalEfficiencyStatus": "Valid",
        "dataReductionWithoutSystemDataStatus": "NoTargetData",
        "softwareSavingWithoutSystemDataStatus": "NoTargetData",
        "totalEfficiency": 9223372036854775807,
        "dataReductionWithoutSystemData": -1,
        "softwareSavingWithoutSystemData": -1,
        "calculationStartTime": "2023-12-04T13:16:15Z",
        "calculationEndTime": "2023-12-04T13:16:22Z"
    },
    "configStatus": [],
    "numberOfVolumes": 10,
    "numberOfTiers": 0,
    "tiers": [],
    "subscriptionLimit": {
        "isEnabled": false,
        "currentRate": 928
    },
    "containsCapacitySavingVolume": true
}
],
"count": 1
}

```

属性	型	説明
id	int	プール ID
name	string	プール名
status	string	プールの状態 <ul style="list-style-type: none"> <li>Normal : 正常</li> <li>ExceededThreshold: プールの使用率がしきい値を超えている</li> <li>Error : プールが満杯でエラー状態</li> </ul>
encryption	string	プールの暗号化状態 <ul style="list-style-type: none"> <li>DISABLED : プールの暗号化が無効</li> <li>UNKNOWN_EXTERNAL_VOLUME : プールに外部ボリュームがあるため、プールの暗号化の状態が不明</li> <li>UNKNOWN : プールが閉塞しているためプールの暗号化の状態が不明</li> </ul>

属性	型	説明
totalCapacity	long	プールの総容量 (MiB) プールを拡張すると、拡張後の容量が属性 totalCapacity として出力されます。
effectiveCapacity	long	プールの有効容量 (MiB) プールの拡張中は、属性 totalCapacity と effectiveCapacity の値が異なります。 プールの拡張が完了すると、これらの属性の値は同じになります。
usedCapacity	long	プールの使用容量 (MiB)
freeCapacity	long	プールの空き容量 (MiB)
capacityManage	object	プールの容量に関する設定情報が出力されます。 <ul style="list-style-type: none"> <li>usedCapacityRate (int) プールの使用率 (%)</li> <li>thresholdWarning (int) プールの警告しきい値 (%)</li> <li>thresholdDepletion (int) プールの枯渇しきい値 (%)</li> </ul>
savingEffects	object	プールに関する削減効果の比率などの情報が出力されます。 <ul style="list-style-type: none"> <li>efficiencyDataReduction (int) ※1 容量削減機能（重複排除および圧縮機能）によって削減されたプールの削減効果の比率 削減後の値を 100 として計算した削減前の値が出力されます。 (例) 105 (削減前) : 100 (削減後) の場合、 「"efficiencyDataReduction": 105」と出力されます。 -1 が出力された場合、その値は無効な情報です。</li> <li>efficiencyFmdSaving (int) 無効な値を示す-1 が出力されます。</li> <li>preCapacityFmdSaving (long) 属性 efficiencyFmdSaving に無効な値-1 が出力されている場合、この属性には 0 が出力されます。</li> <li>postCapacityFmdSaving (long) 属性 efficiencyFmdSaving に無効な値-1 が出力されている場合、この属性には 0 が出力されます。</li> <li>isTotalEfficiencySupport (boolean) 合計効果をサポートしているかどうか出力されます。  <ul style="list-style-type: none"> <li>true : サポート</li> <li>false : 未サポート</li> </ul> </li> <li>totalEfficiencyStatus (string) プールに格納されているすべてのボリュームに対する合計効果の状態  <ul style="list-style-type: none"> <li>Valid : 有効</li> <li>NotSupported : 未サポート</li> <li>CalculationInProgress : 集計中</li> </ul> </li> </ul>

属性	型	説明
		<ul style="list-style-type: none"> <li>- NoTargetData : 集計対象のデータがない</li> <li>- Unknown : 不明</li> </ul> <ul style="list-style-type: none"> <li>• dataReductionWithoutSystemDataStatus (string) 容量削減機能（重複排除および圧縮機能）によるプールの削減効果の状態 <ul style="list-style-type: none"> <li>- Valid : 有効</li> <li>- NotSupported : 未サポート</li> <li>- CalculationInProgress : 集計中</li> <li>- NoTargetData : 集計対象のデータがない</li> <li>- Unknown : 不明</li> </ul> </li> <li>• softwareSavingWithoutSystemDataStatus (string) 容量削減機能（重複排除および圧縮機能）による、容量削減機能が有効なボリュームに対するプールの削減効果の状態 <ul style="list-style-type: none"> <li>- Valid : 有効</li> <li>- NotSupported : 未サポート</li> <li>- CalculationInProgress : 集計中</li> <li>- NoTargetData : 集計対象のデータがない</li> <li>- Unknown : 不明</li> </ul> </li> <li>• totalEfficiency (long) ※2 プールに格納されているすべてのボリュームに対する合計効果の比率 削減後の値を 100 として計算した削減前の値が出力されます。 (例) 6742 (削減前) : 100 (削減後) の場合、「"totalEfficiency": 6742」と出力されます。 プールからボリュームを作成したあと、データが書き込まれる前は、最大値 9223372036854775807 が出力されます。 計算を開始した日時、および計算を終了した日時は、属性 calculationStartTime および calculationEndTime で確認できます。 次の場合、無効値を示す-1 が出力されます。 <ul style="list-style-type: none"> <li>- 属性 isTotalEfficiencySupport が false</li> <li>- 属性 totalEfficiencyStatus が Valid 以外</li> </ul> </li> <li>• dataReductionWithoutSystemData (long) ※2 容量削減機能（重複排除および圧縮機能）によるプールの削減効果の比率 削減後の値を 100 として計算した削減前の値が出力されます。 (例) 138 (削減前) : 100 (削減後) の場合、「"dataReductionWithoutSystemData": 138」と出力されます。 計算を開始した日時、および計算を終了した日時は、属性 calculationStartTime および calculationEndTime で確認できます。 次の場合、無効値を示す-1 が出力されます。 <ul style="list-style-type: none"> <li>- 属性 isTotalEfficiencySupport が false</li> </ul> </li> </ul>

属性	型	説明
		<ul style="list-style-type: none"> <li>- 属性 <code>dataReductionWithoutSystemDataStatus</code> が <code>Valid</code> 以外</li> <li>• <code>softwareSavingWithoutSystemData</code> (long) ※2 容量削減機能（重複排除および圧縮機能）による、容量削減機能が有効なボリュームに対するプールの削減効果の比率 削減後の値を 100 として計算した削減前の値が出力されます。 (例) 138 (削減前) : 100 (削減後) の場合、 「<code>softwareSavingWithoutSystemData</code>: 138」が出力されます。 次の場合、無効値を示す -1 が出力されます。 <ul style="list-style-type: none"> <li>- 属性 <code>isTotalEfficiencySupport</code> が <code>false</code></li> <li>- 属性 <code>softwareSavingWithoutSystemDataStatus</code> が <code>Valid</code> 以外</li> </ul> </li> <li>• <code>calculationStartTime</code> (ISO8601string) 計算を開始した日時 (UTC) が <code>YYYY-MM-DDThh:mm:ssZ</code> 形式で出力されます。</li> <li>• <code>calculationEndTime</code> (ISO8601string) 計算を終了した日時 (UTC) が <code>YYYY-MM-DDThh:mm:ssZ</code> 形式で出力されます。</li> </ul>
<code>configStatus</code>	<code>string[]</code>	プールを構成するボリュームの状態 <ul style="list-style-type: none"> <li>• <code>PartiallyBlocked</code> : プールを構成するボリュームが一部閉塞している</li> </ul>
<code>numberOfVolumes</code>	<code>int</code>	プールに関連づけられたボリュームの数
<code>numberOfTiers</code>	<code>int</code>	現在使用しない属性です。
<code>tiers</code>	<code>object[]</code>	現在使用しない属性です。
<code>subscriptionLimit</code>	<code>object</code>	現在使用しない属性です。
<code>containsCapacitySavingVolume</code>	<code>boolean</code>	容量削減機能（重複排除および圧縮機能）が有効なボリュームがプールに含まれているかどうか <ul style="list-style-type: none"> <li>• <code>true</code> : 含まれている</li> <li>• <code>false</code> : 含まれていない</li> </ul>

**注※1**

ユーザデータのほかにストレージシステムが生成するメタデータやガベージデータなどを含んだ値で計算します。そのため、削減後の値の方が大きくなる場合があります。

**注※2**

ストレージシステムが生成するメタデータやガベージデータなどは含まれません。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192
.0.2.100/ConfigurationManager/simple/v1/objects/pools
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[データオブジェクト \(17 ページ\)](#)

## 4.2.2 特定のプールの情報を取得する

プール ID を指定して、特定のプールの情報を取得します。

### 実行権限

ストレージ管理者 (参照)

### リクエストライン

```
GET <ベース URL>/simple/v1/objects/pools/<オブジェクト ID>
```

### リクエストメッセージ

#### オブジェクト ID

プールの情報取得で取得した id の値を指定します。

属性	型	説明
id	int	(必須) プール ID

#### クエリパラメータ

なし。

#### ボディ

なし。

### レスポンスメッセージ

#### ボディ

プール ID が 7 のプールの情報を取得した出力例を次に示します。

```
{
  "id": 7,
  "name": "test",
  "status": "Normal",
  "encryption": "DISABLED",
  "totalCapacity": 11673600,
  "effectiveCapacity": 11673600,
  "usedCapacity": 0,
  "freeCapacity": 11673600,
  "capacityManage": {
    "usedCapacityRate": 0,
    "thresholdWarning": 70,
    "thresholdDepletion": 80
  },
  "savingEffects": {
    "efficiencyDataReduction": 220,
    "efficiencyFmdSaving": 120,
    "preCapacityFmdSaving": 0,
    "postCapacityFmdSaving": 0,
    "isTotalEfficiencySupport": true,
    "totalEfficiencyStatus": "Valid",
    "dataReductionWithoutSystemDataStatus": "Valid",
    "softwareSavingWithoutSystemDataStatus": "Valid",
    "totalEfficiency": 12345,
    "dataReductionWithoutSystemData": 102,
    "softwareSavingWithoutSystemData": 103,
    "calculationStartTime": "2023-12-07T00:46:02Z",
    "calculationEndTime": "2023-12-07T00:56:02Z"
  },
  "configStatus": [],
  "numberOfVolumes": 0,
  "numberOfTiers": 0,
  "numberOfDriveTypes": 1,
  "tiers": [],
  "drives": [
    {
      "driveType": "SSD",
      "driveInterface": "NVMe",
      "driveRpm": "NUMBER_0",
      "driveCapacity": 1900,
      "displayDriveCapacity": "1.9 TB",
      "totalCapacity": 17100,
      "numberOfDrives": 9,
      "locations": [
        "57-4",
        "57-5",
        "57-6",
        "57-7",
        "57-8",
        "57-9",
        "57-10",
        "57-11",
        "57-12"
      ],
      "raidLevel": "RAID6",
      "parityGroupType": "ADP"
    }
  ],
}
```



```

    "subscriptionLimit": {
        "isEnabled": false,
        "currentRate": 0
    },
    "containsCapacitySavingVolume": false
}

```

取得する属性については、プールの情報を取得する場合の属性に加えて、次に示す属性が取得されます。

属性	型	説明
numberOfDriveTypes	int	ドライブタイプの数
drives	object[]	<p>ドライブに関する情報が出力されます。</p> <ul style="list-style-type: none"> <li>• driveType (string) ドライブタイプ <ul style="list-style-type: none"> <li>- SSD</li> <li>- HDD (iStorage V110/V310 でのみ表示されます)</li> </ul> </li> <li>• driveInterface (string) ドライブインタフェース <ul style="list-style-type: none"> <li>- NVMe</li> <li>- SAS (iStorage V110/V310 でのみ表示されます)</li> </ul> </li> <li>• driveRpm (string) ドライブの回転数 (rpm) <ul style="list-style-type: none"> <li>- NUMBER_0</li> <li>- NUMBER_7200</li> <li>- NUMBER_10000</li> <li>- NUMBER_15000</li> <li>- Unknown</li> </ul> </li> <li>• driveCapacity (int) ドライブの容量 (GB)</li> <li>• displayDriveCapacity (string) ドライブの容量が単位付き (GB または TB) で出力されます。</li> <li>• totalCapacity (long) ドライブの総容量 (MiB)</li> <li>• numberOfDrives (int) プールに属しているドライブタイプのドライブ数</li> <li>• locations (string[]) ドライブの位置</li> <li>• raidLevel (string) RAID レベル <ul style="list-style-type: none"> <li>- RAID6</li> </ul> </li> <li>• dynamicDriveProtection (string) ADP の有効化状態 <ul style="list-style-type: none"> <li>- ENABLED : ADP が有効</li> <li>- DISABLED : ADP が無効※</li> <li>- MIXED : ADP の有効と無効が混在している※</li> </ul> </li> </ul>

**注※**

保守で使用する属性値です。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192
.0.2.100/ConfigurationManager/simple/v1/objects/pools/0
```

---

**関連リンク**

---

[HTTP ステータスコード \(13 ページ\)](#)

[プールの情報を取得する \(50 ページ\)](#)

---

### 4.2.3 プールを作成する

プール名とドライブ情報を指定して、プールを作成します。

暗号化の設定を有効にしたプールを作成したい場合、暗号化の仕様やシステム要件など、Encryption License Key の機能の詳細については、マニュアル『Encryption License Key ユーザガイド』を参照してください。

**⚠ 注意**

プールにスナップショットの差分データを格納する場合、プールの使用率が枯渇しきい値を超えると作成したスナップショットが使用できなくなるおそれがあります。枯渇しきい値を変更したい場合は、プールの作成後にプールの設定を変更してください。

---

## 実行権限

ストレージ管理者（プロビジョニング）

## リクエストライン

```
POST <ベース URL>/simple/v1/objects/pools
```

## リクエストメッセージ

### オブジェクト ID

なし。

### クエリパラメータ

なし。

### ボディ

プール名とドライブ情報を次のとおり指定して、暗号化のプールを作成する例を示します。

- プール名：NASOS
- プールの暗号化の設定：true
- ドライブタイプコード：SNR5A-R960NC
- データドライブの数：9
- RAID レベル：RAID6

```
{
  "name": "NASOS",
  "isEncryptionEnabled": true,
  "drives": [
    {
      "driveTypeCode": "SNB5B-R1R9NC",
      "dataDriveCount": 9,
      "raidLevel": "RAID6",
      "parityGroupType": "ADP"
    }
  ]
}
```

属性	型	説明
name	string	(必須) プール名を 1～32 文字で指定します。 使用できる文字は次のとおりです。 0～9 A～Z a～z , - . / : @ \ _ 半角スペース 英字は、大文字と小文字を区別します。 半角スペースは、使用できる文字と文字の間には使用できませんが、先頭と末尾には使用できません。 名前の先頭にハイフン (-) は使用できません。
isEncryptionEnabled	boolean	(任意) プールの暗号化の設定を有効にするかどうかを指定します。 <ul style="list-style-type: none"> <li>• false：無効にする</li> </ul> この属性を省略した場合、false が指定されたと見なされます。
drives	object[]	プールに使用するドライブ情報を指定します。 <ul style="list-style-type: none"> <li>• (必須) driveTypeCode (string)</li> </ul>

属性	型	説明
		<p>ドライブタイプコードを 12 文字で指定します。※1 例：SNB5B-R1R9NC</p> <ul style="list-style-type: none"> <li>• (必須) dataDriveCount (int) データドライブの数を 1～1440 までの整数で指定します。 ADP プールの場合、データドライブの数は最低でも 9 以上を指定してください。※2</li> <li>• (必須) raidLevel (string) RAID レベル 指定できる値は次のとおりです。 <ul style="list-style-type: none"> <li>- RAID6</li> </ul> </li> <li>• (任意) parityGroupType (string) プール作成時に指定するパリティグループの種別 <ul style="list-style-type: none"> <li>- ADP：ADP を有効にしたプールを作成する※3</li> <li>- DEFAULT：ADP が無効なプールを作成する</li> </ul> </li> </ul> <p>この属性は、RAID レベルが RAID6 の場合に ADP を指定できます。それ以外の場合、またはこの属性を省略したときには、DEFAULT が指定されたと見なされます。</p>

**注※1**

ドライブの条件（ドライブタイプ、ドライブインタフェース、ドライブの容量）が同じでも、ドライブタイプコードの異なるドライブがストレージシステム内に混在している場合があります。この場合、どのドライブタイプコードを指定してもドライブの条件に一致するドライブを使用してプールが作成されます。

作成されたプールに属するドライブの情報を確認するには、ドライブの情報を取得する API のクエリパラメータにプール ID を指定して実行してください。

**注※2**

ADP プールにおいて、ドライブの条件（ドライブタイプ、ドライブインタフェース、ドライブの容量）が同じドライブごとに、ドライブ数が 32 の倍数を超える場合は、32 の倍数+9 個以上を指定する必要があります。下記の例を参考に指定してください。

- 9～32 個（32×1 倍）

指定できます。

- 33～40 個

指定しないでください（32 の倍数を超えた場合、さらに 9 個以上のドライブが必要なため）。

- 41～64 個（32×2 倍）

指定できます。

- 65～72 個

指定しないでください（32 の倍数を超えた場合、さらに 9 個以上のドライブが必要のため）。

### 注※3

Advanced Dynamic Provisioning (ADP) は、ドライブ分散型の RAID 機能です。ADP を使用すると、スペア領域を各ドライブに分散させるためスペア専用ドライブが不要となります。また、リビルド負荷を全ドライブに分散させることで、リビルド時間が短縮します。

ADP の詳細については、マニュアル『システム構築ガイド』を参照してください。

## レスポンスメッセージ

### ボディ

属性	型	説明
statusResource	string	作成したプールの実行結果の問い合わせに使用する URL

### メモ

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H "Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/simple/v1/objects/pools
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

[ドライブの情報を取得する \(38 ページ\)](#)

[プールの設定を変更する \(62 ページ\)](#)

## 4.2.4 プールの設定を変更する

プール ID を指定して、プールの設定（プール名やしきい値など）を変更します。

### ⚠ 注意

プールにスナップショットの差分データを格納している場合、プールの使用率が枯渇しきい値を超えると作成したスナップショットが使用できなくなるおそれがあります。必要に応じて枯渇しきい値を変更してください。

## 実行権限

ストレージ管理者（プロビジョニング）

## リクエストライン

```
PATCH <ベース URL>/simple/v1/objects/pools/<オブジェクト ID>
```

## リクエストメッセージ

### オブジェクト ID

プールの情報取得で取得した id の値を指定します。

属性	型	説明
id	int	(必須) プール ID

### クエリパラメータ

なし。

### ボディ

指定した属性だけが変更されます。

プール ID が 63 のプールの、警告しきい値の設定値を 80 および枯渇しきい値の設定値を 90 に変更する例を次に示します。

```
{
  "thresholdWarning": 80,
  "thresholdDepletion": 90
}
```

属性	型	説明
name	string	(任意) プール名を 1~32 文字で指定します。 使用できる文字は次のとおりです。 0~9 A~Z a~z - . / : @ \ _ 英字は、大文字と小文字を区別します。

属性	型	説明
		半角スペースは、使用できる文字と文字の間には使用できますが、先頭と末尾には使用できません。 名前の先頭にハイフン (-) は使用できません。
thresholdWarning	int	(任意) プールの警告しきい値 (%) 1～100 までの整数で指定します。 この属性を指定する場合、thresholdDepletion も必ず指定してください。
thresholdDepletion	int	(任意) プールの枯渇しきい値 (%) 1～100 までの整数で指定します。 この属性を指定する場合、thresholdWarning も必ず指定してください。その場合、thresholdWarning で指定した値と同じ値、または大きい値を指定する必要があります。

## レスポンスメッセージ

### ボディ

属性	型	説明
affectedResources	string[]	変更したプールの情報を参照するための URL の一覧
operationDetails	object[]	変更したプールの詳細情報 詳細については、commandStatus オブジェクトの属性 operationDetails の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X PATCH --data-bi
nary @./InputParameters.json https://192.0.2.100/ConfigurationManager/simpl
e/v1/objects/pools/63
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[commandStatus オブジェクト \(19 ページ\)](#)

[特定のプールの情報を取得する \(55 ページ\)](#)

## 4.2.5 プールを削除する

プール ID を指定して、プールを削除します。

## 実行権限

ストレージ管理者（プロビジョニング）

## リクエストライン

```
DELETE <ベース URL>/simple/v1/objects/pools/<オブジェクト ID>
```

## リクエストメッセージ

### オブジェクト ID

プールの情報取得で取得した id の値を指定します。

属性	型	説明
id	int	(必須) プール ID

### クエリパラメータ

なし。

### ボディ

なし。

## レスポンスメッセージ

### ボディ

属性	型	説明
statusResource	string	削除したプールの実行結果の問い合わせに使用する URL

### メモ

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。



## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H "Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X DELETE https://192.0.2.100/ConfigurationManager/simple/v1/objects/pools/63
```

---

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

---

## 4.2.6 プールにドライブを追加する

プール ID を指定して、プールにドライブを追加することでプールの容量を拡張します。

parityGroupType に ADP を設定したドライブと DEFAULT を設定したドライブを同時に指定することはできません。

詳細は、マニュアル『システム構築ガイド』を参照してください。

HA Storage Manager Embedded のプール拡張時にドライブが ADP に割り当てられる動作については、マニュアル『システム構築ガイド』を参照してください。

### 注意

HA Storage Manager Embedded の API を使用してプールを拡張すると、RAID Manager を使用した場合に比べて拡張に使用できるドライブ数が少なくなる場合があります。マニュアル『システム構築ガイド』で、プール容量を拡張する際の注意事項をご確認の上、拡張されることをお勧めします。

---

### メモ

拡張中のプールにドライブを追加することはできません。

プールが拡張中かどうかは、プールの情報を取得する API で属性 totalCapacity と effectiveCapacity の値を確認してください。

---

## 実行権限

ストレージ管理者 (プロビジョニング)

## リクエストライン

```
POST <ベース URL>/simple/v1/objects/pools/<オブジェクト ID>/actions/expand/invoke
```

## リクエストメッセージ

### オブジェクト ID

プールの情報取得で取得した id の値を指定します。

属性	型	説明
id	int	(必須) プール ID

### クエリパラメータ

なし。

### ボディ

プール ID が 63 のプールに、ドライブタイプコードが SLB5F-M480SS のドライブのデータドライブを 9 台追加する例を示します。

```
{
  "additionalDrives": [
    {
      "driveTypeCode": "SLB5F-M480SS",
      "dataDriveCount": 9,
      "raidLevel": "RAID6",
      "parityGroupType": "ADP"
    }
  ]
}
```

属性	型	説明
additionalDrives	object[]	<p>プールに追加するドライブの情報</p> <ul style="list-style-type: none"> <li>(必須) driveTypeCode (string) ドライブタイプコードを 12 文字で指定します。※1 例: SNB5B-R1R9NC</li> <li>(必須) dataDriveCount (int) データドライブの数を 1~1440 までの整数で指定します。※2 非 ADP プールにおいて、RAID レベルに RAID6 を指定する場合、データドライブの数は最低でも 8 以上を指定してください。</li> <li>(必須) raidLevel (string) 次に示す RAID レベルを指定します。 <ul style="list-style-type: none"> <li>RAID6</li> </ul> </li> <li>(任意) parityGroupType (string) プール拡張時に指定するパリティグループの種別 <ul style="list-style-type: none"> <li>ADP: プール拡張時に ADP を有効にする※3</li> <li>DEFAULT: プール拡張時に ADP を有効にしない</li> </ul> </li> </ul> <p>この属性は、RAID レベルが RAID6 の場合に指定できます。この属性を省略したときには、DEFAULT が指定されたと見なされます。</p>

**注※1**

- ドライブの条件（ドライブタイプ、ドライブインタフェース、ドライブの容量）が同じでも、ドライブタイプコードの異なるドライブがストレージシステム内に混在している場合があります。この場合、どのドライブタイプコードを指定してもドライブの条件に一致するドライブが追加されてプールが拡張されます。

プールに追加されたドライブの情報を確認するには、ドライブの情報を取得する API のクエリパラメータにプール ID を指定して実行してください。

- 拡張するプールにスナップショットが作成されている場合、条件が異なるドライブを指定できません。

**注※2**

ADP プールにおいて、ドライブの条件（ドライブタイプ、ドライブインタフェース、ドライブの容量）が同じドライブごとに、拡張後のドライブ数が 32 の倍数を超える場合には、32 の倍数+9 個以上となるように指定する必要があります。下記の例を参考に指定してください。

- 拡張後のドライブ数：～32 個（32×1 倍）  
指定できます。
- 拡張後のドライブ数：33～40 個  
指定しないでください（32 の倍数を超えた場合、さらに 9 個以上のドライブが必要なため）。
- 拡張後のドライブ数：41～64 個（32×2 倍）  
指定できます。
- 拡張後のドライブ数：65～72 個  
指定しないでください（32 の倍数を超えた場合、さらに 9 個以上のドライブが必要なため）。

**注※3**

Advanced Dynamic Provisioning (ADP) は、ドライブ分散型の RAID 機能です。ADP を使用すると、スペア領域を各ドライブに分散させるためスペア専用ドライブが不要となります。また、リビルド負荷を全ドライブに分散させることで、リビルド時間が短縮します。

## レスポンスメッセージ

### ボディ

属性	型	説明
statusResource	string	ドライブを追加したプールの実行結果の問い合わせに使用する URL

### メモ

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H "Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/simple/v1/objects/pools/63/actions/expand/invoke
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

[ドライブの情報を取得する \(38 ページ\)](#)

[特定のプールの情報を取得する \(55 ページ\)](#)

## 4.3 ポートの管理

REST API を使用してポートに関する情報を一覧で取得したり、特定のポートの情報を取得します。また、ポートに関する設定を変更する操作について説明します。

### 4.3.1 ポートの情報を取得する

ポートの情報を一覧で取得します。

## 実行権限

ストレージ管理者（参照）

## リクエストライン

```
GET <ベース URL>/simple/v1/objects/ports
```

## リクエストメッセージ

### オブジェクト ID

なし。

### クエリパラメータ

パラメータ	型	フィルタ条件
protocol	string	(任意) 取得するプロトコルとして、次の値を指定します。 <ul style="list-style-type: none"> <li>• FC</li> <li>• iSCSI</li> </ul>

### ボディ

なし。

## レスポンスメッセージ

### ボディ

すべてのポートの情報を取得した場合の出力例を次に示します。

```
{
  "data": [
    {
      "id": "CL1-A",
      "protocol": "FC",
      "portWwn": "50060e8023000100",
      "portSpeed": "NUMBER_8",
      "actualPortSpeed": "8G",
      "portSecurity": true,
      "fcInformation": {
        "alPa": "EF",
        "fabricSwitchSetting": false,
        "connectionType": "FC_AL",
        "sfpDataTransferRate": "NUMBER_16",
        "portMode": "SCSI"
      }
    }
  ],
  {
```

```

        "id": "CL2-B",
        "protocol": "iSCSI",
        "portIscsiName": "iqn.2001-03.jp.nec:storage01.h9s.t.30001.2b",
        "portSpeed": "NUMBER_10",
        "actualPortSpeed": "10G",
        "portSecurity": true,
        "iscsiInformation": {
            "vlanUse": false,
            "ipMode": "ipv4v6",
            "ipv4Information": {
                "address": "192.168.0.188",
                "subnetMask": "255.255.255.0",
                "defaultGateway": "0.0.0.0"
            },
            "ipv6Information": {
                "linklocal": "Auto",
                "linklocalAddress": "fe80::21f:67ff:feaf:c26d",
                "linklocalAddressStatus": "VALID",
                "global": "Manual",
                "globalAddress": "fe84::1111",
                "defaultGateway": "::"
            },
            "isIpv6Updating": false,
            "tcpPort": 3260,
            "selectiveAck": true,
            "delayedAck": true,
            "windowSize": "NUMBER_64K",
            "mtuSize": "NUMBER_1500",
            "linkMtuSize": "NUMBER_1500",
            "keepAliveTimer": 60,
            "isnsServerMode": false,
            "isnsServerIpAddress": "0.0.0.0",
            "isnsServerPort": 3205,
            "virtualPortEnabled": false
        }
    }

    ],
    "count": 3
}

```

プロトコルが iSCSI（クエリパラメータ protocol に iSCSI を指定）のポートの情報を取得した出力例を次に示します。

```

{
  "data": [
    {
      "id": "CL2-B",
      "protocol": "iSCSI",
      "portIscsiName": "iqn.2001-03.jp.nec:storage01.h9s.t.30001.2b",
      "portSpeed": "NUMBER_10",
      "actualPortSpeed": "10G",
      "portSecurity": true,
      "iscsiInformation": {
        "vlanUse": false,
        "ipMode": "ipv4v6",
        "ipv4Information": {

```

```

        "address": "10.1.1.52",
        "subnetMask": "255.255.0.0",
        "defaultGateway": "0.0.0.0"
    },
    "ipv6Information": {
        "linklocal": "Auto",
        "linklocalAddress": "fe80::21f:67ff:feaf:c278",
        "linklocalAddressStatus": "VALID",
        "global": "Auto",
        "globalAddress": "::",
        "globalAddressStatus": "INVALID",
        "defaultGateway": "::"
    },
    "isIpv6Updating": false,
    "tcpPort": 3260,
    "selectiveAck": true,
    "delayedAck": true,
    "windowSize": "NUMBER_64K",
    "mtuSize": "NUMBER_9000",
    "linkMtuSize": "NUMBER_9000",
    "keepAliveTimer": 60,
    "isnsServerMode": false,
    "isnsServerIpAddress": "0.0.0.0",
    "isnsServerPort": 3205,
    "virtualPortEnabled": false
    }
}
],
"count": 1
}

```

属性	型	説明
id	string	ポート ID
protocol	string	プロトコル 次に示す値が出力されます。 <ul style="list-style-type: none"> <li>FC</li> <li>iSCSI</li> </ul>
portWwn	string	ポートの WWN プロトコルが FC の場合に出力されます。
portIscsiName	string	ポートの iSCSI ネーム プロトコルが iSCSI の場合に出力されます。
portSpeed	string	ポートのデータ転送速度 <ul style="list-style-type: none"> <li>NUMBER_0 : 自動 (Auto)</li> <li>NUMBER_8 : 8Gbps</li> <li>NUMBER_10 : 10Gbps</li> <li>NUMBER_16 : 16Gbps</li> <li>NUMBER_25 : 25Gbps</li> <li>NUMBER_32 : 32Gbps</li> <li>NUMBER_64 : 64Gbps</li> <li>NUMBER_100 : 100Gbps</li> </ul>
actualPortSpeed	string	ポートの実際に動作しているデータ転送速度 <ul style="list-style-type: none"> <li>LINK_DOWN : リンクダウンまたはリンクアップ処理中</li> </ul>

属性	型	説明
		<ul style="list-style-type: none"> <li>• 8G : 8Gbps</li> <li>• 10G : 10Gbps</li> <li>• 16G : 16Gbps</li> <li>• 25G : 25Gbps</li> <li>• 32G : 32Gbps</li> <li>• 64G : 64Gbps</li> <li>• 100G : 100Gbps</li> </ul>
portSecurity	boolean	ポートのセキュリティの設定が有効かどうか出力されます。 <ul style="list-style-type: none"> <li>• true : 有効</li> <li>• false : 無効</li> </ul>
fcInformation	object	プロトコルが FC の場合に、FC に関する情報が出力されます。 <ul style="list-style-type: none"> <li>• alPa (string) ポートのアドレス (AL-PA)</li> <li>• fabricSwitchSetting (boolean) Fabric スイッチの設定の有効/無効               <ul style="list-style-type: none"> <li>- true : 有効</li> <li>- false : 無効</li> </ul> </li> <li>• connectionType (string) トポロジ設定               <ul style="list-style-type: none"> <li>- Point_To_Point</li> <li>- FC_AL</li> </ul> </li> <li>• sfpDataTransferRate (string) 転送レート               <ul style="list-style-type: none"> <li>- NUMBER_16 : 16Gbps</li> <li>- NUMBER_32 : 32Gbps</li> <li>- NUMBER_64 : 64Gbps</li> </ul> </li> <li>• portMode (string) ポートモード ポートの動作モードを示します。               <ul style="list-style-type: none"> <li>- SCSI</li> </ul> </li> </ul>
iscsiInformation	object	プロトコルが iSCSI の場合に、iSCSI に関する情報が出力されます。 <ul style="list-style-type: none"> <li>• vlanUse (boolean) VLAN の有効/無効               <ul style="list-style-type: none"> <li>- true : 有効</li> <li>- false : 無効</li> </ul> </li> <li>• vlanId (int) VLAN ID VLAN の設定が有効な場合に出力されます。</li> <li>• ipMode (string) IP アドレスの形式               <ul style="list-style-type: none"> <li>- ipv4</li> <li>- ipv4v6</li> </ul> </li> <li>• ipv4Information (object)</li> </ul>



属性	型	説明
		<p>IPv4 に対応する情報</p> <ul style="list-style-type: none"> <li>- address (string) IP アドレス</li> <li>- subnetMask (string) サブネットマスク</li> <li>- defaultGateway (string) デフォルトゲートウェイのアドレス</li> </ul> <p>• ipv6Information (object)</p> <p>IPv6 に対応する情報</p> <ul style="list-style-type: none"> <li>- linklocal (string) リンクローカルアドレスの設定方法 <ul style="list-style-type: none"> <li>• Auto</li> <li>• Manual</li> </ul> </li> <li>- linklocalAddress (string) リンクローカルアドレス</li> <li>- linklocalAddressStatus (string) リンクローカルアドレスの状態 <ul style="list-style-type: none"> <li>• INVALID : 無効</li> <li>• VALID : 有効</li> <li>• ACQUIRING : 取得中</li> <li>• DUPLICATED : 重複</li> </ul> <p>属性 linklocal が Auto の場合に出力されます。</p> </li> <li>- global (string) グローバルアドレスの設定方法 <ul style="list-style-type: none"> <li>• Auto</li> <li>• Manual</li> </ul> </li> <li>- globalAddress (string) グローバルアドレス</li> <li>- globalAddressStatus (string) グローバルアドレスの状態 <ul style="list-style-type: none"> <li>• INVALID : 無効</li> <li>• VALID : 有効</li> <li>• ACQUIRING : 取得中</li> <li>• DUPLICATED : 重複</li> </ul> <p>属性 global が Auto の場合に出力されます。</p> </li> <li>- defaultGateway (string) ゲートウェイアドレス</li> </ul> <p>• isIpv6Updating (boolean)</p> <p>IPv6 に対応する情報の設定処理が処理中かどうか出力されます。</p> <ul style="list-style-type: none"> <li>- true : 処理中 true (処理中) の場合、次に示す属性には設定前の値が出力されます。 <ul style="list-style-type: none"> <li>• linklocalAddress</li> <li>• linklocalAddressStatus</li> <li>• globalAddress</li> </ul> </li> </ul>

属性	型	説明
		<ul style="list-style-type: none"> <li>• globalAddressStatus</li> <li>• defaultGateway</li> <li>• linkMtuSize</li> <li>- false: 処理中でない</li> <li>• tcpPort (int) iSCSI 通信時の TCP ポート番号</li> <li>• selectiveAck (boolean) 選択型 ACK の有効/無効 <ul style="list-style-type: none"> <li>- true: 有効</li> <li>- false: 無効</li> </ul> </li> <li>• delayedAck (boolean) 遅延 ACK の有効/無効 <ul style="list-style-type: none"> <li>- true: 有効</li> <li>- false: 無効</li> </ul> </li> <li>• windowSize (string) Window サイズの値</li> <li>• mtuSize (string) MTU サイズの値</li> <li>• linkMtuSize (string) リンク MTU サイズの値</li> <li>• keepAliveTimer (int) iSCSI 通信時の Keep Alive タイマーの設定値 (秒)</li> <li>• isnsServerMode (boolean) iSNS サーバモードの有効/無効 <ul style="list-style-type: none"> <li>- true: 有効</li> <li>- false: 無効</li> </ul> </li> <li>• isnsServerIpAddress (string) iSNS サーバの IP アドレス (IPv4 または IPv6) iSNS サーバモードの設定が有効な場合に有効な IP アドレスの値が出力されます。 仮想ポートが有効な場合、この属性は出力されません。</li> <li>• isnsServerPort (int) iSNS サーバの TCP ポート番号 iSNS サーバモードの設定が有効な場合に有効な TCP ポート番号の値が出力されます。 仮想ポートが有効な場合、この属性は出力されません。</li> <li>• virtualPortEnabled (boolean) 仮想ポートの有効/無効 <ul style="list-style-type: none"> <li>- true: 有効 仮想ポートが true (有効) の場合、属性 iscsiInformation には、仮想ポート番号が 0 の情報が出力されます。</li> <li>- false: 無効</li> </ul> </li> </ul>

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H
"Content-Type:application/json" -H "Authorization:Session
b74777a3f9f04ea8bd8f09847fac48d3" -X GET
https://192.0.2.100/ConfigurationManager/simple/v1/objects/ports
```

---

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[データオブジェクト \(17 ページ\)](#)

---

### 4.3.2 特定のポートの情報を取得する

ポート ID を指定して、特定のポートに関する情報を取得します。

## 実行権限

ストレージ管理者 (参照)

## リクエストライン

```
GET <ベース URL>/simple/v1/objects/ports/<オブジェクト ID>
```

## リクエストメッセージ

### オブジェクト ID

ポートの情報取得で取得した id の値を指定します。

属性	型	説明
id	int	(必須) ポート ID

### クエリパラメータ

なし。

### ボディ

なし。

## レスポンスメッセージ

### ボディ

FC 接続のポート（ポート ID：CL1-A）の情報を取得した出力例を次に示します。

```
{
  "id": "CL1-A",
  "protocol": "FC",
  "portWwn": "50060e8023000100",
  "portSpeed": "NUMBER_8",
  "actualPortSpeed": "8G",
  "portSecurity": true,
  "fcInformation": {
    "alPa": "EF",
    "fabricSwitchSetting": false,
    "connectionType": "FC_AL",
    "sfpDataTransferRate": "NUMBER_16",
    "portMode": "SCSI"
  }
}
```

iSCSI 接続のポート（ポート ID：CL2-B）の情報を取得した出力例を次に示します。

```
{
  "id": "CL2-B",
  "protocol": "iSCSI",
  "portIscsiName": "iqn.2001-03.jp.nec:storage01.h9s.t.20160.2b",
  "portSpeed": "NUMBER_10",
  "actualPortSpeed": "10G",
  "portSecurity": true,
  "iscsiInformation": {
    "vlanUse": false,
    "ipMode": "ipv4v6",
    "ipv4Information": {
      "address": "10.1.1.52",
      "subnetMask": "255.255.0.0",
      "defaultGateway": "0.0.0.0"
    },
    "ipv6Information": {
      "linklocal": "Auto",
      "linklocalAddress": "fe80::21f:67ff:feaf:c278",
      "linklocalAddressStatus": "VALID",
      "global": "Auto",
      "globalAddress": ":::",
      "globalAddressStatus": "INVALID",
      "defaultGateway": ":::"
    },
    "isIpv6Updating": false,
    "tcpPort": 3260,
    "selectiveAck": true,
    "delayedAck": true,
    "windowSize": "NUMBER_64K",
    "mtuSize": "NUMBER_9000",
    "linkMtuSize": "NUMBER_9000",
    "keepAliveTimer": 60,
    "isnsServerMode": false,
  }
}
```

```

    "isnsServerIpAddress": "0.0.0.0",
    "isnsServerPort": 3205,
    "virtualPortEnabled": false
  }
}

```

取得する属性については、ポートの情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```

curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192
.0.2.100/ConfigurationManager/simple/v1/objects/ports/CL1-A

```

---

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[ポートの情報を取得する \(68 ページ\)](#)

---

### 4.3.3 ポートの設定を変更する

ポート ID を指定して、ポートの設定に関する値を変更します。

## 実行権限

ストレージ管理者（プロビジョニング）

## リクエストライン

```
PATCH <ベース URL>/simple/v1/objects/ports/<オブジェクト ID>
```

## リクエストメッセージ

### オブジェクト ID

ポートの情報取得で取得した id の値を指定します。

属性	型	説明
id	int	(必須) ポート ID

## クエリパラメータ

なし。

## ボディ

指定した属性だけが変更されます。

FC 接続のポート（ポート ID：CL1-A）の設定を次の様に変更する例を示します。

- Fabric スイッチの設定を true に変更
- トポロジを Point\_To\_Point に変更

```
{
  "fcInformation": {
    "fabricSwitchSetting": true,
    "connectionType": "Point_To_Point"
  }
}
```

iSCSI 接続のポート（ポート ID：CL1-B）の設定を次の様に変更する例を示します。

- IPv6 のリンクローカルアドレスの設定を Auto に変更
- IPv6 のグローバルアドレスの設定 Auto に変更
- 選択型 ACK の設定を true に変更
- MTU サイズの値を NUMBER\_4500 に変更

```
{
  "iscsiInformation": {
    "ipv6Information": {
      "linklocal": "Auto",
      "global": "Auto"
    },
    "selectiveAck": true,
    "mtuSize": "NUMBER_4500"
  }
}
```

属性	型	説明
portSpeed	string	<p>（任意）ポートのデータ転送速度 指定できる値は次のとおりです。</p> <ul style="list-style-type: none"> <li>• NUMBER_0：自動</li> <li>• NUMBER_8：8Gbps</li> <li>• NUMBER_10：10Gbps</li> <li>• NUMBER_16：16Gbps</li> <li>• NUMBER_25：25Gbps</li> <li>• NUMBER_32：32Gbps</li> <li>• NUMBER_64：64Gbps</li> <li>• NUMBER_100：100Gbps</li> </ul>

属性	型	説明
		<p>データ転送速度と FC の設定の組み合わせの詳細については、マニュアル『システム構築ガイド』を参照してください。</p> <p>チャンネルボード (10Gbps iSCSI(Optical)) のポートの場合、[ポートスピード]には NUMBER_10 (10Gbps) を設定します。そのほかの値を指定しても無視されます。</p>
portSecurity	boolean	<p>(任意) ポートのセキュリティの設定を有効にするかどうかを指定します。</p> <ul style="list-style-type: none"> <li>• true : 有効</li> <li>• false : 無効</li> </ul>
fcInformation	object	<p>FC に関する設定を変更します。プロトコルが FC の場合にだけ指定できます。</p> <ul style="list-style-type: none"> <li>• (任意) alPa (string) ポートのアドレス (AL-PA) を 2 桁の 16 進数 (01～EF) で指定します。</li> <li>• (任意) fabricSwitchSetting (boolean) Fabric スイッチの設定を有効にするかどうかを指定します。 <ul style="list-style-type: none"> <li>- true : 有効</li> <li>- false : 無効</li> </ul> </li> <li>• (任意) connectionType (string) トポロジの設定を指定します。 <ul style="list-style-type: none"> <li>- Point_To_Point</li> <li>- FC_AL</li> </ul> </li> </ul>
iscsiInformation	object	<p>iSCSI に関する設定を変更します。プロトコルが iSCSI の場合にだけ指定できます。</p> <ul style="list-style-type: none"> <li>• (任意) vlanUse (boolean) VLAN の設定を有効にするかどうかを指定します。 <ul style="list-style-type: none"> <li>- true : 有効</li> <li>- false : 無効</li> </ul> </li> <li>• (任意) addVlanId (int) 追加する VLAN ID を 1～4094 までの整数で指定します。削除する VLAN ID とは異なる値を指定してください。</li> <li>• (任意) deleteVlanId (int) 削除する VLAN ID を 1～4094 までの整数で指定します。</li> <li>• (任意) ipMode (string) IP アドレスの形式を指定します。 <ul style="list-style-type: none"> <li>- ipv4 : IPv4 を使用する</li> <li>- ipv4v6 : IPv4 および IPv6 を使用する</li> </ul> </li> <li>• ipv4Information (object) IPv4 に対応する情報を設定します。 <ul style="list-style-type: none"> <li>- (任意) address (string) ※1 IPv4 形式の IP アドレスを 7～15 文字で指定します。</li> <li>- (任意) subnetMask (string) サブネットマスクを 7～15 文字で指定します。</li> </ul> </li> </ul>

属性	型	説明
		<ul style="list-style-type: none"> <li>- (任意) defaultGateway (string) デフォルトゲートウェイのアドレスを 7～15 文字で指定します。</li> <li>• ipv6Information (object) IPv6 に対応する情報を設定します。 <ul style="list-style-type: none"> <li>- (任意) linklocal (string) ※2 Auto (自動)、またはリンクローカルアドレスを 2～45 文字で指定します。</li> <li>- (任意) global (string) ※2 Auto (自動)、またはグローバルアドレスを 2～45 文字で指定します。</li> <li>- (任意) defaultGateway (string) ※2 デフォルトゲートウェイのアドレスを 2～45 文字で指定します。 チャンネルボード (25Gbps iSCSI (Optical)) のポートの場合で、Router Advertisement(RA)を使用したゲートウェイアドレスを利用するときは、この属性は指定しないでください。 同一ネットワーク内に複数の IPv6 のゲートウェイアドレスがある場合、性能遅延を引き起こすおそれがあります。</li> </ul> </li> <li>• (任意) tcpPort (int) iSCSI 通信時の TCP ポート番号を 1～65535 までの整数で指定します。</li> <li>• (任意) selectiveAck (boolean) 選択型 ACK を有効、または無効に設定します。 <ul style="list-style-type: none"> <li>- true : 有効</li> <li>- false : 無効</li> </ul> チャンネルボード (25Gbps iSCSI (Optical)) のポートの場合、無効に設定できません。 </li> <li>• (任意) delayedAck (boolean) 遅延 ACK を有効、または無効に設定します。 <ul style="list-style-type: none"> <li>- true : 有効</li> <li>- false : 無効</li> </ul> </li> <li>• (任意) windowSize (string) Window サイズの値 指定できる値は、NUMBER_64K、NUMBER_128K、NUMBER_256K、NUMBER_512K、または NUMBER_1024K です。</li> <li>• (任意) mtuSize (string) MTU サイズの値 指定できる値は、NUMBER_1500、NUMBER_4500、または NUMBER_9000 です。</li> <li>• (任意) keepAliveTimer (int) iSCSI 通信時の Keep Alive タイマーの設定値 (秒) を 30～64800 までの整数で指定します。</li> <li>• (任意) isnsServerMode (boolean) iSNS サーバモードを有効、または無効に設定します。 <ul style="list-style-type: none"> <li>- true : 有効</li> </ul> </li> </ul>



属性	型	説明
		<ul style="list-style-type: none"> <li>- false : 無効</li> <li>• (任意) isnsServerIpAddress (string) iSNS サーバの IP アドレスを IPv4 形式、または IPv6 形式で 2~45 文字で指定します。</li> <li>• (任意) isnsServerPort (int) iSNS サーバの TCP ポート番号を 1~65535 までの整数で指定します。</li> </ul>

**注※1**

この属性を指定する場合、次のアドレスは指定できません。

- ネットワークアドレス (例 : 192.168.10.0)
- ブロードキャストアドレス (例 : 255.255.255.255)
- ループバックアドレス (例 : 127.0.0.1)

**注※2**

この属性を指定する場合、次のアドレスは指定できません。

- アドレス未指定 (例 : ::)
- マルチキャストアドレス (例 : ff00:1024:1215::01)
- ループバックアドレス (例 : ::1)

## レスポンスメッセージ

**ボディ**

属性	型	説明
affectedResources	string[]	変更したポートの情報を参照するための URL の一覧
operationDetails	object[]	変更したポートの詳細情報 詳細については、commandStatus オブジェクトの属性 operationDetails の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X PATCH --data-bi
```

```
nary @./InputParameters.json https://192.0.2.100/ConfigurationManager/simple/v1/objects/ports/CL1-A
```

---

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[commandStatus オブジェクト \(19 ページ\)](#)

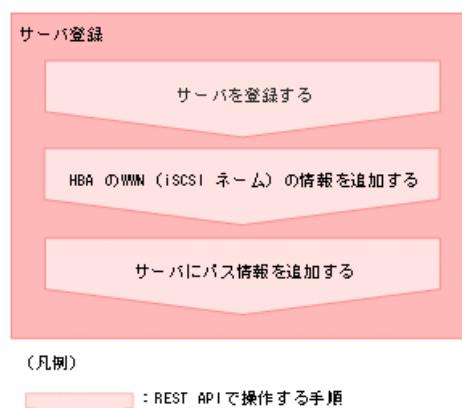
---

## 4.4 サーバの管理

REST API を使用してサーバに関する情報を取得したり、特定のサーバに関する詳細な情報を取得します。ストレージシステムにサーバの情報を登録、サーバとポート間のパス情報を追加、HBA の WWN (iSCSI ネーム) を追加する操作などについて説明します。

### サーバを登録する流れ

ストレージシステムにサーバの情報を登録します。また、サーバのニックネームだけを指定して、ホストグループを追加するためのサーバを作成することもできます。



- サーバを登録する

ストレージシステムにサーバの情報を登録します。

- HBA の WWN (iSCSI ネーム) の情報を追加する

サーバ ID を指定して、サーバに HBA の WWN (または iSCSI ネーム) の情報を追加します。

- サーバにパス情報を追加する

サーバ ID を指定して、特定のサーバにパス情報を追加します。

各種管理リソースの情報を取得したり、設定変更をしたりできます。

- スナップショットグループ名の一覧を取得する
- サーバの情報を取得する
- 特定のサーバの情報を取得する

- サーバの設定を変更する
- サーバを削除する
- サーバにホストグループ (iSCSI ターゲット) を追加する
- ホストグループ名 (iSCSI ターゲット名) をサーバのニックネームと同期する
- HBA の WWN (iSCSI ネーム) の情報を取得する
- 特定のサーバの HBA の WWN (iSCSI ネーム) の情報を取得する
- サーバから HBA の WWN (iSCSI ネーム) の情報を削除する
- サーバのパス情報の一覧を取得する
- サーバと特定のポート間のパス情報を取得する
- サーバからパス情報を削除する
- iSCSI ターゲットの情報を取得する
- 特定の iSCSI ターゲットの情報を取得する
- iSCSI ターゲットの設定を変更する

#### 4.4.1 サーバの情報を取得する

HA Storage Manager Embedded で管理しているサーバに関する情報を取得します。条件を指定して、絞り込んだ情報を取得したりできます。

### 実行権限

ストレージ管理者 (参照)

### リクエストライン

```
GET <ベース URL>/simple/v1/objects/servers
```

### リクエストメッセージ

#### オブジェクト ID

なし。

#### クエリパラメータ

パラメータ	型	フィルタ条件
nickname	string	(任意) サーバのニックネーム 指定した値と完全に一致するサーバの情報が取得され ます。

パラメータ	型	フィルタ条件
hbaWwn	string	(任意) HBA の WWN 指定した値と完全に一致するサーバの情報が取得されます。
iscsiName	string	(任意) iSCSI ネーム 指定した値と完全に一致するサーバの情報が取得されます。

## ボディ

なし。

## レスポンスメッセージ

### ボディ

サーバに関する情報を取得した出力例を次に示します。

```
{
  "data": [
    {
      "id": 10,
      "nickname": "hostA",
      "protocol": "FC",
      "osType": "Linux",
      "totalCapacity": 1024,
      "usedCapacity": 42,
      "numberOfPaths": 2,
      "isInconsistent": false,
      "modificationInProgress": false,
      "isReserved": false,
      "hasUnalignedOsTypes": false
    },
    {
      "id": 11,
      "nickname": "hostB",
      "protocol": "iSCSI",
      "osType": "Linux",
      "totalCapacity": 1024,
      "usedCapacity": 42,
      "numberOfPaths": 2,
      "isInconsistent": false,
      "modificationInProgress": false,
      "isReserved": false,
      "hasUnalignedOsTypes": false
    }
  ],
  "count": 2
}
```

ニックネームが「hostC」のサーバ（クエリパラメータ nickname に hostC を指定）に関する情報を取得した出力例を次に示します。

この例では属性 `isReserved` に `true` が出力されているので、ホストグループを追加するサーバであることを示しています。

```
{
  "data": [
    {
      "id": 8,
      "nickname": "hostC",
      "protocol": "Undefined",
      "osType": "Undefined",
      "totalCapacity": 0,
      "usedCapacity": 0,
      "numberOfPaths": 0,
      "isInconsistent": false,
      "modificationInProgress": false,
      "isReserved": true,
      "hasUnalignedOsTypes": false
    }
  ],
  "count": 1
}
```

属性	型	説明
id	int	サーバ ID
nickname	string	サーバのニックネーム
protocol	string	プロトコル 次に示す値が出力されます。 <ul style="list-style-type: none"> <li>FC</li> <li>iSCSI</li> <li>Undefined</li> </ul> ホストグループを追加するサーバの場合に出力されます。
osType	string	OS タイプ 次に示す値が出力されます。 <ul style="list-style-type: none"> <li>Linux</li> <li>DeprecatedVMware</li> <li>DeprecatedWindows</li> <li>VMware</li> <li>Windows</li> <li>Undefined</li> </ul> ホストグループを追加するサーバの場合に出力されます。 <ul style="list-style-type: none"> <li>Unknown</li> </ul>
totalCapacity	long	割り当てられているボリュームの総容量 (MiB)
usedCapacity	long	割り当てられているボリュームの使用済み容量 (MiB)
numberOfPaths	int	サーバに登録されている HBA の数
isInconsistent	boolean	サーバの構成情報が矛盾しているかどうか出力されます。 <ul style="list-style-type: none"> <li>true: 矛盾している</li> </ul>

属性	型	説明
		<ul style="list-style-type: none"> <li>• false : 矛盾していない</li> </ul>
modificationInProgress	boolean	現在、使用しない属性です。
isReserved	boolean	ホストグループを追加するサーバかどうか出力されます。 <ul style="list-style-type: none"> <li>• true : ホストグループを追加するサーバである</li> <li>• false : ホストグループを追加するサーバでない</li> </ul>
hasUnalignedOsTypes	boolean	OS タイプが定義されているサーバに、この値と異なるホストモードが定義されているホストグループ（または iSCSI ターゲット）を追加した場合に、サーバ側の情報が混在しているかどうか出力されます。 <ul style="list-style-type: none"> <li>• true : 混在している</li> <li>• false : 混在していない</li> </ul>

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192
.0.2.100/ConfigurationManager/simple/v1/objects/servers
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[データオブジェクト \(17 ページ\)](#)

## 4.4.2 特定のサーバの情報を取得する

サーバ ID を指定して、特定のサーバの情報を取得します。

## 実行権限

ストレージ管理者（参照）

## リクエストライン

```
GET <ベース URL>/simple/v1/objects/servers/<オブジェクト ID>
```

## リクエストメッセージ

### オブジェクト ID

サーバの情報取得で取得した id の値を指定します。

属性	型	説明
id	int	(必須) サーバ ID

### クエリパラメータ

なし。

### ボディ

なし。

## レスポンスメッセージ

### ボディ

FC 接続のサーバ（サーバ ID : 10）の情報を取得した出力例を次に示します。

```
{
  "id": 10,
  "nickname": "hostA",
  "protocol": "FC",
  "osType": "Linux",
  "osTypeOptions": [2, 22, 25, 68],
  "totalCapacity": 1024,
  "usedCapacity": 42,
  "numberOfVolumes": 2,
  "numberOfPaths": 2,
  "paths": [
    {
      "hbaWwn": "0000000102ccec9",
      "portIds": [
        "CL1-A"
      ]
    },
    {
      "hbaWwn": "1111111111111111",
      "portIds": [
        "CL1-A"
      ]
    }
  ],
  "isInconsistent": false,
  "modificationInProgress": false,
  "isReserved": false,
  "hasNonFullmeshLuPaths": false,
  "hasUnalignedOsTypes": false,
}
```

```
"hasUnalignedOsTypeOptions": false
}
```

iSCSI 接続のサーバ（サーバ ID：11）の情報を取得した出力例を次に示します。

```
{
  "id": 11,
  "nickname": "hostB",
  "protocol": "iSCSI",
  "osType": "Linux",
  "osTypeOptions": [2,22,25,68],
  "totalCapacity": 1024,
  "usedCapacity": 42,
  "numberOfPaths": 2,
  "paths": [
    {
      "iscsiName": "iqn.1994-05.com.redhat:496799ba93",
      "portIds": [
        "CL1-B"
      ]
    },
    {
      "iscsiName": "iqn.1994-05.com.redhat:a7526e46aa",
      "portIds": [
        "CL1-B"
      ]
    }
  ],
  "isInconsistent": false,
  "modificationInProgress": false,
  "isReserved": false,
  "hasNonFullmeshLuPaths": false,
  "hasUnalignedOsTypes": false,
  "hasUnalignedOsTypeOptions": false
}
```

取得する属性については、サーバの情報を取得する場合の属性に加えて、次に示す属性が取得されます。

属性	型	説明
osTypeOptions	int[]	OS タイプのオプション
numberOfVolumes	int	割り当てられているボリュームの数
paths	object[]	HBA の WWN に定義されているパスの情報 <ul style="list-style-type: none"> <li>• hbaWwn (string) HBA の WWN プロトコルが FC の場合に出力されます。</li> <li>• iscsiName (string) iSCSI ネーム プロトコルが iSCSI の場合に出力されます。</li> <li>• portIds (string[]) 割り当て先のポート ID</li> </ul>
hasNonFullmeshLuPaths	boolean	指定したサーバ内において登録されているすべてのポートと、サーバに接続されているすべてのボリューム間で、



属性	型	説明
		<p>パスが定義されていないボリュームがあるかどうか出力されます。</p> <ul style="list-style-type: none"> <li>• true: パスが定義されていないボリュームが1つでもある</li> <li>• false: すべてのボリュームにパスが定義されている</li> </ul>
hasUnalignedOsTypeOptions	boolean	<p>OS タイプのオプションが定義されているサーバに、この値と異なるホストモードオプションが定義されているホストグループ（または iSCSI ターゲット）を登録した場合に、サーバ側の情報が混在しているかどうか出力されます。</p> <ul style="list-style-type: none"> <li>• true: 混在している</li> <li>• false: 混在していない</li> </ul> <p>詳細は、マニュアル『システム構築ガイド』を参照してください。OS タイプのオプションは、RAID Manager ではホストモードオプションに対応します。</p>

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192
.0.2.100/ConfigurationManager/simple/v1/objects/servers/10
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)  
[サーバの情報を取得する \(83 ページ\)](#)

## 4.4.3 サーバを登録する

ストレージシステムに、HA Storage Manager Embedded で管理するためのサーバの情報を登録します。また、サーバのニックネームだけを指定して、ホストグループを追加するためのサーバを作成することもできます。

## 実行権限

ストレージ管理者（プロビジョニング）

## リクエストライン

```
POST <ベース URL>/simple/v1/objects/servers
```

## リクエストメッセージ

### オブジェクト ID

なし。

### クエリパラメータ

なし。

### ボディ

FC 接続のサーバの情報を次のとおり指定してストレージシステムに登録した例を次に示します。

- サーバのニックネーム：hostA
- プロトコル：FC
- OS タイプ：Linux

```
{
  "serverNickname": "hostA",
  "protocol": "FC",
  "osType": "Linux"
}
```

iSCSI 接続のサーバの情報を次のとおり指定してストレージシステムに登録した例を次に示します。

- サーバのニックネーム：hostB
- プロトコル：iSCSI
- OS タイプ：Linux

```
{
  "serverNickname": "hostB",
  "protocol": "iSCSI",
  "osType": "Linux"
}
```

ホストグループを追加するサーバを作成する例を次に示します。

```
{
  "serverNickname": "hostC",
  "isReserved": true
}
```

属性	型	説明
serverNickname	string	(必須) サーバのニックネームを 1～229 文字で指定します。 使用できる文字は次のとおりです。 0～9 A～Z a～z , - . / : @ \ _ 半角スペース

属性	型	説明
		<p>英字は、大文字と小文字を区別します。</p> <p>半角スペースは、使用できる文字と文字の間には使用できますが、先頭と末尾には使用できません。</p> <p>名前の先頭にハイフン (-) は使用できません。</p> <p>iSCSI 接続の場合、コンマ (,)、スラッシュ (/)、円マークまたはバックスラッシュ (\) は使用できません。</p>
protocol	string	<p>(任意) プロトコル</p> <p>指定できる値は次のとおりです。</p> <ul style="list-style-type: none"> <li>• FC</li> <li>• iSCSI</li> </ul> <p>属性 isReserved に true を指定しない場合は、この属性を必ず指定してください。</p> <p>属性 isReserved に true を指定した場合で、この属性を指定したときは指定した値は無視されます。</p>
osType	string	<p>(任意) OS タイプ</p> <p>指定できる値は次のとおりです。</p> <ul style="list-style-type: none"> <li>• Linux</li> <li>• VMware</li> <li>• Windows</li> </ul> <p>属性 isReserved に true を指定しない場合は、この属性を必ず指定してください。</p> <p>属性 isReserved に true を指定した場合で、この属性を指定したときは指定した値は無視されます。</p>
osTypeOptions	int[]	<p>(任意) OS タイプのオプションを 0~255 の整数で指定します。</p> <p>指定できる値については、マニュアル『システム構築ガイド』を参照してください。この属性は、RAID Manager ではホストモードオプションに対応します。</p> <p>OS タイプを指定して、この属性を省略した場合、指定した OS タイプに対応する値が自動で設定されます。</p> <p>属性 isReserved に true を指定した場合で、この属性を指定したときは指定した値は無視されます。</p>
isReserved	boolean	<p>(任意) ストレージシステムにホストグループを追加するサーバを作成するかどうかを指定します。</p> <ul style="list-style-type: none"> <li>• true : ホストグループを追加するサーバを作成する</li> <li>• false : ホストグループを追加するサーバを作成しない</li> </ul> <p>この属性を省略した場合、false が指定されたと見なされます。</p>

## レスポンスメッセージ

### ボディ

属性	型	説明
statusResource	string	登録したサーバの実行結果の問い合わせに使用する URL

## メモ

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-binary
@./InputParameters.json https://192.0.2.100/ConfigurationManager/simple
/v1/objects/servers
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

[サーバにホストグループ \(iSCSI ターゲット\) を追加する \(96 ページ\)](#)

## 4.4.4 サーバの設定を変更する

サーバ ID を指定して、特定のサーバの設定に関する値を変更します。

## 実行権限

ストレージ管理者 (プロビジョニング)

## リクエストライン

```
PATCH <ベース URL>/simple/v1/objects/servers/<オブジェクト ID>
```

## リクエストメッセージ

### オブジェクト ID

サーバの情報取得で取得した id の値を指定します。

属性	型	説明
id	int	(必須) サーバ ID

## クエリパラメータ

なし。

## ボディ

指定した属性だけが変更されます。

FC 接続のサーバ（サーバ ID : 10）の OS タイプを Linux に変更する例を示します。

```
{
  "osType": "Linux"
}
```

iSCSI 接続のサーバ（サーバ ID : 11）のニックネームを My\_REST\_API\_HOST に変更する例を示します。

```
{
  "nickname": "My_REST_API_HOST"
}
```

属性	型	説明
nickname	string	<p>（任意）サーバのニックネームを 1～229 文字で指定します。</p> <p>使用できる文字は次のとおりです。</p> <p>0～9 A～Z a～z , - . / : @ \ _ 半角スペース</p> <p>英字は、大文字と小文字を区別します。</p> <p>半角スペースは、使用できる文字と文字の間には使用できますが、先頭と末尾には使用できません。</p> <p>名前の先頭にハイフン (-) は使用できません。</p> <p>iSCSI 接続の場合、コンマ (,)、スラッシュ (/)、円マークまたはバックスラッシュ (\) は使用できません。</p> <p>次の属性と同時に指定できません。</p> <ul style="list-style-type: none"> <li>osType</li> <li>osTypeOptions</li> </ul>
osType	string	<p>（任意）OS タイプ</p> <p>指定できる値は次のとおりです。</p> <ul style="list-style-type: none"> <li>Linux</li> <li>VMware</li> <li>Windows</li> </ul> <p>次の属性と同時に指定できません。</p> <ul style="list-style-type: none"> <li>nickname</li> </ul>
osTypeOptions	int[]	<p>（任意）OS タイプのオプションを 0～255 の整数で指定します。</p> <p>指定できる値については、マニュアル『システム構築ガイド』を参照してください。この属性は、RAID Manager ではホストモードオプションに対応します。</p> <p>OS タイプのオプションを指定しない場合、現在設定されている OS タイプに対応する値で上書きされます。すでに値を設定していて、上書きさせたくない場合は、その値を指定してください。</p>

属性	型	説明
		次の属性と同時に指定できません。 <ul style="list-style-type: none"><li>• nickname</li></ul>

## レスポンスメッセージ

### ボディ

属性	型	説明
statusResource	string	変更したサーバの実行結果の問い合わせに使用する URL

### メモ

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H "Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X PATCH --data-binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/simple/v1/objects/servers/10
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

### 4.4.5 サーバを削除する

サーバ ID を指定して、ストレージシステムに登録されているサーバの情報を削除します。

## 実行権限

ストレージ管理者（プロビジョニング）

## リクエストライン

```
DELETE <ベース URL>/simple/v1/objects/servers/<オブジェクト ID>
```

## リクエストメッセージ

### オブジェクト ID

サーバの情報取得で取得した id の値を指定します。

属性	型	説明
id	int	(必須) サーバ ID

### クエリパラメータ

なし。

### ボディ

属性	型	説明
keepLunConfig	boolean	<p>(任意) リソースの割り当て情報を保持したまま、サーバの情報を削除するかどうかを指定します。</p> <ul style="list-style-type: none"> <li>• true: 保持する</li> <li>• false: 保持しない</li> </ul> <p>ホストグループを誤ってサーバに追加した場合は、true を指定してサーバの情報を削除したあと、サーバを再作成して正しいホストグループを追加し直してください。</p> <p>true を指定した場合、リソースの割り当て情報を管理できなくなります。通常はこの属性を指定しないでください。</p>

## レスポンスメッセージ

### ボディ

属性	型	説明
statusResource	string	削除したサーバの情報の実行結果の問い合わせに使用する URL

### メモ

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H "Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X DELETE https://192.0.2.100/ConfigurationManager/simple/v1/objects/servers/10
```

---

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

---

## 4.4.6 サーバにホストグループ (iSCSI ターゲット) を追加する

サーバ ID を指定して、特定のサーバにホストグループ (または iSCSI ターゲット) を追加します。

### 重要

---

次の条件に該当するホストグループ (または iSCSI ターゲット) は追加できません。

- ホストグループ (または iSCSI ターゲット) のプロトコルが追加するサーバのプロトコルと異なっている
  - すでに別のサーバに追加されているホストグループ (または iSCSI ターゲット) である
  - 追加するサーバとホストグループ (または iSCSI ターゲット) に関連付けられたポートがすでに接続されている
  - 追加するサーバとホストグループ (または iSCSI ターゲット) に登録されている WWN (または iSCSI ネーム) の合計が 32 個を超えている
  - ホストグループ (または iSCSI ターゲット) 内にプールから作成された仮想的なボリュームではないボリュームがある
  - ホストグループ (または iSCSI ターゲット) に関連付けられたポートのセキュリティが無効である
  - ホストグループ番号 (または iSCSI ターゲット番号) が「0」のホストグループ (または iSCSI ターゲット) である
  - 仮想ストレージマシンに割り当てられたホストグループ (または iSCSI ターゲット) である
- 

## 実行権限

ストレージ管理者 (プロビジョニング)



## リクエストライン

```
POST <ベース URL>/simple/v1/objects/servers/<オブジェクト ID>/actions/add-host-groups/invoke
```

## リクエストメッセージ

### オブジェクト ID

サーバの情報取得で取得した id の値を指定します。

属性	型	説明
id	int	(必須) サーバ ID

### クエリパラメータ

なし。

### ボディ

ホストグループ ID を指定してホストグループを追加する例を次に示します。

```
{
  "hostGroups": [
    {
      "portId": "CL1-C",
      "hostGroupId": 1
    }
  ]
}
```

ホストグループ名を指定して iSCSI ターゲットを追加する例を次に示します。

```
{
  "hostGroups": [
    {
      "portId": "CL1-D",
      "hostGroupName": "My_REST_API_HOST"
    }
  ]
}
```

属性	型	説明
hostGroups	object[]	追加するホストグループ（または iSCSI ターゲット）の情報 <ul style="list-style-type: none"> <li>(必須) portId: (string) ポート ID を 5 文字で指定します。</li> <li>(任意) hostGroupId: (int) ホストグループ（または iSCSI ターゲット）ID を 1～254 までの整数で指定します。</li> </ul>

属性	型	説明
		hostGroupId または hostGroupName のどちらか一方を必ず指定してください。また、これらの属性を同時には指定しないでください。 <ul style="list-style-type: none"><li>• (任意) hostGroupName : (string)</li></ul> ホストグループ名（または iSCSI ターゲット名）を 1～64 文字で指定します。 hostGroupId または hostGroupName のどちらか一方を必ず指定してください。また、これらの属性を同時には指定しないでください。

## レスポンスメッセージ

### ボディ

属性	型	説明
statusResource	string	追加したホストグループ（または iSCSI ターゲット）の実行結果の問い合わせに使用する URL

### メモ

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H "Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/simple/v1/objects/servers/8/actions/add-host-groups/invoke
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

### 4.4.7 ホストグループ名（iSCSI ターゲット名）をサーバのニックネームと同期する

サーバ ID を指定して、サーバのニックネームをサーバと関連づけられたホストグループ（または iSCSI ターゲット）のホストグループ名（または iSCSI ターゲット名）に同期します。

同期するホストグループ名（または iSCSI ターゲット名）は次に示す条件によって異なります。

- サーバのニックネームが 24 文字以下の場合で、同一ポート内に同期するサーバのニックネームと同じホストグループ名（または iSCSI ターゲット名）がないとき

サーバのニックネームがホストグループ名（または iSCSI ターゲット名）にそのまま同期されます。

- サーバのニックネームが 24 文字以下の場合で、同一ポート内に同期するサーバのニックネームと同じホストグループ名（または iSCSI ターゲット名）がすでにあるとき

サーバのニックネームに加えて 25 文字目までは「\_（アンダーバー）」が付与され、26 文字目から 32 文字目までは、すでにあるホストグループ名（または iSCSI ターゲット名）と重複しないように識別文字が付与されて同期されます。

例：サーバのニックネームが「abcdefghijklmnpqrst」の場合

```
abcdefghijklmnpqrst_____0001_AA
```

- サーバのニックネームが 25 文字以上の場合

サーバのニックネームの先頭から 24 文字（25 文字目以降は切り捨てられます。） + 「\_（アンダーバー）」に加えて、26 文字目から 32 文字目までは、ホストグループ名（または iSCSI ターゲット名）が重複しないように識別文字が付与されて同期されます。

例：サーバのニックネームが「abcdefghijklmnpqrstuvwxy」の場合

```
abcdefghijklmnpqrstuvw_0001_AA
```

## 実行権限

ストレージ管理者（プロビジョニング）

## リクエストライン

```
POST <ベース URL>/simple/v1/objects/servers/<オブジェクト ID>/actions/sync-host-group-names/invoke
```

## リクエストメッセージ

### オブジェクト ID

サーバの情報取得で取得した id の値を指定します。

属性	型	説明
id	int	(必須) サーバ ID

### クエリパラメータ

なし。

### ボディ

なし。

## レスポンスメッセージ

### ボディ

属性	型	説明
statusResource	string	同期したサーバの実行結果の問い合わせに使用する URL

### メモ

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H "Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/simple/v1/objects/servers/10/actions/sync-host-group-names/invoke -d ""
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

## 4.4.8 HBA の WWN (iSCSI ネーム) の情報を取得する

サーバ ID を指定して、サーバの HBA の WWN (または iSCSI ネーム) の情報を取得します。

## 実行権限

ストレージ管理者（参照）

## リクエストライン

```
GET <ベース URL>/simple/v1/objects/servers/<オブジェクト ID>/hbas
```

## リクエストメッセージ

### オブジェクト ID

サーバの情報取得で取得した id の値を指定します。

属性	型	説明
id	int	(必須) サーバ ID

### クエリパラメータ

なし。

### ボディ

なし。

## レスポンスメッセージ

### ボディ

FC 接続のサーバ（サーバ ID : 10）の HBA の WWN 情報を取得した出力例を示します。

```
{
  "data": [
    {
      "serverId": 10,
      "hbaWwn": "000000102cceccc9",
      "portIds": [
        "CL1-A"
      ]
    },
    {
      "serverId": 10,
      "hbaWwn": "1111111111111111",
      "portIds": [
        "CL1-A"
      ]
    }
  ],
}
```

```
"count": 2
}
```

iSCSI 接続のサーバ（サーバ ID : 11）の iSCSI ネームの情報を取得した出力例を示します。

```
{
  "data": [
    {
      "serverId": 11,
      "iscsiName": "iqn.1994-05.com.redhat:496799ba93",
      "portIds": [
        "CL1-B"
      ]
    },
    {
      "serverId": 11,
      "iscsiName": "iqn.1994-05.com.redhat:a7526e46aa",
      "portIds": [
        "CL1-B"
      ]
    }
  ],
  "count": 2
}
```

属性	型	説明
serverId	int	サーバ ID
hbaWwn	string	HBA の WWN
iscsiName	string	iSCSI ネーム
portIds	string[]	割り当て先のポート ID のリスト

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192
.0.2.100/ConfigurationManager/simple/v1/objects/servers/10/hbas
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[データオブジェクト \(17 ページ\)](#)

## 4.4.9 特定のサーバの HBA の WWN (iSCSI ネーム) の情報を取得する

特定のサーバの、HBA の WWN (または iSCSI ネーム) の情報を取得します。

### 実行権限

ストレージ管理者 (参照)

### リクエストライン

```
GET <ベース URL>/simple/v1/objects/servers/<オブジェクト ID>/hbas/<オブジェクト ID>
```

### リクエストメッセージ

#### オブジェクト ID

複数のオブジェクト ID を組み合わせて指定します。

- HBA の WWN を指定する場合

HBA の WWN (iSCSI ネーム) の情報取得で取得した `serverId` の値、および `hbaWwn` の値を次の形式で指定します。

```
< serverId >/hbas/< hbaWwn >
```

属性	型	説明
<code>serverId</code>	int	(必須) サーバ ID
<code>hbaWwn</code>	string	(必須) HBA の WWN

- iSCSI ネームを指定する場合

HBA の WWN (iSCSI ネーム) の情報取得で取得した `serverId` の値、および `iscsiName` の値を次の形式で指定します。

```
< serverId >/hbas/< iscsiName >
```

属性	型	説明
<code>serverId</code>	int	(必須) サーバ ID
<code>iscsiName</code>	string	(必須) iSCSI ネーム

#### クエリパラメータ

なし。

## ボディ

なし。

## レスポンスメッセージ

### ボディ

FC 接続のサーバ（サーバ ID : 10）で HBA の WWN が 000000102ccec9 の情報を取得した出力例を次に示します。

```
{
  "serverId": 10,
  "hbaWwn": "000000102ccec9",
  "portIds": [
    "CL1-A"
  ]
}
```

iSCSI 接続のサーバ（サーバ ID : 11）で iSCSI ネームが iqn.1994-05.com.redhat:496799ba93 の情報を取得した出力例を次に示します。

```
{
  "serverId": 11,
  "iscsiName": "iqn.1994-05.com.redhat:496799ba93",
  "portIds": [
    "CL1-B"
  ]
}
```

属性	型	説明
serverId	int	サーバ ID
hbaWwn	string	HBA の WWN HBA の WWN を指定した場合に有効な値が出力されます。
iscsiName	string	iSCSI ネーム iSCSI ネームを指定した場合に有効な値が出力されます。
portIds	string[]	割り当て先のポート ID のリスト

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192
```



```
.0.2.100/ConfigurationManager/simple/v1/objects/servers/10/hbas/000000102cceccc9
```

---

#### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[HBA の WWN \(iSCSI ネーム\) の情報を取得する \(100 ページ\)](#)

---

## 4.4.10 HBA の WWN (iSCSI ネーム) の情報を追加する

サーバ ID を指定して、サーバに HBA の WWN (または iSCSI ネーム) の情報を追加します。

### メモ

クラスタ構成の場合、クラスタを構成するすべてのサーバ (ノード) の WWN (または iSCSI ネーム) を同じサーバ ID のサーバに追加してください。

---

## 実行権限

ストレージ管理者 (プロビジョニング)

## リクエストライン

```
POST <ベース URL>/simple/v1/objects/servers/<オブジェクト ID>/hbas
```

## リクエストメッセージ

### オブジェクト ID

サーバの情報取得で取得した id の値を指定します。

属性	型	説明
id	int	(必須) サーバ ID

### クエリパラメータ

なし。

### ボディ

FC 接続のサーバ (サーバ ID : 10) に HBA の WWN 「210003e08b0256f9」を追加する例を次に示します。

```
{
  "hbas": [
    {
```

```

        "hbaWwn": "210003e08b0256f9"
    }
}

```

iSCSI 接続のサーバ（サーバ ID：11）に iqn 形式の iSCSI ネーム「iqn.myrestapiiscsi20150907」を追加する例を次に示します。

```

{
  "hbas": [
    {
      "iscsiName": "iqn.myrestapiiscsi20150907"
    }
  ]
}

```

属性	型	説明
hbas	object[]	<p>HBA の WWN または iSCSI ネームの情報</p> <ul style="list-style-type: none"> <li>（任意）hbaWwn（string） HBA の WWN を 16 文字の小文字で指定します。</li> <li>（任意）iscsiName（string） iSCSI ネームを指定します。 iqn 形式または eui 形式で指定してください。 <ul style="list-style-type: none"> <li>- iqn 形式 iqn. に続けて全体で 223 文字以内の値で指定します。 使用できる文字は次のとおりです。 0～9 A～Z a～z - . : 指定例：iqn.rest.example.of.iqn.form</li> <li>- eui 形式 eui. に続けて 16 進数で指定します。全体で 20 文字の値を指定してください。 使用できる文字は次のとおりです。 0～9 A～F a～f 指定例：eui.0900ABDC32598D26</li> </ul> </li> </ul> <p>hbaWwn または iscsiName のどちらか一方を必ず指定してください。また、これらの属性を同時には指定しないでください。</p>

## レスポンスメッセージ

### ボディ

属性	型	説明
statusResource	string	HBA の WWN（または iSCSI ネーム）を追加したサーバの実行結果の問い合わせに使用する URL

---

## メモ

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

---

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H "Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/simple/v1/objects/servers/10/hbas
```

---

## 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

---

### 4.4.11 サーバから HBA の WWN (iSCSI ネーム) の情報を削除する

HBA の WWN (または iSCSI ネーム) の情報を、サーバから削除します。

## 実行権限

ストレージ管理者 (プロビジョニング)

## リクエストライン

```
DELETE <ベース URL>/simple/v1/objects/servers/<オブジェクト ID>/hbas/<オブジェクト ID>
```

## リクエストメッセージ

### オブジェクト ID

複数のオブジェクト ID を組み合わせて指定します。

- HBA の WWN を指定する場合

HBA の WWN (iSCSI ネーム) の情報取得で取得した `serverId` の値、および `hbaWwn` の値を次の形式で指定します。

```
< serverId > / hbas / < hbaWwn >
```

属性	型	説明
<code>serverId</code>	<code>int</code>	(必須) サーバ ID
<code>hbaWwn</code>	<code>string</code>	(必須) HBA の WWN

- iSCSI ネームを指定する場合

HBA の WWN (iSCSI ネーム) の情報取得で取得した `serverId` の値、および `iscsiName` の値を次の形式で指定します。

```
< serverId > / hbas / < iscsiName >
```

属性	型	説明
<code>serverId</code>	<code>int</code>	(必須) サーバ ID
<code>iscsiName</code>	<code>string</code>	(必須) iSCSI ネーム

## クエリパラメータ

なし。

## ボディ

なし。

## レスポンスメッセージ

### ボディ

属性	型	説明
<code>statusResource</code>	<code>string</code>	情報を削除したサーバの実行結果の問い合わせに使用する URL

### メモ

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X DELETE https://
192.0.2.100/ConfigurationManager/simple/v1/objects/servers/10/hbas/210003e0
8b0256f9
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

## 4.4.12 サーバのパス情報の一覧を取得する

サーバ ID を指定して、特定のサーバのパス情報を取得します。条件を指定して、絞り込んだ情報を取得したりできます。

### 実行権限

ストレージ管理者 (参照)

### リクエストライン

```
GET <ベース URL>/simple/v1/objects/servers/<オブジェクト ID>/paths
```

### リクエストメッセージ

#### オブジェクト ID

サーバの情報取得で取得した id の値を指定します。

属性	型	説明
id	int	(必須) サーバ ID

#### クエリパラメータ

パラメータ	型	フィルタ条件
hbaWwn	string	(任意) HBA の WWN 指定した値と完全に一致するサーバのパス情報が取得されます。
iscsiName	string	(任意) iSCSI ネーム 指定した値と完全に一致するサーバのパス情報が取得されます。
portId	string	(任意) 割り当て先のポート ID 指定した値と完全に一致するサーバのパス情報が取得されます。

## ボディ

なし。

## レスポンスメッセージ

### ボディ

FC 接続のサーバ（サーバ ID : 10）のパス情報を取得した出力例を次に示します。

```
{
  "data": [
    {
      "id": "0000000102ccec9,CL1-A",
      "serverId": 10,
      "hbaWwn": "0000000102ccec9",
      "portId": "CL1-A"
    },
    {
      "id": "111111111111111,CL1-A",
      "serverId": 10,
      "hbaWwn": "111111111111111",
      "portId": "CL1-A"
    }
  ],
  "count": 2
}
```

iSCSI 接続のサーバ（サーバ ID : 11）のパス情報を取得した出力例を次に示します。

```
{
  "data": [
    {
      "id": "iqn.1994-05.com.redhat:496799ba93,CL1-B",
      "serverId": 11,
      "iscsiName": "iqn.1994-05.com.redhat:496799ba93",
      "portIds": "CL1-B"
    },
    {
      "id": "iqn.1994-05.com.redhat:a7526e46aa,CL1-B",
      "serverId": 11,
      "iscsiName": "iqn.1994-05.com.redhat:a7526e46aa",
      "portIds": "CL1-B"
    }
  ],
  "count": 2
}
```

属性	型	説明
id	string	HBA の WWN（または iSCSI ネーム）および割り当て先のポート ID がコンマで連結した形式で出力されます。
serverId	int	サーバ ID
hbaWwn	string	HBA の WWN

属性	型	説明
iscsiName	string	iSCSI ネーム
portId	string	割り当て先のポート ID

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192
.0.2.100/ConfigurationManager/simple/v1/objects/servers/10/paths
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[データオブジェクト \(17 ページ\)](#)

## 4.4.13 サーバと特定のポート間のパス情報を取得する

サーバ ID、HBA の WWN（または iSCSI ネーム）、およびポート ID を指定して、サーバと特定のポート間のパス情報を取得します。

## 実行権限

ストレージ管理者（参照）

## リクエストライン

```
GET <ベース URL>/simple/v1/objects/servers/<オブジェクト ID>/paths/<オブジェ
クト ID>
```

## リクエストメッセージ

### オブジェクト ID

複数のオブジェクト ID を組み合わせて指定します。

サーバのパスの情報取得で取得した serverId の値および id の値を、次の形式で指定します。

```
< serverId > /paths /< id >
```

<id>の値には、hbaWwn（または iscsiName）の値および portId の値を連結した形式でも指定できます。

- HBA の WWN を指定する場合

```
<serverId>/paths/<hbaWwn>,<portId>
```

属性	型	説明
serverId	int	(必須) サーバ ID
hbaWwn	string	(必須) HBA の WWN
portId	string	(必須) ポート ID

- iSCSI ネームを指定する場合

```
<serverId>/paths/<iscsiName>,<portId>
```

属性	型	説明
serverId	int	(必須) サーバ ID
iscsiName	string	(必須) iSCSI ネーム
portId	string	(必須) ポート ID

## クエリパラメータ

なし。

## ボディ

なし。

## レスポンスメッセージ

### ボディ

FC 接続のサーバ（サーバ ID : 10、HBA の WWN : 000000102ccec9）とポート間（ポート ID : CL1-A）のパス情報を取得した出力例を次に示します。

```
{
  "id": "000000102ccec9,CL1-A",
  "serverId": 10,
  "hbaWwn": "000000102ccec9",
  "portId": "CL1-A"
}
```

iSCSI 接続のサーバ（サーバ ID : 11、iSCSI ネーム : iqn.1994-05.com.redhat:496799ba93）とポート間（ポート ID : CL1-B）のパス情報を取得した出力例を次に示します。

```
{
  "id": "iqn.1994-05.com.redhat:496799ba93,CL1-B",
```



```

"serverId": 11,
"iscsiName": "iqn.1994-05.com.redhat:496799ba93",
"portId": "CL1-B"
}

```

取得する属性については、サーバのパス情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```

curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192
.0.2.100/ConfigurationManager/simple/v1/objects/servers/10/paths/000000102c
ceccc9,CL1-A

```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[サーバのパス情報の一覧を取得する \(109 ページ\)](#)

## 4.4.14 サーバにパス情報を追加する

サーバ ID を指定して、特定のサーバにパス情報を追加します。

## 実行権限

ストレージ管理者 (プロビジョニング)

## リクエストライン

```
POST <ベース URL>/simple/v1/objects/servers/<オブジェクト ID>/paths
```

## リクエストメッセージ

### オブジェクト ID

サーバの情報取得で取得した id の値を指定します。

属性	型	説明
id	int	(必須) サーバ ID

## クエリパラメータ

なし。

## ボディ

FC 接続のサーバ（サーバ ID : 10）にパス情報を追加する例を次に示します。

```
{
  "hbaWwn": "210003e08b0256f9",
  "portIds": [
    "CL1-A"
  ]
}
```

iSCSI 接続のサーバ（サーバ ID : 11）にパス情報を追加する例を次に示します。

```
{
  "iscsiName": "iqn.myrestapiiscsi20150907",
  "portIds": [
    "CL1-B"
  ]
}
```

属性	型	説明
hbaWwn	string	(任意) HBA の WWN を 16 文字の小文字で指定します。 hbaWwn または iscsiName のどちらか一方を必ず指定してください。また、これらの属性を同時には指定しないでください。
iscsiName	string	(任意) iSCSI ネームを指定します。 iqn 形式または eui 形式で指定してください。 <ul style="list-style-type: none"> <li>iqn 形式 iqn. に続けて全体で 223 文字以内の値で指定します。 使用できる文字は次のとおりです。 0～9 A～Z a～z - . : 指定例 : iqn.rest.example.of.iqn.form</li> <li>eui 形式 eui. に続けて 16 進数で指定します。全体で 20 文字の値を指定してください。 使用できる文字は次のとおりです。 0～9 A～F a～f 指定例 : eui.0900ABDC32598D26</li> </ul> hbaWwn または iscsiName のどちらか一方を必ず指定してください。また、これらの属性を同時には指定しないでください。
portIds	string[]	(必須) 割り当て先のポート ID を指定します。

## レスポンスメッセージ

### ボディ

属性	型	説明
statusResource	string	パス情報を追加したサーバの実行結果の問い合わせに使用する URL

### メモ

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H "Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/simple/v1/objects/servers/10/paths
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

### 4.4.15 サーバからパス情報を削除する

特定のサーバからパス情報を削除します。

## 実行権限

ストレージ管理者 (プロビジョニング)

## リクエストライン

```
DELETE <ベース URL>/simple/v1/objects/servers/<オブジェクト ID>/paths/<オブジェクト ID>
```

## リクエストメッセージ

### オブジェクト ID

複数のオブジェクト ID を組み合わせて指定します。

サーバのパスの情報取得で取得した `serverId` の値および `id` の値を、次の形式で指定します。

```
< serverId > /paths /< id >
```

< id > の値には、`hbaWwn`（または `iscsiName`）の値および `portId` の値を連結した形式でも指定できます。

- HBA の WWN を指定する場合

```
< serverId > /paths /< hbaWwn > , < portId >
```

属性	型	説明
<code>serverId</code>	int	(必須) サーバ ID
<code>hbaWwn</code>	string	(必須) HBA の WWN
<code>portId</code>	string	(必須) ポート ID

- iSCSI ネームを指定する場合

```
< serverId > /paths /< iscsiName > , < portId >
```

属性	型	説明
<code>serverId</code>	int	(必須) サーバ ID
<code>iscsiName</code>	string	(必須) iSCSI ネーム
<code>portId</code>	string	(必須) ポート ID

### クエリパラメータ

なし。

### ボディ

なし。

## レスポンスメッセージ

### ボディ

属性	型	説明
<code>statusResource</code>	string	パス情報を削除したサーバの実行結果の問い合わせに使用する URL

## メモ

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X DELETE https://
192.0.2.100/ConfigurationManager/simple/v1/objects/servers/10/paths/210003e
08b0256f9,CL1-A
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

## 4.4.16 iSCSI ターゲットの情報を取得する

サーバ ID を指定して、iSCSI ターゲットの情報を取得します。

## 実行権限

ストレージ管理者 (参照)

## リクエストライン

```
GET <ベース URL>/simple/v1/objects/servers/<オブジェクト ID>/target-iscsi-po
rts
```

## リクエストメッセージ

### オブジェクト ID

サーバの情報取得で取得した id の値を指定します。

属性	型	説明
id	int	(必須) サーバ ID

## クエリパラメータ

なし。

## ボディ

なし。

## レスポンスメッセージ

### ボディ

iSCSI 接続のサーバ（サーバ ID : 11）の iSCSI ターゲットの情報を取得した出力例を次に示します。

```
{
  "data": [
    {
      "portId": "CL1-B",
      "targetIscsiName": "iqn.rest.example.of.iqn.hostB"
    }
  ],
  "count": 1
}
```

属性	型	説明
portId	string	割り当て先のポート ID
targetIscsiName	string	iSCSI ターゲットの iSCSI ネーム

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192
.0.2.100/ConfigurationManager/simple/v1/objects/servers/11/target-iscsi-por
ts
```

---

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[データオブジェクト \(17 ページ\)](#)

---

## 4.4.17 特定の iSCSI ターゲットの情報を取得する

サーバ ID およびポート ID を指定して、iSCSI ターゲットの情報を取得します。

### 実行権限

ストレージ管理者（参照）

### リクエストライン

```
GET <ベース URL>/simple/v1/objects/servers/<オブジェクト ID>/target-iscsi-ports/<オブジェクト ID>
```

### リクエストメッセージ

#### オブジェクト ID

サーバの情報取得で取得した id の値および portId の値を、次の形式で指定します。

```
< id > /target-iscsi-ports /< portId >
```

属性	型	説明
id	int	(必須) サーバ ID
portId	string	(必須) ポート ID

#### クエリパラメータ

なし。

#### ボディ

なし。

### レスポンスメッセージ

#### ボディ

iSCSI 接続のサーバ（サーバ ID : 11、ポート ID : CL1-B）の iSCSI ターゲットの情報を取得した出力例を次に示します。

```
{
  "portId": "CL1-B",
  "targetIscsiName": "iqn.rest.example.of.iqn.hostB"
}
```

取得する属性については、iSCSI ターゲットの情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192
.0.2.100/ConfigurationManager/simple/v1/objects/servers/11/target-iscsi-ports/CL1-B
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[iSCSI ターゲットの情報を取得する \(117 ページ\)](#)

## 4.4.18 iSCSI ターゲットの設定を変更する

サーバ ID およびポート ID を指定して、iSCSI ターゲットの iSCSI ネームを変更します。

## 実行権限

ストレージ管理者 (プロビジョニング)

## リクエストライン

```
PATCH <ベース URL>/simple/v1/objects/servers/<オブジェクト ID>/target-iscsi-ports/<オブジェクト ID>
```

## リクエストメッセージ

### オブジェクト ID

複数のオブジェクト ID を組み合わせて指定します。

サーバの情報取得で取得した id の値および portId の値を、次の形式で指定します。

```
< id >/target-iscsi-ports/< portId >
```

属性	型	説明
id	int	(必須) サーバ ID
portId	string	(必須) ポート ID



## クエリパラメータ

なし。

## ボディ

iSCSI 接続のサーバ（サーバ ID：11、ポート ID：CL1-B）の iSCSI ターゲットの iSCSI ネームを「iqn.rest.example.of.iqn.hostB」に変更する例を次に示します。

```
{
  "targetIscsiName": "iqn.rest.example.of.iqn.hostB"
}
```

属性	型	説明
targetIscsiName	string	<p>(必須) iSCSI ターゲットの iSCSI ネームを指定します。iqn 形式または eui 形式で指定してください。</p> <ul style="list-style-type: none"> <li>• iqn 形式 iqn. に続けて全体で 223 文字以内の値で指定します。使用できる文字は次のとおりです。 0～9 A～Z a～z - . : 指定例：iqn.rest.example.of.iqn.form</li> <li>• eui 形式 eui. に続けて 16 進数で指定します。全体で 20 文字の値を指定してください。使用できる文字は次のとおりです。 0～9 A～F a～f 指定例：eui.0900ABDC32598D26</li> </ul>

## レスポンスメッセージ

### ボディ

属性	型	説明
statusResource	string	iSCSI ネームを変更したサーバの実行結果の問い合わせに使用する URL

### メモ

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X PATCH --data-binary
@./InputParameters.json https://192.0.2.100/ConfigurationManager/simple/v1/objects/servers/11/target-iscsi-ports/CL1-B
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

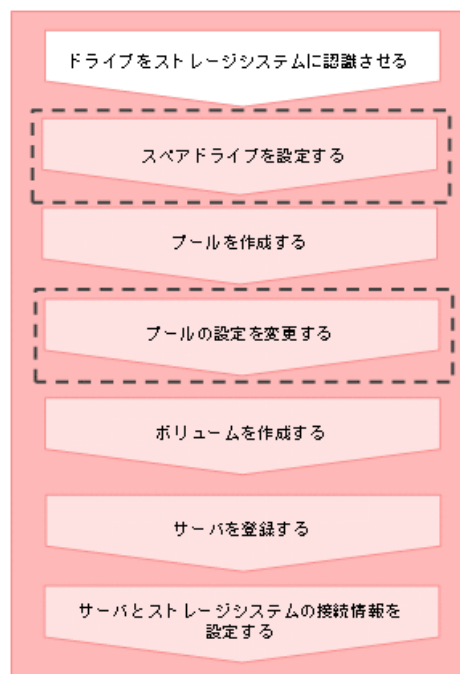
[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

## 4.5 ボリュームの管理

REST API を使用してボリュームの情報を取得、新規にボリュームを作成、作成済みのボリュームをサーバに割り当てたり、使用率の高いボリュームの容量を拡張したりする操作について説明します。

### ボリュームを利用するための準備の流れ

ストレージシステムの容量をボリュームとして利用するための準備をします。また、ボリュームを割り当てるためにサーバを登録したり、ストレージシステムのポートとの接続情報を設定したりします。



(凡例)

: REST API で操作する手順

: REST API 以外で操作する手順

: 必要に応じて実施する操作

---

## メモ

---

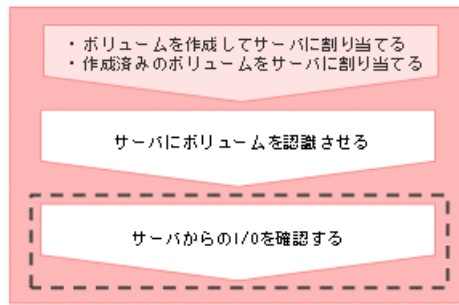
サーバの登録後に WWN または iSCSI イニシエータ名を追加したり、削除したりすることもできます。

---




- ドライブをスペアドライブに設定する  
ドライブの数を指定して、スペアドライブに設定します。
- プールを作成する  
プール名とドライブ情報を指定して、プールを作成します。
- プールの設定を変更する  
プール ID を指定して、プールの設定（プール名やしきい値など）を変更します。
- ボリュームを作成する  
容量およびボリュームに付与するニックネームを指定して、ボリュームを作成します。
- サーバを登録する  
ストレージシステムにサーバの情報を登録します。また、サーバのニックネームだけを指定して、ホストグループを追加するためのサーバを作成することもできます。
- サーバにパスの情報を追加する  
サーバ ID を指定して、特定のサーバにパス情報を追加します。
- iSCSI ターゲットの設定を変更する  
サーバ ID およびポート ID を指定して、iSCSI ターゲットの iSCSI ネームを変更します。
- HBA の WWN (iSCSI ネーム) の情報を追加する  
サーバ ID を指定して、サーバに HBA の WWN (または iSCSI ネーム) の情報を追加します。
- サーバから HBA の WWN (iSCSI ネーム) の情報を削除する  
HBA の WWN (または iSCSI ネーム) の情報を、サーバから削除します。

## ボリューム割り当ての流れ

サーバにボリュームを割り当てます。割り当て後のボリュームを認識するために、サーバで OS に応じた操作を実施します。



(凡例)

-  : REST APIで操作する手順
-  : REST API以外で操作する手順
-  : 必要に応じて実施する操作

- ボリュームを作成する

容量およびボリュームに付与するニックネームを指定して、ボリュームを作成します。

- ボリュームとサーバを接続する

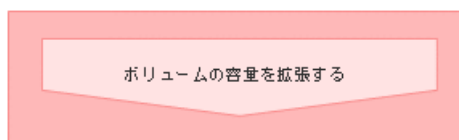
サーバとボリューム間のすべてのパスに対して接続設定します。

- サーバからの I/O を確認する

サーバからの I/O の状態を確認して、ボリュームが正しく割り当てられているかどうかを確認します。

## ボリュームの容量を拡張する流れ

ボリュームの容量が不足した場合、容量を拡張します。



(凡例)

-  : REST APIで操作する手順

- ボリュームの容量を拡張する

ボリューム ID を指定して、ボリュームの容量を拡張します。

- サーバからの I/O を確認する

サーバからの I/O の状態を確認して、ボリュームが正しく割り当てられているかどうかを確認します。

### 4.5.1 ボリュームの情報を取得する

ボリュームに関する情報を取得します。ボリュームの条件を指定して、絞り込んだ情報を取得することもできます。

## 実行権限

ストレージ管理者（参照）

## リクエストライン

```
GET <ベース URL>/simple/v1/objects/volumes
```

## リクエストメッセージ

### オブジェクト ID

なし。

### クエリパラメータ

1 回のリクエストで取得できるボリュームの情報は最大で 500 個です。501 個以上のボリュームの情報を取得する場合は、count パラメータと startVolumeId パラメータを組み合わせて複数回に分けて API を実行してください。count パラメータを指定することで取得するボリュームの情報をフィルタリングすることもできます。

パラメータ	型	フィルタ条件
poolId	int	(任意) ボリュームが属するプール ID poolName と同時に指定しないでください。
poolName	string	(任意) ボリュームが属するプール名 指定した値の一部がプール名と一致する場合もボリュームの情報が取得されます。 poolId と同時に指定しないでください。poolId を指定している場合、このパラメータは無視されます。
serverId	int	(任意) ボリュームが割り当てられているサーバ ID serverNickname と同時に指定しないでください。
serverNickname	string	(任意) ボリュームが割り当てられているサーバのニックネーム 指定した値の一部がサーバのニックネームと一致する場合もボリュームの情報が取得されます。 serverId と同時に指定しないでください。serverId を指定している場合、このパラメータは無視されます。
nickname	string	(任意) ボリュームのニックネーム 指定した値の一部がボリュームのニックネームと一致する場合もボリュームの情報が取得されます。
minTotalCapacity	long	(任意) ボリュームの最小容量 (MiB) 取得するボリュームの最小容量として 47～268435456 までの整数で指定します。
maxTotalCapacity	long	(任意) ボリュームの最大容量 (MiB) 取得するボリュームの最大容量として 47～268435456 までの整数で指定します。

パラメータ	型	フィルタ条件
minUsedCapacity	long	(任意) ボリュームの最小使用容量 (MiB) 取得するボリュームの最小使用容量として 0～268435456 までの整数で指定します。
maxUsedCapacity	long	(任意) ボリュームの最大使用容量 (MiB) 取得するボリュームの最大使用容量として 0～268435456 までの整数で指定します。
startVolumeId	int	(任意) 取得を開始するボリュームの ID を 0～65279 の値で指定します。
count	int	(任意) 取得するボリュームの情報の個数を 1～500 の値で指定します。 このパラメータを省略した場合、500 が指定されたと見なされます。

## ボディ

なし。

## レスポンスメッセージ

### ボディ

プール ID が 0 のプール (クエリパラメータ poolId に 0 を指定) に属するボリュームの情報を取得した出力例を次に示します。

```
{
  "data": [
    {
      "id": 1,
      "nickname": "volume_01",
      "poolId": 0,
      "poolName": "test_pool1",
      "totalCapacity": 1024,
      "usedCapacity": 0,
      "savingSetting": "DEDUPLICATION_AND_COMPRESSION",
      "isDataReductionShareEnabled": true,
      "compressionAcceleration": false,
      "capacitySavingStatus": "Enabled",
      "numberOfConnectingServers": 0,
      "numberOfSnapshots": 0,
      "volumeTypes": []
    },
    {
      "id": 2,
      "nickname": "volume_02",
      "poolId": 0,
      "poolName": "test_pool1",
      "totalCapacity": 1024,
      "usedCapacity": 0,
      "savingSetting": "COMPRESSION",
      "isDataReductionShareEnabled": true,
      "compressionAcceleration": false,

```

```

        "capacitySavingStatus": "Enabled",
        "numberOfConnectingServers": 0,
        "numberOfSnapshots": 0,
        "volumeTypes": []
    },
    ],
    "count": 2,
    "totalCount": 2,
    "hasNext": false
}

```

属性	型	説明
id	int	ボリューム ID
nickname	string	ニックネーム
poolId	int	ボリュームが属するプール ID
poolName	string	ボリュームが属するプール名 ボリュームを作成中または削除中の場合、この属性は出力されません。
totalCapacity	long	ボリュームの総容量 (MiB)
usedCapacity	long	ボリュームの使用済み容量 (MiB) ボリュームを作成中または削除中の場合、0 が出力されます。
savingSetting	string	容量削減機能（重複排除および圧縮機能）の適用状況 <ul style="list-style-type: none"> <li>• DEDUPLICATION_AND_COMPRESSION：重複排除および圧縮が有効</li> <li>• COMPRESSION：圧縮が有効</li> <li>• DISABLE：設定なし</li> </ul>
isDataReductionShareEnabled	boolean	スナップショットと削減データを共有するかどうかの設定 <ul style="list-style-type: none"> <li>• true：有効 (データ削減共有ボリューム)</li> <li>• false：無効 (dedupe and compression による容量削減ボリューム)</li> </ul> 容量削減機能が有効なボリュームのときに出力されます。
compressionAcceleration	boolean	圧縮アクセラレータの適用状況 <ul style="list-style-type: none"> <li>• true：有効</li> <li>• false：無効</li> </ul> 容量削減機能が有効なボリュームのときに出力されます。圧縮アクセラレータ搭載装置でのみ意味のある値です。
capacitySavingStatus	string	容量削減機能（重複排除および圧縮機能）の状態 <ul style="list-style-type: none"> <li>• Disabled：容量削減機能が無効</li> <li>• Enabled：容量削減機能が有効</li> <li>• Rehydrating：容量削減機能を無効化中（データ伸長中）</li> <li>• Enabling：容量削減機能を有効化中</li> <li>• Deleting：容量削減機能が有効なボリュームを削除中</li> <li>• Converting：容量削減機能の圧縮方式を変更中</li> </ul>

属性	型	説明
		<ul style="list-style-type: none"> <li>Failed: 重複排除用システムデータボリューム内のデータが不正な状態</li> </ul>
numberOfConnectingServers	int	ボリュームと接続している、HA Storage Manager Embedded で管理しているサーバの数
numberOfSnapshots	int	スナップショットの数
volumeTypes	string[]	ボリュームタイプ <ul style="list-style-type: none"> <li>Snapshot: ボリューム ID が定義されているスナップショットの作成先ボリューム</li> <li>System: 重複排除用システムデータボリュームまたはジャーナルボリューム</li> <li>Command Device: コマンドデバイスとして使用しているボリューム</li> <li>SLU: SCSI アーキテクチャモデルである Conglomerate LUN structure に使われるボリューム</li> <li>Attached Unmanaged: ほかのツールで割り当て済みのボリューム</li> </ul>

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H "Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192.0.2.100/ConfigurationManager/simple/v1/objects/volumes?poolId=63
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[データオブジェクト \(17 ページ\)](#)

## 4.5.2 特定のボリュームの情報を取得する

ボリューム ID を指定して、特定のボリュームの情報を取得します。

## 実行権限

ストレージ管理者 (参照)

## リクエストライン

```
GET <ベース URL>/simple/v1/objects/volumes/<オブジェクト ID>
```



## リクエストメッセージ

### オブジェクト ID

ボリュームの情報取得で取得した `id` の値を指定します。

属性	型	説明
<code>id</code>	<code>int</code>	(必須) ボリューム ID

### クエリパラメータ

なし。

### ボディ

なし。

## レスポンスメッセージ

### ボディ

ボリューム ID が 1 のボリュームの情報取得した例を次に示します。

```
{
  "id": 1,
  "nickname": "volume_01",
  "poolId": 0,
  "poolName": "test_pool1",
  "totalCapacity": 1024,
  "usedCapacity": 0,
  "freeCapacity": 1024,
  "reservedCapacity": 0,
  "savingSetting": "DEDUPLICATION_AND_COMPRESSION",
  "isDataReductionShareEnabled": true,
  "compressionAcceleration": false,
  "compressionAccelerationStatus": "DISABLED",
  "capacitySavingStatus": "Enabled",
  "numberOfConnectingServers": 0,
  "numberOfSnapshots": 0,
  "luns": [],
  "volumeTypes": []
}
```

取得する属性については、ボリュームの情報取得する場合の属性に加えて、次に示す属性が取得されます。

属性	型	説明
<code>freeCapacity</code>	<code>long</code>	ボリュームの空き容量 (MiB) ボリュームを作成中または削除中の場合、ボリュームの総容量が出力されます。
<code>reservedCapacity</code>	<code>long</code>	ボリュームのページ予約容量 (MiB)

属性	型	説明
compressionAccelerationStatus	string	ボリュームに格納されているデータの圧縮アクセラレータ適用の状態 <ul style="list-style-type: none"> <li>DISABLED: 圧縮アクセラレータが適用されていない</li> <li>ENABLED: 圧縮アクセラレータが適用されている</li> <li>HYBRID: 圧縮アクセラレータが有効なデータと無効なデータが、ボリューム内に混在している</li> </ul> 容量削減機能が有効なボリュームまたは重複排除用システムデータボリューム（データストア）のときに出力されます。
capacitySavingProgress	int	容量削減機能（重複排除および圧縮機能）の状態の進捗率（%）
luns	object[]	LUN の設定情報 <ul style="list-style-type: none"> <li>lun (int) LU 番号</li> <li>serverId (int) サーバ ID</li> <li>portId (string) 割り当て先のポート ID</li> </ul>

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H "Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192.0.2.100/ConfigurationManager/simple/v1/objects/volumes/100
```

### —— 関連リンク ——

[HTTP ステータスコード \(13 ページ\)](#)

[ボリュームの情報を取得する \(124 ページ\)](#)

## 4.5.3 ボリュームを作成する

容量およびボリュームに付与するニックネームを指定して、ボリュームを作成します。

## 実行権限

ストレージ管理者（プロビジョニング）

## リクエストライン

POST <ベース URL>/simple/v1/objects/volumes

## リクエストメッセージ

### オブジェクト ID

なし。

### クエリパラメータ

なし。

### ボディ

プール ID が 63 のプールに 100MiB のボリュームを作成する例を次に示します。

```
{
  "capacity": 100,
  "number": 1,
  "nicknameParam": {
    "baseName": "REST_API_100MVolume"
  },
  "savingSetting": "DEDUPLICATION_AND_COMPRESSION",
  "isDataReductionShareEnabled": "true",
  "poolId": 63
}
```

属性	型	説明
capacity	long	(必須) 作成するボリュームの容量 (MiB) 47～268435456 までの整数で指定します。
number	int	(任意) 作成するボリュームの個数 1～1000 までの整数で指定します。 この属性を省略した場合、作成されるボリュームの数は 1 個です。
nicknameParam	object	作成するボリュームに付与するニックネームの情報 <ul style="list-style-type: none"> <li>(必須) baseName (string) ニックネームを 1～32 文字で指定します。 使用できる文字は次のとおりです。 0～9 A～Z a～z ! # \$ % &amp; ' ( ) + , - . / : = @ [ \ ] ^ _ ` { } ~ 半角スペース 英字は、大文字と小文字が区別されます。 半角スペースは、使用できる文字と文字の間には使用できますが、先頭と末尾には使用できません。</li> <li>(任意) startNumber (int) ニックネームのあとに続けて付与したい番号の開始位置を 0～65279 までの整数で指定します。</li> <li>(任意) numberOfDigits (int)</li> </ul>

属性	型	説明
		<p>ニックネームのあとに続けて付与したい番号の桁数を1～5までの整数で指定します。</p> <p>この属性を指定する場合、startNumber も必ず指定します。startNumber を指定して、この属性を省略した場合、1 が指定されたと見なされます。</p> <p>例：baseName に AAA、startNumber に 1、および numberOfDigits に 3 を指定して、作成するボリュームの個数として number に 100 を指定した場合、AAA001～AAA100 のニックネームが付与されます。</p> <p>baseName の文字数および startNumber と numberOfDigits の指定から生成される番号の桁数の合計が 32 文字以下になるように指定してください。</p>
savingSetting※	string	<p>(任意) 容量削減機能（重複排除および圧縮機能）の設定次に示す値を指定します。</p> <ul style="list-style-type: none"> <li>• DEDUPLICATION_AND_COMPRESSION：重複排除および圧縮を有効にする</li> <li>• COMPRESSION：圧縮を有効にする</li> <li>• DISABLE：設定しない</li> </ul> <p>この属性を省略した場合、DISABLE が指定されたと見なされます。DEDUPLICATION_AND_COMPRESSION や COMPRESSION は、SSD を使用したプール上にボリュームを作成する場合にのみ指定可能です。</p>
isDataReductionShareEnabled	boolean	<p>(任意) スナップショットと削減データを共有するかどうかの設定</p> <ul style="list-style-type: none"> <li>• true：有効 (データ削減共有ボリューム)</li> <li>• false：無効 (dedupe and compression による容量削減ボリューム)</li> </ul> <p>この属性を省略した場合、false が指定されたと見なされますが、true を指定することを推奨します。また、ボリュームを作成したあとの、属性値の変更はできません。容量削減機能が有効なボリュームの詳細については、マニュアル『システム構築ガイド』を参照してください。</p>
poolId	int	<p>(必須) ボリュームを作成するプールのプール ID を 0～127 までの整数で指定します。</p>

**注※**

ストレージシステムが iStorage V310F で容量削減機能の設定を有効にすると、圧縮アクセラレータの設定が自動で有効になります。

## レスポンスメッセージ

**ボディ**

属性	型	説明
statusResource	string	作成したボリュームの実行結果の問い合わせに使用する URL

---

## メモ

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

---

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H "Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/simple/v1/objects/volumes
```

---

## 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

[特定のボリュームの情報を取得する \(128 ページ\)](#)

---

### 4.5.4 ボリュームの設定を変更する

ボリューム ID を指定して、ニックネームの変更や、容量削減機能（重複排除および圧縮機能）の設定を変更します。

## 実行権限

ストレージ管理者（プロビジョニング）

## リクエストライン

```
PATCH <ベース URL>/simple/v1/objects/volumes/<オブジェクト ID>
```

## リクエストメッセージ

### オブジェクト ID

ボリュームの情報取得で取得した id の値を指定します。

属性	型	説明
id	int	(必須) ボリューム ID

## クエリパラメータ

なし。

## ボディ

指定した属性だけが変更されます。

ニックネームを変更する例を次に示します。

```
{
  "nickname": "REST_API_10GVolume"
}
```

属性	型	説明
nickname	string	<p>(任意) 変更するニックネームを 1～32 文字で指定します。</p> <p>使用できる文字は次のとおりです。</p> <p>0～9 A～Z a～z ! # \$ % &amp; ' ( ) + , - . / : = @ [ \ ] ^ _ ` { } ~ 半角スペース</p> <p>英字は、大文字と小文字が区別されます。</p> <p>半角スペースは、使用できる文字と文字の間には使用できますが、先頭と末尾には使用できません。</p> <p>この属性を指定する場合、savingSetting を同時に指定しないでください。</p>
savingSetting	string	<p>(任意) 容量削減機能（重複排除および圧縮機能）の設定</p> <ul style="list-style-type: none"> <li>• DEDUPLICATION_AND_COMPRESSION：重複排除および圧縮を有効にする</li> <li>• COMPRESSION：圧縮を有効にする（重複排除は無効のまま）</li> <li>• DISABLE：重複排除および圧縮を無効にする</li> </ul> <p>この属性を指定する場合、ボリュームの設定変更に時間が掛かることがあります。容量削減機能の詳細はマニュアル『システム構築ガイド』を参照してください。</p> <p>この属性を指定する場合、nickname を同時に指定しないでください。</p> <p>DEDUPLICATION_AND_COMPRESSION や COMPRESSION は、SSD を使用したプール上のボリュームである場合にのみ指定可能です。</p> <p>DEDUPLICATION_AND_COMPRESSION(重複排除および圧縮)が設定されているボリュームを COMPRESSION(圧縮)に変更することはできません。</p> <p>データ削減共有ボリュームに対して次の制限があります。</p> <ul style="list-style-type: none"> <li>• COMPRESSION(圧縮)が設定されているボリュームを DISABLED(無効)に変更することはできません。</li> <li>• DEDUPLICATION_AND_COMPRESSION(重複排除および圧縮)が設定されているボリュームを DISABLE(無効)に変更することはできません。</li> </ul>

属性	型	説明
		<ul style="list-style-type: none"> <li>DISABLE(無効)のボリュームに COMPRESSION や DEDUPLICATION_AND_COMPRESSION を指定できません。</li> </ul>
compressionAcceleration	boolean	(任意) 圧縮アクセラレータの設定 <ul style="list-style-type: none"> <li>true: 有効</li> <li>false: 無効</li> </ul> ストレージシステムが iStorage V310F の場合に指定できます。 この属性を指定する場合、nickname を同時に指定しないでください。

## レスポンスメッセージ

### ボディ

- ニックネームを変更した場合

属性	型	説明
affectedResources	string[]	変更したボリュームの情報を参照するための URL の一覧
operationDetails	object[]	変更したボリュームの詳細情報 詳細については、commandStatus オブジェクトの属性 operationDetails の説明を参照してください。

- 容量削減機能（重複排除および圧縮機能）の設定を変更した場合

属性	型	説明
statusResource	string	変更したボリュームの情報を参照するための URL

### メモ

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X PATCH --data-bi
nary @./InputParameters.json https://192.0.2.100/ConfigurationManager/simpl
e/v1/objects/volumes/100
```

---

## 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[commandStatus オブジェクト \(19 ページ\)](#)

[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

[特定のボリュームの情報を取得する \(128 ページ\)](#)

---

## 4.5.5 ボリュームの容量を拡張する

ボリューム ID を指定して、ボリュームの容量を拡張します。

### 実行権限

ストレージ管理者 (プロビジョニング)

### リクエストライン

```
POST <ベース URL>/simple/v1/objects/volumes/<オブジェクト ID>/actions/expand
/invoke
```

### リクエストメッセージ

#### オブジェクト ID

ボリュームの情報取得で取得した id の値を指定します。

属性	型	説明
id	int	(必須) ボリューム ID

#### クエリパラメータ

なし。

#### ボディ

ボリュームの容量を、100MiB 追加する例を次に示します。

```
{
  "capacity": 100
}
```

属性	型	説明
capacity	long	(必須) ボリュームの容量 (MiB) 追加する容量を、1~268435409 までの整数で指定します。



## レスポンスメッセージ

### ボディ

属性	型	説明
statusResource	string	容量を追加したボリュームの情報を参照するための URL の一覧

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-bin
ary @./InputParameters.json https://192.0.2.100/ConfigurationManager/simple
/v1/objects/volumes/100/actions/expand/invoke
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

## 4.5.6 ボリュームを削除する

ボリューム ID を指定して、ボリュームを削除します。

### 実行権限

ストレージ管理者 (プロビジョニング)

## リクエストライン

```
DELETE <ベース URL>/simple/v1/objects/volumes/<オブジェクト ID>
```

## リクエストメッセージ

### オブジェクト ID

ボリュームの情報取得で取得した id の値を指定します。

属性	型	説明
id	int	(必須) ボリューム ID

### クエリパラメータ

なし。

### ボディ

なし。

## レスポンスメッセージ

### ボディ

属性	型	説明
statusResource	string	削除したボリュームの情報を参照するための URL

### メモ

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H "Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X DELETE https://192.0.2.100/ConfigurationManager/simple/v1/objects/volumes/100
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

[特定のボリュームの情報を取得する \(128 ページ\)](#)

## 4.5.7 特定のボリュームの QoS の設定情報を取得する

ボリューム ID を指定して、ボリュームの QoS の設定に関する情報を取得します。

## 実行権限

ストレージ管理者（参照）

## リクエストライン

```
GET <ベース URL>/simple/v1/objects/volumes/<オブジェクト ID>/qos-setting
```

## リクエストメッセージ

### オブジェクト ID

ボリュームの情報取得で取得した id の値を指定します。

属性	型	説明
id	int	(必須) ボリューム ID

### クエリパラメータ

なし。

### ボディ

なし。

## レスポンスメッセージ

### ボディ

ボリューム ID が 100 のボリュームの QoS の設定情報を取得した例を次に示します。

```
{
  "volumeId": 100,
  "threshold": {
    "isUpperIopsEnabled": true,
    "upperIops": 2000,
    "isUpperTransferRateEnabled": true,
    "upperTransferRate": 20000,
    "isLowerIopsEnabled": true,
    "lowerIops": 1000,
    "isLowerTransferRateEnabled": true,
    "lowerTransferRate": 10000,
    "isResponsePriorityEnabled": false
  },
  "alertSetting": {
    "isUpperAlertEnabled": true,
    "upperAlertAllowableTime": 10,
    "isLowerAlertEnabled": true,
    "lowerAlertAllowableTime": 20,
    "isResponseAlertEnabled": false
  }
}
```

```

    },
    "alertTime": {}
}

```

属性	型	説明
volumeId	int	ボリューム ID
threshold	object	<p>ボリュームの QoS の設定に関する情報が出力されます。</p> <ul style="list-style-type: none"> <li>• isUpperIopsEnabled (boolean) IOPS の上限値の設定が有効かどうか <ul style="list-style-type: none"> <li>- true : 有効</li> <li>- false : 無効</li> </ul> </li> <li>• upperIops (int) IOPS の上限値 属性 isUpperIopsEnabled が true の場合に出力されます。</li> <li>• isUpperTransferRateEnabled (boolean) データ転送量の上限値の設定が有効かどうか <ul style="list-style-type: none"> <li>- true : 有効</li> <li>- false : 無効</li> </ul> </li> <li>• upperTransferRate (int) データ転送量の上限値 (MiBps) 属性 isUpperTransferRateEnabled が true の場合に出力されます。</li> <li>• isLowerIopsEnabled (boolean) IOPS の下限値の設定が有効かどうか <ul style="list-style-type: none"> <li>- true : 有効</li> <li>- false : 無効</li> </ul> </li> <li>• lowerIops (int) IOPS の下限値 属性 isLowerIopsEnabled が true の場合に出力されます。</li> <li>• isLowerTransferRateEnabled (boolean) データ転送量の下限値の設定が有効かどうか <ul style="list-style-type: none"> <li>- true : 有効</li> <li>- false : 無効</li> </ul> </li> <li>• lowerTransferRate (int) データ転送量の下限値 (MiBps) 属性 isLowerTransferRateEnabled が true の場合に出力されます。</li> <li>• isResponsePriorityEnabled (boolean) I/O 処理の優先度の設定が有効かどうか <ul style="list-style-type: none"> <li>- true : 有効</li> <li>- false : 無効</li> </ul> </li> <li>• responsePriority (int) I/O 処理の優先度 値が大きいほど優先度が高く、値が小さいほど優先度が低いことを示します。 属性 isResponsePriorityEnabled が true の場合に出力されます。</li> </ul>

属性	型	説明
		<ul style="list-style-type: none"> <li>• <code>targetResponseTime</code> (int) 応答時間の目標値 (ミリ秒) 属性 <code>isResponsePriorityEnabled</code> が <code>true</code> の場合に出力されます。</li> </ul>
<code>alertSetting</code>	<code>object</code>	<p>ボリュームのアラート設定に関する情報が出力されます。</p> <ul style="list-style-type: none"> <li>• <code>isUpperAlertEnabled</code> (boolean) IOPS またはデータ転送量が継続して上限値を上回る場合に、アラートが出力されるかどうか  <ul style="list-style-type: none"> <li>- <code>true</code> : 出力される</li> <li>- <code>false</code> : 出力されない</li> </ul> </li> <li>• <code>upperAlertAllowableTime</code> (int) 上限値を超過してからアラートが出力されるまでの許容時間 (秒) 属性 <code>isUpperAlertEnabled</code> が <code>true</code> の場合に出力されます。</li> <li>• <code>isLowerAlertEnabled</code> (boolean) IOPS またはデータ転送量が継続して下限値を下回る場合に、アラートが出力されるかどうか  <ul style="list-style-type: none"> <li>- <code>true</code> : 出力される</li> <li>- <code>false</code> : 出力されない</li> </ul> </li> <li>• <code>lowerAlertAllowableTime</code> (int) 下限値未達の場合にアラートが出力されるまでの許容時間 (秒) 属性 <code>isLowerAlertEnabled</code> が <code>true</code> の場合に出力されます。</li> <li>• <code>isResponseAlertEnabled</code> (boolean) 平均応答時間が継続して目標値未達成の場合にアラートが出力されるかどうか  <ul style="list-style-type: none"> <li>- <code>true</code> : 出力される</li> <li>- <code>false</code> : 出力されない</li> </ul> </li> <li>• <code>responseAlertAllowableTime</code> (int) 平均応答時間の目標値未達成の場合にアラートが出力されるまでの許容時間 (秒) 属性 <code>isResponseAlertEnabled</code> が <code>true</code> の場合に出力されます。</li> </ul>
<code>alertTime</code>	<code>object</code>	<p>ボリュームのアラート発生時刻※に関する情報が出力されます。</p> <ul style="list-style-type: none"> <li>• <code>upperAlertTime</code> (ISO8601string) IOPS またはデータ転送量が継続して上限値を上回る場合のアラート最終発生時刻 (UTC) アラートが出力される設定をしていない、またはアラートが発生していない場合、この属性は出力されません。</li> <li>• <code>lowerAlertTime</code> (ISO8601string) IOPS またはデータ転送量が継続して下限値を下回る場合のアラート最終発生時刻 (UTC)</li> </ul>

属性	型	説明
		<p>アラートが出力される設定をしていない、またはアラートが発生していない場合、この属性は出力されません。</p> <ul style="list-style-type: none"> <li>responseAlertTime (ISO8601string)</li> </ul> <p>平均応答時間が継続して目標値未達成の場合のアラート最終発生時刻 (UTC)</p> <p>アラートが出力される設定をしていない、またはアラートが発生していない場合、この属性は出力されません。</p>

**注※**

ストレージシステムのタイムゾーンを変更すると、タイムゾーンを変更する前のアラート発生時刻が不正な値となります。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192
.0.2.100/ConfigurationManager/simple/v1/objects/volumes/100/qos-setting
```

### —— 関連リンク ——

[HTTP ステータスコード \(13 ページ\)](#)

[ボリュームの情報を取得する \(124 ページ\)](#)

## 4.5.8 ボリュームの QoS の設定を変更する

ボリューム ID を指定して QoS の設定を変更します。

## 実行権限

ストレージ管理者 (システムリソース管理)

## リクエストライン

```
PATCH <ベース URL>/simple/v1/objects/volumes/<オブジェクト ID>/qos-setting
```

## リクエストメッセージ

### オブジェクト ID

ボリュームの情報取得で取得した id の値を指定します。

属性	型	説明
id	int	(必須) ボリューム ID

### クエリパラメータ

なし。

### ボディ

属性 threshold と属性 alertSetting は同時に指定できません。

指定した属性だけが変更されます。

QoS の設定（属性 threshold）を次のとおり指定して変更する例を次に示します。

- isUpperIopsEnabled : true
- upperIops (IOPS の上限値) : 2000
- isLowerIopsEnabled : true
- lowerIops (IOPS の下限値) : 1000
- isResponsePriorityEnabled : true
- responsePriority (I/O 処理の優先度) : 3 (高い)

```
{
  "threshold": {
    "isUpperIopsEnabled": true,
    "upperIops": 2000,
    "isLowerIopsEnabled": true,
    "lowerIops": 1000,
    "isResponsePriorityEnabled": true,
    "responsePriority": 3
  }
}
```

アラートの設定（属性 alertSetting）を次のとおり指定して変更する例を次に示します。

- isUpperAlertEnabled : true
- upperAlertAllowableTime (アラート出力までの許容時間) : 10
- isLowerAlertEnabled : true
- lowerAlertAllowableTime (アラート出力までの許容時間) : 20
- isResponseAlertEnabled : true

- responseAlertAllowableTime (アラート出力までの許容時間): 30

```
{
  "alertSetting": {
    "isUpperAlertEnabled": true,
    "upperAlertAllowableTime": 10,
    "isLowerAlertEnabled": true,
    "lowerAlertAllowableTime": 20,
    "isResponseAlertEnabled": true,
    "responseAlertAllowableTime": 30
  }
}
```

属性	型	説明
threshold	object	<p>ボリュームの QoS の設定に関する情報</p> <ul style="list-style-type: none"> <li>• isUpperIopsEnabled (boolean) (任意) IOPS の上限値の設定を有効にするかどうかを指定します。 <ul style="list-style-type: none"> <li>- true: 有効にする</li> <li>- false: 無効にする</li> </ul> この属性に true を指定する場合、属性 upperIops も必ず指定してください。</li> <li>• upperIops (int) (任意) IOPS の上限値 100～2147483647 までの整数で指定します。 この属性を指定する場合、属性 isUpperIopsEnabled に true を指定してください。</li> <li>• isUpperTransferRateEnabled (boolean) (任意) データ転送量の上限値の設定を有効にするかどうかを指定します。 <ul style="list-style-type: none"> <li>- true: 有効にする</li> <li>- false: 無効にする</li> </ul> この属性に true を指定する場合、属性 upperTransferRate も必ず指定してください。</li> <li>• upperTransferRate (int) (任意) データ転送量の上限値 (MiBps) 1～2097151 までの整数で指定します。 この属性を指定する場合、属性 isUpperTransferRateEnabled に true を指定してください。</li> <li>• isLowerIopsEnabled (boolean) (任意) IOPS の下限値の設定を有効にするかどうかを指定します。 <ul style="list-style-type: none"> <li>- true: 有効にする</li> <li>- false: 無効にする</li> </ul> この属性に true を指定する場合、属性 lowerIops も必ず指定してください。</li> <li>• lowerIops (int) (任意) IOPS の下限値の設定 10～2147483647 までの整数で指定します。</li> </ul>



属性	型	説明
		<p>この属性を指定する場合、属性 <code>isLowerIopsEnabled</code> に <code>true</code> を指定してください。</p> <ul style="list-style-type: none"> <li>• <code>isLowerTransferRateEnabled</code> (boolean) (任意) データ転送量の下限値の設定を有効にするかどうかを指定します。 <ul style="list-style-type: none"> <li>- <code>true</code> : 有効にする</li> <li>- <code>false</code> : 無効にする</li> </ul> </li> </ul> <p>この属性に <code>true</code> を指定する場合、属性 <code>lowerTransferRate</code> も必ず指定してください。</p> <ul style="list-style-type: none"> <li>• <code>lowerTransferRate</code> (int) (任意) データ転送量の下限値 (MiBps) 1~2097151 までの整数で指定します。</li> </ul> <p>この属性を指定する場合、属性 <code>isLowerTransferRateEnabled</code> に <code>true</code> を指定してください。</p> <ul style="list-style-type: none"> <li>• <code>isResponsePriorityEnabled</code> (boolean) (任意) I/O 処理の優先度の設定を有効にするかどうかを指定します。 <ul style="list-style-type: none"> <li>- <code>true</code> : 有効にする</li> <li>- <code>false</code> : 無効にする</li> </ul> </li> </ul> <p>この属性に <code>true</code> を指定する場合、属性 <code>responsePriority</code> も必ず指定してください。</p> <ul style="list-style-type: none"> <li>• <code>responsePriority</code> (int) (任意) I/O 処理の優先度 1~3 までの整数で指定します。</li> </ul> <p>値が大きいくほど、優先度が高くなります。</p> <p>この属性を指定した場合、応答時間の目標値が自動で設定されます。</p> <p>この属性を指定する場合、属性 <code>isResponsePriorityEnabled</code> に <code>true</code> を指定してください。</p>
<code>alertSetting</code>	<code>object</code>	<p>ボリュームのアラート設定に関する情報</p> <ul style="list-style-type: none"> <li>• <code>isUpperAlertEnabled</code> (boolean) (任意) IOPS またはデータ転送量が継続して上限値を上回る場合にアラートの出力を有効にするかどうかを指定します。 <ul style="list-style-type: none"> <li>- <code>true</code> : 有効にする</li> <li>- <code>false</code> : 無効にする</li> </ul> </li> </ul> <p>この属性は、IOPS またはデータ転送量に上限値を設定している場合に有効にできます。</p> <p>この属性に <code>true</code> を指定する場合、属性 <code>upperAlertAllowableTime</code> も必ず指定してください。</p> <ul style="list-style-type: none"> <li>• <code>upperAlertAllowableTime</code> (int) (任意) IOPS またはデータ転送量の上限値を超過してからアラートが出力されるまでの許容時間 (秒) 1~600 までの整数で指定します。</li> </ul> <p>この属性を指定する場合、属性 <code>isUpperAlertEnabled</code> に <code>true</code> を指定してください。</p>

属性	型	説明
		<ul style="list-style-type: none"> <li>• <code>isLowerAlertEnabled</code> (boolean)  (任意) IOPS またはデータ転送量が継続して下限値を下回る場合にアラートの出力を有効にするかどうかを指定します。  - <code>true</code>: 有効にする  - <code>false</code>: 無効にする  この属性は、IOPS またはデータ転送量に下限値を設定している場合に有効にできます。  この属性に <code>true</code> を指定する場合、属性 <code>lowerAlertAllowableTime</code> も必ず指定してください。</li> <li>• <code>lowerAlertAllowableTime</code> (int)  (任意) IOPS またはデータ転送量の下限値未達の場合にアラートが出力されるまでの許容時間 (秒)  1~600 までの整数で指定します。  この属性を指定する場合、属性 <code>isLowerAlertEnabled</code> に <code>true</code> を指定してください。</li> <li>• <code>isResponseAlertEnabled</code> (boolean)  (任意) 平均応答時間が継続して目標値未達成の場合にアラートの出力を有効にするかどうかを指定します。  - <code>true</code>: 有効にする  - <code>false</code>: 無効にする  この属性は、平均応答時間の目標値を設定している場合に有効にできます。  この属性に <code>true</code> を指定する場合、属性 <code>responseAlertAllowableTime</code> も必ず指定してください。</li> <li>• <code>responseAlertAllowableTime</code> (int)  (任意) 応答時間の目標値未達の場合にアラートが出力されるまでの許容時間 (秒)  1~600 までの整数で指定します。  この属性を指定する場合、<code>isResponseAlertEnabled</code> に <code>true</code> を指定してください。</li> </ul>

## レスポンスメッセージ

### ボディ

属性	型	説明
<code>statusResource</code>	<code>string</code>	変更したボリュームの情報を参照するための URL

### メモ

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H "Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X PATCH --data-binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/simple/v1/objects/volumes/100/qos-setting
```

---

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[commandStatus オブジェクト \(19 ページ\)](#)

---

## 4.5.9 ボリュームとサーバの接続情報を取得する

クエリパラメータにサーバ ID、またはサーバのニックネームを指定して、ボリュームとサーバの接続情報の一覧を取得します。

## 実行権限

ストレージ管理者 (参照)

## リクエストライン

```
GET <ベース URL>/simple/v1/objects/volume-server-connections
```

## リクエストメッセージ

### オブジェクト ID

なし。

### クエリパラメータ

1 回のリクエストで取得できるボリュームとサーバの接続情報は最大で 2048 個です。2049 個以上のボリュームとサーバの接続情報を取得する場合は、count パラメータと startVolumeId パラメータを組み合わせる複数回に分けて API を実行してください。count パラメータを指定することで取得するボリュームとサーバの接続情報をフィルタリングすることもできます。

パラメータ	型	フィルタ条件
serverId	int	(任意) ボリュームと接続されているサーバのサーバ ID を指定します。 指定したサーバ ID の接続情報を取得します。 serverId または serverNickname のどちらか一方を必ず指定してください。また、これらのパラメータを同時には指定しないでください。
serverNickname	string	(任意) ボリュームと接続されているサーバのニックネームを指定します。 指定したサーバのニックネームの接続情報を取得します。 serverId または serverNickname のどちらか一方を必ず指定してください。また、これらのパラメータを同時には指定しないでください。
startVolumeId	int	(任意) ボリュームとサーバの接続情報の取得を開始するボリューム ID を 0～65279 の値で指定します。
count	int	(任意) ボリュームとサーバの接続情報を取得する個数を 1～2048 の値で指定します。 省略した場合、2048 が指定されたと見なされます。

## ボディ

なし。

## レスポンスメッセージ

## ボディ

サーバ ID が 10（クエリパラメータ serverId に 10 を指定）のサーバのボリュームとサーバの接続情報を取得した出力例を次に示します。

```

{
  "data": [
    {
      "id": "100,10",
      "volumeId": 100,
      "serverId": 10,
      "luns": [
        {
          "lun": 1,
          "portId": "CL1-A"
        }
      ]
    },
    {
      "id": "101,10",
      "volumeId": 101,
      "serverId": 10,
      "luns": [
        {
          "lun": 2,
          "portId": "CL1-A"
        }
      ]
    }
  ]
}

```

```

    }
  ]
}
],
"count": 2,
"totalCount": 2,
"hasNext": false
}

```

属性	型	説明
id	string	ボリューム ID とサーバ ID がコンマで連結した形式で出力されます。
volumeId	int	ボリューム ID
serverId	int	サーバ ID
luns	object[]	LUN の設定情報 <ul style="list-style-type: none"> <li>lun (int) LU 番号</li> <li>portId (string) 割り当て先のポート ID</li> </ul>

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```

curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192
.0.2.100/ConfigurationManager/simple/v1/objects/volume-server-connections?s
erverId=10

```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[データオブジェクト \(17 ページ\)](#)

## 4.5.10 ボリュームと特定のサーバの接続情報を取得する

ボリューム ID とサーバ ID を指定して、ボリュームと特定のサーバの接続情報を取得します。

## 実行権限

ストレージ管理者 (参照)

## リクエストライン

```
GET <ベース URL>/simple/v1/objects/volume-server-connections/<オブジェクト ID>
```

## リクエストメッセージ

### オブジェクト ID

ボリュームとサーバの接続情報取得で取得した id の値を指定します。取得した属性 id の値は、ボリューム ID およびサーバ ID をコンマで連結した形式で出力されます。ボリュームとサーバの接続情報取得で取得した volumeId の値と serverId の値をコンマで連結した形式でも指定できます。

```
< volumeId >,< serverId >
```

属性	型	説明
volumeId	int	(必須) ボリューム ID
serverId	int	(必須) サーバ ID

### クエリパラメータ

なし。

### ボディ

なし。

## レスポンスメッセージ

### ボディ

ボリューム ID が 100 のボリュームと、サーバ ID が 10 のサーバの接続情報を取得する出力例を次に示します。

```
{
  "id": "100,10",
  "volumeId": 100,
  "serverId": 10,
  "luns": [
    {
      "lun": 1,
      "portId": "CL1-A"
    }
  ]
}
```

取得する属性については、ボリュームとサーバの接続情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H "Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192.0.2.100/ConfigurationManager/simple/v1/objects/volume-server-connections/100,10
```

---

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[ボリュームとサーバの接続情報を取得する \(147 ページ\)](#)

---

## 4.5.11 ボリュームとサーバを接続する

サーバとボリューム間のすべてのパスに対して接続設定します。

### メモ

仮想ストレージマシンに割り当てられたホストグループ（または iSCSI ターゲット）が追加されているサーバと、ボリューム間のパスに対して接続設定はできません。

---

## 実行権限

ストレージ管理者（プロビジョニング）

## リクエストライン

```
POST <ベース URL>/simple/v1/objects/volume-server-connections
```

## リクエストメッセージ

### オブジェクト ID

なし。

### クエリパラメータ

なし。

## ボディ

ボリューム ID が 100 のボリュームと、サーバ ID が 10 のサーバ間のパスに対して接続設定をする例を次に示します。

```
{
  "volumeIds": [
    100
  ],
  "serverIds": [
    10
  ]
}
```

属性	型	説明
volumeIds	int[]	(必須) 割り当て対象のボリューム ID を 0～65279 までの値で指定します。
serverIds	int[]	(必須) 割り当てサーバ ID ボリュームを割り当てる対象のサーバ ID を 0～4095 までの値ですべて指定します。

## レスポンスメッセージ

### ボディ

属性	型	説明
statusResource	string	サーバと接続したボリュームの実行結果の問い合わせに使用する URL

### メモ

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-bin
ary @./InputParameters.json https://192.0.2.100/ConfigurationManager/simple
/v1/objects/volume-server-connections
```



---

**関連リンク**


---

[HTTP ステータスコード \(13 ページ\)](#)
[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)


---

## 4.5.12 ボリュームと特定のサーバの接続を解除する

ボリューム ID とサーバ ID を指定して、特定のサーバとボリューム間に登録されているすべてのパスに対して接続設定を解除します。

### 実行権限

ストレージ管理者 (プロビジョニング)

### リクエストライン

```
DELETE <ベース URL>/simple/v1/objects/volume-server-connections/<オブジェクト ID>
```

### リクエストメッセージ

#### オブジェクト ID

ボリュームとサーバの接続情報取得で取得した id の値を指定します。取得した属性 id の値は、ボリューム ID およびサーバ ID をコンマで連結した形式で出力されます。ボリュームとサーバの接続情報取得で取得した volumeId の値と serverId の値をコンマで連結した形式でも指定できます。

```
< volumeId > , < serverId >
```

属性	型	説明
volumeId	int	(必須) ボリューム ID
serverId	int	(必須) サーバ ID

#### クエリパラメータ

なし。

#### ボディ

なし。

## レスポンスメッセージ

### ボディ

属性	型	説明
statusResource	string	サーバとの接続を解除したボリュームの実行結果の問い合わせに使用する URL

### メモ

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H "Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X DELETE https://192.0.2.100/ConfigurationManager/simple/v1/objects/volume-server-connections/100,10
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

## 第5章

# スナップショットの操作

この章では、ストレージシステム内のボリュームのスナップショットを作成してコストパフォーマンスの良い複製を作成したり、スナップショットをマッピングしてボリュームが使用できる状態にする API などの操作について説明します。

### 5.1 スナップショットの流れ

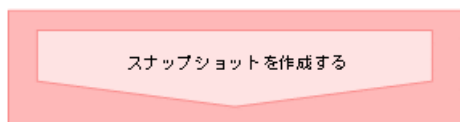
スナップショットを作成してリストアするまでの流れを説明します。

#### スナップショットによるバックアップの流れ

データをバックアップするために、ボリュームのスナップショットを作成します。スナップショットは特定の時点のボリュームのイメージです。スナップショットを作成したタイミングで、元のボリュームとの差分データがプールに格納されます。スナップショットを作成してリストアするまでの流れを説明します。

スナップショットに関する機能については、マニュアル『Snapshot Advanced ユーザガイド』を参照してください。

#### スナップショットを作成する



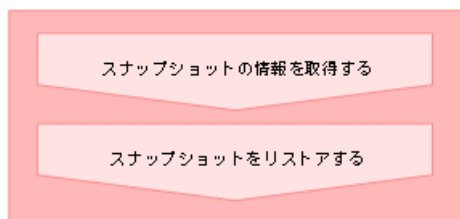
(凡例)

： REST API で操作する手順

- スナップショットを作成する

データをバックアップ、またはバックアップしたデータを二次的に利用するためのスナップショットを作成します。

#### スナップショットをリストアする



(凡例)

： REST API で操作する手順

- スナップショットの情報を取得する

スナップショットに関する情報を取得します。

- スナップショットをリストアする

スナップショットの作成元ボリューム ID、およびスナップショット ID を指定してスナップショットをリストアします。リストアすると、指定した作成元ボリュームのスナップショットデータを上書きします。

### スナップショットを削除する



- スナップショットの情報を取得する

スナップショットに関する情報を取得します。

- スナップショットを削除する

スナップショットの作成元ボリューム ID、およびスナップショット ID を指定してスナップショットを削除します。

### メモ

スナップショットの作成日時は、協定世界時（UTC）に基づく日時が返ります。

## 5.2 スナップショットグループ名の一覧を取得する

スナップショットグループ名の一覧を取得します。

### 実行権限

ストレージ管理者（参照）

### リクエストライン

```
GET <ベース URL>/simple/v1/objects/snapshot-groups
```

## リクエストメッセージ

### オブジェクト ID

なし。

### クエリパラメータ

なし。

### ボディ

なし。

## レスポンスメッセージ

### ボディ

```
{
  "data": [
    {
      "name": "snapshotGroup"
    },
    {
      "name": "snapshotGroup2"
    }
  ],
  "count": 2
}
```

属性	型	説明
name	string	スナップショットグループ名

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192
.0.2.100/ConfigurationManager/simple/v1/objects/snapshot-groups
```

---

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[データオブジェクト \(17 ページ\)](#)

---

## 5.3 特定のスナップショットグループの情報を取得する

スナップショットグループ名を指定して、スナップショットグループの情報を取得します。

### 実行権限

ストレージ管理者（参照）

### リクエストライン

```
GET <ベース URL>/simple/v1/objects/snapshot-groups/<オブジェクト ID>
```

### リクエストメッセージ

#### オブジェクト ID

スナップショットグループ名の一覧取得で取得した name の値を指定します。

属性	型	説明
name	string	(必須) スナップショットグループ名

#### メモ

次に示すスナップショットグループ名は指定できません。

- スラッシュまたはバックスラッシュを含むスナップショットグループ名
- ピリオド (.) のみのスナップショットグループ名

該当のスナップショットグループの情報を取得する場合は、クエリパラメータ snapshotGroupName を指定して、スナップショットの情報を取得する API を実行することでスナップショットグループの情報が取得できます。

#### クエリパラメータ

1 回のリクエストで取得できるスナップショットグループ内のスナップショットの情報は最大で 1000 個です。1001 個以上のスナップショットを取得する場合は、count パラメータと startSnapshotId パラメータを組み合わせる複数回に分けて API を実行してください。count パラメータを指定することで取得するスナップショットをフィルタリングすることもできます。

パラメータ	型	フィルタ条件
startSnapshotId	string	(任意) 取得を開始するスナップショットの ID を指定します。 このパラメータを省略した場合、"0,0" が指定されたと見なされます。

パラメータ	型	フィルタ条件
count	int	(任意) 取得するスナップショットの個数を 1～1000 の値で指定します。 このパラメータを省略した場合、1000 が指定されたと見なされます。

## ボディ

なし。

## レスポンスメッセージ

### ボディ

スナップショットグループ名が snapshotGroup のスナップショットグループの情報を取得した出力例を次に示します。

```
{
  "name": "snapshotGroup",
  "snapshots": [
    {
      "masterVolumeId": 100,
      "snapshotId": 3
    },
    {
      "masterVolumeId": 100,
      "snapshotId": 4
    }
  ],
  "count": 2,
  "totalCount": 2,
  "hasNext": false
}
```

属性	型	説明
name	string	スナップショットグループ名
snapshots	object[]	スナップショットに関する情報が出力されます。 <ul style="list-style-type: none"> <li>masterVolumeId (int) スナップショットの作成元ボリューム ID</li> <li>snapshotId (int) スナップショット ID</li> <li>MU (ミラーユニット) 番号が出力されます。</li> </ul>
count	int	スナップショットの情報の格納数
totalCount	int	スナップショットグループ内のスナップショット総数
hasNext	boolean	取得できていない情報があるかどうか出力されます。 <ul style="list-style-type: none"> <li>true : 取得できていない情報がある</li> <li>false : すべての情報が取得できている</li> </ul> 取得できていない情報がある場合、クエリパラメータを指定して取得する情報をフィルタリングするか、取得で

属性	型	説明
		きている情報のうち、最大のスナップショットの ID 以降を複数回に分けて API を実行し取得してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H "Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192.0.2.100/ConfigurationManager/simple/v1/objects/snapshot-groups/snapshotGroup
```

---

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[スナップショットグループ名の一覧を取得する \(156 ページ\)](#)

[スナップショットの情報を取得する \(160 ページ\)](#)

---

## 5.4 スナップショットの情報を取得する

スナップショットに関する情報を取得します。スナップショットの条件を指定して、絞り込んだ情報を取得することもできます。

### 実行権限

ストレージ管理者 (参照)

### リクエストライン

```
GET <ベース URL>/simple/v1/objects/snapshots
```

### リクエストメッセージ

#### オブジェクト ID

なし。



## クエリパラメータ

1 回のリクエストで取得できるスナップショットの情報は最大で 1000 個です。1001 個以上のスナップショットの情報を取得する場合は、count パラメータと startId パラメータを組み合わせて複数回に分けて API を実行してください。count パラメータを指定することで取得するボリュームの情報をフィルタリングすることもできます。

パラメータ	型	フィルタ条件
masterVolumeId	int	(任意) スナップショットの作成元ボリューム ID
snapshotDateFrom	ISO8601string	(任意) 取得するスナップショットの日時を YYYY-MM-DDThh:mm:ssZ 形式で指定します。 このパラメータで指定された日時以降のスナップショットの情報を取得します。
snapshotDateTo	ISO8601string	(任意) 取得するスナップショットの日時を YYYY-MM-DDThh:mm:ssZ 形式で指定します。 このパラメータで指定された日時以前のスナップショットの情報を取得します。
snapshotGroupName	string	(任意) スナップショットグループ名 指定した値と完全に一致するスナップショットの情報が取得されます。
startId	string	(任意) 取得を開始するスナップショットの情報を、スナップショットの作成元ボリューム ID とスナップショット ID をコンマで連結した形式で指定します。 <masterVolumeId>,<snapshotId> このパラメータを省略した場合、"0,0"が指定されたと見なされます。
count	int	(任意) 取得するスナップショットの情報の個数を 1～1000 の値で指定します。 このパラメータを省略した場合、1000 が指定されたと見なされます。

## ボディ

なし。

## レスポンスメッセージ

### ボディ

```
{
  "data": [
    {
      "id": "100,3",
      "masterVolumeId": 100,
      "snapshotId": 3,
      "status": "Completed",
      "snapshotDate": "2015-03-20T09:27:35Z",
      "snapshotGroupName": "snapshotGroup",
      "rootVolumeId": 100,
      "type": "Snapshot",

```

```

    "poolId": 10,
    "usedCapacityPerRootVolume": 126,
    "isVolumeCapacityExpanding": false,
    "retentionPeriod": 10
  },
  {
    "id": "100,4",
    "masterVolumeId": 100,
    "snapshotId": 4,
    "status": "Completed",
    "snapshotDate": "2015-03-20T09:27:35Z",
    "snapshotGroupName": "snapshotGroup",
    "rootVolumeId": 100,
    "type": "Snapshot",
    "poolId": 10,
    "usedCapacityPerRootVolume": 126,
    "isVolumeCapacityExpanding": false,
    "retentionPeriod": 10
  },
],
"count": 2,
"totalCount": 2,
"hasNext": false
}

```

属性	型	説明
id	string	スナップショットの作成元ボリューム ID、およびスナップショット ID がコンマで連結した形式で出力されます。
masterVolumeId	int	スナップショットの作成元ボリューム ID
snapshotId	int	スナップショット ID MU（ミラーユニット）番号が出力されます。
status	string	スナップショットの状態 <ul style="list-style-type: none"> <li>Creating：スナップショット作成中</li> <li>In Sync：スナップショット同期処理完了</li> <li>Splitting：スナップショット分割中</li> <li>Completed：スナップショット作成済</li> <li>Restoring：スナップショットリストア中</li> <li>Deleting：スナップショット削除中</li> <li>Relocating：スナップショット再配置中</li> <li>Error：障害が発生している</li> </ul>
snapshotDate	ISO8601string	スナップショットの作成日時
snapshotGroupName	string	スナップショットグループ名
mappedVolumeId	int	スナップショットの作成先ボリューム ID スナップショットの作成先ボリューム ID が定義されている場合に出力されます。
rootVolumeId	int	スナップショットのルートボリューム ID
poolId	int	スナップショットの差分データが格納されているプールのプール ID
usedCapacityPerRootVolume	long	スナップショットの差分データが格納されているプールにおいて、同じボリュームから作成されたスナップ

属性	型	説明
		ショット群が使用している差分情報と制御情報の総容量 (MiB)
isVolumeCapacityExpanding	boolean	スナップショットの容量拡張の状態を示すフラグ <ul style="list-style-type: none"> <li>• true: 容量拡張の処理中である スナップショットにボリュームをマッピングしている場合、スナップショットの作成元ボリュームとスナップショットのマッピング先ボリュームの容量が異なるときに true (容量拡張の処理中) となります。 スナップショットにボリュームをマッピングしていない場合、スナップショットの作成元ボリュームが容量拡張の処理中のときだけ true (容量拡張の処理中) となります。</li> <li>• false: 容量拡張の処理中でない</li> </ul>
type	string	スナップショットのタイプ <ul style="list-style-type: none"> <li>• Snapshot: スナップショット、またはボリューム ID が定義されているスナップショット</li> </ul>
retentionPeriod	int	スナップショットの残り保護時間 (hours) スナップショットデータ保護期間が設定されていない、またはスナップショットデータ保護期間が満了している場合は 0 が出力されます。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H "Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192.0.2.100/ConfigurationManager/simple/v1/objects/snapshots
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[データオブジェクト \(17 ページ\)](#)

## 5.5 特定のスナップショットの情報を取得する

スナップショットの作成元ボリューム ID、およびスナップショット ID を指定して、スナップショットに関する情報を取得します。

### 実行権限

ストレージ管理者 (参照)

## リクエストライン

```
GET <ベース URL>/simple/v1/objects/snapshots/<オブジェクト ID>
```

## リクエストメッセージ

### オブジェクト ID

スナップショットの情報取得で取得した id の値を指定します。取得した属性 id の値は、スナップショットの作成元ボリューム ID およびスナップショット ID をコンマで連結した形式で取得されます。スナップショットの情報取得で取得した masterVolumeId の値と snapshotId の値をコンマで連結した形式でも指定できます。

```
< masterVolumeId >, < snapshotId >
```

属性	型	説明
masterVolumeId	int	(必須) スナップショットの作成元ボリューム ID
snapshotId	int	(必須) スナップショット ID

### クエリパラメータ

なし。

### ボディ

なし。

## レスポンスメッセージ

### ボディ

特定のスナップショット（スナップショットの作成元ボリューム ID：100、スナップショット ID：3）の情報を取得した出力例を次に示します。

```
{
  "id": "100,3",
  "masterVolumeId": 100,
  "snapshotId": 3,
  "status": "Completed",
  "snapshotDate": "2015-03-20T09:27:35Z",
  "snapshotGroupName": "snapshotGroup",
  "rootVolumeId": 100,
  "type": "Snapshot",
  "poolId": 10,
  "usedCapacityPerRootVolume": 126,
  "isVolumeCapacityExpanding": false,
  "retentionPeriod": 10
}
```

取得する属性については、スナップショットの情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H "Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192.0.2.100/ConfigurationManager/simple/v1/objects/snapshots/100,3
```

---

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[スナップショットの情報を取得する \(160 ページ\)](#)

---

## 5.6 スナップショットを作成する

データをバックアップするためのスナップショットを作成します。

### 実行権限

ストレージ管理者（ローカルバックアップ管理）

### リクエストライン

```
POST <ベース URL>/simple/v1/objects/snapshots
```

### リクエストメッセージ

#### オブジェクト ID

なし。

#### クエリパラメータ

なし。

## ボディ

プール ID が 13 のプールに、スナップショットの作成元ボリューム（ボリューム ID : 100、スナップショットグループ名 : snapshotGroup）からスナップショットを作成する例を次に示します。

```
{
  "params": [
    {
      "masterVolumeId": 100,
      "poolId": 13,
      "snapshotGroupName": "snapshotGroup",
      "type": "Snapshot"
    }
  ]
}
```

プール ID が 13 のプールに、スナップショットの作成元ボリューム（ボリューム ID : 100、スナップショットグループ名 : snapshotGroup）からスナップショットを作成し、スナップショットの作成先ボリュームにボリューム ID を定義する例を次に示します。

```
{
  "params": [
    {
      "masterVolumeId": 100,
      "poolId": 13,
      "snapshotGroupName": "snapshotGroup",
      "type": "Mapped Snapshot"
    }
  ]
}
```

属性	型	説明
params	object[]	<p>作成するスナップショットの情報</p> <ul style="list-style-type: none"> <li>masterVolumeId (int) (必須) スナップショットの作成元ボリューム ID を 0 ~ 65279 の整数で指定します。</li> <li>poolId (int) (必須) スナップショットの差分データを格納するプールのプール ID を 0 ~ 127 の整数で指定します。 プールの使用率が枯渇しきい値を超えると作成済みのものを含むスナップショットが使用できなくなるおそれがあります。指定するプールに十分な空き容量があることを確認してください。 プールの使用率を確認するには、プール ID を指定して特定のプールの情報を取得する API を実行してください。 スナップショットと削減データを共有するボリュームの場合、ボリュームが属するプール ID を指定してください。 スナップショットと削減データを共有するボリュームの詳細については『Snapshot Advanced ユーザガイド』を参照してください。</li> <li>snapshotGroupName (string)</li> </ul>

属性	型	説明
		<p>(必須) スナップショットグループ名を 1~32 文字で指定します。</p> <p>使用できる文字は次のとおりです。</p> <p>0~9 A~Z a~z , - . / : @ \ _ 半角スペース</p> <p>英字は、大文字と小文字を区別します。</p> <p>半角スペースは、使用できる文字と文字の間には使用できますが、先頭と末尾には使用できません。</p> <p>また、詳細な情報を取得したり、構成変更する操作に対応した API (リクエストラインに「simple」を含まない API) を使用する場合、スナップショットグループ名に半角スペースが含まれていると、スナップショットグループ名およびスナップショットに関する情報を取得できません。</p> <p>名前の先頭にハイフン (-) は使用できません。</p> <p>新規のスナップショットグループ名を指定すると、同時にスナップショットグループも作成します。</p> <ul style="list-style-type: none"> <li>• type (string)</li> </ul> <p>(必須) スナップショットのタイプ</p> <ul style="list-style-type: none"> <li>- Snapshot : スナップショットを作成する</li> <li>- Mapped Snapshot : スナップショットを作成してスナップショットにボリューム ID を定義する</li> </ul> <p>属性値に Mapped Snapshot を指定した場合、次の形式でスナップショットの作成先ボリュームにニックネームが自動で付与されます。</p> <p><b>Mapped Snapshot を指定した場合 :</b></p> <p>「Snapshot of ID: 作成元ボリューム ID」</p>

## レスポンスメッセージ

### ボディ

属性	型	説明
statusResource	string	作成したスナップショットの実行結果の問い合わせに使用する URL

### メモ

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-bin
ary @./InputParameters.json https://192.0.2.100/ConfigurationManager/simple
/v1/objects/snapshots
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

[特定のプールの情報を取得する \(55 ページ\)](#)

## 5.7 スナップショットをマッピングする

スナップショットの作成元ボリューム ID、およびスナップショット ID を指定してスナップショットをマッピングします。マッピングすると、指定したスナップショットの作成先ボリュームが使用できる状態になります。

### 実行権限

ストレージ管理者（ローカルバックアップ管理）

### リクエストライン

```
POST <ベース URL>/simple/v1/objects/snapshots/<オブジェクト ID>/actions/map/
invoke
```

### リクエストメッセージ

#### オブジェクト ID

スナップショットの情報取得で取得した id の値を指定します。取得した属性 id の値は、スナップショットの作成元ボリューム ID およびスナップショット ID をコンマで連結した形式で取得されます。スナップショットの情報取得で取得した masterVolumeId の値と snapshotId の値をコンマで連結した形式でも指定できます。

```
< masterVolumeId > , < snapshotId >
```

属性	型	説明
masterVolumeId	int	(必須) スナップショットの作成元ボリューム ID
snapshotId	int	(必須) スナップショット ID



## クエリパラメータ

なし。

## ボディ

```
{
  "poolId": 13
}
```

属性	型	説明
poolId	int	(必須) スナップショットをマッピングするボリュームの作成元プールのプール ID を 0～127 までの整数で指定します。

## メモ

スナップショットをマッピングすると、次の形式でスナップショットの作成先ボリュームにニックネームが自動で付与されます。

「Snapshot of ID: 作成元ボリューム ID」

## レスポンスメッセージ

## ボディ

属性	型	説明
statusResource	string	マッピングしたスナップショットの作成先ボリュームの実行結果の問い合わせに使用する URL

## メモ

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-bin
ary @./InputParameters.json https://192.0.2.100/ConfigurationManager/simple
/v1/objects/snapshots/100,3/actions/map/invoke
```

---

**関連リンク**

---

[HTTP ステータスコード \(13 ページ\)](#)[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

---

## 5.8 スナップショットをリストアする

スナップショットの作成元ボリューム ID、およびスナップショット ID を指定してスナップショットをリストアします。リストアすると、指定した作成元ボリュームのスナップショットデータを上書きします。

### 実行権限

ストレージ管理者（ローカルバックアップ管理）

### リクエストライン

```
POST <ベース URL>/simple/v1/objects/snapshots/<オブジェクト ID>/actions/rest
ore/invoke
```

### リクエストメッセージ

#### オブジェクト ID

スナップショットの情報取得で取得した id の値を指定します。取得した属性 id の値は、スナップショットの作成元ボリューム ID およびスナップショット ID をコンマで連結した形式で取得されます。スナップショットの情報取得で取得した masterVolumeId の値と snapshotId の値をコンマで連結した形式でも指定できます。

```
< masterVolumeId > , < snapshotId >
```

属性	型	説明
masterVolumeId	int	(必須) スナップショットの作成元ボリューム ID
snapshotId	int	(必須) スナップショット ID

#### クエリパラメータ

なし。

#### ボディ

なし。

## レスポンスメッセージ

### ボディ

属性	型	説明
statusResource	string	リストアしたスナップショットの情報の実行結果の問い合わせに使用する URL

### メモ

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST https://19
2.0.2.100/ConfigurationManager/simple/v1/objects/snapshots/100,3/actions/re
store/invoke -d ""
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

## 5.9 スナップショットグループを削除する

スナップショットグループ名を指定して、スナップショットグループおよびスナップショットグループ内のすべてのスナップショットを削除します。

### 実行権限

ストレージ管理者（ローカルバックアップ管理）

### リクエストライン

```
DELETE <ベース URL>/simple/v1/objects/snapshot-groups/<オブジェクト ID>
```

## リクエストメッセージ

### オブジェクト ID

スナップショットグループ名の一覧取得で取得した `name` の値を指定します。

属性	型	説明
<code>name</code>	<code>string</code>	(必須) スナップショットグループ名

### メモ

スラッシュまたはバックスラッシュを含むスナップショットグループ名は指定できません。

該当するスナップショットグループ名のスナップショットグループを削除する場合、スナップショットを削除する API を実行してください。

スナップショットグループに含まれるすべてのスナップショットを削除すると、スナップショットグループは自動的に削除されます。

### クエリパラメータ

なし。

### ボディ

なし。

## レスポンスメッセージ

### ボディ

属性	型	説明
<code>statusResource</code>	<code>string</code>	削除したスナップショットグループの実行結果の問い合わせに使用する URL

### メモ

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

# コード例


```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X DELETE https://
192.0.2.100/ConfigurationManager/simple/v1/objects/snapshot-groups/snapshot
Group
```

— 関連リンク —

[HTTP ステータスコード \(13 ページ\)](#)  
[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)  
[スナップショットを削除する \(173 ページ\)](#)

## 5.10 スナップショットを削除する

スナップショットの作成元ボリューム ID、およびスナップショット ID を指定してスナップショットを削除します。

 **注意**

スナップショットに保護期間が設定されている場合は、スナップショットを削除できません。

# 実行権限

ストレージ管理者（ローカルバックアップ管理）

# リクエストライン

```
DELETE <ベース URL>/simple/v1/objects/snapshots/<オブジェクト ID>
```

# リクエストメッセージ

## オブジェクト ID

スナップショットの情報取得で取得した id の値を指定します。取得した属性 id の値は、スナップショットの作成元ボリューム ID およびスナップショット ID をコンマで連結した形式で取得されます。スナップショットの情報取得で取得した masterVolumeId の値と snapshotId の値をコンマで連結した形式でも指定できます。

```
< masterVolumeId > , < snapshotId >
```

属性	型	説明
masterVolumeId	int	(必須) スナップショットの作成元ボリューム ID
snapshotId	int	(必須) スナップショット ID

## クエリパラメータ

なし。

## ボディ

なし。

## レスポンスメッセージ

### ボディ

属性	型	説明
statusResource	string	削除したスナップショットの実行結果の問い合わせに使用する URL

### メモ

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H "Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X DELETE https://192.0.2.100/ConfigurationManager/simple/v1/objects/snapshots/100,3
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

## 第6章

# Active Mirror を利用するための環境構築

この章では、Active Mirror を利用するための環境構築のうち、REST API を使用して外部ボリュームの作成および Quorum ディスクの登録について説明します。

## 6.1 Active Mirror を利用するための環境構築の流れ

Active Mirror を利用すると、2 台のストレージシステムの間でデータをボリューム単位に冗長化し、ストレージシステムの可用性を向上させることができます。

一方のストレージシステムのボリュームにデータが書き込まれると自動的にもう一方のストレージシステムのボリュームに同期されるため、ユーザはサーバがアクセスするボリュームの構成を意識することなく利用できます。

ここでは、Active Mirror の環境構築までの流れを記載します。環境構築後の Active Mirror の設定については、マニュアル『Active Mirror ユーザガイド』を参照してください。

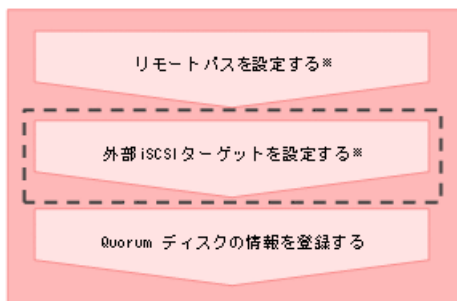
Active Mirror を利用するための設定は、Active Mirror を構成するストレージシステムの双方で必要です。

### ⚠ 注意

外部接続用のポートを、外部ストレージシステムからローカルストレージシステムに対する I/O パスの終点として使用している場合、I/O パスが切断されるおそれがあります。外部パスとして使用できる経路を使用してください。外部パスに関する詳細については、マニュアル『Universal Volume Manager ユーザガイド』を参照してください。

### メモ

REST API では、仮想ポートが有効なポートを Quorum ディスクのための外部接続用のポートとして使用する操作はできません。



〈凡例〉

- : REST API で操作する手順
- : 必要に応じて実施する操作

注※

リクエストラインに「simple」を含まないAPIで操作してください。

- リモートパスを設定する

リモート接続を作成する API の操作方法については、マニュアル『REST API リファレンスガイド』を参照してください。リクエストラインを次に示します。

```
POST <ベース URL>/v1/objects/remotepath-groups
```

- 外部 iSCSI ターゲットを設定する

外部ボリュームを作成して Quorum ディスクを設定する場合、FC や iSCSI で外部ストレージシステムと接続します。iSCSI で接続する場合は、あらかじめ外部接続用のポート（ローカルポート）に、外部ストレージシステムの iSCSI ターゲットの情報を登録しておく必要があります。

ローカルストレージシステム側ポートに外部ストレージシステムの iSCSI ネームを登録する API の操作方法については、マニュアル『REST API リファレンスガイド』を参照してください。リクエストラインを次に示します。

```
POST <ベース URL>/v1/objects/iscsi-ports/<オブジェクト ID>/actions/register/invoke
```

- Quorum ディスクの情報を登録する

Active Mirror で使用する Quorum ディスクに関する情報をストレージシステムに登録します。

## 6.2 外部ボリュームの情報を取得する

外部ボリュームに関する情報を取得します。外部ボリュームの条件を指定して、絞り込んだ情報を取得することもできます。

### メモ

- 外部ボリュームグループ：外部接続したストレージシステムのボリュームをマッピングして作成されたオブジェクトです。外部ボリュームを管理するために使用します。
- 外部ボリューム：外部ボリュームグループから作成されたボリュームです。
- 外部パスグループ：外部接続用ポートと、外部ストレージシステムのポート間を接続した経路のことを外部パスと呼びます。同じ外部パスを使用する外部ボリュームをグルーピングしたものを外部パスグループと呼びます。

## 実行権限

ストレージ管理者（参照）



## リクエストライン

```
GET <ベース URL>/simple/v1/objects/external-volumes
```

## リクエストメッセージ

### オブジェクト ID

なし。

### クエリパラメータ

1 回のリクエストで取得できる外部ボリュームの情報は最大で 200 個です。201 個以上の外部ボリュームの情報を取得する場合は、count パラメータと startVolumeId パラメータを組み合わせて複数回に分けて API を実行してください。count パラメータを指定することで取得する外部ボリュームの情報をフィルタリングすることもできます。

パラメータ	型	フィルタ条件
nickname	string	(任意) 外部ボリュームのニックネーム 指定した値の一部が外部ボリュームのニックネームと一致する場合も外部ボリュームの情報が取得されます。
minTotalCapacity	long	(任意) 外部ボリュームの最小容量 (MiB) 取得する外部ボリュームの最小容量として 47～268435456 までの整数で指定します。
maxTotalCapacity	long	(任意) 外部ボリュームの最大容量 (MiB) 取得する外部ボリュームの最大容量として 47～268435456 までの整数で指定します。
status	string	(任意) 取得する外部ボリュームの状態として、次の値を指定します。 <ul style="list-style-type: none"> <li>• Normal : 正常</li> <li>• Blockade : 閉塞</li> </ul>
volumeType	string	(任意) 取得する外部ボリュームタイプとして、Quorum Disk (Active Mirror で使用する Quorum ディスク) を指定します。
externalParityGroupId	string	(任意) 外部ボリュームグループ ID 「X-Y」形式で指定します。 X に指定できる値は 1～16384 までの整数です。 Y に指定できる値は 1～4096 までの整数です。
externalPathGroupId	int	(任意) 外部パスグループ ID を 0～63231 までの整数で指定します。
startVolumeId	int	(任意) 取得を開始する外部ボリュームの ID を 0～65279 の値で指定します。
count	int	(任意) 取得する外部ボリュームの情報の個数を 1～200 の値で指定します。 このパラメータを省略した場合、200 が指定されたと見なされます。

## ボディ

なし。

## レスポンスメッセージ

### ボディ

外部ボリュームタイプが **Quorum** ディスク（クエリパラメータ `volumeType` に `Quorum Disk` を指定）の外部ボリュームの情報を取得した出力例を次に示します。

```
{
  "data": [
    {
      "id": 1655,
      "nickname": "externalVolumeFc",
      "totalCapacity": 16384,
      "Status": "Normal",
      "externalParityGroupId": "42-3",
      "externalPathGroupId": 91,
      "externalPathOfVolume": [
        {
          "portId": "CL5-A",
          "portProtocol": "FC",
          "externalPortWwn": "50060e8012000c60",
          "lun": 1463
        }
      ],
      "volumeTypes": [
        "Quorum Disk"
      ],
      "externalStorage": {
        "model": "iStorage V310",
        "serial": "800012",
        "vendor": "NEC"
      },
      "quorumSetting": {
        "quorumId": 31,
        "pairedStorageModel": "M8",
        "pairedStorageSerial": "801026"
      }
    },
    {
      "id": 1656,
      "nickname": "externalVolumeIscsi",
      "totalCapacity": 16384,
      "status": "Normal",
      "externalParityGroupId": "9-1025",
      "externalPathGroupId": 1025,
      "externalPathOfVolume": [
        {
          "portId": "CL1-B",
          "portProtocol": "iSCSI",
          "externalPortIpAddress": "10.1.2.122",
          "externalPortIscsiName": "iqn.2001-03.jp.nec:storage01.h9s.t.000"
        }
      ]
    }
  ]
}
```

```

12.2d00b",
    "externalTcpPortNumber": 3260,
    "lun": 1025,
    "virtualPortNumber": 1
  }
],
"volumeTypes": [
  "Quorum Disk"
],
"externalStorage": {
  "model": "iStorage V310",
  "serial": "800012",
  "vendor": "NEC"
},
"quorumSetting": {
  "quorumId": 12,
  "pairedStorageModel": "M8",
  "pairedStorageSerial": "801206"
}
}
],
"count": 2,
"totalCount": 2,
"hasNext": false
}

```

属性	型	説明
id	int	外部ボリューム ID
nickname	string	外部ボリュームのニックネーム 外部ボリュームのニックネームが定義されている場合に出力されます。
totalCapacity	long	外部ボリュームの総容量 (MiB)
status	string	外部ボリュームの状態 <ul style="list-style-type: none"> <li>• Normal : 正常</li> <li>• Busy : 状態遷移中</li> <li>• Blockade : 閉塞</li> </ul>
externalParityGroupId	string	外部ボリュームグループ ID
externalPathGroupId	int	外部パスグループ ID
externalPathOfVolume	object[]	外部ボリュームの外部パス情報が、外部パスごとに出力されます。 <ul style="list-style-type: none"> <li>• portId (string) 外部接続用のポート ID</li> <li>• portProtocol (string) ポートのプロトコル <ul style="list-style-type: none"> <li>- FC</li> <li>- iSCSI</li> </ul> </li> <li>• externalPortWwn (string) 外部ストレージシステムのポートの WWN iSCSI ポートの場合、この属性は出力されません。</li> <li>• externalPortIpAddress (string) 外部ストレージシステムのポートの IP アドレス</li> </ul>

属性	型	説明
		<p>IPv4 または IPv6 形式で出力されます。</p> <p>FC ポートの場合、この属性は出力されません。</p> <ul style="list-style-type: none"> <li>• externalPortIscsiName (string) 外部ストレージシステムのポートの iSCSI ターゲット名 FC ポートの場合、この属性は出力されません。</li> <li>• externalTcpPortNumber (int) 外部ストレージシステムのポートの TCP ポート番号 FC ポートの場合、この属性は出力されません。</li> <li>• virtualPortNumber (int) 外部接続用のポートの仮想ポート番号 次の場合、この属性は出力されません。 <ul style="list-style-type: none"> <li>- FC ポートの場合</li> <li>- iSCSI ポートの場合で、仮想ポートが無効のとき</li> </ul> </li> <li>• lun (int) 外部ストレージシステムのポートに割り当てられている LUN</li> </ul>
volumeTypes	object[]	<p>外部ボリュームタイプ</p> <ul style="list-style-type: none"> <li>• Quorum Disk : Active Mirror で使用する Quorum ディスク</li> </ul>
externalStorage	object	<p>外部ストレージシステムの情報が出力されます。</p> <ul style="list-style-type: none"> <li>• model (string) 外部ストレージシステムのモデル 出力される外部ストレージシステムの表記については、マニュアル『Universal Volume Manager ユーザガイド』を参照してください。 有効な値が取得できない場合、この属性は出力されません。</li> <li>• serial (string) 外部ストレージシステムのシリアル番号 有効な値が取得できない場合、この属性は出力されません。</li> <li>• vendor (string) 外部ストレージシステムのベンダー識別子 有効な値が取得できない場合、この属性は出力されません。</li> </ul>
quorumSetting	object	<p>Quorum ディスクに関する情報が設定されている場合に、Quorum ディスクに関する情報が出力されます。</p> <ul style="list-style-type: none"> <li>• quorumId (int) Quorum ディスク ID</li> <li>• pairedStorageModel (string) Quorum ディスクを共有する相手側のストレージシステムのモデル <ul style="list-style-type: none"> <li>- M8 : iStorage V110、iStorage V310、iStorage V310F</li> </ul> モデルの情報が取得できない場合、Unknown が出力されます。</li> <li>• pairedStorageSerial (string)</li> </ul>

属性	型	説明
		Quorum ディスクを共有する相手側のストレージシステムのシリアル番号

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192
.0.2.100/ConfigurationManager/simple/v1/objects/external-volumes?volumeType
="Quorum Disk"
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[データオブジェクト \(17 ページ\)](#)

## 6.3 特定の外部ボリュームの情報を取得する

外部ボリューム ID を指定して、特定の外部ボリュームの情報を取得します。

### 実行権限

ストレージ管理者 (参照)

### リクエストライン

```
GET <ベース URL>/simple/v1/objects/external-volumes/<オブジェクト ID>
```

### リクエストメッセージ

#### オブジェクト ID

外部ボリュームの情報取得で取得した id の値を指定します。

属性	型	説明
id	int	(必須) 外部ボリューム ID

### クエリパラメータ

なし。

### ボディ

なし。

## レスポンスメッセージ

### ボディ

外部ボリューム ID が 3 のボリュームの情報を取得した例を次に示します。

```
{
  "id": 3,
  "nickname": "externalVolumeForQuorum",
  "totalCapacity": 20480,
  "status": "Normal",
  "volumeIdentifier": "NEC 5040000C1965",
  "externalParityGroupId": "8-501",
  "externalPathGroupId": 501,
  "externalPathOfVolume": [
    {
      "portId": "CL5-A",
      "portProtocol": "FC",
      "externalPortWwn": "50060e8012000c60",
      "lun": 501
    }
  ],
  "volumeTypes": [],
  "externalStorage": {
    "model": "iStorage V310",
    "serial": "800012",
    "vendor": "NEC"
  }
}
```

取得する属性については、外部ボリュームの情報を取得する場合の属性に加えて、次に示す属性が取得されます。

属性	型	説明
volumeIdentifier	string	外部ボリュームを特定するための識別子 例：NEC 5040000C1965

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192
.0.2.100/ConfigurationManager/simple/v1/objects/external-volumes/3
```

---

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[外部ボリュームの情報を取得する \(176 ページ\)](#)

---

## 6.4 外部ボリュームを作成する

ストレージシステムに、外部接続されたストレージシステムのボリュームをマッピングして外部ボリュームグループを作成し、外部ボリュームグループから外部ボリュームを作成します。同時に外部パスグループと外部パスも作成されます。

### メモ

- 外部ボリュームの容量が 4194304MiB (4TiB) を超える場合、4194304MiB (4TiB) の外部ボリュームとして作成されます。
  - iSCSI で接続する場合は、あらかじめ外部ストレージシステムの iSCSI ターゲットの情報を、外部接続用の iSCSI ポート（ローカルポート）に登録してください。詳細はマニュアル『HA Storage Manager Embedded ユーザガイド』を参照してください。
- 

## 実行権限

ストレージ管理者（プロビジョニング）

## リクエストライン

```
POST <ベース URL>/simple/v1/objects/external-volumes
```

## リクエストメッセージ

### オブジェクト ID

なし。

### クエリパラメータ

なし。

## ボディ

外部ボリュームを作成する例を次に示します。

- FC の場合

```
{
  "externalParityGroupId": "8-501",
  "externalPathGroupId": 501,
  "portId": "CL5-A",
  "externalPortWwn": "50060e8012000c60",
  "lun": 501
}
```

- iSCSI の場合

```
{
  "externalParityGroupId": "9-1025",
  "externalPathGroupId": 1025,
  "portId": "CL1-B",
  "externalPortIpAddress": "10.1.2.122",
  "externalPortIscsiName": "iqn.2001-03.jp.nec:storage01.h9s.t.00012.2d00b",
  "lun": 1025
}
```

属性	型	説明
externalParityGroupId	string	(必須) 外部ボリュームグループ ID 使用されていない外部ボリュームグループ ID を指定してください。 「x-y」形式で指定します。 x に指定できる値は 1～16384 までの整数です。 y に指定できる値は 1～4096 までの整数です。 例 : 1-1
externalPathGroupId	int	(必須) 外部パスグループ ID 使用されていない外部パスグループ ID を指定してください。 0～63231 までの整数で指定します。
portId	string	(必須) 外部接続用のポート ID
externalPortWwn	string	(任意) 外部ストレージシステムのポートの WWN FC ポートの場合は、この属性を必ず指定してください。 iSCSI ポートの場合で、この属性を指定したときは指定した値は無視されます。
externalPortIpAddress	string	(任意) 外部ストレージシステムのポートの IP アドレス IPv4 または IPv6 の IP アドレスを指定します。 iSCSI ポートの場合は、この属性を必ず指定してください。 FC ポートの場合で、この属性を指定したときは指定した値は無視されます。
externalPortIscsiName	string	(任意) 外部ストレージシステムのポートの iSCSI ターゲット名



属性	型	説明
		iSCSI ポートの場合は、この属性を必ず指定してください。 FC ポートの場合で、この属性を指定したときは指定した値は無視されます。
lun	int	(必須) 外部ストレージシステムのポートに割り当てられている LUN

## レスポンスメッセージ

### ボディ

属性	型	説明
statusResource	string	作成した外部ボリュームの実行結果の問い合わせに使用する URL

### メモ

- 非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。
- 外部ボリュームグループや外部パスグループが正しく作成されているかどうかは、外部ボリュームグループの情報を取得する API、および外部パスグループの情報を取得する API を実行して確認してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H "Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/simple/v1/objects/external-volumes
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

[特定の外部ボリュームグループの情報を取得する \(190 ページ\)](#)

[特定の外部パスグループの情報を取得する \(195 ページ\)](#)

## 6.5 外部ボリュームのニックネームを変更する

外部ボリューム ID を指定して、外部ボリュームのニックネームを変更します。

### 実行権限

ストレージ管理者（プロビジョニング）

### リクエストライン

```
PATCH <ベース URL>/simple/v1/objects/external-volumes/<オブジェクト ID>
```

### リクエストメッセージ

#### オブジェクト ID

外部ボリュームの情報取得で取得した id の値を指定します。

属性	型	説明
id	int	(必須) 外部ボリューム ID

#### クエリパラメータ

なし。

#### ボディ

```
{
  "nickname": "externalVolumeForQuorum2"
}
```

属性	型	説明
nickname	string	<p>(必須) 変更するニックネームを 1~32 文字で指定します。</p> <p>使用できる文字は次のとおりです。</p> <p>0~9 A~Z a~z ! # \$ % &amp; ' ( ) + , - . / : = @ [ \ ] ^ _ ` { } ~ 半角スペース</p> <p>英字は、大文字と小文字が区別されます。</p> <p>半角スペースは、使用できる文字と文字の間には使用できますが、先頭と末尾には使用できません。</p>

## レスポンスメッセージ

### ボディ

属性	型	説明
affectedResources	string[]	変更した外部ボリュームの情報を参照するための URL の一覧
operationDetails	object[]	変更した外部ボリュームの詳細情報 詳細については、commandStatus オブジェクトの属性「operationDetails」の説明を参照してください。

### ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

### コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X PATCH --data-binary
@./InputParameters.json https://192.0.2.100/ConfigurationManager/simple/v1/objects/external-volumes/3
```

#### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[commandStatus オブジェクト \(19 ページ\)](#)

## 6.6 外部ボリュームグループの情報を取得する

外部ボリュームグループの情報の一覧を取得します。

### 実行権限

ストレージ管理者 (参照)

### リクエストライン

```
GET <ベース URL>/simple/v1/objects/external-parity-groups
```

## リクエストメッセージ

### オブジェクト ID

なし。

### クエリパラメータ

1 回のリクエストで取得できる外部ボリュームグループの情報は最大で 200 個です。201 個以上の外部ボリュームグループの情報を取得する場合は、count パラメータと startIndex パラメータを組み合わせる複数回に分けて API を実行してください。count パラメータを指定することで取得する外部ボリュームグループの情報をフィルタリングすることもできます。

パラメータ	型	フィルタ条件
startIndex	int	(任意) 取得を開始する外部ボリュームグループのインデックス番号を 0～65535 の値で指定します。
count	int	(任意) 取得する外部ボリュームグループの情報の個数を 1～200 の値で指定します。 省略した場合、200 が指定されたと見なされます。

### ボディ

なし。

## レスポンスメッセージ

### ボディ

```
{
  "data": [
    {
      "id": "1-500",
      "status": "Normal",
      "externalPathGroupId": 500,
      "externalPath": [
        {
          "portId": "CL5-A",
          "portProtocol": "FC",
          "externalPortWwn": "50060e8012000c60",
          "lun": 1754,
          "status": "Normal"
        }
      ],
      "index": 2073
    },
    {
      "id": "9-1200",
      "status": "Normal",
      "externalPathGroupId": 1200,
      "externalPath": [
```

```

    {
      "portId": "CL1-B",
      "portProtocol": "iSCSI",
      "externalPortIpAddress": "10.1.2.122",
      "externalPortIscsiName": "iqn.2001-03.jp.nec:storage01.h9s.t.0
0012.2d011",
      "externalTcpPortNumber": 3260,
      "lun": 1200,
      "virtualPortNumber": 1,
      "status": "Normal"
    }
  ],
  "index": 2274
}
],
"count": 2,
"totalCount": 2,
"hasNext": false
}

```

属性	型	説明
id	string	外部ボリュームグループ ID
status	string	外部ボリュームグループの状態 <ul style="list-style-type: none"> <li>• Normal : 正常</li> <li>• Checking : 外部パスの状態を確認中</li> <li>• Cache Destaging : キャッシュ内のデータをボリュームに書き込み中</li> <li>• Disconnect : 外部パスが未接続</li> <li>• Blockade : 外部パスが閉塞</li> <li>• Warning : 外部パスの一部に問題がある</li> <li>• Unknown : 不明</li> </ul>
externalPathGroupId	int	外部パスグループ ID
externalPath	object[]	外部ボリュームグループ内の外部パスの情報が、外部パスごとに出力されます。 <ul style="list-style-type: none"> <li>• portId (string) 外部接続用のポート ID</li> <li>• portProtocol (string) ポートのプロトコル               <ul style="list-style-type: none"> <li>- FC</li> <li>- iSCSI</li> </ul> </li> <li>• externalPortWwn (string) 外部ストレージシステムのポートの WWN iSCSI ポートの場合、この属性は出力されません。</li> <li>• externalPortIpAddress (string) 外部ストレージシステムのポートの IP アドレス IPv4 または IPv6 形式で出力されます。 FC ポートの場合、この属性は出力されません。</li> <li>• externalPortIscsiName (string) 外部ストレージシステムのポートの iSCSI ターゲット名 FC ポートの場合、この属性は出力されません。</li> </ul>

属性	型	説明
		<ul style="list-style-type: none"> <li>• externalTcpPortNumber (int) 外部ストレージシステムのポートの TCP ポート番号 FC ポートの場合、この属性は出力されません。</li> <li>• virtualPortNumber (int) 外部接続用のポートの仮想ポート番号 次の場合、この属性は出力されません。 <ul style="list-style-type: none"> <li>- FC ポートの場合</li> <li>- iSCSI ポートの場合で、仮想ポートが無効のとき</li> </ul> </li> <li>• lun (int) 外部ストレージシステムのポートに割り当てられている LUN 外部パスの状態が不明 (Unknown) の場合、この属性は出力されません。</li> <li>• status (string) 外部パスの状態 <ul style="list-style-type: none"> <li>- Normal : 正常</li> <li>- Disconnect : 未接続</li> <li>- Temporary Blockade : ポートが閉塞 (一時閉塞)</li> <li>- Blockade : 閉塞</li> <li>- Unknown : 不明</li> </ul> </li> </ul>
index	int	外部ボリュームグループのインデックス番号

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192
.0.2.100/ConfigurationManager/simple/v1/objects/external-parity-groups
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[データオブジェクト \(17 ページ\)](#)

## 6.7 特定の外部ボリュームグループの情報を取得する

外部ボリュームグループ ID を指定して、特定の外部ボリュームグループの情報を取得します。

## 実行権限

ストレージ管理者（参照）

## リクエストライン

```
GET <ベース URL>/simple/v1/objects/external-parity-groups/<オブジェクト ID>
```

## リクエストメッセージ

### オブジェクト ID

外部ボリュームグループの情報取得で取得した id の値を指定します。

属性	型	説明
id	string	(必須) 外部ボリュームグループ ID

### クエリパラメータ

なし。

### ボディ

なし。

## レスポンスメッセージ

### ボディ

外部ボリュームグループ ID が 9-500 の外部ボリュームグループの情報を取得した例を次に示します。

```
{
  "id": "9-500",
  "status": "Normal",
  "externalPathGroupId": 500,
  "externalPath": [
    {
      "portId": "CL5-A",
      "portProtocol": "FC",
      "externalPortWwn": "50060e8012000c60",
      "lun": 3,
      "status": "Normal"
    }
  ],
  "index": 2051
}
```

取得する属性については、外部ボリュームグループの情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192
.0.2.100/ConfigurationManager/simple/v1/objects/external-parity-groups/9-50
0
```

---

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[外部ボリュームグループの情報を取得する \(187 ページ\)](#)

---

## 6.8 外部パスグループの情報を取得する

外部パスグループの情報を一覧で取得します。関連する外部パスの情報も取得します。

### 実行権限

ストレージ管理者 (参照)

### リクエストライン

```
GET <ベース URL>/simple/v1/objects/external-path-groups
```

### リクエストメッセージ

#### オブジェクト ID

なし。

#### クエリパラメータ

1 回のリクエストで取得できる外部パスグループの情報は最大で 200 個です。201 個以上の外部パスグループの情報を取得する場合は、count パラメータと startId パラメータ



タを組み合わせ、複数回に分けて API を実行してください。count パラメータを指定することで取得する外部パスグループの情報をフィルタリングすることもできます。

パラメータ	型	フィルタ条件
startId	int	(任意) 取得を開始する外部パスグループの ID を 0～63231 の値で指定します。
count	int	(任意) 取得する外部パスグループの情報の個数を 1～200 の値で指定します。 省略した場合、200 が指定されたと見なされます。

## ボディ

なし。

## レスポンスメッセージ

## ボディ

```
{
  "data": [
    {
      "id": 1235,
      "externalPaths": [
        {
          "portId": "CL5-A",
          "portProtocol": "FC",
          "externalPortWwn": "50060e8012000c60"
        }
      ]
    },
    {
      "id": 1200,
      "externalPaths": [
        {
          "portId": "CL1-B",
          "portProtocol": "iSCSI",
          "externalPortIpAddress": "10.1.9.122",
          "externalPortIscsiName": "iqn.2001-03.jp.nec:storage01.h9s.t.0012.2d011",
          "externalTcpPortNumber": 3260,
          "virtualPortNumber": 1
        }
      ]
    }
  ],
  "count": 2,
  "totalCount": 2,
  "hasNext": false
}
```

属性	型	説明
id	int	外部パスグループ ID

属性	型	説明
externalPaths	object[]	<p>外部パスグループに関連する外部パスの情報が、外部パスごとに出力されます。</p> <ul style="list-style-type: none"> <li>• portId (string) 外部接続用のポート ID</li> <li>• portProtocol (string) ポートのプロトコル <ul style="list-style-type: none"> <li>- FC</li> <li>- iSCSI</li> </ul> </li> </ul> <p>ポートの情報が取得できない場合、この属性は出力されません。</p> <ul style="list-style-type: none"> <li>• externalPortWwn (string) 外部ストレージシステムのポートの WWN ポートの情報が取得できない場合、または iSCSI ポートのときには、この属性は出力されません。</li> <li>• externalPortIpAddress (string) 外部ストレージシステムのポートの IP アドレス IPv4 または IPv6 形式で出力されます。 FC ポートの場合、この属性は出力されません。</li> <li>• externalPortIscsiName (string) 外部ストレージシステムのポートの iSCSI ターゲット名 FC ポートの場合、この属性は出力されません。</li> <li>• externalTcpPortNumber (int) 外部ストレージシステムのポートの TCP ポート番号 FC ポートの場合、この属性は出力されません。</li> <li>• virtualPortNumber (int) 外部接続用のポートの仮想ポート番号 次の場合、この属性は出力されません。 <ul style="list-style-type: none"> <li>- FC ポートの場合</li> <li>- iSCSI ポートの場合で、仮想ポートが無効のとき</li> </ul> </li> </ul>

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192
.0.2.100/ConfigurationManager/simple/v1/objects/external-path-groups
```

### —— 関連リンク ——

[HTTP ステータスコード \(13 ページ\)](#)

# 6.9 特定の外部パスグループの情報を取得する

外部パスグループ ID を指定して、特定の外部パスグループの情報を取得します。

## 実行権限

ストレージ管理者 (参照)

## リクエストライン

GET <ベース URL>/simple/v1/objects/external-path-groups/<オブジェクト ID>

## リクエストメッセージ

### オブジェクト ID

外部パスグループの情報取得で取得した id の値を指定します。

属性	型	説明
id	int	(必須) 外部パスグループ ID

### クエリパラメータ

なし。

### ボディ

なし。

## レスポンスメッセージ

### ボディ

外部パスグループ ID が 600 の外部パスグループの情報を取得した例を次に示します。

```
{
  "id": 600,
  "externalPaths": [
    {
      "portId": "CL5-A",
      "portProtocol": "FC",
      "externalPortWwn": "50060e8012000c60"
    },
    {
```

```

        "portId": "CL6-A",
        "portProtocol": "FC",
        "externalPortWwn": "50060e8012000c70"
    }
]
}

```

取得する属性については、外部パスグループの情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```

curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192
.0.2.100/ConfigurationManager/simple/v1/objects/external-path-groups/600

```

---

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[外部パスグループの情報を取得する \(192 ページ\)](#)

---

## 6.10 Quorum ディスクの情報を取得する

Quorum ディスクに関する情報を取得します。外部ボリュームを使用しないで Quorum ディスクの情報を登録している場合も情報を取得できます。

### 実行権限

ストレージ管理者 (参照)

### リクエストライン

```
GET <ベース URL>/simple/v1/objects/quorum-disks
```

### リクエストメッセージ

#### オブジェクト ID

なし。

## クエリパラメータ

なし。

## ボディ

なし。

## レスポンスメッセージ

### ボディ

```
{
  "data": [
    {
      "quorumId": 6,
      "pairedStorageModel": "M8",
      "pairedStorageSerial": "800001"
    },
    {
      "quorumId": 5,
      "pairedStorageModel": "M8",
      "pairedStorageSerial": "800032"
    },
    {
      "quorumId": 2,
      "pairedStorageModel": "M8",
      "pairedStorageSerial": "800501"
    },
    {
      "quorumId": 3,
      "externalVolumeId": 15,
      "externalVolumeNickname": "ExternalVolume2",
      "externalVolumeStatus": "Normal",
      "pairedStorageModel": "M8",
      "pairedStorageSerial": "800401"
    },
    {
      "quorumId": 1,
      "externalVolumeId": 6,
      "externalVolumeNickname": "ExternalVolume1",
      "externalVolumeStatus": "Normal",
      "pairedStorageModel": "M8",
      "pairedStorageSerial": "800108"
    },
    {
      "quorumId": 7,
      "pairedStorageModel": "M8",
      "pairedStorageSerial": "800512"
    },
    {
      "quorumId": 4,
      "externalVolumeId": 16,
      "externalVolumeNickname": "ExternalVolume3",
      "externalVolumeStatus": "Normal",
```

```

        "pairedStorageModel": "M8",
        "pairedStorageSerial": "800108"
    },
    ],
    "count": 7
}

```

属性	型	説明
quorumId	int	Quorum ディスク ID
externalVolumeId※	int	Quorum ディスクの情報が登録されている外部ボリューム ID
externalVolumeNickname※	string	Quorum ディスクの情報が登録されている外部ボリュームのニックネーム 外部ボリュームのニックネームが定義されている場合に出力されます。
externalVolumeStatus※	string	Quorum ディスクの情報が登録されている外部ボリュームの状態 <ul style="list-style-type: none"> <li>• Normal : 正常</li> <li>• Busy : 状態遷移中</li> <li>• Blockade : 閉塞</li> </ul>
pairedStorageModel	string	Quorum ディスクを共有する相手側のストレージシステムのモデル <ul style="list-style-type: none"> <li>• M8 : iStorage V110、iStorage V310、iStorage V310F</li> </ul> モデルの情報が取得できない場合、Unknown が出力されます。
pairedStorageSerial	string	Quorum ディスクを共有する相手側のストレージシステムのシリアル番号

### 注※

外部ボリュームを使用しないで Quorum ディスクの情報を登録している場合は出力されません。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```

curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192
.0.2.100/ConfigurationManager/simple/v1/objects/quorum-disks

```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[データオブジェクト \(17 ページ\)](#)

## 6.11 特定の Quorum ディスクの情報を取得する

Quorum ディスク ID を指定して、Quorum ディスクに関する情報を取得します。

### 実行権限

ストレージ管理者（参照）

### リクエストライン

```
GET <ベース URL>/simple/v1/objects/quorum-disks/<オブジェクト ID>
```

### リクエストメッセージ

#### オブジェクト ID

Quorum ディスクの情報取得で取得した quorumId の値を指定します。

属性	型	説明
quorumId	int	(必須) Quorum ディスク ID

#### クエリパラメータ

なし。

#### ボディ

なし。

### レスポンスメッセージ

#### ボディ

Quorum ディスク ID が 4 の Quorum ディスクに関する情報を取得した例を次に示します。

このレスポンスメッセージでは、外部ボリュームに関する属性（externalVolumeId、externalVolumeNickname、および externalVolumeStatus）が出力されているため、外部ボリュームを使用して Quorum ディスクの情報が登録されていることを示しています。

```
{
  "quorumId": 4,
  "externalVolumeId": 84,
  "externalVolumeNickname": "Quorum_R800_10",
  "externalVolumeStatus": "Normal",
  "pairedStorageModel": "M8",
```

```
"pairedStorageSerial": "801305"
}
```

Quorum ディスク ID が 30 の Quorum ディスクに関する情報を取得した例を次に示します。

このレスポンスメッセージでは、外部ボリュームに関する属性（externalVolumeId、externalVolumeNickname、および externalVolumeStatus）が出力されていないため、外部ボリュームを使用しないで Quorum ディスクの情報が登録されていることを示しています。

```
{
  "quorumId": 30,
  "pairedStorageModel": "M8",
  "pairedStorageSerial": "810011"
}
```

取得する属性については、Quorum ディスクの情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://192
.0.2.100/ConfigurationManager/simple/v1/objects/quorum-disks/7
```

---

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[Quorum ディスクの情報を取得する \(196 ページ\)](#)

---

## 6.12 Quorum ディスクの情報を登録する

Active Mirror で使用する Quorum ディスクに関する情報をストレージシステムに登録します。

### 実行権限

ストレージ管理者（プロビジョニング）



## リクエストライン

```
POST <ベース URL>/simple/v1/objects/quorum-disks
```

## リクエストメッセージ

### オブジェクト ID

なし。

### クエリパラメータ

なし。

### ボディ

外部ボリュームを使用して、Quorum ディスクに関する情報を次のとおり指定して登録する例を次に示します。

- Quorum ディスク ID : 7
- 外部ボリューム ID : 6400
- 相手側のストレージシステムのモデル : M8
- 相手側のストレージシステムのシリアル番号 : 801026

```
{
  "quorumId": 7,
  "externalVolumeId": 6400,
  "pairedStorageModel": "M8",
  "pairedStorageSerial": "801026"
}
```

属性	型	説明
quorumId	int	(必須) Quorum ディスク ID 登録する Quorum ディスク ID を 0～31 までの整数で指定します。 使用されていない Quorum ディスク ID を指定してください。
externalVolumeId	int	(任意) 外部ボリューム ID 登録する外部ボリューム ID を 0～65279 までの整数で指定します。 外部ボリュームを使用して Quorum ディスクの情報を登録する場合、この属性は必ず指定してください。
pairedStorageModel	string	(必須) Quorum ディスクを共有する相手側のストレージシステムのモデル 指定できる値は次のとおりです。 <ul style="list-style-type: none"> <li>• M8 : iStorage V110、iStorage V310、iStorage V310F</li> </ul>
pairedStorageSerial	string	(必須) Quorum ディスクを共有する相手側のストレージシステムのシリアル番号

## レスポンスメッセージ

### ボディ

属性	型	説明
statusResource	string	登録した Quorum ディスクの情報の実行結果の問い合わせに使用する URL

### メモ

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-bin
ary @./InputParameters.json https://192.0.2.100/ConfigurationManager/simple
/v1/objects/quorum-disks
```

### 関連リンク

[HTTP ステータスコード \(13 ページ\)](#)

[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

## 6.13 Quorum ディスクの情報を削除する

Quorum ディスク ID を指定して、ストレージシステムから Quorum ディスクに関する情報を削除します。

### メモ

外部ボリュームまで削除する場合、外部ボリュームのマッピングを解除する API を実行して削除する必要があります。操作方法については、マニュアル『REST API リファレンスガイド』を参照してください。

## 実行権限

ストレージ管理者（プロビジョニング）

## リクエストライン

```
DELETE <ベース URL>/simple/v1/objects/quorum-disks/<オブジェクト ID>
```

## リクエストメッセージ

### オブジェクト ID

Quorum ディスクの情報取得で取得した quorumId の値を指定します。

属性	型	説明
quorumId	int	(必須) Quorum ディスク ID

### クエリパラメータ

なし。

### ボディ

なし。

## レスポンスメッセージ

### ボディ

属性	型	説明
statusResource	string	削除した Quorum ディスクの情報の実行結果の問い合わせに使用する URL

### メモ

非同期処理の API のステータス情報を取得する API を実行して、実行結果のステータス情報を参照することもできます。詳細は非同期処理の API のステータス情報を取得する API の説明を参照してください。

## ステータスコード

この操作のリクエストに対するステータスコードについては、HTTP ステータスコードの説明を参照してください。

## コード例

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X DELETE https://
192.0.2.100/ConfigurationManager/simple/v1/objects/quorum-disks/7
```

---

—— 関連リンク ——

---

[HTTP ステータスコード \(13 ページ\)](#)

[非同期処理の API のステータス情報を取得する \(24 ページ\)](#)

---

# 付録 A. 制限事項および注意事項

REST API を使用する際の注意事項を示します。

マニュアル『システム管理者ガイド』も参照して、事前に注意事項などを確認しておいてください。

## A.1 制限事項および注意事項

REST API の操作中に、次の事象が発生した場合の対処方法について示します。

### サーバとストレージシステムの接続情報を設定する場合にエラーとなる時

iSCSI 接続の場合、サーバにパス情報を追加する操作でエラーになるとき、サーバ名に、コンマ (,)、スラッシュ (/)、円マークまたはバックスラッシュ (\) を使用していないかを確認してください。使用している場合、サーバ名を変更してから再度サーバとストレージシステムの接続情報を設定する操作をしてください。

### REST API が定期的にエラーになる場合

次の条件に当てはまる場合で、REST API が利用できないとき、現象が発生している CTL の ESM リブートを実施してください。

- API を実行すると、エラーメッセージが通知される、かつ、その現象が実行した API や操作した画面にかかわらず定期的に発生する
- 通知されるエラーメッセージに対処できない、または対処しても現象が解消しない

## 付録 B. このマニュアルの参考情報

このマニュアルを読むに当たっての参考情報を示します。

### B.1 このマニュアルで使用している略語

このマニュアルで使用する英略語を次に示します。

略語	正式名称
AL-PA	Arbitrated-Loop Physical Address
API	Application programming interface
ESM	Embedded Storage Manager
FC	Fibre Channel
HBA	Host Bus Adapter
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
iSCSI	Internet Small Computer System Interface
JSON	JavaScript Object Notation
Node	Node
REST	Representational State Transfer
RFC	Request for Comments
RI	Read Intensive
SIM	Service Information Message
SLU	Subsidiary Logical Unit
SSD	Solid-State Drive
SSL	Secure Sockets Layer
TLS	Transport Layer Security
URL	Uniform Resource Locator
WWN	World Wide Name

### B.2 KB（キロバイト）などの単位表記について

このマニュアルでは、1KB（キロバイト）、1MB（メガバイト）、1GB（ギガバイト）、1TB（テラバイト）はそれぞれ 1,000 バイト、1,000KB、1,000MB、1,000GB です。

1KiB（キビバイト）、1MiB（メビバイト）、1GiB（ギビバイト）、1TiB（テビバイト）は、それぞれ 1,024 バイト、1,024KiB、1,024MiB、1,024GiB です。

---

# 索引

## A

API の実行ユーザのロールとアクセス権限.....	2
Authorization ヘッダ .....	10

## C

commandStatus オブジェクト .....	19
----------------------------	----

## H

HTTP ステータスコード .....	13
---------------------	----

## Q

Quorum ディスクの情報を削除する .....	202
Quorum ディスクの情報を取得する .....	196
Quorum ディスクの情報を登録する .....	200

## S

SSL 通信を利用する場合の要件 .....	2
------------------------	---

## U

URL の形式 .....	3
---------------	---

## あ

エラーオブジェクト .....	21
オブジェクト ID の指定方法 .....	9

## か

外部パスグループの情報を取得する .....	192
外部ボリューム	
特定の外部ボリュームの情報を取得する .....	181
外部ボリュームグループの情報を取得する ..	187
外部ボリュームの情報を取得する .....	176
外部ボリュームのニックネームを変更する ..	186
外部ボリュームを作成する .....	183
概要	
Active Mirror を利用するための環境構築 .....	175

スナップショット .....	155
----------------	-----

## 共通仕様

API の実行ユーザのロールとアクセス権限 .....	2
commandStatus オブジェクト .....	19
HTTP ステータスコード .....	13
SSL 通信を利用する場合の要件 .....	2
URL の形式 .....	3
エラーオブジェクト .....	21
オブジェクト ID の指定方法 .....	9
クエリパラメータ .....	15
サポートする HTTP メソッド .....	9
出力形式 .....	17
データオブジェクト .....	17
データ型 .....	16
ユーザ認証 .....	10
リクエストおよびレスポンスのフォーマット .....	15
リクエストヘッダ .....	12
レスポンスヘッダ .....	13
クエリパラメータ .....	15

## さ

### サーバ

HBA の WWN (iSCSI ネーム) の情報を取得する .....	100
iSCSI ターゲットの情報を取得する .....	117
iSCSI ターゲットの設定を変更する .....	120
サーバ ID を指定して HBA の WWN (iSCSI ネーム) の情報を追加する .....	105
サーバからパス情報を削除する .....	115
サーバにパス情報を追加する .....	113
サーバにホストグループ (iSCSI ターゲッ ト) を追加する .....	96
サーバの情報を取得する .....	83
サーバの設定を変更する .....	92
サーバのパス情報を取得する .....	109
サーバを削除する .....	94
サーバを登録する .....	89

特定の iSCSI ターゲットの情報を取得する.....	119
特定のサーバから HBA の WWN (iSCSI ネーム) の情報を削除する.....	107
特定のサーバの HBA の WWN (iSCSI ネー ム) の情報を取得する.....	103
特定のサーバの情報を取得する.....	86
特定のサーバのパス情報を取得する.....	111
サポートする HTTP メソッド.....	9
システム構成.....	1
出力形式.....	17
ストレージシステム	
ストレージシステムの情報を取得する....	29
スナップショットとプールの稼働状態の情 報を取得する.....	35
スナップショット	
スナップショットの情報を取得する.....	160
スナップショットを削除する.....	173
スナップショットを作成する.....	165
スナップショットをマッピングする.....	168
スナップショットをリストアする.....	170
特定のスナップショットの情報を取得する .....	163
スナップショットグループ	
スナップショットグループ名の一覧を取得 する.....	156
スナップショットグループを削除する..	171
特定のスナップショットグループの情報を 取得する.....	158
た	
データオブジェクト.....	17
データ型.....	16
特定の Quorum ディスクの情報を取得する..	199
特定の外部パスグループの情報を取得する..	195
特定の外部ボリュームグループの情報を取得す る.....	190
ドライブ	
スペアドライブの設定を解除する.....	45
特定のドライブの情報を取得する.....	41
ドライブの情報を取得する.....	38
ドライブをスペアドライブに設定する....	43

## は

非同期処理の API のステータス情報を削除す る.....	27
非同期処理の API のステータス情報を取得す る.....	24
プール	
特定のプールの情報を取得する.....	55
プールにドライブを追加する.....	65
プールの情報を取得する.....	50
プールの設定を変更する.....	62
プールを削除する.....	63
プールを作成する.....	58
ポート	
特定のポートの情報を取得する.....	75
ポートの情報を取得する.....	68
ポートの設定を変更する.....	77
ホストグループ名 (iSCSI ターゲット名) をサー バのニックネームと同期する.....	99
ボリューム	
特定のボリュームの QoS の設定情報を取 得する.....	138
特定のボリュームの情報を取得する.....	128
ボリュームとサーバの接続情報を取得する .....	147
ボリュームとサーバを接続する.....	151
ボリュームと特定のサーバの接続情報を取 得する.....	149
ボリュームと特定のサーバの接続を解除す る.....	153
ボリュームの QoS の設定情報を変更する... .....	142
ボリュームの情報を取得する.....	124
ボリュームの設定を変更する.....	133
ボリュームの容量を拡張する.....	136
ボリュームを削除する.....	137
ボリュームを作成する.....	130

## や

ユーザ認証.....	10
------------	----



---

ら

リクエストおよびレスポンスのフォーマット15

リクエストヘッダ .....12

---

**iStorage V110/V310/V310F  
HA Storage Manager Embedded  
REST API リファレンスガイド**

**IV-UG-030-004-02**

**2025 年 1 月 第 2 版 発行**

**日本電気株式会社**

---

**© NEC Corporation 2024-2025**