

**Oracle® ZFS Storage Appliance RESTful
API ガイド、Release OS8.8.x**



Part No: F39466-01
2021 年 8 月

Part No: F39466-01

Copyright © 2014, 2021, Oracle and/or its affiliates.

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクルまでご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

このソフトウェアまたはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアまたはハードウェアは、危険が伴うアプリケーション(人的傷害を発生させる可能性があるアプリケーションを含む)への用途を目的として開発されていません。このソフトウェアまたはハードウェアを危険が伴うアプリケーションで使用する際、安全に使用するために、適切な安全装置、バックアップ、冗長性(redundancy)、その他の対策を講じることは使用者の責任となります。このソフトウェアまたはハードウェアを危険が伴うアプリケーションで使用了ことに起因して損害が発生しても、Oracle Corporationおよびその関連会社は一切の責任を負いかねます。

OracleおよびJavaはオラクルおよびその関連会社の登録商標です。その他の社名、商品名等は各社の商標または登録商標である場合があります。

Intel、Intel Insideは、Intel Corporationの商標または登録商標です。すべてのSPARCの商標はライセンスをもとに使用し、SPARC International, Inc.の商標または登録商標です。AMD、Epyc、AMDロゴは、Advanced Micro Devices, Inc.の商標または登録商標です。UNIXは、The Open Groupの登録商標です。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。適用されるお客様とOracle Corporationとの間の契約に別段の定めがある場合を除いて、Oracle Corporationおよびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。適用されるお客様とOracle Corporationとの間の契約に定めがある場合を除いて、Oracle Corporationおよびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

ドキュメントのアクセシビリティについて

オラクルのアクセシビリティについての詳細情報は、Oracle Accessibility ProgramのWebサイト(<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>)を参照してください。

Oracle Supportへのアクセス

サポートをご契約のお客様には、My Oracle Supportを通して電子支援サービスを提供しています。詳細情報は「<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>」か、聴覚に障害のあるお客様は「<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>」を参照してください。

目次

Oracle ZFS Storage Appliance RESTful API の概要	15
RESTful API 認証	15
RESTful API バージョン	16
RESTful API サービスバージョン	16
RESTful API サービスバージョン 2.0	17
スクリプト可能な値	17
一貫した値	18
一般的な RESTful 操作	19
HTTP 応答本文	19
HTTP 応答ヘッダー	20
問合せパラメータ	21
問合せパラメータ: props	21
問合せパラメータ: start	21
問合せパラメータ: end	22
問合せパラメータ: limit	22
問合せパラメータ: depth	23
問合せパラメータ: match	25
アプライアンスエラー	25
セキュリティープロトコルおよび暗号の設定	26
パスワードの複雑性	27
 RESTful API の操作	31
サービスへのアクセス	31
サービスの一覧表示	31
サービス操作の一覧表示	32
認証トークン	33
非永続ログイントークンの作成	34
ログアウトおよび非永続ログイントークンの削除	34

証明書の管理	37
証明書の一覧表示	38
証明書を PEM 形式で返す	41
サーバー証明書の作成	42
リクエストテンプレートを返す	42
リクエストの取り込みおよびアップロード	43
リクエストの CA への転送	45
鍵または証明書のアップロード	45
証明書を信頼する必要があるサービスの指定	47
システムのデフォルト証明書の設定	48
証明書の破棄	48
HTTP Strict Transport Security の有効化	48
 RESTful API 警告サービス	51
警告サービスコマンド	51
アラートアクション	52
すべてのアラートアクションの一覧表示	54
単一アラートアクションの一覧表示	55
アラートアクションの作成	56
アラートアクションの変更	57
イベントの応答の指定	57
イベントの応答の変更	58
イベントの応答の削除	59
アラートアクションの削除	59
カスタムアラート	59
カスタムアラートを作成および送信するための承認の構成	60
カスタムアラートの作成	60
カスタムアラートの送信	61
しきい値アラート	61
しきい値アラートの一覧表示	62
しきい値アラートの作成	63
しきい値アラートの変更	64
しきい値アラートの削除	65
 Analytics サービス	67
Analytics コマンド	67
Analytics の設定	68
設定の取得	68

設定の変更	69
Analytics のワークシート	70
ワークシートの一覧表示	70
分析ワークシートの取得	71
ワークシートの作成	71
ワークシート名の変更	72
ワークシートの破棄	72
ワークシートデータセットの一覧表示	73
ワークシートデータセットの追加	73
ワークシートデータセットの変更	74
Analytics のデータセット	74
データセットの一覧表示	76
データセットの取得	77
データセットの作成	77
データセットの変更	78
データセットの破棄	78
データセットの保存	79
データセットデータの除去	79
データセットデータの取得	80
 ハードウェアサービス	85
クラスタ	85
クラスタのプロパティの一覧表示	85
クラスタリソースの変更	87
クラスタリンクのステータス	87
クラスタ管理のコマンド	88
クラスタの設定	88
シャーシ	89
シャーシの一覧表示	89
シャーシのコンポーネントの取得	91
ハードウェアコンポーネントの取得	92
コンポーネントのプロパティの変更	93
 ログのコマンド	95
ログの管理コマンド	95
ログの一覧表示	95
ログエントリの取得	96
ログのダウンロード	98

ログのダウンロード	98
ネットワークのコマンド	99
ネットワーク構成	99
ネットワークデータリンク	100
ネットワークデータリンクの一覧表示	101
ネットワークデータリンクの取得	102
ネットワークデータリンクの作成	102
ネットワークデータリンクの変更	103
ネットワークデータリンクの削除	104
ネットワークデバイス	104
ネットワークデバイスの一覧表示	105
ネットワークデバイスの取得	105
ネットワークインタフェース	106
ネットワークインタフェースの一覧表示	107
ネットワークインタフェースの取得	107
ネットワークインタフェースの作成	108
ネットワークインタフェースの変更	108
ネットワークインタフェースの削除	109
ネットワークルート	109
ルートの一覧表示	110
ルートの取得	111
ルートの追加	111
ルートの削除	112
RESTful API クラウドサービス	113
クラウドサービス操作	113
クラウドサービスの有効化	115
クラウドサービスログファイルの表示	116
クラウドサービスのプロパティの一覧表示	116
クラウドサービスのプロパティの変更	117
ターゲットの一覧表示	117
ターゲットの作成	118
ターゲットの変更	119
ターゲットの削除	120
クラウドバックアップの一覧表示	120
クラウドバックアップの削除	123
クラウドバックアップの復元	124

ジョブの一覧表示	124
ジョブの取り消しまたは再起動	126
スナップショットバックアップ操作	127
スナップショットバックアップの一覧表示	128
スナップショットバックアップの作成	128
増分スナップショットバックアップの作成	129
増分スナップショットバックアップの親の検出	130
スナップショットバックアップの削除	132
RESTful API 問題サービス	133
問題サービスコマンド	133
すべての問題の一覧表示	133
一つの問題の一覧表示	134
問題の修復	135
問題通知の一時停止	135
通知の一時停止のステータスの表示	135
通知の一時停止	136
通知の再開	136
RESTful API SAN サービス	137
SAN の概要	137
SAN イニシエータ	138
イニシエータの一覧表示	139
イニシエータの詳細の取得	139
イニシエータの作成	140
イニシエータの変更	140
イニシエータの削除	141
イニシエータグループ	141
イニシエータグループの一覧表示	142
イニシエータグループの詳細の取得	143
イニシエータグループの作成	143
イニシエータグループの削除	144
ターゲット	144
ターゲットの一覧表示	145
ターゲットの詳細の取得	146
ターゲットの作成	147
ターゲットの変更	147
ターゲットの削除	148

ターゲットグループ	148
ターゲットグループの一覧表示	149
ターゲットグループの取得	150
ターゲットグループの作成	150
ターゲットグループの削除	151
 サービスコマンド	153
サービスコマンド	153
サービスの一覧表示	153
サービスの取得	154
サービスの状態の変更	155
サービス構成の変更	155
サービスリソース	158
 RESTful API ストレージサービス	159
ストレージプールの操作	159
プールの一覧表示	159
プールの取得	161
プールの構成	161
プールへのストレージの追加	163
プールからのストレージの削除	163
プールのスクラブ	164
プールの構成解除	165
プロジェクトの操作	166
プロジェクトの一覧表示	168
プロジェクトのプロパティの取得	169
プロジェクトの作成	170
プロジェクトの変更	171
プロジェクトの削除	172
プロジェクトの使用状況	172
ファイルシステムの操作	172
ファイルシステムの一覧表示	174
ファイルシステムの取得	175
ファイルシステムの作成	177
ファイルシステムの変更	178
ファイルシステムの削除	179
ファイルシステムの割り当て制限および使用状況	179
LUN 操作	179

LUNを一覧表示します	181
LUNの取得	182
新しいLUNの作成	183
LUNの変更	184
LUNの削除	185
スナップショットおよびクローンの操作	185
スナップショットの一覧表示	190
スナップショットの取得	190
スナップショットの作成	191
スナップショットの名前変更	192
スナップショットのクローン作成	192
スナップショットのロールバック	193
スナップショットの削除	194
スナップショット依存の一覧表示	195
スキーマ	196
プロパティの一覧表示	197
プロパティの取得	198
プロパティの作成	198
プロパティの変更	199
プロパティの削除	199
レプリケーション	199
レプリケーションサービスのプロパティの一覧表示	200
レプリケーションサービスのプロパティの変更	201
レプリケーションターゲット	201
レプリケーションターゲットの一覧表示	202
指定されたレプリケーションターゲットの一覧表示	202
レプリケーションターゲットの作成	203
ターゲット証明書の検証	204
レプリケーションターゲットの変更	205
レプリケーションターゲットの削除	206
レプリケーションアクション	206
フラットアクションインタフェースの使用	206
プロジェクト、ファイルシステム、またはLUNコンテキストでのレプリケーションアクション	208
レプリケーションアクションの一覧表示	211
レプリケーションアクションの取得	211
レプリケーションアクションの作成	213
レプリケーションアクションの変更	214
レプリケーションアクションの進行状況のモニター	216

更新の取り消し	218
更新の送信	218
レプリケーションアクションの削除	218
レプリケーションパッケージ	219
レプリケーションソースの一覧表示	223
レプリケーションパッケージの一覧表示	224
パッケージの変更	224
パッケージの削除	225
更新の取り消し	225
パッケージのクローン作成	226
パッケージの切断	227
パッケージの逆方向化	227
ストレージの暗号化	229
暗号化鍵の管理	229
ローカルキーストアの構成	230
OKM キーストアの構成	231
KMIP キーストアの構成	232
暗号化鍵の作成	233
暗号化鍵の一覧表示	233
指定した鍵で暗号化されたストレージの一覧表示	234
鍵の削除	235
暗号化されたプール、プロジェクト、またはシェアの作成	235
システムコマンド	237
アプライアンスシステムコマンド	237
バージョンの取得	238
システムの電源切断	238
システムのリブート	239
システム管理の再起動	239
診断リブート	239
出荷時リセット	240
システムのサポートバンドル	240
サポートバンドルの作成	240
サポートバンドルの一覧表示	241
サポートバンドルの取得	242
サポートバンドルの取り消し	243
サポートバンドルのアップロードの再試行	243

サポートバンドルのアップロード	244
サポートバンドルの削除	244
システムの更新	244
システム更新の一覧表示	245
システム更新の取得	246
プラットフォームファームウェアの更新ステータスの取得	247
コンポーネントファームウェアの更新ステータスの取得	247
システム更新のアップロード	248
アップグレード	248
ロールバック	249
更新イメージの削除	249
RESTful API ユーザーサービス	251
ユーザーサービスのコマンド	251
ユーザーサービスのプロパティ	252
ユーザープロパティ	252
ユーザーのロールおよび例外	254
ユーザー設定のプロパティ	254
CLI タイムアウト	255
SSH 鍵	256
REST ログイントークン	256
ユーザーの一覧表示	257
特定のユーザーの一覧表示	257
ユーザーの作成	258
ユーザープロパティの変更	260
ユーザーの削除	261
トークンの管理	261
RESTful API ロールサービス	265
ロールサービスコマンド	265
ロールサービスのプロパティ	266
ロールの一覧表示	266
ロールの取得	267
ロールの作成	267
ロールの変更	268
ロールの取り消し	269
ロールの削除	269
ロールの承認の一覧表示	269

ロールの承認の作成	270
ロールの承認の変更	270
ロールの承認の削除	271
ワークフローおよびスクリプトコマンド	273
ワークフローおよびスクリプトサービスコマンド	273
ワークフローのアップロード	274
ワークフローの一覧表示	274
ワークフローの取得	275
ワークフローの変更	277
ワークフローの実行	277
ワークフローの削除	278
スクリプトのアップロードおよび実行	278
実行中のすべてのスクリプトの一覧表示	279
実行中のスクリプトへの再接続	280
実行中のスクリプトの停止	281
RESTful クライアント	283
curl REST クライアント	283
リソースデータの取得	283
新しいリソースの作成	284
既存のリソースの変更	285
既存のリソースの削除	285
Python RESTful クライアント	285
リソースの取得	286
リソースの作成	286
リソースの変更	287
既存のリソースの削除	288

Oracle ZFS Storage Appliance RESTful API の概要

Oracle ZFS Storage Appliance は、ネットワーク経由で効率的なファイルおよびブロックデータサービスを提供します。このガイドでは、アプライアンスの管理に使用できる Oracle ZFS Storage Appliance RESTful アプリケーションプログラミングインタフェース (API) について説明します。RESTful アーキテクチャーは、クライアントを構成せずに標準のハブ、ルーター、およびその他のネットワークシステムを介してサービスを透過的にリダイレクトする階層化されたクライアントサーバーモデルに基づいています。

RESTful API 認証

Oracle ZFS Storage Appliance RESTful API では、ブラウザユーザーインタフェース (BUI) およびコマンド行インタフェース (CLI) と同じ認証資格を使用します。外部クライアントからのリクエストはすべて、アプライアンスの資格を使って個々に認証され、ポート 215 の HTTPS 接続を介して実行されます。RESTful API は、ユーザーが 15 分のタイムアウトを定義できる HTTPS セッションをサポートしています。

認証は次のいずれかの形式にすることができます。

- **基本認証** – 各リクエストにユーザーログインが含まれている必要があります。認証文字列は、`username:password` を連結したもので、Base64でエンコードされます。

HTTP ヘッダーの例:

```
Authorization: Basic Tm8gcGV1a2luZyE=
```

- **ユーザー認証** – BUI または CLI ログイン資格証明を認証に使用します。この場合、X-Auth-User ヘッダーにログイン名が含まれている必要があり、X-Auth-Key ヘッダーにログインパスワードが含まれている必要があります。

HTTP ヘッダーの例:

```
X-Auth-User: login-name  
X-Auth-Key: password-xxx
```

- **トークン認証** – トークンが認証されると、トークンが期限切れになるまで、トークンヘッダーを使用して、コマンドを実行し続けることができます。トークンの期限

が切れたあとは、コマンドが受け付けられる前に再度認証を実行する必要があります。

トークンヘッダーの例:

```
X-Auth-Session: qYftpufrrTx1DztkMh11LoyTfSDUSIR
```

RESTful API バージョン

ある特定のアプライアンスリリースの RESTful API バージョンには、アプライアンスソフトウェアのバージョンに一致するグローバルバージョン番号があります。このバージョン番号はすべてのリクエストの応答ヘッダーで返されます。

```
X-Zfssa-Version: nas.2013.1.1
```

RESTful API サービスバージョン

各 RESTful API サービスは、サービスにアクセスするための URI (Uniform Resource Identifier) の一部としてバージョン番号を持ちます。このバージョンにはメジャー番号とマイナー番号があります。リクエストではメジャーバージョン番号を指定する必要がありますが、マイナーバージョン番号はオプションで、指定しない場合は、デフォルトで 0 の値に設定されます。メジャー番号はサービスのメジャー番号と一致する必要があります。マイナー番号はサービスのマイナー番号以下である必要があります。

たとえば、次の表は、バージョン 2.1 が動作しているサービスをクライアントがリクエストする場合、指定されたバージョンをクライアントのリクエストで使用できるかどうかを示しています。

リクエストのバージョン	許可
1	いいえ: メジャーバージョンが、サービスで動作しているバージョンと一致していません。
2	はい: メジャーバージョンが一致しており、マイナーバージョン (デフォルトは 0) には下位互換性があります。
2.1	はい: メジャーバージョンおよびマイナーバージョンの値が、サービスで動作しているバージョンと一致しています。
2.2	いいえ: メジャーバージョンは一致していますが、マイナーバージョンが、サービスで動作しているバージョンよりも新しいものです。

次のプロパティの変更には、サービスの API バージョンの変更は必要ありません。使用可能なプロパティを判断するために、アプライアンスのバージョン番号とモデ

ルを使用する必要があります。これらのプロパティの変更は、CLI および BUI にも反映され、そのアプライアンスインスタンスの機能を示します。

- 新しい出力プロパティ (古いプロパティの削除なし)。
- 既存のコマンドに追加された、コマンドに以前のバージョンと同じ動作をさせるデフォルト値を持つ新しい入力プロパティ。

下位互換性のあるコマンドの新しいバージョンでは、追加のプロパティを返すことができるため、クライアントは新しいプロパティを無視するようにコーディングしてください。マイナー番号は、サービス API への下位互換性のある変更に対して増分されます。

- 既存のサービスに新しいコマンドを追加する。
- サービスのコマンドに新しい問合せパラメータを追加する。

メジャー番号は、サービス API への互換性のない変更に対して増分されます。

- コマンド問合せパラメータの削除。
- 既存のサービスからのコマンドの削除。

アプライアンスソフトウェアのメジャーリリースには、互換性のないバージョンの変更が含まれることがあります。メジャー更新時には、指定されたサービスの古いバージョンが存在する場合と存在しない場合があります。各コマンドの応答には、指定されたモジュールに対応するアプライアンス API の現在のバージョンを含めた HTTP ヘッダーを含める必要があります。

X-Zfssa-Nas-API: 1.1

RESTful API サービスバージョン 2.0

このセクションでは、RESTful API サービスバージョン 2 と RESTful API サービスバージョン 1 の違いについて説明します。

- [17 ページの「スクリプト可能な値」](#)
- [18 ページの「一貫した値」](#)

RESTful API バージョン 2 と RESTful API バージョン 1 の両方が同時に使用可能であり、このガイドの残りの部分では、バージョン 1 の例を示します。リクエスト URI のサービスバージョン部分 (v1 または v2) を使用して、使用する REST API バージョンを選択します。

スクリプト可能な値

RESTful API バージョン 2 の操作は、常にスクリプト可能な値を返します。スクリプト可能な値には、プロパティのタイプごとに同一の安定した形式があります。

RESTful API バージョン 1 の操作は、多くの場合、スクリプト可能な値を返しますが、必ずしもそうではありません。たとえば、RESTful API バージョン 1 では、完全な Javascript 日付形式で日時文字列を返す場合と、ISO 8601 日時形式で日時文字列を返す場合があります。RESTful API バージョン 2 では、常に ISO 8601 日時形式の日時文字列を返します。

次の例では、GET /api/system/v1/updates 操作は完全な Javascript 日付形式で日時文字列を返し、GET /api/system/v2/updates 操作は ISO 8601 日時形式で日時文字列を返します。

```
GET /api/system/v1/updates
{
  "updates": [{
    "status": "previous",
    "href": "/api/system/v1/updates/ak-nas@2013.06.05.4.0,1-1.7",
    "release_date": "Fri May 01 2015 20:13:00 GMT+0000 (UTC)",
    "install_date": "Tue Nov 15 2016 01:01:07 GMT+0000 (UTC)",
    "version": "2013.06.05.4.0,1-1.7",
    "date": "Fri May 01 2015 20:13:00 GMT+0000 (UTC)"
  }]
}
GET /api/system/v2/updates
{
  "updates": [{
    "status": "previous",
    "href": "/api/system/v2/updates/ak-nas@2013.06.05.4.0,1-1.7",
    "release_date": "2015-05-01T20:13:00Z",
    "install_date": "2016-11-15T01:01:07Z",
    "version": "2013.06.05.4.0,1-1.7",
    "date": "2015-05-01T20:13:00Z"
  }]
}
```

一貫した値

RESTful API バージョン 1 の操作では、プロパティへのアクセス方法に応じて、同じプロパティに対して異なる値が返されることがあります。RESTful API バージョン 2 の操作では、プロパティへのアクセス方法に関係なく、一貫した値が返されます。

次の例では、すべてのレプリケーションアクションが一覧表示されている場合、max_bandwidth プロパティの値は -1 として返されます。

```
GET /api/storage/v1/replication/actions
{
  "actions": [{
    "id": "71b1b8b9-9c57-c969-aab9-f96d5f4e5d54",
    ...
    "max_bandwidth": -1,
    ...
  }]
}
```

レプリケーションアクションが 1 つのみ指定されている場合、基になる値が変更されていなくても、max_bandwidth プロパティの値は 0 として返されます。

```
GET /api/storage/v1/replication/actions/71b1b8b9-9c57-c969-aab9-f96d5f4e5d54
{
  "action": {
    "id": "71b1b8b9-9c57-c969-aab9-f96d5f4e5d54",
    ...
    "max_bandwidth": 0,
    ...
  }
}
```

RESTful API バージョン 2 の操作では、特定のプロパティ値へのアクセス方法に関係なく、そのプロパティに対して常に同じ値を返します。

```
GET /api/storage/v2/replication/actions
{
  "actions": [{
    "id": "71b1b8b9-9c57-c969-aab9-f96d5f4e5d54",
    ...
    "max_bandwidth": -1,
    ...
  }]
}
GET /api/storage/v2/replication/actions/71b1b8b9-9c57-c969-aab9-f96d5f4e5d54
{
  "action": {
    "id": "71b1b8b9-9c57-c969-aab9-f96d5f4e5d54",
    ...
    "max_bandwidth": -1,
    ...
  }
}
```

一般的な RESTful 操作

次の表は指定されたリソースに対する一般的な RESTful 操作を示しています。

表 1 一般的な RESTful 操作

リクエスト	パス	説明
GET	resources	すべてのリソースを一覧表示します
GET	resources/name	選択されたリソースを記述する JSON オブジェクトを取得します
POST	resources	新しいリソースを作成します
PUT	resources/name	選択されたリソースを変更します
DELETE	resources/name	選択されたリソースを削除します

HTTP 応答本文

すべての応答データは、[RFC 4627](#) で定義された JSON 形式でエンコードされます。ほかに指定がないかぎり、単一のリソースに対するコマンドは、プロパティとしてリ

ソース名を持つ単一の JSON 結果オブジェクトを返します。各コマンドセクションでは、この JSON 結果オブジェクトで返されるプロパティ名を示しています。

ほかに記述がないかぎり、作成 (POST) および変更 (PUT) コマンドは、作成または変更されたリソースのプロパティを返します。内容は GET リクエストによって返される値と一致します。

本文の例:

```
{
  "resource_name": {
    "href": "path/to/this/resource",
    "property_01": "value_01",
    "property_02": "value_01"
  }
}
```

一部の GET コマンドはリソースのリストを返します。

```
{
  "resource_list_name": [
    {
      "href": "path/to/resource_01",
      "property_01": "value_01"
    }, {
      "href": "path/to/resource_02",
      "property_02": "value_02"
    }
  ]
}
```

注記 - このドキュメント全体で、コマンドでは、読みやすくするために改行とスペースを追加して書式設定された JSON 戻り結果を示しています。実際の出力ではこの書式設定が含まれません。

HTTP 応答ヘッダー

データを送信するすべてのアプライアンスサービスコマンドは JSON データ形式を使用し、次のヘッダー値が必要です。

```
Accept: application/json
Content-Type: application/json
```

応答ヘッダーには次の情報が含まれます。

```
Date: Tue, 23 Jul 2013 13:07:37 GMT X-Zfs-Sa-Appliance-API: 1.0 Content-Type: application/
json Content-Length: 357
```

リストの結果の場合、データが返送されるまで、内容の長さがわからない場合があります。内容の長さが指定されていない場合、クライアントは、返されるすべてのデータを読み取るために、EOF まで応答本文を読み取る必要があります。

問合せパラメータ

一部のリクエストは、返されるデータや操作されるデータを選択するオプションの問合せパラメータを受け取ることができます。このセクションでは、複数のタイプのリソースで利用できる問合せパラメータについて説明します。各リソースに固有の問合せパラメータ、およびこのセクションで説明する問合せパラメータの特殊な使用方法については、各リソースのドキュメントを参照してください。

表 2 一般的な問合せパラメータ

パラメータ	説明
<code>props=true</code>	リソースのプロパティメタデータを一覧表示します。デフォルト値は <code>false</code> です。
<code>start=index</code>	指定された時間またはオブジェクトIDの後に返す最も古いデータまたはオブジェクトを指定します。
<code>end=index</code>	指定された時間またはオブジェクトIDの前に返す最も新しいデータまたはオブジェクトを指定します。
<code>limit=n</code>	n 個未満のリスト要素を返します。
<code>depth=n</code>	返されるデータの詳細レベルを指定します。
<code>match_property-name=value</code>	指定されたプロパティを指定された値で持っているリストオブジェクトを返します。

問合せパラメータ: props

`props` 問合せパラメータは、プロパティメタデータ値を表示します。データを変更したり、新しいデータを作成したりする操作で `props=true` を使用すると、プロパティとメタデータが表示され、操作は実行されません。これにより、リソースの変更や作成に役立つ現在のデータ値を表示できます。

表 3 プロパティメタデータ値

プロパティ	説明
<code>name</code>	プロパティ名
<code>label</code>	プロパティの説明
<code>immutable</code>	プロパティを変更できないことを示すフラグ
<code>type</code>	プロパティの型 (String、Integer、Boolean、ChooseOne など)
<code>choices</code>	列挙されたプロパティの場合、使用可能な値の配列

問合せパラメータ: start

`start` 問合せパラメータには、オブジェクト索引番号または時間を指定します。

- オブジェクト索引番号を指定すると、その索引によって選択されたオブジェクトと、その指定されたオブジェクトの作成後に作成された最も古いオブジェクトを含むリストを返します。
- UTC 時間を指定して、指定した時刻以降に作成された最も古いオブジェクトまたはデータのリストを返します。一部のリソースは、**start** 問合せパラメータの時間値をサポートしていません。

時間値は、次の表に示す形式で UTC 時間で表す必要があります。

サービスのバージョン	時間値の形式	時間値の例
v1 paths	%Y%m%dT%H:%M:%SZ	20200723T14:11:49
v2 paths	%Y-%m-%dT%H:%M:%SZ	2020-07-23T14:11:49Z

start 問合せパラメータを使用した例については、次のセクションを参照してください。

- [80 ページの「データセットデータの取得」](#)
- [120 ページの「クラウドバックアップの一覧表示」](#)

問合せパラメータ: end

end 問合せパラメータには、オブジェクト索引番号または時間を指定します。

- オブジェクト索引番号を指定すると、その索引によって選択されたオブジェクトと、その指定されたオブジェクトの作成前に作成された最も新しいオブジェクトを含むリストを返します。
- UTC 時間を指定して、指定した時刻以前に作成された最も新しいオブジェクトまたはデータのリストを返します。

時間値は、%Y-%m-%dT%H:%M:%SZ の形式で UTC 時間で表す必要があります。

end 問合せパラメータの使用例については、[120 ページの「クラウドバックアップの一覧表示」](#)を参照してください。

問合せパラメータ: limit

limit 問合せパラメータは、返す要素の最大数を指定します。

start も **end** も指定されていない場合、**limit=n** は *n* 個の最新の要素を返します。

問合せパラメータ: depth

depth 問合せパラメータは、返されるリソースのリストの詳細レベルを指定します。次の表に示すように、depth の値が大きいほど、より詳細な情報が返されます。

depth の値	リストに返される情報
depth=0	ノードのプロパティおよび子の名前
depth=1	ノードのプロパティ、子の名前とプロパティ、および孫の名前
depth=2	ノードのプロパティ、子の名前とプロパティ、孫の depth=0 出力

注記 - depth 問合せパラメータは、/api/log/v{1|2} を使用したログの一覧表示、または /api/storage/v{1|2} を使用したプール、プロジェクト、ファイルシステム、または LUN の一覧表示ではサポートされていません。

問合せパラメータ depth を使用したユーザーのリストのリクエストの例:

```
GET /api/user/v1/users?depth=2 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-User: root
X-Auth-Key: password-xxx
```

レスポンスの例:

簡潔にするために、追加のユーザーは省略しています。

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 1558
X-Zfssa-Access-API: 1.0

{
  "users": [
    {
      "name": "root",
      ...
    },
    {
      "name": "firstlast",
      "properties": {
        "logname": "firstlast",
        "type": "directory",
        "uid": uid,
        "fullname": "First Last",
        "require_annotation": false,
        "roles": [
          "basic"
        ],
        "kiosk_mode": false,
        "kiosk_screen": "status/dashboard"
      },
      "children": [
        {
          "name": "exceptions",
          "properties": {},
          ...
        }
      ]
    }
  ]
}
```

```
    "children": [],
    "list": [
      {
        "name": "auth-000",
        "properties": {
          "scope": "ad",
          "name": "*",
          "allow_domain": true,
          "allow_workgroup": false
        },
        "children": [],
        "list": []
      },
      {
        "name": "auth-001",
        "properties": {
          "scope": "alert",
          "allow_configure": true,
          "allow_post": true
        },
        "children": [],
        "list": []
      }
    ]
  },
  {
    "name": "preferences",
    "properties": {
      "locale": "C",
      "login_screen": "configuration/preferences",
      "session_timeout": 15,
      "cli_idle_timeout": "infinite",
      "advanced_analytics": false
    },
    "children": [
      {
        "name": "keys",
        "properties": {},
        "children": [],
        "list": []
      },
      {
        "name": "tokens",
        "properties": {},
        "children": [],
        "list": []
      }
    ],
    "list": []
  },
  "list": [],
  "href": "/api/user/v1/users/firstlast"
},
{
  ...
}
]
```

問合せパラメータ: match

`matchproperty-name=value` 問合せパラメータは、指定されたプロパティ名を指定された値で持つリソースのリストを返します。

次の例では、`kiosk_mode` プロパティの値が `true` であるユーザーのリストを返します。

```
match_kiosk_mode=true
```

次の例では、`roles` プロパティの値が `super` を含み、`require_annotation` プロパティの値が `true` であるユーザーのリストを返します。

```
match_roles='*super*&match_require_annotation=true
```

注記 - `match_property-name=value` 問合せパラメータは、`/api/log/v{1|2}` を使用したログの一覧表示、または `/api/storage/v{1|2}` を使用したプール、プロジェクト、ファイルシステム、または LUN の一覧表示ではサポートされていません。

アプライアンスエラー

エラーでは、次の障害応答のペイロードとともに、エラーを示す HTTP ステータスコードを返します。

JSON 障害応答:

```
{
  fault: {
    message: 'ERR_INVALID_ARG',
    details: 'Error Details...',
    code: 500
  }
}
```

表 4 一般的なエラーコード

エラー	コード	説明
ERR_INVALID_ARG	400	無効な入力引数
ERR_UNKNOWN_ARG	400	余分な未処理の入力引数
ERR_MISSING_ARG	400	必要な入力引数がありません
ERR_UNAUTHORIZED	401	このユーザーはコマンドを実行する権限がありません
ERR_DENIED	403	操作は拒否されました
ERR_STATE_CHANGED		システム状態の競合

エラー	コード	説明
ERR_NOT_FOUND	404	リクエストされた項目が見つかりませんでした
ERR_OBJECT_EXISTS	409	リクエストはすでに存在するオブジェクトを作成します
ERR_CONFIRM_REQUIRED	409	リクエストを完了するために ?confirm=true 問合せパラメータが必要です
ERR_OVER_LIMIT	413	入力されたリクエストは処理するには大きすぎます
ERR_UNSUPPORTED_MEDIA	415	リクエストされたメディアタイプがリクエストでサポートされていません
ERR_NOT_IMPLEMENTED	501	操作は実装されていません
ERR_BUSY	503	リソースが制限されているためサービスを使用できません

セキュリティープロトコルおよび暗号の設定

プロトコルバージョンおよび関連する暗号のコマンドは、アプライアンスにアクセスするための SSL/TLS プロトコルバージョンおよび暗号を管理します。

デフォルトでは、SSL/TLS プロトコルバージョン TLSv1.1、TLSv1.2、およびそれらに関連する暗号が有効になっています。HTTPS サービスに PUT リクエストを送信して `tls_version` プロパティを設定することにより、TLSv1.0 を有効にできます。

リクエストの例:

```
PUT /api/service/v1/services/https HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json

{ "tls_version": ["TLSv1.0", "TLSv1.1", "TLSv1.2"] }
```

結果の例 (簡潔にするために、出力は省略しています):

```
HTTP/1.1 202 Accepted
Content-Length: 1265
X-Zfssa-Service-API: 1.1
X-Zfssa-API-Version: 1.0
Content-Type: application/json; charset=utf-8

{
  "service": {
    "href": "/api/service/v1/services/https",
    "<status>": "online",
    "tls_version": "TLSv1 TLSv1.1 TLSv1.2",
    "ciphers": "SRP-DSS-AES-256-CBC-SHA:SRP-RSA-AES-256-CBC-SHA:SRP-AES-256-CBC-SHA:
    ...
    3DES-EDE-CBC-SHA:EDH-RSA-DES-CBC3-SHA:EDH-DSS-DES-CBC3-SHA:DH-RSA-DES-CBC3-SHA:
    DH-DSS-DES-CBC3-SHA:DES-CBC3-SHA"
  }
}
```

TLSv1.0 のみを有効にするには、`ciphers` プロパティを、TLSv1.0 のみで使用可能な暗号のリストに設定します。

リクエストの例 (簡潔にするために、出力は省略しています):

```
PUT /api/service/v1/services/https HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json

{
  "tls_version": ["TLSv1.0"],
  "ciphers": ["SRP-DSS-AES-256-CBC-SHA", "SRP-RSA-AES-256-CBC-SHA", "SRP-AES-256-CBC-SHA",
  ...
  "EDH-RSA-DES-CBC3-SHA", "EDH-DSS-DES-CBC3-SHA", "DH-RSA-DES-CBC3-SHA", "DH-DSS-DES-CBC3-SHA",
  "DES-CBC3-SHA"]
}
```

結果の例 (簡潔にするために、出力は省略しています):

```
HTTP/1.1 202 Accepted
Content-Length: 809
X-Zfssa-Service-API: 1.1
X-Zfssa-API-Version: 1.0
Content-Type: application/json; charset=utf-8

{
  "service": {
    "href": "/api/service/v1/services/https",
    "<status>": "online", "tls_version": "TLSv1",
    "ciphers": "SRP-DSS-AES-256-CBC-SHA:SRP-RSA-AES-256-CBC-SHA:SRP-AES-256-CBC-SHA:
    ...
    3DES-EDE-CBC-SHA:SRP-3DES-EDE-CBC-SHA:EDH-RSA-DES-CBC3-SHA:EDH-DSS-DES-CBC3-SHA:DH-
    RSA-DES-CBC3-SHA:DH-DSS-DES-CBC3-SHA:DES-CBC3-SHA"
  }
}
```

注記 - RESTful API または BUI を使用できなくなることを防ぐために、特に必要がある場合または Oracle サポートから指示された場合を除き、`tls_version` および `ciphers` プロパティのデフォルト設定を保持してください。

パスワードの複雑性

パスワードの RESTful API を使用すると、`changeProperties` 承認を持っているユーザーがすべてのローカルユーザーのパスワード複雑性規則を設定できます。ユーザー承認の詳細は、[265 ページの「RESTful API ロールサービス」](#)を参照してください。

ローカルユーザーがパスワードを変更する際に、パスワードの要件が強制されます。パスワードの規則が変更されても、既存のパスワードは影響を受けません。

変更可能なプロパティを表示するには、次のコマンドを使用します。

リクエストの例:

```
GET /api/setting/v2/password HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Date: Fri, 14 May 2021 17:07:39 GMT
Content-Type: application/json; charset=utf-8
X-Zfssa-API-Version: 2.0
X-Zfssa-Setting-API: 2.0
Content-Length: 196
```

```
{
  "complexity": {
    "href": "/api/setting/v2/password",
    "passlength": 8,
    "min_letters": 0,
    "min_upper": 0,
    "min_lower": 0,
    "min_digit": 0,
    "min_punctuation": 0,
    "max_repeats": 0,
    "namecheck": true
  }
}
```

これらのプロパティの説明については、『[Oracle ZFS Storage Appliance 管理ガイド、Release OS8.8.x](#)』の「パスワード複雑性のプロパティ」を参照してください。

次の例では、大文字、小文字、数字、句読文字がそれぞれ1つ以上必要になるようにパスワードの規則を変更します。min_letters の値は、min_upper および min_lower の新しい値に合わせて更新する必要があります。

リクエストの例:

```
PUT /api/setting/v2/password HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
```

```
{"min_letters": 2, "min_upper": 1, "min_lower": 1, "min_digit": 1, "min_punctuation": 1}
```

結果例:

```
HTTP/1.1 202 Accepted
Date: Fri, 14 May 2021 17:38:40 GMT
Content-Type: application/json; charset=utf-8
X-Zfssa-API-Version: 2.0
X-Zfssa-Setting-API: 2.0
Content-Length: 196
```

```
{
  "complexity": {
    "href": "/api/setting/v2/password",
    "passlength": 8,
    "min_letters": 2,
    "min_upper": 1,
    "min_lower": 1,
  }
}
```

```
    "min_digit": 1,  
    "min_punctuation": 1,  
    "max_repeats": 0,  
    "namecheck": true  
  }  
}
```


RESTful API の操作

アクセスサービスは、Oracle ZFS Storage Appliance 上のすべての RESTful API サービスに対するエントリポイントです。このサービスは、ユーザー資格証明を認証したり、使用可能な RESTful API サービスをバージョンおよびアクセスポイントとともに一覧表示したりするために使用されます。

サービスへのアクセス

サービスにアクセスするには、次のいずれかの URL を使用します。

- `http://hostname:215/api/access/v2`
- `http://hostname:215/api/access/v1`

その他のサービスにアクセスするには、アクセスサービスを使用してログインし、使用可能なサービスの場所およびバージョンを取得してから、返された URI を使用してそれらのサービスにアクセスします。サービスの場所は、現在のアプライアンス構成またはリリースレベルに基づいて変更できます。

表 5 アクセスサービスのコマンド

リクエスト	パス	説明
GET	<code>/api/access/v{1 2}</code>	RESTful API サービスのアクセスポイントを一覧表示します
POST	<code>/api/access/v{1 2}</code>	非永続ログイントークンを作成します
DELETE	<code>/api/access/v{1 2}</code>	ログアウトし、非永続ログイントークンを削除します

サービスの一覧表示

`list services` コマンドは、使用可能なサービスアクセス URI を一覧表示します。ログインセッションが必要ない場合は、`list services` で適切な資格証明を使用して使用可能なサービスアクセス URI を一覧表示します。このコマンドは、アプライアンスで使用可能なすべての RESTful API サービスおよびバージョンを一覧表示します。

リクエストの例:

```
GET /api/access/v1 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-User: admin1
X-Auth-Key: password
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 190
X-Zfssa-Access-API: 1.0

{
  "services": [
    {
      "version": "1.1",
      "name": "hardware",
      "uri": "https://hostname:215/api/hardware/v1"
    },
    {
      "version": "1.0",
      "name": "san",
      "uri": "https://hostname:215/api/san/v1"
    },
    {
      "version": "1.0",
      "name": "network",
      "uri": "https://hostname:215/api/network/v1"
    }
  ]
}
```

サービス操作の一覧表示

このコマンドは、指定されたサービスで使用可能な操作 (メソッド) のリストを返します。該当する場合、このコマンドは、指定されたサービスのリソースに関する情報を返します。次の例では、クラスタハードウェアコンポーネントに、さらに調査および構成できるリソースがあります。

リクエストの例。このリクエストでは認証トークンを使用していることに注意してください。 [33 ページの「認証トークン」](#) を参照してください。

```
GET /api/hardware/v1 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-Session: puPnHZKgSrUmXqYz0wFCrGcLOGwPODj
```

結果の例。簡潔にするために、この出力のほとんどを省略しています。

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 204
X-Zfssa-Access-API: 1.0

{
```

```

"service": {
  "methods": [
    {
      "path": "",
      "href": "/api/hardware/v1",
      "request": "GET",
      "description": "List the hardware service commands."
    },
    {
      "path": "/cluster",
      "href": "/api/hardware/v1/cluster",
      "request": "GET",
      "description": "Get cluster properties and cluster resource list"
    },
    {
      "path": "/cluster/resources/<resource:path>",
      "href": "/api/hardware/v1/cluster/resources/<resource:path>",
      "request": "GET",
      "description": "Get properties for the specified cluster resource"
    },
    {
      "path": "/cluster/resources/<resource:path>",
      "href": "/api/hardware/v1/cluster/resources/<resource:path>",
      "request": "PUT",
      "description": "Modify the specified cluster resource"
    },
    {
      "path": "/chassis",
      "href": "/api/hardware/v1/chassis",
      "request": "GET",
      "description": "List hardware chassis"
    }
  ],
  "version": "1.1",
  "name": "hardware",
  "uri": "https://hostname:215/api/hardware/v1"
}

```

認証トークン

非永続ログイントークンは、POST リクエストを送信することでアクセスサービスから取得します。この非永続ログイントークンは、他のすべてのサービスでアイデンティティ資格証明として使用できます。非永続ログイントークンは、ユーザーのセッションタイムアウトプロパティによって設定されるタイムアウト期間後に無効化されます。通常、デフォルトは、15 分です。DELETE リクエストは、ログアウトして非永続ログイントークンを無効化するために使用できます。

この非永続ログイントークンは、以前の認証セッション ID と同等です。これは、RESTful API バージョン 2 と RESTful API バージョン 1 の両方でサポートされています。これは、ID が作成されたクラスターノードに固有のもので、クラスターピア間では同期されません。

ユーザーは、RESTful API にアクセスするための永続トークンを作成することもできます。永続トークンの作成は、RESTful API バージョン 2 以降でのみサポートされて

います。永続トークンはクラスタピア間で同期されているため、あるクラスタノードで作成し、他のクラスタノードとの通信に使用できます。[251 ページの「RESTful API ユーザーサービス」](#)を参照してください。

非永続ログイントークンの作成

POST リクエストは、新しい非永続ログイントークンをリクエストします。成功すると、使用可能な RESTful API サービスのリストが格納された単一プロパティ `access` を持つ JSON オブジェクトとともに、HTTP ステータス 201 が返されます。トークンの名前を設定するために、オプションのプロパティ `name` を使用できます。

作成リクエストの例:

```
POST /api/access/v2 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-User: root
X-Auth-Key: password-xxx
```

リクエストに成功すると、HTTP ステータス 201 (Created) と、X-Auth-Session HTTP ヘッダーを介して非永続ログイントークンを返します。レスポンス本文には、このログインを介してアクセス可能なサービスのリストが含まれます。

レスポンスヘッダー:

```
HTTP/1.1 201 Created
X-Auth-Session: puPnHZKgSrUmXqYzOwFCrGcLOGwPODj
X-Auth-Name: REST-YG02oRod
Content-Type: application/json
Content-Length: 378
X-Zfssa-Access-API: 1.0
```

```
{
  "access": {
    "services": [{
      ...
    }]
  }
}
```

ログアウトおよび非永続ログイントークンの削除

空の DELETE は、ログアウトして非永続ログイントークンを無効にするリクエストを送信します。

ログアウトリクエストの例:

```
DELETE /api/access/v2 HTTP/1.1
X-Auth-Session: puPnHZKgSrUmXqYzOwFCrGcLOGwPODj
```

結果例:

HTTP/1.1 204 No Content
X-Zfssa-Access-API: 1.0

証明書の管理

RESTful API を使用すると、証明書署名リクエスト (CSR)、システムまたは信頼できるユーザー証明書、および認証局 (CA) 証明書を管理できます。

次の表では、*request* は CSR の *uuid* であり、*certificate* はシステムまたは信頼できるユーザー証明書、あるいは CA 証明書の *uuid* です。

表 6 証明書操作

リクエスト	パス <i>/api/setting/v{1 2}</i> の後ろに追加	説明
GET	<i>/certificates/system/template</i>	テンプレートの CSR を返します。
POST	<i>/certificates/system</i>	CSR を作成します。 新しいシステム証明書をアップロードします。
GET	<i>/certificates/system/request</i>	指定された CSR のプロパティを一覧表示します。 CSR を PEM 形式で返します。
GET	<i>/certificates/system</i>	すべてのシステム証明書およびリクエストのプロパティを一覧表示します。
PUT	<i>/certificates/system</i>	デフォルトのシステム証明書の値を設定します。
GET	<i>/certificates/system/certificate</i>	指定されたシステム証明書のプロパティを一覧表示します。 証明書を PEM 形式で返します。
DELETE	<i>/certificates/system/certificate</i>	指定されたシステム証明書を破棄します。
GET	<i>/certificates/trusted</i>	すべての信頼できる証明書のプロパティを一覧表示します。
POST	<i>/certificates/trusted</i>	新しい信頼できる証明書をアップロードします。
GET	<i>/certificates/trusted/certificate</i>	指定された信頼できる証明書のプロパティを一覧表示します。 証明書を PEM 形式で返します。
PUT	<i>/certificates/trusted/certificate</i>	指定された信頼できる証明書の <i>services</i> プロパティの値を設定します。
DELETE	<i>/certificates/trusted/certificate</i>	指定された信頼できる証明書を破棄します。

証明書の一覧表示

次のリクエストは、ホスト上のすべてのシステム証明書のプロパティを一覧表示し、**default** プロパティの値を一覧表示します。

リクエストの例:

```
GET /api/setting/v2/certificates/system HTTP/1.1
Host: alice.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

結果例:

次の例の最初の証明書は、(ドメインまたは IP アドレスに基づいて) 自動生成された従来の証明書です。2 番目の証明書は、アプライアンスシリアル番号 (ASN) UUID に基づいて自動生成された証明書です。

この結果の最後にある **default** プロパティの値は、システムのデフォルト証明書が自動的に選択されることを示しています。

```
HTTP/1.1 200 OK
Date: Sat, 08 May 2021 00:37:21 GMT
Content-Type: application/json; charset=utf-8
X-Zfssa-API-Version: 2.0
X-Zfssa-Setting-API: 2.0
Content-Length: 1975

{
  "certificates": [
    {
      "uuid": "system-cert1-uuid",
      "type": "cert",
      "data": {
        "subject": [
          {
            "countryName": "US"
          },
          {
            "stateOrProvinceName": "CA"
          },
          {
            "localityName": "Exampleton"
          },
          {
            "organizationName": "Example Corp, Inc"
          },
          {
            "commonName": "alice.example.com"
          }
        ],
        "issuer": [
          {
            "countryName": "US"
          },
          {
            "stateOrProvinceName": "AK"
          },
          {
            "localityName": "Trustville"
          }
        ]
      }
    }
  ]
}
```

```

    },
    {
      "organizationName": "Totally Trustworthy Certificates, Inc"
    },
    {
      "commonName": "Most Trusted Certificate"
    }
  ],
  "serialNumber": "64",
  "validity": {
    "notBefore": "20210520T21:08:27",
    "notAfter": "20220520T21:08:27"
  },
  "extensions": {
    "basicConstraints": {
      "value": [
        {
          "CA": false
        }
      ]
    },
    "subjectKeyIdentifier": {
      "value": "subjectKeyIdentifier"
    },
    "authorityKeyIdentifier": {
      "value": [
        {
          "keyid": "authorityKeyIdentifier"
        }
      ]
    },
    "subjectAltName": {
      "value": [
        {
          "DNS": "alice.example.com"
        },
        {
          "IP": "alice.example.com-ipaddr"
        }
      ]
    }
  }
},
"sha1fingerprint": "sha1fingerprint",
"href": "/api/setting/v2/certificates/system/system-cert1-uuid"
},
{
  "uuid": "system-cert2-uuid",
  "type": "cert",
  "asn": "8bf7f9bc-8b3a-4064-e59f-bf09e3dba275",
  "data": {
    "subject": [
      {
        "commonName": "8bf7f9bc-8b3a-4064-e59f-bf09e3dba275"
      }
    ],
    "issuer": [
      {
        "commonName": "8bf7f9bc-8b3a-4064-e59f-bf09e3dba275"
      }
    ],
    "serialNumber": "59:8A:73:7B:00:00:00:07",
    "validity": {
      "notBefore": "20060215T18:00:00",

```

```

        "notAfter": "20380119T03:14:07"
      },
      "extensions": {
        "nsComment": {
          "value": "Automatically generated"
        },
        "subjectAltName": {
          "critical": true,
          "value": [
            {
              "DirName": "8bf7f9bc-8b3a-4064-e59f-bf09e3dba275"
            }
          ]
        }
      }
    },
    "sha1fingerprint": "sha1fingerprint",
    "href": "/api/setting/v2/certificates/system/system-cert2-uuid"
  },
  "default": "auto"
}

```

次のリクエストは、指定された信頼できる証明書 *trusted-cert1-uuid* のプロパティを一覧表示します。

リクエストの例:

```

GET /api/setting/v2/certificates/trusted/trusted-cert1-uuid HTTP/1.1
Host: alice.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json

```

結果例:

```

HTTP/1.1 200 OK
Date: Sat, 08 May 2021 00:37:57 GMT
Content-Length: 984
Content-Type: application/json; charset=utf-8
X-Zfssa-Setting-API: 2.0
X-Zfssa-API-Version: 2.0

{
  "certificate": {
    "uuid": "trusted-cert1-uuid",
    "type": "cert_ca",
    "data": {
      "subject": [
        {
          "countryName": "US"
        },
        {
          "stateOrProvinceName": "AK"
        },
        {
          "localityName": "Trustville"
        },
        {
          "organizationName": "Totally Trustworthy Certificates, Inc"
        },
        {
          "commonName": "Most Trusted Certificate"
        }
      ]
    }
  }
}

```

```

    ],
    "issuer": [
      {
        "countryName": "US"
      },
      {
        "stateOrProvinceName": "AK"
      },
      {
        "localityName": "Trustville"
      },
      {
        "organizationName": "Totally Trustworthy Certificates, Inc"
      },
      {
        "commonName": "Most Trusted Certificate"
      }
    ],
    "serialNumber": "83:F7:79:02:5F:44:4D:60",
    "validity": {
      "notBefore": "20210316T17:28:37",
      "notAfter": "20210415T17:28:37"
    },
    "extensions": {
      "subjectKeyIdentifier": {
        "value": "subjectKeyIdentifier"
      },
      "authorityKeyIdentifier": {
        "value": [
          {
            "keyid": "authorityKeyIdentifier"
          }
        ]
      },
      "basicConstraints": {
        "value": [
          {
            "CA": true
          }
        ]
      }
    }
  },
  "sha1fingerprint": "sha1fingerprint",
  "services": [
    "ldap",
    "cloud"
  ],
  "href": "/api/setting/v2/certificates/trusted/trusted-cert1-uuid"
}

```

証明書 を PEM 形式で返す

証明書を PEM 形式で返すには、次のいずれかの値を Accept ヘッダーに指定します。

```

application/pkix-cert
application/x-x509-ca-cert
application/x-x509-user-cert
application/x-pem-file

```

リクエストの例:

```
GET /api/setting/v2/certificates/system/system-cert1-uuid HTTP/1.1
Host: alice.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/x-pem-file
```

結果例:

```
HTTP/1.1 200 OK
Date: Thu, 13 May 2021 06:29:33 GMT
Content-Length: 1440
Content-Type: application/x-pem-file; charset=utf-8
X-Zfssa-Setting-API: 2.0
X-Zfssa-API-Version: 2.0

-----BEGIN CERTIFICATE-----
MIID+TCCAUGgAwIBAgIIXKTieQAAAAIwDQYJKoZIhvcNAQELBQAwVjEgMB4GA1UE
...
sUSSZgilvMJ4G8jtx6JSbG4DzDkvo8vq7GSika7h+hi5cbDiZds0L9kDtBIqSAVN
Z1gjaFpzgio6wRvaIA==
-----END CERTIFICATE-----
```

サーバー証明書の作成

サーバー証明書を作成する最初のステップは、証明書署名リクエスト (CSR) を作成することです。CSR をアプライアンスでポストし、CA に送信します。署名付き証明書を CA から受信したら、[45 ページの「鍵または証明書のアップロード」](#)で説明されているように、その署名付き証明書をアップロードします。署名付き証明書がリクエストに置き換わります。

リクエストテンプレートを返す

`template` コマンドは、最低限必要なプロパティのデフォルト値も含めて、CSR のフレームワークを返します。

リクエストの例:

```
GET /api/setting/v2/certificates/system/template HTTP/1.1
Host: alice.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Date: Thu, 13 May 2021 08:03:03 GMT
Content-Length: 261
Content-Type: application/json; charset=utf-8
X-Zfssa-Setting-API: 2.0
X-Zfssa-API-Version: 2.0
```

```
{
  "request": {
    "type": "request",
    "data": {
      "subject": [
        {
          "commonName": "alice.example.com"
        }
      ],
      "extensions": {
        "subjectAltName": {
          "value": [
            {
              "IP": "alice.example.com-ipaddr"
            },
            {
              "DNS": "alice.example.com"
            }
          ]
        }
      ]
    }
  },
  "href": "/api/setting/v2/certificates/system/template"
}
```

リクエストの取り込みおよびアップロード

この `template` 出力を使用する場合、`data` 要素のみを含めます。

CSR に指定する必要がある追加のプロパティについては、[38 ページの「証明書の一覧表示」](#)に示すように、既存のシステム証明書のプロパティを一覧表示します。

CSR が完成したら、次の例に示すように、CSR をホストにアップロードします。CSR は、一度アップロードすると変更できません。

リクエストの例:

```
POST /api/setting/v2/certificates/system HTTP/1.1
Host: alice.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-type: application/json
```

```
{
  "data": {
    "subject": [
      {
        "commonName": "alice.example.com"
      },
      {
        "organizationName": "Example Corp, Inc"
      },
      {
        "localityName": "Exampleron"
      }
    ]
  }
}
```

```

        "stateOrProvinceName": "CA"
      },
      {
        "countryName": "US"
      }
    ],
    "extensions": {
      "subjectAltName": {
        "value": [
          {
            "DNS": "alice.example.com"
          },
          {
            "IP": "alice.example.com-ipaddr"
          }
        ]
      }
    }
  }
}

```

結果例:

```

HTTP/1.1 201 Created
Date: Fri, 14 May 2021 01:17:45 GMT
Content-Type: application/json; charset=utf-8
X-Zfssa-API-Version: 2.0
X-Zfssa-Setting-API: 2.0
Location: /api/setting/v2/certificates/system/65119889-98d3-4fc4-bff5-f007a55f6cb3
Content-Length: 379

```

```

{
  "request": {
    "uuid": "csr-uuid",
    "type": "request",
    "data": {
      "subject": [
        {
          "commonName": "alice.example.com"
        },
        {
          "organizationName": "Example Corp, Inc"
        },
        {
          "localityName": "Exampleron"
        },
        {
          "stateOrProvinceName": "CA"
        },
        {
          "countryName": "US"
        }
      ],
      "extensions": {
        "subjectAltName": {
          "value": [
            {
              "DNS": "alice.example.com"
            },
            {
              "IP": "alice.example.com-ipaddr"
            }
          ]
        }
      }
    }
  }
}

```

```

    },
    "href": "/api/setting/v2/certificates/system/csr-uuid"
  }
}

```

リクエストの CA への転送

アップロードした CSR には、プロパティーの表示や PEM 形式でのリクエストの取得に使用できる UUID があります。

CSR を PEM 形式で返すには、次のいずれかの値を Accept ヘッダーに指定します。

```

application/pkcs10
application/x-pem-file

```

リクエストの例:

```

GET /api/setting/v2/certificates/system/csr-uuid HTTP/1.1
Host: alice.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/x-pem-file

```

結果例:

```

HTTP/1.1 200 OK
Date: Fri, 14 May 2021 03:47:21 GMT
Content-Type: application/x-pem-file; charset=utf-8
X-Zfssa-API-Version: 2.0
X-Zfssa-Setting-API: 2.0
Content-Length: 997

-----BEGIN CERTIFICATE REQUEST-----
MIICpjCCAY4CAQAwJDEiMCAgA1UEAwZyXJkb2NoLWt6LTludWsub3JhY2x1LmNv
...
Bc0Q9FVRVv89AkmeAlF7727aIqmgmFcIUIIrEKTG4PSacedaoBsbjpvrizCuMhyo
vgUk0PE/0xLAfw==
-----END CERTIFICATE REQUEST-----

```

前述の方法で、CSR を CA に転送します。署名付き証明書を CA から受信したら、[45 ページの「鍵または証明書のアップロード」](#)に示すように、その署名付き証明書をアップロードします。

鍵または証明書のアップロード

署名付き証明書を CA から受信したら、次のコマンドを使用して証明書をアップロードします。次のいずれかの値を Content-Type ヘッダーに指定します:

```

application/pkix-cert
application/x-x509-ca-cert
application/x-x509-user-cert
application/x-pem-file

```

リクエストの例:

```
POST /api/setting/v2/certificates/system HTTP/1.1
Host: alice.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/x-pem-file

-----BEGIN CERTIFICATE-----
MIIDdTCCAl2gAwIBAgICH5cWdQYJKoZIhvcNAQELBQAwazELMAkGA1UEBhMCdXMx
...
cgfvd1NUEvSdlb2+cjRbD9uSdtLfv7H5BKTkEd0Xikv9+f150MytMEo4ABt0pEyp
/KwtRsdIxzmzAjmNqfQPR6eAHVfQ/CGwh6Q==
-----END CERTIFICATE-----
```

結果例:

```
HTTP/1.1 200 OK
Date: Tue, 11 May 2021 18:04:22 GMT
Content-Type: application/json; charset=utf-8
X-Zfssa-API-Version: 2.0
X-Zfssa-Setting-API: 2.0
Location: /api/setting/v2/certificates/system/system-cert3-uuid
Content-Length: 987

{
  "certificate": {
    "uuid": "system-cert3-uuid",
    "type": "cert",
    "data": {
      "subject": [
        {
          "countryName": "US"
        },
        {
          "stateOrProvinceName": "CA"
        },
        {
          "localityName": "Exampleton"
        },
        {
          "organizationName": "Example Corp, Inc"
        },
        {
          "commonName": "alice.example.com"
        }
      ],
      "issuer": [
        {
          "countryName": "US"
        },
        {
          "stateOrProvinceName": "AK"
        },
        {
          "localityName": "Trustville"
        },
        {
          "organizationName": "Totally Trustworthy Certificates, Inc"
        },
        {
          "commonName": "Most Trusted Certificate"
        }
      ],
      "serialNumber": "64",
```

```

    "validity": {
      "notBefore": "20210520T21:08:27",
      "notAfter": "20220520T21:08:27"
    },
    "extensions": {
      "basicConstraints": {
        "value": [
          {
            "CA": false
          }
        ]
      },
      "subjectKeyIdentifier": {
        "value": "subjectKeyIdentifier"
      },
      "authorityKeyIdentifier": {
        "value": [
          {
            "keyid": "authorityKeyIdentifier"
          }
        ]
      },
      "subjectAltName": {
        "value": [
          {
            "DNS": "alice.example.com"
          },
          {
            "IP": "alice.example.com-ipaddr"
          }
        ]
      }
    }
  },
  "sha1fingerprint": "sha1fingerprint",
  "href": "/api/setting/v2/certificates/system/system-cert3-uuid"
}

```

証明書を信頼する必要があるサービスの指定

システム証明書または CSR のプロパティは変更できません。CSR をポストする前に、CSR のプロパティを設定します。

信頼できる証明書の `services` プロパティの値は設定できます。`services` プロパティは、証明書を信頼する必要があるサービスのリストです。

次の例では、信頼できる証明書の `services` プロパティを設定します。

```

PUT /api/setting/v2/certificates/trusted/trusted-cert2-uuid HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: alice.example.com:215
Content-Type: application/json

```

```

{"certificate": { "services": ["ldap"] }}

```

次の例では、証明書を信頼する必要がある複数のサービスを設定します。

```
PUT /api/setting/v2/certificates/trusted/trusted-cert2-uuid HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: alice.example.com:215
Content-Type: application/json

{"certificate": {"services": [ "ldap", "cloud" ] }}
```

システムのデフォルト証明書の設定

次の例では、デフォルトのシステム証明書を割り当てます。

```
PUT /api/setting/v2/certificates/system HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: alice.example.com:215
Accept: application/json
Content-Type: application/json

{ "default": "system-cert1-uuid" }
```

証明書の破棄

DELETE コマンドは、指定された証明書、リクエストまたは鍵を破棄します。成功すると、HTTP ステータス 204 (No Content) が返されます。

```
DELETE /api/setting/v2/certificates/system/system-cert2-uuid HTTP/1.1
```

HTTP Strict Transport Security の有効化

HTTP Strict Transport Security (HSTS) では、指定した期間にわたりセキュアな HTTPS 接続のみが許可され、HTTP 接続は許可されません。HSTS を使用する前に、HSTS の前提条件をよく確認し、HSTS が有効になっているブラウザの動作について理解し、認証局によって署名された証明書をインストールしてください。

注記 - 証明書を有効で適切な状態に維持しないと、HSTS のセキュリティ上の利点を無駄にしたり、ブラウザがアプライアンスに接続できなくなったりすることがあります。

次の例に示すように、HSTS が有効のままになるデフォルトの最長時間は 63072000 秒 (2 年) です。

リクエストの例:

```
GET /api/setting/v2/security HTTP/1.1
Host: alice.example.com:215
```

```
Authorization: Basic Tm8gcGVla2luZyE=  
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK  
Date: Fri, 14 May 2021 19:22:27 GMT  
Content-Type: application/json; charset=utf-8  
X-Zfssa-API-Version: 2.0  
X-Zfssa-Setting-API: 2.0  
Content-Length: 109
```

```
{  
  "Security settings": {  
    "href": "/api/setting/v2/security",  
    "hsts_enable": false,  
    "hsts_max_age": 63072000  
  }  
}
```

このアプライアンスに対して HSTS を有効にするには、`hsts_enable` プロパティを `true` に設定します。

```
PUT /api/setting/v2/security HTTP/1.1  
Host: alice.example.com:215  
Content-Type: application/json
```

```
{"hsts_enable": true}
```


RESTful API 警告サービス

ハードウェアおよびソフトウェアの障害などの重要なアプライアンスイベントでは、アラートがトリガーされます。アラートはログに表示され、電子メールの送信やデータセットの再開など、他のアラートアクションを実行するように構成することもできます。

RESTful API 警告サービスを使用すると、カスタムアラートアクション (イベントアラートへの応答) およびカスタム分析統計しきい値アラートを作成できます。

警告サービスコマンド

次の表に、警告サービスコマンドを示します。1つのアラートイベントに対して複数のアラートアクション (応答) を定義できます。たとえば、1つのアラートに回答して、2つの異なるグループに電子メールを送信し、syslog メッセージを書き込み、データセットを再開することができます。actions-### オブジェクトは、アラートイベントとアラートに対するすべての応答のセットです。action-### オブジェクトは、応答の1つです。

表 7 警告サービスコマンド

リクエスト	パス /api/alert/v{1 2} の後ろに追加	説明
GET	/api/alert/v{1 2} のみを使用	警告サービスコマンドを一覧表示します
GET	/actions	すべてのアラートアクションオブジェクトを一覧表示します
POST	/actions	新しいアラートアクションを作成します
GET	/actions/actions-###	指定されたアラートアクションプロパティを一覧表示します
PUT	/actions/actions-###	指定されたアラートアクションオブジェクトを変更します
DELETE	/actions/actions-###	指定されたアクションオブジェクトを破棄します
POST	/actions/actions-###	新しいアラートアクションのアクションを作成します
GET	/actions/actions-###/action-###	指定されたアラートアクションのアクションプロパティを一覧表示します

リクエスト	パス <code>/api/alert/v{1 2}</code> の後ろに追加	説明
PUT	<code>/actions/actions-###/action-###</code>	指定されたアラートアクションのアクションオブジェクトを変更します
POST	<code>/postalert</code>	カスタムアラートを送信します
DELETE	<code>/actions/actions-###/action-###</code>	指定されたアラートアクションのアクションオブジェクトを破棄します
GET	<code>/thresholds</code>	すべてのしきい値アラートオブジェクトを一覧表示します
POST	<code>/thresholds</code>	新しいしきい値アラートを作成します
GET	<code>/thresholds/threshold-alert-uuid</code>	指定されたしきい値アラートプロパティを一覧表示します
PUT	<code>/thresholds/threshold-alert-uuid</code>	指定されたしきい値アラートオブジェクトを変更します
DELETE	<code>/thresholds/threshold-alert-uuid</code>	指定されたしきい値アラートオブジェクトを破棄します
GET	<code>/events</code>	新しい警告イベントを待機します

アラートアクション

アラートアクションは、イベントアラートに対する応答です。アラートアクションを作成するには、1つ以上のイベントを指定し、電子メールの送信やワークフローの実行など、そのイベントに対してアラートが送信されたときに実行する1つ以上のアクションを指定します。特定のイベントアラートに対して複数のアラートアクションを指定できます。

category プロパティは、アラートアクションが実行されるイベントのカテゴリを指定します。各カテゴリには、1つ以上のイベントが含まれます。アラートアクションを作成または一覧表示すると、各カテゴリ内のイベントが一覧表示されます。デフォルトでは、アラートアクションはカテゴリ内のすべてのイベントに対して実行されます。アラートアクションがイベントのサブセットに対してのみ実行される場合、このアラートアクションを実行することのないイベントを表すプロパティの値を **false** に変更してください。

次の表では、指定できるイベントカテゴリについて説明します。

表 8 アラートアクションイベントカテゴリ

カテゴリ	説明
ad	Active Directory または SMB Kerberos クライアント認証が低下しました。
all	すべてのアラートまたは不具合、サービスアラート、ハードウェア障害などの高レベルイベント。
analytics	データセットの自動一時停止警告、メモリ合計超過、使用量超過などの高レベルイベント。特定の分析統計のイベントについては、 61 ページの「しきい値アラート」 を参照してください。

カテゴリ	説明
appliance_software	ソフトウェア更新を妨げたり、カーネルパニックを引き起こしたりするイベント。
cloud_snapshot	
cluster	クラスタイベント (リンクの障害やピアのエラーなど)
custom	ワークフローで指定されたカスタムイベントのアラートアクション。59 ページの「カスタムアラート」を参照してください。
hardware	アプライアンスのブートおよびハードウェア構成の変更
hardware_faults	任意のハードウェア障害。
ndmp backup restore	NDMP の TAR または DUMP のバックアップおよび復元の開始イベントと終了イベント。
network	ネットワークポート、データリンク、および IP インタフェースのイベントと障害
scrk	バンドルのアップロードイベントをサポートします
replication replication_source replication_target	イベントおよび障害を送受信します。
smf	ソフトウェアサービス障害のイベント
shadow	移行エラーまたは移行完了。
thresholds	61 ページの「しきい値アラート」で説明されているように、既存のしきい値イベントアラートにアクションを追加できるようにします。
zfs_pool	ストレージプールのイベント (スクラブやホットスペースの起動など)

handler プロパティは、指定されたイベントが発生したときに実行するアクションのタイプを指定します。次の表に示すように、handler のほとんどの値には、追加のプロパティを設定する必要があります。

表 9 アラートアクションの応答タイプ

応答タイプ (handler)	ハンドラのプロパティ	応答タイプの説明
email	address	指定した件名で指定した受信者に電子メールを送信します。
	subject	受信者は 1 人のみ入力します。受信者のリストを入力しないでください。複数の個別受信者に送信するには、このイベントアラートに追加アラートアクションを作成します。 SMTP サービスを使用して電子メールの送信方法を構成します。153 ページの「サービスの一覧表示」を参照してください。
snmp_trap	なし	アラートの詳細を含む SNMP トラップを送信します。

応答タイプ (handler)	ハンドラのプロパティ	応答タイプの説明
		SNMP サービスを使用して SNMP トラップ宛先を構成します。153 ページの「サービスの一覧表示」を参照してください。
syslog	なし	アラートの詳細を含むシステムメッセージを 1 つ以上のリモートシステムに送信します。 syslog サービスを使用して syslog 宛先を構成します。153 ページの「サービスの一覧表示」を参照してください。 syslog メッセージの送信の詳細は、『Oracle ZFS Storage Appliance 管理ガイド、Release OS8.8.x』の「syslog 構成」を参照してください。
resume_dataset	dataset	分析データセットを再開します。データセットの再開および一時停止は、断続的なパフォーマンス問題を診断する場合や、データセットを有効にしたまま保持することが適切ではない場合などに役立ちます。 詳細は、『Oracle ZFS Storage Appliance Analytics ガイド、Release OS8.8.x』の「分析データセットについて」を参照してください。
suspend_dataset	dataset	分析データセットを一時停止します。
resume_worksheet	worksheet	分析ワークシートを再開します。ワークシートの再開および一時停止は、データセットの再開および一時停止と同じ理由で役立ちます。ワークシートには大量のデータセットが含まれることがあります。 詳細は、『Oracle ZFS Storage Appliance Analytics ガイド、Release OS8.8.x』の「ワークシートのグラフとプロット」を参照してください。
suspend_worksheet	worksheet	分析ワークシートを一時停止します。
execute_workflow	workflow	指定したワークフローを実行します。アラートアクションの対象となるワークフローの詳細は、『Oracle ZFS Storage Appliance 管理ガイド、Release OS8.8.x』の「警告アクションでのワークフローの使用」を参照してください。

すべてのアラートアクションの一覧表示

すべてのアラートアクションを一覧表示すると、各アラートアクションについて、イベントカテゴリとそのカテゴリ内の各イベントのみが一覧表示されます。アラートアクションに対して定義されている応答も表示するには、55 ページの「単一アラートアクションの一覧表示」を参照してください。

警告アクションを取得するリクエストの例:

```
GET /api/alert/v1/actions HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
```

Accept: application/json

結果例:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 1395

{
  "actions": [
    {
      "category": "smf",
      "uuid": "actions-uuid1",
      "failed_services": true,
      "degraded_services": true,
      "repaired_services": false,
      "actions": "actions-000",
      "href": "/api/alert/v2/actions/actions-000"
    },
    {
      "category": "analytics",
      "uuid": "actions-uuid2",
      "analytics_datasets_auto-suspend_notify": false,
      "analytics_datasets_auto-suspend_warning": false,
      "analytics_memory_total_exceeded": true,
      "analytics_memory_total_normal": false,
      "analytics_usage_exceeded": true,
      "analytics_usage_normal": false,
      "actions": "actions-001",
      "href": "/api/alert/v2/actions/actions-001"
    }
  ]
}
```

単一アラートアクションの一覧表示

一覧表示する特定のアラートアクションを指定すると、そのアラートアクションについて、イベントカテゴリ、そのカテゴリ内の各イベント、および各応答やアクションが一覧表示されます。

次のアラートアクションには3つの応答があり、true イベントの1つが発生したときにすべて実行されます。

リクエストの例:

```
GET /api/alert/v1/actions/actions-000 HTTP/1.1
```

結果例:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 331

{
  "actions": {
```

```

    "href": "/api/alert/v2/actions/actions-000",
    "category": "smf",
    "uuid": "actions-uuid1",
    "failed_services": true,
    "degraded_services": true,
    "repaired_services": false,
    "action-000": {
      "handler": "email",
      "address": "admin@example.com",
      "subject": "failed or degraded service",
      "href": "/api/alert/v2/actions/actions-000/action-000"
    },
    "action-001": {
      "handler": "email",
      "address": "it-team@example.com",
      "subject": "failed or degraded service",
      "href": "/api/alert/v2/actions/actions-000/action-001"
    },
    "action-002": {
      "handler": "syslog",
      "href": "/api/alert/v2/actions/actions-000/action-002"
    }
  }
}

```

アラートアクションの作成

アラートアクションを作成する際には、このカスタム応答を定義するイベントのカテゴリである `category` プロパティの値を指定する必要があります。`category` の値のリストについては、[表8](#)を参照してください。

リクエストの例:

```

POST /api/alert/v1/actions HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-Session: uerqghq84vbdv
Content-Type: application/json
Content-Length: 30

```

```

{"category": "smf"}

```

結果例:

結果には、指定されたイベントカテゴリのすべてのイベントが一覧表示されます。デフォルトでは、カテゴリ内のすべてのイベントによって応答アクションが実行されます (これらはすべて `true` に設定されています)。

```

HTTP/1.1 201 Created
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 118
Location: /api/alert/v1/actions/actions-000

{
  "actions": {
    "href": "/api/alert/v2/actions/actions-000",
    "category": "smf",

```

```

        "uuid": "actions-uuid",
        "failed_services": true,
        "degraded_services": true,
        "repaired_services": true
    }
}

```

アラートアクションの変更

指定されたイベントカテゴリの一部のイベントで応答アクションが実行されないようにする場合は、それらのイベントを表すプロパティを `false` に設定します。

次の例では、修復されたサービスに対して、障害が発生したサービスや劣化したサービスとは異なる応答を定義できます。

リクエストの例:

```

PUT /api/alert/v1/actions/actions-000 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-Session: uerqghq84vbdv
Content-Type: application/json
Content-Length: 30

```

```

{"repaired_services": false}

```

結果例:

```

HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 195

{
  "actions": {
    "href": "/api/alert/v2/actions/actions-000",
    "category": "smf",
    "uuid": "actions-uuid",
    "failed_services": true,
    "degraded_services": true,
    "repaired_services": false
  }
}

```

イベントの応答の指定

デフォルトでは、イベントアラートはアラートログに記録されます。前述の例では、アラートログへの送信に加えて、応答を定義するイベントを指定しました。これらのイベントへの応答を定義するには、特定のアラートアクションの `handler` プロパティの値を指定します。`handler` の値のリストについては、[表9](#)を参照してください。

リクエストの例:

この例では、actions-000 アラートの admin に電子メールを送信するアラートアクションを作成します。

```
POST /api/alert/v1/actions/actions-000 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-Session: uerqghq84vbdv
Content-Type: application/json
Content-Length: 68

{"handler": "email", "address": "admin@example.com", "subject": "failed or degraded service"}
```

結果例:

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 177
Location: /api/alert/v1/actions/actions-000/action-000

{
  "action": {
    "href": "/api/alert/v1/actions/actions-002/action-000",
    "handler": "email",
    "address": "admin@example.com",
    "subject": "failed or degraded service"
  }
}
```

同じイベントに対して追加の応答を指定するには、同じアラートアクションに対して POST リクエストを再度発行し、別のハンドラを指定するか、同じハンドラとそのハンドラの別の引数を指定します。1つのリクエストで複数の handler プロパティを指定した場合、最後のプロパティを除くすべての handler プロパティは無視されます。

次の例は短縮されています。これらのリクエストは、[55 ページの「単一アラートアクションの一覧表示」](#)に示すように、/api/alert/v1/actions/actions-000/action-001 および /api/alert/v1/actions/actions-000/action-002 を作成します。

```
POST /api/alert/v1/actions/actions-002 HTTP/1.1
...
{"handler": "email", "address": "it-team@example.com", "subject": "failed or degraded service"}

POST /api/alert/v1/actions/actions-002 HTTP/1.1
...
{"handler": "syslog"}
```

イベントの応答の変更

応答を変更するには、変更する応答の HREF を指定します。

```
PUT /api/alert/v1/actions/actions-000/action-001 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-Session: uerqghq84vbdv
```

```
Content-Type: application/json
Content-Length: 28

{"address": "it-group@example.com"}
```

イベントの応答の削除

応答を削除するには、削除する応答の HREF を指定します。

```
DELETE /api/alert/v1/actions/actions-000/action-000 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-Session: uerqghq84vbdv

HTTP/1.1 204 No Content
```

アラートアクションの削除

アラートアクションを削除するには、削除するアラートアクションの HREF を指定します。

リクエストの例:

```
DELETE /api/alert/v1/actions/actions-003 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
```

結果例:

```
HTTP/1.1 204 No Content
X-Zfssa-Appliance-API: 1.0
```

カスタムアラート

カスタムアラートとは、カスタムイベントに対するアラートアクションのことです。category の値に custom を指定します。

カスタムイベントはワークフローで定義されます。事前定義されたイベントから選択するのではなく、ワークフローで定義されたイベントを説明する次のプロパティを指定します。

表 10 カスタムアラートのプロパティ

プロパティ	型	説明
severity	String	オプション。警告の原因となったイベントの重要度。有効な値は、Minor、Major、または Critical です。
description	String	必須。アラートの原因となったイベントの説明。

プロパティ	型	説明
response	String	オプション。このイベントの影響を軽減するためにシステムが実行するアクションの説明。
impact	String	オプション。このイベントのアプライアンスに対する影響の説明。
recommended_action	String	オプション。このイベントの影響を軽減するために管理者が実行するアクションの説明。

カスタムアラートを作成および送信するための承認の構成

カスタムアラートを作成するには、ユーザーは、alert スコープの `allow_configure` 承認が必要です。カスタムアラートを送信するには、ユーザーは、alert スコープの `allow_post` 承認が必要です。『[Oracle ZFS Storage Appliance 管理ガイド、Release OS8.8.x](#)』の「[カスタムアラートを使用するために必要な承認](#)」を参照してください。

カスタムアラートの作成

`category` の値に `custom` を指定し、[表10](#)から少なくとも `description` プロパティの値を設定します。

リクエストの例:

```
POST /api/alert/v1/actions
Accept: application/json
Content-Type: application/json
```

```
{"category": "custom", "severity": "Minor", "description": "Custom alert description",
"response": "What the system will do", "impact": "What happened to the appliance",
"recommended_action": "What the administrator should do"}
```

結果例:

出力のアラートの `uuid` を書き留めておきます。これはアラートを送信する際に必要になります。

```
{
  "actions": {
    "href": "/api/alert/v1/actions/actions-004",
    "category": "custom",
    "severity": "Minor",
    "description": "Custom alert description",
    "response": "What the system will do",
    "impact": "What happened to the appliance",
    "recommended_action": "What the administrator should do",
    "uuid": "custom-alert-uuid"
  }
}
```

57 ページの「アラートアクションの変更」で説明されているように、他のアラートを変更するのと同じ方法で、このカスタムアラートを変更し、severity、description、response、impact、または recommended_action プロパティの値を追加または変更します。

57 ページの「イベントの応答の指定」および58 ページの「イベントの応答の変更」で説明されているように、他のアラートへの応答を指定するのと同じ方法で、このカスタムアラートへの応答を指定します。

59 ページの「イベントの応答の削除」および59 ページの「アラートアクションの削除」で説明されているように、他のアラートと同じ方法で、アラートの応答またはカスタムアラートを削除します。

カスタムアラートの送信

送信するカスタムアラートの UUID を指定する必要があります。UUID は、アラートを作成するとき、およびアラートを一覧表示するときに表示されます。60 ページの「カスタムアラートの作成」および55 ページの「単一アラートアクションの一覧表示」を参照してください。

category の値が custom であるアラートのみを送信できます。

アラートの UUID に加えて、表10に一覧表示されているプロパティのうち、アラートで指定されていないものを指定する必要があります。アラートで指定されたプロパティには、アラートが送信されたときに新しい値を指定できます。

リクエストの例:

```
POST /api/alert/v1/postalert
Accept: application/json
Content-Type: application/json

{"uuid": "custom-alert-uuid"}
```

結果例:

```
{
  "uuid": "posted-alert-uuid"
}
```

しきい値アラート

しきい値アラートは、しきい値が特定の分析統計に対して定義されているカスタムアラートであり、統計値がそのしきい値の範囲外になるとアラートアクションが実行されます。Oracle ZFS Storage Appliance Analytics ガイド、Release OS8.8.x も参照してください。

次の表では、分析統計の指定、その統計のしきい値の定義、およびそのしきい値アラートに対するアラートアクションの実行時期の定義を行うために設定するプロパティについて説明します。

表 11 しきい値アラートのプロパティ

プロパティ	型	説明
statname	AnalyticsStatistics	必須。モニターする分析統計。
limit	PositiveInteger	必須。アラートをトリガーするしきい値。1 秒あたりのバイト、操作、アクセス、リクエストの割合または数。
type	ChooseOne	しきい値 (limit) を現在の統計値と比較する方法。 ■ normal – 現在の統計値がしきい値を超えています。これはデフォルトです。 ■ inverted – 現在の統計値がしきい値を下回っています。
minpost	Duration	アラートアクションを実行するまでに統計値がしきい値条件を維持する必要がある秒単位の時間の長さ。デフォルト値は 5 分です。
days	ChooseOne	これらのアラートを送信する日: all の日、weekdays、または weekends。デフォルト値は all です。
window_start window_end	TimeOfDay	このアラートアクションを実行する期間を示す時間ウィンドウ。UTC の 00:00 から 23:30 までの時間を 30 分単位で指定します。条件が満たされたらいつでもこのアラートアクションを実行するには、開始時間または終了時間として none を指定します。デフォルト値は、window_start が none、window_end が 00:00 です。
frequency	Duration	統計値がしきい値条件を維持している間にアラートアクションを再実行する秒単位の時間間隔の長さ。デフォルト値は 5 分です。
minclear	Duration	フォローアップアラートアクションを実行するまでに統計値がしきい値条件の範囲外になる必要がある秒単位の時間の長さ。デフォルト値は 5 分です。

しきい値アラートの一覧表示

次のリクエストは、構成済みのすべてのしきい値アラートを一覧表示します。

リクエストの例:

```
GET /api/alert/v1/thresholds HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Date: Tue, 15 Oct 2019 10:38:40 GMT
X-Zfssa-Appliance-API: 1.0
```

```

Content-Type: application/json
Content-Length: 689

{
  "thresholds": [
    {
      "uuid": "threshold-uuid1",
      "statname": "cpu.utilization",
      "type": "normal",
      "limit": 80,
      "minpost": 300,
      "days": "weekdays",
      "window_start": "08:00",
      "window_end": "19:30",
      "frequency": 300,
      "minclear": 300,
      "threshold": "threshold-000",
      "href": "/api/alert/v1/thresholds/threshold-uuid1"
    },
    {
      "uuid": "threshold-uuid2",
      "statname": "cap.meta_percentused[pool]",
      "type": "normal",
      "limit": 85,
      "minpost": 300,
      "days": "all",
      "window_start": "none",
      "window_end": "00:00",
      "frequency": 0,
      "minclear": 0,
      "threshold": "threshold-001",
      "href": "/api/alert/v1/thresholds/threshold-uuid2"
    }
  ]
}

```

次のリクエストを使用して、指定されたしきい値アラートのみのすべてのプロパティを一覧表示します。

```
GET /api/alert/v1/thresholds/threshold-uuid HTTP/1.1
```

しきい値アラートの作成

この例では、1秒あたりのデータリンクバイト数が100000KBを超えるイベントのしきい値アラートを作成します。他のすべてのプロパティにはデフォルト値があります。

カスタムしきい値アラートを作成するには、ユーザーは、alert スコープの `allow_configure` 承認が必要です。

リクエストの例:

```

POST /api/alert/v1/thresholds HTTP/1.1
Host: zfs-storage.example.com
X-Auth-User: root
X-Auth-Key: password
Content-Type: application/json

```

Content-Length: 50

```
{"statname": "datalink.kilobytes", "limit": 100000}
```

結果例:

HTTP/1.1 201 Created
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 321
Location: /api/alert/v1/thresholds/threshold-uuid

```
{
  "threshold": {
    "href": "/api/alert/v1/thresholds/threshold-uuid",
    "uuid": "threshold-uuid",
    "statname": "datalink.kilobytes",
    "type": "normal",
    "limit": 100000,
    "minpost": 300,
    "days": "all",
    "window_start": "none",
    "window_end": "00:00",
    "frequency": 300,
    "minclear": 300
  }
}
```

しきい値アラートの変更

このコマンドを使用して、指定されたしきい値アラートのプロパティを変更します。

リクエストの例:

PUT /api/alert/v1/thresholds/threshold-uuid HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215

```
{"days": "weekdays", "minpost": 120}
```

結果例:

HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 326

```
{
  "threshold": {
    "href": "/api/alert/v1/thresholds/threshold-uuid",
    "uuid": "threshold-uuid",
    "statname": "datalink.kilobytes",
    "type": "normal",
    "limit": 100000,
    "minpost": 120,
    "days": "weekdays",
    "window_start": "none",
    "window_end": "00:00",
  }
}
```

```
    "frequency": 300,  
    "minclear": 300  
  }  
}
```

しきい値アラートの削除

指定されたしきい値アラートを削除します

リクエストの例:

```
DELETE /api/alert/v1/thresholds/threshold-uuid HTTP/1.1  
Authorization: Basic Tm8gcGVla2luZyE=  
Host: zfs-storage.example.com:215
```

結果例:

```
HTTP/1.1 204 No Content  
X-Zfssa-Appliance-API: 1.0
```


Analytics サービス

Analytics を使用すると、さまざまな統計をリアルタイムでグラフ化したり、あとで検索するためにデータを記録したりできます。長期のモニタリングおよび短期の分析のどちらを行うこともできます。Analytics では、DTrace を使用してカスタム統計を動的に作成し、オペレーティングシステムスタックの異なる層を詳細に分析できます。

Analytics コマンド

`http://hostname/api/analytics/v{1|2}` から次の Analytics サービスを入手できます。

表 12 Analytics コマンド

リクエスト	パス <code>/analytics/v{1 2}</code> の後ろに追加	説明
GET	<code>/analytics/v{1 2}</code> のみを使用	分析サービス情報を一覧表示します
POST	<code>/worksheets</code>	新しい分析データセットを作成します
GET	<code>/worksheets/worksheet</code>	指定された分析データセットプロパティを取得します
GET	<code>/worksheets</code>	すべての分析データセットオブジェクトを一覧表示します
PUT	<code>/worksheets/worksheet</code>	指定された分析データセットオブジェクトを変更します
DELETE	<code>/worksheets/worksheet</code>	指定されたワークシートオブジェクトを破棄します
PUT	<code>/worksheets/worksheet/suspend</code>	すべてのワークシートデータセットを一時停止します
PUT	<code>/worksheets/worksheet/resume</code>	すべてのワークシートデータセットを再開します
POST	<code>/worksheets/worksheet/datasets</code>	新しいワークシートデータセットを作成します
GET	<code>/worksheets/worksheet/datasets/dataset</code>	指定されたワークシートデータセットプロパティを取得します
GET	<code>/worksheets/worksheet/datasets</code>	すべてのワークシートデータセットオブジェクトを一覧表示します

リクエスト	パス <code>/analytics/v{1 2}</code> の後ろに追加	説明
PUT	<code>/worksheets/worksheet/datasets/dataset</code>	指定されたワークシートデータセットオブジェクトを変更します
DELETE	<code>/worksheets/worksheet/datasets/dataset</code>	指定されたデータセットオブジェクトを破棄します
POST	<code>/datasets</code>	新しい分析データセットを作成します
GET	<code>/datasets/dataset</code>	指定された分析データセットプロパティを取得します
GET	<code>/datasets</code>	すべての分析データセットオブジェクトを一覧表示します
PUT	<code>/datasets/dataset</code>	指定された分析データセットオブジェクトを変更します
DELETE	<code>/datasets/dataset</code>	指定されたデータセットオブジェクトを破棄します
PUT	<code>/datasets</code>	すべてのデータセットを一時停止または再開します
PUT	<code>/datasets/dataset/data</code>	このデータセットを保存します (未保存の場合)
DELETE	<code>/datasets/dataset/data</code>	指定された [粒度] でこのデータセットからデータを削除します
GET	<code>/settings</code>	分析設定を一覧表示します
PUT	<code>/settings</code>	分析設定を変更します

Analytics の設定

次の表で説明しているプロパティを使用すると、すべての分析データを収集したり、データを保持する時間数を設定したり、ホスト名検索ポリシーを設定したりできます。

プロパティ	説明
<code>retain_second_data</code>	秒あたりのデータの保持間隔 (時間)
<code>retain_minute_data</code>	分あたりのデータの保持間隔 (時間)
<code>retain_hour_data</code>	時間あたりのデータの保持間隔 (時間)
<code>hostname_lookup</code>	ホスト名検索ポリシー

設定の取得

このコマンドは、分析プロパティの現在の値を取得します。

リクエストの例:

```
GET /api/analytics/v1/settings HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 131
X-Zfssa-Analytics-API: 1.0

{
  "settings": {
    "href": "/api/analytics/v1/settings",
    "retain_hour_data": 600,
    "retain_minute_data": 400,
    "retain_second_data": 200,
    "hostname_lookup": true
  }
}
```

設定の変更

設定の変更コマンドは、データ保持値やホスト名検索ポリシーなどの分析設定を変更するために使用します。

リクエストの例:

```
PUT /api/analytics/v1/settings HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Content-Type: application/json
Content-Length: 60

{"retain_hour_data":600, "retain_minute_data":400, "retain_second_data":200,
 "hostname_lookup":true}
```

結果例:

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 101
X-Zfssa-Analytics-API: 1.0

{
  "settings": {
    "href": "/api/analytics/v1/settings",
    "retain_hour_data": 600,
    "retain_minute_data": 400,
    "retain_second_data": 200,
    "hostname_lookup": true
  }
}
```

Analytics のワークシート

ワークシートは統計がグラフ化される BUI 画面です。複数の統計を同時にプロットでき、ワークシートにタイトルを付けてあとで表示するために保存できます。ワークシートを保存する操作によって、オープン中のすべての統計のアーカイブアクションが自動的に実行されます。つまり、統計が開いていれば、統計の読み取りおよびアーカイブが際限なく続けられます。ワークシートコマンドは、BUI から使用可能なワークシートを管理するために使用できます。

次の表に、分析ワークシートで使用するプロパティを示します。

プロパティ	説明
ctime	このワークシートが作成された日時
mtime	このワークシートが最後に変更された日時
name	このワークシートの名前
owner	このワークシートの所有者
uuid	このワークシートの汎用一意識別子

ワークシートの一覧表示

現在構成済みのすべての分析ワークシートを一覧表示します。

リクエストの例:

```
GET /api/analytics/v1/worksheets HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 237
X-Zfssa-Analytics-API: 1.0

{
  "worksheets": [{
    "href": "/api/analytics/v1/worksheets/ab59bcbc...",
    "uuid": "ab59bcbc-080a-cf1a-98c9-9f485bc3a43d"
  }, {
    "href": "/api/analytics/v1/worksheets/bb3ee729...",
    "uuid": "bb3ee729-080a-cf1a-98c9-9f485bc3a43d"
  }]
}
```

分析ワークシートの取得

単一の分析ワークシートを取得します。

リクエストの例:

```
GET /api/analytics/v1/worksheets/ab59bcbc-080a-cf1a-98c9-9f485bc3a43d
HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 237
X-Zfssa-Analytics-API: 1.0
```

```
{
  "worksheet": {
    "ctime": "Thu Jun 13 2013 02:17:14 GMT+0000 (UTC)",
    "href": "/api/analytics/v1/worksheets/ab59bcbc-080a-cf1a-98c9-9f485bc3a43d",
    "mtime": "Sun Jun 23 2013 16:22:01 GMT+0000 (UTC)",
    "name": "myworksheet",
    "owner": "root",
    "uuid": "ab59bcbc-080a-cf1a-98c9-9f485bc3a43d"
  }
}
```

ワークシートの作成

新しい分析ワークシートを作成します。

リクエストの例:

```
POST /api/analytics/v1/worksheets HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Content-Length: 26
```

```
{"name": "myworksheet"}
```

結果例:

```
HTTP/1.1 201 Created
Content-Length: 280
Location: /api/analytics/v1/worksheets/bb3ee729-4480-4609-89b2-fae2dc016bec
```

```
{
  "worksheet": {
    "uuid": "bb3ee729-4480-4609-89b2-fae2dc016bec",
    "name": "myworksheet",
    "owner": "root",
    "ctime": "Fri Aug 23 2013 20:35:00 GMT+0000 (UTC)",

```

```
      "mtime": "Fri Aug 23 2013 20:35:00 GMT+0000 (UTC)",
      "href": "/api/analytics/v1/worksheets
              /bb3ee729-4480-4609-89b2-fae2dc016bec"
    }
  }
```

ワークシート名の変更

保存したワークシートの名前を変更します。

リクエストの例:

```
PUT /api/analytics/v1/worksheets/a442e761-4048-4738-b95f-be0824d7ed09
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Content-Length: 26
```

```
{"name": "test"}
```

結果例:

```
HTTP/1.1 202 Accepted
Date: Tue, 20 Dec 2016 00:33:06 GMT
Server: TwistedWeb/192.0.2
Content-Length: 279
X-Zfssa-Analytics-API: 1.1
X-Zfssa-API-Version: 1.0
Content-Type: application/json; charset=utf-8

{
  "worksheet": {
    "href": "/api/analytics/v1/worksheets/a442e761-4048-4738-b95f-be0824d7ed09",
    "uuid": "a442e761-4048-4738-b95f-be0824d7ed09",
    "name": "test",
    "owner": "root",
    "ctime": "Wed Dec 14 2016 03:58:28 GMT+0000 (UTC)",
    "mtime": "Tue Dec 20 2016 00:25:57 GMT+0000 (UTC)"
  }
}
```

ワークシートの破棄

分析ワークシートを破棄します。この例では、ワークシート名がワークシート識別子として使用されていますが、href で指定される uuid も使用できます。このコマンドの動作は、ワークシートを破棄する CLI コマンドの動作と一致します。

リクエストの例:

```
DELETE /api/analytics/v1/worksheets/name=myworksheet HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Content-Length: 26
```

結果例:

HTTP/1.1 204 No Content
X-Zfssa-Analytics-API: 1.0

ワークシートデータセットの一覧表示

指定されたワークシート内のすべてのデータセットを一覧表示します。

次の表に、データセット構成で使用するプロパティを示します。

プロパティ	説明
name	このデータセットの基になる統計の名前
drilldowns	現在強調表示されているドリルダウン (ある場合)
seconds	このデータセットに表示されている秒数

リクエストの例:

```
GET /api/analytics/v1/worksheets/name=myworksheet/datasets HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

ワークシートデータセットの追加

ワークシートデータセットを作成します。

リクエストの例:

```
POST /api/analytics/v1/worksheets/name=myworksheet/datasets HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Content-Length: 26
```

```
{"name": "nfs4.ops", "seconds": 300}
```

結果例:

```
HTTP/1.1 201 Created
Content-Type: application/json
X-Zfssa-Analytics-API: 1.0
Location: /api/analytics/v1/worksheets/name=me/datasets/nfs4.ops
Content-Length: 162
```

```
{
  "dataset": {
    "href": "/api/analytics/v1/worksheets/name=me/datasets/dataset-008",
    "name": "nfs4.ops",
    "width": 0,
    "drilldowns": [],
    "seconds": 300,
  }
}
```

```

    "time": ""
  }
}

```

ワークシートデータセットの変更

既存のワークシートデータセットを変更します。

リクエストの例:

```

PUT /api/analytics/v1/worksheets/name=myworksheet/datasets/dataset-008
HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Content-Length: 26

```

```

{"seconds": 60}

```

結果例:

```

HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 161
X-Zfssa-Analytics-API: 1.0

{
  "dataset": {
    "href": "/api/analytics/v1/worksheets/name=me/datasets/dataset-008",
    "name": "nfs4.ops",
    "width": 0,
    "drilldowns": [],
    "seconds": 60,
    "time": ""
  }
}

```

Analytics のデータセット

Analytics のデータセットでは、次のプロパティを使用します。suspended を除くすべてのプロパティは変更できません。

プロパティ	説明
name	このデータセットの基になる統計の名前
grouping	この統計が属するグループ
explanation	基になる統計の説明
incore	コア中のデータセットデータのバイト数
size	ディスク上のデータセットデータのバイト数
suspended	データセットが現在一時停止中かどうかを示すブール値

プロパティ	説明
activity	保留中のデータセットアクティビティフラグ

使用可能なデータセット:

- ad.avglatency
- ad.avglatency[op]
- ad.avglatency[result]
- ad.binds
- ad.binds[hostname]
- ad.binds[result]
- ad.ops
- ad.ops[op]
- ad.ops[result]
- arc.accesses[hit/miss]
- arc.l2_accesses[hit/miss]
- arc.l2_size
- arc.size
- arc.size[component]
- cpu.utilization
- cpu.utilization[mode]
- dnlc.accesses[hit/miss]
- fc.bytes
- fc.ops
- ftp.kilobytes
- http.reqs
- io.bytes
- io.bytes[op]
- io.disks[utilization=95][disk]
- io.ops
- io.ops[disk]
- io.ops[op]
- iscsi.bytes
- iscsi.ops
- metacap.bytesused
- metacap.percentused
- ndmp.diskkb
- nfs2.ops

- nfs2.ops[op]
- nfs3.ops
- nfs3.ops[op]
- nfs4.ops
- nfs4.ops[op]
- nfs4-1.ops
- nfs4-1.bytes
- nic.kilobytes
- nic.kilobytes[device]
- nic.kilobytes[direction]
- sftp.kilobytes
- smb.ops
- smb.ops[op]

データセットの一覧表示

構成済みのすべての分析データセットを一覧表示します。

リクエストの例:

```
GET /api/analytics/v1/datasets HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 237
X-Zfssa-Analytics-API: 1.0

{
  "datasets": [{
    "dataset": "dataset-000",
    "href": "/api/analytics/v1/datasets/arc.accesses[hit/miss]",
    "name": "arc.accesses[hit/miss]"
  }, {
    "dataset": "dataset-001",
    "href": "/api/analytics/v1/datasets/arc.l2_accesses[hit/miss]",
    "name": "arc.l2_accesses[hit/miss]",
  }, {
    "dataset": "dataset-002",
    "href": "/api/analytics/v1/datasets/arc.l2_size",
    "name": "arc.l2_size",
  }, {
    "dataset": "dataset-003",
    "href": "/api/analytics/v1/datasets/arc.size",
    "name": "arc.size",
  }, {
    "dataset": "dataset-004",
```

```

        "href": "/api/analytics/v1/datasets/arc.size[component]",
        "name": "arc.size[component]",
      }, {
        ...
      }]
    }
  }
}

```

データセットの取得

指定されたデータセットからプロパティを取得します。

リクエストの例:

```

GET /api/analytics/v1/datasets/nfs4.ops HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json

```

結果例:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 237
X-Zfssa-Analytics-API: 1.0

{
  "dataset": {
    "activity": "none",
    "dataset": "dataset-030",
    "explanation": "NFSv4 operations per second",
    "grouping": "Protocol",
    "href": "/api/analytics/v1/datasets/nfs4.ops",
    "incore": 296128,
    "name": "nfs4.ops",
    "size": 53211540,
    "suspended": false
  }
}

```

データセットの作成

新しいデータセットを作成します。

リクエストの例:

```

POST /api/analytics/v1/datasets HTTP/1.1
X-Auth-User: root
X-Auth-Key: password
Content-Type: application/json
Content-Length: 26

{"statistic": "test.sine"}

```

結果例:

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 200
Location: /api/analytics/v1/datasets/test.sine
```

```
{
  "dataset": {
    "href": "/api/analytics/v1/datasets",
    "name": "test.sine",
    "grouping": "Test",
    "explanation": "sine per second",
    "incore": 34752,
    "size": 31912,
    "suspended": false,
    "activity": "none"
  }
}
```

データセットの変更

データセットの変更コマンドは、単一データセットのデータ収集を一時停止または再開するために使用します。

一時停止リクエストの例:

```
POST /api/analytics/v1/datasets/nfs4.ops
{"suspended":true}
```

再開リクエストの例:

```
POST /api/analytics/v1/datasets/nfs4.ops
{"suspended":false}
```

結果例:

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 228
X-Zfssa-Analytics-API: 1.0
```

```
{
  "dataset" {
    ...
    "suspended": false
  }
}
```

データセットの破棄

データセットを破棄します。

リクエストの例:

```
DELETE /api/analytics/v1/datasets/test.sine HTTP/1.1
```

結果例:

```
HTTP/1.1 204 No Content
X-Zfssa-Analytics-API: 1.0
```

データセットの保存

データセットを保存します。

リクエストの例:

```
PUT /api/analytics/v1/datasets/nfs4.ops/data
```

結果例:

```
HTTP/1.1 202 Accepted
```

データセットデータの除去

次の表に、データセットの除去で使用する問合せプロパティを示します。

パラメータ	説明
granularity	除去の粒度。second、minute、または hour という粒度値で、データセット内のデータを除去できます。
endtime	指定された endtime の前に収集されたデータを除去します。この時間値の形式については、 21 ページの「問合せパラメータ: start」 を参照してください。

リクエストの例:

次の例では、nfs4.ops データセット内のすべての秒単位、分単位、および時間単位のデータを削除します。毎日、毎週、毎月、または毎年収集されたデータは、データセットに残ります。

```
DELETE /api/analytics/v1/datasets/nfs4.ops/data?granularity=hour
```

結果例:

```
HTTP/1.1 204 No Content
```

リクエストの例:

次の例では、指定された endtime より前に収集された nfs4.ops データセット内のすべての秒単位、分単位、および時間単位のデータを削除します。

```
DELETE /api/analytics/v1/datasets/nfs4.ops/data?granularity=hour&endtime=20130910T00:00:00
```

データセットデータの取得

アクティブな分析データセットからデータを返します。秒単位と粗粒度の両方のデータ取得がサポートされています。

次の表に、データセットデータを取得するための時間ベースの問合せパラメータを示します。

パラメータ	説明
start	サンプルデータの収集を開始する時間、またはデータの収集を開始するサンプル索引。start 時間値は、特定の時間にすることも、キーワード now にすることもできます。特定の時間値の形式については、 21 ページの「問合せパラメータ: start」 を参照してください。デフォルトの start の時刻は、現在の時刻から seconds の値を引いたものです。
seconds	サンプルデータを収集する秒数。デフォルト値は 1 です。span および granularity パラメータが指定された場合、seconds パラメータは無視されます。
span	サンプルデータを収集する期間 (minute、hour、day、week、month、または year)。
granularity	指定された期間内での粒度 (minute、hour、day、week、month、または year)。その粒度からデータポイントの平均が表示されます。

開始時間は未来にすることができません。データを収集する秒数が現在の時間を過ぎてしまう場合、サーバーはデータを返す前に各サンプルを待機します。

粒度データを取得するには、パラメータ span と interval の組み合わせを使用します。span と granularity が使用された場合、seconds パラメータは無視されます。span または granularity が正しく入力されない場合、そのリクエストは無視され、代わりに seconds パラメータが使用されます。不適切な、つまりサポートされないリクエストを入力した場合、「Input span and granularity are not supported.」というエラーメッセージが表示されます。

span および granularity パラメータは、次の方法で組み合わせることができます。

- span が minute の場合、granularity には minute のみを指定できます。
- span が hour の場合、granularity には minute または hour を指定できます。
- span が day の場合、granularity には minute、hour、または day を指定できます。
- span が week の場合、granularity には hour、day、または week を指定できます。

- `span` が `month` の場合、`granularity` には `day`、`week`、または `month` を指定できます。
- `span` が `year` の場合、`granularity` には `week`、`month`、または `year` を指定できます。

次の表に、返されるデータセットデータのプロパティを示します。

プロパティ	説明
<code>data</code>	サンプルデータの配列
<code>sample</code>	サンプルデータの索引番号
<code>startTime</code>	<code>sample</code> が返された時間
<code>min</code>	指定された粒度内での 1 秒あたりの最小値
<code>max</code>	指定された粒度内での 1 秒あたりの最大値

2 秒分のライブデータを収集するリクエストの例:

```
GET /api/analytics/v1/datasets/io.ops[op]/data?start=now&seconds=2 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: text/x-yaml
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: text/x-yaml
X-Zfssa-Analytics-API: 1.0
Transfer-Encoding: chunked

{
  "data": [
    {
      "sample": 457642682,
      "data": {
        "data": [
          {
            "key": "write",
            "value": 199
          }
        ],
        "value": 199
      },
      "startTime": "20200818T18:43:47",
      "samples": 457642683
    },
    {
      "sample": 457642683,
      "data": {
        "data": [
          {
            "key": "write",
            "value": 289
          }
        ],
        "value": 289
      }
    }
  ]
}
```

```

    },
    "startTime": "20200818T18:43:48",
    "samples": 457642684
  }
]
}

```

1 週間の期間内に 7 日分のデータを収集するリクエストの例:

```

GET /api/analytics/v1/datasets/io.ops[op]/data?
start=20200811T15:00:00&granularity=day&span=week
HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: text/x-yaml

```

次のリクエストでは、`io.ops[op]` データセットが「操作のタイプごとに分類された 1 秒あたりの I/O 操作」(読み取りまたは書き込み)であることが報告されています。

```
GET /api/analytics/v1/datasets/io.ops[op]
```

結果の例。簡潔にするために、7 つの出力サンプルのうち 5 つは省略しています。

```

HTTP/1.1 200 OK
Content-Type: text/x-yaml
X-Zfssa-Analytics-API: 1.0
Transfer-Encoding: chunked

{
  "data": [
    {
      "sample": 457197423,
      "data": {
        "max": 3156,
        "data": [
          {
            "max": 588,
            "key": "read",
            "value": 6,
            "min": 0
          },
          {
            "max": 3156,
            "key": "write",
            "value": 45,
            "min": 0
          }
        ],
        "value": 52,
        "min": 0
      },
      "startTime": "20200811T15:00:00",
      "samples": 457644011
    },
    {
      "sample": 457283823,
      "data": {
        "max": 3675,
        "data": [
          {
            "max": 588,
            "key": "read",
            "value": 6,

```

```
        "min": 0
      },
      {
        "max": 3675,
        "key": "write",
        "value": 45,
        "min": 0
      }
    ],
    "value": 52,
    "min": 0
  },
  "startTime": "20200812T15:00:23",
  "samples": 457644011
}
]
```

sample の値を start の値として使用することもできます。次のリクエストの結果は、指定された 1 秒のサンプルからのデータです。

```
GET /api/analytics/v1/datasets/io.ops[op]/data?start=457642682 HTTP/1.1
```


ハードウェアサービス

このセクションでは、ハードウェアクラスタ、シャーシ、およびコンポーネントの管理について説明します。

クラスタ

`ccluster` コマンドは、クラスタ化を設定し、クラスタ化されたリソースを管理します。

表 13 クラスタのコマンド

リクエスト	パス <code>/hardware/v{1 2}</code> の後ろに追加	説明
GET	<code>/cluster</code>	クラスタのプロパティとリソースを一覧表示します
GET	<code>/cluster/resources/resource-path</code>	指定されたクラスタリソースのプロパティを一覧表示します
PUT	<code>/cluster/resources/resource-path</code>	指定されたクラスタリソースを変更します
PUT	<code>/cluster/failback</code>	クラスタピアに割り当てられているすべてのリソースをフェイルバックします
PUT	<code>/cluster/takeover</code>	クラスタピアに割り当てられているすべてのリソースをテイクオーバーします
PUT	<code>/cluster/unconfig</code>	クラスタ化されたアプライアンスをスタンドアロンモードに構成解除します
GET	<code>/cluster/links</code>	クラスタカードリンクステータスを表示します
PUT	<code>/cluster/setup</code>	初期クラスタ設定を実行します

クラスタのプロパティの一覧表示

`ccluster` コマンドは、現在のクラスタ構成状態とリソースのプロパティを一覧表示します。

リクエストの例:

```
GET /api/hardware/v2/cluster HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 529
X-Zfssa-API: 1.0
```

```
{
  "cluster": {
    "state": "AKCS_OWNER",
    "description": "Active (takeover completed)",
    "peer_asn": "d23331e6-41f4-6a15-ac09-a4353e33b43a",
    "peer_hostname": "peer-1",
    "peer_state": "AKCS_STRIPPED",
    "peer_description": "Ready (waiting for failback)",
    "resources": [
      {
        "owner": "peer-1",
        "type": "private",
        "user_label": "peer-1",
        "details": [
          "ipaddr"
        ],
        "href": "/api/hardware/v2/cluster/resources/net/vnic1"
      },
      {
        "owner": "peer-1",
        "type": "singleton",
        "user_label": "",
        "details": [
          "8.03T"
        ],
        "href": "/api/hardware/v2/cluster/resources/zfs/cas1"
      },
      {
        "owner": "peer-2",
        "type": "singleton",
        "user_label": "",
        "details": [
          "18.7T"
        ],
        "href": "/api/hardware/v2/cluster/resources/zfs/cas2"
      }
    ]
  }
}
```

次の例に示すように、`cluster` コマンドで一覧表示されたリソースの 1 つから `href` プロパティを使用して、その特定のクラスタリソースのみのプロパティを一覧表示します。

```
GET /api/hardware/v2/cluster/resources/net/vnic1 HTTP/1.1
```

クラスタリソースの変更

`cluster` コマンドで一覧表示されたリソースの 1 つから `href` プロパティを指定した PUT リクエストを使用して、そのクラスタリソースのプロパティを設定します。

クラスタリンクのステータス

`cluster/links` コマンドは、クラスタカードの現在のリンクステータスを返します。

クラスタの初期設定を実行する前に、`cluster/links` を使用して、すべてのリンクが `AKCIOS_ACTIVE` 状態になっていることを確認します。`AKCIOS_ACTIVE` 状態ではない接続とは、もう一方のシステムが再起動/リブートしていることを意味します。または、リンクが正しく配線されていないか、クラスタケーブルがコネクタでセキュアではないこともあります。クラスタ配線の手順については、『[Oracle ZFS Storage Appliance 配線ガイド](#)』の「[クラスタケーブルの接続](#)」を参照してください。

`cluster/links` コマンドの出力は、コントローラモデルごとに異なります。ZS9-2 コントローラの場合、2 つのクラスタ I/O リンクのステータスが返されます。他のすべてのコントローラの場合、3 つのクラスタ I/O リンクのステータスが返されます。クラスタ I/O リンクの説明は、『[Oracle ZFS Storage Appliance 管理ガイド、Release OS8.8.x](#)』の「[クラスタ相互接続 I/O](#)」を参照してください。

リクエストの例:

```
GET /api/hardware/v2/cluster/links HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Oracle ZFS Storage ZS9-2 の結果の例:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 84

{
  "links": {
    "lio_dev/i40e0 = AKCIOS_ACTIVE\nlio_dev/i40e1 = AKCIOS_ACTIVE"
  }
}
```

ZS7-2 または ZS5-2 コントローラの結果の例:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 181

{
  "links": "\n\tclustron_ng3:0/clustron_uart:0 = AKCIOS_ACTIVE\n\tclustron_ng3:0/clustron_uart:1 = AKCIOS_ACTIVE"
```

```
}          \n\tclustron_ng3:0/dlpi:0 = AKC_IOS_ACTIVE\n\n"
```

その他のコントローラでも同様の `links` 出力が表示されます。唯一の違いは、`/clustron` または `/dlpi` に先行する部分です。たとえば、前述の例の `clustron3_ng3:0` は、ZS5-4 コントローラでは `clustron3:0` に、ZS4-4 コントローラでは `clustron2:0` に、ZS3-2 コントローラでは `clustron2_embedded:0` になります。

クラスタ管理のコマンド

クラスタ管理には、`failback`、`takeover`、および `unconfig` があります。成功すると、コマンドは HTTP ステータス 202 (Accepted) を返します。クラスタがコマンドを受け入れる適切な状態ではない場合、HTTP ステータス 409 (Conflict) が返されます。

テイクオーバーは、ピアの障害が検出されるたびに自動的に試みられます。テイクオーバーは、管理者が実行することもできます。

フェイルバックは、管理者が実行する必要があります。フェイルバック操作は非同期です。REST クライアントが `failback` コマンドを送信すると、リクエストが正常に受信された場合は HTTP ステータス 202 が返されます。フェイルバックの進行状況をモニターするために、クライアントはアラートを待機したり、クラスタ状態をポーリングしたりできます。

テイクオーバーおよびフェイルバックの詳細は、[『Oracle ZFS Storage Appliance 管理ガイド、Release OS8.8.x』](#) の「[クラスタのテイクオーバーとフェイルバック](#)」を参照してください。

リクエストの例:

```
PUT /api/hardware/v2/cluster/failback HTTP/1.1
```

クラスタノードを構成解除すると、ノードがスタンドアロン操作に構成されます。原則として、ユーザー自身でクラスタノードを構成解除しないでください。クラスタノードの構成解除は破壊的な操作です。構成解除には、単なる `unconfig` コマンド以上の影響があります。詳細については、[『Oracle ZFS Storage Appliance 管理ガイド、Release OS8.8.x』](#) の「[クラスタノードの構成解除](#)」を参照してください。

クラスタの設定

設定は、初期クラスタ構成の 1 つのステップです。詳細については、[『Oracle ZFS Storage Appliance 管理ガイド、Release OS8.8.x』](#) の「[クラスタ化構成へのスタンドアロンアプライアンスのアップグレード \(BUI\)](#)」を参照してください。

`cluster/setup` コマンドは、システムの初期クラスタ構成を実行します。`nodename` および `password` プロパティの値を指定します。設定に成功すると、HTTP ステータス 202 (Accepted) が返されます。

cluster/setup コマンドは、次の両方の条件が揃わないと失敗します。

- すべてのクラスタリンクが AKCIOS_ACTIVE 状態です。87 ページの「クラスタリンクのステータス」を参照してください。
- ピアの電源はオンになっていますが、構成されていません。

注記 - 初期クラスタ構成の設定は、完了するのに数分かかることがあります。

リクエストの例:

```
PUT /api/hardware/v2/cluster/setup HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json

{"nodename": "zfs-storage-2", "password": "password"}
```

シャーシ

ハードウェアのコマンドは、アプライアンスハードウェアシャーシおよびコンポーネントのリストを取得するために使用します。

表 14 ハードウェアのコマンド

リクエスト	パス /hardware/v{1 2} の後ろに追加	説明
GET	/chassis	ハードウェアシャーシを一覧表示します
GET	/chassis/chassis	指定されたハードウェアシャーシのプロパティを取得します
PUT	/chassis/chassis	指定されたハードウェアシャーシのプロパティを変更します
GET	/chassis/chassis/fru_type	ハードウェアシャーシのコンポーネントを一覧表示します
GET	/chassis/chassis/fru_type/fru	指定されたシャーシコンポーネントのプロパティを取得します
PUT	/chassis/chassis/fru_type/fru	ハードウェアシャーシコンポーネントのプロパティを変更します

シャーシの一覧表示

シャーシの取得コマンドは、引数を取らず、システムシャーシオブジェクトのリストを返します。コマンドに成功すると、HTTP ステータス 200 (OK) が返されます。

プロパティ	型	説明
name	string	シャーシ名
model	string	シャーシのモデル番号
manufacturer	string	シャーシの製造元
serial	string	シャーシのシリアル番号
revision	string	シャーシのリビジョンレベル
part	string	シャーシの交換パーツ番号
type	string	シャーシのストレージタイプ
faulted	boolean	障害インジケータ
uuid	string	シャーシの UUID 識別子

リクエストの例:

```
GET /api/hardware/v1/chassis HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Content-Length: 788
Content-Type: application/json
X-Zfssa-Appliance-API: 1.0
```

```
{
  "hardware": [{
    "faulted": false,
    "href": "/api/hardware/v1/chassis/chassis-000",
    "manufacturer": "Oracle",
    "model": "Oracle ZFS Storage ZS3-2",
    "name": "hostname",
    "rpm": "--",
    "serial": "1211FM200C",
    "type": "system"
  }, {
    "faulted": false,
    "href": "/api/hardware/v1/chassis/chassis-001",
    "locate": false,
    "manufacturer": "Oracle",
    "model": "Oracle Storage DE2-24C",
    "name": "1235FM4002",
    "part": "7046842",
    "path": 2,
    "revision": "0010",
    "rpm": 7200,
    "serial": "1235FM4002",
    "type": "storage"
  }, {
    "faulted": false,
    "href": "/api/hardware/v1/chassis/chassis-002",
    "locate": false,
    "manufacturer": "Oracle",
    "model": "Oracle Storage DE2-24P",
    "name": "50050cc10c206b96",
    "part": "7046836",
    "path": 2,
```

```

        "revision": "0010",
        "rpm": 10000,
        "serial": "50050cc10c206b96",
        "type": "storage"
    }
}

```

シャーシのコンポーネントの取得

このコマンドは、指定されたシャーシ内のすべてのハードウェアコンポーネントを返します。コマンドに成功すると、HTTP ステータス 200 (OK) が返されます。

リクエストの例:

```

GET /api/hardware/v1/chassis/chassis-001 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json

```

結果例:

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "chassis": {
    "type": "storage"
    "faulted": false,
    "href": "/api/hardware/v1/chassis/chassis-001",
    "locate": false,
    "manufacturer": "Oracle",
    "model": "Oracle Storage DE2-24C",
    "name": "1235FM4002",
    "part": "7046842",
    "path": 2,
    "revision": "0010",
    "rpm": 7200,
    "serial": "1235FM4002",
    "disk": [{
      "device": "c0t5000CCA01A76A2B8d0",
      "faulted": false,
      "href": "/api/hardware/v1/chassis/chassis-001/disk/disk-000",
      "interface": "SAS",
      "label": "HDD 0",
      "locate": false,
      "offline": false,
      "readytoremove": false,
      "manufacturer": "HITACHI",
      "model": "H7230AS60SUN3.0T",
      "pathcount": 2,
      "present": true,
      "revision": "A310",
      "rpm": 7200,
      "serial": "001210R37LVD-----YHJ37LVD",
      "size": 3000592982016,
      "type": "data",
      "use": "peer"
    }, {
      "href": "/api/hardware/v1/chassis/chassis-001/disk/disk-001",
      ...
    }, {

```

```

        "href": "/api/hardware/v1/chassis/chassis-001/disk/disk-002",
        ...
    }, {
        "href": "/api/hardware/v1/chassis/chassis-001/disk/disk-023",
        ...
    }],
    "fan": [
        {
            "href": "/api/hardware/v1/chassis/chassis-001/fan/fan-000",
            ...
        }, {
            "href": "/api/hardware/v1/chassis/chassis-001/fan/fan-007",
            ...
        }],
    "psu": [
        {
            "href": "/api/hardware/v1/chassis/chassis-001/psu/psu-000",
            ...
        }, {
            "href": "/api/hardware/v1/chassis/chassis-001/psu/psu-001",
            ...
        }, {
            "href": "/api/hardware/v1/chassis/chassis-001/psu/psu-002",
            ...
        }, {
            "href": "/api/hardware/v1/chassis/chassis-001/psu/psu-003",
            ...
        }],
    "slot": [{
        "href": "/api/hardware/v1/chassis/chassis-001/slot/slot-000",
        ...
    }, {
        "href": "/api/hardware/v1/chassis/chassis-001/slot/slot-001",
        ...
    }],
    }
}

```

ハードウェアコンポーネントの取得

このコマンドは、1つのハードウェアコンポーネントからのプロパティを返します。コマンドに成功すると、HTTP ステータス 200 (OK) が返されます。レスポンスオブジェクトには、次の表に示すコンポーネントのプロパティが含まれます。

`offline`、`readytoremove`、および `use` プロパティは、プール内のディスクにのみ適用されます。

プロパティ	型	説明
<code>device</code>	string	FRU デバイス ID
<code>faulted</code>	boolean	FRU に障害が発生しているかどうか
<code>interface</code>	string	FRU のインタフェースタイプ
<code>label</code>	string	FRU の場所ラベル
<code>locate</code>	boolean	FRU 位置特定 LED がオンであるかどうか
<code>manufacturer</code>	string	FRU の製造元
<code>model</code>	string	FRU モデル
<code>offline</code>	boolean	ディスクがオフラインであるかどうか
<code>pathcount</code>	integer	ディスクシェルフへの I/O パスの合計数

プロパティ	型	説明
present	boolean	FRU プレゼンスインジケータ
readytoremove	boolean	障害後にディスクドライブが取り外し可能かどうか
revision	string	FRU のファームウェアまたはハードウェアリビジョン
rpm	number	プラッタの RPM (ディスクのみ)
serial	string	FRU シリアル番号
size	number	FRU サイズ (容量)
type	string	コンポーネントタイプ
use	string	コンポーネント使用量の列挙

リクエストの例:

```
GET /api/hardware/v1/chassis/chassis-001/disk/disk-011 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "disk": {
    "device": "c0t5000CCA01A764FB0d0",
    "faulted": false,
    "href": "/api/hardware/v1/chassis/chassis-001/disk/disk-011",
    "interface": "SAS",
    "label": "HDD 11",
    "locate": false,
    "offline": false,
    "readytoremove": false,
    "manufacturer": "HITACHI",
    "model": "H7230AS60SUN3.0T",
    "pathcount": 2,
    "present": true,
    "revision": "A310",
    "rpm": 7200,
    "serial": "001210R322ED-----YHJ322ED",
    "size": 3000592982016,
    "type": "data",
    "use": "peer"
  }
}
```

コンポーネントのプロパティの変更

PUT リクエストは、選択したハードウェアコンポーネントでプロパティを設定するために使用できます。リクエストに成功すると、HTTP ステータス 201 (Accepted) が JSON 形式のコンポーネントプロパティとともに返されます。

リクエストの例:

```
PUT /api/hardware/v1/chassis/chassis-001/disk/disk-011 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-User: root
X-Auth-Key: password
Accept: application/json
Content-Type: application/json
Content-Length: 16

{"locate": true}
```

JSON レスポンスの例:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
Content-Length: 403
Content-Type: application/json

{
  "disk": {
    "href": "/api/hardware/v1/chassis/chassis-001/disk/disk-011",
    "...",
    "locate": true
  }
}
```

ログのコマンド

ログのコマンドは、CLI「maintenance logs」メニューで使用可能なログを管理します。個々のサービスログ情報については、サービス API を参照してください。

ログの管理コマンド

次の表に、ログの管理コマンドの呼び出し方法を示します。

表 15 ログの管理コマンド

リクエスト	パス <code>/api/log/v{1 2}</code> の後ろに追加	説明
GET	<code>/api/log/v{1 2}</code> のみを使用	ログサービスのコマンドを一覧表示します
GET	<code>/logs</code>	すべてのログタイプを一覧表示します
GET	<code>/logs/?start=index/time&limit=entry limit</code>	選択した範囲のログエントリを取得します
GET	<code>/logs/alert</code>	すべてのアラートログを一覧表示します
GET	<code>/logs/alert?start=index/time&limit=entry limit</code>	選択した範囲のログエントリを取得します
GET	<code>/collect</code>	すべてのログエントリのコレクションをダウンロードします
GET	<code>/collect?start=index/time&limit=entry limit</code>	選択した範囲のログエントリのコレクションをダウンロードします

ログの一覧表示

このコマンドは、アプライアンスで使用可能なすべてのログを一覧表示します。各ログは、ログのエントリ数と最後のエントリのタイムスタンプを返します。

注記 - `depth` 問合せパラメータおよび `match_property-name=value` 問合せパラメータはサポートされていません。

リクエストの例:

```
GET /api/log/v1/logs HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 532
X-Zfssa-API: 1.0
```

```
{
  "logs": [
    {
      "href": "/api/log/v1/logs/fault",
      "name": "faults",
      "size": 16,
      "updated": "20130614T22:51:48"
    },
    {
      "href": "/api/log/v1/logs/audit",
      "name": "audits",
      "size": 460149,
      "updated": "20130730T22:10:41"
    },
    {
      "href": "/api/log/v1/logs/alert",
      "name": "alerts",
      "size": 13054,
      "updated": "20130728T00:06:10"
    },
    {
      "href": "/api/log/v1/logs/phone-home",
      "name": "phone-home",
      "size": 249,
      "updated": "20130730T03:22:35"
    },
    {
      "href": "/api/log/v1/logs/system",
      "name": "system",
      "size": 344,
      "updated": "20130724T03:21:55"
    }
  ]
}
```

ログエントリの取得

ログエントリは、指定されたアプライアンスログから返すことができます。各ログエントリは、エントリの日付および時間と、ログ固有の内容のプロパティを返します。

注記 - ログの数によっては、メモリーの制約が原因で、古いログエントリを利用できないことがあります。この同じ制限が BUI と CLI で発生します。すべてのシステムログを取得するには、95 ページの「ログの管理コマンド」で説明されている collect 関数を使用します。

パラメータ	説明
start=index	指定された索引/時間からログの返送を開始します
limit=number	返されるログエントリ数を制限します

開始索引のデフォルトは値 0 であり、生成された最初のログを返します。負の値、およびログサイズ以上の値は使用できません。開始索引は、時間文字列 (例: 20130724T03:21:55) にすることもできます。

注記 - REST では UTC 時間のみ受け入れられます。現在時間より 1 か月以上古い時間値は使用できません。古いログを取得するには、開始値に索引値を使用する必要があります。limit 値は、指定されたリクエストに対して返されるログの数を制限します。指定された limit 値以上は返されません。

リクエストの例:

```
GET /api/log/v1/logs/audit?limit=4&start=1000 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
X-Zfssa-API: development
Transfer-Encoding: chunked

{
  "logs": [
    {
      "address": "192.0.2.0",
      "annotation": "",
      "summary": "User logged in",
      "timestamp": "20131022T22:54:19",
      "user": "root"
    }, {
      "address": "192.0.2.0",
      "annotation": "",
      "summary": "Destroyed share \"zfs-storage-1:tst.volumes.py.34111.project/tst.volumes.py.34111.lun.7\"",
      "timestamp": "20131022T22:52:34",
      "user": "root"
    }, {
      "summary": "Joined workgroup \"RESTTESTWG\"",
      "timestamp": "20131022T22:54:23",
      "user": "<system>"
    }
  ]
}
```

```
    }, {  
      "address": "192.0.2.0",  
      "annotation": "",  
      "summary": "User logged in",  
      "timestamp": "20131022T22:54:19",  
      "user": "root"  
    }  
  ]  
}
```

ログのダウンロード

ログのダウンロードコマンドは、すべてのシステムログが含まれる gzip 圧縮された tar ファイルを返します。ファイルの処理名は `logs.tar.gz` に設定されます。データはリアルタイムで作成および送信されるため、ダウンロードを再開することはできません。

ログのダウンロード

1 つのログタイプのみをダウンロードする場合は、表に示すように、その名前を `collect` リソースに付加できます。ログのテキストはクライアントに返送されます。gzip 圧縮がリクエストされた場合、テキストストリームは gzip で圧縮されます。その他の圧縮タイプはサポートされず、無視されます。

ネットワークのコマンド

このセクションで説明するネットワークのコマンドは、ネットワークアドレスおよびデバイスを表示したり、ネットワークデータリンク、インタフェース、およびルートを構成したりするために使用されます。

ネットワーク構成

ネットワーク構成機能を使用すると、物理的なネットワークポートから、リンクアグリゲーション、仮想 NIC (VNIC)、仮想 LAN (VLAN)、IP ネットワークマルチパス (IPMP) グループなどのさまざまな高度なネットワーク設定を作成できます。次に、システム上のさまざまなデータサービスへの接続で使用するために、これらの抽象化のための任意の数の IPv4 および IPv6 アドレスを定義できます。

システムのネットワーク構成には、次の 4 つのコンポーネントがあります。

- **デバイス** – 物理的なネットワーク接続または InfiniBand (IPoIB) パーティション上の IP に対応する物理的なネットワークポート。
- **データリンク** – パケットを送受信するための基本的な構成体。データリンクは、デバイス (つまり、物理的なネットワークポート) または IB パーティションと 1:1 に対応させることができます。または、ほかのデバイスとデータリンクで構成されたアグリゲーション、VLAN、および VNIC データリンクを定義できます。
- **インタフェース** – IP の構成およびアドレス指定のための基本的な構成体。各 IP インタフェースは 1 つのデータリンクに関連付けられるか、またはほかのインタフェースで構成された IPMP グループとして定義されます。
- **ルーティング** – システムが IP パケットを送信する方法を制御する IP ルーティング構成。

このモデルでは、ネットワークデバイスは使用可能なハードウェアを表し、構成可能な設定を持っていません。データリンクはレイヤー 2 のエンティティであり、これらのネットワークデバイスに LACP などの設定が適用されるように作成する必要があります。インタフェースは、IP 設定が含まれたレイヤー 3 のエンティティです。これらの設定は、データリンクを介して使用可能になります。このモデルでは、ネットワークインタフェースの設定が、レイヤー 2 の設定のためのデータリンクと、レイヤー 3 の設定のためのインタフェースという 2 つの部分に分離されています。

ネットワークデータリンク

ネットワークデータリンクのコマンドは、アプライアンスでのデータリンク管理を提供します。データリンクリソースを一覧表示、変更、作成、および削除できます。

表 16 ネットワークデータリンクのコマンド

リクエスト	パス <code>/network/v{1 2}</code> の後ろに追加	説明
POST	<code>/datalinks</code>	新しいネットワークデータリンクを作成します
GET	<code>/datalinks/datalink</code>	指定されたネットワークデータリンクのプロパティを取得します
GET	<code>/datalinks</code>	すべてのネットワークデータリンクオブジェクトを一覧表示します
PUT	<code>/datalinks/datalink</code>	指定されたネットワークデータリンクオブジェクトを変更します
DELETE	<code>/datalinks/datalink</code>	指定されたデータリンクオブジェクトを破棄します

表 17 物理デバイスデータリンクのプロパティ

プロパティ	型	説明
<code>class</code>	String	「device」(「変更不可」)
<code>label</code>	NetworkLabel	Label
<code>links</code>	ChooseOne	リンク [「igb1」、「igb0」、「ixgbe2」、「ixgbe3」、「igb4」、「igb3」、「ixgbe1」、「igb2」、「igb5」]
<code>jumbo</code>	Boolean	Jumbo Frame を使用します [「true」、「false」] (「非推奨」)
<code>mtu</code>	PositiveInteger	最大転送単位 (MTU)
<code>speed</code>	ChooseOne	リンク速度 [「auto」、「10」、「100」、「1000」、「10000」]
<code>duplex</code>	ChooseOne	リンク二重 [「auto」、「half」、「full」]

表 18 VNIC デバイスデータリンクのプロパティ

プロパティ	型	説明
<code>class</code>	String	「vnic」(「変更不可」)
<code>label</code>	NetworkLabel	Label
<code>links</code>	ChooseOne	リンク [「ixgbe0」]
<code>mtu</code>	PositiveInteger	最大転送単位 (MTU)
<code>id</code>	VLAN	VLAN ID

表 19 VLAN デバイスデータリンクのプロパティ

プロパティ	型	説明
class	String	「vlan」(「変更不可」)
label	NetworkLabel	Label
links	ChooseOne	リンク [「ixgbe0」]
mtu	PositiveInteger	最大転送単位 (MTU)
id	VLAN	VLAN ID

表 20 アグリゲーションベースデバイスデータリンクのプロパティ

プロパティ	型	説明
class	String	「aggregation」(「変更不可」)
label	NetworkLabel	Label
links	ChooseN	リンク [「igb1」、「igb0」、「ixgbe2」、「ixgbe3」、「igb4」、「igb3」、「ixgbe1」、「igb2」、「igb5」]
jumbo	Boolean	Jumbo Frame を使用します [「true」、「false」] (「非推奨」)
mtu	PositiveInteger	最大転送単位 (MTU)
policy	ChooseOne	ポリシー [「L2」、「L3」、「L4」、「L2+L3」、「L2+L4」、「L3+L4」]
mode	ChooseOne	モード [「active」、「passive」、「off」]
timer	ChooseOne	タイマー [「short」、「long」]
key	Integer	アグリゲーションキー (「変更不可」)

表 21 IB パーティションベースデバイスデータリンクのプロパティ

プロパティ	型	説明
class	String	「partition」(「変更不可」)
label	NetworkLabel	Label
links	ChooseOne	リンク
pkey	Pkey	パーティションキー
linkmode	ChooseOne	リンクモード [「cm」、「ud」]

ネットワークデータリンクの一覧表示

アプライアンスですべての構成済みデータリンクを一覧表示します。データリンクリストの各オブジェクトには、単一データリンクリソース上の操作を取得する href と、データリンクのプロパティが含まれます。

リクエストの例:

```
GET /api/network/v1/datalinks HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

JSON データの例:

```
{
  "datalinks": [{
    "href": "/api/network/v1/datalinks/ixgbe0",
    ...
  }, {
    "href": "/api/network/v1/datalinks/ixgbe1",
    ...
  }, {
    "href": "/api/network/v1/datalinks/ixgbe2",
    ...
  }, {
    "href": "/api/network/v1/datalinks/ixgbe3",
    ...
  }]
}
```

ネットワークデータリンクの取得

GET メソッドは、datalink プロパティとデータリンクオブジェクトのリストが含まれる JSON オブジェクトを返します。

```
GET /api/network/v1/datalinks/ixgbe0 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

JSON データの例:

```
{
  "datalink": {
    "class": "device",
    "datalink": "ixgbe0",
    "duplex": "auto",
    "href": "/api/network/v1/datalinks/ixgbe0",
    "jumbo": false,
    "label": "Untitled Datalink",
    "links": [
      "ixgbe0"
    ],
    "mac": "0:21:28:a1:d9:68",
    "mtu": 1500,
    "speed": "auto"
  }
}
```

ネットワークデータリンクの作成

POST コマンドは、新しいデータリンクを作成します。新しいデータリンクを作成するときに必要な追加のプロパティの 1 つが class プロパティで、作成するデータリ

リンクのクラスを定義します。datalinks クラスは、データリンク作成中に定義され、次のクラスタイプのいずれかになります。

- device – デバイスベースのデータリンクを作成します
- vnic – VNIC ベースのデータリンクを作成します
- vlan – VLAN ベースのデータリンクを作成します
- aggregation – アグリゲーションベースのデータリンクを作成します
- partition – IB パーティションのデータリンクを作成します

プロパティは、「configuration net datalinks」メニューで使用可能なものと同じ CLI プロパティにマップされます。

リクエストの例:

```
POST /api/network/v1/datalinks HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-User: root
X-Auth-Key: password
Content-Type: application/json
Content-Length: 78
```

```
{
  "class": "device",
  "jumbo": true,
  "links": ["ixgbe2"],
  "label": "TestDataLink"
}
```

結果例:

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-API: 1.0
Location: /api/network/v1/datalinks/ixgbe2
```

ネットワークデータリンクの変更

PUT メソッドは、データリンクのプロパティを変更するために使用されます。データリンクの設定の詳細については、CLI のドキュメントを参照してください。

リクエストの例:

```
PUT /api/network/v1/datalinks/ixgbe2 HTTP/1.1
{"jumbo": true}
```

結果例:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 219
{
```

```

    "datalink": {
      "href": "/api/network/v1/datalinks/ixgbe2",
      "class": "device",
      "label": "MyDataLink",
      "links": ["ixgbe2"],
      "mac": "0:21:28:a1:d9:6a",
      "mtu": 9000,
      "duplex": "auto",
      "jumbo": true,
      "speed": "auto"
    }
  }
}

```

ネットワークデータリンクの削除

このコマンドは、システムからデータリンクを削除します。指定されたデータリンクを削除するには、href パスを使用します。

リクエストの例:

```
DELETE /api/network/v1/datalinks/ixgbe2 HTTP/1.1
```

結果例:

```
HTTP/1.1 204 No Content
```

ネットワークデバイス

これらのコマンドは、システムの物理ネットワークデバイスを一覧表示します。物理ネットワークデバイスに変更可能なプロパティはありません。

表 22 ネットワークデバイスのコマンド

リクエスト	パス <code>/network/v{1 2}</code> の後ろに追加	説明
GET	<code>/devices/device</code>	指定されたネットワークデバイスのプロパティを取得します
GET	<code>/devices</code>	すべてのネットワークデバイスオブジェクトを一覧表示します

表 23 ネットワークデバイスのプロパティ

プロパティ	説明
active	デバイスがアクティブかどうかを示すブール型フラグ
duplex	デバイスの全二重/半二重
factory_mac	工場出荷時の MAC アドレス

プロパティ	説明
media	デバイスのメディア
speed	デバイス速度 (M ビット/秒)
up	デバイスが動作可能かどうかを示すブール型フラグ

ネットワークデバイスの一覧表示

このコマンドは、すべてのネットワークデバイスを一覧表示します。

リクエストの例:

```
GET /api/network/v1/devices HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 412
X-Zfssa-Gns-API: 1.0

{
  "devices": [{
    "href": "/api/network/v1/devices/ixgbe0",
    ...
  }, {
    "href": "/api/network/v1/devices/ixgbe1",
    ...
  }, {
    "href": "/api/network/v1/devices/ixgbe2",
    ...
  }, {
    "href": "/api/network/v1/devices/ixgbe3",
    ...
  }]
}
```

ネットワークデバイスの取得

このコマンドは、1つのネットワークデバイスからプロパティを取得します。

リクエストの例:

```
GET /api/network/v1/devices/ixgbe0 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
```

```

Content-Type: application/json
Content-Length: 178
X-Zfssa-Gns-API: 1.0

{
  "devices": {
    "active": false,
    "device": "ixgbe0",
    "duplex": "full-duplex",
    "factory_mac": "0:21:28:a1:d9:68",
    "href": "/api/network/v1/devices/ixgbe0",
    "media": "Ethernet",
    "speed": "1000 Mbit/s",
    "up": true
  }
}

```

ネットワークインタフェース

表 24 ネットワークインタフェースのコマンド

リクエスト	パス <code>/api/network/v{1 2}</code> の後ろに追加	説明
POST	<code>/interfaces</code>	新しいネットワークインタフェースを作成します
GET	<code>/interfaces/interface</code>	指定されたネットワークインタフェースのプロパティを取得します
GET	<code>/interfaces</code>	すべてのネットワークインタフェースオブジェクトを一覧表示します
PUT	<code>/interfaces/interface</code>	指定されたネットワークインタフェースオブジェクトを変更します
DELETE	<code>/interfaces/interface</code>	指定されたインタフェースオブジェクトを破棄します

表 25 ネットワークインタフェースのプロパティ

プロパティ	説明
<code>admin</code>	このインタフェースで管理が許可されているかどうかを示すフラグ
<code>class</code>	クラスタイプ (「ip」、「ipmp」) (作成後は変更不可)
<code>curaddr</code>	現在の IP アドレス (変更不可)
<code>enable</code>	このインタフェースが有効かどうかを示すフラグ
<code>label</code>	インタフェースのユーザーラベル
<code>links</code>	このインタフェースのネットワークリンクを選択します
<code>state</code>	インタフェースの状態 (変更不可)
<code>v4addr</code>	IPv4 アドレス

プロパティ	説明
v6dhcp	IPv4 DHCP フラグ
v6addrs	IPv6 アドレス
v6dhcp	IPv6 DHCP フラグ

ネットワークインタフェースの一覧表示

このコマンドは、構成済みのすべてのネットワークインタフェースを一覧表示します。

リクエストの例:

```
GET /api/network/v1/interfaces HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 369

{
  "interfaces": [
    {
      "href": "/api/network/v1/interfaces/ixgbe0",
      "v4addrs": ["ipaddr-1"]
    },
    {
      "href": "/api/network/v1/interfaces/ixgbe1",
      "v4addrs": ["ipaddr-2"]
    },
    {
      "href": "/api/network/v1/interfaces/ixgbe2",
      "v4addrs": ["ipaddr-3"]
    },
    {
      "href": "/api/network/v1/interfaces/ixgbe3",
      "v4addrs": ["ipaddr-4"]
    }
  ]
}
```

ネットワークインタフェースの取得

このコマンドは、指定されたネットワークインタフェースのプロパティの完全なリストを取得します。

リクエストの例:

```
GET /api/network/v1/interfaces/ixgbe0 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
```

```
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 292
```

```
{
  "interface": {
    "admin": true,
    "class": "ip",
    "curaddrs": ["ipaddr-1"],
    "enable": true,
    "href": "/api/network/v1/interfaces/ixgbe0",
    "interface": "ixgbe0",
    "label": "Untitled Interface",
    "links": ["ixgbe0"],
    "state": "up",
    "v4addrs": ["ipaddr-1"],
    "v4dhcp": false,
    "v6addrs": [],
    "v6dhcp": false
  }
}
```

ネットワークインタフェースの作成

このコマンドは、新しいネットワークインタフェースを作成します。

リクエストの例:

```
POST /api/network/v1/interfaces HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-User: root
X-Auth-Key: password
Content-Type: application/json
Content-Length: 78
```

```
{
  "class": "ip",
  "links": ["ixgbe3"],
  "v4addrs": "192.0.2.0/24"
}
```

結果例:

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-API: 1.0
Location: /api/network/v1/interfaces/ixgbe3
```

ネットワークインタフェースの変更

このコマンドは、既存のネットワークインタフェースを変更します。

リクエストの例:

```
PUT /api/network/v1/interfaces/ixgbe3 HTTP/1.1

{
  "v4addrs": ["192.0.2.0/24"],
  "interface": "Demo Rest"
}
```

結果例:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 219

{
  "admin": true,
  "class": "ip",
  "curaddrs": ["192.0.2.0/24"],
  "enable": true,
  "href": "/api/network/v1/interfaces/ixgbe3",
  "interface": "ixgbe3",
  "label": "Demo Rest",
  "links": ["ixgbe3"],
  "state": "failed",
  "v4addrs": ["192.0.2.0/24"],
  "v4dhcp": false,
  "v6addrs": [],
  "v6dhcp": false
}
```

ネットワークインタフェースの削除

このコマンドは、既存のネットワークインタフェースを削除します。

注記 - インタフェースが削除されると、そのインタフェースに関連付けられたすべてのルートも削除されます。

リクエストの例:

```
DELETE /api/network/v1/interfaces/ixgbe3 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
```

結果例:

```
HTTP/1.1 204 No Content
```

ネットワークルート

これらのコマンドは、ネットワークルートを管理します。

表 26 ネットワークルートの管理

リクエスト	パス <code>/api/network/v{1 2}</code> の後ろに追加	説明
POST	<code>/routes</code>	新しいネットワークルートを作成します
GET	<code>/routes/route</code>	指定されたネットワークルートのプロパティを取得します
GET	<code>/routes</code>	すべてのネットワークルートオブジェクトを一覧表示します
DELETE	<code>/routes/route</code>	指定されたルートオブジェクトを破棄します
GET	<code>/routing</code>	ネットルーティングプロパティを取得します
PUT	<code>/routing</code>	ネットルーティングプロパティを変更します

表 27 ネットワークルートの管理プロパティ

プロパティ	説明
<code>type</code>	ルートのタイプ (「system」、「static」など (変更不可))
<code>family</code>	アドレスファミリー (IPv4 または IPv6)
<code>destination</code>	ルーティング先アドレス
<code>gateway</code>	ゲートウェイアドレス
<code>interface</code>	ネットワークデータリンクのインタフェース

各ルートへの href パスでは、CLI で設定されたルート ID を使用しますが、これらの値はルートの変更に伴って変更されることがあります。API は、ルート内で固有のプロパティを使用することで、単一ルートの選択をサポートします。その構文は、`routes/route-###` に対して `routes/name=value` になります。

ルートの一覧表示

アプライアンス上で作成されたすべてのネットワークルートを一覧表示します。

リクエストの例:

```
GET /api/network/v1/routes HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 192

{
```

```

"routes": [{
  "destination": "ipaddr-0",
  "family": "IPv4",
  "gateway": "ipaddr-1",
  "href": "/api/network/v1/routing/route-000",
  "interface": "ixgbe0",
  "mask": 0,
  "route": "route-000",
  "type": "static"
}, {
  "destination": "ipaddr-2",
  "family": "IPv4",
  "gateway": "ipaddr-3",
  "href": "/api/network/v1/routes/route-001",
  "interface": "ixgbe0",
  "mask": 24,
  "route": "route-001",
  "type": "system"
}]
}

```

ルートの取得

単一ルートのプロパティを取得します。

リクエストの例:

```

GET /api/network/v1/routes/destination=ipaddr-1 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json

```

結果例:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 192

{
  "route": {
    "destination": "ipaddr-1",
    "family": "IPv4",
    "gateway": "ipaddr-2",
    "href": "/api/network/v1/routes/route-001",
    "interface": "ixgbe0",
    "mask": 24,
    "route": "route-001",
    "type": "system"
  }
}

```

ルートの追加

新しいネットワークルートを作成します。ルート href の値は、ほかのルートがシステムに追加されると変更されることがあります。作成時には、返されるプロパティ

は入力プロパティと同一であるため、ルート情報は返されません。作成に成功すると、HTTP ステータス 204 (Created) が返されます。

静的ルートを作成するリクエストの例:

```
POST /api/network/v1/routes HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Content-Type: application/json
Content-Length: 164
```

```
{
  "family": "IPv4",
  "destination": "ipaddr-0",
  "mask": "0",
  "gateway": "ipaddr-1",
  "interface": "ixgbe0"
}
```

結果例:

```
HTTP/1.1 201 Created
```

ルートの削除

既存のネットワークルートを削除します。

リクエストの例:

```
DELETE /api/network/v1/routes/route-001 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
```

結果例:

```
HTTP/1.1 204 No Content
```

RESTful API クラウドサービス

RESTful API クラウドサービスを使用すると、アプライアンス管理者は次のタスクを行うことができます。

- アプライアンスから Oracle Cloud Infrastructure Object Storage に ZFS スナップショットデータをバックアップします。
- クラウドで使用可能なスナップショットバックアップを一覧表示します。
- スナップショットバックアップを新しいシェアとしてアプライアンスに復元します。
- スナップショットバックアップデータが不要になったら、削除します。

複数のアプライアンスが同じクラウドターゲットにバックアップできます。クラウドバックアップは、クラウドターゲットにアクセスできる任意のアプライアンスに新しくクローニングされたシェアとして復元できます。

次のトピックを含む、クラウドバックアップの詳細は、[『Oracle ZFS Storage Appliance 管理ガイド、Release OS8.8.x』](#) の「クラウドバックアップの構成」を参照してください。

- 「標準」または「アーカイブ」層を含む、Oracle Cloud Infrastructure アカウントの構成
- データバケット、メタデータバケット、およびアーカイブバケットの管理
- クラスタ構成によるクラウドバックアップ操作への影響

次の操作を含む Oracle Cloud Infrastructure Object Storage サービスの詳細は、[『Oracle ZFS Storage Appliance Object API Guide for Oracle Cloud Infrastructure Object Storage Service Support, Release OS8.8.x』](#) の「Oracle ZFS Storage Appliance RESTful API Support for the Oracle Cloud Infrastructure Object Storage Service」を参照してください。

- Oracle Cloud Infrastructure サービスを有効にします。
- ユーザー鍵を作成または削除します。ユーザー鍵の操作権限を変更します。

クラウドサービス操作

クラウドサービス操作を使用して、Oracle Cloud Infrastructure Object Storage のアプライアンススナップショットのバックアップを管理します。Oracle Cloud Infrastructure

Object Storage のターゲットとバックアップの一覧表示、ターゲットの削除、バックアップの削除、バックアップを新しいアプライアンスシェアとして復元、およびクラウドサービスジョブの取り消しや再起動を行うことができます。

バックアップを作成するには、[128 ページの「スナップショットバックアップの作成」](#) in [127 ページの「スナップショットバックアップ操作」](#) を参照してください。バックアップは、シェアスナップショットの完全バックアップまたは増分バックアップです。スナップショットを作成するには、[185 ページの「スナップショットおよびクローンの操作」](#) を参照してください。

クラウドサービス操作の表では、次のパラメータが使用されます。

<i>format</i>	バックアップが Oracle Cloud Infrastructure Object Storage に保存された形式。 <i>format</i> の値は、<code>zfs</code> または <code>tar</code> のいずれかです。スナップショットバックアップの作成に <i>format</i> が指定されていない場合、<code>zfs</code> がデフォルトとなります。<code>zfs</code> 形式はファイルシステムと LUN の両方のスナップショットをサポートし、<code>tar</code> 形式はファイルシステムのスナップショットのみをサポートします。<code>zfs</code> および <code>tar</code> 形式の詳細は、『Oracle ZFS Storage Appliance 管理ガイド、Release OS8.8.x』の「クラウドバックアップの作成 (CLI)」を参照してください。
<i>backup-id</i>	Oracle Cloud Infrastructure Object Storage に保存されているアプライアンススナップショットのコピーの識別子。<i>backup-id</i> の値は、120 ページの「クラウドバックアップの一覧表示」 に示すように、バックアップを一覧表示するときの <code>id</code> プロパティの値です。 完全スナップショットバックアップの場合、次の例に示すように、<i>backup-id</i> の値は、<code>pool_id/snapshot_id</code> です。 <pre>3e035b7e546e0d02/1cbfdb5ff2259b76</pre> 増分スナップショットバックアップの場合、次の例に示すように、<i>backup-id</i> の値は、<code>pool_id/child_snapshot_id-parent_snapshot_id</code> です。 <pre>6913a5703bee98dc/46be95ced54e99d9-667f3eb88fd209e1</pre>
<i>target-id</i>	バックアップが保存される Oracle Cloud Infrastructure Object Storage の場所。特定のバックアップを複数のターゲットに保存できます。つまり、同じ <i>backup-id</i> が異なる <i>target-id</i> の場所に表示される可能性があります。 <i>target-id</i> の値は、バックアップを一覧表示するときの <code>target</code> プロパティの値です。

job-id 実行中のジョブの識別子。*job-id* の値は、[124 ページの「ジョブの一覧表示」](#)に示すように、ジョブを一覧表示するときの *id* プロパティの値です。

表 28 クラウドサービスコマンド

リクエスト	パス <i>/api/service/v2/services</i> の後ろに追加	説明
GET	<i>/cloud</i>	ターゲット、バックアップ、およびジョブのプロパティとサマリーデータを一覧表示します。
PUT	<i>/cloud</i>	プロパティを変更します。
POST	<i>/cloud/targets</i>	新しいターゲットを作成します。
GET	<i>/cloud/targets</i>	ターゲットを一覧表示します。
GET	<i>/cloud/targets/target-id</i>	指定されたターゲットのプロパティを一覧表示します。
PUT	<i>/cloud/targets/target-id</i>	指定されたターゲットのプロパティを変更します。
DELETE	<i>/cloud/targets/target-id</i>	指定されたターゲットをサービスから削除します。
GET	<i>/cloud/backups</i>	任意の形式の完了したすべてのバックアップを一覧表示します。
GET	<i>/cloud/backups/format/backup-id/target-id</i>	指定されたバックアップを一覧表示します。
DELETE	<i>/cloud/backups/format/backup-id/target-id</i>	指定されたバックアップを削除します (ジョブリクエストを発行します)。
POST	<i>/cloud/backups/format/backup-id/target-id/restore</i>	指定されたバックアップを復元します (ジョブリクエストを発行します)。
GET	<i>/cloud/jobs</i>	実行中のジョブおよび最近完了したジョブを一覧表示します。
GET	<i>/cloud/jobs/job-id</i>	指定されたジョブのプロパティを一覧表示します。
PUT	<i>/cloud/jobs/job-id/cancel</i>	指定された実行中のジョブを取り消します。
PUT	<i>/cloud/jobs/job-id/restart</i>	指定された中断したジョブを再起動します。

クラウドサービスの有効化

クラウドサービスを有効にするには、次の例に示すように、*status* を *enable* に設定します。

リクエストの例:

```
PUT /api/service/v2/services/cloud HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
Content-Type: application/json
Content-Length: 22
```

```
{ "<status>": "enable" }
```

クラウドサービスログファイルの表示

次のリクエストを使用して、クラウドサービスのログファイルを表示します。

```
GET /api/log/v1/logs/appliance-kit-cloud:default HTTP/1.1
```

クラウドサービスのプロパティの一覧表示

リクエストの例:

```
GET /api/service/v2/services/cloud HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
```

結果例:

```
HTTP/1.1 200 OK
Date: Wed, 24 Jul 2019 20:30:59 GMT
Content-Length: 843
X-Zfssa-Service-API: 2.0
X-Zfssa-API-Version: 2.0
Content-Type: application/json; charset=utf-8

{
  "service": {
    "href": "/api/service/v2/services/cloud",
    "<status>": "online",
    "tls_version": [
      "TLSv1.1",
      "TLSv1.2"
    ],
    "ciphers": [
      "ECDHE-RSA-AES128-GCM-SHA256",
      "ECDHE-RSA-AES256-GCM-SHA384",
      "DHE-RSA-AES128-GCM-SHA256",
      "DHE-RSA-AES256-GCM-SHA384",
      "AES128-SHA",
      "AES256-SHA",
      "DES-CBC3-SHA"
    ],
    "targets": {
      "href": "/api/service/v2/services/cloud/targets",
      "entries": 2
    },
    "backups": {
      "href": "/api/service/v2/services/cloud/backups",
      "entries": 2548
    },
    "jobs": {
      "href": "/api/service/v2/services/cloud/jobs",
      "entries": 0
    }
  }
}
```

```
}
```

クラウドサービスのプロパティの変更

リクエストの例:

```
PUT /api/service/v2/services/cloud HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
Content-Type: application/json
Content-Length: 48
```

```
{
  "tls_version": [
    "TLSv1.0", "TLSv1.1", "TLSv1.2"
  ]
}
```

ターゲットの一覧表示

次の例では、すべてのターゲットを一覧表示します。

リクエストの例:

```
GET /api/service/v2/services/cloud/targets HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
```

結果例:

```
HTTP/1.1 200 OK
Date: Wed, 24 Jul 2019 21:06:18 GMT
Content-Length: 1086
X-Zfssa-Service-API: 2.0
X-Zfssa-API-Version: 2.0
Content-Type: application/json; charset=utf-8

{
  "targets": [{
    "bucket": "pl-test",
    "href": "/api/service/v2/services/cloud/targets/target-id1",
    "id": "target-id1",
    "key": true,
    "location": "https://objectstorage.us-ashburn-1.oraclecloud.com",
    "name": "oci-ashburn",
    "online": true,
    "proxy_host": "",
    "proxy_on": false,
    "proxy_password": false,
    "proxy_user": "",
    "tenancy": "ocid1.tenancy.oc1..tenancy-id",
    "user": "ocid1.user.oc1..user-id"
  }, {
    "bucket": "pl-test2",
```

```

    "href": "/api/service/v2/services/cloud/targets/target-id2",
    "id": "target-id2",
    "key": true,
    "location": "https://objectstorage.us-phoenix-1.oraclecloud.com",
    "name": "oci-phoenix",
    "online": true,
    "proxy_host": "www-proxy.example.com:80",
    "proxy_on": true,
    "proxy_password": false,
    "proxy_user": "",
    "tenancy": "ocid1.tenancy.oc1..tenancy-id",
    "user": "ocid1.user.oc1..user-id"
  }
}

```

次の例では、指定されたターゲットを一覧表示します。

```

GET /api/service/v2/services/cloud/targets/target-id HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*

```

ターゲットの作成

クラウドターゲットを作成するには、次のパラメータが必要です。

パラメータ	説明
<i>key</i>	<i>user-id</i> の Oracle Cloud Infrastructure アカウントユーザー鍵
<i>tenancy-id</i>	Oracle Cloud Infrastructure アカウントテナンシ名の OCID
<i>user-id</i>	Oracle Cloud Infrastructure アカウントユーザー名の OCID

次の例では、ターゲットを作成します。

リクエストの例:

```

POST /api/service/v2/services/cloud/targets HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=

{
  "name": "oci4",
  "location": "https://objectstorage.us-phoenix-1.oraclecloud.com",
  "user": "ocid1.user.oc1..user-id",
  "bucket": "test-bucket3",
  "tenancy": "ocid1.tenancy.oc1..tenancy-id",
  "key": "key",
  "proxy_on": false,
  "readlimit": -1,
  "writelimit": -1
}

```

結果例:

```

HTTP/1.1 201 Created

```

```
Date: Wed, 24 Jul 2019 21:14:39 GMT
Content-Length: 568
X-Zfssa-Service-API: 2.0
Location: /api/service/v2/services/cloud/targets/target-id
X-Zfssa-API-Version: 2.0
Content-Type: application/json; charset=utf-8
```

```
{
  "target": {
    "bucket": "test-bucket3",
    "href": "/api/service/v2/services/cloud/targets/target-id",
    "id": "target-id",
    "key": true,
    "location": "https://objectstorage.us-phoenix-1.oraclecloud.com",
    "name": "oci4",
    "proxy_on": false,
    "readlimit": -1,
    "state": "offline",
    "tenancy": "ocid1.tenancy.oc1..tenancy-id",
    "user": "ocid1.user.oc1..user-id",
    "writelimit": -1
  }
}
```

ターゲットの変更

次の表に、クラウドターゲットで変更できるプロパティを示します。

プロパティ	説明
name	このクラウドターゲットの名前。アプライアンスごとに一意である必要があります。
proxy_on	true の場合、インターネットとのシステム通信にプロキシを使用します。 proxy_on の値が true の場合は、proxy_host の値を指定する必要があります。
proxy_host	プロキシホスト名およびポート番号。
proxy_password	オプション。プロキシパスワード。
proxy_user	オプション。プロキシユーザー名。
readlimit	オプション。データがクラウドターゲットから読み取られる最大レート (バイト/秒)。この値は、クラウドバックアップをクラウドから復元する際に使用されます。たとえば、値が 4194304 の場合は、クラウドターゲットからの読み取り速度が 4 メガバイト/秒に制限されます。値 -1 は I/O が制限されていないことを意味します。
writelimit	オプション。データがクラウドターゲットに書き込まれる最大レート (バイト/秒)。この値は、クラウドスナップショットをクラウドターゲットにアップロードする際に使用されます。たとえば、値が 5242880 の場合は、クラウドターゲットへの書き込み速度が 5 メガバイト/秒に制限されます。値 -1 は I/O が制限されていないことを意味します。

リクエストの例:

```
PUT /api/service/v2/services/cloud/targets/target-id HTTP/1.1
Host: hostname:215
```

```
Authorization: Basic Tm8gcGVla2luZyE=
User-Agent: curl/7.54.0
Accept: */*
Content-Type: application/json
Content-Length: 19
{
  "proxy_on": true,
  "proxy_host": "www-proxy.example.com:80",
  "readlimit": 4194304,
  "writelimit": 5242880
}
```

ターゲットの削除

ターゲットを削除する前に、次のチェックを実行します。

- このターゲットへのバックアップが進行中であるかどうかを確認します。[124 ページの「ジョブの一覧表示」](#)を参照してください。
- このターゲットにバックアップがあるかどうかを判断します。[120 ページの「クラウドバックアップの一覧表示」](#)に示すように、`target` フィルタを使用して、このターゲットに保存されているバックアップを一覧表示します。

次の例では、指定されたクラウドターゲットをサービスから削除します。

リクエストの例:

```
DELETE /api/service/v2/services/cloud/targets/target-id HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
```

結果例:

```
HTTP/1.1 204 No Content
Date: Wed, 24 Jul 2019 21:20:27 GMT
X-Content-Type-Options: nosniff
X-Zfssa-Service-API: 2.0
X-Zfssa-API-Version: 2.0
Content-Type: application/json; charset=utf-8
```

クラウドバックアップの一覧表示

次の問合せを使用して、任意の形式の完了したすべてのバックアップを、新しいバックアップから順に一覧表示します。進行中のバックアップに関する情報を取得するには、[124 ページの「ジョブの一覧表示」](#)を参照してください。

リクエストの例:

```
GET /api/service/v2/services/cloud/backups HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
```

結果例:

この例では、一覧表示されている最初のバックアップは、一覧表示されている 2 番目のバックアップの子です。1 つ目のバックアップの `parent` の値は、2 つ目のバックアップ (`app-data-fullsnap`) の `dataset` の値と同じであり、2 つ目のバックアップの `parent` の値は `null` です。子のバックアップは、親と同じ `format` になります。この例では、子のバックアップと親のバックアップの両方が `tar` 形式です。

```
HTTP/1.1 200 OK
Date: Wed, 22 Jan 2020 21:22:40 GMT
X-Zfssa-Service-API: 2.0
X-Zfssa-API-Version: 2.0
Content-Type: application/json; charset=utf-8
Transfer-Encoding: chunked

{
  "backups": [{
    "target": "target-id2",
    "parent": "app-data-fullsnap",
    "started": "2020-01-06T20:03:32Z",
    "completed": "2020-01-06T20:04:36Z",
    "id": "backup-id2",
    "source": "cloudsnap0",
    "href": "/api/service/v2/services/cloud/backups/tar/backup-id2/target-id2",
    "tier": "standard",
    "dataset": "p1/local/default/app-data@app-data-incsnap",
    "format": "tar",
    "size": 3224982536.0
  }, {
    "target": "target-id1",
    "parent": "",
    "started": "2020-01-06T20:01:16Z",
    "completed": "2020-01-06T20:01:17Z",
    "id": "backup-id1",
    "source": "cloudsnap0",
    "href": "/api/service/v2/services/cloud/backups/tar/backup-id1/target-id1",
    "tier": "standard",
    "dataset": "p1/local/default/app-data@app-data-fullsnap",
    "format": "tar",
    "size": 2149988056.0
  }
]
```

次のリクエストを使用して、指定されたバックアップを一覧表示します。

```
GET /api/service/v2/services/cloud/backups/format/backup-id/target-id HTTP/1.1
```

クラウドバックアップの一覧表示の問合せパラメータ

クラウドバックアップのリストをフィルタ処理するために、次の問合せパラメータがサポートされています。すべてのパラメータを同じ問合せで使用できます。表の後の例に示すように、問合せパラメータ間で AND 演算 (&) を使用します。

表 29 クラウドバックアップの一覧表示の問合せパラメータ

プロパティ	説明
start	start の値には、次のいずれかを指定します。

プロパティ	説明
	■ バックアップ索引。この索引の値は、116 ページの「クラウドサービスのプロパティの一覧表示」に示されているリクエストの <code>backups</code> セクションの <code>entries</code> プロパティの値よりも 0 から 1 小さい値になります。 指定された索引によって選択されたバックアップと、指定されたバックアップの作成後に作成された最も古いバックアップを一覧表示します。 ■ <code>%Y-%m-%dT%H:%M:%SZ</code> 形式の時間。これは、<code>started</code> プロパティと <code>completed</code> プロパティの値に使用されるのと同じ形式です。この時間は、現在の時間までの任意のリアルタイムにすることができます。たとえば、<code>2019-09-00T00:00:00Z</code> はリアルタイムではありません。 指定した時刻以降に作成された最も古いバックアップを一覧表示します。
<code>end</code>	<code>end</code> の値は、<code>%Y-%m-%dT%H:%M:%SZ</code> の形式の時間です。これは、<code>started</code> プロパティと <code>completed</code> プロパティの値に使用されるのと同じ形式です。この時間は、現在の時間までの任意のリアルタイムにすることができます。たとえば、<code>2019-09-00T00:00:00Z</code> はリアルタイムではありません。 指定した時刻以前に作成された最も新しいバックアップを一覧表示します。
<code>limit</code>	指定された数未満のバックアップを一覧表示します。 <code>limit</code> パラメータにはデフォルト値がありません。
<code>target</code>	<code>target</code> プロパティの値。そのターゲットの最も新しいバックアップを一覧表示します。
<code>source</code>	<code>source</code> プロパティの値。そのソースの最も新しいバックアップを一覧表示します。
<code>dataset</code>	@ の前の <code>dataset</code> プロパティの値。たとえば、データセット <code>p1/local/default/app-data@app-data-incsnap</code> の場合は、<code>p1/local/default/app-data</code> または <code>app-data</code> を指定します。 指定されたデータセット内の最も新しいバックアップを一覧表示します。 注記 - データセットバックアップのリクエストには、リクエストされたデータセット名を含む名前を持つすべてのデータセットのすべてのバックアップが一覧表示されます。たとえば、<code>p1/local/default/app-data@app-data-incsnap</code> のバックアップをリクエストすると、<code>app-data-incsnap</code>、<code>app-data-incsnap-1</code>、および <code>myproj-app-data-incsnap</code> という名前のデータセットのバックアップも返されます。
<code>format</code>	<code>format</code> プロパティの値。 <code>zfs</code> または <code>tar</code> のいずれかです。指定されたバックアップ形式の最も新しいバックアップを一覧表示します。

次の例では、索引番号 2000 のバックアップから始まる最も古いバックアップを一覧表示します。クラウドサービスのプロパティリストの `backups` セクションにある `entries` プロパティの値が 2865 の場合、次の例では、バックアップ 2000 からバックアップ 2864 までの 865 個のバックアップを示しています。

```
GET /api/service/v2/services/cloud/backups?start=2000
```

次の例では、索引番号 2000 のバックアップのみを一覧表示します。

```
GET /api/service/v2/services/cloud/backups?start=2000&limit=1
```

次の例では、指定した時刻以降に作成された最も古い 500 個のバックアップを一覧表示します。

```
GET /api/service/v2/services/cloud/backups?start=2019-07-12T00:00:00Z&limit=500
```

次の例では、指定した時刻以前に作成された最も新しい 500 個のバックアップを一覧表示します。

```
GET /api/service/v2/services/cloud/backups?end=2019-07-12T00:00:00Z&limit=500
```

次の例では、指定した **start** 時間以降に作成され、指定した **end** 時間までに作成されたすべてのバックアップを一覧表示します。

```
GET /api/service/v2/services/cloud/backups?start=2019-07-11T00:00:00Z&end=2019-07-12T00:00:00Z
```

次の例では、**tar** バックアップ形式で、名前に **app-data** を含むデータセットの最も新しいバックアップを一覧表示します。

```
GET /api/service/v2/services/cloud/backups?dataset=app-data&format=tar
```

次の例では、**target-id** ターゲット上の、名前に **app-data** を含むデータセットの最も新しいバックアップを一覧表示します。

```
GET /api/service/v2/services/cloud/backups?dataset=app-data&target=target-id
```

次の例では、**cloudsnap0** をソースとする **target-id** ターゲット上の、名前に **app-data** を含むデータセットの最も新しいバックアップを一覧表示します。

```
GET /api/service/v2/services/cloud/backups?dataset=app-data&target=target-id&source=cloudsnap0
```

クラウドバックアップの削除

クラウドバックアップを削除する前に、次のチェックを実行します。

- このバックアップの復元が進行中であるかどうかを確認します。[124 ページの「ジョブの一覧表示」](#)を参照してください。
- このバックアップに子があるかどうかを判断します。子バックアップを持つバックアップは削除できません。このバックアップの子には、削除するバックアップの **dataset** 値と同じ **parent** 値があります。[120 ページの「クラウドバックアップの一覧表示」](#)を参照してください。

次の例では、Oracle Cloud Infrastructure Object Storage から指定されたターゲット上の指定されたバックアップを削除するジョブリクエストを送信します。アプライアンスは、バックアップが別のアプライアンスで作成された場合でも、アプライアンスがアクセスできる任意のターゲット上の任意のバックアップで動作します。

この操作を、アプライアンス上のシェアスナップショットバックアップを削除する方法を示した[132 ページの「スナップショットバックアップの削除」](#)と比較してみてください。

リクエストの例:

```
DELETE /api/service/v2/services/cloud/backups/format/backup-id/target-id HTTP/1.1
```

```
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
```

結果例:

```
{
  "action": "job-id"
}
```

クラウドバックアップの削除の進行状況を表示するには、[124 ページの「ジョブの一覧表示」](#)を使用して、前述の *job-id* のジョブを表示します。

クラウドバックアップの復元

クラウドバックアップは、そのクラウドターゲットにアクセスできる任意のアプライアンス上に新しくクローニングされたシェアとして復元できます。

次の例では、指定されたバックアップを復元するジョブリクエストを送信します。バックアップを復元するプールとプロジェクトを指定し、新しいシェアの名前を指定します。復元の進行状況を表示するには、[124 ページの「ジョブの一覧表示」](#)を使用して、返された *job-id* のジョブを表示します。

ターゲットに `readlimit` プロパティが設定されている場合、ターゲットから読み取られるのは 1 秒あたり `readlimit` バイト以下です。[118 ページの「ターゲットの作成」](#)を参照してください。

リクエストの例:

```
POST /api/service/v2/services/cloud/backups/format/backup-id/target-id/restore HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
Content-Length: 55
```

```
{
  "pool": "p1",
  "project": "default",
  "share": "restore6"
}
```

結果例:

```
{
  "action": "job-id"
}
```

ジョブの一覧表示

次の例では、実行中のすべてのジョブと最近完了したジョブを、新しいジョブから順に一覧表示します。

リクエストの例:

```
GET /api/service/v2/services/cloud/jobs HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
```

結果例:

この例では、2つのバックアップジョブが2つの異なるバックアップを同じターゲットに保存しています。

```
HTTP/1.1 200 OK
Date: Wed, 22 Jan 2020 21:37:52 GMT
Content-Length: 983
X-Zfssa-Service-API: 2.0
X-Zfssa-API-Version: 2.0
Content-Type: application/json; charset=utf-8

{
  "jobs": [{
    "href": "/api/service/v2/services/cloud/jobs/job-id2",
    "op": "backup",
    "target": "target-id",
    "targetName": "oci-ashburn",
    "created": "2020-02-06T16:52:42Z",
    "updated": "2020-02-06T16:52:48Z",
    "id": "job-id2",
    "status": "in-progress",
    "rate": 10002432,
    "transferred": 80019456,
    "estimated_size": 43088792088,
    "dataset": "p1/local/default/f-1",
    "backup": "backup-id2",
    "snapshot": "snap3",
    "format": "tar",
    "details": "uploading backup to zfs/backups/tar/backup-id2/000000001"
  }, {
    "href": "/api/service/v2/services/cloud/jobs/job-id1",
    "op": "backup",
    "target": "target-id",
    "targetName": "oci-ashburn",
    "created": "2020-02-06T16:52:28Z",
    "updated": "2020-02-06T16:52:48Z",
    "id": "job-id1",
    "status": "in-progress",
    "rate": 1942,
    "transferred": 3884,
    "estimated_size": 0,
    "dataset": "p1/local/default/f-1",
    "backup": "backup-id1",
    "snapshot": "snap2",
    "format": "zfs",
    "details": "uploading backup to zfs/backups/zfs/backup-id1/000000001"
  }
]
```

次の例では、指定されたジョブを一覧表示します。

```
GET /api/service/v2/services/cloud/jobs/job-id2 HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
```

クラウドバックアップジョブの一覧表示の問合せパラメータ

クラウドバックアップジョブのリストをフィルタ処理するために、start および limit 問合せパラメータがサポートされています。

- start パラメータの値はジョブ索引です。この索引の値は、[116 ページの「クラウドサービスのプロパティの一覧表示」](#)に示されているリクエストの jobs セクションの entries プロパティの値よりも 0 から 1 小さい値になります。
- limit パラメータの値は、一覧表示するジョブの最大数です。

両方のパラメータを同じ問合せで使用できます。次の例に示すように、問合せパラメータ間で AND 演算 (&) を使用します。

次の例では、実行中の最も新しい 100 個のジョブと最近完了したジョブを、新しいジョブから順に一覧表示します。

```
GET /api/service/v2/services/cloud/jobs?limit=100
```

次の例では、実行中の最も古いジョブまたは最近完了したジョブのみを一覧表示します。

```
GET /api/service/v2/services/cloud/jobs?start=0&limit=1
```

次の例では、実行中のすべてのジョブと、ジョブ索引番号が 4 以上の最近完了したジョブを、新しいジョブから順に一覧表示します。

```
GET /api/service/v2/services/cloud/jobs?start=4
```

ジョブの取り消しまたは再起動

次の例では、指定されたクラウドサービスジョブを取り消します。

リクエストの例:

```
PUT /api/service/v2/services/cloud/jobs/job-id/cancel HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
```

結果例:

```
HTTP/1.1 202 Accepted
Date: Wed, 24 Jul 2019 21:50:29 GMT
Content-Length: 0
X-Zfssa-Service-API: 2.0
X-Zfssa-API-Version: 2.0
Content-Type: application/json; charset=utf-8
```

次の例では、指定されたクラウドサービスジョブを再起動します。

リクエストの例:

```
PUT /api/service/v2/services/cloud/jobs/job-id/restart HTTP/1.1
Host: hostname:215
```

```
Authorization: Basic Tm8gcGVla2luZyE=
User-Agent: curl/7.54.0
Accept: */*
```

結果例:

```
HTTP/1.1 202 Accepted
Date: Wed, 24 Jul 2019 21:51:08 GMT
Content-Length: 0
X-Zfssa-Service-API: 2.0
X-Zfssa-API-Version: 2.0
Content-Type: application/json; charset=utf-8
```

スナップショットバックアップ操作

スナップショットバックアップ操作を使用して、Oracle Cloud Infrastructure Object Storage でのシェアスナップショットバックアップを管理します。スナップショットを作成するには、[185 ページの「スナップショットおよびクローンの操作」](#)を参照してください。

各スナップショットバックアップには一意の識別子が割り当てられているため、異なるローカルシステム上で同じ名前のスナップショットを同じクラウドターゲットにバックアップできます。

同じファイルシステムスナップショットを、2つの異なる形式の2つのクラウドバックアップに使用できます。

ローカルスナップショットをクラウドにバックアップした後、そのスナップショットを削除できます。ただし、親スナップショットになる可能性のあるローカルスナップショットは、将来の増分スナップショットのために保持しておいてください。

表 30 スナップショットバックアップのコマンド

リクエスト	次の後ろに追加:	説明
	<code>/api/storage/v2/pools/<i>pool</i>/projects/<i>project</i></code> 次のいずれかを加えます。 ■ <code>/filesystems/<i>fs</i></code> ■ <code>/luns/<i>lun</i></code>	
GET	<code>/snapshots/<i>snapshot</i>/backups</code>	任意の形式のすべてのスナップショットバックアップを一覧表示します。
GET	<code>/snapshots/<i>snapshot</i>/backups/<i>format</i>/<i>backup-id</i>/<i>target-id</i></code>	指定されたスナップショットバックアップを一覧表示します。
DELETE	<code>/snapshots/<i>snapshot</i>/backups/<i>format</i>/<i>backup-id</i>/<i>target-id</i></code>	指定されたスナップショットバックアップを削除します。
POST	<code>/snapshots/<i>snapshot</i>/backups</code>	新しいスナップショットバックアップを作成します。

スナップショットバックアップの一覧表示

次の例では、ファイルシステム `f-1` 上の任意の形式のスナップショット `snap0` のすべてのクラウドバックアップを一覧表示します。

リクエストの例:

```
GET /api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap0/backups
HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
User-Agent: curl/7.54.0
Accept: */*
```

結果例:

```
HTTP/1.1 200 OK
Date: Wed, 07 Jan 2020 20:54:47 GMT
Content-Length: 708
X-Zfssa-Storage-API: 2.0
Content-Type: application/json; charset=utf-8
X-Zfssa-API-Version: 2.0

{
  "backups": [{
    "finished": "2020-01-07T21:02:14Z",
    "format": "tar",
    "href": "/api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap0/backups/tar/backup-id/target-id",
    "id": "backup-id",
    "status": "completed",
    "target": "target-id",
    "targetName": "oci-ashburn"
  }]
}
```

次のリクエストでは、指定されたスナップショットバックアップを一覧表示します。

```
GET /api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap0/backups/format/backup-id/target-id HTTP/1.1
```

スナップショットバックアップの作成

次の例では、スナップショット `snap0` のバックアップを `tar` 形式で作成し、バックアップをターゲット `oci-phoenix` に保存します。形式が指定されていない場合、バックアップは `zfs` 形式で作成されます。スナップショットバックアップの進行状況を表示するには、[124 ページの「ジョブの一覧表示」](#)を使用して、返された `job-id` のジョブを表示します。

ターゲットに `writelimit` プロパティが設定されている場合、ターゲットに書き込まれるのは 1 秒あたり `writelimit` バイト以下です。[118 ページの「ターゲットの作成」](#)を参照してください。

リクエストの例:

```
POST /api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap0/backups
HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
{
  "target": "oci-phoenix",
  "format": "tar"
}
```

結果例:

```
{
  "action": "job-id"
}
```

増分スナップショットバックアップの作成

増分スナップショットバックアップを作成するには、`incremental` プロパティの値として `true` を指定し、比較に使用する親スナップショットを指定します。

- 親スナップショットは、指定された増分スナップショットバックアップと同じローカルシステムと同じクラウドターゲットの両方に存在する必要があります。次の例では、スナップショット `snap0` が、ローカルシステムとクラウドターゲットの両方のファイルシステム `f-1` に存在する必要があります。
- 親ファイルシステムと増分ファイルシステムのスナップショットは、`zfs` または `tar` の同じ形式である必要があります。

次のリクエストでは、ファイルシステム `f-1` のバックアップを作成します。これは、`snap0` とファイルシステム `f-1` の現在の状態との違いです。増分スナップショットバックアップ `snap1` は、ターゲット `oci-ashburn` に保存されます。増分スナップショットバックアップは、親スナップショットバックアップと同じ形式です。

リクエストの例:

```
POST /api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap1/backups
HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
{
  "target": "oci-ashburn",
  "incremental": true,
  "parent": "snap0"
}
```

結果例:

```
{
  "action": "job-id"
}
```

```
}

```

増分スナップショットバックアップの親の検出

次の例では、指定されたターゲット上の指定された増分スナップショットバックアップの親を識別します。この例では、ターゲット `oci-ashburn` 上のスナップショット `snap2` のバックアップの親は、スナップショット `snap1` および `snap0` です。結果は、このアプライアンスが `oci-ashburn` ターゲットと `oci-phoenix` ターゲットの両方にアクセスできることを示しています。`oci-ashburn` の結果がリクエストされているため、`oci-phoenix` の結果は表示されません。

リクエストの例:

```
POST /api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap2/backups?
props=true HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
```

```
{
  "target": "oci-ashburn"
}
```

結果例:

```
HTTP/1.1 200 OK
Date: Wed, 22 Jan 2020 22:02:17 GMT
Content-Length: 316
X-Zfssa-Storage-API: 2.0
Content-Type: application/json; charset=utf-8
X-Zfssa-API-Version: 2.0
```

```
{
  "props": [{
    "choices": [
      "oci-ashburn",
      "oci-phoenix"
    ],
    "data_type": "string",
    "label": "Backup target",
    "name": "target"
  }, {
    "choices": [
      "zfs",
      "tar"
    ],
    "data_type": "string",
    "label": "format",
    "name": "format"
  }, {
    "choices": [
      true,
      false
    ],
    "data_type": "boolean",
    "label": "Incremental",
    "name": "incremental"
  }, {
```

```

        "choices": [
            "snap1",
            "snap0"
        ],
        "data_type": "string",
        "label": "Parent",
        "name": "parent"
    }
}

```

次の例では、指定された増分スナップショットバックアップの親を識別するための代替方法として、POST の代わりに GET を使用しています。この形式では、ターゲットを指定する必要はありません。結果は、oci-phoenix ターゲット上に snap2 の親がないことを示しており、これは oci-phoenix 上に snap2 の増分バックアップを作成できないことを意味します。oci-phoenix 上に snap2 の完全バックアップを作成できます。

リクエストの例:

```

GET /api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap2/targets
HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*

```

結果例:

```

HTTP/1.1 200 OK
Date: Wed, 07 Jan 2020 22:04:08 GMT
Content-Length: 329
X-Zfssa-Storage-API: 2.0
Content-Type: application/json; charset=utf-8
X-Zfssa-API-Version: 2.0

{
  "targets": [{
    "format": "zfs",
    "href": "/api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap2/targets/zfs/target-id1",
    "id": "target-id1",
    "name": "oci-ashburn",
    "parents": [
      "snap0",
      "snap1"
    ]
  }, {
    "format": "tar",
    "href": "/api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap2/targets/tar/target-id1",
    "id": "target-id1",
    "name": "oci-ashburn",
    "parents": [
      "snap0",
      "snap1"
    ]
  }, {
    "id": "target-id2",
    "name": "oci-phoenix",
    "parents": [],
    "href": "/api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap2/targets/target-id2"
  }
]

```

```
}    }]  
}
```

次のリクエストを使用して、指定されたスナップショットバックアップの親を表示します。

```
GET /api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap2/  
targets/format/target-id1 HTTP/1.1
```

スナップショットバックアップの削除

次の例では、指定されたスナップショットバックアップを削除します。バックアップ削除の進行状況を表示するには、[124 ページの「ジョブの一覧表示」](#)を使用して、返された *job-id* のジョブを表示します。

リクエストの例:

```
DELETE /api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap0/  
backups/format/backup-id/target-id HTTP/1.1  
Host: hostname:215  
Authorization: Basic Tm8gcGVla2luZyE=  
Accept: */*
```

結果例:

```
{  
  "action": "job-id"  
}
```

RESTful API 問題サービス

RESTful API 問題サービスは、アプライアンス障害マネージャーによって検出された問題を表示および管理するために使用されます。

問題サービスコマンド

表 31 問題サービスコマンド

リクエスト	パス <code>/problem/v{1 2}</code> の後ろに追加	説明
GET	<code>/problem/v{1 2}</code> のみを使用	問題サービスコマンドを一覧表示します
GET	<code>/problems</code>	すべての現在の問題を一覧表示します
GET	<code>/problems/problem</code>	指定された <code>uuid</code> を持つ問題の詳細プロパティを取得します
PUT	<code>/problems/problem/markrepaired</code>	指定された問題 <code>uuid</code> を修復済みとしてマークします
GET	<code>/suspend_notification</code>	通知が一時停止されているかどうかを表示します
PUT	<code>/suspend_notification/enable</code>	通知を一時停止します
PUT	<code>/suspend_notification/disable</code>	通知を再開します

すべての問題の一覧表示

このコマンドはアプライアンスで現在アクティブなすべての問題を一覧表示します。コマンドに成功すると、HTTP ステータス 200 (OK) が返されます。

リクエストの例:

```
GET /api/problem/v1/problems HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "problems": [{
    "code": "AK-8003-Y6",
    "description": "The device configuration for JBOD
                  '1204FMD063' is invalid.",
    "impact": "The disks contained within the enclosure
              cannot be used as part of a storage pool.",
    "uuid": "0d30be41-b50d-4d03-ddb4-edb69ee080f8",
    "repairable": false,
    "type": "Defect",
    "timestamp": "2013-2-21 17:37:12",
    "severity": "Major",
    "components": [{
      "certainty": 100,
      "status": "degraded",
      "uuid": "b4fd328f-92d6-4f0e-fb86-e3967a5473e7",
      "chassis": "1204FMD063",
      "label": "hc:///chassis-mfg=SUN
                :chassis-name=SUN-Storage-J4410
                :chassis-part=unknown
                :chassis-serial=1204FMD063
                :fru-serial=1204FMD063
                :fru-part=7041262
                :fru-revision=3529/ses-enclosure=0",
      "revision": "3529",
      "part": "7041262",
      "model": "Sun Disk Shelf (SAS-2)",
      "serial": "1204FMD063",
      "manufacturer": "Sun Microsystems, Inc."
    }]
  }]
}
```

一つの問題の一覧表示

一つの問題の一覧表示コマンドは単一の問題を一覧表示します。コマンドに成功すると、HTTP ステータス 200 (OK) が返されます。

問題の一覧表示コマンドでは、1 つの問題の UUID である `uuid` 入力パラメータを使用します。

リクエストの例:

```
GET /api/problem/v1.0/problems/0d30be41-b50d-4d03-ddb4-edb69ee080f8
HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "problem": {
    "uuid": "0d30be41-b50d-4d03-ddb4-edb69ee080f8",
    ...
  }
}
```

```
}
```

問題の修復

問題の修復コマンドは、問題を修復済みとしてマークします。

問題の修復コマンドでは、修復済みとしてマークされる問題の UUID である `uuid` 入力パラメータを使用します。

リクエストの例:

```
PUT /api/problem/v1/problems/0d30be41-b50d-4d03-ddb4-edb69ee080f8/repaired
HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

成功した応答は HTTP ステータス 202 (Accepted) を返します。

```
HTTP/1.1 202 Accepted
```

問題通知の一時停止

アプライアンスを処理すると、誤った障害が発生することがあります。不要な SR の送信を回避するために、サービスを実行している間、すべての通知を一時停止できます。

問題通知が一時停止された場合の説明については、「[Suspending Problem Notification](#)」 in 『[Oracle ZFS Storage Appliance Customer Service Manual](#)』を参照してください。

通知の一時停止のステータスの表示

次のコマンドを使用して、問題の通知が一時停止されているかどうかを確認します。

リクエストの例:

```
GET /api/problem/v2/suspend_notification HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

`period` プロパティは、通知の一時停止の残り時間 (分) を示します。

```
{
  "suspend_notification": {
```

```
    "href": "/api/problem/v2/suspend_notification",  
    "suspend_notification": "enabled",  
    "period": 472  
  }  
}
```

通知の一時停止

次のコマンドを使用して、問題の通知を一時停止します。操作が成功すると、HTTP コード 202 (Accepted) を返します。

リクエストの例:

```
PUT /api/problem/v2/suspend_notification/enable HTTP/1.1  
Host: zfs-storage.example.com:215  
Accept: application/json
```

通知の再開

次のコマンドを使用して、問題の通知を再開します。操作が成功すると、HTTP コード 202 (Accepted) を返します。

リクエストの例:

```
PUT /api/problem/v2/suspend_notification/disable HTTP/1.1  
Host: zfs-storage.example.com:215  
Accept: application/json
```

RESTful API SAN サービス

RESTful API SAN サービスでは、ストレージエリアネットワーク (SAN) にアプライアンスを接続できます。

SAN の概要

SAN には次の基本的なコンポーネントがあります。

- ネットワークストレージにアクセスするクライアント
- ネットワークストレージを提供するストレージアプライアンス
- クライアントをストレージにリンクするネットワーク

これらの3つのコンポーネントは、ネットワーク上で使用されているプロトコルには関係なく同じです。場合によっては、ネットワークがイニシエータとターゲットの間のケーブルであることもありますが、ほとんどの場合は、何らかの種類の切り換えが必要になります。RESTful API SAN サービスは、サポートされているプロトコルごとに4つのタイプの SAN リソースを管理します。

- **イニシエータ** – SCSI セッションを開始し、SCSI コマンドと入出力リクエストを送信できるアプリケーションまたは本番システムエンドポイント。イニシエータは、一意のアドレス指定方式によっても識別されます。
- **イニシエータグループ** – イニシエータのセット。イニシエータグループが論理ユニット番号 (LUN) に関連付けられている場合は、そのグループのイニシエータだけがその LUN にアクセスできます。
- **ターゲット** – イニシエータからの SCSI コマンドと入出力リクエストを処理するサービスを提供するストレージシステムエンドポイント。ターゲットはストレージシステム管理者によって作成され、一意のアドレス指定方式によって識別されます。構成されたターゲットは、0 個以上の論理ユニットから構成されます。
- **ターゲットグループ** – ターゲットのセット。LUN は、1 つの特定のターゲットグループ内のすべてのターゲットにエクスポートされます。

SAN イニシエータ

SAN イニシエータの管理には、次のコマンドが使用されます。

これらのコマンドは次の URI パラメータを使用します。

protocol イニシエータ用の NAS プロトコル: *fc*、*iscsi*、または *srp*

initiator イニシエータの IQN、WWN、または EUI

表 32 イニシエータコマンド

リクエスト	パス <i>/san/v{1 2}</i> の後ろに追加	説明
GET	<i>/protocol/initiators</i>	指定されたプロトコル (<i>fc</i> 、 <i>iscsi</i> 、 <i>srp</i>) オブジェクトのすべての SAN イニシエータを一覧表示します
GET	<i>/protocol/initiators/initiator</i>	指定されたプロトコル (<i>fc</i> 、 <i>iscsi</i> 、 <i>srp</i>) プロパティの指定された SAN イニシエータを取得します
POST	<i>/protocol/initiators</i>	指定されたプロトコル (<i>fc</i> 、 <i>iscsi</i> 、 <i>srp</i>) で新しい SAN イニシエータを作成します
PUT	<i>/protocol/initiators/initiator</i>	指定されたプロトコル (<i>fc</i> 、 <i>iscsi</i> 、 <i>srp</i>) オブジェクトの指定された SAN イニシエータを変更します
DELETE	<i>/protocol/initiators/initiator</i>	指定されたイニシエータオブジェクトを破棄します

イニシエータコマンドの多くは、戻り値として次の表に示すプロパティを使用します。作成および変更コマンドも入力値としてプロパティを使用します。

表 33 イニシエータのプロパティ

プロパティ	プロトコル	説明
<i>alias</i>	<i>all</i>	このイニシエータの別名
<i>initiator</i>	<i>fc</i>	このイニシエータのポート World Wide Name (WWN)
<i>iqn</i>	<i>iscsi</i>	このイニシエータの iSCSI 修飾名
<i>chapuser</i>	<i>iscsi</i>	チャレンジハンドシェイク <i>auth</i> プロトコル (CHAP) ユーザー名
<i>chapsecret</i>	<i>iscsi</i>	チャレンジハンドシェイク <i>auth</i> プロトコル (CHAP) シークレット
<i>initiator</i>	<i>srp</i>	EUI (Extended Unique Identifier)

イニシエータの一覧表示

指定されたプロトコルタイプのアプライアンス上に構成されたすべてのイニシエータを一覧表示します。応答本文に、JSON 形式で「initiators」という名前のイニシエータプロパティの配列が含まれます。

iSCSI イニシエータを一覧表示するリクエストの例:

```
GET /api/san/v1/iscsi/initiators HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "initiators": [{
    "alias": "init-02",
    "href": "/api/san/v1/iscsi/initiators/iqn.zfs-storage.example.com.sun:02:02",
    "initiator": "iqn.zfs-storage.example.com.sun:02:02",
    "chapsecret": "",
    "chapuser": ""
  }, {
    "alias": "init-01",
    "initiator": "iqn.zfs-storage.example.com.sun:02:01",
    "href": "/api/san/v1/iscsi/initiators/iqn.zfs-storage.example.com.sun:02:01",
    "chapsecret": "",
    "chapuser": ""
  }
]}
```

イニシエータの詳細の取得

単一の iSCSI イニシエータの詳細を一覧表示します。応答本文に、JSON 形式で「initiator」という名前のオブジェクトとして iSCSI イニシエータプロパティが含まれます。

リクエストの例:

```
GET /api/san/v1/iscsi/initiators/iqn.zfs-storage.example.com.sun:02:01 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "initiator": {
    "alias": "init-01",
    "href": "/api/san/v1/iscsi/initiators/iqn.zfs-storage.example.com.sun:02:01"
```

```
    "initiator": "iqn.zfs-storage.example.com.sun:02:01",  
    "chapsecret": "",  
    "chapuser": ""  
  }  
}
```

イニシエータの作成

新しい iSCSI イニシエータを作成します。iSCSI 修飾名 (IQN) を指定する必要があります。リクエスト本文に JSON 形式で iSCSI イニシエータプロパティーが含まれます。応答には、成功時に HTTP ヘッダーに新しい iSCSI イニシエータの場所の URI とステータスコード 201 (Created) が含まれます。応答本文に、JSON 形式で「initiator」という名前のオブジェクトとして iSCSI イニシエータプロパティーが含まれます。

リクエストの例:

```
POST /api/san/v1.0/iscsi/initiators HTTP/1.1  
Host: zfs-storage.example.com:215  
Content-Type: application/json  
Accept: application/json  
  
{  
  "initiator": "iqn.zfs-storage.example.com.sun:02:02",  
  "alias": "init-02"  
}
```

結果例:

```
HTTP/1.1 201 Created  
Content-Type: application/json  
Content-Length: 181  
X-Zfssa-San-API: 1.0  
Location: /api/san/v1/iscsi/initiators/iqn.zfs-storage.example.com.sun:02:02  
  
{  
  "initiator": {  
    "alias": "init-02",  
    "href": "/api/san/v1/iscsi/initiators/iqn.zfs-storage.example.com.sun:02:02",  
    "initiator": "iqn.zfs-storage.example.com.sun:02:02",  
    "chapsecret": "",  
    "chapuser": ""  
  }  
}
```

イニシエータの変更

このコマンドは既存のイニシエータを変更します。リクエスト本文に JSON 形式で変更すべきイニシエータプロパティーが含まれます。イニシエータの IQN は URI で提供されます。成功時に HTTP ステータス 202 (Accepted) が返されます。応答本文に、JSON 形式で initiator という名前のオブジェクトとして、新しい iSCSI イニシエータプロパティーが含まれます。

リクエストの例:

```
PUT /api/san/v1/iscsi/initiators/iqn.zfs-storage.example.com.sun:01 HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json
```

```
{
  "alias": "init-01-secure",
  "chapuser": "admin4",
  "chapsecret": "secret"
}
```

結果例:

```
HTTP/1.1 202 Accepted
Content-Length: 167
Content-Type: application/json
X-Zfs-Sa-Nas-API: 1.0
```

```
{
  "initiator": {
    "alias": "init-01-secure",
    "href": "/api/san/v1/iscsi/initiators/iqn.zfs-storage.example.com.sun:01",
    "iqn": "iqn.zfs-storage.example.com.sun:1",
    "chapsecret": "secret",
    "chapuser": "admin4"
  }
}
```

イニシエータの削除

アプライアンスからイニシエータを削除します。

リクエストの例:

```
DELETE /api/san/v1/iscsi/initiators/iqn.zfs-storage.example.com.sun:01 HTTP/1.1
Host: zfs-storage.example.com:215
```

Delete が成功すると HTTP コード 204 (No Content) を返します。

```
HTTP/1.1 204 No-Content
```

イニシエータグループ

iSCSI イニシエータコマンドはアプライアンス上の iSCSI イニシエータと iSCSI イニシエータグループの管理に使用します。使用可能なコマンドを以下の表に示します。

これらのコマンドは次の URI パラメータを使用します。

protocol イニシエータ用の NAS プロトコル: fc、iscsi、または srp

name イニシエータグループの名前

各イニシエータグループには、*name* プロパティと、イニシエータグループ内のイニシエータのリストを含む *initiators* プロパティがあります。

表 34 イニシエータグループコマンド

リクエスト	パス <i>/san/v{1 2}</i> の後ろに追加	説明
GET	<i>/protocol/initiator-groups</i>	指定されたプロトコル (<i>fc</i> 、 <i>iscsi</i> 、 <i>srp</i>) オブジェクトのすべての SAN イニシエータグループを一覧表示します
GET	<i>/protocol/initiator-groups/name</i>	指定されたプロトコル (<i>fc</i> 、 <i>iscsi</i> 、 <i>srp</i>) プロパティの指定された SAN イニシエータグループを取得します
POST	<i>/protocol/initiator-groups</i>	指定されたプロトコル (<i>fc</i> 、 <i>iscsi</i> 、 <i>srp</i>) の新しい SAN イニシエータグループを作成します
PUT	<i>/protocol/initiator-groups/name</i>	指定されたプロトコル (<i>fc</i> 、 <i>iscsi</i> 、 <i>srp</i>) オブジェクトの指定された SAN イニシエータグループを変更します
DELETE	<i>/protocol/initiator-groups/name</i>	指定された名前オブジェクトを破棄します

イニシエータグループの一覧表示

使用可能なすべての iSCSI イニシエータグループを一覧表示します。成功時、HTTP ステータス 200 (OK) が返され、本文にイニシエータグループオブジェクトの配列を含む「*groups*」という名前のプロパティを持つ JSON オブジェクトが含まれます。

リクエストの例:

```
GET /api/san/v1/iscsi/initiator-groups HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "groups": [{
    "href": "/san/v1/iscsi/initiator-groups/p1-initiators-0",
    "initiators": ["iqn.zfs-storage.example.com.sun:0"],
    "name": "p1-initiators-0"
  }, {
    "href": "/san/v1/iscsi/initiator-groups/p1-initiators-1",
    "initiators": ["iqn.zfs-storage.example.com.sun:1"],
    "name": "p1-initiators-1"
  }]
}
```

イニシエータグループの詳細の取得

単一の iSCSI イニシエータグループから詳細情報を取得します。グループには `list initiator group` コマンドで返された `href` プロパティに従ってアクセスできます。

リクエストの例:

```
GET /api/san/v1/iscsi/initiator-groups/test-group HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "group": {
    "href": "/api/san/v1/iscsi/initiator-groups/test-group"
    "initiators": ["iqn.zfs-storage.example.com.sun:02:01"],
    "name": "test-group"
  }
}
```

イニシエータグループの作成

メンバーを含まない iSCSI イニシエータグループを作成します。リクエスト本文には、グループ名を含む単一の `name` パラメータを持つ JSON オブジェクトが含まれます。

表 35 イニシエータグループ作成プロパティ

プロパティ	型	説明
<code>name</code>	string	イニシエータグループの名前
<code>initiators</code>	配列	既存のイニシエータ IQN プロパティの配列

リクエストの例:

```
POST /api/san/v1/iscsi/initiator-groups HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Content-Length: 64
Accept: application/json
```

```
{
  "name": "group-01",
  "initiators": ["iqn.zfs-storage.example.com.sun:02"]
}
```

結果例:

```

HTTP/1.1 201 Created
Content-Type: application/json
Location: /api/san/v1/iscsi/initiator-groups/test-group

{
  "group": {
    "href": "/api/san/v1/iscsi/initiator-groups/test-group",
    "initiators": ["iqn.zfs-storage.example.com.sun:02"],
    "name": "group-01"
  }
}

```

イニシエータグループの削除

アプライアンスからイニシエータグループを削除します。

リクエストの例:

```

DELETE /api/san/v1.0/iscsi/initiator-groups/group-01 HTTP/1.1
Host: zfs-storage.example.com:215

```

削除に成功すると、HTTP ステータス 204 (No Content) が返されます。

```

HTTP/1.1 204 No-Content

```

ターゲット

iSCSI ターゲットコマンドは、iSCSI ターゲットおよび iSCSI ターゲットグループを管理するために使用します。使用可能なコマンドを以下の表に示します。

ターゲットコマンドは次の URI パラメータを取ります。

protocol SAN プロトコル: *fc*、*iscsi*、または *srp*

target ターゲット ID: IQN、WWN、EUI

表 36 ターゲットコマンド

リクエスト	パス <i>/san/v{1 2}</i> の後ろに追加	説明
GET	<i>/protocol/targets</i>	指定されたプロトコル (<i>fc</i> 、 <i>iscsi</i> 、 <i>srp</i>) オブジェクトのすべての SAN ターゲットを一覧表示します
GET	<i>/protocol/targets/target</i>	指定されたプロトコル (<i>fc</i> 、 <i>iscsi</i> 、 <i>srp</i>) プロパティの指定された SAN ターゲットを取得します
POST	<i>/protocol/targets</i>	指定されたプロトコル (<i>fc</i> 、 <i>iscsi</i> 、 <i>srp</i>) で新しい SAN ターゲットを作成します
PUT	<i>/protocol/targets/target</i>	指定されたプロトコル (<i>fc</i> 、 <i>iscsi</i> 、 <i>srp</i>) オブジェクトの指定された SAN ターゲットを変更します

リクエスト	パス <code>/san/v{1 2}</code> の後ろに追加	説明
DELETE	<code>/protocol/targets/target</code>	指定されたターゲットオブジェクトを破棄します

ターゲットの取得コマンドは、ターゲットプロパティを返します。ターゲットの作成および変更コマンドは、次の表に示すプロパティを入力として使用します。

表 37 ターゲットの入力プロパティ

プロパティ	プロトコル	説明
alias	iscsi	簡単な判読可能な名前
iqn	iscsi	iSCSI 修飾名
state	iscsi	iSCSI ターゲットの状態 (「online」、「offline」)
auth	iscsi	オプションの認証タイプ (「none」、「chap」)
targetchapuser	iscsi	オプションの CHAP ユーザー認証
targetchapsecret	iscsi	オプションの CHAP シークレット認証
interfaces	iscsi	ターゲットが使用可能なネットワークインタフェースのリスト
wwn	fc	このターゲットの World Wide Name
port	fc	ポートの物理的な場所
mode	fc	このポートのモード (イニシエータまたはターゲット)
speed	fc	このポートのネゴシエートされた速度
discovered_ports	fc	検出されたリモートイニシエータポートの数
alias	srp	SRP ターゲットの別名
eui	srp	このターゲットの EUI (Extended Unique Identifier)

次のプロパティは iSCSI ターゲットグループ情報を取得するために使用します。

表 38 ターゲットグループのプロパティ

プロパティ	型	説明
protocol	string	ターゲットグループプロトコル: FC、iSCSI、または SRP
name	string	iSCSI ターゲットグループ名
targets	配列	iSCSI ターゲット IQN グループメンバーのリスト

ターゲットの一覧表示

アプライアンスで利用できる指定されたプロトコルのすべての SAN ターゲットを一覧表示します。

リクエストの例:

```
GET /api/san/v1/iscsi/targets HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 1337

{
  "size": 7,
  "targets": [{
    "alias": "tst.volumes.py.12866.target",
    "href": "/api/san/v1/iscsi/targets/iqn.zfs-storage.example.com.sun:02:72b6fa9a-96c4-e511-db19-aadb9bac2052",
    "iqn": "iqn.zfs-storage.example.com.sun:02:72b6fa9a-96c4-e511-db19-aadb9bac2052",
    ...
  }, {
    "alias": "tst.volumes.py.96238.target",
    "href": "/api/san/v1/iscsi/targets/iqn.zfs-storage.example.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008",
    "iqn": "iqn.zfs-storage.example.com.sun:31d26d2e-6aa0-6054-fe58-8b1fb508b008",
    ...
  }
  ...]
}
```

ターゲットの詳細の取得

単一のターゲットからプロパティを取得します。ターゲットを選択するには、「iqn」プロパティを使用するか、「alias=*alias*」を使用します。

リクエストの例:

```
GET /api/san/v1/iscsi/targets/alias=test-target HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 251

{
  "target": {
    "alias": "test-target",
    "auth": "none",
    "href": "/api/san/v1/iscsi/targets/alias=test-target",
    "interfaces": ["ixgbe0"],
    "iqn": "iqn.zfs-storage.example.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008",
  }
}
```

```

        "targetchapsecret": "",
        "targetchapuser": ""
    }
}

```

ターゲットの作成

新しいターゲットを作成します。リクエスト本文には、新しい iSCSI ターゲットグループの名前である単一の `name` プロパティを持つ JSON オブジェクトが含まれます。

リクエストの例:

```

POST /api/san/v1/iscsi/targets HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Content-Length: 23
Accept: application/json

```

```

{"alias": "test-target"}

```

結果例:

```

HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 233
X-Zfssa-San-API: 1.0
Location: /api/san/v1/iscsi/targets/iqn.zfs-storage.example.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008

{
  "target": {
    "href": "/api/san/v1/iscsi/targets/iqn.zfs-storage.example.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008",
    "alias": "test-target",
    "iqn": "iqn.zfs-storage.example.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008",
    "auth": "none",
    "targetchapuser": "",
    "targetchapsecret": "",
    "interfaces": ["ixgbe0"]
  }
}

```

ターゲットの変更

既存の iSCSI ターゲットを変更します。リクエスト本文には、変更される iSCSI ターゲットプロパティを含む JSON オブジェクトが含まれます。成功時に HTTP ステータス 202 (Accepted) が返されます。応答本文には、JSON オブジェクトでエンコードされたターゲットの結果の iSCSI ターゲットプロパティが含まれます。

リクエストの例:

```
PUT /api/san/v1/iscsi/targets/alias=test-target HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Content-Length: 54
Accept: application/json
```

```
{"targetchapsecret": "secret", "auth": "chap",
 "targetchapuser": "admin5"}
```

結果例:

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 189
X-Zfssa-San-API: 1.0
```

```
{
  "target": {
    "href": "/api/san/v1/iscsi/targets/alias=test-target",
    "auth": "chap",
    "targetchapsecret": "secret",
    "alias": "test-target",
    "iqn": "iqn.zfs-storage.example.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008",
    "targetchapuser": "admin5",
    "interfaces": ["ixgbe0"]
  }
}
```

ターゲットの削除

システムから SAN ターゲットを削除します。

リクエストの例:

```
DELETE /api/san/v1/iscsi/targets/iqn.zfs-storage.example.com.sun:02:e7e688b1 HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
```

Delete が成功すると HTTP コード 204 (No Content) を返します。

```
HTTP/1.1 204 No-Content
```

ターゲットグループ

ターゲットグループはターゲットのコレクションです。ターゲットグループコマンドを以下の表に示します。

ターゲットグループコマンドは次の URI パラメータを取ります。

<i>protocol</i>	イニシエータ用の NAS プロトコル: fc、iscsi、または srp
-----------------	--------------------------------------

`target-group` ターゲットグループの名前

表 39 ターゲットグループのコマンド

リクエスト	パス <code>/san/v{1 2}</code> の後ろに追加	説明
GET	<code>/protocol/target-groups</code>	指定されたプロトコル (<code>fc</code> 、 <code>iscsi</code> 、 <code>srp</code>) オブジェクトのすべての SAN ターゲットグループを一覧表示します
GET	<code>/protocol/target-groups/target-group</code>	指定されたプロトコル (<code>fc</code> 、 <code>iscsi</code> 、 <code>srp</code>) プロパティの指定された SAN ターゲットグループを取得します
POST	<code>/protocol/target-groups</code>	指定されたプロトコル (<code>fc</code> 、 <code>iscsi</code> 、 <code>srp</code>) の新しい SAN ターゲットグループを作成します
PUT	<code>/protocol/target-groups/target-group</code>	指定されたプロトコル (<code>fc</code> 、 <code>iscsi</code> 、 <code>srp</code>) オブジェクトの指定された SAN ターゲットグループを変更します
DELETE	<code>/protocol/target-groups/target-group</code>	指定された <code>target-group</code> オブジェクトを破棄します

ターゲットグループの一覧表示

アプライアンスで使用可能なすべてのターゲットグループを一覧表示します。成功時、HTTP ステータス 200 (OK) が返され、本文にターゲットグループオブジェクトの配列を含む `groups` という名前のプロパティを持つ JSON オブジェクトが含まれます。

リクエストの例:

```
GET /api/san/v1/iscsi/target-groups
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 237

{
  "groups": [{
    "href": "/api/san/v1/iscsi/target-groups/test-group",
    "name": "test-group",
    "targets": [
      "iqn.zfs-storage.example.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008"
    ]
  }, {
    "href": "/api/san/v1/iscsi/target-groups/alt-group",
    ...
  }]
}
```

ターゲットグループの取得

単一のターゲットグループを取得します。リクエストは、ターゲットグループ名である単一の URI パラメータを取ります。応答本文には、ターゲットグループプロパティを含む group という名前の JSON オブジェクトプロパティが含まれます。

リクエストの例:

```
GET /api/san/v1/iscsi/target-groups/test-group
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "group": {
    "href": "/api/san/v1/iscsi/target-groups/test-group",
    "name": "test-group",
    "targets": [
      "iqn.zfs-storage.example.com.sun:02:0d5a0ed8-44b6-49f8-a594-872bf787ca5a"
    ]
  }
}
```

ターゲットグループの作成

新しい iSCSI ターゲットグループを作成します。リクエスト本文は、新しいグループの名前である単一の name プロパティを持つ JSON オブジェクトです。

リクエストの例:

```
POST /api/san/v1/iscsi/target-groups HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
Content-Type: application/json
Content-Length: 97

{"name": "test-group",
 "targets": ["iqn.zfs-storage.example.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008"]}
```

結果例:

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 154
X-Zfssa-San-API: 1.0
Location: /api/san/v1/iscsi/target-groups/test-group

{
  "group": {
    "href": "/api/san/v1/iscsi/target-groups/test-group",
```

```
    "name": "test-group",  
    "targets": [  
      "iqn.zfs-storage.example.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008"  
    ]  
  }  
}
```

ターゲットグループの削除

既存のターゲットグループを削除します。

リクエストの例:

```
DELETE /api/san/v1.0/iscsi/target-groups/test-group
```

削除に成功すると、HTTP ステータス 204 (No Content) が返されます。

```
HTTP/1.1 204 No-Content
```


サービスコマンド

サービス RESTful API はアプライアンスで実行しているソフトウェアサービスを一覧表示し、管理するために使用します。

サービスコマンド

次のサービスコマンドを使用できます。

表 40 サービスコマンド

リクエスト	パス <code>/service/v{1 2}</code> の後ろに追加	説明
GET	<code>/service/v{1 2}</code> のみを使用	サービスコマンドを一覧表示します
GET	<code>/services</code>	すべてのサービスを一覧表示します
GET	<code>/services/service</code>	指定されたサービスの構成およびステータスを取得します
PUT	<code>/services/service</code>	指定されたサービスの構成を変更します
PUT	<code>/services/service/enable</code>	指定されたサービスを有効にします
PUT	<code>/services/service/disable</code>	指定されたサービスを無効にします

サービスの一覧表示

このコマンドは、ストレージアプライアンス上で使用可能な構成可能なサービスのリストをそれらの有効のステータスとともに返します。成功したコマンドに対し、HTTP ステータス 200 (OK) が返されます。

リクエストの例:

```
GET /api/service/v1/services HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

結果の例。簡潔にするために、エントリのほとんどを省略しています。

```
HTTP/1.1 200 OK
```

```

Content-Type: application/json; charset=utf-8
Transfer-Encoding: chunked
X-Zfssa-Service-API: 1.0

{
  "services": [
    {
      "<status>": "disabled",
      "href": "/api/service/v1/services/ad",
      "name": "ad",
      "log": {
        "href": "/api/log/v1/logs/appliance-kit-adstat:default",
        "size": 2
      }
    },
    {
      "<status>": "online",
      "href": "/api/service/v1/services/nfs",
      "name": "nfs",
      "log": {
        "href": "/api/log/v1/logs/appliance-kit-nfsconf:default",
        "size": 8
      }
    },
    {
      "<status>": "online",
      "href": "/api/service/v1/services/ssh",
      "name": "ssh",
      "log": {
        "href": "/api/log/v1/logs/network-ssh:default",
        "size": 134
      }
    }
  ]
}

```

サービスの取得

このコマンドは、単一のサービスからその状態と構成を含む詳細を取得します。

リクエストの例:

```

GET /api/service/v1/services/ndmp HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json

```

結果例:

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "service": {
    "cram_md5_password": "",
    "cram_md5_username": "",
    "dar_support": true,
    "default_pools": [],
    "drive_type": "sysv",
    "href": "/api/service/v1/services/ndmp",

```

```

        "ignore_ctime": false,
        "name": "ndmp",
        "restore_fullpath": false,
        "status": "online",
        "tcp_port": 10000,
        "version": 4,
        "zfs_force_override": "off",
        "zfs_token_support": false
    }
}

```

サービスの状態の変更

このコマンドは指定されたサービスの状態を変更します。次の URI パラメータを使用します。

<i>service</i>	サービスの名前
<i>state</i>	新しいサービスの状態: enable または disable

リクエストの例:

```

PUT /api/service/v1/services/replication/enable HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json

```

成功した応答は HTTP ステータス 202 (Accepted) を返します。

サービスに JSON リクエストを送信して、サービスを有効または無効にすることもできます。

JSON を使用したリクエストの例:

```

PUT /api/service/v1/services/replication HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 22

```

```

{"<status>": "enable"}

```

サービスを無効にするには、次の JSON を送信します。

```

{"<status>": "disable"}

```

サービス構成の変更

指定されたサービスの構成プロパティを変更するには、ヘッダーに新しいプロパティ値を定義して、PUT リクエストを送信します。一部のサービスにはサブリソー

スがあることがあり、それらもサブリソースに定義された href に従って変更できます。成功した応答は HTTP ステータス 202 (Accepted) を返します。

次の例では、LDAP サービスのサーバーのリストを並べ替えて、サーバーのリストが優先順位の高い順になるように指定しています。LDAP 構成の詳細は、[『Oracle ZFS Storage Appliance 管理ガイド、Release OS8.8.x』](#) の「LDAP 構成」を参照してください。

現在の状態:

```
GET /api/service/v1/services/ldap HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json

HTTP/1.1 200 OK
Content-Type: application/json

{
  "service": {
    "href": "/api/service/v1/services/ldap",
    "<status>": "online",
    "servers": [
      "ldap-server2.us.example.com:484",
      "ldap-server1.us.example.com:636"
    ],
    "use_server_order": false,
    "proxy_dn": "",
    "proxy_password": false,
    "base_dn": "dc=us,dc=oracle,dc=com",
    "search_scope": "one",
    "cred_level": "proxy",
    "auth_method": "simple",
    "use_tls": false,
    "user_search": [
    ],
    "user_mapattr": [
    ],
    "user_mapobjclass": [
    ],
    "group_search": [
    ],
    "group_mapattr": [
    ],
    "group_mapobjclass": [
    ],
    "netgroup_search": [
    ],
    "netgroup_mapattr": [
    ],
    "netgroup_mapobjclass": [
    ],
    "server-000": {
      "host": "ldap-server2.us.example.com",
      "port": 484,
      "status": "online",
      "last_seen": "142s",
      "rtt": "70.285ms",
      "err_msg": "",
      "href": "/api/service/v1/services/ldap/server-000"
    },
    "server-001": {
```

```

        "host": "ldap-server1.us.example.com",
        "port": 636,
        "status": "online",
        "last_seen": "142s",
        "rtt": "126.013ms",
        "err_msg": "",
        "href": "/api/service/v1/services/ldap/server-001"
    }
}

```

リクエストの例:

```

PUT api/service/v1/services/ldap HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json

```

```

{
  "servers": ["ldap-server1.us.example.com:636", "ldap-server2.us.example.com:484"],
  "use_server_order": true
}

```

結果例:

```

HTTP/1.1 202 Accepted
Content-Length: 1295
Content-Type: application/json; charset=utf-8
X-Zfssa-Service-API: 1.0

```

```

{
  "service": {
    "href": "/api/service/v1/services/ldap",
    "<status>": "online",
    "servers": [
      "ldap-server1.us.example.com:636",
      "ldap-server2.us.example.com:484"
    ],
    "use_server_order": true,
    "proxy_dn": "",
    "proxy_password": false,
    "base_dn": "dc=us,dc=oracle,dc=com",
    "search_scope": "one",
    "cred_level": "proxy",
    "auth_method": "simple",
    "use_tls": false,
    "user_search": [
    ],
    "user_mapattr": [
    ],
    "user_mapobjclass": [
    ],
    "group_search": [
    ],
    "group_mapattr": [
    ],
    "group_mapobjclass": [
    ],
    "netgroup_search": [
    ],
    "netgroup_mapattr": [
    ],
    "netgroup_mapobjclass": [
    ],
    "server-000": {

```

```

    "host": "ldap-server1.us.example.com",
    "port": 636,
    "status": "online",
    "last_seen": "142s",
    "rtt": "126.013ms",
    "err_msg": "",
    "href": "/api/service/v1/services/ldap/server-000"
  },
  "server-001": {
    "host": "ldap-server2.us.example.com",
    "port": 484,
    "status": "online",
    "last_seen": "142s",
    "rtt": "70.285ms",
    "err_msg": "",
    "href": "/api/service/v1/services/ldap/server-001"
  }
}

```

サービスリソース

一部のサービスにはサブリソースがあります。使用可能なサブリソースを確認するには、各サービスまたはサービスのリストのコマンドで返されるデータを参照してください。

表 41 サービスサブリソースのコマンド

リクエスト	パス	説明
GET	<i>/services/service/resource</i>	サービスのサブリソースを一覧表示します
PUT	<i>/services/service/resource/href</i>	サブリソースを変更します
POST	<i>/services/service/resource</i>	新しいサブリソースを作成します
DELETE	<i>/services/service/resource/href</i>	サブリソースを破棄します

これらの各コマンドは、ほかの RESTful API コマンドと同じパターンに従います。ここで GET は指定されたサブリソースのタイプを一覧表示または取得するために使用し、POST は新しいサブリソースタイプを作成するために使用し、PUT はサブリソースを変更するために使用し、DELETE は指定されたサブリソースを破棄するために使用します。

サブリソースと、各サブリソースで使用可能なプロパティおよびコマンドのリストについては、CLI「構成サービス」のドキュメントを参照してください。

RESTful API ストレージサービス

RESTful API ストレージサービスは、ストレージプール、プロジェクト、ファイルシステム、および LUN についての構成の表示および管理に使用します。また、スナップショットやレプリケーションも管理します。

ストレージプールの操作

Oracle ZFS Storage Appliance では、NAS はすべての LUN およびファイルシステム間での同じデータ冗長性特性を特徴とするプール内に構成され、プール操作は、アプライアンスストレージ構成を取得するために使用されます。

表 42 ストレージプールコマンド

リクエスト	パス <code>/api/storage/v{1 2}</code> の後ろに追加	説明
GET	<code>/pools</code>	すべてのストレージプールを一覧表示します
GET	<code>/pools/pool</code>	ストレージプールの詳細を取得します
POST	<code>/pools</code>	新しいストレージプールを構成します
PUT	<code>/pools/pool</code>	プールに対してストレージを追加または削除します。
PUT	<code>/pools/pool/scrub</code>	指定されたプールでデータのスクラブを開始します
DELETE	<code>/pools/pool/scrub</code>	指定されたプールでデータのスクラブジョブを停止します
DELETE	<code>/pools/pool</code>	指定されたストレージプールを構成解除します

プールの一覧表示

このコマンドは、システム上のすべてのストレージプールのプロパティを一覧表示します。成功したコマンドに対し、HTTP ステータス 200 (OK) が返されます。HTTP 本文には、JSON オブジェクトのリストが含まれ、各プールについて説明しています。次の表に、プロパティ名を示します。

注記 - depth 問合せパラメータおよび match_property-name=value 問合せパラメータはサポートされていません。

表 43 ストレージプールのプロパティ

プロパティ	型	説明
asn	string	プールを所有するアプライアンスのシリアル番号
name	string	ターゲットのプール名
owner	string	プールを所有するシステムのホスト名
peer	string	クラスタ化されたシステムでの、アプライアンスクラスタのパートナーヘッド
profile	string	データデバイスのプロファイル
scrub_schedule	string	プールの定期スクラブ操作間の日数、またはプールの定期スクラブを無効にします。許可される値、および他のプールのスクラブのプロパティについては、 164 ページの「プールのスクラブ」 を参照してください。
state	string	プールの状態: online、offline、exported

リクエストの例:

```
GET /api/storage/v1/pools HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "pools": [
    {
      "profile": "mirror3",
      "name": "m1",
      "peer": "peer-hostname",
      "state": "online",
      "owner": "system-hostname",
      "asn": "appliance-serial-number",
      "scrub_schedule": "30 days"
    },
    {
      "profile": "raidz1",
      "name": "r1",
      "peer": "peer-hostname",
      "state": "online",
      "owner": "system-hostname",
      "asn": "appliance-serial-number",
      "scrub_schedule": "30 days"
    }
  ]
}
```

プールの取得

このコマンドは、単一のストレージプールからのプロパティをプールのストレージ利用状況の情報とともに返します。成功したコマンドに対し、HTTP ステータス 200 (OK) が返されます。

リクエストの例:

```
GET /api/storage/v1/pools/p1 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "pool": {
    "profile": "raidz1",
    "name": "p1",
    "usage": {
      "available": 57454799311352.0,
      "compression": 1.0,
      "dedupratio": 672791,
      "free": 57454799311352.0,
      "total": 74732430950400.0,
      "usage_child_reservation": 0.0,
      "usage_data": 16011663438848.0,
      "usage metasize": 0.0,
      "usage metaused": 0.0,
      "usage_replication": 1693675705344.0,
      "usage_reservation": 0.0,
      "usage_snapshots": 123913627136.0,
      "usage_total": 17829252771328.0,
      "used": 17829252771328.0
    },
    "peer": "00000000-0000-0000-0000-000000000000",
    "state": "online",
    "owner": "admin1",
    "asn": "2f4aeeb3-b670-ee53-e0a7-d8e0ae410749"
  }
}
```

プールの構成

プールを構成します。プールの作成に必要なパラメータについては、CLI の構成ストレージコマンドを参照してください。使用可能なプロパティ名および値が返される、プール作成のためのドライランリクエストを実行できます。これを行うには、props 問合せパラメータのプロパティを true に設定します。

リクエストの例:

```
POST /api/storage/v1/pools?props=true HTTP/1.1
Host: zfs-storage.example.com:215
```

```
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Accept: application/json
```

```
{
  "name": "p1",
}
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
"props": [{
  "choices": ["custom"],
  "label": "Chassis 0",
  "name": "0",
  "type": "ChooseOne"
}, {
  "choices": ["custom"],
  "label": "Chassis 1",
  "name": "1",
  "type": "ChooseOne"
}, {
  "choices": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12],
  "label": "Chassis 1 data",
  "name": "1-data",
  "type": "ChooseOne"
}, {
  "choices": ["mirror", "mirror3", "raidz1",
    "raidz2", "raidz3_max", "stripe"],
  "label": "Data Profile",
  "name": "profile",
  "type": "ChooseOne"
}]
}
```

リクエストの例 (シャーシ [1] から 8 台のディスクを使用するプールを作成する):

```
POST /api/storage/v1/pools HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Accept: application/json
```

```
{
  "name": "p1",
  "profile": "stripe",
  "1-data": 8
}
```

結果例:

```
HTTP/1.1 201 Created
Content-Type: application/json
```

```
{
  "pool": {
    "asn": "314d252e-c42b-e844-dab1-a3bca680b563",
    "errors": [],
    "name": "p1",
    "owner": "zfs-storage",
    "peer": "00000000-0000-0000-0000-000000000000",
    "profile": "stripe",
  }
}
```

```

    "status": "online",
    "usage": {
      "available": 1194000466944.0,
      "dedupratio": 100,
      "total": 1194000908288.0,
      "used": 441344.0
    }
  }
}

```

プールへのストレージの追加

このコマンドは、プールの作成または構成に類似しています。ストレージの追加により、既存のプールにストレージデバイスがさらに追加されます。プールに追加する必要のあるストレージデバイスの数を含む本文と一緒に、`href pool/add` を送信します。

リクエストの例:

```

PUT /api/storage/v1/pools/p1/add HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Accept: application/json

```

```

{
  "2-data": 8
}

```

結果例:

```

HTTP/1.1 202 Accepted

```

プールからのストレージの削除

このコマンドは、プールにストレージを追加するコマンドに似ています。ストレージを削除すると、キャッシュおよびログストレージデバイスが既存のプールから削除されます。目的のタイプ、シャース番号、およびプールから削除するストレージデバイスの数を含む本文と一緒に、`href pool/remove` を送信します。

リクエストの例:

```

PUT /api/storage/v1/pools/p1/remove HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Accept: application/json

```

```

{
  "0-cache" : 2
}

```

結果例:

```

HTTP/1.1 202 Accepted

```

削除可能なデバイスの数を表示するには、`props` 問合せパラメータを `true` に設定します。

リクエストの例:

```
PUT /api/storage/v1/pools/p1/remove?props=true HTTP/1.1
Host: zfs-storage.example.com
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "props": [
    {
      "choices": [
        "0",
        "1",
        "2"
      ],
      "type": "ChooseOne",
      "name": "0-cache",
      "label": "Chassis 0 cache"
    },
    {
      "choices": [
        "0",
        "1",
        "2"
      ],
      "type": "ChooseOne",
      "name": "1-log",
      "label": "Chassis 1 log"
    }
  ]
}
```

プールのスクラブ

`pool/scrub` PUT リクエストを送信すると、プールスクラブ操作が開始されます。`pool/scrub` DELETE リクエストを送信すると、実行中のスクラブ操作が停止します。プールのスクラブの詳細は、[『Oracle ZFS Storage Appliance 管理ガイド、Release OS8.8.x』の「ストレージプールのスクラブ - 手動 \(CLI\)」](#)を参照してください。

デフォルトでは、ストレージプールの定期スクラブは有効化され、30 日ごとの頻度に設定されます。`scrub_schedule` プロパティは、プールの定期スクラブ操作間の日数を指定したり、プールの定期スクラブを無効にしたりします。`scrub_schedule` のデフォルト値は 30 です。

- 定期スクラブを無効にするには、たとえば手動スクラブを実行する場合は、`scrub_schedule` プロパティの値を `off` に設定します。

- 定期スクラブ操作間の日数を変更するには、`scrub_schedule` プロパティの値を 15、30、45、60、75、または 90 に設定します。

プールの定期スクラブの詳細は、『[Oracle ZFS Storage Appliance 管理ガイド、Release OS8.8.x](#)』の「[ストレージプールのスクラブ - 定期 \(CLI\)](#)」を参照してください。

`scrub` オブジェクトは、定期または手動による最新のプールスクラブについて報告します。

- `complete` プロパティが `false` の場合、スクラブはまだ実行中です。検出されたエラーの数 (`errors`) および修復されたエラーの数 (`repaired`) が表示されます。
- `complete` プロパティが `true` の場合、スクラブは終了しています。表示される追加情報には、スクラブが開始された時間 (`op_start`) とスクラブが終了した時間 (`last_end`) があります。時間は GMT です。

`scrub_started` および `scrub_finished` は、`zfs_pool` アラートアクションイベントカテゴリのイベントであり、これらのイベントにカスタムアクションを指定できることに注意してください。[51 ページの「RESTful API 警告サービス」](#)を参照してください。

次の例は、スクラブ後のプールリストの一部を示しています。

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "pool": {
    "status": "online",
    "profile": "mirror:log_stripe:cache_stripe",
    "scrub": {
      "errors": 0,
      "op_start": "20190520T16:09:41",
      "complete": true,
      "seq_resilver": 0,
      "type": "everything",
      "examined": 403968,
      "repaired": 0,
      "last_end": "20190520T16:17:59"
    },
    "scrub_schedule": "30 days",
    "name": "p0",
    "peer": "peer-hostname",
    "owner": "system-hostname",
    "asn": "appliance-serial-number"
  }
}
```

プールの構成解除

このコマンドはシステムからプールを削除します。

プールを削除するリクエスト:

```
DELETE /api/storage/v1/pools/p1 HTTP/1.1
```

```
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
```

結果例:

```
HTTP/1.0 204 No Content
Date: Fri, 02 Aug 2013 22:31:06 GMT
X-Zfssa-Nas-API: 1.0
Content-Length: 0
```

プロジェクトの操作

プロジェクトの操作はすべて、特定のプールヘスコープ指定できます。すべてのプロジェクトを対象に操作するコマンドでは URI の後ろに `/projects` を追加し、1 つのプロジェクトを対象に操作するコマンドでは `/projects/project` を追加します。

表 44 プロジェクトコマンド

リクエスト	パス <code>/api/storage/v{1 2}</code> の後ろに追加	説明
GET	<code>/projects</code>	すべてのプロジェクトを一覧表示します
GET	<code>/pools/pool/projects</code>	プロジェクトを一覧表示します
GET	<code>/pools/pool/projects?snaps=true</code>	スナップショットを含む、すべてのプロジェクトを一覧表示します
GET	<code>/pools/pool/projects/project</code>	プロジェクトの詳細を取得します
POST	<code>/pools/pool/projects</code>	プロジェクトを作成します
PUT	<code>/pools/pool/projects/project</code>	プロジェクトを変更します
DELETE	<code>/pools/pool/projects/project</code>	プロジェクトを破棄します
GET	<code>/pools/pool/projects/project/usage/groups</code>	プロジェクトグループの使用状況を取得します
GET	<code>/pools/pool/projects/project/usage/groups/group</code>	指定されたグループのプロジェクトの使用状況を取得します
GET	<code>/pools/pool/projects/project/usage/users</code>	プロジェクトユーザーの使用状況を取得します
GET	<code>/pools/pool/projects/project/usage/users/user</code>	指定されたユーザーのプロジェクトの使用状況を取得します

次の表に、プロジェクトのリソース内で編集可能なプロパティのリストを示します。

表 45 プロジェクトのプロパティ

プロパティ	型	説明
<code>aclinherit</code>	string	ACL の継承動作 (「discard」、「noallow」、「restricted」、「passthrough」、「passthrough-x」、「passthrough-mode-preserve」)

プロパティ	型	説明
aclmode	string	モード変更での ACL 動作 (「discard」、 「mask」、「passthrough」)
atime	boolean	読み取りフラグでのアクセス時間を更新します
canonical_name	string	正規名
checksum	string	ブロックチェックサム (「fletcher2」、 「fletcher4」、「sha256」)
compression	string	データ圧縮設定 (「off」、「lzb」、 「gzip-2」、 「gzip」、 「gzip-9」)
copies	number	追加のレプリケーションコピーの数
creation	datetime	プロジェクト (または LUN、ファイルシステム) 作 成の日付と時間
dedup	boolean	データ複製解除フラグ
default_group	string	プロジェクトのデフォルトのファイルシステムグ ループ: 「other」
default_permissions	string	プロジェクトのデフォルトのファイルシステム権限 「700」
default_sparse	boolean	プロジェクトのデフォルトの LUN スパースデータ フラグ
default_user	string	プロジェクトのデフォルトのファイルシステムユー ザー: 「nobody」
default_volblocksize	number	プロジェクトのデフォルトの LUN ブロックサイズ: 8192
default_volsize	number	プロジェクトのデフォルトの LUN サイズ
exported	boolean	エクスポート済みフラグ
logbias	string	同期書き込みバイアス (「latency」、 「throughput」)
mountpoint	string	シェアマウントポイントのデフォルト 「/export/ proj-01」
name	string	プロジェクト名
nbmand	boolean	非ブロックの必須ロックフラグ
nodestroy	boolean	破棄防止フラグ
quota	number	プロジェクトの割り当て制限サイズ (バイト)
origin	string	クローン元
pool	string	プール名
readonly	boolean	true に設定されている場合データは読み取り専用で す
recordsize	string	データベースのレコードサイズ 「128k」
reservation	number	データ予約サイズ
rstchown	boolean	所有権の変更制限フラグ
secondarycache	string	セカンダリキャッシュの使用状況 (「all」、 「metadata」、 「none」)

プロパティ	型	説明
sharedav	string	HTTP シェア (「off」、「rw」、「ro」)
shareftp	string	FTP シェア (「off」、「rw」、「ro」)
sharenfs	string	NFS シェア (「off」、「on」、「ro」、「rw」)
sharesftp	string	SFTP シェア (「off」、「rw」、「ro」)
sharesmb	string	SMB/CIFS シェア (「off」、「rw」、「ro」)
sharetftp	string	TFTP シェア (「off」、「rw」、「ro」)
snapdir	string	.zfs/snapshot の可視性 (「hidden」、「visible」)
snaplabel	string	定期スナップショットラベル
vscan	boolean	ウイルススキャンフラグ

プロジェクトの一覧表示

このコマンドを実行すると、指定のプール内のすべてのプロジェクトが表示されます。リクエストは、ストレージプール名である単一の URI パラメータを取ります。返されるプロジェクトのそれぞれには、上記で示された変更可能なプロパティのリストのほかに、プール名、作成時間、ロード状態、レプリケーションアクション、データ使用状況も含まれます。

注記 - depth 問合せパラメータおよび match_property-name=value 問合せパラメータはサポートされていません。

リクエストパラメータ: **filter** – プロジェクト内のプロパティがその値内に同じフィルタ文字列を含んでいることを必要とする簡易的な文字列照合フィルタ。

リクエストの例:

```
GET /api/storage/v1/pools/p1/projects HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

get が成功すると、HTTP コード 200 (OK) が JSON 形式のプロジェクトプロパティの配列とともに返されます。

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "projects": [{
    "name": "proj-01",
    ...
  }, {
    "name": "proj-02",
    ...
  }
}]
```

すべてのプール全体のすべてのプロジェクトのリストもサポートされます。URI には /projects パスのみが含まれます。

プロパティーの一部として backup を持つすべてのプロジェクトを取得するリクエストの例:

```
GET /projects?filter=backup HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

プロジェクトのプロパティーの取得

このコマンドを実行すると、指定されたプール内の単一のプロジェクトのプロパティーが一覧表示されます。get が成功すると、HTTP コード 200 (OK) が JSON 形式のプロジェクトプロパティーとともに返されます。

zfs-storage-1 プール内の proj-01 というプロジェクトを一覧表示するリクエストの例:

```
GET /api/storage/v1/pools/p1/projects/proj-01 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "project": {
    "default_volblocksize": 8192.0,
    "logbias": "latency",
    "creation": "20130411T20:02:35",
    "nodeestroy": false,
    "dedup": false,
    "sharenfs": "on",
    "sharesmb": "off",
    "default_permissions": "700",
    "mountpoint": "/export",
    "snaplabel": "",
    "id": "042919bb-0882-d903-0000-000000000000",
    "readonly": false,
    "rrsrc_actions": [],
    "compression": "off",
    "shareftp": "",
    "default_sparse": false,
    "snapdir": "hidden",
    "aclmode": "discard",
    "copies": 1,
    "aclinherit": "restricted",
    "shareftp": "",
    "canonical_name": "zfs-storage-1/local/default",
    "recordsize": 131072.0,
    "usage": {
      "available": 1758424767306.0,
      "loading": false,
      "quota": 0.0,

```

```
        "snapshots": 0.0,  
        "compressratio": 100.0,  
        "child_reservation": 0.0,  
        "reservation": 0.0,  
        "total": 45960.0,  
        "data": 45960.0  
    },  
    "default_volsize": 0.0,  
    "secondarycache": "all",  
    "collection": "local",  
    "exported": true,  
    "vscan": false,  
    "reservation": 0.0,  
    "atime": true,  
    "pool": "p1",  
    "default_user": "nobody",  
    "name": "default",  
    "checksum": "fletcher4",  
    "default_group": "other",  
    "sharesftp": "",  
    "nbmand": false,  
    "sharedav": "",  
    "rstchown": true  
  }  
}
```

プロジェクトの作成

プロジェクトの作成コマンドを実行すると、指定されたストレージプールに指定された名前のプロジェクトが作成されます。リクエストは、ストレージプール名である単一の URI パラメータを取ります。デフォルトのプロパティーを持つ新しいプロジェクトが返されます。

JSON の本文リクエストパラメータ:

- 名前 – プロジェクトを作成するには、プロジェクト名を指定する必要があります。
- プロジェクトプロパティー – 任意のプロジェクトプロパティーを新しいプロジェクトの初期値として設定できます。

proj-01 という名前のプロジェクトを作成するリクエストの例:

```
POST /api/storage/v1/pools/p1/projects HTTP/1.1  
Host: zfs-storage.example.com:215  
Content-Type: application/json  
Accept: application/json  
  
{  
  "name": "proj-01",  
  "sharenfs": "ro"  
}
```

作成に成功すると、新しいプロジェクトの URI を含むロケーションヘッダーとともに HTTP ステータス 201 (Created) が返されます。本文には、JSON 形式のプロジェクトプロパティーがすべて含まれます。

結果例:

```

HTTP/1.1 201 Created
Content-Type: application/json
Location: http://zfs-storage.example.com:215
          /pools/p1/projects/proj-01

{
  "project": {
    "name": "proj-01",
    "href": "/api/storage/v1/pools/p1/projects/proj-01",
    "mountpoint": "/export/acme/zfs-storage-1",
    ...
  }
}

```

プロジェクトの変更

プロジェクトの変更コマンドは、既存のプロジェクトの属性を変更します。次の URI パラメータを使用します。

pool ストレージプール名

project プロジェクト名

リクエストパラメータ: プロジェクトプロパティ – 任意のプロジェクトプロパティを新しいプロジェクトの初期値として設定できます。

プロジェクト名を **proj-01** から **new-name** に変更するリクエストの例:

```

POST /api/storage/v1/pools/p1/projects/proj-01 HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json

```

```

{
  "name": "new-name",
  "sharenfs": "rw",
  "compression": "gzip-9"
}

```

応答に成功すると、HTTP ステータス 202 (Accepted) が返され、すべてのプロジェクトプロパティが一覧表示されます。

結果例:

```

HTTP/1.1 201 Created
Content-Type: application/json
Location: /api/storage/v1/pools/p1/projects/new-name

```

```

{
  "project": {
    "name": "new-name",
    "sharenfs": "rw",
    "compression": "gzip-9",
    ...
  }
}

```

プロジェクトの削除

プロジェクトの削除コマンドを実行すると、指定されたプール内の単一のプロジェクトが削除されます。次の URI パラメータを使用します。

pool ストレージプール名

project プロジェクト名

「非同期データセット削除」遅延更新 (OS8.7.0) が受け入れられた場合に、ストレージプール内の回収されるべき領域の量をモニターするには、`pools/pool` に対する GET コマンドを入力します。プロパティ `async_destroy_reclaim_space` の領域の量に注目します。操作が完了すると、0 が表示されます。

リクエストの例:

```
DELETE /api/storage/v1/pools/p1/projects/proj-01 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

プロジェクトの使用状況

プロジェクトの使用状況リソースの取得リクエストは、プロジェクトのユーザー当たりまたはグループ当たりの使用状況データの取得に使用できます。

ファイルシステムの操作

ファイルシステム操作では、ファイルシステム共有を一覧表示および管理します。すべてのコマンドは、特定のストレージプールまたはプロジェクトへスコープ指定されます。

`service_uri/pools/pool/project/project`

表 46 ファイルシステムのコマンド

リクエスト	パス <code>/api/storage/v{1 2}</code> の後ろに追加	説明
GET	<code>/filesystems</code>	すべてのファイルシステムを一覧表示します
GET	<code>/pools/pool/projects/project/filesystems</code>	指定されたファイルシステムを一覧表示します
GET	<code>/pools/pool/projects/project/filesystems?snaps=true</code>	スナップショットを含む、すべてのファイルシステムを一覧表示します
GET	<code>/pools/pool/projects/project/filesystems/filesystem</code>	ファイルシステムの詳細を取得します

リクエスト	パス <code>/api/storage/v{1 2}</code> の後ろに追加	説明
POST	<code>/pools/pool/projects/project/filesystems</code>	ファイルシステムを作成します
PUT	<code>/pools/pool/projects/project/filesystems/filesystem</code>	ファイルシステムを変更します
DELETE	<code>/pools/pool/projects/project/filesystems/filesystem</code>	ファイルシステムを破棄します
GET	<code>/pools/pool/projects/project/filesystems/filesystem/usage/groups</code>	ファイルシステムグループの使用状況を取得します
GET	<code>/pools/pool/projects/project/filesystems/filesystem/usage/groups/group</code>	指定されたグループのファイルシステムの使用状況を取得します
POST	<code>/pools/pool/projects/project/filesystems/filesystem/usage/groups</code>	ファイルシステムグループの割り当て制限を作成します
PUT	<code>/pools/pool/projects/project/filesystems/filesystem/usage/groups/name</code>	ファイルシステムグループの割り当て制限を変更します
GET	<code>/pools/pool/projects/project/filesystems/filesystem/usage/users</code>	ファイルシステムユーザーの使用状況を取得します
GET	<code>/pools/pool/projects/project/filesystems/filesystem/usage/users/user</code>	指定されたユーザーのファイルシステムの使用状況を取得します
POST	<code>/pools/pool/projects/project/filesystems/filesystem/usage/users</code>	ファイルシステムユーザーの割り当て制限を作成します
PUT	<code>/pools/pool/projects/project/filesystems/filesystem/usage/users/name</code>	ファイルシステムユーザーの割り当て制限を変更します
GET	<code>/pools/pool/projects/project/filesystems/filesystem/shadow/errors</code>	シャドウ移行エラーを一覧表示します

各ファイルシステムにはプロジェクトからのプロパティーが含まれ、次のファイルシステム固有のプロパティーがあります。

表 47 ファイルシステムのプロパティー

プロパティー	型	説明
<code>casesensitivity</code>	string	大/小文字の区別の設定: <code>mixed</code> 、 <code>sensitive</code> 、または <code>insensitive</code>
<code>group</code>	string	グループ名
<code>normalization</code>	string	正規化
<code>permissions</code>	string	ファイルシステムの権限
<code>project</code>	string	プロジェクト名
<code>quota_snap</code>	boolean	割り当て制限にスナップショットを含めるフラグ
<code>reservation_snap</code>	boolean	予約にスナップショットを含めるフラグ
<code>shadow</code>	string	データ移行ソース
<code>errors</code>	string	データ移行エラー
<code>sharesmb_name</code>	string	SMB シェアの名前
<code>source</code>	オブジェクト	プロジェクト継承プロパティー

すべてのプールおよびプロジェクトにわたるすべてのファイルシステムのリストもサポートされます。その場合、URI は `/api/storage/v{1|2}/filesystems` となります。

プロパティーの一部として「abcd」の文字列を持つすべてのファイルシステムを取得するリクエストの例:

```
GET /api/storage/v1/filesystems?filter=abcd HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

ファイルシステムの取得

ファイルシステムの取得コマンドを実行すると、指定されたプールまたはプロジェクト内の単一のファイルシステムのプロパティーが返されます。次の URI パラメータを使用します。

<i>pool</i>	ストレージプール名
<i>project</i>	プロジェクト名
<i>filesystem</i>	ファイルシステム名

proj-01 という名前のプロジェクトを一覧表示するリクエストの例:

```
GET /api/storage/v1/pools/p1/projects/proj-01 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

get が成功すると、HTTP ステータス 200 (OK) が JSON 形式のファイルシステムプロパティーとともに返されます。

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "filesystem": {
    "logbias": "latency",
    "creation": "20130423T21:30:34",
    "nodeldestroy": false,
    "dedup": false,
    "sharenfs": "on",
    "sharesmb": "off",
    "mountpoint": "/export/mnt1",
    "snaplabel": "",
    "id": "424ca2ec-b3fa-df86-0000-000000000000",
    "readonly": false,
    "rrsrc_actions": [],
    "compression": "off",
    "shareftp": "",
    "source": {
      "logbias": "default",
      "dedup": "default",
```

```

    "sharenfs": "inherited",
    "sharesmb": "off",
    "mountpoint": "inherited",
    "rrsrc_actions": "local",
    "compression": "default",
    "shareftp": "inherited",
    "snapdir": "default",
    "aclmode": "default",
    "copies": "default",
    "aclinherit": "default",
    "shareftp": "inherited",
    "readonly": "default",
    "secondarycache": "default",
    "exported": "inherited",
    "vscan": "default",
    "reservation": "local",
    "atime": "default",
    "recordsize": "default",
    "checksum": "inherited",
    "sharesftp": "inherited",
    "nbmand": "default",
    "rstchown": "default"
  },
  "snapdir": "hidden",
  "aclmode": "discard",
  "copies": 1,
  "aclinherit": "restricted",
  "shareftp": "",
  "canonical_name": "p1/local/default/mnt1",
  "recordsize": 131072.0,
  "usage": {
    "available": 880395477504.0,
    "loading": false,
    "quota": 0.0,
    "snapshots": 18432.0,
    "compressratio": 100.0,
    "reservation": 0.0,
    "total": 50176.0,
    "data": 31744.0
  },
  "secondarycache": "all",
  "collection": "local",
  "exported": true,
  "vscan": false,
  "reservation": 0.0,
  "shadow": "none",
  "atime": true,
  "pool": "p1",
  "quota_snap": true,
  "name": "mnt1",
  "checksum": "fletcher4",
  "project": "default",
  "sharesftp": "",
  "nbmand": false,
  "reservation_snap": true,
  "sharedav": "",
  "rstchown": true,
  "root_acl": {
    "owner@:cc:fd:deny",
    "everyone@:rw:fd:allow",
    "user:admin1:rw:allow",
  }
}
"smbshareacl": {
  "owner@:cc:fd:deny",

```

```

        "everyone@:rw:fd:allow",
        "user:admin1:rw:allow",
    }
}

```

ファイルシステムの作成

ファイルシステムの作成コマンドを実行すると、指定されたストレージプールまたはプロジェクトに指定された名前のファイルシステムが作成されます。デフォルトのプロパティーを持つ新しいファイルシステムが返されます。

次の URI パラメータを使用します。

<i>pool</i>	ストレージプール名
<i>project</i>	プロジェクト名
<i>filesystem</i>	ファイルシステム名

リクエストパラメータ:

- 名前 – 新しいファイルシステムを作成するには、ファイルシステム名を指定する必要があります。
- ファイルシステムプロパティー – ファイルシステムプロパティーまたはプロジェクトプロパティーに一覧表示された任意のプロパティーを初期値として設定できます。

リクエストの例 (share-01 という名前で、ユーザー admin1 が所有するファイルシステムを作成する):

```

POST /api/storage/v1/pools/p1/projects/proj-01/filesystems HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json

{
  "name": "share-01",
  "root_user": "admin1"
}

```

作成に成功すると、HTTP ステータス 201 (Created) が、新しいファイルシステムの URI を含むロケーションヘッダーとともに返されます。本文には、JSON 形式のファイルシステムプロパティーがすべて含まれます。

結果例:

```

HTTP/1.1 201 Created
Content-Type: application/json
Location: /api/storage/v1/pools/p1/projects/proj-01/filesystems/share-01

{

```

```

    "filesystem": {
      "name": "share-01",
      "pool": "p1",
      "collection": "local",
      "project": "proj-01",
      "root_user": "admin1"
      ...
    }
  }
}

```

ファイルシステムの変更

ファイルシステムの変更コマンドは、既存のファイルシステムの属性を変更します。応答に成功すると、HTTP ステータス 202 (Accepted) が返され、すべてのファイルシステムプロパティーが一覧表示されます。

次の URI パラメータを使用します。

pool ストレージプール名

project プロジェクト名

filesystem ファイルシステム名

リクエストパラメータ: ファイルシステムプロパティー – 任意のファイルシステムプロパティーまたはプロジェクトプロパティーを変更できます。

リクエストの例 (ファイルシステム名を `share-01` から `new-name` に変更し、所有者を `nobody` に変更する):

```

PUT /api/storage/v1/pools/p1/projects/proj-01/filesystems/share-01 HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json

```

```

{
  "name": "new-name",
  "root_user": "nobody",
}

```

結果例:

```

HTTP/1.1 202 Accepted
Content-Type: application/json
Location: http://zfs-storage.example.com:215/pools/p1/projects/proj-01/filesystems/share-01

```

```

{
  "filesystem": {
    "name": "new-name",
    "pool": "p1",
    "collection": "local",
    "project": "proj-01",
    "root_user": "nobody"
    ...
  }
}

```

```
}
}
```

ファイルシステムの削除

ファイルシステムの削除コマンドを実行すると、指定されたプールまたはプロジェクト内の単一のファイルシステムを削除します。

次の URI パラメータを使用します。

<i>pool</i>	ストレージプール名
<i>project</i>	プロジェクト名
<i>filesystem</i>	ファイルシステム名

ストレージプール内の回収されるべき領域の量をモニターするには、*pools/pool* に対する GET コマンドを入力します。プロパティ *async_destroy_reclaim_space* の領域の量に注目します。操作が完了すると、0 が表示されます。

リクエストの例:

```
DELETE /api/storage/v1/pools/p1/projects/proj-01/filesystems/share-01 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

削除に成功すると、HTTP ステータス 204 (No Content) が返されます。

結果例:

```
HTTP/1.1 204 No-Content
```

ファイルシステムの割り当て制限および使用状況

ユーザーまたはグループの割り当て制限は、それぞれ POST または PUT リクエストを使用して作成または変更できます。ファイルシステム使用リソースへの GET リクエストは、プロジェクトのユーザー当たりまたはグループ当たりの使用状況データの取得に使用できます。

LUN 操作

LUN またはボリュームの操作はすべて、指定されたプールまたはプロジェクトにスコープ指定されます。次の LUN コマンドを使用できます。

表 48 ボリュームのコマンド

リクエスト	パス <code>/api/storage/v{1 2}</code> の後ろに追加	説明
GET	<code>/luns</code>	すべての LUN を一覧表示します
GET	<code>/pools/pool/projects/project/luns</code>	LUN を一覧表示します
GET	<code>/pools/pool/projects/project/luns?snaps=true</code>	スナップショットを含む、すべての LUN を一覧表示します
GET	<code>/pools/pool/projects/project/luns/lun</code>	LUN の詳細を取得します
POST	<code>/pools/pool/projects/project/luns</code>	LUN を作成します
PUT	<code>/pools/pool/projects/project/luns/lun</code>	LUN を変更します
DELETE	<code>/pools/pool/projects/project/luns/lun</code>	LUN を破棄します

次の表に、LUN プロパティを示します。ボリュームは、プロジェクトのプロパティを継承またはオーバーライドすることもできます。

表 49 ボリュームのプロパティ

プロパティ	型	説明
<code>assignednumber</code>	数値または数値のリスト	割り当てられた LU 番号。複数のイニシエータグループに提示される場合、型は数値のリストです。 複数のイニシエータグループに提示される場合、 <code>assignednumber</code> と <code>initiatorgroups</code> は同じ順序で並べられます。たとえば、 <code>assignednumber</code> リストの最初の項目は、 <code>initiatorgroups</code> リストの最初の項目に関連します。
<code>fixednumber</code>	boolean	現在の値で LU 番号を修正するフラグ
<code>initiatorgroups</code>	文字列のリスト	イニシエータグループ。 LUN が複数のイニシエータグループに提示される場合、 <code>assignednumber</code> と <code>initiatorgroups</code> は同じ順序で並べられます。たとえば、 <code>assignednumber</code> リストの最初の項目は、 <code>initiatorgroups</code> リストの最初の項目に関連します。
<code>lunguid</code>	string	STMF の GUID
<code>lunumber</code>	数値または文字列	LU 番号。数字または <code>auto</code> のどちらか。
<code>project</code>	string	プロジェクト名 (変更不可)
<code>source</code>	オブジェクト	プロパティのソース (<code>local</code> または <code>inherited</code>) を一覧表示します
<code>sparse</code>	boolean	シンプロビジョニングを有効にするフラグ
<code>status</code>	string	論理ユニットのステータス: <code>online</code> または <code>offline</code>
<code>targetgroup</code>	string	ターゲットグループ
<code>usage</code>	オブジェクト	LUN 使用状況の統計を一覧表示します
<code>volblocksize</code>	number	ボリュームのブロックサイズ
<code>volsize</code>	number	ボリュームのサイズ
<code>writecache</code>	boolean	書き込みキャッシュを有効にするフラグ

一部のプロパティは、プロジェクトから継承することもできます。ソースオブジェクトは、これらのプロパティのそれぞれを一覧表示し、プロパティが LUN にとってローカルなものであるか、それともプロジェクトから継承されたものであるかを特定します。デフォルトでは、これらのプロパティはプロジェクトから継承されます。設定されると、LUN にとってローカルになります。ソースオブジェクトは変更できません。ソースを継承されたものに戻すには、プロパティを設定解除します。

圧縮を設定解除する JSON リクエストの例:

```
{"unset": ["compression"]}
```

LUN を一覧表示します

LUN の一覧表示コマンドを実行すると、指定されたプールまたはプロジェクト内で使用可能な LUN のリストが返されます。

注記 - `depth` 問合せパラメータおよび `match_property-name=value` 問合せパラメータはサポートされていません。

次の URI パラメータを使用します。

<i>pool</i>	ストレージプール名
<i>project</i>	プロジェクト名
<i>filesystem</i>	ファイルシステム名

プロジェクト `proj-01` 内の LUN を一覧表示するリクエストの例:

```
GET /api/storage/v1/pools/p1/projects/proj-01/luns HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

`get` が成功すると、HTTP ステータス 200 (OK) が JSON 形式の LUN プロパティとともに返されます。

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "luns": [{
    "id": "fa4ac6fb-0bcc-d2e3-0000-000000000000",
    "name": "vol-01"
  }, {
    "id": "690ae407-7c4d-b5d2-0000-000000000000",
    "name": "vol-01",
  }
]
```

LUN の取得

LUN の取得コマンドを実行すると、指定されたプールまたはプロジェクト内の単一の LUN のプロパティが返されます。

次の URI パラメータを使用します。

pool ストレージプール名

project プロジェクト名

lun LUN の名前

リクエストの例 (「vol-01」という名前の LUN を取得する):

```
GET /api/storage/v1/pools/p1/projects/proj-01/lun/vol-01 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

get が成功すると、HTTP ステータス 200 (OK) が JSON 形式の LUN プロパティとともに返されます。

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "lun": {
    "logbias": "latency",
    "creation": "20130423T21:31:17",
    "nodeestroy": false,
    "dedup": false,
    "rrsrc_actions": [],
    "id": "e3045406-319b-cf7a-0000-000000000000",
    "writecache": false,
    "compression": "off",
    "copies": 1,
    "stmfguid": "600144F0D8E0AE4100005176FDA60001",
    "source": {
      "compression": "default",
      "checksum": "inherited",
      "logbias": "default",
      "dedup": "default",
      "copies": "default",
      "exported": "inherited",
      "rrsrc_actions": "inherited",
      "secondarycache": "default"
    },
    "canonical_name": "p1/local/default/disk1",
    "snaplabel": "",
    "usage": {
      "available": 881469214720.0,
      "loading": false,
      "snapshots": 0.0,
      "compressratio": 100.0,
      "total": 1073758208.0,

```

```

        "data": 1073758208.0
      },
      "secondarycache": "all",
      "collection": "local",
      "exported": true,
      "volsize": 1073741824.0,
      "pool": "p1",
      "volblocksize": 8192,
      "checksum": "fletcher4",
      "project": "default",
      "sparse": false
    }
  }
}

```

新しい LUN の作成

このコマンドを実行すると、新しい LUN が作成されます。新しい LUN に対してサイズまたはクローニングソースを指定する必要があります。

次の URI パラメータを使用します。

pool ストレージプール名

project プロジェクト名

リクエストのパラメータ:

- 名前 – 新しい LUN を作成するには、LUN 名を指定する必要があります。
- ボリュームプロパティ – LUN プロパティまたはプロジェクトプロパティに一覧表示された任意のプロパティを初期値として設定できます。

リクエストの例:

```

POST /api/storage/v1/pools/p1/projects/proj-01/luns HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json

```

Request JSON:

```

{
  name : "vol-001",           // Volume name (required)

  size : 500000,              // New Volume size
  blocksize : 8192,           // New Volume block size
  sparse : true,              // New Volume sparse data flag

  initiatorgroup : 'default', // Initiator group name
  targetgroup : 'default',    // Target group name
  lunumber : 'auto',          // Volume LU number
  status : 'online',          // Initial Status ('online', 'offline')
  fixednumber : false,

  "source": {
    "snapshot_id" : "76b8950a-8594-4e5b-8dce-0dfa9c696358",
    "snapshot": "/pool-001/local/proj-001/snap-001"
  }
}

```

作成に成功すると、HTTP ステータス 201 (Created) が、新しい LUN の URI を含むロケーションヘッダーとともに返されます。本文には、JSON 形式の LUN プロパティーがすべて含まれます。

結果例:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://zfs-storage.example.com:215
          /pools/p1/projects/proj-01/luns/vol-001
```

```
{
  "lun": {
    "name": "vol-001",
    ...
  }
}
```

LUN の変更

LUN の変更コマンドは、既存の LUN の属性を変更します。

次の URI パラメータを使用します。

pool ストレージプール名

project プロジェクト名

lun LUN の名前

リクエストパラメータ: ボリュームプロパティー – 任意の LUN プロパティーまたはプロジェクトプロパティーを変更できます。

LUN 名を *vol-01* から *new-name* に変更するリクエストの例:

```
POST /api/storage/v1/pools/p1/projects/proj-01/luns/vol-01 HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json
```

```
{
  "name": "new-name",
}
```

応答に成功すると、HTTP ステータス 202 (Accepted) が返され、すべての LUN プロパティーが一覧表示されます。

結果例:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: /api/storage/v1/pools/p1/projects/proj-01/luns/new-name
```

```
{
  "lun": {
```

```

    "name": "new-name",
    "pool": "p1",
    "collection": "local",
    "project": "proj-01",
    ...
  }
}

```

LUN の削除

LUN の削除コマンドを実行すると、指定されたプールまたはプロジェクト内の単一の LUN が削除されます。

次の URI パラメータを使用します。

<i>pool</i>	ストレージプール名
<i>project</i>	プロジェクト名
<i>lun</i>	LUN の名前

ストレージプール内の回収されるべき領域の量をモニターするには、*pools/pool* に対する GET コマンドを入力します。プロパティ *async_destroy_reclaim_space* の領域の量に注目します。操作が完了すると、0 が表示されます。

リクエストの例:

```

DELETE /pools/p1/projects/proj-01/luns/lun-01 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json

```

get が成功すると、HTTP ステータス 204 (No Content) が返されます。

結果例:

```

HTTP/1.1 204 No-Content

```

スナップショットおよびクローンの操作

スナップショットの操作はすべて、指定されたプールまたはプロジェクトにスコープ指定されます。スナップショット操作は、ファイルシステムまたは LUN レベルにもスコープ指定できます。

- すべてのプロジェクトベースのスナップショット操作の URI は `/api/storage/v{1|2}/pools/pool/projects/project` で始まります。
- すべてのファイルシステムベースのスナップショット操作の URI は `/api/storage/v{1|2}/pools/pool/projects/project/filesystems/filesystem` で始まります。

- すべての LUN ベースのスナップショット操作の URI は `/api/storage/v{1|2}/pools/pool/projects/project/luns/lun` で始まります。

スナップショットをクラウドにバックアップする、またはスナップショットバックアップを新しいシェアとしてアプライアンスに復元するには、[113 ページの「RESTful API クラウドサービス」](#)を参照してください。

表 50 スナップショットおよびクローンのコマンド

リクエスト	パス <code>/api/storage/v{1 2}</code> の後ろに追加	説明
GET	<code>/snapshots</code>	すべてのローカルスナップショットを一覧表示します
GET	<code>/pools/pool/projects?snaps=true</code>	スナップショットを含む、すべてのプロジェクトを一覧表示します
GET	<code>/pools/pool/projects/project/filesystems?snaps=true</code>	スナップショットを含む、すべてのファイルシステムを一覧表示します
GET	<code>/pools/pool/projects/project/luns?snaps=true</code>	スナップショットを含む、すべての LUN を一覧表示します
GET	<code>/pools/pool/projects/project/snapshots</code>	プロジェクトのすべてのスナップショットを一覧表示します
GET	<code>/pools/pool/projects/project/filesystems/filesystem/snapshots</code>	ファイルシステムのすべてのスナップショットを一覧表示します
GET	<code>/pools/pool/projects/project/luns/lun/snapshots</code>	LUN のすべてのスナップショットを一覧表示します
GET	<code>/pools/pool/projects/project/snapshots/snapshot</code>	プロジェクトスナップショットの詳細を取得します
GET	<code>/pools/pool/projects/project/filesystems/filesystem/snapshots/snapshot</code>	ファイルシステムスナップショットの詳細を取得します
GET	<code>/pools/pool/projects/project/luns/lun/snapshots/snapshot</code>	LUN スナップショットの詳細を取得します
POST	<code>/pools/pool/projects/project/snapshots</code>	プロジェクトスナップショットを作成します
POST	<code>/pools/pool/projects/project/filesystems/filesystem/snapshots</code>	ファイルシステムスナップショットを作成します
POST	<code>/pools/pool/projects/project/luns/lun/snapshots</code>	LUN スナップショットを作成します
PUT	<code>/pools/pool/projects/project/snapshots/snapshot</code>	プロジェクトスナップショットを変更します
PUT	<code>/pools/pool/projects/project/filesystems/filesystem/snapshots/snapshot</code>	ファイルシステムスナップショットを変更します
PUT	<code>/pools/pool/projects/project/luns/lun/snapshots/snapshot</code>	LUN スナップショットを変更します
PUT	<code>/pools/pool/projects/project/filesystems/filesystem/snapshots/snapshot/clone</code>	ファイルシステムスナップショットをクローン作成します
PUT	<code>/pools/pool/projects/project/luns/lun/snapshots/snapshot/clone</code>	LUN スナップショットをクローン作成します

リクエスト	パス <code>/api/storage/v{1 2}</code> の後ろに追加	説明
PUT	<code>/pools/pool/projects/project/filesystems/filesystem/snapshots/snapshot/rollback</code>	データを指定されたファイルシステムスナップショットにロールバックします
PUT	<code>/pools/pool/projects/project/lun/lun/snapshots/snapshot/rollback</code>	データを指定された LUN スナップショットにロールバックします
DELETE	<code>/pools/pool/projects/project/snapshots/snapshot</code>	プロジェクトスナップショットを破棄します
DELETE	<code>/pools/pool/projects/project/filesystems/filesystem/snapshots/snapshot</code>	ファイルシステムスナップショットを破棄します
DELETE	<code>/pools/pool/projects/project/luns/lun/snapshots/snapshot</code>	LUN スナップショットを破棄します
GET	<code>/pools/pool/projects/project/snapshots/snapshot/dependents</code>	プロジェクトスナップショットの依存を一覧表示します
GET	<code>/pools/pool/projects/project/filesystems/filesystem/snapshots/snapshot/dependents</code>	ファイルシステムスナップショットの依存を一覧表示します
GET	<code>/pools/pool/projects/project/lun/lun/snapshots/snapshot/dependents</code>	LUN スナップショットの依存を一覧表示します
POST	<code>/pools/pool/projects/project/automatic</code>	新しいプロジェクト自動スナップショットオブジェクトを作成します
POST	<code>/pools/pool/projects/project/automatic?convert=true</code>	新しいプロジェクト自動スナップショットオブジェクトを作成します。新しいスケジュールに適合しない自動生成された既存のスナップショットは、手動スナップショットに変換されます。 <code>convert</code> プロパティを除外すると、自動生成された既存のスナップショットは破棄されます。
GET	<code>/pools/pool/projects/project/automatic/automatic</code>	指定されたプロジェクトの自動スナップショットプロパティを取得します
GET	<code>/pools/pool/projects/project/automatic</code>	すべてのプロジェクト自動スナップショットオブジェクトを一覧表示します
PUT	<code>/pools/pool/projects/project/automatic/automatic</code>	指定されたプロジェクトの自動スナップショットオブジェクトを変更します
PUT	<code>/pools/pool/projects/project/automatic/automatic?convert=true</code>	指定されたプロジェクト自動スナップショットスケジュールオブジェクトを変更します。新しいスケジュールに適合しない自動生成された既存のスナップショットは、手動スナップショットに変換されます。 <code>convert</code> プロパティを除外すると、自動生成された既存のスナップショットは破棄されます。

スナップショットおよびクローンの操作

リクエスト	パス <i>/api/storage/v{1 2}</i> の後ろに追加	説明
DELETE	<i>/pools/pool/projects/project/automatic/automatic</i>	指定された自動オブジェクトを破棄します
DELETE	<i>/pools/pool/projects/project/automatic/automatic?convert=true</i>	指定された自動スナップショットスケジュールオブジェクトを破棄します。新しいスケジュールに適合しない自動生成された既存のスナップショットは、手動スナップショットに変換されます。 <i>convert</i> プロパティを除外すると、自動生成された既存のスナップショットは破棄されます。
POST	<i>/pools/pool/projects/project/filesystems/filesystem/automatic</i>	新しいファイルシステム自動スナップショットオブジェクトを作成します
POST	<i>/pools/pool/projects/project/filesystems/filesystem/automatic?convert=true</i>	新しいファイルシステム自動スナップショットオブジェクトを作成します。新しいスケジュールに適合しない自動生成された既存のスナップショットは、手動スナップショットに変換されます。 <i>convert</i> プロパティを除外すると、自動生成された既存のスナップショットは破棄されます。
GET	<i>/pools/pool/projects/project/filesystems/filesystem/automatic/automatic</i>	指定されたファイルシステムの自動スナップショットプロパティを取得します
GET	<i>/pools/pool/projects/project/filesystems/filesystem/automatic</i>	すべてのファイルシステムの自動スナップショットオブジェクトを一覧表示します
PUT	<i>/pools/pool/projects/project/filesystems/filesystem/automatic/automatic</i>	指定されたファイルシステムの自動スナップショットオブジェクトを変更します
PUT	<i>/pools/pool/projects/project/filesystems/filesystem/automatic/automatic?convert=true</i>	指定されたファイルシステム自動スナップショットスケジュールオブジェクトを変更します。新しいスケジュールに適合しない自動生成された既存のスナップショットは、手動スナップショットに変換されます。 <i>convert</i> プロパティを除外すると、自動生成された既存のスナップショットは破棄されます。
DELETE	<i>/pools/pool/projects/project/filesystems/filesystem/automatic/automatic</i>	指定された自動スナップショットスケジュールオブジェクトを破棄します
DELETE	<i>/pools/pool/projects/project/filesystems/filesystem/automatic/automatic?convert=true</i>	指定されたファイルシステム自動スナップショットスケジュールオブジェクトを破棄します。新しいスケジュールに適合しない自動生成され

リクエスト	パス <code>/api/storage/v{1 2}</code> の後ろに追加	説明
		た既存のスナップショットは、手動スナップショットに変換されます。 <code>convert</code> プロパティを除外すると、自動生成された既存のスナップショットは破棄されます。
POST	<code>/pools/pool/projects/project/luns/lun/automatic</code>	新しい LUN の自動スナップショットを作成します
POST	<code>/pools/pool/projects/project/luns/lun/automatic?convert=true</code>	新しい LUN 自動スナップショットスケジュールを作成します。新しいスケジュールに適合しない自動生成された既存のスナップショットは、手動スナップショットに変換されます。 <code>convert</code> プロパティを除外すると、自動生成された既存のスナップショットは破棄されます。
GET	<code>/pools/pool/projects/project/luns/lun/automatic/automatic</code>	指定された LUN の自動スナップショットプロパティを取得します
GET	<code>/pools/pool/projects/project/luns/lun/automatic</code>	すべての LUN の自動スナップショットオブジェクトを一覧表示します
PUT	<code>/pools/pool/projects/project/luns/lun/automatic/automatic</code>	指定された LUN の自動スナップショットオブジェクトを変更します
PUT	<code>/pools/pool/projects/project/luns/lun/automatic/automatic?convert=true</code>	指定された LUN 自動スナップショットスケジュールオブジェクトを変更します。新しいスケジュールに適合しない自動生成された既存のスナップショットは、手動スナップショットに変換されます。 <code>convert</code> プロパティを除外すると、自動生成された既存のスナップショットは破棄されます。
DELETE	<code>/pools/pool/projects/project/luns/lun/automatic/automatic</code>	指定された LUN 自動オブジェクトを破棄します
DELETE	<code>/pools/pool/projects/project/luns/lun/automatic/automatic?convert=true</code>	指定された LUN 自動スナップショットスケジュールオブジェクトを破棄します。新しいスケジュールに適合しない自動生成された既存のスナップショットは、手動スナップショットに変換されます。 <code>convert</code> プロパティを除外すると、自動生成された既存のスナップショットは破棄されます。

スナップショットの一覧表示

アプライアンス上の使用可能なスナップショットを一覧表示します。リクエスト URI に応じて、リストにはプロジェクト、ファイルシステム、または LUN のスナップショットが含まれます。

表 51 スナップショットの一覧表示コマンドの形式

コマンド	パス <code>/api/storage/v{1 2}/pools/pool/projects/project</code> の後ろに追加
プロジェクトスナップショットの一覧表示	<code>/snapshots</code>
ファイルシステムスナップショットの一覧表示	<code>/filesystems/share/snapshots</code>
LUN スナップショットの一覧表示	<code>/lun/share/snapshots</code>

リクエストの例:

```
GET /api/storage/v1/pools/p1/projects/default/snapshots
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "snapshots": [{
    "id": "3fbbcccf-d058-4502-8844-6feeffdf4cb5",
    "display_name": "snap-001",
    "display_description": "Daily backup",
    "volume_id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
    "status": "available",
    "size": 30,
    "created_at": "2012-02-29T03:50:07Z"
  }, {
    "id": "e479997c-650b-40a4-9dfe-77655818b0d2",
    "display_name": "snap-002",
    "display_description": "Weekly backup",
    "volume_id": "76b8950a-8594-4e5b-8dce-0dfa9c696358",
    "status": "available",
    "size": 25,
    "created_at": "2012-03-19T01:52:47Z"
  }]
}
```

スナップショットの取得

単一のスナップショットに関するすべての情報を表示します。成功すると、HTTP ステータス 200 (OK) が返されます。

リクエストの例:

```
GET /api/storage/v1/pools/p1/projects/default/snapshots/snap-001
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "snapshot": {
    "id": "3fbbcccf-d058-4502-8844-6feeffdf4cb5",
    "display_name": "snap-001",
    "display_description": "Daily backup",
    "volume_id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
    "status": "available",
    "size": 30,
    "created_at": "2012-02-29T03:50:07Z"
  }
}
```

スナップショットの作成

スナップショットの作成コマンドを実行すると、プロジェクト、ファイルシステム、または LUN のスナップショットが作成されます。

- プロジェクトスナップショットの作成 –
POST /pools/pool/projects/project/snapshots
- ファイルシステムスナップショットの作成 –
POST /pools/pool/projects/project/filesystems/share/snapshots
- ボリュームスナップショットの作成 –
POST /pools/pool/projects/project/luns/lun/snapshots

リクエストの例:

```
POST /api/storage/v1/pools/p1/projects/default/snapshots
Content-Type: application/json
```

```
{"name": "initial-backup"}
```

結果例:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: /pools/p1/projects/default/
snapshot/initial-backup
```

```
{
  "snapshot": {
    "name": "initial-backup",
    "numclones": 0,
    "creation": "20130610T21:00:49",
    "collection": "local",
    "project": "default",
    "canonical_name": "zfs-storage-1/local/default@initial-backup",
    "usage": {
      "unique": 0.0,
      "loading": false,

```

```

        "data": 145408.0
      },
      "type": "snapshot",
      "id": "a26abd24-e22b-62b2-0000-000000000000",
      "pool": "p1"
    }
  }
}

```

スナップショットの名前変更

既存のスナップショットの名前を変更します。

- **リクエスト URI** – スナップショット、現在のスナップショット名
- **リクエスト本文** – 新しいスナップショット名を含む名前パラメータを持つ JSON オブジェクト

リクエストの例:

```

PUT /api/storage/v1/pools/p1/projects/default/snapshots/initial-snapshot
Content-Type: application/json
Accept: application/json

{"name": "old-snapshot"}

```

結果例:

```

HTTP/1.1 202 Accepted
Content-Type: application/json
Location: /pools/p1/projects/default/snapshot/initial-backup

```

スナップショットのクローン作成

既存のスナップショットから新しいファイルシステムまたは LUN を作成します。

次の URI パラメータを使用します。

<i>pool</i>	ソースプール名
<i>project</i>	ソースプロジェクト名
<i>filesystem</i>	ファイルシステムスナップショットのソースシェア名
<i>lun</i>	LUN スナップショットのソースシェア名
<i>snapshot</i>	ソーススナップショット名

ファイルシステムのクローン作成:

```

PUT /pools/pool/projects/project/filesystems/share/snapshots/snapshot/clone

```

ボリュームのクローン作成:

```
PUT /pools/pool/projects/project/luns/lun/snapshots/snapshot/clone
```

リクエストの本文には、次のプロパティを持つ JSON オブジェクトが含まれます。

表 52 スナップショットのクローン作成のプロパティ

プロパティ	型	説明
pool	string	宛先クローンプール名
project	string	宛先クローンプロジェクト名
lun	string	LUN スナップショットの宛先 LUN 名

リクエストの例:

```
PUT /api/storage/v1/pools/p1/projects/default/filesystems/fs01/
    snapshots/snap01/clone
```

```
{"project": "rest", "share": "snap01clone01", "compression": "gzip-9"}
```

結果例:

```
HTTP/1.1 201 Created
Content-Length: 2035
X-Zfssa-Storage-API: 1.0
Location: /api/storage/v1/pools/p1/projects/rest/filesystem/snap01clone01
Content-Type: application/json; charset=utf-8
```

```
{
  "filesystem": {
    "origin": {
      "project": "default",
      "share": "fs01",
      "snapshot": "snap01",
      "pool": "p1",
      "collection": "local"
    },
    "href": "/api/storage/v1/pools/p1/projects/rest/filesystems/snap01clone01",
    "mountpoint": "/export/snap01clone01",
    "compression": "gzip-9",
    "source": {
      "compression": "local",
      ...
    },
    ...
  },
  "canonical_name": "zfs-storage-1/local/rest/snap01clone01"
}
```

スナップショットのロールバック

スナップショットのロールバックでは、ソースのファイルシステムまたは LUN が、スナップショットが作成されたときの状態に戻されます。応答に成功すると、HTTP ステータス 202 (Accepted) が JSON 形式のスナップショットプロパティとともに返されます。

次の URI パラメータを使用します。

<i>pool</i>	ソースプール名
<i>project</i>	ソースプロジェクト名
<i>filesystem</i>	ファイルシステムスナップショットのソースファイルシステム名
<i>lun</i>	LUN スナップショットのソース LUN 名
<i>snapshot</i>	ソーススナップショット名

ファイルシステムスナップショットのロールバック:

```
PUT /pools/pool/projects/project/filesystems/filesystem/snapshots/snapshot/rollback
```

LUN スナップショットのロールバック:

```
PUT /pools/pool/projects/project/luns/lun/snapshots/snapshot/rollback
```

リクエストの例:

```
PUT /api/storage/v1/pools/p1/projects/default/filesystems/fs-01/snapshots/initial-backup/rollback
```

結果例:

```
HTTP/1.1 202 Accepted
Location: /pools/p1/projects/default/filesystems/fs-01/snapshot/fs-01-initial-clone
Content-Type: application/json
```

```
{
  "snapshot": {
    "name": "fs-01-initial-clone",
    "numclones": 0,
    "creation": "20130610T21:00:49",
    "filesystem": "fs-01",
    "collection": "local",
    "project": "default",
    "canonical_name": "zfs-storage-1/local/default/fs-01@fs-01-initial-clone",
    "usage": {
      "unique": 0.0,
      "loading": false,
      "data": 31744.0
    },
    "type": "snapshot",
    "id": "5c9bda07-21c1-2238-0000-000000000000",
    "pool": "p1"
  }
}
```

スナップショットの削除

スナップショットの DELETE コマンドを実行すると、システムからプロジェクト、ファイルシステム、または LUN のスナップショットが削除されます。

次の URI パラメータを使用します。

<i>pool</i>	ソースプール名
<i>project</i>	ソースプロジェクト名
<i>filesystem</i>	ソースファイルシステム名
<i>lun</i>	LUN の名前
<i>snapshot</i>	ソーススナップショット名

プロジェクトスナップショットの削除:

```
DELETE /api/storage/v1/pools/pool/projects/project/snapshots/snapshot
```

ファイルシステムスナップショットの削除:

```
DELETE /api/storage/v1/pools/pool/projects/project/filesystems/filesystem/snapshots/snapshot
```

ファイルシステム LUN の削除:

```
DELETE /api/storage/v1/pools/pool/projects/projectsnapshot
```

スナップショットに NDMP ホールドがある場合は、`?confirm=true` を `DELETE` コマンドに追加してください。ただし、これは NDMP の操作に悪影響を及ぼす可能性があります。詳細については、『[Oracle ZFS Storage Appliance 管理ガイド、Release OS8.8.x](#)』の「[NDMP 構成](#)」を参照してください。

リクエストの例:

```
DELETE /pools/p1/projects/default/filesystems/fs-01/snapshots/initial-backup?confirm=true
```

`?confirm=true` が追加されない場合の結果の例:

スナップショットに NDMP ホールドが存在するときに `?confirm=true` が追加されない場合、コマンドは失敗し、次の出力が表示されます (読みやすさのために人為的に改行されています)。

```
HTTP/1.1 409 Conflict
```

```
{"fault": {"message": "request requires confirm=true to complete (confirmation\nneeded for scripted command (scripted commands must be prefixed with \"confirm\"\nto automatically confirm or \"deny\" to automatically deny) (encountered while\nattempting to run command \"confirm destroy snap\")", "code": 409, "name":\n"ERR_CONFIRM_REQUIRED"}}
```

スナップショット依存の一覧表示

ファイルシステムまたはボリュームの依存を一覧表示します。次の URI パラメータを使用します。

<i>pool</i>	システムのストレージプール名
<i>project</i>	プロジェクト名
<i>filesystem</i>	ファイルシステム名
<i>lun</i>	LUN の名前
<i>snapshot</i>	スナップショット名

ファイルシステムの依存の一覧表示:

```
GET /api/storage/v1/pools/pool/projects/project/filesystems/filesystem/snapshots/snapshot/dependents
```

ボリューム依存の一覧表示:

```
GET /api/storage/v1/pools/pool/projects/project/lun/lun/snapshots/snapshot/dependents
```

リクエストの例:

```
GET /api/storage/v1/pools/p1/projects/default/filesystems/fs01/snapshots/snap01/dependents
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
X-Zfssa-Storage-API: 1.0
Content-Type: application/json; charset=utf-8
X-Zfssa-API-Version: 1.0

{
  "dependents": [
    {
      "project": "rest",
      "href": "/api/storage/v1/pools/p1/projects/rest/filesystems/snap01clone01",
      "share": "snap01clone01"
    },
    {
      "project": "rest",
      "href": "/api/storage/v1/pools/p1/projects/rest/filesystems/snap01clone02",
      "share": "snap01clone02"
    },
    {
      "project": "rest",
      "href": "/api/storage/v1/pools/p1/projects/rest/filesystems/snap01clone03",
      "share": "snap01clone03"
    }
  ]
}
```

スキーマ

カスタムのスキーマプロパティを管理します。

表 53 スキーマコマンド

リクエスト	パス <code>/api/storage/v{1 2}</code> の後ろに追加	説明
GET	<code>/schema</code>	すべての NAS スキーマプロパティオブジェクトを一覧表示します
GET	<code>/schema/property</code>	指定された NAS スキーマプロパティのプロパティを取得します
POST	<code>/schema</code>	新しい NAS スキーマプロパティを作成します
PUT	<code>/schema/property</code>	指定された NAS スキーマプロパティオブジェクトを変更します
DELETE	<code>/schema/property</code>	指定された NAS スキーマプロパティオブジェクトを削除します

カスタムプロパティ名に接頭辞 `custom:` を追加すると、プロジェクト、ファイルシステム、および LUN に各カスタムのスキーマプロパティを設定できます。

たとえば、次の PUT 本文は、`priority` というカスタムの `int` プロパティを変更します。

```
{"custom:priority": 5}
```

表 54 スキーマパラメータ

パラメータ	説明
<code>property</code>	プロパティの名前 (変更不可)
<code>description</code>	プロパティの説明 (ブラウザインタフェースの場合)
<code>type</code>	型 (「String」、「Integer」、「PositiveInteger」、「Boolean」、「EmailAddress」、「Host」)

プロパティの一覧表示

スキーマプロパティを一覧表示します。

リクエストの例:

```
GET /api/storage/v1/schema
```

結果例:

```
{
  "properties": [{
    "description": "bob",
    "href": "/api/storage/v1/schema/bob",
    "property": "bob",
    "type": "String"
  }], {
```

```
        "description": "boo",
        "href": "/api/storage/v1/schema/boo",
        "property": "boo",
        "type": "String"
    }
}
```

プロパティの取得

スキーマプロパティを取得します。

リクエストの例:

```
GET /api/storage/v1/schema/priority
```

結果例:

```
{
  "property": {
    "description": "priority",
    "href": "/api/storage/v1/schema/priority",
    "property": "bob",
    "type": "Integer"
  }
}
```

プロパティの作成

新しいスキーマプロパティを作成します。

リクエストの例:

```
POST /api/storage/v1/schema HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Content-Length: 64
```

```
{"property":"priority", "type":"Integer", "description":"Oh my"}
```

結果例:

```
HTTP/1.1 201 Created
Content-Length: 89
X-Zfssa-Nas-API: 1.0
Content-Type: application/json
Location: /api/storage/v1/schema/priority
```

```
{
  "property": {
    "href": "/api/storage/v1/schema",
    "type": "Integer",
    "description": "Oh my"
  }
}
```

プロパティの変更

スキーマプロパティを変更します。

リクエストの例:

```
PUT /api/storage/v1/schema/priority
{"description": "My custom priority level"}
```

結果例:

```
HTTP/1.1 202 Accepted
X-Zfssa-Nas-API: 1.0
Content-Type: application/json
Content-Length: 90

{
  "property": {
    "href": "/api/storage/v1/schema/priority",
    "type": "Integer",
    "description": "My custom priority level"
  }
}
```

プロパティの削除

スキーマプロパティを削除します。

リクエストの例:

```
DELETE /api/storage/v1/schema/me HTTP/1.1
```

結果例:

```
HTTP/1.1 204 No Content
```

レプリケーション

リモートレプリケーションは、アプライアンス間のプロジェクトとシェアのレプリケーションを容易にします。

注記 - レプリケーションは Oracle ZFS Storage Appliance の特定のモデルでライセンス付与された機能で、レプリケーション RESTful API はその機能を管理します。サービスは URI: <https://hostname:215/api/storage/v{1|2}/replication> から使用できます。ライセンスの詳細は、「オラクル社のソフトウェアライセンス契約書 (「SLA」) およびハードウェアシステムと組み込みのソフトウェアオプションの権利書」およびソフトウェアリリースのライセンス情報ユーザーマニュアルを参照してください。

レプリケーション RESTful API は次のリソースを管理します。

- **レプリケーションサービス** – レプリケーションタスクを管理するサービス。
- **レプリケーションターゲット** – 別のアプライアンスピア (ソース) からレプリケートされるデータを受け取って保存するアプライアンスピア。この用語は、別のアプライアンスへのレプリケーションを可能にする、アプライアンス上の構成オブジェクトを指すこともあります。
- **レプリケーションアクション** – プロジェクトまたはシェア、ターゲットアプライアンス、およびポリシーオプション (更新の送信頻度、ワイヤ上でのデータの暗号化の有無など) を指定する、ソースアプライアンス上の構成オブジェクト。
- **レプリケーションパッケージ** – ターゲット側でアクションに相当するもの。特定のソースの特定のアクションの一部としてレプリケートされるデータを管理する、ターゲットアプライアンス上の構成オブジェクト。ソースアプライアンス上の各アクションには、ターゲットアプライアンス上の正確に 1 つのパッケージが関連付けられます。逆も同様です。どちらかのオブジェクトが失われた場合は、アクションとパッケージのペア (および完全なレプリケーション更新) を新たに作成する必要があります。

この API では、レプリケーションアクションおよびレプリケーションパッケージに対するレプリケーション操作が提供されます。サービス API は、レプリケーションサービスとレプリケーションのソースおよびターゲットの管理に使用します。

表 55 レプリケーションサービスコマンド

リクエスト	パス /api/service/v{1 2}/services の後ろに追加	説明
GET	/replication	レプリケーションサービスの状態プロパティを取得します
PUT	/replication/enable	レプリケーションサービスを有効にします
PUT	/replication/disable	レプリケーションサービスを無効にします

レプリケーションサービスのプロパティの一覧表示

レプリケーションサービスの状態を取得します。

リクエストの例:

```
GET /api/service/v2/services/replication HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
```

```
{
  "service": {
    "href": "/api/service/v2/services/replication",
    "<status>": "online",
    "enable_start_finish_alerts": true
  }
}
```

レプリケーションサービスのプロパティーの変更

レプリケーションサービスの状態はその他のサービスと同様に変更できます。詳細は、サービス RESTful API を参照してください。

レプリケートしているプロジェクトの数とレプリケーションスケジュールの頻度によっては、定期更新の開始および終了アラートの数によってほかの重要なアラートがわかりにくくなる可能性があります。定期更新の開始および終了アラートを無効にするには、`enable_start_finish_alerts` プロパティーを `false` に設定します。

```
PUT /api/service/v2/services/replication
Host: zfs-storage.example.com:215
Content-Type: application/json

{ "enable_start_finish_alerts":false }
```

レプリケーションターゲット

次の表に、使用可能なレプリケーションターゲットのコマンドを示します。

表 56 レプリケーションターゲットコマンド

リクエスト	パス <code>/api/storage/v{1 2}</code> の後ろに追加	説明
POST	<code>/replication/targets</code>	新しいレプリケーションターゲットを作成します
GET	<code>/replication/targets/target</code>	指定されたレプリケーションターゲットプロパティーを取得します
GET	<code>/replication/targets</code>	すべてのレプリケーションターゲットオブジェクトを一覧表示します
PUT	<code>/replication/targets/target</code>	指定されたレプリケーションターゲットオブジェクトを変更します
DELETE	<code>/replication/targets/target</code>	指定されたターゲットオブジェクトを破棄します

次の表に、レプリケーションターゲットのプロパティーを示します。

プロパティ	説明
label	表示するターゲット名。
hostname	ターゲットアプライアンスの完全修飾ドメイン名または IPv4 アドレス。
host_match	ホスト名の検証を実行またはバイパスします。 204 ページの「ターゲット証明書の検証」 を参照してください。
auto_accept_cert	ターゲットの証明書を自動的に受け入れます。 204 ページの「ターゲット証明書の検証」 を参照してください。

レプリケーションターゲットの一覧表示

システム上の使用可能なレプリケーションターゲットをすべて一覧表示します。

リクエストの例:

```
GET /api/storage/v2/replication/targets HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
X-Zfssa-Replication-API: 1.0
Content-Type: application/json
Content-Length: 529

{
  "targets": [{
    "address": "ipaddr-1",
    "label": "zfs-storage-1",
    "hostname": "ipaddr-2",
    "asn": "9d7a7543-ca83-68f5-a8fc-f818f65e1cfc",
    "actions": 0,
    "target": "target-000",
    "href": "/api/storage/v2/replication/targets/zfs-storage-1"
  }, {
    "address": "ipaddr-3",
    "label": "zfs-storage-2",
    "hostname": "ipaddr-4",
    "asn": "16a4c82c-26c1-4a50-e317-ac53181f2e86",
    "actions": 0,
    "target": "target-001",
    "href": "/api/storage/v2/replication/targets/zfs-storage-2"
  }]
}
```

指定されたレプリケーションターゲットの一覧表示

このコマンドは、単一のレプリケーションターゲットの詳細を表示します。ターゲットはそのホスト名によってアクセスされます。

リクエストの例:

```
GET /api/storage/v2/replication/targets/zfs-storage-1 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
X-Zfssa-Replication-API: 1.0
Content-Type: application/json
Content-Length: 337

{
  "target": {
    "href": "/api/storage/v2/replication/targets/zfs-storage-1",
    "address": "ipaddr-1",
    "label": "zfs-storage-1",
    "hostname": "ipaddr-2",
    "asn": "9d7a7543-ca83-68f5-a8fc-f818f65e1cfc",
    "actions": 0
  }
}
```

レプリケーションターゲットの作成

`targets` コマンドは、リモートレプリケーション用に新しいレプリケーションターゲットを作成します。

レプリケーショントラフィックが特定のネットワークインタフェースを通過するようになる必要がある場合は、[111 ページの「ルートの追加」](#)に示すように、そのインタフェースを指定するターゲットの静的ルートを設定します。

リクエストの例:

`hostname` および `auto_accept_cert` プロパティの詳細は、[204 ページの「ターゲット証明書の検証」](#)を参照してください。

```
POST /api/replication/v2/targets HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 54
```

```
{
  "label": "zfs-storage-3",
  "hostname": "zfs-storage-3.example.com",
  "root_password": "root-password",
  "auto_accept_cert": true
}
```

結果例:

```
HTTP/1.1 201 Created
```

```
Content-Length: 135
Content-Type: application/json
Location: /service/v2/services/replication/targets/target-000
X-Zfssa-Replication-API: 1.0
{
  "target": {
    "actions": 0,
    "address": "123.45.78.9:216",
    "asn": "fa5bf303-0dcb-e20d-ac92-cd129ccd2c81",
    "auto_accept_cert": true,
    "hostname": "zfs-storage-3.example.com",
    "href": "/service/v2/services/replication/targets/target-000",
    "label": "zfs-storage-3"
  }
}
```

ターゲット証明書の検証

レプリケーションターゲットを作成すると、証明書の検証が実行されます。証明書の検証は、次のステップで構成されます。

1. 証明書ホスト名チェック
2. 証明書信頼チェック

ホスト名チェックまたは証明書信頼性チェックのいずれかが失敗した場合、ターゲットは作成されません。

ホスト名チェック

`hostname` プロパティの値は、完全修飾ドメイン名または IPv4 アドレスにすることができます。推奨される値は、ターゲットの完全修飾ドメイン名です。

ホスト名チェックは、ターゲットの `hostname` プロパティで指定されたホスト名が証明書で指定されたホストと一致することを確認します。`hostname` に IP アドレスまたは非修飾ドメイン名を指定し、証明書に完全修飾ドメイン名しかない場合、ホスト名チェックは失敗し、ターゲットは作成されません。

ターゲットが ASN ベースの証明書を使用している場合は、`hostname` プロパティの値にターゲットの完全修飾ドメイン名を指定します。

ホスト名チェックはデフォルトで実行されます。`host_match` プロパティを `false` に設定すると、ホスト名チェックは実行されません。

セキュリティを強化するには、`hostname` プロパティの値をターゲットの完全修飾ドメイン名に設定し、`host_match` プロパティが `true` に設定されていることを確認してください。

証明書信頼チェック

証明書信頼チェックでは、次の証明書のいずれかがソースの信頼できる証明書リストに追加され、ピアでの使用が有効になっていることを確認します。

- ターゲットアプライアンスの証明書。
- ターゲットアプライアンスの証明書を発行した認証局の証明書。

証明書が信頼されていない場合、HTTP ステータス 409 (Conflict) が返され、ターゲットは作成されません。次の例では、読みやすくするためにメッセージ行を改行しています。

```
{
  "fault": {
    "code": 409,
    "name": "ERR_ILLEGAL_STATE",
    "message": "operation failed due to illegal state (Certificate is not trusted
               (encountered while attempting to run command \"commit\"))"
  }
}
```

このソースに対してこのターゲットを初めて作成するとき、ターゲットホストの証明書が信頼されているかどうかはわかりません。RESTful API は証明書の確認を求めるプロンプトを表示できないため、`auto_accept_cert` プロパティを `true` に設定して、ターゲットの証明書を自動的に受け入れます。

ターゲットが作成された後、その証明書は信頼できなくなることがあります。たとえば、ソースの管理者が信頼できる証明書のリストから証明書を削除したり、ターゲットの管理者が証明書を置き換えたりすることができます。ターゲットを変更して、`auto_accept_cert` プロパティを `true` に設定し、この証明書を信頼できるものとして受け入れるようにソースに指示します。

証明書信頼チェックは、すべてのピアおよびレプリケーション接続に対して実行されます。証明書が信頼されていない場合、ソースは接続を拒否します。

フィンガープリントや発行者の `commonName`、`SubjectAltName` などの証明書のプロパティを確認したり、証明書をアップロードしたり、証明書を削除したりするには、[37 ページの「証明書の管理」](#)を参照してください。

レプリケーションターゲットの変更

このコマンドは、レプリケーションターゲットのプロパティを変更します。

リクエストの例:

```
PUT /api/replication/v2/targets/target-001 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 78

{ "hostname": "zfs-storage-3.example.com" }
```

レプリケーションターゲットの削除

このコマンドは、既存のレプリケーションターゲットを削除します。

リクエストの例:

```
DELETE /service/v2/services/replication/targets/target-000 HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
```

削除に成功すると、HTTP ステータス 204 (No Content) が返されます。

レプリケーションアクション

レプリケーションアクションは、レプリケーションターゲットにデータをレプリケートするための規則を定義します。次のコマンドは、レプリケーションアクションを管理します。

フラットアクションインタフェースの使用

プロジェクトまたはシェアを指定せずに、レプリケーションアクションを管理するようにアプライアンスに直接リクエストを行うことができます。

次の表には、レプリケーションアクションを管理するための基本コマンドを一覧表示します。

表 57 基本アクションインタフェース

リクエスト	パス <code>/api/storage/v{1 2}</code> の後ろに追加	説明
GET	<code>/replication/actions</code>	すべてのレプリケーションアクションオブジェクトを一覧表示します
GET	<code>/replication/actions/ra_id</code>	指定されたレプリケーションアクションプロパティを取得します
PUT	<code>/replication/actions/ra_id</code>	指定されたレプリケーションアクションオブジェクトを変更します
DELETE	<code>/replication/actions/ra_id</code>	指定されたレプリケーションアクションオブジェクトを削除します
PUT	<code>/replication/actions/ra_id/sendupdate</code>	選択したレプリケーションアクションを開始します
PUT	<code>/replication/actions/ra_id/cancelupdate</code>	選択したレプリケーションアクションを停止します

次の表に、レプリケーションアクションのスケジュールを管理するためのコマンドを一覧表示します。

表 58 アクションスケジュールへのアクセス

リクエスト	パス <code>/api/storage/v{1 2}</code> の後ろに追加	説明
GET	<code>/replication/actions/ra_id/schedules</code>	すべてのレプリケーションアクションスケジュールオブジェクトを一覧表示します
GET	<code>/replication/actions/ra_id/schedules/ra_schedule</code>	指定されたレプリケーションアクションスケジュールプロパティを取得します
POST	<code>/replication/actions/ra_id/schedules</code>	新しいレプリケーションアクションスケジュールを作成します
PUT	<code>/replication/actions/ra_id/schedules/ra_schedule</code>	指定されたレプリケーションアクションスケジュールオブジェクトを変更します
DELETE	<code>/replication/actions/ra_id/schedules/ra_schedule</code>	指定されたレプリケーションアクションスケジュールオブジェクトを削除します

次の表には、レプリケーションの自動スナップショットを管理するためのコマンドを一覧表示します。

注記 - プロジェクトレベルのレプリケーションアクション内に構成されているシェアレベルの自動スナップショットスケジュールには、次のコマンドを使用してアクセスできません。プロジェクトレベルのアクションには、複数のシェア内の複数の自動スナップショットスケジュールを含めることができますが、このインタフェースでは、すべての組み合わせを識別する最終的な方法が提供されていません。

表 59 レプリケーションの自動スナップショット構成へのアクセス

リクエスト	パス <code>/api/storage/v{1 2}</code> の後ろに追加	説明
GET	<code>/replication/actions/ra_id/autosnaps</code>	選択されたレプリケーションアクションの自動スナップショット構成を取得します
GET	<code>/replication/actions/ra_id/autosnaps/autosnaps_id</code>	指定されたレプリケーションアクションの自動スナップショットオブジェクトを取得します
PUT	<code>/replication/actions/ra_id/autosnaps</code>	指定されたレプリケーションアクションの自動スナップショットプロパティを変更します
PUT	<code>/replication/actions/ra_id/autosnaps/autosnaps_id</code>	指定されたレプリケーションアクションの自動スナップショットオブジェクトを変更します
DELETE	<code>/replication/actions/ra_id/autosnaps/autosnaps_id</code>	指定されたレプリケーションアクションの自動スナップショットオブジェクトを削除します

次の表に、レプリケーションアクション統計にアクセスするためのコマンドを一覧表示します。

表 60 レプリケーションアクション統計へのアクセス

リクエスト	パス <code>/api/storage/v{1 2}</code> の後ろに追加	説明
GET	<code>/replication/actions/ra_id/stats</code>	選択されたレプリケーションアクションの読み取り専用レプリケーション統計を取得します

プロジェクト、ファイルシステム、または LUN コンテキストでのレプリケーションアクション

特定のプロジェクト、ファイルシステム、または LUN のコンテキストで、レプリケーションアクションを管理するようにリクエストを行うこともできます。

次の表には、レプリケーションアクションを管理するための基本コマンドを一覧表示します。

- プロジェクトベースの操作の URI は次で始まります。

`/api/storage/v{1|2}/pools/pool/projects/project`

- ファイルシステムベースの操作の URI は次で始まります。

`/api/storage/v{1|2}/pools/pool/projects/project/filesystems/filesystem`

- LUN ベースの操作の URI は次で始まります。

`/api/storage/v{1|2}/pools/pool/projects/project/luns/lun`

レプリケーションアクションを管理するには、上記の目的のコンテキスト URI に次の基本コマンドを追加します。

表 61 プロジェクト、ファイルシステム、または LUN のレプリケーションアクションの基本インタフェース

リクエスト	上記のプロジェクト、ファイルシステム、または LUN の URI に追加	説明
GET	<code>/replication/actions</code>	すべてのレプリケーションアクションオブジェクトを一覧表示します
GET	<code>/replication/actions/ra_id</code>	指定されたレプリケーションアクションプロパティを取得します
POST	<code>/replication/actions</code>	新しいレプリケーションアクションを作成します
PUT	<code>/replication/actions/ra_id</code>	指定されたレプリケーションアクションオブジェクトを変更します

リクエスト	上記のプロジェクト、ファイルシステム、または LUN の URI に追加	説明
DELETE	/replication/actions/ra_id	指定されたレプリケーションアクションオブジェクトを削除します
PUT	/replication/actions/ra_id/sendupdate	選択したレプリケーションアクションを開始します
PUT	/replication/actions/ra_id/cancelupdate	選択したレプリケーションアクションを停止します

次の表に、レプリケーションスケジュールを管理するための基本コマンドを一覧表示します。

- プロジェクトベースの操作の URI は次で始まります。

/api/storage/v{1|2}/pools/pool/projects/project

- ファイルシステムベースの操作の URI は次で始まります。

/api/storage/v{1|2}/pools/pool/projects/project/filesystems/filesystem

- LUN ベースの操作の URI は次で始まります。

/api/storage/v{1|2}/pools/pool/projects/project/luns/lun

レプリケーションスケジュールを管理するには、上記の目的のコンテキスト URI に次の基本コマンドを追加します。

表 62 プロジェクト、ファイルシステム、または LUN のレプリケーションアクションのスケジュール

リクエスト	上記のプロジェクト、ファイルシステム、または LUN の URI に追加	説明
GET	/replication/actions/ra_id/schedules	すべてのレプリケーションアクションスケジュールオブジェクトを一覧表示します
GET	/replication/actions/ra_id/schedules/ra_schedule	指定されたレプリケーションアクションスケジュールプロパティを取得します
POST	/replication/actions/ra_id/schedules	新しいレプリケーションアクションスケジュールを作成します
PUT	/replication/actions/ra_id/schedules/ra_schedule	指定されたレプリケーションアクションスケジュールオブジェクトを変更します
DELETE	/replication/actions/ra_id/schedules/ra_schedule	指定されたレプリケーションアクションスケジュールオブジェクトを削除します

次の表に、レプリケーションの自動スナップショット構成を管理するための基本コマンドを一覧表示します。

- プロジェクトベースの操作の URI は次で始まります。

/api/storage/v{1|2}/pools/pool/projects/project

- ファイルシステムベースの操作の URI は次で始まります。

```
/api/storage/v{1|2}/pools/pool/projects/project/filesystems/filesystem
```

- LUN ベースの操作の URI は次で始まります。

```
/api/storage/v{1|2}/pools/pool/projects/project/luns/lun
```

レプリケーションの自動スナップショット構成を管理するには、上記の目的のコンテキスト URI に次の基本コマンドを追加します。

注記 - プロジェクトレベルのレプリケーションアクション内に構成されているシェアレベルの自動スナップショットスケジュールには、次のプロジェクトベースの操作を使用してアクセスできません。プロジェクトレベルのアクションには、複数のシェア内の複数の自動スナップショットスケジュールを含めることができますが、このインタフェースでは、すべての組み合わせを識別する最終的な方法が提供されていません。

表 63 プロジェクト、ファイルシステム、または LUN レプリケーションの自動スナップショット構成

リクエスト	上記のプロジェクト、ファイルシステム、または LUN の URI に追加	説明
GET	/replication/actions/ra_id/autosnaps	プロジェクトまたはシェアの選択されたレプリケーションアクションの自動スナップショット構成を取得します
GET	/replication/actions/ra_id/autosnaps/autosnaps_id	プロジェクトまたはシェアの指定されたレプリケーションアクションの自動スナップショット構成を取得します
POST	/replication/actions/ra_id/autosnaps	新しいプロジェクトまたはシェアレベルのレプリケーションアクションの自動スナップショットオブジェクトを作成します
PUT	/replication/actions/ra_id/autosnaps	プロジェクトまたはシェアの指定されたレプリケーションアクションのターゲット自動スナップショット保持ポリシーを変更します。
PUT	/replication/actions/ra_id/autosnaps/autosnaps_id	指定されたレプリケーションアクションの自動スナップショットオブジェクトを変更します
DELETE	/replication/actions/ra_id/autosnaps/autosnaps_id	指定されたレプリケーションアクションの自動スナップショットオブジェクトを削除します

次の表に、レプリケーションアクション統計にアクセスするための基本コマンドを一覧表示します。

- プロジェクトベースの操作の URI は次で始まります。

```
/api/storage/v{1|2}/pools/pool/projects/project
```

- ファイルシステムベースの操作の URI は次で始まります。

```
/api/storage/v{1|2}/pools/pool/projects/project/filesystems/filesystem
```

- LUN ベースの操作の URI は次で始まります。

```
/api/storage/v{1|2}/pools/pool/projects/project/luns/lun
```

レプリケーションアクション統計にアクセスするには、上記の目的のコンテキスト URI に次の基本コマンドを追加します。

表 64 レプリケーションアクション統計へのアクセス

リクエスト	上記のプロジェクト、ファイルシステム、または LUN の URI に追加	説明
GET	/replication/actions/ra_id/stats	選択されたレプリケーションアクションの読み取り専用レプリケーション統計を取得します

レプリケーションアクションの一覧表示

すべての使用可能なレプリケーションアクションのリストを取得します。

リクエストの例:

```
GET /api/storage/v2/replication/actions HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
X-Zfssa-Replication-API: 1.0
Content-Type: application/json
Content-Length: 529
```

```
{
  "actions": [{
    "href": ""
    ...
  }, {
    "href": "",
    ...
  }]
}
```

レプリケーションアクションの取得

レプリケーションアクションステータスの取得コマンドは、ID で指定された単一のレプリケーションアクションのステータスを返します。

リクエストの例:

```
GET /api/storage/v2/replication/actions/1438ed7f-aad3-c631-d869-9e85cd7f15b4 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
X-Zfssa-Replication-API: 1.0
Content-Type: application/json
Content-Length: 529

{
  "action": {
    "average_throughput": 0.0,
    "bytes_sent": 0.0,
    "collection": "local",
    "compression": true,
    "continuous": false,
    "enabled": true,
    "estimated_size": 0.0,
    "estimated_time_left": 0.0,
    "href": "/api/storage/v2/replication/actions",
    "id": "8373d331-de60-e590-90e8-9ad69fcb4aec",
    "include_clone_origin_as_data": false,
    "include_snaps": true,
    "last_sync": "20130916T21:36:50",
    "last_try": "20130916T21:36:50",
    "max_bandwidth": 0,
    "pool": "p1",
    "project": "proj-01",
    "retain_user_snaps_on_target": false,
    "share": "fs1",
    "state": "sending",
    "target": "38094753-6c90-49ed-aa92-995a296d432a",
    "use_ssl": true
  }
}
```

リクエストの例:

次のレプリケーションアクションの応答は、回復ポイント目標 (RPO) の例と関連するレプリカ遅延警告およびアラートを示しています。

```
GET /api/storage/v2/replication/actions HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
```

結果例:

```
HTTP/1.1 200 OK
X-Zfssa-Replication-API: 1.0
Content-Type: application/json
Content-Length: 529

{
  "action": {
    "id": "12d981c3-b098-c65a-e1e9-a6b8263a0f6a",
    "target_id": "4fd305ac-4af5-c34a-87c3-88203207305b",
    ...
    "replica_lag": "42:25:31",
  }
}
```

```

        "recovery_point_objective": 0,
        "replica_lag_warning_alert": 0,
        "replica_lag_error_alert": 0,
        "replica_lag_over_warning_limit": false,
        "replica_lag_over_error_limit": false,
        "project": "default"
    }
}

```

レプリケーションアクションの作成

新しいレプリケーションアクションを作成します。

プロパティの作成:

```

Initial values:
    target = cleo
    enabled = true
    continuous = false
    include_snaps = true
    retain_user_snaps_on_target = false
    dedup = true
    include_clone_origin_as_data = false
    max_bandwidth = unlimited
    bytes_sent = 0
    estimated_size = 0
    estimated_time_left = 0
    average_throughput = 0
    use_ssl = true
    compression = on

```

リクエストの例:

```

POST /api/storage/v2/replication/actions HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Content-Length: 121
Accept: application/json

```

```

{
  "pool": "p1",
  "project": "proj-01",
  "share": "fs1",
  "target_pool": "pool1",
  "target": "38094753-6c90-49ed-aa92-995a296d432a"
}

```

結果例:

```

HTTP/1.1 201 Created
Content-Length: 506
Content-Type: application/json
Location: /api/storage/v2/replication/action/8373d331-de60-e590-90e8-9ad69fcb4aec
X-Zfssa-Replication-API: 1.0

```

```

{
  "action": {

```

```

    "project": "blue1",
    "target": "38094753-6c90-49ed-aa92-995a296d432a",
    "bytes_sent": 0.0,
    "compression": true,
    "continuous": false,
    "enabled": true,
    "dedup": false,
    "max_bandwidth": 0,
    "collection": "local",
    "estimated_size": 0.0,
    "state": "idle",
    "href": "/api/storage/v2/replication/pools/p1/projects/blah1/shares/fs1/
            actions/8373d331-de60-e590-90e8-9ad69fcb4aec",
    "average_throughput": 0.0,
    "use_ssl": true,
    "estimated_time_left": 0.0,
    "retain_user_snaps_on_target": false,
    "share": "fs1",
    "id": "8373d331-de60-e590-90e8-9ad69fcb4aec",
    "pool": "p1",
    "include_clone_origin_as_data": false,
    "include_snaps": true
  }
}

```

レプリケーションアクションのスケジュールを作成します。

リクエストの例:

```

POST /api/storage/v2/replication/actions/b77bd8cd-17ed-69da-9e4b-aebe3cc63755/schedules
HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
Content-Type: application/json
Content-Length: 65

{"frequency": "month", "day": "5th", "hour": "auto", "minute": "auto"}

```

結果例:

```

HTTP/1.1 201 Created
Date: Thu, 12 Jan 2017 17:35:48 GMT
Server: TwistedWeb/192.0.2
Content-Length: 0
X-Zfssa-Storage-API: 1.1
Content-Type: application/json; charset=utf-8
X-Zfssa-API-Version: 1.0
X-Zfssa-Version: user/generic@2016.12.08,1-0

```

レプリケーションアクションの変更

既存のレプリケーションアクションを変更します。

リクエストの例:

```

PUT /api/storage/v2/replication/actions/c141d88d-ffd2-6730-d489-b71905f340cc HTTP/1.1

```

```
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
```

```
{"use_ssl": false}
```

結果例:

```
HTTP/1.1 202 Accepted
X-Zfssa-Replication-API: 1.0
Content-Type: application/json
Content-Length: 620
```

```
{
  "action": {
    "target_id": "407642ae-91b5-681c-de5e-afcd5cbf2974",
    "compression": true,
    "continuous": false,
    "enabled": true,
    "max_bandwidth": 0,
    "dedup": false,
    "retain_user_snaps_on_target": false,
    "use_ssl": false,
    "id": "c141d88d-ffd2-6730-d489-b71905f340cc",
    "include_clone_origin_as_data": false,
    "include_snaps": true
  }
}
```

リクエストの例:

```
PUT /api/storage/v2/replication/actions/action_id -d '{"recovery_point_objective":
60}'HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
```

結果例:

```
X-Zfssa-Replication-API: 1.0
Content-Type: application/json
Content-Length: 620
```

```
{
  "action": {
    "state_description": "Idle (no update in progress)",
    "recovery_point_objective": 60,
    "replica_lag_over_warning_limit": false,
    "bytes_sent": "0",
    "last_try": "Mon Nov 21 2016 23:25:59 GMT+0000 (UTC)",
    "max_bandwidth": 0,
    "estimated_size": "0",
    "href": "/api/storage/v2/replication/actions/12d981c3-b098-c65a-e1e9-a6b8263a0f6a",
    "estimated_time_left": 0,
    "use_ssl": true,
    "id": "12d981c3-b098-c65a-e1e9-a6b8263a0f6a",
    "stats": {"total_logical_bytes": 40656,
    "last_dd_table_build": 9169029,
    "total_after_dedup": 18476,
    "last_try": "Mon Nov 21 2016 23:25:59 GMT+0000 (UTC)",
    "dd_total_updates": 1,
    "href": "/api/storage/v2/replication/actions/12d981c3-b098-c65a-e1e9-a6b8263a0f6a/"
  },
  "stats",
```

```

        "dd_total_duration": 47149245470,
        "last_logical_bytes": 40656,
        "dd_total_table_mem": 2097152,
        "last_result": "success",
        "last_after_dedup": 18476,
        "last_duration": 47149245470,
        {"dd_total_logical_bytes": 40656,
         "total_updates": 1,
         "total_duration": 47149245470,
         "replica_data_timestamp": "Mon Nov 21 2016 23:25:12 GMT+0000 (UTC)",
         "total_to_network": 9623,
         "dd_total_table_build": 9169029,
         "dd_total_phys_bytes": 16800,
         "last_to_network": 9623,
         "total_phys_bytes": 16800,
         "last_phys_bytes": 16800,
         "last_sync": "Mon Nov 21 2016 23:25:59 GMT+0000 (UTC)",
         "last_dd_table_mem": 2097152,
         "dd_total_after_dedup": 18476,
         "dd_total_to_network": 9623},
        "compression": "on",
        "dedup": true,
        "replica_lag_warning_alert": 0,
        "last_result": "success",
        "include_clone_origin_as_data": false,
        "state": "idle",
        "offline": false,
        "export_path": "",
        "export_pending": false,
        "autosnaps": {"autosnaps_retention_policies":
        "synchronized",
        "href": "/api/storage/v2/replication/actions/12d981c3-b098-c65a-e1e9-a6b8263a0f6a/
autosnaps"},
        "replica_data_timestamp": "Mon Nov 21 2016 23:25:12 GMT+0000 (UTC)",
        "continuous": false,
        "target_id": "4fd305ac-4af5-c34a-87c3-88203207305b",
        {"average_throughput": "0B/s",
         "next_update": "Sync now",
         "pool": "p1",
         "replica_lag_over_error_limit": false,
         "target": "pool1",
         "replica_lag": "42:28:24",
         "retain_user_snaps_on_target": false,
         ...
        }
    }
}

```

レプリケーションアクションの進行状況のモニター

レプリケーションアクションステータスの取得コマンドは、ID で指定された単一のレプリケーションアクションのステータスを返します。レプリケーションの進行状況を確認するには、state および state_description を調べます。

state プロパティの値は次のとおりです。

- sending
- idle

state_description プロパティの値は次のとおりです。

- Connecting to replication target
- Receiving checkpoint from target
- Estimating size of update
- Building deduplication tables

このプロパティ値は、複製解除されたレプリケーションストリームでのみ使用されます。

リクエストの例:

```
GET /api/storage/v2/replication/actions/1438ed7f-aad3-c631-d869-9e85cd7f15b4 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
X-Zfssa-Replication-API: 1.0
Content-Type: application/json
Content-Length: 529

{
  "action": {
    "id": "1438ed7f-aad3-c631-d869-9e85cd7f15b4",
    "target_id": "4fd3483e-b1f5-4bdc-9be3-b3a4becd0c42",
    "target": "cleo",
    "pool": "p0",
    "replication_of": "testproj",
    "enabled": true,
    "continuous": false,
    "include_snaps": true,
    "retain_user_snaps_on_target": false,
    "dedup": true,
    "include_clone_origin_as_data": false,
    "max_bandwidth": -1,
    "bytes_sent": 0,
    "estimated_size": 0,
    "estimated_time_left": 0,
    "average_throughput": 0,
    "use_ssl": true,
    "compression": "on",
    "export_path": "",
    "state": "sending",
    "state_description": "Receiving checkpoint from target",
    "export_pending": false,
    "offline": false,
    "next_update": "Sync now",
    "replica_data_timestamp": "Thu Apr 28 2016 22:38:03 GMT+0000 (UTC)",
    "last_sync": "<unknown>",
    "last_try": "<unknown>",
    "last_result": "<unknown>",
    "replica_lag": "00:00:18",
    "recovery_point_objective": 0,
    "replica_lag_warning_alert": 0,
    "replica_lag_error_alert": 0,
    "replica_lag_over_warning_limit": false,
    "replica_lag_over_error_limit": false,
    "project": "testproj"
  }
}
```

```
}  
}
```

更新の取り消し

進行中のレプリケーション更新を取り消します。

リクエストの例:

```
PUT /api/storage/v2/replication/actions/c141d88d-ffd2-6730-d489-b71905f340cc/cancelupdate  
HTTP/1.1  
Host: zfs-storage.example.com:215  
Authorization: Basic Tm8gcGVla2luZyE=
```

結果例:

```
HTTP/1.1 202 Accepted  
X-Zfssa-Replication-API: 1.0
```

更新の送信

レプリケーション更新を可能なかぎりすぐを開始するようにスケジュールします。

リクエストの例:

```
PUT /api/storage/v2/replication/actions/c141d88d-ffd2-6730-d489-b71905f340cc/sendupdate  
HTTP/1.1  
Authorization: Basic Tm8gcGVla2luZyE=
```

結果例:

```
HTTP/1.1 202 Accepted  
X-Zfssa-Replication-API: 1.0
```

レプリケーションアクションの削除

既存のレプリケーションアクションを削除します。

リクエストの例:

```
DELETE /api/storage/v2/replication/actions/e7e688b1-ff07-474f-d5cd-cac08293506e HTTP/1.1  
Host: zfs-storage.example.com:215  
Authorization: Basic Tm8gcGVla2luZyE=
```

削除に成功すると、HTTP ステータス 204 (No Content) が返されます。

結果例:

```
HTTP/1.1 204 No-Content  
X-Zfssa-Replication-API: 1.0
```

レプリケーションパッケージ

このセクションでは、レプリケーションパッケージおよびソースのコマンドについて詳しく説明します。

表 65 レプリケーションパッケージのコマンド

リクエスト	パス <code>/api/storage/v{1 2}/replication</code> の後ろに追加	説明
GET	<code>/packages</code>	すべてのレプリケーションパッケージを一覧表示します
GET	<code>/packages/package</code>	指定されたレプリケーションパッケージを取得します
PUT	<code>/packages/package</code>	指定されたレプリケーションパッケージを変更します
DELETE	<code>/packages/package</code>	指定されたレプリケーションパッケージを破棄します
PUT	<code>/packages/package/cancelupdate</code>	指定されたパッケージで <code>cancelupdate</code> を実行します
PUT	<code>/packages/package/sever</code>	指定されたパッケージで切断を実行します
PUT	<code>/packages/package/pkgreverse</code>	指定されたパッケージでの逆方向化を実行します
PUT	<code>/packages/package/clone</code>	指定されたパッケージのクローンを作成します
GET	<code>/packages/package/clone/conflicts</code>	シェアプロパティの競合を一覧表示します
GET	<code>/packages/package/projects</code>	パッケージプロジェクトを一覧表示します
GET	<code>/packages/package/projects/project</code>	パッケージプロジェクトを取得します
PUT	<code>/packages/package/projects/project</code>	パッケージプロジェクトを変更します
GET	<code>/packages/package/projects/project/usage/groups</code>	パッケージプロジェクトグループの使用状況を取得します
GET	<code>/packages/package/projects/project/usage/users</code>	パッケージプロジェクトユーザーの使用状況を取得します
GET	<code>/packages/package/projects/project/snapshots</code>	すべてのスナップショットオブジェクトを一覧表示します
GET	<code>/packages/package/projects/project/snapshots/snapshot</code>	指定されたスナップショットプロパティを取得します
DELETE	<code>/packages/package/projects/project/snapshots/snapshot</code>	指定されたスナップショットオブジェクトを破棄します
PUT	<code>/packages/package/projects/project/snapshots/snapshot</code>	パッケージプロジェクトスナップショットの名前を変更します

レプリケーションパッケージ

リクエスト	パス <i>/api/storage/v{1 2}/replication</i> の後ろに追加	説明
GET	<i>/packages/package/projects/project/automatic</i>	すべてのパッケージプロジェクト自動スナップショットオブジェクトを一覧表示します
GET	<i>/packages/package/projects/project/automatic/automatic</i>	指定されたパッケージプロジェクトの自動スナップショットプロパティを取得します
GET	<i>/packages/package/projects/project/filesystems</i>	パッケージファイルシステムを一覧表示します
GET	<i>/packages/package/projects/project/filesystems/filesystem</i>	パッケージファイルシステムを取得します
PUT	<i>/packages/package/projects/project/filesystems/filesystem</i>	パッケージファイルシステムを変更します
GET	<i>/packages/package/projects/project/filesystems/filesystem/usage/groups</i>	パッケージファイルシステムグループの使用状況を取得します
GET	<i>/packages/package/projects/project/filesystems/filesystem/usage/users</i>	パッケージファイルシステムユーザーの使用状況を取得します
GET	<i>/packages/package/projects/project/filesystems/filesystem/snapshots/snapshot</i>	指定されたスナップショットプロパティを取得します
GET	<i>/packages/package/projects/project/filesystems/filesystem/snapshots</i>	すべてのスナップショットオブジェクトを一覧表示します
DELETE	<i>/packages/package/projects/project/filesystems/filesystem/snapshots/snapshot</i>	指定されたスナップショットオブジェクトを破棄します
PUT	<i>/packages/package/projects/project/filesystems/filesystem/snapshots/snapshot</i>	パッケージファイルシステムスナップショットの名前を変更します
GET	<i>/packages/package/projects/project/filesystems/filesystem/automatic</i>	すべてのパッケージファイルシステムの自動スナップショットオブジェクトを一覧表示します
GET	<i>/packages/package/projects/project/filesystems/filesystem/automatic/automatic</i>	指定されたパッケージファイルシステムの自動スナップショットプロパティを取得します
GET	<i>/packages/package/projects/project/luns</i>	パッケージ LUN を一覧表示します
GET	<i>/packages/package/projects/project/luns/lun</i>	パッケージ LUN を取得します
PUT	<i>/packages/package/projects/project/luns/lun</i>	パッケージ LUN を変更します
GET	<i>/packages/package/projects/project/luns/lun/usage/groups</i>	パッケージ LUN グループの使用状況を取得します
GET	<i>/packages/package/projects/project/luns/lun/usage/users</i>	パッケージ LUN ユーザーの使用状況を取得します
GET	<i>/packages/package/projects/project/luns/lun/snapshots/snapshot</i>	指定されたスナップショットプロパティを取得します
GET	<i>/packages/package/projects/project/luns/lun/snapshots</i>	すべてのスナップショットオブジェクトを一覧表示します
DELETE	<i>/packages/package/projects/project/luns/lun/snapshots/snapshot</i>	指定されたスナップショットオブジェクトを破棄します

リクエスト	パス <code>/api/storage/v{1 2}/replication</code> の後ろに追加	説明
PUT	<code>/packages/package/projects/project/luns/lun/snapshots/snapshot</code>	パッケージ LUN スナップショットの名前を変更します
GET	<code>/packages/package/projects/project/luns/lun/automatic</code>	すべてのパッケージ LUN の自動スナップショットオブジェクトを一覧表示します
GET	<code>/packages/package/projects/project/luns/lun/automatic/automatic</code>	指定されたパッケージ LUN の自動スナップショットプロパティを取得します

次のコマンドを使用して、レプリケーションソースおよびそれに対応するパッケージにアクセスすることもできます。

表 66 レプリケーションソースのコマンド

リクエスト	パス <code>/api/storage/v{1 2}/replication/sources</code> の後ろに追加	説明
GET	<code>/api/storage/v{1 2}/replication/sources</code> のみを使用	レプリケーションソースを一覧表示します
GET	<code>/source</code>	レプリケーションソースの詳細を一覧表示します
GET	<code>/source/packages/package</code>	指定されたレプリケーションパッケージを取得します
PUT	<code>/source/packages/package</code>	指定されたレプリケーションパッケージを変更します
DELETE	<code>/source/packages/package</code>	指定されたレプリケーションパッケージを破棄します
PUT	<code>/source/packages/package/cancelupdate</code>	指定されたパッケージで <code>cancelupdate</code> を実行します
PUT	<code>/source/packages/package/sever</code>	指定されたパッケージで切断を実行します
PUT	<code>/source/packages/package/pkgreverse</code>	指定されたパッケージでの逆方向化を実行します
PUT	<code>/source/packages/package/clone</code>	指定されたパッケージのクローンを作成します
GET	<code>/source/packages/package/clone/conflicts</code>	シェアプロパティの競合を一覧表示します
GET	<code>/source/packages/package/projects</code>	パッケージプロジェクトを一覧表示します
GET	<code>/source/packages/package/projects/project</code>	パッケージプロジェクトを取得します
PUT	<code>/source/packages/package/projects/project</code>	パッケージプロジェクトを変更します
GET	<code>/source/packages/package/projects/project/usage/groups</code>	パッケージプロジェクトグループの使用状況を取得します

レプリケーションパッケージ

リクエスト	パス <i>/api/storage/v{1 2}/replication/sources</i> の後ろに追加	説明
GET	<i>/source/packages/package/projects/project/usage/users</i>	パッケージプロジェクトユーザーの使用状況を取得します
GET	<i>/source/packages/package/projects/project/snapshots/snapshot</i>	指定されたスナップショットプロパティを取得します
GET	<i>/source/packages/package/projects/project/snapshots</i>	すべてのスナップショットオブジェクトを一覧表示します
DELETE	<i>/source/packages/package/projects/project/snapshots/snapshot</i>	指定されたスナップショットオブジェクトを破棄します
PUT	<i>/source/packages/package/projects/project/snapshots/snapshot</i>	パッケージプロジェクトスナップショットの名前を変更します
GET	<i>/source/packages/package/projects/project/automatic</i>	すべてのパッケージプロジェクト自動スナップショットオブジェクトを一覧表示します
GET	<i>/source/packages/package/projects/project/automatic/automatic</i>	指定されたパッケージプロジェクトの自動スナップショットプロパティを取得します
GET	<i>/source/packages/package/projects/project/filesystems</i>	パッケージファイルシステムを一覧表示します
GET	<i>/source/packages/package/projects/project/filesystems/filesystem</i>	パッケージファイルシステムを取得します
PUT	<i>/source/packages/package/projects/project/filesystems/filesystem</i>	パッケージファイルシステムを変更します
GET	<i>/source/packages/package/projects/project/filesystems/filesystem/usage/groups</i>	パッケージファイルシステムグループの使用状況を取得します
GET	<i>/source/packages/package/projects/project/filesystems/filesystem/usage/users</i>	パッケージファイルシステムユーザーの使用状況を取得します
GET	<i>/source/packages/package/projects/project/filesystems/filesystem/snapshots/snapshot</i>	指定されたスナップショットプロパティを取得します
GET	<i>/source/packages/package/projects/project/filesystems/filesystem/snapshots</i>	すべてのスナップショットオブジェクトを一覧表示します
DELETE	<i>/source/packages/package/projects/project/filesystems/filesystem/snapshots/snapshot</i>	指定されたスナップショットオブジェクトを破棄します
PUT	<i>/source/packages/package/projects/project/filesystems/filesystem/snapshots/snapshot</i>	パッケージファイルシステムスナップショットの名前を変更します
GET	<i>/source/packages/package/projects/project/filesystems/filesystem/automatic</i>	すべてのパッケージファイルシステムの自動スナップショットオブジェクトを一覧表示します
GET	<i>/source/packages/package/projects/project/filesystems/filesystem/automatic/automatic</i>	指定されたパッケージファイルシステムの自動スナップ

リクエスト	パス <code>/api/storage/v{1 2}/replication/sources</code> の後ろに追加	説明
		ショットプロパティを取得します
GET	<code>/source/packages/package/projects/project/luns</code>	パッケージ LUN を一覧表示します
GET	<code>/source/packages/package/projects/project/luns/lun</code>	パッケージ LUN を取得します
PUT	<code>/source/packages/package/projects/project/luns/lun</code>	パッケージ LUN を変更します
GET	<code>/source/packages/package/projects/project/luns/lun/usage/groups</code>	パッケージ LUN グループの使用状況を取得します
GET	<code>/source/packages/package/projects/project/luns/lun/usage/users</code>	パッケージ LUN ユーザーの使用状況を取得します
GET	<code>/source/packages/package/projects/project/luns/lun/snapshots/snapshot</code>	指定されたスナップショットプロパティを取得します
GET	<code>/source/packages/package/projects/project/luns/lun/snapshots</code>	すべてのスナップショットオブジェクトを一覧表示します
DELETE	<code>/source/packages/package/projects/project/luns/lun/snapshots/snapshot</code>	指定されたスナップショットオブジェクトを破棄します
PUT	<code>/source/packages/package/projects/project/luns/lun/snapshots/snapshot</code>	パッケージ LUN スナップショットの名前を変更します
GET	<code>/source/packages/package/projects/project/luns/lun/automatic</code>	すべてのパッケージ LUN の自動スナップショットオブジェクトを一覧表示します
GET	<code>/source/packages/package/projects/project/luns/lun/automatic/automatic</code>	指定されたパッケージ LUN の自動スナップショットプロパティを取得します

レプリケーションソースの一覧表示

すべての使用可能なレプリケーションソースを一覧表示します。

リクエストの例:

```
GET /api/storage/v2/replication/sources HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

出力例:

```
HTTP/1.1 200 OK
X-Zfssa-Replication-API: 1.0
Content-Type: application/json
Content-Length: 529

{
  "sources": [{
    "asn": "314d252e-c42b-e844-dab1-a3bca680b563",
    "href": "/api/storage/v2/replication/sources/zfs-repl-host",
```

```

        "ip_address": "ipaddr-1",
        "name": "zfs-repl-host",
        "source": "source-000"
    }
}

```

レプリケーションパッケージの一覧表示

すべてのレプリケーションパッケージを一覧表示します。

リクエストの例:

```

GET /api/storage/v2/replication/packages HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json

```

結果例:

```

HTTP/1.1 200 OK
X-Zfssa-Replication-API: 1.0
Content-Type: application/json
Content-Length: 529

{
  "packages": [
    {
      "href": "/api/storage/v2/replication/packages/0efaab49-7b22-4d4a-def8-813c27780894",
      "id": "0efaab49-7b22-4d4a-def8-813c27780894",
      "source_name": "sourceA",
      "source_asn": "8a22f6e0-4ee4-4b85-f141-e152f5fac961",
      "source_ip": "ipaddr-1",
      "source_pool": "poolA",
      "target_pool": "poolA",
      "replica_of": "projTest",
      "enabled": true,
      "state": "idle",
      "state_description": "Idle (no update in progress)",
      "offline": false,
      "import_path": "",
      "data_timestamp": "2017-03-09T22:36:12Z",
      "last_sync": "2017-03-09T22:36:22Z",
      "last_try": "2017-03-09T22:36:22Z",
      "last_result": "success"
    }
  ]
}

```

パッケージの変更

パッケージプロパティを変更します。

プロパティ	型	説明
enabled	boolean	レプリケーション更新の現在の状態

リクエストの例:

```
PUT /api/storage/v2/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json

{"enabled": false}
```

結果例:

```
HTTP/1.1 202 Accepted
X-Zfssa-Replication-API: 1.0
```

リクエストの例:

```
PUT /api/storage/v2/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec/pkgreverse
  HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json

{"new_project_name":"restrev", "enable_action_upon_reversal":"true"}
```

結果例:

```
HTTP/1.1 202 Accepted
X-Zfssa-Replication-API: 1.0
```

パッケージの削除

レプリケーションパッケージを破棄します。

リクエストの例:

```
DELETE /api/storage/v2/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
```

削除に成功すると、HTTP ステータス 204 (No Content) が返されます。

結果例:

```
HTTP/1.1 204 No-Content
X-Zfssa-Replication-API: 1.0
```

更新の取り消し

このパッケージに対する進行中の更新を取り消します。

リクエストの例:

```
PUT /api/storage/v2/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec/cancelupdate
  HTTP/1.1
```

```
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
```

進行中の更新がない場合、HTTP ステータス 409 (Conflict) が返されます。

結果例:

```
HTTP/1.1 409 Conflict
X-Zfssa-Replication-API: 1.0
Content-Type: application/json
Content-Length: 137

{
  "cancelupdate": {
    "AKSH_ERROR": "EAK_NAS_REPL_BADSTATE",
    "message": "operation illegal for state"
  }
}
```

結果例:

```
HTTP/1.1 202 Accepted
X-Zfssa-Replication-API: 1.0
```

パッケージのクローン作成

パッケージプロジェクトのクローンを作成します。

リクエストの例:

```
PUT /api/v2/storage/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec/clone
HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
```

結果例:

```
HTTP/1.1 202 Accepted
X-Zfssa-Replication-API: 1.0
```

クローン作成に成功すると、HTTP ステータス 202 (Accepted) が返されます。ヘルパーコマンドを使用すると、クローン操作との競合が発生しているかどうかを確認できます。

クローンの競合リクエストの例:

```
GET /api/storage/v2/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec/clone/
conflicts HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
```

クローンまたは競合が競合を返します。

```
HTTP/1.1 200 OK
X-Zfssa-Replication-API: 1.0
Content-Type: application/json
```

Content-Length: 58

```
{
  "conflicts": "There are no conflicts."
}
```

プロパティー:

```
Default settings:
  target_project = (unset)
  original_mountpoint = /export
  override_mountpoint = false
  mountpoint =
```

パッケージの切断

レプリケーション接続を切断し、パッケージの内容を新しいプロジェクトに移動します。このアクションは、このパッケージとそのレプリケートされたシェアをソースシステムから完全に切断し、このシステム上のローカルプロジェクトとします。その後のどちらの方向のレプリケーション更新でも、新しいアクションを定義し、完全な更新を送信することが必要になります。

リクエストの例:

```
PUT /api/storage/v2/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec/sever
HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=

{"projname":"restsev"}
```

成功のレスポンス:

```
HTTP/1.1 202 Accepted
X-Zfssa-Replication-API: 1.0
```

パッケージの逆方向化

レプリケーションの方向を逆にします。このアクションはこのパッケージのレプリケーションを無効にし、このパッケージの内容を、ソースにレプリケートして戻すように構成された新しいローカルプロジェクトに移動します。最後の成功した更新以降にソースに対して行われたメタデータまたはデータの変更は、新しいプロジェクトが最初にソースにレプリケートされて戻されたときに失われます。

リクエストの例:

```
PUT /api/storage/v2/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec/reverse
HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
```

```
{"projname":"restrev"}
```

成功のレスポンス:

```
HTTP/1.1 202 Accepted  
X-Zfssa-Replication-API: 1.0
```

ストレージの暗号化

プール、プロジェクト、およびシェアは暗号化できます。プールが暗号化されている場合、各子プロジェクトとシェアは暗号化され、プールから暗号化プロパティ値を継承します。プロジェクトに暗号化プロパティ値が指定されている場合、継承された値は使用されず、子シェアはプロジェクトに指定されたローカル値を継承します。暗号化されたプロジェクトは、暗号化されていないプールに作成できます。暗号化されていないプロジェクトは、暗号化されたプールに作成できません。

プロジェクトが暗号化されている場合、各子シェアは暗号化され、プロジェクトから暗号化プロパティ値を継承します。シェアに暗号化プロパティ値が指定されている場合、継承された値は使用されません。暗号化されたシェアは、暗号化されていないプロジェクトに作成できます。暗号化されていないシェアは、暗号化されたプロジェクトに作成できません。

暗号化鍵の管理

注記 - 暗号化は、特定のモデルに対してライセンス付与された機能です。詳細は、「オラクル社のソフトウェアライセンス契約書(「SLA」)およびハードウェアシステムと組み込みのソフトウェアオプションの権利書」およびソフトウェアリリースのライセンス情報ユーザーマニュアルを参照してください。

Oracle ZFS Storage Appliance は、プール、プロジェクト、および個々のシェア(ファイルシステムおよび LUN) に対する透過的なデータ暗号化を提供します。アプライアンスには組み込みのローカルキーストアが含まれており、Oracle Key Manager (OKM) と Key Management Interoperability Protocol (KMIP) 暗号化もサポートされています。各暗号化プロジェクトまたはシェアでは、ローカルキーストア、OKM キーストアまたは KMIP キーストアからのラッピング鍵が必要とされます。データ暗号化鍵はストレージアプライアンスによって管理され、ラッピング鍵によって暗号化された状態で永続的に保存されます。

暗号化されたプール、プロジェクト、またはシェアを作成する前に、暗号化鍵を作成する必要があります。

- キーストアはプールの作成前に構成する必要があるため、初期システム構成時または出荷時リセット後に暗号化プールを作成することはできません。

- 暗号化プールでシェアまたはプロジェクトのレプリケーションを設定する前に、ソースで使用する暗号化鍵がターゲットでも使用できることを確認してください。

次の表では、暗号化鍵を管理するために使用できる RESTful API リクエストについて説明します。表では、*keystore* の値は *local*、*okm* または *kmip* です。*key* の値は *keyname* プロパティの値です。鍵のプロパティについては、[233 ページの「暗号化鍵の一覧表示」](#)で説明されています。

表 67 暗号化鍵の操作

リクエスト	パス <i>/api/storage/v{1 2}</i> の後ろに追加	説明
GET	<i>/encryption/keystore</i>	すべての <i>keystore</i> プロパティを一覧表示します
PUT	<i>/encryption/keystore</i>	<i>keystore</i> プロパティを変更します。
GET	<i>/encryption/keystore/keys</i>	すべての <i>keystore</i> 鍵を一覧表示します。
GET	<i>/encryption/keystore/keys/key</i>	指定された鍵の詳細を取得します。
POST	<i>/encryption/keystore/keys</i>	鍵を作成します。
DELETE	<i>/encryption/keystore/keys/key</i>	鍵を破棄します。
GET	<i>/encryption/keystore/keys/key/dependents</i>	この鍵に依存しているシェアを一覧表示します

ローカルキーストアの構成

ローカルキーストアを構成するには、マスターパスフレーズを設定します。ローカルキーストアの場合、一覧表示または変更できるプロパティは、*master_passphrase* のみです。

master_passphrase が設定されているかどうかを確認します。

```
GET /api/storage/v2/encryption/local HTTP/1.1
```

出力:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "keystore": {
    "href": "/api/storage/v2/encryption/local",
    "master_passphrase": false,
    "keys": []
  }
}
```

master_passphrase の値を指定します。

```
PUT /api/storage/v2/encryption/local HTTP/1.1
```

```
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json
```

```
{
  "master_passphrase": "passphrase"
}
```

master_passphrase が設定されていることを確認します。

```
GET /api/storage/v2/encryption/local HTTP/1.1
```

出力:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "keystore": {
    "href": "/api/storage/v2/encryption/local",
    "master_passphrase": true,
    "keys": []
  }
}
```

OKM キーストアの構成

次の表では、OKM キーストアを構成するために設定する必要があるプロパティについて説明します。

表 68 OKM キーストアのプロパティ

プロパティ	型	説明
agent_id	string	エージェント ID。
registration_pin	string	この値は、OKM セキュリティー担当者から支給されます。
server_addr	string	OKM サーバーの IP アドレス。

前述のプロパティが設定されているかどうかを確認します。

```
GET /api/storage/v2/encryption/okm HTTP/1.1
```

出力:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "keystore": {
    "href": "/api/storage/v2/encryption/okm",
    "agent_id": "",
    "registration_pin": false,
    "server_addr": "",
    "keys": []
  }
}
```

```
}
}
```

agent_id、registration_pin、および server_addr の値を指定します。

```
PUT /api/storage/v2/encryption/okm HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json
```

```
{
  "agent_id": "agent-id",
  "registration_pin": "reg-pin",
  "server_addr": "ipaddr"
}
```

KMIP キーストアの構成

KMIP キーストアは、Oracle Key Vault などの KMIP 準拠のサーバーとともに使用されます。Oracle Key Vault は、専用サーバーにインストールされ、OASIS KMIP 標準をサポートするソフトウェアアプライアンスです。

KMIP を使用した暗号化を構成するには、[45 ページの「鍵または証明書のアップロード」](#)で説明されているように、KMIP 管理者から受信した鍵および証明書をアップロードします。

鍵および証明書をアップロードしたら、KMIP サーバー、クライアント証明書および鍵名を指定します。

次の表では、KMIP キーストアを構成するために設定するプロパティについて説明します。

表 69 KMIP キーストアのプロパティ

プロパティ	型	説明
server_list	list	KMIP サーバーの IP アドレスまたはホスト名。このプロパティには複数の値を設定できます。
client_cert	string	KMIP サーバー管理者によって提供されたファイルから作成した証明書。
host_match	boolean	サーバーのホスト名をサーバー証明書内のサーバーのアイデンティティに対して検証します。
destroy_key_on_remove	boolean	KMIP サーバー上の鍵が、アプライアンス上で削除された鍵である場合に、それを破棄または保持します。

host_match および destroy_key_on_remove の詳細は、"Key Management Interoperability Protocol (KMIP) Keystore" in [Oracle ZFS Storage Appliance 管理ガイド、Release OS8.8.x](#) を参照してください。

前述のプロパティーが設定されているかどうかを確認します。

```
GET /api/storage/v2/encryption/kmip HTTP/1.1

出力:

HTTP/1.1 200 OK
Content-Type: application/json

{
  "keystore": {
    "href": "/api/storage/v2/encryption/kmip",
    "server_list": [
      "ipaddr-or-hostname"
    ],
    "client_cert": "134a9138-29a0-4720-80bb-ec2b13457c39",
    "host_match": false,
    "destroy_key_on_remove": true,
    "keys": [],
    ... detailed information about the private key, certificate, and certificate authority ...
  }
}
```

暗号化鍵の作成

キーストアの構成後、鍵を作成するには、鍵名を設定するだけです。次の例では、新しい KMIP 鍵を作成します。結果例については、[233 ページの「暗号化鍵の一覧表示」](#)を参照してください。

```
リクエストの例:

POST /api/storage/v2/encryption/kmip/keys HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Accept: application/json
{
  "keyname": "atz-1-27-2021"
}
```

暗号化鍵の一覧表示

このコマンドは、すべての暗号化鍵のプロパティーを一覧表示します。成功したコマンドに対し、HTTP ステータス 200 (OK) が返されます。HTTP 本文には、JSON 形式の鍵の配列が含まれています。次の表では、暗号化鍵のプロパティーについて説明します。

表 70 暗号化鍵のプロパティー

プロパティー	型	説明
cipher	string	AES 暗号化タイプ。

プロパティ	型	説明
key	string	(ローカル専用) 暗号化された状態で保存された 16 進エンコード raw 256 ビット鍵。値を指定しない場合、この値は自動的に生成されます。
keyname	string	特定の鍵。
href	string	鍵へのパス。

次の例では、すべてのローカル鍵、OKM 鍵および KMIP 鍵を一覧表示します。

リクエストの例:

```
GET /api/storage/v2/encryption/local/keys /api/storage/v2/encryption/okm/keys /api/storage/v2/encryption/kmip/keys HTTP/1.1
```

結果例:

この結果は、このアプライアンスにローカル鍵または OKM 鍵が存在せず、KMIP 鍵が 1 つ存在することを示しています。

```
{
  "keys": {
  }
}
{
  "keys": {
  }
}
{
  "keys": [{
    "cipher": "AES",
    "keyname": "atz-1-27-2021",
    "href": "/api/storage/v2/encryption/kmip/keys/key-000"
  }]
}
```

指定した鍵で暗号化されたストレージの一覧表示

dependents 問合せは、指定した鍵で暗号化されたシェア、プロジェクトまたはプールを一覧表示します。

次の例は、デフォルトプール内のファイルシステム fs-enc のみが atz-1-27-2021 KMIP 鍵で暗号化されていることを示しています。

リクエストの例:

```
GET /api/storage/v2/encryption/kmip/keys/atz-1-27-2021/dependents HTTP/1.1
```

結果例:

```
{
  "dependents": [
    "pool-0/local/default/fs-enc"
  ]
}
```

```
    }
}
```

鍵の削除

鍵を削除するには、(鍵名ではなく) 鍵の href プロパティの値を使用します。削除に成功すると、HTTP ステータス 204 (No Content) が返されます。

鍵を削除すると、その鍵を使用しているすべてのプールおよびシェアのすべてのデータにアクセスできなくなります。これは、セキュアなデータ破棄と同等であり、鍵をバックアップして鍵復元に備えている場合を除き、永続的で取り消し不能です。

リクエストの例:

```
DELETE /api/storage/v2/encryption/kmip/keys/key-000 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

暗号化されたプール、プロジェクト、またはシェアの作成

暗号化されたプール、プロジェクト、またはシェアを作成するには、[161 ページの「プールの構成」](#)、[170 ページの「プロジェクトの作成」](#)、[177 ページの「ファイルシステムの作成」](#)、および[183 ページの「新しい LUN の作成」](#)で説明されている必須のプロパティに加えて、`encryption`、`keystore`、および `keyname` の各プロパティに値を指定します。

次の表では、特定のプール、プロジェクト、またはシェアの暗号化プロパティについて説明します。

表 71 プール、プロジェクト、またはシェアの暗号化プロパティ

プロパティ	型	説明
<code>encryption</code>	string	AES 暗号化のタイプおよび鍵の長さ。
<code>keystore</code>	string	キーストアのタイプ: <code>local</code> 、 <code>okm</code> または <code>kmip</code> 。
<code>keyname</code>	string	特定の鍵名。
<code>keylastchanged</code>	string	(読み取り専用) 鍵が最後に変更された日付。この値は、v2 では ISO-8601 日時形式であり、v1 では Javascript 日時形式です。値が空の場合、この鍵は作成されてから変更されていません。
<code>keystatus</code>	string	(読み取り専用) <code>available</code> 、 <code>unavailable</code> 、または <code>none</code> のいずれかです。このプロパティの値が <code>unavailable</code> の場合、鍵は削除されています。

リクエストの例:

```
POST /api/storage/v2/pools/p1/projects HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json
```

```
{
  "name": "proj-enc",
  "encryption": "aes-128-ccm",
  "keystore": "local",
  "keyname": "Key-0"
}
```

結果例:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://zfs-storage.example.com:215/pools/p1/projects/proj-enc
```

```
{
  "project": {
    "name": "proj-enc",
    "href": "/api/storage/v2/pools/p1/projects/proj-enc",
    ...
    "encryption": "aes-128-ccm",
    "keystore": "local",
    "keychangedate": "",
    "keystatus": "available",
    "keyname": "Key-0",
    ...
  }
}
```

システムコマンド

システムコマンドは、システムアイデンティティ情報の取得および最上位システムの管理コマンドの実行に使用されます。次の表に、使用可能なシステムコマンドを示します。

アプライアンスシステムコマンド

次のシステムコマンドを使用できます。

表 72 アプライアンスシステムコマンド

リクエスト	パス <code>/api/system/v{1 2}</code> の後ろに追加	説明
GET	<code>/version</code>	アプライアンスハードウェアおよびソフトウェアのバージョン情報を一覧表示します
PUT	<code>/reboot</code>	アプライアンスをリブートします。キューに入っているプラットフォーム更新は、このリブート中に適用されます。
PUT	<code>/reboot?skip_update=true</code>	キューに入っているプラットフォーム更新を適用せずに、アプライアンスをリブートします。
PUT	<code>/reboot?diag=true</code>	診断リブート: アプライアンスをリブートし、プロセスで追加の診断情報を収集します。
PUT	<code>/poweroff</code>	アプライアンスをオフにします
PUT	<code>/restart</code>	管理インタフェースをリブートし、診断情報を収集します
PUT	<code>/factoryreset</code>	アプライアンス構成を出荷時の設定にリセットします
GET	<code>/disks</code>	すべてのシステムディスクを一覧表示します
GET	<code>/disks/disk</code>	指定されたシステムディスクプロパティを一覧表示します
GET	<code>/memory</code>	システムメモリのステータスレポート

バージョンの取得

このコマンドは、システムのアイデンティティ情報を含むシステム構造を返します。成功したコマンドに対し、HTTP ステータス 200 (OK) が返されます。

リクエストの例:

```
GET /api/system/v1/version HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "version": {
    "hw_csn": "1211FM2009",
    "updated": "20130528T16:21:17",
    "fw_vendor": "American Megatrends Inc.",
    "os_isa": "i386",
    "os_boot": "20130528T16:25:44",
    "hw_product": "Sun Netra X4270 M3",
    "http_version": "Apache/2.2.24 (Unix)",
    "hw_asn": "2f4aeeb3-b670-ee53-e0a7-d8e0ae410749",
    "ssl_version": "OpenSSL 1.0.0k 5 Feb 2013",
    "os_machine": "i86pc",
    "os_nodename": "admin1",
    "os_version": "nas/generic@2013.05.16,1-0",
    "ak_product": "SUNW,iwashig2",
    "fw_version": "21000208",
    "os_release": "5.11",
    "installed": "20130411T19:50:16",
    "sp_version": "3.1.2.0",
    "os_platform": "i86pc",
    "fw_release": "10/22/2012"
  }
}
```

システムの電源切断

このコマンドは、アプライアンスの正常な停止を実行します。アプライアンスがクラスタの一部でないかぎり、すべてのデータサービスは永久に使用不可になります。システムの電源を入れ直すには、電源スイッチへのサービスプロセッサアクセスまたは物理アクセスが必要になります。このコマンドは非同期に実行し、HTTP ステータス 202 (Accepted) を返します。アプライアンスは、実際のコマンドのステータスに従うようモニターする必要があります。

リクエストの例:

```
PUT /api/system/v1/poweroff HTTP/1.1
Host: zfs-storage.example.com:215
```

システムのリブート

このコマンドは、アプライアンスの正常な電源再投入を実行します。すべてのサービスは一時的に使用できなくなります。このコマンドは非同期に実行し、HTTP ステータス 202 (Accepted) を返します。アプライアンスは、実際のコマンドのステータスに従うようモニターする必要があります。

注記 - 保留中のプラットフォーム更新がアプライアンスで利用可能な場合は、このリブート中に適用されます。保留中のプラットフォーム更新を適用せずにリブートを実行するには、代わりに `/reboot?skip_update=true` コマンドを使用します。

リクエストの例:

```
PUT /api/system/v1/reboot HTTP/1.1
Host: zfs-storage.example.com:215
```

リクエストの例:

```
PUT /api/system/v1/reboot?skip_update=true HTTP/1.1
Host: zfs-storage.example.com:215
```

システム管理の再起動

このコマンドは、管理インタフェースを再起動し、診断情報を収集します。このコマンドは非同期に実行し、HTTP ステータス 202 (Accepted) を返します。アプライアンスは、実際のコマンドのステータスに従うようモニターする必要があります。

リクエストの例:

```
PUT /api/system/v1/restart HTTP/1.1
Host: zfs-storage.example.com:215
```

診断リブート

このコマンドは、アプライアンスをリブートし、プロセスで追加の診断情報を収集します。このコマンドは非同期に実行し、HTTP ステータス 202 (Accepted) を返します。アプライアンスは、実際のコマンドのステータスに従うようモニターする必要があります。

注記 - 保留中のプラットフォーム更新がアプライアンスで利用可能な場合、この診断リブート中には適用されません。

リクエストの例:

```
PUT /api/system/v1/reboot?diag=true HTTP/1.1
```

Host: zfs-storage.example.com :215

出荷時リセット

このコマンドは、アプライアンス構成を元の出荷時の設定に復元します。構成の変更はすべて失われ、アプライアンスには、最初にインストールされたときの設定を行う必要があります。このコマンドは非同期に実行し、HTTP ステータス 202 (Accepted) を返します。アプライアンスは、実際のコマンドのステータスに従うようモニターする必要があります。このコマンドではすべての構成データが失われる可能性があるため、コマンドを成功させるために問合せパラメータ `?confirm=true` に設定する必要があります。

リクエストの例:

```
PUT /api/system/v1/factoryreset?confirm=true HTTP/1.1
Host: zfs-storage.example.com:215
```

システムのサポートバンドル

次のサポートバンドルコマンドを使用できます。

表 73 サポートバンドルコマンド

リクエスト	パス <code>/api/system/v{1 2}</code> の後ろに追加	説明
GET	<code>/bundles</code>	すべてのサポートバンドルを一覧表示します
GET	<code>/bundles/bundle</code>	指定されたバンドルデータまたはプロパティを取得します
POST	<code>/bundles</code>	サポートバンドルを構築し、それを Oracle サポートにアップロードします。
PUT	<code>/bundles/bundle/retry</code>	指定されたバンドルのアップロードを再試行します
PUT	<code>/bundles/bundle/cancel</code>	指定されたバンドルのアップロードを取り消します
PUT	<code>/bundles/bundle/send</code>	指定されたバンドルをオプションの SR 番号を使って Oracle Support にアップロードします。
DELETE	<code>/bundles/bundle</code>	指定されたバンドルを破棄します

サポートバンドルの作成

サービスリクエストを解決するために、新しいサポートバンドルを作成します。サポートバンドルをオープンサービスリクエストに関連付け、それを Oracle Support に送信するには、サービスリクエスト (SR) 番号を指定する必要があります。SR 番号の

形式は、3-`nnnnnnnnnn` する必要があります。サポートバンドルが自動的に Oracle サポートにアップロードされるには、フォンホーム設定がアップロード権限のある有効な MOS 資格に登録される必要があります。

リクエストの例:

```
POST /api/system/v1/bundles HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 23
```

```
{"srn": "3-0123456789"}
```

結果例:

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-API: 1.0
```

サービスリクエスト番号 (SRN) が指定されなかった場合、システムは代わりにローカルバンドルを構築します。

リクエストの例:

```
POST /api/system/v1/bundles HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 23
```

結果例:

```
{
  "bundle": {
    "status": "",
    "uuid": "d4431d57-ba4f-4f37-fa1e-a09fcbf3e56b",
    "associated_bundle": [
      {
        "href": "/api/system/v1/bundles/4050963a-4082-663f-99c0-fee915f2839c"
      }
    ],
    "srn": null,
    "filename": "ak.d4431d57-ba4f-4f37-fa1e-a09fcbf3e56b.tar.gz",
    "href": "/api/system/v1/bundles/d4431d57-ba4f-4f37-fa1e-a09fcbf3e56b",
    "date": "Thu Mar 10 2016 19:38:58 GMT+0000 (UTC)",
    "type": "User initiated"
  }
}
```

サポートバンドルの一覧表示

このコマンドは、システムによって処理または収集されるすべてのサポートバンドルを一覧表示します。サポートバンドルが Oracle サポートにアップロードされたあと、サポートバンドルはシステムから削除されます。

リクエストの例:

```
GET /api/system/v1/bundles HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: */*
```

結果例:

```
{
  "bundles": [{
    "status": "building",
    "step_progress": 6.25,
    "srn": "3-0123456789",
    "filename": "/upload/issue/3-0123456789/3-0123456789_ak.ba8ebd55-2349-c31c-cde3-
acf3fb0c3389.tar.gz",
    "href": "/api/system/v1/bundles/ba8ebd55-2349-c31c-cde3-acf3fb0c3389",
    "date": "Wed Apr 30 2014 19:31:06 GMT+0000 (UTC)",
    "type": "User initiated",
    "uuid": "ba8ebd55-2349-c31c-cde3-acf3fb0c3389"
  }],
}
```

サポートバンドルの取得

単一のバンドルからプロパティを取得します。

リクエストの例:

```
GET /api/system/v1/bundles/9604155c-928b-cf97-c826-cda9fc17ac57 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: */*
```

結果例:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 165

{
  "bundle": {
    "status": "building",
    "step_progress": 62.5,
    "srn": "3-0123456789",
    "filename": "/upload/issue/3-0123456789/3-0123456789_ak.ba8ebd55-2349-c31c-cde3-
acf3fb0c3389.tar.gz",
    "href": "/api/system/v1/bundles/ba8ebd55-2349-c31c-cde3-acf3fb0c3389",
    "date": "Wed Apr 30 2014 19:31:06 GMT+0000 (UTC)",
    "type": "User initiated",
    "uuid": "ba8ebd55-2349-c31c-cde3-acf3fb0c3389"
  }
}
```

サポートバンドルの取り消し

このコマンドは、サポートバンドルの自動アップロードを取り消します。

リクエストの例:

```
PUT /api/system/v1/bundles/9aef7c38-073c-603f-f35c-be64e26e90e3/cancel HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
```

結果例:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
```

サポートバンドルのアップロードの再試行

このコマンドは、Oracle サポートにバンドルのアップロードを試行する、新しいバンドルのアップロードジョブを作成します。バンドルの取得コマンドを使用して、サポートバンドルのアップロードのステータスをモニターできます。

リクエストの例:

```
PUT /api/system/v1/bundles/9aef7c38-073c-603f-f35c-be64e26e90e3/retry HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
```

結果例:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
```

バンドルのアップロードを別のサービスリクエスト (SR) 番号を使って再試行するには、`send` コマンドを使用します。SR 番号が指定されなかった場合、システムは元の SR 番号を使ってアップロードを再試行します。

注記 - ローカルで生成されたバンドルで `send` を実行する場合、SR 番号を指定する必要があります。そうしないと、エラーがスローされます。

リクエストの例:

```
PUT /api/system/v1/bundles/9aef7c38-073c-603f-f35c-be64e26e90e3/send HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
```

```
{"srn": "3-0123456789"}
```

結果例:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
```

サポートバンドルのアップロード

Oracle サポートに自動的にアップロードされていないサポートバンドルは、手動でアップロードできます。

注記 - ローカルで生成されたバンドルで `send` を実行する場合、SR 番号を指定する必要があります。そうしないと、エラーがスローされます。

リクエストの例:

```
PUT /api/system/v1/bundles/9aef7c38-073c-603f-f35c-be64e26e90e3/send HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215

{"srn": "3-0123456789"}
```

結果例:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
```

サポートバンドルの削除

このコマンドは、アプライアンスからサポートバンドルを削除します。

リクエストの例:

```
DELETE /api/system/v1/bundles/9aef7c38-073c-603f-f35c-be64e26e90e3 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
```

結果例:

```
HTTP/1.1 204 No Content
X-Zfssa-Appliance-API: 1.0
```

システムの更新

これらのコマンドは、システムの更新イメージを管理します。

表 74 更新コマンド

リクエスト	パス <code>/api/system/v{1 2}</code> の後ろに追加	説明
GET	<code>/updates</code>	すべてのシステム更新を一覧表示します
GET	<code>/updates/update</code>	指定されたシステム更新プロパティを取得します

リクエスト	パス /api/system/v{1 2} の後ろに追加	説明
GET	/update/platform	プラットフォームファームウェア (コントローラのサービスプロセッサおよびシステムボードのファームウェアを指す) の更新ステータスを表示します
GET	/update/firmware	コンポーネントファームウェア (ディスクと SSD のファームウェア、およびディスクエンクロージャー IOM ファームウェアを指す) の更新ステータスを表示します
PUT	/updates/update	更新設定を変更します
PUT	/updates/update/upgrade	指定された更新イメージへアップグレードします
PUT	/updates/update/check	指定された更新イメージに対してアップグレード健全性検査を実行します
PUT	/updates/update/rollback	指定された更新イメージにロールバックします
PUT	/updates-apply	互換性のない遅延更新を適用します
DELETE	/updates/update	指定されたシステム更新を破棄します
POST	/updates	更新イメージをアプライアンスにロードします

表 75 システム更新プロパティ

プロパティ	型	説明
version	String	媒体バージョンを更新します
release_date	DateTime	リリース日を更新します
install_date	DateTime	最新のインストール日を更新します。インストールされていない場合はアプライアンスへのダウンロード日です
status	String	媒体ステータスを更新します (変更不可)
update_deferred	ChooseOne	遅延設定: onreboot または onrequest

遅延更新の通知:

The following updates enable features that are incompatible with earlier software versions. As these updates cannot be reverted once committed, and peer system resources are updated across a cluster, verifying first that the system upgrade is functioning properly before applying deferred updates is advised.

システム更新の一覧表示

システム更新取得リクエストの例:

```
GET /api/system/v1/updates HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Length: 541
Content-Type: application/json

{
  "updates": [
    {
      "release_date": "Tue Aug 13 2013 17:47:32 GMT+0000 (UTC)",
      "install_date": "Wed Aug 14 2013 12:33:08 GMT+0000 (UTC)",
      "href": "/api/system/v1/updates/nas@2013.08.13,1-0",
      "status": "previous",
      "version": "2013.08.13,1-0"
    },
    {
      "release_date": "Sat Aug 24 2013 17:54:23 GMT+0000 (UTC)",
      "install_date": "Sun Aug 25 2013 11:30:14 GMT+0000 (UTC)",
      "href": "/api/system/v1/updates/nas@2013.08.24,1-0",
      "status": "current",
      "version": "2013.08.24,1-0"
    },
    {
      "release_date": "Sun Aug 25 2013 12:56:57 GMT+0000 (UTC)",
      "install_date": "Mon Aug 26 2013 18:50:33 GMT+0000 (UTC)",
      "href": "/api/system/v1/updates/nas@2013.08.25,1-0",
      "status": "waiting",
      "version": "2013.08.25,1-0"
    }
  ]
}
```

システム更新の取得

単一の更新イメージのプロパティを取得します。

リクエストの例:

```
GET /api/system/v1/updates/nas@2013.08.25,1-0 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Length: 541
Content-Type: application/json

{
  "update": {
    "release_date": "Sat Aug 24 2013 17:54:23 GMT+0000 (UTC)",
    "install_date": "Sun Aug 25 2013 11:30:14 GMT+0000 (UTC)",
    "href": "/api/system/v1/updates/nas@2013.08.24,1-0",
    "status": "current",
    "version": "2013.08.24,1-0",
    "update_deferred": "on_request"
  }
}
```

```
}
```

プラットフォームファームウェアの更新ステータスの取得

保留中のプラットフォームファームウェア更新の更新ステータスを取得します。プラットフォームファームウェアとは、コントローラのサービスプロセッサおよびシステムボードのファームウェアを指す総称です。

リクエストの例:

```
GET /api/system/v1/update/platform HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Length: 541
Content-Type: application/json

{
  "platform": {
    "href": "/api/system/v1/update/platform",
    "power_down_needed": true,
    "update_needed": "true"
  }
}
```

コンポーネントファームウェアの更新ステータスの取得

保留中、失敗、および進行中のコンポーネントファームウェア更新の数を取得します。コンポーネントファームウェアとは、ディスクと SSD のファームウェア、およびディスクエンクロージャー IOM ファームウェアを指す総称です。

リクエストの例:

```
GET /api/system/v1/update/firmware HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Length: 541
Content-Type: application/json

{
```

```
    "firmware": {
      "href": "/api/system/v1/update/firmware",
      "upgrades_pending": 0,
      "upgrades_failed": 0,
      "upgrades_in_progress": 0
    }
  }
```

システム更新のアップロード

このコマンドは、新しいシステム更新イメージをアップロードします。

curl を使用したアップロードコマンドの例:

```
$ curl --user root:root-password -k --data-binary @nas@2013.08.24,1-0.pkg.gz \
  --header "Content-Type: application/octet-stream" \
  https://zfs-storage.example.com/api/system/v1/updates
```

イメージがアップロードされ解凍されると、更新イメージのプロパティが返されます。成功すると、HTTP ステータスが 201 (Created) に設定され、新しいイメージの相対的な場所がロケーションヘッダーで返されます。

結果例:

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-API: 1.0
Content-Length: 541
Content-Type: application/json
Location: /api/system/v1/updates/nas@2013.08.24,1-0

{
  "update": {
    "release_date": "Sat Aug 24 2013 17:54:23 GMT+0000 (UTC)",
    "install_date": "Sun Aug 25 2013 11:30:14 GMT+0000 (UTC)",
    "href": "/api/system/v1/updates/nas@2013.08.24,1-0",
    "status": "current",
    "version": "2013.08.24,1-0",
    "update_deferred": "on_request"
  }
}
```

アップグレード

このコマンドは、更新イメージをロードし、アプライアンスをリブートして指定された更新イメージにします。指定されたイメージのステータスは、「previous」と等しい必要があり、そうでない場合コマンドは失敗します。

リクエストの例:

```
PUT /api/system/v1/updates/nas@2013.08.25,1-0/upgrade
Host: zfs-storage.example.com:215
```

```
Authorization: Basic Tm8gcGVla2luZyE=  
Content-Length: 0
```

結果例:

```
HTTP/1.1 202 Accepted  
X-Zfssa-Appliance-API: 1.0
```

ロールバック

ロールバックは、アプライアンスをリブートして、以前の更新イメージにします。

リクエストの例:

```
PUT /api/system/v1/updates/nas@2013.08.24,1-0/rollback  
Host: zfs-storage.example.com:215  
Authorization: Basic Tm8gcGVla2luZyE=  
Content-Length: 0
```

結果例:

```
HTTP/1.1 202 Accepted  
X-Zfssa-Appliance-API: 1.0
```

更新イメージの削除

アプライアンスから未使用の更新イメージを削除します。

リクエストの例:

```
DELETE /api/system/v1/updates/nas@2013.08.13,1-0 HTTP/1.1  
Host: zfs-storage.example.com:215  
Authorization: Basic Tm8gcGVla2luZyE=
```

結果例:

```
HTTP/1.1 204 No Content  
X-Zfssa-Appliance-API: 1.0
```


RESTful API ユーザーサービス

RESTful API ユーザーサービスは、アプライアンスでユーザーを構成します。

ユーザーサービスのコマンド

次のユーザーサービスのコマンドを使用できます。

表 76 ユーザーサービスのコマンド

リクエスト	パス <code>/api/user/v{1 2}</code> の後ろに追加	説明
GET	<code>/api/user/v{1 2}</code> のみを使用	ユーザーサービスコマンドを一覧表示します
GET	<code>/users</code>	すべてのアプライアンスユーザーを一覧表示します
GET	<code>/users/user</code>	指定されたユーザーのプロパティを一覧表示します
POST	<code>/users</code>	新しいアプライアンスユーザーを作成します
PUT	<code>/users/user</code>	指定されたユーザーのプロパティを変更します
DELETE	<code>/users/user</code>	アプライアンスから指定されたユーザーを削除します
GET	<code>/users/user/preferences</code>	指定されたユーザーのプリファレンスのプロパティを一覧表示します
PUT	<code>/users/user/preferences</code>	指定されたユーザーのプリファレンスのプロパティを変更します
GET	<code>/users/user/exceptions</code>	指定されたユーザーのすべての承認例外を一覧表示します
GET	<code>/users/user/exceptions/auth</code>	指定されたユーザーの <code>auth</code> 承認を一覧表示します
POST	<code>/users/user/exceptions</code>	指定されたユーザーの新しい承認例外を作成します
PUT	<code>/users/user/exceptions/auth</code>	指定されたユーザーの指定された承認を変更します
DELETE	<code>/users/user/exceptions/auth</code>	指定されたユーザーの指定された承認を破棄します

リクエスト	パス <code>/api/user/v{1 2}</code> の後ろに追加	説明
GET	<code>/users/user/preferences/keys</code>	指定されたユーザーのすべての SSH 鍵を一覧表示します
GET	<code>/users/user/preferences/keys/key</code>	指定されたユーザーの指定された SSH 鍵のプロパティを一覧表示します
POST	<code>/users/user/preferences/keys</code>	指定されたユーザーの新しい SSH 鍵を作成します
PUT	<code>/users/user/preferences/keys/key</code>	指定されたユーザーの指定された SSH 鍵を変更します
DELETE	<code>/users/user/preferences/keys/key</code>	指定されたユーザーの指定された SSH 鍵を破棄します
GET	<code>/users/user/preferences/tokens</code>	指定されたユーザーのすべての REST ログイントークンを一覧表示します
GET	<code>/users/user/preferences/tokens?token=token</code>	トークン値別の REST ログイントークンを一覧表示します
GET	<code>/users/user/preferences/tokens/token-id</code>	トークン ID 別の REST ログイントークンを一覧表示します
POST	<code>/users/user/preferences/tokens</code>	指定されたユーザーの REST ログイントークンを作成します
DELETE	<code>/users/user/preferences/tokens?token=token</code>	トークン値別の REST ログイントークンを削除します
DELETE	<code>/users/user/preferences/tokens/token-id</code>	トークン ID 別の REST ログイントークンを削除します

ユーザーサービスのプロパティ

ユーザー名とパスワードに加えて、ユーザーサービスのプロパティは、ユーザーに付与される承認、ユーザーに課される制限、ユーザーのロケールなどの特性を定義します。

ユーザープロパティ

ユーザーは次のプロパティを設定できます。一部のプロパティは、特定のタイプのユーザーのみが使用できます。

表 77 ユーザープロパティ

プロパティ	型	説明
logname	string	一意のユーザー名。logname は、ユーザーが作成された後は変更できません。

プロパティー	型	説明
type	string	ユーザーのタイプ: local、directory、data、または nologin。type は、ユーザーが作成された後は変更できません。デフォルト: local
uid	number	ユーザー ID。自分でユーザー ID を指定することも、システムにユーザー ID を割り当てさせることもできます。ユーザー ID を指定する場合、ユーザー ID を 100 より小さくすること、2147483646 より大きくすること、60001、60002、または 65534 と等しくすることはできません。uid は、ユーザーが作成された後は変更できません。
fullname	string	ユーザーのフルネームまたは実際の名前。BUI では、ダッシュボードの上部にある「ログアウト」ボタンの左側にフルネームが表示され、ブラウザのタブに表示されることもあります。デフォルト: logname と同じ
initial_password	string	ローカルユーザーおよびデータユーザーのパスワード。
require_annotation	boolean	true の場合: ■ BUI – ユーザーは、最初の BUI ページを表示する前にコメントを入力する必要があります。 ■ CLI – ユーザーは、CLI プロンプトを表示する前にコメントを入力する必要があります。 ■ REST – 未承認としてリクエストは失敗します。 セッション注釈は監査ログに表示されます。
roles	list	ディレクトリユーザーまたはローカルユーザーに割り当てられたロール。
kiosk_mode	boolean	true の場合、このユーザーはキオスクユーザーになります。 ■ BUI – ユーザーは、kiosk_screen プロパティーの値である画面のみの表示に制限されます。 ■ CLI – ログインに失敗します。 ■ REST – 未承認としてリクエストは失敗します。
kiosk_screen	string	kiosk_mode が true の場合に、このユーザーに許可される BUI 画面。デフォルト: status/dashboard
exceptions	list	ディレクトリユーザーまたはローカルユーザーに割り当てられた追加の承認、またはロールで割り当てられた承認に対する制限。
preferences	list	ユーザー環境のプリファレンス (ロケール、BUI 開始ページ、タイムアウト、SSH 公開鍵、REST ログイントークンなど)。

これらのプロパティーの詳細は、次のドキュメントを参照してください。

- 『Oracle ZFS Storage Appliance 管理ガイド、Release OS8.8.x』の「ユーザーとロールについて」

- 『Oracle ZFS Storage Appliance 管理ガイド、Release OS8.8.x』の「ユーザープロパティ」

ユーザーのロールおよび例外

`roles` プロパティを使用して、既存のロールをユーザーに割り当てます。ロールを作成するには、[265 ページの「RESTful API ロールサービス」](#)を参照してください。

`exceptions` プロパティを使用して、ユーザーの承認を追加します。このメカニズムを使用して、承認を削除することもできます。たとえば、ユーザーに特定の承認を付与するロールが割り当てられている場合、そのロールに含まれている他の承認を保持したまま、例外リストで個々の承認を `false` に設定できます。

次のリクエストでは、ユーザー `user1` の Kerberos サービスを再起動するための承認を追加します。

POST /api/user/v1/users/user1/exceptions HTTP/1.1

```
{
  "scope": "svc",
  "service": "kerberos",
  "allow_restart": true
}
```

次の結果は、`user1` の Kerberos サービスで利用できるすべての承認を示しています。この例では、他の承認のデフォルト値は `false` のままです。

```
{
  "auth": {
    "href": "/api/user/v1/users/user1/exceptions/auth-001",
    "scope": "svc",
    "service": "kerberos",
    "allow_administer": false,
    "allow_configure": false,
    "allow_restart": true
  }
}
```

ユーザー設定のプロパティ

ログインしているユーザー、または `allow_changePreferences` 承認を持っている任意のユーザーのプリファレンスを設定できます。`allow_changePreferences` 承認を得るには、[254 ページの「ユーザーのロールおよび例外」](#)を参照してください。

表 78 ユーザー設定のプロパティ

プロパティ	型	説明
locale	string	地域。デフォルト: C

プロパティー	型	説明
login_screen	string	ページが URL で指定されていない場合、ログインの成功時に表示される BUI ページです。デフォルト: <code>status/dashboard</code>
session_timeout	integer	ユーザーが BUI から移動したあとにブラウザが自動的にセッションからログアウトするまでの分数。デフォルト: 15
cli_idle_timeout	integer	セッションが中断されるまでに CLI をアイドル状態にできる時間の長さ (秒)。デフォルト値の -1 の場合、CLI はアイドル時に自動的に閉じられません。
advanced_analytics	string	使用可能な高度な分析統計を作成します
keys	list	RSA/DSA 公開鍵
tokens	list	REST ログイントークン

これらのプロパティーの詳細は、次のセクションを参照してください。

- [255 ページの「CLI タイムアウト」](#)
- [256 ページの「SSH 鍵」](#)
- [256 ページの「REST ログイントークン」](#)

CLI タイムアウト

デフォルトでは、CLI をアイドル状態にできる時間の長さに制限はありません (`cli_idle_timeout` の値は -1 です)。CLI をアイドル状態にできる時間の長さに制限を設定するには、`cli_idle_timeout` 秒に正の整数値を設定します。タイムアウトの制限に到達すると、CLI は閉じられます。

次の例では、CLI タイムアウトを 1 時間に設定します。

```
PUT /api/user/v1/users/user1/preferences HTTP/1.1

{ "cli_idle_timeout": 3600 }

{
  "preferences": {
    "href": "/api/user/v1/users/user1/preferences",
    "locale": "C",
    "login_screen": "configuration/preferences",
    "session_timeout": 15,
    "cli_idle_timeout": 3600,
    "advanced_analytics": false,
    "keys": [],
    "tokens": []
  }
}
```

タイムアウトを無効にするには、次の例に示すように、`cli_idle_timeout` の値を -1 に設定するか、`unset` を使用します。

```
PUT /api/user/v1/users/user1/preferences HTTP/1.1
```

```
{ "<unset>": ["cli_idle_timeout"] }
```

unset を山カッコで囲むと、unset という名前のプロパティが存在するという問題を回避できます。

SSH 鍵

SSH 公開鍵は、パスワードを使わずに SSH 接続を許可するのに使用できます。

表 79 SSH 鍵プロパティ

プロパティ	型	説明
type	string	SSH 鍵のタイプです。RSA または DSA
key	string	SSH 鍵の内容
comment	string	SSH 鍵に関連付けられたコメント

REST ログイントークン

永続的または非永続的な REST ログイントークンを作成し、返されたトークン値またはトークン ID を介してトークンにアクセスできます。

表 80 REST ログイントークンのプロパティ

プロパティ	型	説明
name	string	トークン名。
token_username	string	このログイントークンを使用できるユーザーの名前。この値はリクエストパスに設定されます。
preserve	boolean	false の場合、デフォルトの expiration の値を使用します。true の場合、カスタムの expiration の値を設定します。デフォルト: false
expiration	number	トークンを作成する場合、この値はトークンの有効期限が切れるまでの秒数です。preserve が true に設定されている場合は、このプロパティを設定する必要があります。preserve が false に設定されている場合、expiration の値は 900 です。 トークンのプロパティを一覧表示する場合、この値はこのトークンの有効期限が切れる日時です。

REST ログイントークンの問合せパラメータ

token=token 問合せパラメータを GET コマンドとともに使用して、そのトークンのプロパティ値を表示します。token=token 問合せパラメータを DELETE コマンドと

ともに使用して、そのトークンを削除します。*token* は、create token 応答の X-Auth-Session の値です。

token は、トークンが作成されたときに表示され、再度表示されることはありません。必ず *token* をコピーして保管しておいてください。[例3「REST ログイントークンの作成」](#)を参照してください。

ユーザーの一覧表示

次の例では、root ユーザーはこのアプライアンスのローカルであり、user1 ユーザーは、LDAP、NIS、または AD ユーザーです。directory タイプのユーザーの場合、ユーザー ID、フルネーム、およびパスワードはディレクトリサービスから取得され、パスワードのプロパティは一覧表示されません。ユーザータイプの詳細は、『[Oracle ZFS Storage Appliance 管理ガイド、Release OS8.8.x](#)』の「[ユーザーとロールについて](#)」を参照してください。

リクエストの例:

```
GET /api/user/v1/users HTTP/1.1
```

結果例:

```
{
  "users": [
    {
      "logname": "root",
      "type": "local",
      "uid": 0,
      "fullname": "Super-User",
      "initial_password": true,
      "require_annotation": false,
      "href": "/api/user/v2/users/root"
    }, {
      "logname": "user1",
      "type": "directory",
      "uid": user1uid,
      "fullname": "Real Name",
      "require_annotation": false,
      "roles": ["basic"],
      "kiosk_mode": false,
      "kiosk_screen": "status/dashboard",
      "href": "/api/user/v2/users/user1"
    }
  ]
}
```

特定のユーザーの一覧表示

特定のユーザーに関する情報には、ユーザー設定および承認の例外が含まれます。この例では、user1 に auth-000 という名前の承認例外があります。これは user1 にア

ラートを構成して送信する機能を付与します。user1 の設定はすべてデフォルト値です。

リクエストの例:

```
GET /api/user/v1/users/user1 HTTP/1.1
```

結果例:

```
{
  "user": {
    "href": "/api/user/v2/users/user1",
    "logname": "user1",
    "type": "directory",
    "uid": user1uid,
    "fullname": "Real Name",
    "require_annotation": false,
    "roles": ["basic"],
    "kiosk_mode": false,
    "kiosk_screen": "status/dashboard",
    "exceptions": [
      {
        "scope": "alert",
        "allow_configure": true,
        "allow_post": true,
        "href": "/api/user/v2/users/user1/exceptions/auth-000"
      }
    ],
    "preferences": {
      "href": "/api/user/v2/users/user1/preferences",
      "locale": "C",
      "login_screen": "status/dashboard",
      "session_timeout": 15,
      "cli_idle_timeout": -1,
      "advanced_analytics": false,
      "keys": [],
      "tokens": []
    }
  }
}
```

ユーザーの作成

新しいユーザーを作成するには、少なくともユーザー名 (logname) を指定する必要があります。タイプを指定しない場合、新しいユーザーはタイプ `local` になります。次のユーザータイプのリストで説明されているように、他のプロパティは、ユーザーのタイプに応じて必要になります。

- **directory** – ユーザー名は、既存の NIS、LDAP、または AD ユーザーである必要があります。UID、パスワード、およびフルネームは、NIS、LDAP、または AD から取得されます。
- **local** および **data** – ユーザー名とパスワードを指定します。UID を指定することもできますが、UID は自動的に割り当てられます。
- **nologin** – ユーザーを指定します。UID を指定することもできますが、UID は自動的に割り当てられます。

ユーザーやユーザータイプについてより深く理解するには、『[Oracle ZFS Storage Appliance 管理ガイド、Release OS8.8.x](#)』の「ユーザーとロールについて」を参照してください。

例 1 ローカルユーザーの作成

リクエストの例:

POST /api/user/v1/users HTTP/1.1

```
{
  "logname": "test_user",
  "initial_password": "password"
}
```

結果例:

```
{
  "user": {
    "href": "/api/user/v1/users/test_user",
    "logname": "test_user",
    "type": "local",
    "uid": 2000000002,
    "fullname": "test_user",
    "initial_password": true,
    "require_annotation": false,
    "roles": ["basic"],
    "kiosk_mode": false,
    "kiosk_screen": "status/dashboard",
    "exceptions": [],
    "preferences": {
      "href": "/api/user/v1/users/test_user/preferences",
      "locale": "C",
      "login_screen": "status/dashboard",
      "session_timeout": 15,
      "cli_idle_timeout": -1,
      "advanced_analytics": false,
      "keys": [],
      "tokens": []
    }
  }
}
```

例 2 ユーザーのクローニング

既存のユーザーと同じタイプで、同じロールと承認が割り当てられた新しいユーザーを作成するには、次のプロパティを指定します。

- user – クローニングするユーザーのユーザー名。
- clonename – 新しくクローニングされたユーザーのユーザー名。
- password – 新しくクローニングされたユーザーの初期パスワード。

リクエストの例:

POST /api/user/v1/users HTTP/1.1

```
{
  "user": "test_user",
  "clonename": "clone_user",
  "password": "password"
}
```

結果例:

```
{
  "user": {
    "href": "/api/user/v1/users/clone_user",
    "logname": "clone_user",
    "type": "local",
    "uid": 2000000003,
    "fullname": "clone_user",
    "initial_password": true,
    "require_annotation": false,
    "roles": ["basic"],
    "kiosk_mode": false,
    "kiosk_screen": "status/dashboard",
    "exceptions": [],
    "preferences": {
      "href": "/api/user/v1/users/clone_user/preferences",
      "locale": "C",
      "login_screen": "status/dashboard",
      "session_timeout": 15,
      "cli_idle_timeout": -1,
      "advanced_analytics": false,
      "keys": [],
      "tokens": []
    }
  }
}
```

ユーザープロパティの変更

ユーザーのプロパティを直接変更します。logname、type、および uid は変更できません。

ロール、例外、設定、SSH 鍵、またはログイントークンを追加、変更、または削除するには、次のドキュメントを参照してください。

- [265 ページの「RESTful API ロールサービス」](#)
- [252 ページの「ユーザーサービスのプロパティ」](#)

リクエストの例:

```
PUT /api/user/v1/users/test_user HTTP/1.1
{"fullname": "Test A. User", "require_annotation": true}
```

結果例:

```
{
  "user": {
```

```

    "href": "/api/user/v1/users/test_user",
    "logname": "test_user",
    "type": "local",
    "uid": 2000000002,
    "fullname": "Test A. User",
    "initial_password": true,
    "require_annotation": true,
    "roles": ["basic"],
    "kiosk_mode": false,
    "kiosk_screen": "status/dashboard",
    "exceptions": [],
    "preferences": {
      "href": "/api/user/v2/users/test_user/preferences",
      "locale": "C",
      "login_screen": "status/dashboard",
      "session_timeout": 15,
      "cli_idle_timeout": -1,
      "advanced_analytics": false,
      "keys": [],
      "tokens": []
    }
  }
}

```

ユーザーの削除

システムからユーザーを削除します。

リクエストの例:

```
DELETE /api/user/v1/users/clone_user HTTP/1.1
```

結果例:

```
HTTP/1.1 204 No Content
```

トークンの管理

永続的および非永続的な REST ログイントークンの作成、トークンのプロパティの表示、およびトークンの削除を行うことができます。REST ログイントークンのプロパティは、トークンが作成された後は読み取り専用です。返されたトークン値またはトークン ID を介してトークンにアクセスできます。

例 3 REST ログイントークンの作成

REST ログイントークンを作成するには、ユーザー名とパスワードが必要です。

トークン name を設定します。デフォルトでは、`preserve` の値は `false` であり、`expiration` の値は 900 です。`preserve` を `true` に設定する場合は、`expiration` を秒数に設定する必要があります。

結果の **X-Auth-Session** の値を必ず保存してください。詳細については、[256 ページの「REST ログイントークン」](#) を参照してください。

リクエストの例:

```
POST /api/user/v2/users/test_user/preferences/tokens HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-User: test_user
X-Auth-Key: password-xxx
Content-Type: application/json
```

```
{"name": "Test Token"}
```

結果例:

```
HTTP/1.1 201 Created
...
X-Auth-Session: JjZJsZrVQfbZULyAuvSJjTftnBHCCQT

{
  "token": {
    "href":
"/api/user/v2/users/test_user/preferences/tokens/fb65a127-a04c-4f58-bc52-efa884447efb",
    "name": "Test Token",
    "token_username": "test_user",
    "preserve": false,
    "expiration": "2020-04-30T02:33:44Z",
    "id": "fb65a127-a04c-4f58-bc52-efa884447efb"
  }
}
```

リクエストの例:

```
POST /api/user/v2/users/test_user/preferences/tokens HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-User: test_user
X-Auth-Key: password-xxx
Content-Type: application/json
```

```
{"name": "Another Token", "preserve": true, "expiration": 3600}
```

結果例:

```
HTTP/1.1 201 Created
...
X-Auth-Session: pviHrthBGQhGZHoPuqxFQrDcCPZgwEK

{
  "token": {
    "href":
"/api/user/v2/users/test_user/preferences/tokens/21f981ad-6221-4fb4-a7d1-dd5560256dfb",
    "name": "Another Token",
    "token_username": "test_user",
    "preserve": true,
    "expiration": "2020-04-30T03:20:31Z",
    "id": "21f981ad-6221-4fb4-a7d1-dd5560256dfb"
  }
}
```

例 4 指定されたユーザーのすべてのトークンの一覧表示

リクエストの例:

```
GET /api/user/v2/users/test_user/preferences/tokens HTTP/1.1
X-Auth-Session: pviHrthBGQhGZHoPuqxFQrDccPZgwEK
```

結果例:

```
{
  "tokens": [{
    "name": "Another Token",
    "token_username": "test_user",
    "preserve": true,
    "expiration": "2020-04-30T03:20:31Z",
    "id": "21f981ad-6221-4fb4-a7d1-dd5560256dfb",
    "href": "/api/user/v2/users/testuser1/preferences/tokens/21f981ad-6221-4fb4-a7d1-dd5560256dfb"
  }, {
    "name": "Test Token",
    "token_username": "test_user",
    "preserve": false,
    "expiration": "2020-04-30T02:33:44Z",
    "id": "fb65a127-a04c-4f58-bc52-efa884447efb",
    "href": "/api/user/v2/users/testuser1/preferences/tokens/fb65a127-a04c-4f58-bc52-efa884447efb"
  }]
}
```

例 5 トークン値別の特定のトークンの一覧表示

リクエストの例:

```
GET /api/user/v2/users/test_user/preferences/tokens?token=pviHrthBGQhGZHoPuqxFQrDccPZgwEK
HTTP/1.1
X-Auth-Session: pviHrthBGQhGZHoPuqxFQrDccPZgwEK
```

結果例:

```
{
  "token": {
    "href":
"/api/user/v2/users/test_user/preferences/tokens/21f981ad-6221-4fb4-a7d1-dd5560256dfb",
    "name": "Another Token",
    "token_username": "test_user",
    "preserve": true,
    "expiration": "2020-04-30T03:20:31Z",
    "id": "21f981ad-6221-4fb4-a7d1-dd5560256dfb"
  }
}
```

例 6 トークン ID 別の特定のトークンの一覧表示

```
GET /api/user/v2/users/test_user/preferences/tokens/21f981ad-6221-4fb4-a7d1-dd5560256dfb
HTTP/1.1
X-Auth-Session: pviHrthBGQhGZHoPuqxFQrDccPZgwEK
```

結果は前述の場合と同じです。

例 7 トークン値別のトークンの削除

リクエストの例:

```
DELETE /api/user/v2/users/test_user/preferences/tokens?  
token=pviHrthBGQhGZHoPuqxFQrDcCPZgwEK HTTP/1.1  
X-Auth-Session: pviHrthBGQhGZHoPuqxFQrDcCPZgwEK
```

結果例:

```
HTTP/1.1 204 No Content
```

例 8 トークン ID 別のトークンの削除

```
DELETE /api/user/v2/users/test_user/preferences/tokens/21f981ad-6221-4fb4-a7d1-dd5560256dfb  
HTTP/1.1  
X-Auth-Session: pviHrthBGQhGZHoPuqxFQrDcCPZgwEK
```

RESTful API ロールサービス

ロールとは、ユーザーに割り当てることができる承認の名前付きコレクションです。さまざまな目的のために、さまざまな承認を持つロールを作成できます。アプリケーションユーザーには、必要なロールを割り当てることができます。ロールを使用することは、管理者パスワードを共有するよりも安全です。ロールは、ユーザーに必要な承認だけに制限し、監査ログでユーザーのアクションをそのユーザー名に結び付けます。

「基本管理」ロールは既存のものであり、デフォルトですべてのユーザーに割り当てられています。

ロールサービスコマンド

次のリストにロールサービスコマンドを示します。

表 81 ロールサービスコマンド

リクエスト	パス <code>/role/v{1 2}</code> の後ろに追加	説明
GET	<code>/role/v{1 2}</code> のみを使用	ロールサービスコマンドを一覧表示します
GET	<code>/roles</code>	すべてのロールを一覧表示します
GET	<code>/roles/role</code>	指定されたロールのプロパティを一覧表示します
POST	<code>/roles</code>	新しいロールを作成します
PUT	<code>/roles/role</code>	指定されたロールのプロパティを変更します
PUT	<code>/roles/role/revoke</code>	すべてのユーザーから指定されたロールを削除します
DELETE	<code>/roles/role</code>	指定されたロールを破棄します
GET	<code>/roles/role/authorizations</code>	指定されたロールのすべての承認を一覧表示します
GET	<code>/roles/role/authorizations/auth</code>	指定されたロールの承認のプロパティを一覧表示します
POST	<code>/roles/role/authorizations</code>	指定されたロールの新しい承認を作成します
PUT	<code>/roles/role/authorizations/auth</code>	指定されたロールの承認のプロパティを変更します

リクエスト	パス <code>/role/v{1 2}</code> の後ろに追加	説明
DELETE	<code>/roles/role/authorizations/auth</code>	指定されたロールの承認を破棄します

ロールサービスのプロパティ

name プロパティと **description** プロパティは、ロールを作成するために必要であり、すべてのロールを一覧表示すると表示されます。承認は、ロールの作成後に追加され、特定のロールを一覧表示すると表示されます。

表 82 ロールのプロパティ

プロパティ	型	説明
name	string	一覧に表示されるロールの名前
description	string	ロールの詳細な説明。
authorizations	list	このロールへの承認

ロールの一覧表示

すべての定義済みのロールを一覧表示します。

リクエストの例:

```
GET /api/role/v1/roles HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
{
  "roles": [{
    "description": "Basic administration",
    "href": "/api/role/v1/roles/basic",
    "name": "basic",
    "role": "basic"
  }, {
    "description": "a",
    "href": "/api/role/v1/roles/rola",
    "name": "rola",
    "role": "rola"
  }]
}
```

ロールの取得

単一のロールのプロパティを取得します。プロパティメタデータを返すには、`props` 問合せパラメータを `true` に設定します。

リクエストの例:

```
GET /api/role/v1/roles/basic?props=true HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 390

{
  "props": [{
    "immutable": true,
    "label": "Role name",
    "name": "name",
    "type": "String"
  }, {
    "label": "A description of this role",
    "name": "description",
    "type": "String"
  }],
  "role": {
    "authorizations": [],
    "description": "Basic administration",
    "href": "/api/role/v1/roles/basic",
    "name": "basic"
  }
}
```

ロールの作成

このコマンドは新しいロールを作成します。

表 83 新しいロールの作成のプロパティ

プロパティ	型	説明
name	string	新しいロールの名前 (必須)
clone	string	元のプロパティのクローンを作成するロールの名前 (オプション)
description	string	ロールの説明 (必須)

リクエストの例:

```
POST /api/role/v1/roles HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 71

{"name":"role_workflow", "description":"Role to run workflows"}
```

結果例:

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 143
Location: /api/role/v1/roles/role_workflow

{
  "role": {
    "authorizations": [],
    "description": "Role to run workflows",
    "href": "/api/role/v1/roles/role_workflow",
    "name": "role_workflow"
  }
}
```

ロールの変更

ロールのプロパティはロールの作成後に変更できます。

リクエストの例:

```
PUT /api/role/v1/roles/role_workflow HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 54

{"description":"Role allowing user to run workflows!"}
```

結果例:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 158

{
  "role": {
    "authorizations": [],
    "description": "Role allowing user to run workflows!",
    "href": "/api/role/v1/roles/role_workflow",
    "name": "role_workflow"
  }
}
```

ロールの取り消し

すべてのユーザーからロールを取り消します。

リクエストの例:

```
PUT /api/role/v1/role_worksheets/revoke HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 0
```

ロールの削除

システムからロールを削除します。ロールがまだ1人以上のユーザーに割り当てられている場合は、`?confirm=true` を DELETE コマンドに追加します。

リクエストの例:

```
DELETE /api/role/v1/roles/rola?confirm=true HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: */*
```

結果例:

```
HTTP/1.1 204 No Content
X-Zfssa-Appliance-API: 1.0
```

ロールの承認の一覧表示

選択されたロールの承認を一覧表示します。

リクエストの例:

```
GET /api/role/v1/roles/role_workflow/authorizations HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
{
```

```

    "authorizations": [{
      "allow_modify": false,
      "allow_read": true,
      "auth": "auth-000",
      "href": "/api/role/v1/roles/role_workflow/authorizations/auth-000",
      "owner": "*",
      "scope": "workflow",
      "uuid": "*"
    }]
  }
}

```

ロールの承認の作成

新しいロールの承認を作成します。入力プロパティは CLI で定義されるものと同じです。各承認には定義済みの **scope** プロパティがあります。その他のプロパティは入力スコープに基づいて設定できます。スコープの値は次のとおりです。

ad	cluster	keystore	role	stmf	user
alert	dataset	nas	schema	svc	workflow
appliance	hardware	net	stat	update	worksheet

リクエストの例:

```

POST /api/role/v1/roles/role_workflow/authorizations HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 41
{"scope": "workflow", "allow_read": true}

```

結果例:

```

HTTP/1.1 201 Created
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 171
Location: /api/role/v1/roles/role_workflow/authorizations/auth-000

{
  "auth": {
    "allow_modify": false,
    "allow_read": true,
    "href": "/api/role/v1/roles/role_workflow/authorizations/auth-000",
    "owner": "*",
    "scope": "workflow",
    "uuid": "*"
  }
}

```

ロールの承認の変更

ロールの承認プロパティを変更できます。

リクエストの例:

```
PUT /api/role/v1/roles/role_workflow/authorizations/auth-000 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 29
```

```
{"allow_modify": true}
```

結果例:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 171
```

```
{
  "auth": {
    "allow_modify": true,
    "allow_read": true,
    "href": "/api/role/v1/roles/role_workflow/authorizations/auth-000",
    "owner": "**",
    "scope": "workflow",
    "uuid": "**"
  }
}
```

ロールの承認の削除

ロールの承認を削除します。

リクエストの例:

```
DELETE /api/role/v1/roles/role_workflow/authorizations/auth-000 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: */*
```

結果例:

```
HTTP/1.1 204 No Content
X-Zfssa-Appliance-API: 1.0
```


ワークフローおよびスクリプトコマンド

このサービスは、ワークフローの管理に使用します。ワークフローとは、アプライアンスにアップロードされ、そのアプライアンスで管理されるスクリプトのことです。ワークフローは、ブラウザインタフェースまたはコマンド行インタフェースのどちらからも優れた方法でパラメータ化し、実行できます。ワークフローはまた、警告アクションとして実行したり、指定された時間に実行したりすることもできます。そのため、ワークフローを使用すると、特定のポリシーや手順を取り込むような方法でアプライアンスを拡張できます。また、特定の組織またはアプリケーションのベストプラクティスを正式にエンコードすることもできます。

ワークフローおよびスクリプトサービスコマンド

次の表に、ワークフローサービスコマンドを示します。

表 84 ワークフローサービスコマンド

リクエスト	パス <code>/api/workflow/v{1 2}</code> の後ろに追加	説明
GET	<code>/api/workflow/v{1 2}</code> のみを使用	ワークフローサービスコマンドを一覧表示します
POST	<code>/workflows</code>	新しいワークフローをアプライアンスにアップロードします
GET	<code>/workflows</code>	すべてのワークフローを一覧表示します
GET	<code>/workflows/workflow</code>	指定されたワークフローのプロパティを一覧表示します
PUT	<code>/workflows/workflow</code>	指定されたワークフローのプロパティを変更します
PUT	<code>/workflows/workflow/execute</code>	指定されたワークフローを実行します
DELETE	<code>/workflows/workflow</code>	指定されたワークフローを破棄します
POST	<code>/scripts</code>	スクリプトをアップロードして実行します
GET	<code>/scripts</code>	実行中のすべてのスクリプトを一覧表示します
GET	<code>/scripts/script</code>	実行中のスクリプトに再接続します
DELETE	<code>/scripts/script</code>	実行中のスクリプトを停止します

ワークフローのアップロード

アプライアンスにワークフローをアップロードします。

リクエストの例:

```
POST /api/workflow/v1/workflows HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/javascript
Content-Length: 290

var workflow = {
  name: 'Echo',
  description: 'Echo bird repeats a song.',
  parameters: {
    song: {
      label: 'Words of a song to sing',
      type: 'String',
    }
  },
  execute: function (params) { return (params.song) }
};
```

結果例:

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 268
X-Zfssa-Version: user/generic@2013.09.14,1-0
Location: /api/workflow/v1/workflows/f4fe892f-cf46-4d6a-9026-cd0c0cce9971

{
  "workflow": {
    "href": "/api/workflow/v1/workflows/f4fe892f-cf46-4d6a-9026-cd0c0cce9971",
    "name": "Echo",
    "description": "Echo bird repeats a song.",
    "uuid": "f4fe892f-cf46-4d6a-9026-cd0c0cce9971",
    "owner": "root",
    "origin": "<local>",
    "setid": false,
    "alert": false,
    "version": "",
    "scheduled": false
  }
}
```

ワークフローの一覧表示

アプライアンスにインストールされたすべてのワークフローを一覧表示します。問合せパラメータ `showhidden=true` が設定されている場合、一覧には通常は隠されているワークフローが含まれます。

リクエストの例:

```
GET /api/workflow/v1/workflows HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json; charset=utf-8
Content-Length: 1908
```

```
{
  "workflows": [{
    "description": "Clear locks held on behalf of an NFS client",
    "href": "/api/workflow/v1/workflows/10f25f2c-3a56-e733-d9c7-d4c6fd84e073",
    ...
  },
  {
    "description": "Sets up environment for Oracle Solaris Cluster NFS",
    "href": "/api/workflow/v1/workflows/2793f2dc-72de-eac4-c58b-cf5e527df92d",
    ...
  },
  {
    "description": "Removes the artifacts from the appliance used by Oracle Solaris
Cluster NFS",
    "href": "/api/workflow/v1/workflows/9e2d5eed-cc72-67b0-e913-bf5ffad1d9e1",
    ...
  },
  {
    "description": "Sets up environment to be monitored by Oracle Enterprise Manager",
    "href": "/api/workflow/v1/workflows/bb5de1b8-b950-6da6-a650-f6fb19f1172c",
    ...
  },
  {
    "description": "Removes the artifacts from the appliance used by Oracle Enterprise
Manager",
    "href": "/api/workflow/v1/workflows/bd7214fc-6bba-c7ad-ed1f-942c0189e757",
    ...
  }
  ]
}
```

ワークフローの取得

単一のワークフローのプロパティを取得します。ヘッダー内で **Accept** に **application/javascript** を指定した場合はワークフローの内容が返され、それ以外の場合はワークフローのプロパティが返されます。

Accept に **application/javascript** を指定した場合のリクエストの例:

```
GET /api/workflow/v1/workflows/cc574599-4763-4523-9e72-b74e1246d448 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/javascript
```

結果例:

```
HTTP/1.1 200 OK
```

```
X-Zfssa-Appliance-API: 1.0
Content-Type: application/javascript; charset=utf-8
Content-Length: 916

var workflow = {
  name: 'Clear locks',
  description: 'Clear locks held on behalf of an NFS client',
  origin: 'Oracle Corporation',
  version: '1.0.0',
  parameters: {
    hostname: {
      label: 'Client hostname',
      type: 'String'
    },
    ipaddrs: {
      label: 'Client IP address',
      type: 'String'
    }
  },
  validate: function (params) {
    if (params.hostname == '') {
      return ({ hostname: 'Hostname cannot be empty.' });
    }

    if (params.ipaddrs == '') {
      return ({ ipaddrs: 'IP address cannot be empty.' });
    }
  },
  execute: function (params) {
    try {
      nas.clearLocks(params.hostname, params.ipaddrs);
    } catch (err) {
      return ('Failed to clear NFS locks: ' + err.message);
    }

    return ('Clear of locks held for ' + params.hostname +
      ' returned success. ');
  }
};
```

Accept に application/json を指定した場合のリクエストの例:

```
GET /api/workflow/v1/workflows/cc574599-4763-4523-9e72-b74e1246d448 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

結果例:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json; charset=utf-8
Content-Length: 649

{
  "workflow": {
    "href": "/api/workflow/v1/workflows/cc574599-4763-4523-9e72-b74e1246d448",
    "name": "Clear locks",
    "description": "Clear locks held on behalf of an NFS client",
    "uuid": "cc574599-4763-4523-9e72-b74e1246d448",
    "checksum": "695d029224f614258e626fe0b3c449c1233dee119571f23b678f245f7748d13c",
    "installdate": "Wed Apr 01 2015 17:59:44 GMT+0000 (UTC)",
    "owner": "root",
    "origin": "Oracle Corporation",
```

```

        "setid": false,
        "alert": false,
        "version": "1.0.0",
        "scheduled": false
    }
}

```

ワークフローの変更

ワークフローリソースに PUT リクエストを送信して、単一のワークフローのプロパティを変更できます。

リクエストの例:

```

PUT /api/workflow/v1/workflows/6c2b6545-fa78-cc7b-8cc1-ff88bd628e7d HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 28

{"setid": false}

```

結果例:

```

HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 234

{
  "workflow": {
    "alert": false,
    "description": "Echo bird repeats a song.",
    "href": "/api/workflow/v1/workflows/448b78e1-f219-e8f4-abb5-e01e09e1fac8",
    "name": "Echo",
    "origin": "<local>",
    "owner": "root",
    "scheduled": false,
    "setid": true,
    "uuid": "448b78e1-f219-e8f4-abb5-e01e09e1fac8",
    "version": ""
  }
}

```

ワークフローの実行

ワークフロースクリプトを実行して結果を返します。ワークフローパラメータは、本文内の JSON オブジェクトで渡される必要があります。成功すると、HTTP ステータス 202 (Accepted) が、ワークフローの出力を含む単一の結果プロパティを持つ JSON オブジェクトとともに返されます。

リクエストの例:

```
PUT /api/workflow/v1/workflows/6c2b6545-fa78-cc7b-8cc1-ff88bd628e7d/run HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 28

{"song": "tweet tweet tweet"}
```

結果例:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 34

{
  "result": "tweet tweet tweet\n"
}
```

ワークフローの削除

アプライアンスからワークフロースクリプトを削除します。

リクエストの例:

```
DELETE /api/workflow/v1/workflows/f4fe892f-cf46-4d6a-9026-cd0c0cce9971 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: */*
```

結果例:

```
HTTP/1.1 204 No Content
X-Zfssa-Appliance-API: 1.0
```

スクリプトのアップロードおよび実行

アプライアンスにスクリプトをアップロードして実行します。

root ユーザーは、アプライアンスにアップロードされているすべてのスクリプトを表示してアクセスできます。root 以外のユーザーは、自分のスクリプトのみを表示してアクセスできます。

スクリプトの詳細については、『[Oracle ZFS Storage Appliance 管理ガイド、Release OS8.8.x](#)』の「[CLI スクリプトの操作](#)」を参照してください。

このスクリプトは、アプライアンス上のすべてのシェアを一覧表示します。

リクエストの例:

```
$ curl -kv --user root:pw --data-binary @listShares.aksh \
https://hostname:215/api/workflow/v1/scripts
```

```
POST /api/workflow/v1/scripts HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
User-Agent: curl/7.45.0
Accept: */*
Content-Length: 12
Content-Type: application/x-www-form-urlencoded
```

結果例:

```
HTTP/1.1 201 Created
Date: Mon, 27 Mar 2017 22:16:38 GMT
X-Zfssa-Workflow-API: 1.1
X-Zfssa-Version: user/generic@2017.02.27,1-0
X-Zfssa-API-Version: 1.0
Content-Type: plain/text; charset=utf-8
Transfer-Encoding: chunked
```

```
default
share1
share2
fs1
lun1
```

実行中のすべてのスクリプトの一覧表示

実行中のすべてのスクリプトを一覧表示するには、次のコマンドを使用します。

root ユーザーは、アプライアンスにアップロードされているすべてのスクリプトを表示してアクセスできます。root 以外のユーザーは、自分のスクリプトのみを表示してアクセスできます。

スクリプトの詳細については、『[Oracle ZFS Storage Appliance 管理ガイド、Release OS8.8.x](#)』の「[CLI スクリプトの操作](#)」を参照してください。

リクエストの例:

```
$ curl -kv --user root:pw https://hostname:215/api/workflow/v1/scripts
```

```
GET /api/workflow/v1/scripts HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
User-Agent: curl/7.45.0
Accept: */*
```

結果例:

```
HTTP/1.1 200 OK
Date: Mon, 27 Mar 2017 22:41:06 GMT
Content-Length: 96
X-Zfssa-Workflow-API: 1.1
```

```
X-Zfssa-API-Version: 1.0
Content-Type: application/json; charset=utf-8

{
  "scripts": [
    {
      "time": 4,
      "href": "/api/workflow/v1/scripts/1",
      "user": "root",
      "script": "1"
    },
    {
      "time": 39,
      "href": "/api/workflow/v1/scripts/9",
      "user": "root",
      "script": "9"
    }
  ]
}
```

実行中のスクリプトへの再接続

root ユーザーは、アプライアンスにアップロードされている実行中のどのスクリプトにも再接続できます。root 以外のユーザーは、実行中の自分のスクリプトのみに再接続できます。

スクリプトの詳細については、『[Oracle ZFS Storage Appliance 管理ガイド、Release OS8.8.x](#)』の「[CLI スクリプトの操作](#)」を参照してください。

リクエストの例:

```
$ curl -kv -H "Accept: text/plain" --user root:pw \
https://hostname:215/api/workflow/v1/scripts/9

GET /api/workflow/v1/scripts/9 HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
User-Agent: curl/7.45.0
Accept: text/plain
```

結果例:

```
{
  "test2": {,
    "str": "this is a string",
    "bool": "True",
    "posint": 994,
    "int": 1123,
    "address": "",
    "host": "192.0.2.0",
    "hostname": "example.com",
    "color": "red",
    "languages": "latin",
    "size": "red",
    "onoff": "off",
    "number": 0,
    "stringlist": "this is another string",
```

```

    "emptystringlist": "this is another string",
    "yetanotherstr": "You can't change me",
    "emptystr": "Any string",
    "password": "password",
    "longpassword": "longpassword",
    "permissions": "022",
    "nonnegativeint": 42,
    "port": 21,
    "time": "Thu Jan 01 1970 15:22:30 GMT+0000 (UTC)",
    "date": "Sun Jun 17 2007 00:00:00 GMT+0000 (UTC)",
    "datetime": "Sun Jun 17 2007 15:22:00 GMT+0000 (UTC)",
    "hostport": "ipaddr-1",
    "dn": "uid=root,ou=people,dc=fishpong,dc=com",
    "commalist": "foo,bar"
  }
},
  "utask": []
}

```

実行中のスクリプトの停止

root ユーザーは、アプライアンスにアップロードされている実行中のどのスクリプトでも削除できます。root 以外のユーザーは、実行中の自分のスクリプトのみにアクセスして削除できます。

スクリプトの詳細については、『[Oracle ZFS Storage Appliance 管理ガイド、Release OS8.8.x](#)』の「[CLI スクリプトの操作](#)」を参照してください。

リクエストの例:

```
$ curl -kv -X DELETE --user root:l1a \
https://hostname:215/api/workflow/v1/scripts/9
```

```
DELETE /api/workflow/v1/scripts/9 HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
User-Agent: curl/7.45.0
Accept: */*
```

結果例:

```
HTTP/1.1 204 No Content
Date: Mon, 27 Mar 2017 22:59:12 GMT
Content-Length: 0
X-Zfssa-Workflow-API: 1.1
X-Zfssa-Version: build/generic@2017.02.27,1-0
X-Zfssa-API-Version: 1.0
Content-Type: application/json; charset=utf-8
```


RESTful クライアント

任意の HTTP クライアントを RESTful クライアントとして使用できます。リソース URL を入力すれば、BUI でも RESTful API GET の結果を返すことができます。Mozilla Firefox には、RESTful リクエストを作成するためにインストールできる RESTful クライアントモジュールがあります (<https://addons.mozilla.org/en-us/firefox/addon/restclient/>)。このモジュールは、PUT、POST、および DELETE リクエストと、通常の HTTP GET リクエストを許容します。

SSLv2/3 プロトコルはサポートされなくなったため、RESTful クライアントは TLS プロトコルを使用する必要があります。curl クライアントは curl バージョン 7.34.0 以上を使用する必要があります。

このセクションには、さまざまな RESTful クライアントに関する詳細な情報が記載されています。

curl REST クライアント

curl クライアントは curl バージョン 7.34.0 以上を使用する必要があります。一般的な 2 つの CLI ベース HTTP クライアントは、wget および curl です。このセクションでは、curl を使用して RESTful API コールを行う例をいくつか示しますが、wget を使用しても同様の機能を実現できます。

リソースデータの取得

この例では、単純な HTTP GET リクエストを使用して JSON データを取得する方法を示します。

```
$ curl --user ${USER}:${PASSWORD} -k \
-i https://hostname:215/api/storage/v1/pools/p1
```

```
HTTP/1.1 200 OK
Date: Tue, 23 Jul 2018 12:57:02 GMT
Server: WSGIServer/0.1 Python/2.6.4
Content-Length: 284
Content-Type: application/json
X-Zfs-Sa-Nas-API: 1.0
```

```
{
  "pool": {
    "profile": "mirror",
    "name": "p1",
    "usage": {
      "available": 895468984832.0,
      "total": 895500681216.0,
      "dedupratio": 100,
      "used": 31696384.0
    },
    "peer": "00000000-0000-0000-0000-000000000000",
    "state": "online",
    "owner": "admin1",
    "asn": "314d252e-c42b-e844-dab1-a3bca680b563"
  }
}
```

新しいリソースの作成

この例では、新しいリソースを作成するためにリクエストで JSON データを送信する方法を示します。

```
$ curl --user ${USER}:${PASSWORD} -s -k -i -X POST -d @- \
  -H "Content-Type: application/json" \
  https://zfs-storage.example.com:215/api/user/v1/users <<JSON
> {"logname": "rest_user",
> "fullname": "REST User",
> "initial_password": "password"}
> JSON
```

```
HTTP/1.1 201 Created
Date: Tue, 23 Jul 2018 13:07:37 GMT
Server: WSGIServer/0.1 Python/2.6.4
X-Zfs-Sa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 357
```

```
{
  "user": {
    "logname": "rest_user",
    "fullname": "REST User",
    "initial_password": "password",
    "require_annotation": false,
    "kiosk_mode": false,
    "kiosk_screen": "status/dashboard",
    "roles": ["basic"],
    "exceptions": {},
    "preferences": {
      "href": "/api/user/v1/users/admin1/preferences",
      "locale": "C",
      "login_screen": "status/dashboard",
      "session_timeout": 15,
      "cli_idle_timeout": -1,
      "advanced_analytics": false,
      "keys": {}
    }
  }
}
```

既存のリソースの変更

この例では、ユーザーのセッションタイムアウトを変更します。

```
$ curl --user admin1:password -3 -s -k -i -X PUT \
  -H "Content-Type: application/json" -d @- \
  https://zfs-storage.example.com:215/api/appliance/v1/users/admin1/preferences <<JSON
> {"session_timeout":60}
> JSON

HTTP/1.1 202 Accepted
Date: Wed, 24 Jul 2018 05:43:17 GMT
X-Zfs-Sa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 0

{
  "preferences": {
    "href": "appliance/v1/users/admin1/preferences",
    "locale": "C",
    "login_screen": "status/dashboard",
    "session_timeout": 60,
    "cli_idle_timeout": -1,
    "advanced_analytics": false,
    "keys": {}
  }
}
```

既存のリソースの削除

このコマンドは、システムからユーザーを除去します。

```
$ curl --user ${USER}:${PASSWORD} -s -k -i -X DELETE \
  https://zfs-storage.example.com:215/api/appliance/v1/users/admin1
HTTP/1.1 204 No Content
Date: Tue, 23 Jul 2018 13:21:11 GMT
Server: WSGIServer/0.1 Python/2.6.4
X-Zfs-Sa-Appliance-API: 1.0
Content-Length: 0
```

Python RESTful クライアント

Python RESTful API クライアントは、RESTful サービスのテスト開発を支援するために REST テストライブラリとともに提供されます。

RESTful クライアントプログラムの例:

```
>>> import urllib2
>>> import json

>>> request = urllib2.Request("https://zfsssa.example:215/api/access/v1", "")
>>> request.add_header("X-Auth-User", "rest_user")
>>> request.add_header("X-Auth-Key", "password")
```

```
>>> response = urllib2.urlopen(request)
>>> response.getcode()
201

>>> info = response.info()
>>>
>>> opener = urllib2.build_opener(urllib2.HTTPHandler)
>>> opener.addheaders = [("X-Auth-Session", info.getheader("X-Auth-Session")),
... ('Content-Type', 'application/json'), ('Accept', 'application/json')]
```

次に、オープナーを使用して、事前に承認済みで、JSON データを送受信する準備ができています。リクエストを開くことができます。

リソースの取得

次の Python コードを使用すると、任意の RESTful API リソースからデータを取得できます。

GET の例:

```
>>> request = urllib2.Request("https://zfs-storage.example.com:215/api/network/v1/routes")
>>> response = opener.open(request)
>>> response.getcode()
200
>>> body = json.loads(response.read())
>>> print json.dumps(body, sort_keys=True, indent=4)
{
  "routes": [
    {
      "destination": "ipaddr-0",
      "family": "IPv4",

      "gateway": "ipaddr-1",
      "href":
        "/api/network/v1/routes/ixgbe0,ipaddr-0,ipaddr-1",

      "interface": "ixgbe0",
      "mask": 0,
      "type": "static"
    }
  ]
}
```

リソースの作成

新規リソースを作成する Python コードの例:

```
>>> action = {'category': 'network'}
>>> post_data = json.dumps(action)
>>> request = urllib2.Request("https://zfs-storage.example.com:215/api/alert/v1/actions",
... post_data)
>>> request.add_header('Content-Type', 'application/json')
```

```

>>> response = opener.open(request)
>>> response.getcode()
201
>>> response.info().getheader('Location')
'/api/alert/v1/actions/actions-001'
>>> body = json.loads(response.read())
>>> print json.dumps(body, sort_keys=True, indent=4)
{
    "actions": {
        "category": "network",
        "datalink_failed": true,

        "datalink_ok": true,
        "href":
        "/api/alert/v1/actions/actions-001",

        "ip_address_conflict": true,

        "ip_address_conflict_resolved": true,

        "ip_interface_degraded": true,
        "ip_interface_failed":
        true,
        "ip_interface_ok": true,

        "network_port_down": true,
        "network_port_up":
        true
    }
}

```

リソースの変更

既存のリソースを変更する Python コードの例:

```

>>> put_data = '{"ip_address_conflict_resolved": false}'
>>>
        request = urllib2.Request("https://zfs-storage.example.com:215/api/alert/v1/
actions/actions-001", put_data)
>>> request.add_header('Content-Type', 'application/json')
>>> request.get_method = lambda: 'PUT'

>>> response = opener.open(request)
>>> response.getcode()
202
>>> body = json.loads(response.read())
>>> print json.dumps(body, sort_keys=True, indent=4)
{
    "actions": {
        "category": "network",
        "datalink_failed": true,

        "datalink_ok": true,
        "href":
        "/api/alert/v1/actions/actions-001",

        "ip_address_conflict": true,

```

```
        "ip_address_conflict_resolved": false,  
  
        "ip_interface_degraded": true,  
        "ip_interface_failed":  
        true,  
        "ip_interface_ok": true,  
  
        "network_port_down": true,  
        "network_port_up":  
        true  
    }  
}
```

既存のリソースの削除

既存のリソースを削除する Python コードの例:

```
>>> request = urllib2.Request("https://zfs-storage.example.com:215/api/alert/v1/actions/  
actions-001")  
>>> request.get_method = lambda: 'DELETE'  
>>> response = opener.open(request)  
>>> response.getcode()  
204
```