

**Oracle® ZFS Storage Appliance RESTful
API ガイド、Release OS8.6.x**

ORACLE®

Part No: E78585-01
2016 年 9 月

Part No: E78585-01

Copyright © 2015, 2016, Oracle and/or its affiliates. All rights reserved.

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 installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. 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場合、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したこと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 Xeon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, Opteron, AMDロゴ、AMD Opteron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	13
RESTful API 認証	13
RESTful API バージョン	14
サービスのバージョン	14
一般的な RESTful 操作	15
HTTP 応答本文	15
HTTP 応答ヘッダー	16
パラメータの問合せ	17
問合せパラメータ: props	17
問合せパラメータ: limit	18
問合せパラメータ: start	18
問い合わせパラメータ: depth	18
問い合わせパラメータ: match	20
アプライアンスエラー	20
RESTful API の操作	23
サービスへのアクセス	23
サービスの一覧表示	23
サービスの取得のコマンド	24
認証セッション	25
ログインセッション	25
ログアウトセッション	26
RESTful API 警告サービス	27
警告サービスコマンド	27
警告しきい値	28
警告しきい値の一覧表示	29
警告しきい値の取得	30
警告しきい値の作成	31

警告しきい値の変更	31
警告しきい値の削除	32
警告アクション	32
警告アクションの一覧表示	42
警告アクションの取得	43
警告アクションの作成	44
警告アクションの変更	44
警告アクションの削除	45
警告アクション項目	45
警告項目の作成	45
警告アクションの変更	46
警告アクション項目の削除	47
Analytics サービス	49
使用可能な Analytics サービス	49
Analytics の設定	50
設定の取得	50
設定の変更	51
Analytics のワークシート	51
ワークシートの一覧表示	52
分析ワークシートの取得	52
ワークシートの作成	53
ワークシートの破棄	53
ワークシートデータセットの一覧表示	54
ワークシートデータセットの追加	54
ワークシートデータセットの変更	55
Analytics のデータセット	55
データセットの一覧表示	57
データセットの取得	57
データセットの作成	58
データセットの変更	59
データセットの破棄	59
データセットの保存	59
データセットデータの除去	60
データセットデータの取得	60
ハードウェアサービス	65
クラスタ	65

クラスタのプロパティの取得	65
クラスタリソースの取得	66
クラスタリソースの変更	66
クラスタのコマンド	67
クラスタリンク	67
クラスタの設定	68
シャーシ	68
シャーシの一覧表示	68
シャーシのコンポーネントの取得	70
ハードウェアコンポーネントの取得	71
コンポーネントのプロパティの変更	72
ログのコマンド	75
ログのコマンド	75
ログの一覧表示	75
ログエントリの取得	76
ログのダウンロード	77
ログのダウンロード	78
ネットワークのコマンド	79
ネットワーク構成	79
ネットワークデータリンク	80
ネットワークデータリンクの一覧表示	82
ネットワークデータリンクの取得	82
ネットワークデータリンクの作成	83
ネットワークデータリンクの変更	83
ネットワークデータリンクの削除	84
ネットワークデバイス	84
ネットワークデバイスの一覧表示	85
ネットワークデバイスの取得	85
ネットワークインタフェース	86
ネットワークインタフェースの一覧表示	87
ネットワークインタフェースの取得	88
ネットワークインタフェースの作成	88
ネットワークインタフェースの変更	89
ネットワークインタフェースの削除	89
ネットワークルート	90
ルートの一覧表示	90

ルートの取得	91
ルートの追加	92
ルートの削除	92
RESTful API 問題サービス	93
問題サービスコマンド	93
すべての問題の一覧表示	93
一つの問題の一覧表示	94
問題の修復	95
RESTful API ロールサービス	97
ロールサービスコマンドの概要	97
List Roles	98
Get Role	99
Create Role	99
Modify Role	100
Revoke Role	101
Delete Role	101
List Role Authorizations	101
Create Role Authorization	102
Modify Role Authorization	102
Delete Role Authorization	103
RESTful API SAN サービス	105
SAN の概要	105
SAN イニシエータ	106
List Initiators	107
Get Initiator Details	107
Create an Initiator	108
Modify an Initiator	108
Delete an Initiator	109
イニシエータグループ	109
List Initiator Groups	110
Get Initiator Group Details	111
Create an Initiator Group	111
Delete an Initiator Group	112
ターゲット	112
List Targets	114

Get Target Details	115
Create a Target	115
Modify a Target	116
Delete a Target	117
ターゲットグループ	117
List Target Groups	118
Get Target Group	118
Create a Target Group	119
Delete a Target Group	119
サービスコマンド	121
サービスコマンド	121
サービスの一覧表示	121
Get Service	124
Change Service State	125
サービス構成の変更	126
サービスリソース	126
RESTful API ストレージサービス	129
ストレージプールの操作	129
プールの一覧表示	130
プールの取得	131
プールの構成	131
プールへのストレージの追加	133
プールからのストレージの削除	133
プールのスクラブ	134
プールの構成解除	135
プロジェクトの操作	135
プロジェクトの一覧表示	137
プロジェクトのプロパティの取得	138
プロジェクトの作成	140
プロジェクトの変更	141
プロジェクトの削除	141
プロジェクトの使用状況	142
ファイルシステムの操作	142
ファイルシステムの一覧表示	144
ファイルシステムの取得	145
ファイルシステムの作成	146

ファイルシステムの変更	147
ファイルシステムの削除	148
ファイルシステムの割り当て制限および使用状況	149
LUN 操作	149
LUN の一覧表示	151
LUN の取得	151
新しい LUN の作成	153
LUN の変更	154
LUN の削除	155
スナップショットおよびクローンの操作	155
スナップショットの一覧表示	158
スナップショットの取得	158
スナップショットの作成	159
スナップショットの名前変更	160
スナップショットのクローン作成	160
スナップショットのロールバック	162
スナップショットの削除	163
スナップショット依存の一覧表示	164
スキーマ	165
プロパティの一覧表示	165
プロパティの取得	166
プロパティの作成	166
プロパティの変更	167
プロパティの削除	167
レプリケーション	167
レプリケーションサービスの取得	168
レプリケーションサービスの状態の変更	169
レプリケーションターゲット	169
レプリケーションターゲットの一覧表示	169
レプリケーションターゲットの取得	170
レプリケーションターゲットの作成	171
レプリケーションターゲットの削除	171
レプリケーションアクション	172
レプリケーションアクションの一覧表示	175
レプリケーションアクションの取得	176
レプリケーションアクションの作成	176
レプリケーションアクションの変更	178
更新の取り消し	178

更新の送信	179
レプリケーションアクションの削除	179
レプリケーションパッケージ	179
レプリケーションソースの一覧表示	183
レプリケーションパッケージの一覧表示	184
パッケージの変更	185
パッケージの削除	185
更新の取り消し	185
パッケージのクローン作成	186
パッケージの切断	187
パッケージの逆方向化	187
暗号化	188
すべてのローカル鍵の一覧表示	189
ローカル鍵の一覧表示	189
すべての OKM 鍵の一覧表示	190
システムコマンド	191
アプライアンスシステムコマンド	191
バージョンの取得	192
システムの電源切断	192
システムのリブート	193
システム管理の再起動	193
診断リブート	193
出荷時リセット	193
システムのサポートバンドル	194
サポートバンドルの作成	194
サポートバンドルの一覧表示	195
サポートバンドルの取得	196
サポートバンドルの取り消し	196
サポートバンドルのアップロードの再試行	197
サポートバンドルのアップロード	197
サポートバンドルの削除	198
システムの更新	198
システム更新の一覧表示	199
システム更新の取得	200
システム更新のアップロード	200
アップグレード	201
ロールバック	201

更新イメージの削除	202
RESTful API ユーザーサービス	203
ユーザーサービスのコマンド	203
ユーザーの一覧表示	204
ユーザーの取得	205
ユーザーの作成	206
ユーザーの変更	210
ユーザーの削除	211
ワークフローコマンド	213
ワークフローサービスコマンド	213
ワークフローの一覧表示	214
ワークフローの取得	214
ワークフローの変更	216
ワークフローの実行	217
ワークフローの削除	217
ワークフローのアップロード	217
RESTful クライアント	219
curl REST クライアント	219
リソースデータの取得	219
新しいリソースの作成	220
既存のリソースの変更	220
既存のリソースの削除	221
Python RESTful クライアント	221
リソースの取得	222
リソースの作成	222
リソースの変更	223
既存のリソースの削除	224

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、BUI および 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。

HTTP ヘッダーの例:

```
Authorization: Basic abcdefgMWE
```

- **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: root
```

```
X-Auth-Key: letmein-xxx
```

- **セッション認証** - セッションが認証されると、セッションが期限切れになるまで、セッショ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。セッショ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。

セッションヘッダーの例:

X-Auth-Session: guigqpQRE4g89ngb

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

サービスのバージョン

各サービスは、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、デフォルトが「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。

例: クライアントがバージョン番号「2.1」を実行しているサービスに対し、いくつかのリクエストを作成します。

リクエストのバージョン	許可
v1	F alse - メジャー番号が一致していません
v2	T ru e - メジャー番号が一致していますが、マイナー番号は下位互換性があります
v2.1	T ru e - メジャー番号とマイナー番号が一致しています
v2.2	F alse - メジャー番号は一致していますが、マイナー番号が新しいリビジョンです

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

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 \(http://tools.ietf.org/html/rfc4627.html\)](http://tools.ietf.org/html/rfc4627.html) 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。

パラメータの問合せ

一部のリクエストでは、返されるデータを変更または拡張するオプションの問合せパラメータを取ります。詳細については、各リソースのドキュメントを参照してください。すべてのリソースがすべての問合せパラメータをサポートしているわけではありません。このセクションでは、リソースが指定された問合せパラメータを実装する場合に使用される一般的な問合せパラメータについてのみ説明します。

表 2 一般的な問合せパラメータ

パラメータ	説明
props=true	リソースのプロパティメタデータを一覧表示します (デフォルトは false)
limit=n	返されるリスト要素の数を制限します
start=n	返される要素データの開始に使われるインデックス番号 (または時間)
depth=n	返されるデータの詳細レベルを指定するインデックス番号
match_Property-Name=Value	指定されたプロパティ名と値に一致するデータを一覧表示します

問合せパラメータ: props

props 問い合わせパラメータは GET、POST、PUT の各コマンドで使用可能であり、エンドユーザーがメタデータにアクセスできるようにします。エンドユーザーからこの機能を要求するには、問い合わせパラメータ props を true に設定します。GET 操作と PUT 操作の場合、返された JSON オブジェクトには、要求したデータのほかに、プロパティのメタデータのリストが含まれています。POST の場合、ユーザーがリソースを適切に作成することを支援するメタデータのみが返されます。

表 3 プロパティメタデータ値

プロパティ	説明
name	プロパティ名
label	プロパティの説明
immutable	プロパティを変更できないことを示すフラグ
type	プロパティの型: String、Integer、Boolean...
choices	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: limit

limit 問合せは、返される要素の最大数を制限するために、多数の要素を返す可能性のある多くの GET コマンドで使用できます。

問合せパラメータ: start

start 問合せパラメータは、limit パラメータをサポートする同じコマンドでサポートされています。返されるデータの開始に使用される開始インデックスを指定します。時間値をサポートするリソースの場合、インデックスは「20130531T01:13:58」などの時間値になることがあります。

問い合わせパラメータ: depth

depth 問い合わせパラメータは、リソースのリストを取得するために GET コマンドで使用できます。これは、返されるリストの詳細レベルを指定するために使用します。depth の数値が大きいほど、より詳しい情報が返されます。

問い合わせパラメータ depth を使用したリクエストの例:

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

この例では、ユーザーのリストが depth=2 までの詳細度で返されます。

レスポンスの例:

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

{"users":
  [
    {
      "name": "root",
      "properties": {
        "logname": "root",
        "fullname": "Super-User",
        "initial_password": "DummyPassword",
        "require_annotation": false
      },
      "children": [
        {
          "name": "preferences",
          "properties": {
            "locale": "C",
            "login_screen": "status/dashboard",
            "session_timeout": 15,
            "advanced_analytics": false
          },
          "children": [

```

```

    "name": "keys",
    "properties": {},
    "children": [],
    "list": []
  }],
  "list": []
}],
"list": [],
"href": "/api/user/v1/users/root"
  },
  {
    "name": "tom",
    "properties": {
      "logname": "tom",
      "fullname": "Tommy",
      "initial_password": "DummyPassword",
      "require_annotation": false,
      "roles": ["basic"],
      "kiosk_mode": false,
      "kiosk_screen": "status/dashboard"
    },
    "children": [{
      "name": "exceptions",
      "properties": {},
      "children": [],
      "list": [{
        "name": "auth-000",
        "properties": {
          "scope": "stat",
          "drilldowns": "*",
          "allow_create": false,
          "allow_read": true
        },
        "children": [],
        "list": []
      },
      {
        "name": "auth-001",
        "properties": {
          "scope": "ad",
          "name": "*",
          "allow_domain": true,
          "allow_workgroup": false
        },
        "children": [],
        "list": []
      }
    ]
  }, {
    "name": "preferences",
    "properties": {
      "locale": "C",
      "login_screen": "status/dashboard",
      "session_timeout": 15,
      "advanced_analytics": false
    },
    "children": [{
      "name": "keys",
      "properties": {},
      "children": [],
      "list": ["key-000"]
    }],
    "list": []
  },
  "list": [],

```

```

    "href": "/api/user/v1/users/tom"
  }]
}

```

問い合わせパラメータ: match

match_Property-Name=Value 問い合わせパラメータは、リソースのリストを取得するために GET コマンドで使用できます。指定されたプロパティ名と値に一致するデータのリストが返されます。

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

名前	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
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	リクエストは、完了するために confirm=true の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

名前	コード	説明
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

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

このサービスにアクセスするには、`http://zfssa.example.com:215/api/access/v1` という URL を使用します

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	/api/access/v1	RESTful API サービスのアクセスポイントを一覧表示します
POST	/api/access/v1	ログインセッションを作成します
DELETE	/api/access/v1	セッションからログアウトします

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
X-Auth-User: joeadmin
X-Auth-Key: letmein
```

結果例:

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

{
  "access": {
    "services": [{
      "version": "1.0",
      "name": "appliance",
      "uri": "https://zfs-storage.example.com:215/api/appliance/v1"
    }, {
      "version": "1.0",
      "name": "nas",
      "uri": "https://zfs-storage.example.com:215/api/nas/v1"
    }, {
      "version": "1.0",
      "name": "replication",
      "uri": "https://zfs-storage.example.com:215/api/replication/v1"
    }, {
      "version": "1.0",
      "name": "san",
      "uri": "https://zfs-storage.example.com:215/api/san/v1"
    } ... ]
  }
}
```

サービスの取得のコマンド

`get service` コマンドは、そのサービスに関する情報 (使用可能なすべてのコマンドのリストを含む) を返します。

リクエストの例:

```
GET /api/appliance/v1 HTTP/1.1
Host: zfs-storage.example.com
X-Auth-Session: guigqpQRE4g89ngb
```

レスポンスの例:

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

{
  "service": {
    "name": "appliance",
    "methods": [
      {
        "description": "Get appliance RESTful services",
        "path": "/apis",

```

```

    "request": "GET"
  },
  {
    "description": "Get appliance RESTful service properties",
    "path": "/apis/<api:path>",
    "request": "GET"
  },
  {
    "description": "Create a new alert threshold watch",
    "path": "/alerts/thresholds",
    "request": "POST"
  }, ... ]
}
}

```

認証セッション

認証セッション ID 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。この認証セッション ID は、その他すべてのサービスで同一の資格証明として使用できます。認証 ID 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、ログアウトしてセッション ID を無効化するために使用できます。

クライアントはリクエストごとに認証情報を再送信できるため、認証セッションは必要ありません。RESTful API 操作はステートレスであるため、認証 ID のみが格納されます。

ログインセッション

空の POST リクエストは、新しいログインセッショ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。

ログインリクエストの例:

```

POST /api/access/v1 HTTP/1.1
Host: zfs-storage.example.com
X-Auth-User: root
X-Auth-Key: letmein-xxx

```

ログインに成功すると、HTTP ステータス 201 (Created) と、X-Auth-Session HTTP ヘッダーを介してセッション ID 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: guigqpQRE4g89ngb

```

```
Content-Type: application/json
Content-Length: 378
X-Zfssa-Access-API: 1.0
```

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

ログアウトセッション

空の DELETE は、ログアウトしてセッションを無効化するための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/v1 HTTP/1.1
X-Auth-Session: guigqpQRE4g89ngb
```

レスポンスの例:

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

RESTful API 警告サービス

警告 RESTful API サービスでは、警告しきい値を構成でき、送信された警告に対して応答します。

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。

表 6 警告サービスコマンド

リクエスト	パス <code>/api/alert/v1</code>	説明
GET		警告サービスコマンドを一覧表示します
POST	<code>/thresholds</code>	新しいアラートしきい値ウォッチを作成します
GET	<code>/thresholds/<threshold></code>	指定されたアラートしきい値ウォッチプロパティを取得します
GET	<code>/thresholds</code>	すべてのアラートしきい値ウォッチオブジェクトを一覧表示します
PUT	<code>/thresholds/<threshold></code>	指定されたアラートしきい値ウォッチオブジェクトを変更します
DELETE	<code>/thresholds/<threshold></code>	指定されたしきい値オブジェクトを破棄します
POST	<code>/actions</code>	新しいアラートアクションを作成します
GET	<code>/actions/<actions></code>	指定されたアラートアクションプロパティを取得します
GET	<code>/actions</code>	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
PUT	<code>/actions/<actions></code>	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	<code>/actions/<actions></code>	指定されたアクションオブジェクトを破棄します

リクエスト	パス <code>/api/alert/v1</code>	説明
POST	<code>/actions/<actions></code>	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	<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プロパティを取得します
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
DELETE	<code>/actions/<actions>/<action></code>	指定されたアクション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

警告しきい値

しきい値を設定して、カスタムの警告ウォッチを作成できます。次の表に、警告のしきい値を管理するための一般的なプロパティを示します。詳細は、CLI のヘルプを参照してください。

表 7 警告しきい値

プロパティ	型	説明
<code>uuid</code>	Default	ウォッチの一意的識別子 (変更不可) です
<code>statname</code>	AnalyticsStatistics	モニターする統計 [“cpu.utilization”, “arc.accesses”, “arc.size”, “arc.l2_bytes”, “arc.l2_accesses”, “arc.l2_size”, “syscap.bytesused”, “syscap.percentused”, “repl.bytes”, “repl.ops”, “shadow.kilobytes”, “shadow.ops”, “shadow.requests”, “io.bytes”, “io.ops”, “datalink.kilobytes”, “nic.kilobytes”, “net.kilobytes”, “ftp.kilobytes”, “fc.bytes”, “fc.ops”, “http.reqs”, “ndmp.bytes”, “ndmp.diskkb”, “ndmp.ops”, “nfs2.bytes”, “nfs2.ops”, “nfs3.bytes”, “nfs3.ops”, “nfs4.bytes”, “nfs4.ops”, “sftp.kilobytes”, “smb.ops”, “srp.bytes”, “srp.ops”, “iscsi.bytes”, “iscsi.ops”]
<code>type</code>	ChooseOne	stat が制限を超える (normal) か、制限を下回る (inverted) [「normal」, 「inverted」] 場合に警告を送信するかどうか
<code>limit</code>	PositiveInteger	統計の制限値です

プロパティ	型	説明
minpost	Duration	警告を送信する前に最小時間条件を保持する必要があります
days	ChooseOne	特定の日 [「all」、「weekdays」, 「weekends」] のみ警告を送信します
window_start	TimeOfDay	window_start と window_end 間 [“none”, “00:00”, “00:30”, “01:00”, “01:30”, “02:00”, “02:30”, “03:00”, “03:30”, “04:00”, “04:30”, “05:00”, “05:30”, “06:00”, “06:30”, “07:00”, “07:30”, “08:00”, “08:30”, “09:00”, “09:30”, “10:00”, “10:30”, “11:00”, “11:30”, “12:00”, “12:30”, “13:00”, “13:30”, “14:00”, “14:30”, “15:00”, “15:30”, “16:00”, “16:30”, “17:00”, “17:30”, “18:00”, “18:30”, “19:00”, “19:30”, “20:00”, “20:30”, “21:00”, “21:30”, “22:00”, “22:30”, “23:00”, “23:30”] のみ警告を送信
window_end	TimeOfDay	window_start と window_end 間 [“none”, “00:00”, “00:30”, “01:00”, “01:30”, “02:00”, “02:30”, “03:00”, “03:30”, “04:00”, “04:30”, “05:00”, “05:30”, “06:00”, “06:30”, “07:00”, “07:30”, “08:00”, “08:30”, “09:00”, “09:30”, “10:00”, “10:30”, “11:00”, “11:30”, “12:00”, “12:30”, “13:00”, “13:30”, “14:00”, “14:30”, “15:00”, “15:30”, “16:00”, “16:30”, “17:00”, “17:30”, “18:00”, “18:30”, “19:00”, “19:30”, “20:00”, “20:30”, “21:00”, “21:30”, “22:00”, “22:30”, “23:00”, “23:30”] (変更不可) のみ警告を送信
frequency	Duration	警告送信前の最小時間です
minclear	Duration	「all clear」アラートを再送する前の正常性の最小時間です

警告しきい値の一覧表示

すべての構成済みの警告しきい値を一覧表示します。

リクエストの例:

```
GET /api/alert/v1/thresholds HTTP/1.1
Authorization: Basic abcd123MWE=
Host: zfssa.example.com:215
Accept: application/json
```

レスポンスの例:

```
HTTP/1.1 200 OK
Date: Tue, 27 Aug 2013 17:38:40 GMT
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 689

{
  "thresholds": [
    {
      "days": "all",
      "frequency": 300,
      "href": "/api/alert/v1/thresholds/
        bec758cb-346e-6a7d-c211-b320c09ef6a6",
      "limit": 500,
      "minclear": 300,
      "minpost": 300,
      "statname": "cpu.utilization",
      "threshold": "threshold-000",
      "type": "normal",
      "uuid": "bec758cb-346e-6a7d-c211-b320c09ef6a6",
      "window_end": 0,
      "window_start": -1
    },
    {
      "days": "all",
      "frequency": 300,
      "href": "/api/alert/v1/thresholds/
        475799d8-32c8-6ff6-882c-aa3b66e3a5a2",
      "limit": 100000,
      "minclear": 600,
      "minpost": 300,
      "statname": "datalink.kilobytes",
      "threshold": "threshold-001",
      "type": "normal",
      "uuid": "475799d8-32c8-6ff6-882c-aa3b66e3a5a2",
      "window_end": 300,
      "window_start": 1200
    }
  ]
}
```

警告しきい値の取得

単一の警告しきい値のプロパティを一覧表示します。

リクエストの例:

```
GET /api/alert/v1/thresholds/1b15d405-75c4-4c0c-e0f6-8a108165b874
HTTP/1.1
Authorization: Basic abcd123MWE=
Host: zfssa.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: 363
```

```
{
  "threshold": {
    "days": "weekdays",
    "frequency": 300,
    "href": "/api/alert/v1/thresholds/
             1b15d405-75c4-4c0c-e0f6-8a108165b874",
    "limit": 100000,
    "minclear": 300,
    "minpost": 300,
    "statname": "datalink.kilobytes",
    "type": "normal",
    "uuid": "1b15d405-75c4-4c0c-e0f6-8a108165b874",
    "window_end": 0,
    "window_start": -1
  }
}
```

警告しきい値の作成

警告しきい値を作成します。

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: letmein
Content-Type: application/json
Content-Length: 50
```

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

レスポンスの例:

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 321
Location: /api/alert/v1/thresholds
           /1b15d405-75c4-4c0c-e0f6-8a108165b874
```

```
{
  "threshold": {
    "href": "/api/alert/v1/alerts/thresholds
             /1b15d405-75c4-4c0c-e0f6-8a108165b874",
    ...
  }
}
```

警告しきい値の変更

指定された警告しきい値のプロパティを変更します。

リクエストの例:

```
PUT /api/alert/v1/thresholds/1b15d405-75c4-4c0c-e0f6-8a108165b874
```

```
HTTP/1.1
Authorization: Basic abcd123MWE=
Host: zfssa.example.com:215
```

```
{"days": "weekdays"}
```

レスポンスの例:

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

```
{
  "threshold": {
    "days": "weekdays",
    ...
  }
}
```

警告しきい値の削除

指定された警告しきい値を削除します。

リクエストの例:

```
DELETE /api/alert/v1/thresholds/475799d8-32c8-6ff6-882c-aa3b66e3a5a2
HTTP/1.1
Authorization: Basic abcd123MWE=
Host: zfssa.example.com:215
```

レスポンスの例:

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

警告アクション

カテゴリプロパティにより、定義される警告アクションのタイプが決定されます。各カテゴリには、定義済みの独自のプロパティセットがあります。

サポートされているカテゴリ:

- ad
- all
- appliance_software
- backup
- cluster
- custom
- hardware

- hardware_faults
- ndmp
- network
- replication
- replication_source
- replication_target
- restore
- scrk
- shadow
- smf
- thresholds
- zfs_pool

表 8 警告アクション「ad」

プロパティ	型	説明
active_directory_degraded	Boolean	フィルタは active_directory_degraded イベント [true または false] に一致する必要があります
smb_kerberos_client_authentication_degraded	Boolean	フィルタは mb_kerberos_client_authentication_degraded イベント [true または false] に一致する必要があります

表 9 警告アクション「all」

プロパティ	型	説明
all_defects	Boolean	フィルタは all_defects イベント [true、false] に一致する必要があります
service_alerts	Boolean	フィルタは service_alerts イベント [true、false] に一致する必要があります
all_hardware_faults	Boolean	フィルタは all_hardware_faults イベント [true、false] に一致する必要があります

表 10 警告アクション「appliance_software」

プロパティ	型	説明
obstacles_to_system_software_update	Boolean	フィルタは obstacles_to_system_software_update イベント [true、false] に一致する必要があります
operating_system_kernel_panic	Boolean	フィルタは operating_system_kernel_panic イベント

プロパティ	型	説明
		フィルタは backup_finished イベント [true または false] に一致する必要があります

表 11 警告アクション「backup」

プロパティ	型	説明
backup_finished	Boolean	フィルタは backup_finished イベント [true または false] に一致する必要があります
backup_started	Boolean	フィルタは backup_started イベント [true または false] に一致する必要があります

表 12 警告アクション「cluster」

プロパティ	型	説明
cluster_i/o_link_down	Boolean	フィルタは cluster_i/o_link_down イベント [true または false] に一致する必要があります
cluster_i/o_link_failed	Boolean	フィルタは cluster_i/o_link_failed イベント [true または false] に一致する必要があります
cluster_i/o_link_up	Boolean	フィルタは cluster_i/o_link_up イベント [true または false] に一致する必要があります
unexpected_peer_error_occurred	Boolean	フィルタは unexpected_peer_error_occurred イベント [true または false] に一致する必要があります
communication_to_peer_lost	Boolean	フィルタは communication_to_peer_lost イベント [true または false] に一致する必要があります
cluster_peer_panicked	Boolean	フィルタは cluster_peer_panicked イベント [true または false] に一致する必要があります
failed_to_set_sp_root_password_on_cluster_peer	Boolean	フィルタは failed_to_set_sp_root_password_on_cluster_peer イベント [true または false] に一致する必要があります
cluster_rejoin_failed_on_peer	Boolean	フィルタは cluster_rejoin_failed_on_peer イベント [true または false] に一致する必要があります
cluster_rejoin_mismatch_on_peer	Boolean	フィルタは cluster_rejoin_mismatch_on_peer イベント [true または false] に一致する必要があります
cluster_rejoin_completed_on_peer	Boolean	フィルタは cluster_rejoin_completed_on_peer イベント [true または false] に一致する必要があります

プロパティ	型	説明
cluster_peer_lost_communication_token	Boolean	フィルタは cluster_peer_lost_communication_token イベント [true または false] に一致する必要があります
cluster_rejoin_failed	Boolean	フィルタは cluster_rejoin_failed イベント [true または false] に一致する必要があります
cluster_rejoin_mismatch	Boolean	フィルタは cluster_rejoin_mismatch イベント [true または false] に一致する必要があります
cluster_rejoin_completed	Boolean	フィルタは cluster_rejoin_completed イベント [true または false] に一致する必要があります
cluster_takeover_complete	Boolean	フィルタは cluster_takeover_complete イベント [true または false] に一致する必要があります
resource_import_failed_during_cluster_takeover	Boolean	フィルタは resource_import_failed_during_cluster_takeover イベント [true または false] に一致する必要があります
local_cluster_communication_token_lost	Boolean	フィルタは local_cluster_communication_token_lost イベント [true または false] に一致する必要があります

表 13 警告アクション「custom」

プロパティ	型	説明
patterns	Default	FMA イベントのパターン

表 14 警告アクション「hardware」

プロパティ	型	説明
fibre_channel_port_down	Boolean	フィルタは fibre_channel_port_down イベント [true または false] に一致する必要があります
multiple_transient_fibre_channel_port_status_changes	Boolean	フィルタは multiple_transient_fibre_channel_port_status_changes イベント [true または false] に一致する必要があります
transient_fibre_channel_port_status_change	Boolean	フィルタは transient_fibre_channel_port_status_change イベント [true または false] に一致する必要があります
fibre_channel_port_up	Boolean	フィルタは fibre_channel_port_up イベント [true または false] に一致する必要があります

プロパティ	型	説明
network_port_down	Boolean	フィルタは network_port_down イベント [true または false] に一致する必要があります
network_port_up	Boolean	フィルタは network_port_up イベント [true または false] に一致する必要があります
chassis_connected_to_system	Boolean	フィルタは chassis_connected_to_system イベント [true または false] に一致する必要があります
chassis_removed	Boolean	フィルタは chassis_removed イベント [true または false] に一致する必要があります
hardware_component_inserted	Boolean	フィルタは hardware_component_inserted イベント [true または false] に一致する必要があります
hardware_component_removed	Boolean	フィルタは hardware_component_removed イベント [true または false] に一致する必要があります
disk_inserted	Boolean	フィルタは disk_inserted イベント [true または false] に一致する必要があります
disk_removed	Boolean	フィルタは disk_removed イベント [true または false] に一致する必要があります
i/o_path_added	Boolean	フィルタは i/o_path_added イベント [true または false] に一致する必要があります
i/o_path_removed	Boolean	フィルタは i/o_path_removed イベント [true または false] に一致する必要があります
service_processor_offline_or_unavailable	Boolean	フィルタは service_processor_offline_or_unavailable イベント [true または false] に一致する必要があります
service_processor_online_after_outage	Boolean	フィルタは service_processor_online_after_outage イベント [true または false] に一致する必要があります
failed_to_set_root_password_on_service_processor	Boolean	フィルタは failed_to_set_root_password_on_service_processor イベント [true または false] に一致する必要があります

表 15 警告アクション「hardware_faults」

プロパティ	型	説明
all_hardware_faults	Boolean	フィルタは all_hardware_faults イベント [true または false] に一致する必要があります

表 16 警告アクション「ndmp」

プロパティ	型	説明
invalid_ndmp_restore	Boolean	フィルタは invalid_ndmp_restore イベント [true または false] に一致する必要があります
backup_finished	Boolean	フィルタは backup_finished イベント [true または false] に一致する必要があります
backup_started	Boolean	フィルタは backup_started イベント [true または false] に一致する必要があります
restore_finished	Boolean	フィルタは restore_finished イベント [true または false] に一致する必要があります
restore_started	Boolean	フィルタは restore_started イベント [true または false] に一致する必要があります

表 17 警告アクション「network」

プロパティ	型	説明
datalink_failed	Boolean	フィルタは datalink_failed イベント [true または false] に一致する必要があります
datalink_ok	Boolean	フィルタは datalink_ok イベント [true または false] に一致する必要があります
network_port_down	Boolean	フィルタは network_port_down イベント [true または false] に一致する必要があります
network_port_up	Boolean	フィルタは network_port_up イベント [true または false] に一致する必要があります
ip_address_conflict	Boolean	フィルタは ip_address_conflict イベント [true または false] に一致する必要があります
ip_address_conflict_resolved	Boolean	フィルタは ip_address_conflict_resolved イベント [true または false] に一致する必要があります
ip_interface_degraded	Boolean	フィルタは ip_interface_degraded イベント [true または false] に一致する必要があります
ip_interface_failed	Boolean	フィルタは ip_interface_failed イベント [true または false] に一致する必要があります

プロパティ	型	説明
ip_interface_ok	Boolean	フィルタは ip_interface_ok イベント [true または false] に一致する必要があります

表 18 警告アクション「replication」

プロパティ	型	説明
receive_failed_(unsupported_version)	Boolean	フィルタは receive_failed_(unsupported_version) イベント [true または false] に一致する必要があります
receive_failed_(cancelled)	Boolean	フィルタは receive_failed_(cancelled) イベント [true または false] に一致する必要があります
receive_failed_(all_others)	Boolean	フィルタは receive_failed_(all_others) イベント [true または false] に一致する必要があります
receive_failed_(out_of_space)	Boolean	フィルタは receive_failed_(out_of_space) イベント [true または false] に一致する必要があります
receive_failed_(package_not_upgraded)	Boolean	フィルタは receive_failed_(package_not_upgraded) イベント [true または false] に一致する必要があります
receive_finished	Boolean	フィルタは receive_finished イベント [true または false] に一致する必要があります
receive_started	Boolean	フィルタは receive_started イベント [true または false] に一致する必要があります
send_failed_(unsupported_version)	Boolean	フィルタは send_failed_(unsupported_version) イベント [true または false] に一致する必要があります
send_failed_(cancelled)	Boolean	フィルタは send_failed_(cancelled) イベント [true または false] に一致する必要があります
send_failed_(all_others)	Boolean	フィルタは send_failed_(all_others) イベント [true または false] に一致する必要があります
send_failed_(connectivity)	Boolean	フィルタは send_failed_(connectivity) イベント [true または false] に一致する必要があります
send_failed_(out_of_space)	Boolean	フィルタは send_failed_(out_of_space) イベント [true また

プロパティ	型	説明
		は false] に一致する必要があります
send_failed_(remote_verification)	Boolean	フィルタは send_failed_(remote_verification) イベント [true または false] に一致する必要があります
send_finished	Boolean	フィルタは send_finished イベント [true または false] に一致する必要があります
send_skipped_(already_running)	Boolean	フィルタは send_skipped_(already_running) イベント [true または false] に一致する必要があります
send_started	Boolean	フィルタは send_started イベント [true または false] に一致する必要があります

表 19 警告アクション「replication_source」

プロパティ	型	説明
send_failed_(unsupported_version)	Boolean	フィルタは send_failed_(unsupported_version) イベント [true または false] に一致する必要があります
send_failed_(cancelled)	Boolean	フィルタは send_failed_(cancelled) イベント [true または false] に一致する必要があります
send_failed_(all_others)	Boolean	フィルタは send_failed_(all_others) イベント [true または false] に一致する必要があります
send_failed_(connectivity)	Boolean	フィルタは send_failed_(connectivity) イベント [true または false] に一致する必要があります
send_failed_(out_of_space)	Boolean	フィルタは send_failed_(out_of_space) イベント [true または false] に一致する必要があります
send_failed_(remote_verification)	Boolean	フィルタは send_failed_(remote_verification) イベント [true または false] に一致する必要があります
send_finished	Boolean	フィルタは send_finished イベント [true または false] に一致する必要があります
send_skipped_(already_running)	Boolean	フィルタは send_skipped_(already_running) イベント [true ま

プロパティ	型	説明
		たは false] に一致する必要があります
send_started	Boolean	フィルタは send_started イベント [true または false] に一致する必要があります

表 20 警告アクション「replication_target」

プロパティ	型	説明
receive_failed_ (unsupported_version)	Boolean	フィルタは receive_failed_ (unsupported_version) イベント [true または false] に一致する必要があります
receive_failed_(cancelled)	Boolean	フィルタは receive_failed_ (cancelled) イベント [true または false] に一致する必要があります
receive_failed_(all_others)	Boolean	フィルタは receive_failed_ (all_others) イベント [true または false] に一致する必要があります
receive_failed_(out_of_space)	Boolean	フィルタは receive_failed_ (out_of_space) イベント [true または false] に一致する必要があります
receive_failed_ (package_not_upgraded)	Boolean	フィルタは receive_failed_ (package_not_upgraded) イベント [true または false] に一致する必要があります
receive_finished	Boolean	フィルタは receive_finished イベント [true または false] に一致する必要があります
receive_started	Boolean	フィルタは receive_started イベント [true または false] に一致する必要があります

表 21 警告アクション「restore」

プロパティ	型	説明
restore_finished	Boolean	フィルタは restore_finished イベント [true または false] に一致する必要があります
restore_started	Boolean	フィルタは restore_started イベント [true または false] に一致する必要があります

表 22 警告アクション「scrk」

プロパティ	型	説明
support_bundle_build_failed	Boolean	フィルタは support_bundle_build_failed イベント [true または false] に一致する必要があります
support_bundle_sent	Boolean	フィルタは support_bundle_sent イベント [true または false] に一致する必要があります
support_bundle_upload_failed	Boolean	フィルタは support_bundle_upload_failed イベント [true または false] に一致する必要があります
an_update_is_available_on_my_oracle_support.	Boolean	フィルタは an_update_is_available_on_my_oracle_support. イベント [true または false] に一致する必要があります
no_updates_available.	Boolean	フィルタは no_updates_available. イベント [true または false] に一致する必要があります
the_appliance_failed_to_verify_if_an_update_is_available.	Boolean	フィルタは the_appliance_failed_to_verify_if_an_update_is_available. イベント [true または false] に一致する必要があります

表 23 警告アクション「shadow」

プロパティ	型	説明
shadow_migration_complete	Boolean	フィルタは shadow_migration_complete イベント [true または false] に一致する必要があります

表 24 警告アクション「smf」

プロパティ	型	説明
service_failures	Boolean	フィルタは service_failures イベント [true または false] に一致する必要があります

表 25 警告アクション「thresholds」

プロパティ	型	説明
thresholdid	Default	警告が一致する必要があるウォッチの UUID です

表 26 警告アクション「zfs_pool」

プロパティ	型	説明
resilver_finished	Boolean	フィルタは resilver_finished イベント [true または false] に一致する必要があります
resilver_started	Boolean	フィルタは resilver_started イベント [true または false] に一致する必要があります
scrub_finished	Boolean	フィルタは scrub_finished イベント [true または false] に一致する必要があります
scrub_started	Boolean	フィルタは scrub_started イベント [true または false] に一致する必要があります
hot_spare_activated	Boolean	フィルタは hot_spare_activated イベント [true または false] に一致する必要があります

警告アクションの一覧表示

警告アクションの一覧表示コマンドは、すべての警告アクションを一覧表示します。単一のリソースに対するデータを取得するには、特定の警告アクションリソースの href プロパティに HTTP GET リクエストを送信します。

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 abcd123MWE=
Host: zfs-storage.example.com:215
Accept: application/json
```

レスポンスの例:

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

{
  "actions": [
    {
      "action": "actions-000",
      "category": "smf",
      "href": "/api/alert/v1/actions/actions-000",
      "service_failures": true
    },
    {
      "action": "actions-001",
      "category": "scrk",
      "href": "/api/alert/v1/actions/actions-001",
      "action-000": {
        "handler": "snmp_trap",

```

```

        "href": "/api/alert/v1/alerts/actions/actions-001
            /action-000"
    },
    "action-001": {
        "address": "Joe.Admin@acme.com",
        "handler": "email",
        "href": "/api/alert/v1/actions/actions-001
            /action-001",
        "subject": "Phone Home Alert"
    },
    "support_bundle_build_failed": true,
    "support_bundle_sent": true,
    "support_bundle_upload_failed": true
},
{
    "action": "actions-002",
    "category": "thresholds",
    "href": "/api/alert/v1/actions/actions-002",
    "action-000": {
        "address": "Joe.Admin@acme.com",
        "handler": "email",
        "href": "/api/alert/v1/actions/actions-002
            /action-000",
        "subject": "CPU Busy Alert"
    },
    "thresholdid": "b182ca05-53d3-6604-b874-ec353335704d"
}
]
}

```

警告アクションの取得

このコマンドは、警告アクションの一覧表示コマンドと似ていますが、指定された警告アクションのみを返します。

リクエストの例:

```
GET /api/alert/v1/actions/actions-002 HTTP/1.1
```

レスポンスの例:

```

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

{
  "action": {
    "category": "thresholds",
    "href": "/api/alert/v1/actions/actions-002",
    "action-000": {
      "address": "Joe.Admin@acme.com",
      "handler": "email",
      "href": "/api/alert/v1/alerts/actions/actions-002
          /action-000",
      "subject": "CPU Busy"
    },
    "thresholdid": "b182ca05-53d3-6604-b874-ec353335704d"
  }
}

```

```
}
```

警告アクションの作成

JSON オブジェクトを含む警告アクション POST リクエストを作成する場合、アクションプロパティを `/api/alert/v1/alerts/actions` に送信する必要があります。作成するアクションのタイプを選択するには、カテゴリプロパティを設定する必要があります。特定のシステムで使用可能なすべてのカテゴリ値については、CLI のドキュメントを参照してください。

カテゴリ値には通常次のものがあります。

```
"ad", "all", "appliance_software", "backup", "cluster", "custom",  
"hardware", "hardware_faults", "ndmp", "network", "replication",  
"replication_source", "replication_target", "restore", "scrk", "shadow",  
"smf", "thresholds" or "zfs_pool"
```

リクエストの例:

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

```
{"category": "hardware_faults"}
```

レスポンスの例:

```
HTTP/1.1 201 Created  
X-Zfssa-Appliance-API: 1.0  
Content-Type: application/json  
Content-Length: 118  
Location: /api/alert/v1/actions/actions-006
```

```
{  
  "action": {  
    "href": "/api/alert/v1/actions/actions-006",  
    "category": "hardware_faults",  
    "all_hardware_faults": true  
  }  
}
```

警告アクションの変更

`list` コマンドにより返された一部のプロパティは、HTTP PUT リクエストを送信することで変更できます。

リクエストの例:

```
PUT /api/alert/v1/actions/actions-001 HTTP/1.1
```

```
Host: zfssa.example.com:215
X-Auth-Session: uerqghq84vbvdv
Content-Type: application/json
Content-Length: 30
```

```
{"support_bundle_sent": false}
```

レスポンスの例:

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

```
{
  "action": {
    "href": "/api/alert/v1/actions/actions-001",
    "category": "scrk",
    "support_bundle_build_failed": true,
    "support_bundle_sent": false,
    "support_bundle_upload_failed": true
  }
}
```

警告アクションの削除

HTTP DELETE リクエストを、警告アクション href またはアクション href に送信すると、指定されたリソースを削除します。削除に成功すると、応答は HTTP ステータス 204 (No Content) になります。

リクエストの例:

```
DELETE /api/alert/v1/actions/actions-003 HTTP/1.1
Authorization: Basic abcd123MWE=
Host: zfssa.example.com:215
```

レスポンスの例:

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

警告アクション項目

各警告アクションリストに個別のアクション項目が追加されます。

警告項目の作成

既存のアクショングループに警告アクションを追加します。

リクエストの例:

```
POST /api/alert/v1/actions/actions-001 HTTP/1.1
Host: zfssa.example.com:215
X-Auth-Session: uerqghq84vbdv
Content-Type: application/json
Content-Length: 68

{"address": "Joe.Admin@acme.com", "handler": "email", "subject": "CPU Busy"}
```

レスポンスの例:

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 177
Location: /api/alert/v1/actions/actions-001/action-001

{
  "action": {
    "href": "/api/alert/v1/actions/actions-001/
             /action-001",
    "handler": "email",
    "address": "Joe.Admin@acme.com",
    "subject": "CPU Busy"
  }
}
```

警告アクションの変更

既存の警告アクションを変更します。

リクエストの例:

```
PUT /api/alert/v1/actions/actions-001/action-000 HTTP/1.1
Host: zfssa.example.com:215
X-Auth-Session: uerqghq84vbdv
Content-Type: application/json
Content-Length: 28

{"address": "Joseph.Admin@acme.com"}
```

レスポンスの例:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 176
X-Zfssa-Version: jkremer/generic@2013.06.08,1-0

{
  "action": {
    "href": "/api/alert/v1/actions/actions-001/
             /action-000",
    "handler": "email",
    "address": "Joseph.Admin@acme.com",
    "subject": "CPU Busy"
  }
}
```

警告アクション項目の削除

特定の警告アクションに対し、単一のアクションを削除できます。アクションを削除するには、アクションの href プロパティに DELETE リクエストを送信します。

アクションを削除するリクエストの例:

```
DELETE /api/alert/v1/actions/actions-001/action-000 HTTP/1.1
Host: zfssa.example.com:215
X-Auth-Session: uerqghq84vbvdv

HTTP/1.1 204 No Content
```


Analytics サービス

Analytics とは、さまざまな統計をリアルタイムでグラフ化したり、データを記録してあとで取得できるようにする機能です。これは長期のモニタリングと短期の分析の両方に提供されます。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。

使用可能な Analytics サービス

次の Analytics サービスを <http://zfssa.example.com/api/analytics/v1.0/> で利用できます

リクエスト	パス /analytics/v1	説明
GET		分析サービス情報を一覧表示します
POST	/worksheets	新しい分析データセットを作成します
GET	/worksheets/<worksheet>	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	/worksheets	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	/worksheets/<worksheet>	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	/worksheets/<worksheet>	指定されたワークシートオブジェクトを破棄します
PUT	/worksheets/<worksheet>/suspend	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	/worksheets/<worksheet>/resume	すべてのワークシートデータセットを再開します
POST	/worksheets/<worksheet>/datasets	新しいワークシートデータセットを作成します
GET	/worksheets/<worksheet>/datasets/<dataset>	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	/worksheets/<worksheet>/datasets	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/v1</code>	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できます。

名前	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)

設定の取得

分析設定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 ab6rt4psMWE=
Host: example.zfssa.com:215
Accept: application/json
```

結果の例:

```
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
  }
}
```

設定の変更

設定の変更コマンドは、データ保持値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 ab6rt4psMWE=
Host: example.zfssa.com:215
Content-Type: application/json
Content-Length: 60
```

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

結果の例:

```
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
  }
}
```

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。

名前	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 ab6rt4psMWE=
Host: example.zfssa.com:215
Accept: application/json
```

レスポンスの例:

```
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 ab6rt4psMWE=
Host: example.zfssa.com:215
Accept: application/json
```

レスポンスの例:

```
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 ab6rt4psMWE=
Content-Type: application/json
Content-Length: 26
```

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

結果の例:

```
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 ab6rt4psMWE=
Content-Type: application/json
Content-Length: 26
```

レスポンスの例:

```
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。データセット構成は、次のプロパティを使用します。

名前	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 ab6rt4psMWE=
Host: example.zfssa.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 ab6rt4psMWE=
Content-Type: application/json
Content-Length: 26
```

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

結果の例:

```
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 ab6rt4psMWE=
Content-Type: application/json
Content-Length: 26

{"seconds": 60}
```

結果の例:

```
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

表 27 データセットのプロパティ

名前	説明
name	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

名前	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
activity	保留中のデータセットアクティビティフラグ

suspended を除くすべてのプロパティは変更できません。

使用可能なデータセット:

- 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
- ndmp.diskkb
- nfs2.ops
- nfs2.ops[op]
- nfs3.ops
- nfs3.ops[op]
- nfs4.ops
- nfs4.ops[op]
- 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 ab6rt4psMWE=
Host: example.zfssa.com:215
Accept: application/json
```

結果の例:

```
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 ab6rt4psMWE=
Host: example.zfssa.com:215
Accept: application/json
```

結果の例:

```
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: letmein
Content-Type: application/json
Content-Length: 26
```

```
{"statistic": "test.sine"}
```

結果の例:

```
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 }
```

レスポンスの例:

```
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
```

レスポンスの例:

```
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
```

レスポンスの例:

```
HTTP/1.1 202 Accepted
```

データセットデータの除去

秒、分、または時間の粒度値で、データセット内のデータを除去できます。

名前	説明
granularity	除去の粒度 (秒、分、時間)
endtime	指定された終了時間の前に収集されたデータを除去します

endtime は、ISO 8601 の日時文字列です (例: 20130531T01:13:58)

リクエストの例:

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

レスポンスの例:

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

データセットデータの取得

分析データセットからデータを取得します。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。

表 28 パラメータの問合せ

名前	説明
start	サンプルデータの収集を開始する時間
seconds	サンプルデータを収集する秒数 (デフォルト = 1)
span	サンプルデータを収集する期間 (minute、hour、day、month)
interval	指定された期間内での間隔 (minute、hour、day、month)。この間隔からデータポイントの平均が与えられます。

start パラメータを指定しなかった場合、指定されたサンプルデータの seconds の数を現在時間から引いた値に、開始時間が設定されます。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。

粗粒度データを取得するには、次の表に示すように、パラメータ `span` と `interval` の組み合わせを使用します。これらのパラメータが使用された場合、`seconds` パラメータが設定されていたとしても無視されます。

表 29 span と interval の組み合わせ

Span	Interval					
	Minute	Hour	Day	Week	Month	Year
Minute	x					
Hour	x	x				
Day	x	x	x			
Week		x	x	x		
Month			x	x	x	
Year				x	x	x

どちらかのパラメータが不適切に入力された場合、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。

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。

表 30 データセットデータのプロパティ

名前	説明
<code>startTime</code>	返される最初のサンプルの時間
<code>sample</code>	返される最初のサンプルのサンプルインデックス
<code>data</code>	サンプルデータの配列
<code>min</code>	指定された間隔内での 1 秒あたりの最小値
<code>max</code>	指定された間隔内での 1 秒あたりの最大値

`startTime` プロパティは次のいずれかの形式になります。

- ISO 8601 の日時文字列 (例: 20130531T01:13:58)
- サンプルのインデックス番号
- 文字列リテラル「now」

リクエストの例 (3 秒分のライブデータを収集する):

```
GET /api/analytics/v1/datasets/nfs4.ops%5Bfile%5D/data?start=now&seconds=3
```

```
HTTP/1.1
Authorization: Basic ab6rt4psMWE=
Host: example.zfssa.com:215
Accept: text/x-yaml
```

結果の例:

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

```
---
data:
- sample: 239024557
  data:
    value: 5
    startTime: 20130912T21:42:38
    samples: 239024558
- sample: 239024558
  data:
    value: 15
    startTime: 20130912T21:42:39
    samples: 239024559
- sample: 239024559
  data:
    value: 25
    startTime: 20130912T21:42:40
    samples: 239024560
size: 3
---
```

リクエストの例 (1 週間の期間内に 7 日分のライブデータを収集する):

```
GET /api/analytics/v1/datasets/nfs4.ops%5Bfile%5D/data?
start=239024557&span=week&interval=day
HTTP/1.1
Authorization: Basic ab6rt4psMWE=
Host: example.zfssa.com:215
Accept: text/x-yaml
```

結果の例:

```
HTTP/1.1 200 OK
Content-Type: text/x-yaml
X-Zfssa-Analytics-API: 1.0
Transfer-Encoding: chunked
---
data:
- sample: 239024557
  data:
    value: 5
    max: 79
    min: 0
    startTime: 20130912T21:42:38
    samples: 240074328
- sample: 239110957
  data:
```

```
    value:      15
    max:        150
    min:         1
    startTime:  20130913T21:42:38
    samples:    240074328
...
- sample: 239629357
  data:
    value:      25
    max:        120
    min:         2
    startTime:  20130914T21:42:38
    samples:    240074328
size:          7
```


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

クラスタコマンドは、クラスタ化の設定およびクラスタ化されたリソースの管理に使用されます。

リクエスト	パス /hardware/v1	説明
GET	/cluster	クラスタのプロパティおよびクラスタリソースリストを取得します
GET	/cluster/resources /<resource:path>	指定されたクラスタリソースのプロパティを取得します
PUT	/cluster/resources /<resource:path>	指定されたクラスタリソースを変更します
PUT	/cluster/failback	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	/cluster/takeover	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	/cluster/unconfigure	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	/cluster/links	クラスタカードリンクステータスを取得します
PUT	/cluster/setup	クラスタの初期設定を実行します

クラスタのプロパティの取得

現在のクラスタ構成状態とリソースのプロパティを取得します。

リクエストの例:

```
GET /api/hardware/v1/cluster HTTP/1.1
Authorization: Basic abcd45sMWE=
Host: tanana:215
Accept: application/json
```

レスポンスの例:

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

```
{
  "cluster": {
    "description": "Clustering is not configured",
    "peer_asn": "",
    "peer_description": "",
    "peer_hostname": "",
    "peer_state": "",
    "resources": {
      "net/ixgbe0": {
        "details": ["10.80.231.58"],
        "href": "/hardware/v1/cluster/resources/resources/net/ixgbe0",
        "owner": "tanana",
        "type": "singleton",
        "user_label": "Untitled Interface"
      },
      "zfs/gold": {
        "details": ["821G"],
        "href": "/hardware/v1/cluster/resources/resources/zfs/gold",
        "owner": "tanana",
        "type": "singleton",
        "user_label": ""
      }
    }
  },
  "state": "AKCS_UNCONFIGURED"
}
```

クラスタリソースの取得

クラスタリソースからの href プロパティに従うことで、その単一クラスタリソースのデータを取得できます。前の例では、`/hardware/v1/cluster/resources/resources/zfs/gold` および `/hardware/v1/cluster/resources/resources/net/ixgbe0` という 2 つのリソースが使用可能です。

クラスタリソースの変更

システムがクラスタ化されている場合、このコマンドを使用して各クラスタリソースのプロパティを変更できます。詳細については、CLI 「`configuration cluster resources`」を参照してください。

クラスタのコマンド

クラスタでサポートされるコマンドは、`failover`、`takeback`、および `unconfigure` です。すべてのコマンドは、クラスタリソースにコマンド名を付加したものに対する PUT リクエストを使用します。成功すると、どちらのコマンドも HTTP ステータス 202 (Accepted) を返します。

リクエストの例:

```
PUT /api/hardware/v1/cluster/failback HTTP/1.1
Authorization: Basic abcd123MWE=
Host: zfssa.example.com:215
```

結果例:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
```

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。

クラスタリンク

このコマンドは、クラスタカードの現在のリンクステータスを返します。出力は、`aksh` コマンド「`configuration cluster links`」と同じです。クラスタの配線に問題がないことを保証するために、このコマンドはクラスタ設定を実行する前に実行することをお勧めします。設定を実行する前に、すべてのリンクが `AKCIOS_ACTIVE` 状態である必要があります。

リクエストの例:

```
GET /api/hardware/v1/cluster/links HTTP/1.1
Authorization: Basic abcd123MWE=
Host: zfssa.example.com:215
Accept: application/json
```

レスポンスの例:

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

{
  "links": {
    "clustron2_embedded:0/clustron_uart:0 = AKCIOS_TIMEDOUT\n
    clustron2_embedded:0/clustron_uart:1 = AKCIOS_TIMEDOU\n
    clustron2_embedded:0/dlpi:0 = AKCIOS_TIMEDOUT"
  }
}
```

クラスタの設定

setup cluster コマンドは、システムの初期クラスタ化を設定します。すべてのクラスタリンクが AKCIOS_ACTIVE 状態である必要があり、ピアシステムに電源が投入されていても構成されていない必要があります。そうでないと、コマンドは失敗します。

リクエストの例:

```
PUT /api/hardware/v1/cluster/setup HTTP/1.1
Authorization: Basic abcd123MWE=
Host: zfssa.example.com:215
Accept: application/json

{"nodename": "zfssa-storage-2", "password": "letmein"}
```

結果例:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
```

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。

リクエスト	パス /hardware/v1.0	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。

型	プロパティ	説明
string	name	シャーシ名
string	model	シャーシのモデル番号
string	manufacturer	シャーシの製造元
string	serial	シャーシのシリアル番号
string	revision	シャーシのリビジョンレベル
string	part	シャーシの交換パーツ番号
boolean	faulted	障害インジケータ
string	fru	シャーシの FMRI 表現
string	uuid	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
```

レスポンスの例:

```
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 ZS3-1",
    "name": "cairo",
    "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/nas/v1/chassis/chassis-001 HTTP/1.1
Host: zfs-storage.example.com
Accept: application/json
```

レスポンスの例:

```
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": "c0t500CCA01A76A2B8d0",
      "faulted": false,
      "href": "/api/hardware/v1/chassis/chassis-001/disk/disk-000",
      "interface": "SAS",
      "label": "HDD 0",
      "locate": false,
      "manufacturer": "HITACHI",
      "model": "H7230AS60SUN3.0T",
      "pathcount": 4,
      "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、次の表に含まれるコンポーネントのプロパティが含まれます。

型	名前	説明
string	device	FRU デバイス ID
boolean	faulted	FRU に障害が発生しているかどうかを示すフラグ
string	fru	FRU の FMRI 表現
string	interface	FRU のインタフェースタイプ
string	label	FRU の場所ラベル
boolean	locate	フラグの位置特定インジケータ
string	manufacturer	FRU の製造元
string	model	FRU モデル
string	part	FRU パート番号
boolean	present	FRU プレゼンスインジケータ
number	rpm	プラッタの RPM (ディスクのみ)

型	名前	説明
string	serial	FRU シリアル番号
number	size	FRU サイズ (容量)
string	type	コンポーネントタイプ
string	use	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
Accept: application/json
```

レスポンスの例:

```
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,
    "manufacturer": "HITACHI",
    "model": "H7230AS60SUN3.0T",
    "pathcount": 4,
    "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: zfsa.example.com:215
X-Auth-User: root
X-Auth-Key: letmein
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。

ログのコマンド

リクエスト	パス /api/log/v1	説明
GET		ログサービスのコマンドを一覧表示します
GET	/logs	すべてのログタイプを一覧表示します
GET	/logs/<log>	選択したログのログエントリを一覧表示します
GET	/collect/<log>	指定したログエントリのコレクションをダウンロードします
GET	/collect	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

ログの一覧表示

アプライアンスで使用可能な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。

リクエストの例:

```
GET /api/log/v1/logs HTTP/1.1
Authorization: Basic ab6rt4psMWE=
Host: example.zfssa.com:215
Accept: application/json
```

結果の例:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
```

ログエントリの取得

```
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、後述する収集関数を使用してダウンロードする必要があります。

プロパティ	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の数を制限します

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 - 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 abcd45sMWE=
Host: tanana: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": "10.159.84.34",
      "annotation": "",
      "summary": "User logged in",
      "timestamp": "20131022T22:54:19",
      "user": "root"
    }, {
      "address": "10.159.84.34",
      "annotation": "",
      "summary": "Destroyed share \"gold: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": "10.159.84.34",
      "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、その名前を収集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)、マルチパス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 IP マルチパス (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 31 ネットワークデータリンクのコマンド

リクエスト	パス /network/v1	説明
POST	/datalinks	新しいネットワークデータリンクを作成します
GET	/datalinks/<datalink>	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	/datalinks	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	/datalinks/<datalink>	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	/datalinks/<datalink>	指定されたデータリンクオブジェクトを破棄します

表 32 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
class	String	「device」(「変更不可」)
label	NetworkLabel	Label
links	ChooseOne	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)
speed	ChooseOne	リンク速度 [「auto」、「10」、「100」、「1000」、「10000」]
duplex	ChooseOne	リンク二重 [「auto」、「half」、「full」]

表 33 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
class	String	「vnic」(「変更不可」)
label	NetworkLabel	Label
links	ChooseOne	リンク [「ixgbe0」]

プロパティ	型	説明
mtu	PositiveInteger	最大転送単位 (MTU)
id	VLAN	VLAN ID

表 34 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 35 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 36 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
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
```

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: zfssa.example.com:215
X-Auth-User: root
X-Auth-Key: letmein
Content-Type: application/json
Content-Length: 78

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

```

レスポンスの例:

```

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}
```

レスポンスの例:

```
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
```

レスポンスの例:

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

ネットワークデバイス

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 37 ネットワークのコマンド

リクエスト	パス /network/v1	説明
GET	/devices/<device>	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	/devices	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 38 ネットワークデバイスのプロパティ

名前	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
media	デバイスのメディア
speed	デバイス速度 (M ビット/秒)
up	デバイスが操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

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 ab6rt4psMWE=
Host: example.zfssa.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

単一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 ab6rt4psMWE=
Host: example.zfssa.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 39 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 <i>/api/network/v1</i>	説明
POST	<i>/interfaces</i>	新しいネットワークインタフェースを作成します
GET	<i>/interfaces/<interface></i>	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	<i>/interfaces</i>	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	<i>/interfaces/<interface></i>	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	<i>/interfaces/<interface></i>	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 40 ネットワークインタフェースのプロパティ

名前	説明
admin	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
class	クラスタイプ (「ip」、 「ipmp」) (作成後は変更不可)

名前	説明
curaddr	現在の IP アドレス (変更不可)
enable	このインタフェースが有効かどうかを示すフラグ
label	インタフェースのユーザーラベル
links	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
state	インタフェースの状態 (変更不可)
v4addr	IPv4 アドレス
v6dhcp	IPv4 DHCP フラグ
v6addr	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

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 abcd1234MWE=
Host: zfssa.example.com:215
Accept: application/json
```

レスポンスの例:

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

{
  "interfaces": [
    {
      "href": "/api/network/v1/interfaces/ixgbe0",
      "v4addr": ["10.80.231.58/24"]
      ...
    },
    {
      "href": "/api/network/v1/interfaces/ixgbe1",
      "v4addr": ["10.80.231.59/24"]
      ...
    },
    {
      "href": "/api/network/v1/interfaces/ixgbe2",
      "v4addr": ["10.80.231.60/24"]
      ...
    },
    {
      "href": "/api/network/v1/interfaces/ixgbe3",
      "v4addr": ["10.80.231.61/24"]
      ...
    }
  ]
}
```

ネットワークインタフェースの取得

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 abcd1234MWE=
Host: zfssa.example.com:215
Accept: application/json
```

レスポンスの例:

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

{
  "interface": {
    "admin": true,
    "class": "ip",
    "curaddrs": ["10.80.231.58/24"],
    "enable": true,
    "href": "/api/network/v1/interfaces/ixgbe0",
    "interface": "ixgbe0",
    "label": "Untitled Interface",
    "links": ["ixgbe0"],
    "state": "up",
    "v4addrs": ["10.80.231.58/24"],
    "v4dhcp": false,
    "v6addrs": [],
    "v6dhcp": false
  }
}
```

ネットワークインタフェースの作成

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: zfssa.example.com:215
X-Auth-User: root
X-Auth-Key: letmein
Content-Type: application/json
Content-Length: 78
```

```
{
  "class": "ip",
  "links": ["ixgbe3"],
  "v4addrs": "192.168.1.9/24"
}
```

レスポンスの例:

```
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

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": ["192.168.1.99/24"],
  "interface": "Demo Rest"
}
```

レスポンスの例:

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

{
  "admin": true,
  "class": "ip",
  "curaddrs": ["192.168.1.9/24"],
  "enable": true,
  "href": "/api/network/v1/interfaces/ixgbe3",
  "interface": "ixgbe3",
  "label": "Demo Rest",
  "links": ["ixgbe3"],
  "state": "failed",
  "v4addrs": ["192.168.1.99/24"]
  "v4dhcp": false,
  "v6addrs": [],
  "v6dhcp": false
}
```

ネットワークインタフェースの削除

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 abcd1234MWE=
Host: zfssa.example.com:215
```

結果例:

ネットワークルート

ネットワークルートを管理します。

表 41 ネットワークルートの管理

リクエスト	パス /api/network/v1	説明
POST	/routes	新しいネットワークルートを作成します
GET	/routes/<route>	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	/routes	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	/routes/<route>	指定されたルートオブジェクトを破棄します
GET	/routing	ネットルーティングプロパティを取得します
PUT	/routing	ネットルーティングプロパティを変更します

表 42 ネットワークルートのプロパティ

名前	説明
type	ルートのタイプ (「system」、「static」など (変更不可))
family	アドレスファミリー (IPv4 または IPv6)
destination	ルーティング先アドレス
gateway	ゲートウェイアドレス
interface	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 abcd1234MWE=
Host: zfssa.example.com:215
Accept: application/json
```

結果例:

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

{
  "routes": [{
    "destination": "0.0.0.0",
    "family": "IPv4",
    "gateway": "10.80.231.1",
    "href": "/api/network/v1/routing/route-000",
    "interface": "ixgbe0",
    "mask": 0,
    "route": "route-000",
    "type": "static"
  }, {
    "destination": "10.80.231.0",
    "family": "IPv4",
    "gateway": "10.80.231.58",
    "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=10.80.231.0 HTTP/1.1
Authorization: Basic abcd1234MWE=
Host: zfssa.example.com:215
Accept: application/json
```

結果例:

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

{
  "route": {
    "destination": "10.80.231.0",
    "family": "IPv4",
    "gateway": "10.80.231.58",
    "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 abcd1234MWE=  
Host: zfssa.example.com:215  
Content-Type: application/json  
Content-Length: 164
```

```
{  
  "family": "IPv4",  
  "destination": "0.0.0.0",  
  "mask": "0",  
  "gateway": "10.11.12.1",  
  "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 abcd1234MWE=  
Host: zfssa.example.com:215
```

結果例:

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

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 43 問題サービスコマンド

リクエスト	パス <i>/problem/v1</i>	説明
GET		問題サービスコマンドを一覧表示します
GET	<i>/problems</i>	すべての現在の問題を一覧表示します
GET	<i>/problems/<problem></i>	指定された uuid を持つ問題の詳細プロパティを取得します
PUT	<i>/problems/<problem> /markrepaired</i>	指定された問題 uuid を修復済みとしてマークします

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。成功したコマンドに対し、200 の HTTP ステータス (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
Accept: application/json
```

レスポンスの例:

```
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。成功したコマンドに対し、200 の HTTP ステータス (OK) が返されます。

URI 入力パラメータ uuid – 単一の問題の UUID。

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
Accept: application/json

```

レスポンスの例:

```

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

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

```

問題の修復

問題は問題の修復コマンドで修復済みとしてマークできます。

入力パラメータ uuid – 修復済みとしてマークされる問題の UUID。

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
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
```


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権限のコレクションです。さまざまな承認レベルを持つ管理者およびオペレータのロールを作成することが望ましい場合があります。スタッフメンバーには、不要な権限を割り当てずに、必要に応じて適切なロールを割り当てることができます。共有の管理者パスワードを使用する(たとえば、root パスワードをすべてのユーザーに割り当てる)よりも、ロールを使用した方がセキュアです。ロールは、ユーザーを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個々のユーザー名に結び付けます。デフォルトで、きわめて基本的な承認を含む「基本管理」というロールが存在します。

システムのロールと承認を管理するには、RESTful API ロールサービスを使用します。

ロールサービスコマンドの概要

次のリストにロールコマンドを示します。

表 44 ロールサービスコマンド

リクエスト	パス /role/v1	説明
GET		ロールサービスコマンドを一覧表示します
GET	/roles/<role>	指定された管理ロールプロパティを取得します
GET	/roles	すべての管理ロールオブジェクトを一覧表示します
PUT	/roles/<role>	指定された管理ロールオブジェクトを変更します
DELETE	/roles/<role>	指定されたロールオブジェクトを破棄します
POST	/roles	新しいロールを作成するか、既存のロールのクローンを作成します
PUT	/roles/<role>/revoke	すべてのユーザーから指定されたロールを削除します
POST	/roles/<role>/authorizations	新しいロールの承認を作成します

リクエスト	パス /role/v1	説明
GET	/roles/<role>/authorizations /<auth>	指定されたロールの承認プロパティを取得します
GET	/roles/<role>/authorizations	すべてのロール承認オブジェクトを一覧表示します
PUT	/roles/<role>/authorizations /<auth>	指定されたロール承認オブジェクトを変更します
DELETE	/roles/<role>/authorizations /<auth>	指定された承認オブジェクトを破棄します

List Roles

各ロールには次のサマリープロパティがあります。ロールのプロパティの詳細な説明については、CLI のヘルプを参照してください。

表 45 ロールのプロパティ

型	プロパティ名	説明
string	name	ロール名 (作成後は変更不可)
string	説明	ロールの説明

リクエストの例:

```
GET /api/role/v1/roles HTTP/1.1
Authorization: Basic abcdefgMWE=
Host: zfssa.example.com:215
Accept: application/json
```

レスポンスの例:

```
{
  "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"
  }
]}
```

Get Role

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、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 abcefgMWE=
Host: zfssa.example.com:215
Accept: application/json
```

レスポンスの例:

```
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"
  }
}
```

Create Role

このコマンドは新しいロールを作成します。

表 46 新しいロールの作成のプロパティ

型	プロパティ名	説明
string	name	新しいロールの名前 (必須)
string	clone	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)
string	説明	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 abcdefgMWE=
Host: zfssa.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"
  }
}
```

Modify Role

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 abcdefgMWE=
Host: zfssa.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"
  }
}
```

Revoke Role

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 abcefgMWE=
Host: zfssa.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
```

Delete Role

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、問合せパラメータ「confirm」を「true」に設定する必要があります。

リクエストの例:

```
DELETE /api/role/v1/roles/rola?confirm=true HTTP/1.1
Authorization: Basic abcefgMWE=
Host: zfssa.example.com:215
Accept: */*
```

結果例:

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

List Role Authorizations

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 abcefgMWE=
Host: zfssa.example.com:215
Accept: application/json
```

レスポンスの例:

```
{
```

```
    "authorizations": [{  
      "allow_modify": false,  
      "allow_read": true,  
      "auth": "auth-000",  
      "href": "/api/role/v1/roles/role_workflow/authorizations/auth-000",  
      "owner": "*",  
      "scope": "workflow",  
      "uuid": "*"   
    }  
  ]  
}
```

Create Role Authorization

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 abcdefgMWE=  
Host: zfssa.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": "*"   
  }  
}
```

Modify Role Authorization

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 abcdefgMWE=
Host: zfssa.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": "**"
  }
}
```

Delete Role Authorization

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 abcdefgMWE=
Host: zfssa.example.com:215
Accept: */*
```

結果例:

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


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、何らかの種類の切り換えが含まれます。RESTful API SAN サービスは、サポートされ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することが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。

表 47 イニシエータコマンド

リクエスト	パス /san/v1.0	説明
GET	/<protocol>/initiators	指定されたプロトコル (FC、iSCSI、SRP) 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	/<protocol>/initiators /<initiator>	指定されたプロトコル (FC、iSCSI、SRP) 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	/<protocol>/initiators	指定されたプロトコル (FC、iSCSI、SRP) の新しい SAN イニシエータを作成します
PUT	/<protocol>/initiators /<initiator>	指定されたプロトコル (FC、iSCSI、SRP) 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	/<protocol>/initiators /<initiator>	指定されたイニシエータオブジェクトを破棄します

これらのコマンドは次の URI パラメータを使用します。

表 48 URI パラメータ

名前	説明
protocol	イニシエータ (FC、iSCSI、SRP) の NAS プロトコル
initiator	イニシエータの iqn、wwn、または eui

イニシエータコマンドの多くは戻り値として次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 49 イニシエータのプロパティ

名前	プロトコル	説明
alias	all	このイニシエータの別名
initiator	fc	このイニシエータのポート World Wide Name (WWN)
iqn	iscsi	このイニシエータの iSCSI 修飾名
chapuser	iscsi	チャレンジハンドシェイク認証プロトコル (CHAP) ユーザー名

名前	プロトコル	説明
chapsecret	iscsi	チャレンジハンドシェイク認証プロトコル (CHAP) シークレット
initiator	srp	EUI (Extended Unique Identifier)

List Initiators

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
Accept: application/json
```

レスポンスの例:

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

```
{
  "initiators": [{
    "alias": "init-02",
    "href": "/api/san/v1/iscsi/initiators/iqn.1986-03.com.sun:02:02",
    "initiator": "iqn.1986-03.com.sun:02:02",
    "chapsecret": "",
    "chapuser": ""
  }, {
    "alias": "init-01",
    "initiator": "iqn.1986-03.com.sun:02:01",
    "href": "/api/san/v1/iscsi/initiators/iqn.1986-03.com.sun:02:01",
    "chapsecret": "",
    "chapuser": ""
  }
]}
```

Get Initiator Details

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.1986-03.com.sun:02:01 HTTP/1.1
Host: zfs-storage.example.com
Accept: application/json
```

レスポンスの例:

```
HTTP/1.1 200 OK
```

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

```
{
  "initiator": {
    "alias": "init-01",
    "href": "/api/san/v1/iscsi/initiators/iqn.1986-03.com.sun:02:01"
    "initiator": "iqn.1986-03.com.sun:02:01",
    "chapsecret": "",
    "chapuser": ""
  }
}
```

Create an Initiator

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

```
{
  "initiator": "iqn.1986-03.com.sun:02:02",
  "alias": "init-02"
}
```

レスポンスの例:

```
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.1986-03.com.sun:02:02
```

```
{
  "initiator": {
    "alias": "init-02",
    "href": "/api/san/v1/iscsi/initiators/iqn.1986-03.com.sun:02:02",
    "initiator": "iqn.1986-03.com.sun:02:02",
    "chapsecret": "",
    "chapuser": ""
  }
}
```

Modify an Initiator

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が含まれます。

リクエストの例:

```
PUT /api/san/v1/iscsi/initiators/iqn.1986-03.com.sun:01 /HTTP/1.1
Host: zfs-storage.example.com
Content-Type: application/json
Accept: application/json
```

```
{
  "alias": "init-01-secure",
  "chapuser": "chuck",
  "chapsecret": "igottheintersect"
}
```

レスポンスの例:

```
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.1986-03.com.sun:01",
    "iqn": "iqn.1986-03.com.sun:1",
    "chapsecret": "igottheintersect",
    "chapuser": "chuck"
  }
}
```

Delete an Initiator

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.1986-03.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次の表に示します。

表 50 イニシエータグループコマンド

リクエスト	パス /san/v1.0	説明
GET	/<protocol>/initiator-groups	指定されたプロトコル (FC、iSCSI、SRP) 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	/<protocol>/initiator-groups /<name>	指定されたプロトコル (FC、iSCSI、SRP) 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	/<protocol>/initiator-groups	指定されたプロトコル (FC、iSCSI、SRP) の新しい SAN イニシエータグループを作成します
PUT	/<protocol>/initiator-groups /<name>	指定されたプロトコル (FC、iSCSI、SRP) 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	/<protocol>/initiator-groups /<name>	指定された名前オブジェクトを破棄します

これらのコマンドは次の URI パラメータを使用します。

表 51 URI パラメータ

名前	説明
protocol	イニシエータ (FC、iSCSI、SRP) の NAS プロトコル
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イニシエータグループのイニシエータのリストを含む「name」プロパティと「initiators」プロパティがあります。

List Initiator Groups

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
Accept: application/json
```

レスポンスの例:

```

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

{
  "groups": [{
    "href": "/san/v1/iscsi/initiator-groups/aktest-initiators-0",
    "initiators": ["iqn.1986-03.com.sun:0"],
    "name": "aktest-initiators-0"
  }, {
    "href": "/san/v1/iscsi/initiator-groups/aktest-initiators-1",
    "initiators": ["iqn.1986-03.com.sun:1"],
    "name": "aktest-initiators-1"
  }]
}

```

Get Initiator Group Details

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
Accept: application/json

```

レスポンスの例:

```

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

{
  "group": {
    "href": "/api/san/v1/iscsi/initiator-groups/test-group"
    "initiators": ["iqn.1986-03.com.sun:02:01"],
    "name": "test-group"
  }
}

```

Create an Initiator Group

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 52 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
Content-Type: application/json
Content-Length: 64
Accept: application/json
```

```
{
  "name": "group-01",
  "initiators": ["iqn.1986-03.com.sun:02"]
}
```

レスポンスの例:

```
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.1986-03.com.sun:02"],
    "name": "group-01"
  }
}
```

Delete an Initiator Group

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
```

delete が成功すると、HTTP ステータス 204 (No Content) を返します。

```
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下に示します。

表 53 ターゲットコマンド

リクエスト	パス / san/v1.0	説明
GET	/ <i><protocol></i> /targets	指定されたプロトコル (FC、iSCSI、SRP) 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

リクエスト	パス /san/v1.0	説明
GET	/<protocol>/targets/<target>	指定されたプロトコル (FC、iSCSI、SRP) プロパティの指定された SAN ターゲットを取得します
POST	/<protocol>/targets	指定されたプロトコル (FC、iSCSI、SRP) の新しい SAN ターゲットを作成します
PUT	/<protocol>/targets/<target>	指定されたプロトコル (FC、iSCSI、SRP) オブジェクトの指定された SAN ターゲットを変更します
DELETE	/<protocol>/targets/<target>	指定されたターゲットオブジェクトを破棄します

コマンドは次の URI パラメータを取ります

表 54 URI パラメータ

型	名前	説明
string	protocol	SAN プロトコル (FC、iSCSI、SRP)
string	target	ターゲット ID (IQN、WWN、EUI)

すべての「get」 target コマンドは target プロパティを返し、「create」および「modify」 target コマンドは入力として次のプロパティを使用します。

表 55 ターゲットのプロパティ

名前	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

名前	プロトコル	説明
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 56 ターゲットグループのプロパティ

型	名前	説明
string	protocol	ターゲットグループプロトコル (FC、iSCSI、SRP)
string	name	iSCSI ターゲットグループ名
配列	targets	iSCSI ターゲット IQN グループメンバーのリスト

List Targets

アプライアンスで使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 abcd123MWE=
Accept: application/json
```

レスポンスの例:

```
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.1986-03.com.sun:02:72b6fa9a-96c4-e511-db19-aadb9bac2052",
    "iqn": "iqn.1986-03.com.sun:02:72b6fa9a-96c4-e511-db19-aadb9bac2052",
```

```

    }, {
      ...
      "alias": "tst.volumes.py.96238.target",
      "href": "/api/san/v1/iscsi/targets/iqn.1986-03.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008",
      "iqn": "iqn.1986-03.com.sun:31d26d2e-6aa0-6054-fe58-8b1fb508b008",
      ...
    }
    ...]
  }
}

```

Get Target Details

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 abcd123MWE=
Accept: application/json

```

レスポンスの例:

```

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.1986-03.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008",
    "targetchapsecret": "",
    "targetchapuser": ""
  }
}

```

Create a Target

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 abcd123MWE=
Content-Type: application/json
Content-Length: 23

```

```
Accept: application/json
```

```
{"alias": "test-target"}
```

レスポンスの例:

```
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.1986-03.com.sun:02:31d26d2e-6aa0-6054-
fe58-8b1fb508b008
```

```
{
  "target": {
    "href": "/api/san/v1/iscsi/targets/iqn.1986-03.com.sun:02:31d26d2e-6aa0-6054-
fe58-8b1fb508b008",
    "alias": "test-target",
    "iqn": "iqn.1986-03.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008",
    "auth": "none",
    "targetchapuser": "",
    "targetchapsecret": "",
    "interfaces": ["ixgbe0"]
  }
}
```

Modify a Target

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 abcd123MWE=
Host: zfs-storage.example.com
Content-Type: application/json
Content-Length: 54
Accept: application/json
```

```
{"targetchapsecret": "letmeinnowplease", "auth": "chap",
 "targetchapuser": "argus"}
```

レスポンスの例:

```
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": "letmeinnowplease",
```

```

    "alias": "test-arget",
    "iqn": "iqn.1986-03.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008",
    "targetchapuser": "argus",
    "interfaces": ["ixgbe0"]
  }
}

```

Delete a Target

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.1986-03.com.sun:02:e7e688b1 HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic abcd123MWE=

```

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。

表 57 ターゲットグループのコマンド

リクエスト	パス /san/v1.0	説明
GET	<code>/<protocol>/target-groups</code>	指定されたプロトコル (FC、iSCSI、SRP) 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 (FC、iSCSI、SRP) 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 (FC、iSCSI、SRP) 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 (FC、iSCSI、SRP) 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 target-group オブジェクトを破棄します

これらのコマンドは次の URI パラメータを使用します。

表 58 URI パラメータ

名前	説明
protocol	イニシエータ (FC、iSCSI、SRP) の NAS プロトコル
name	ターゲットグループの名前

List Target Groups

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 abcd123MWE=
Accept: application/json
```

レスポンスの例:

```
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.1986-03.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008"
    ]
  }, {
    "href": "/api/san/v1/iscsi/target-groups/alt-group",
    ...
  }]
}
```

Get Target Group

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「target-group」という名前の 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/test-group
Host: zfs-storage.example.com:215
```

```
Authorization: Basic abcd123MWE=  
Accept: application/json
```

レスポンスの例:

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

```
{  
  "group": {  
    "href": "/api/san/v1/iscsi/target-groups/test-group",  
    "name": "test-group",  
    "targets": [  
      "iqn.1986-03.com.sun:02:0d5a0ed8-44b6-49f8-a594-872bf787ca5a"]  
    ]  
  }  
}
```

Create a Target Group

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 abcd123MWE  
Accept: application/json  
Content-Type: application/json  
Content-Length: 97
```

```
{"name": "test-group",  
 "targets": ["iqn.1986-03.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008"]}
```

レスポンスの例:

```
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.1986-03.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008"]  
    ]  
  }  
}
```

Delete a Target Group

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/nas/v1.0/iscsi/target-groups/test-group
```

`delete` が成功すると、HTTP ステータス 204 (No Content) を返します。

```
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 59 サービスコマンド

リクエスト	パス /service/v1	説明
GET		サービスコマンドを一覧表示します
GET	/services	すべてのサービスを一覧表示します
GET	/services/<service>	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	/services/<service>	指定されたサービスの構成を変更します
PUT	/services/<service>/enable	指定されたサービスを有効にします
PUT	/services/<service>/disable	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
```

Accept: application/json

レスポンスの例:

```
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"
    },
    {
      "<status>": "online",
      "href": "/api/service/v1/services/smb",
      "log": {
        "href": "/api/log/v1/logs/network-smb:default",
        "size": 2
      },
      "name": "smb"
    },
    {
      "<status>": "online",
      "href": "/api/service/v1/services/dns",
      "log": {
        "href": "/api/log/v1/logs/network-dns-client:default",
        "size": 4
      },
      "name": "dns"
    },
    {
      "<status>": "online",
      "href": "/api/service/v1/services/dynrouting",
      "log": {
        "href": "/api/log/v1/logs/network-routing-route:default",
        "size": 81
      },
      "name": "dynrouting"
    },
    {
      "<status>": "disabled",
      "href": "/api/service/v1/services/ftp",
      "log": {
        "href": "/api/log/v1/logs/network-ftp:proftpd",
        "size": 40
      },
      "name": "ftp"
    },
    {
      "<status>": "disabled",
      "href": "/api/service/v1/services/http",
      "name": "http"
    },
    {
      "<status>": "online",
      "href": "/api/service/v1/services/identity",
      "log": {
        "href": "/api/log/v1/logs/system-identity:node",
        "size": 4
      },
      "name": "identity"
    },
    {
      "<status>": "online",
      "href": "/api/service/v1/services/idmap",
      "log": {
        "href": "/api/log/v1/logs/system-idmap:default",
        "size": 15
      }
    }
  ]
}
```

```
    },
    "name": "idmap"
  }, {
    "<status>": "online",
    "href": "/api/service/v1/services/ipmp",
    "log": {
      "href": "/api/log/v1/logs/network-ipmp:default",
      "size": 3
    },
    "name": "ipmp"
  }, {
    "<status>": "online",
    "href": "/api/service/v1/services/iscsi",
    "log": {
      "href": "/api/log/v1/logs/network-iscsi-target:default",
      "size": 3
    },
    "name": "iscsi"
  }, {
    "<status>": "disabled",
    "href": "/api/service/v1/services/ldap",
    "name": "ldap"
  }, {
    "<status>": "online",
    "href": "/api/service/v1/services/ndmp",
    "log": {
      "href": "/api/log/v1/logs/system-ndmpd:default",
      "size": 11
    },
    "name": "ndmp"
  }, {
    "<status>": "online",
    "href": "/api/service/v1/services/nfs",
    "log": {
      "href": "/api/log/v1/logs/appliance-kit-nfsconf:default",
      "size": 6
    },
    "name": "nfs"
  }, {
    "<status>": "disabled",
    "href": "/api/service/v1/services/nis",
    "log": {
      "href": "/api/log/v1/logs/network-nis-domain:default",
      "size": 3
    },
    "name": "nis"
  }, {
    "<status>": "disabled",
    "href": "/api/service/v1/services/ntp",
    "name": "ntp"
  }, {
    "<status>": "online",
    "href": "/api/service/v1/services/replication",
    "name": "replication"
  }, {
    "<status>": "online",
    "href": "/api/service/v1/services/rest",
    "log": {
      "href": "/api/log/v1/logs/appliance-kit-akrestd:default",
      "size": 10
    },
    "name": "rest"
  }, {
    "<status>": "disabled",
```

```
    "href": "/api/service/v1/services/scrk",
    "name": "scrk"
  }, {
    "<status>": "disabled",
    "href": "/api/service/v1/services/sftp",
    "name": "sftp"
  }, {
    "<status>": "online",
    "href": "/api/service/v1/services/shadow",
    "name": "shadow"
  }, {
    "<status>": "online",
    "href": "/api/service/v1/services/smtp",
    "log": {
      "href": "/api/log/v1/logs/network-smtp:sendmail",
      "size": 6
    },
    "name": "smtp"
  }, {
    "<status>": "disabled",
    "href": "/api/service/v1/services/snmp",
    "name": "snmp"
  }, {
    "<status>": "disabled",
    "href": "/api/service/v1/services/srp",
    "name": "srp"
  }, {
    "<status>": "online",
    "href": "/api/service/v1/services/ssh",
    "log": {
      "href": "/api/log/v1/logs/network-ssh:default",
      "size": 3
    },
    "name": "ssh"
  }, {
    "<status>": "disabled",
    "href": "/api/service/v1/services/syslog",
    "name": "syslog"
  }, {
    "<status>": "online",
    "href": "/api/service/v1/services/tags",
    "name": "tags"
  }, {
    "<status>": "disabled",
    "href": "/api/service/v1/services/tftp",
    "name": "tftp"
  }, {
    "<status>": "disabled",
    "href": "/api/service/v1/services/vscan",
    "log": {
      "href": "/api/log/v1/logs/vscan",
      "size": 0
    },
    "name": "vscan"
  }
}]
}
```

Get Service

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
Accept: application/json
```

レスポンスの例:

```
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
  }
}
```

Change Service State

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。

- サービス - サービスの名前
- 状態 - 新しいサービスの状態 (「enable」、 「disable」)

URI パラメータを使用したリクエストの例:

```
PUT /api/service/v1/services/replication/enable HTTP/1.1
Host: zfs-storage.example.com
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
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。

リクエストの例:

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

```
{"port": 218}
```

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
Content-Length: 162
Content-Type: application/json; charset=utf-8
X-Zfssa-Service-API: 1.0
```

```
{
  "service": {
    "<status>": "disabled",
    "href": "/api/service/v1/services/sftp",
    "keys": [],
    "listen_port": 218,
    "logging_verbosity": "INFO",
    "root_login": false
  }
}
```

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 60 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	/services/<service>/<resource>	サービスのサブリソースを一覧表示します

リクエスト	パス	説明
PUT	/services/<service>/<resource>/<href>	サブリソースを変更します
POST	/services/<service>/<resource>	新しいサブリソースを作成します
DELETE	/services/<service>/<resource>/<href>	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共有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ます。このバージョンの NAS API では、アプライアンスストレージ構成の取得にプール操作が使用されます。

表 61 ストレージプールコマンド

リクエスト	パス <i>/api/storage/v1</i>	説明
GET	<i>/pools</i>	すべてのストレージプールを一覧表示します
GET	<i>/pools/<pool></i>	ストレージプールの詳細を取得します
POST	<i>/pools</i>	新しいストレージプールを構成します
PUT	<i>/pools/<pool></i>	プールに対してストレージを追加または削除します。
PUT	<i>/pools/<pool>/scrub</i>	指定されたプールでデータのスクラブを開始します
DELETE	<i>/pools/<pool>/scrub</i>	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	<i>/pools/<pool></i>	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。

表 62 ストレージプールのプロパティ

型	名前	説明
string	pool	ターゲットのプール名
string	profile	データデバイスのプロファイル
string	state	プールの状態(「online」、 「offline」、 「exported」)
string	asn	プールを所有するアプライアンス のシリアル番号
string	peer	クラスタ化されたシステムにおいて、 ピアノードの ASN
string	owner	プールを所有するシステムのホスト名

リクエストの例:

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

レスポンスの例:

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

{
  "pools": [
    {
      "profile": "mirror3",
      "name": "platinum",
      "peer": "00000000-0000-0000-0000-000000000000",
      "state": "online",
      "owner": "zfs-storage",
      "asn": "2f4aeeb3-b670-ee53-e0a7-d8e0ae410749"
    },
    {
      "profile": "raidz1",
      "name": "gold",
      "peer": "00000000-0000-0000-0000-000000000000",
      "state": "online",
      "owner": "zfs-storage",
      "asn": "2f4aeeb3-b670-ee53-e0a7-d8e0ae410749"
    }
  ]
}
```

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/gold HTTP/1.1
Host: zfs-storage.example.com
Accept: application/json
```

レスポンスの例:

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

{
  "pool": {
    "profile": "raidz1",
    "name": "gold",
    "usage": {
      "available": 2388000726016,
      "total": 2388001816576,
      "dedupratio": 100,
      "used": 1090560,
      "usage_data": 31744.0,
      "usage_snapshots": 0,
      "usage_total": 31744.0
    },
    "peer": "00000000-0000-0000-0000-000000000000",
    "state": "online",
    "owner": "tanana",
    "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。これは、問合せパラメータのプロパティを「true」に設定することで実行されます。

リクエストの例:

```
POST /api/storage/v1/pools?props=true HTTP/1.1
Host: zfs-storage.example.com
Authorization: Basic abhadbfsmwe=
Content-Type: application/json
Accept: application/json
```

```
{
  "name": "silver",
}
```

レスポンスの例:

```
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
Authorization: Basic abhadbfsmWE=
Content-Type: application/json
Accept: application/json
```

```
{
  "name": "silver",
  "profile": "stripe",
  "1-data": 8
}
```

レスポンスの例:

```
HTTP/1.1 201 Created
Content-Type: application/json

{
  "pool": {
    "asn": "314d252e-c42b-e844-dab1-a3bca680b563",
    "errors": [],
    "name": "silver",
    "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が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
Authorization: Basic abhadbfsmWE=
Content-Type: application/json
Accept: application/json

```

```

{
  "2-data": 8
}

```

レスポンスの例:

```

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、シャーシ番号、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
Authorization: Basic abhadbfsmWE=
Content-Type: application/json
Accept: application/json
{
  "0-cache" : 2
}

```

レスポンスの例:

```
HTTP/1.1 202 Accepted
```

削除可能なデバイスの数を表示するには、問い合わせパラメータ `props` 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 abhadbfsMWE=
Content-Type: application/json
Accept: application/json
```

レスポンスの例:

```
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` に `PUT` リクエストまたは `DELETE` リクエストを送信すると、それぞれ、プールのスクラブを開始するか、実行中のスクラブジョブを停止します。詳細は、CLI コマンドの「構成ストレージスクラブ」を参照してください。

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/silver HTTP/1.1
Host: zfs-storage.example.com
Authorization: Basic abhadbfsmWE=
```

レスポンスの例:

```
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、特定のプールへスコープ指定できます。すべてのプロジェクト全体で操作するコマンドは URI に「/projects」を追加し、単一のプロジェクトで操作するコマンドは「/projects/{project}」を追加します。

表 63 プロジェクトコマンド

リクエスト	パス /api/storage/v1	説明
GET	/projects	すべてのプロジェクトを一覧表示します
GET	/pools/<pool>/projects	プロジェクトを一覧表示します
GET	/pools/<pool>/projects /<project>	プロジェクトの詳細を取得します
POST	/pools/<pool>/projects	プロジェクトを作成します
PUT	/pools/<pool>/projects /<project>	プロジェクトを変更します
DELETE	/pools/<pool>/projects /<project>	プロジェクトを破棄します
GET	/pools/<pool>/projects /<project>/usage/groups	プロジェクトグループの使用状況を取得します
GET	/pools/<pool>/projects /<project>/usage/groups/<group>	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	/pools/<pool>/projects /<project>/usage/users	プロジェクトユーザーの使用状況を取得します
GET	/pools/<pool>/projects /<project>/usage/users/<user>	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 64 プロジェクトのプロパティ

型	名前	説明
string	aclinherit	ACL の継承動作 (「discard」 、 「noallow」 、 「restricted」 、 「passthrough」 、 「passthrough-x」 、 「passthrough-mode-preserve」)
string	aclmode	モード変更での ACL 動作 (「discard」 、 「mask」 、 「passthrough」)
boolean	atime	読み取りフラグでのアクセス時間を更新します
string	canonical_name	正規名
string	checksum	ブロックチェックサム (「fletcher2」 、 「fletcher4」 、 「sha256」)
string	compression	データ圧縮設定 (「off」 、 「lzb」 、 「gzip-2」 、 「gzip」 、 「gzip-9」)
number	copies	追加のレプリケーションコピーの数
datetime	creation	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
boolean	dedup	データ複製解除フラグ
string	default_group	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」
string	default_permissions	プロジェクトのデフォルトのファイルシステム権限 「700」
boolean	default_sparse	プロジェクトのデフォルトのLUN スパースデータフラグ
string	default_user	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」
number	default_volblocksize	プロジェクトのデフォルトのLUN ブロックサイズ: 8192
number	default_volsize	プロジェクトのデフォルトのLUN サイズ
boolean	exported	エクスポート済みフラグ
string	logbias	同期書き込みバイアス (「latency」 、 「throughput」)
string	mountpoint	シェアマウントポイントのデフォルト 「/export/proj-01」
string	name	プロジェクト名
boolean	nbmand	非ブロックの必須ロックフラグ
boolean	nodestroy	破棄防止フラグ

型	名前	説明
number	quota	プロジェクトの割り当て制限サイズ(バイト)
string	origin	クローン元
string	pool	プール名
boolean	readonly	true に設定されている場合データは読み取り専用です
string	recordsize	データベースのレコードサイズ 「128k」
number	reservation	データ予約サイズ
boolean	rstchown	所有権の変更制限フラグ
string	secondarycache	セカンダリキャッシュの使用状況 (「all」、「metadata」、「none」)
string	sharedav	HTTP シェア (「off」、「rw」、「ro」)
string	shareftp	FTP シェア (「off」、「rw」、「ro」)
string	sharenfs	NFS シェア (「off」、「on」、「ro」、「rw」)
string	sharesftp	SFTP シェア (「off」、「rw」、「ro」)
string	sharesmb	SMB/CIFS シェア (「off」、「rw」、「ro」)
string	sharetftp	TFTP シェア (「off」、「rw」、「ro」)
string	snapdir	.zfs/snapshot の可視性 (「hidden」、「visible」)
string	snaplabel	定期スナップショットラベル
boolean	vscan	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。返されたプロジェクトのおのおのには、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。

問合せパラメータフィルタ – 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。

表 65 URI パラメータ

パラメータ	説明
pool	ストレージプール名

リクエストの例:

```
GET /api/storage/v1/pools/gold/projects HTTP/1.1
Host: zfs-storage.example.com
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
Accept: application/json
```

プロジェクトのプロパティの取得

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とともに返されます。

表 66 プロジェクト取得の URI パラメータ

パラメータ	説明
pool	ストレージプール名
project	プロジェクト名

「gold」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/gold/projects/proj-01 HTTP/1.1
Host: zfs-storage.example.com
Accept: application/json
```

レスポンスの例:

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

{
  "project": {
    "default_volblocksize": 8192.0,
    "logbias": "latency",
    "creation": "20130411T20:02:35",
    "nodestroy": 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",
    "sharetftp": "",
    "default_sparse": false,
    "snapdir": "hidden",
    "aclmode": "discard",
    "copies": 1,
    "aclinherit": "restricted",
    "shareftp": "",
    "canonical_name": "gold/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": "gold",
    "default_user": "nobody",
    "name": "default",
    "checksum": "fletcher4",
    "default_group": "other",
    "sharesftp": "",
    "nbmand": false,
    "shredav": "",
    "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。デフォルトのプロパティを持つ新しいプロジェクトが返されます。

表 67 URI パラメータ

パラメータ	説明
pool	ストレージプール名

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** – 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。

リクエストの例 (「proj-01」という名前のプロジェクトを作成する):

```
POST /api/storage/v1/pools/gold/projects HTTP/1.1
Host: zfs-storage.example.com
Content-Type: application/json
Accept: application/json
```

```
{
  "name": "proj-01",
  "sharedfs": "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がすべて含まれます。

結果の例:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://zfs-storage.example.com:215
          /pools/gold/projects/proj-01

{
  "project": {
    "name": "proj-01",
    "href": "/api/storage/v1/pools/gold/projects/proj-01",
    "mountpoint": "/export/acme/gold",
    ...
  }
}
```

}

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。

表 68 URI パラメータ

パラメータ	説明
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。

リクエストの例 (プロジェクト名を「proj-01」から「new-name」に変更する):

```
POST /api/storage/v1/pools/gold/projects/proj-01 HTTP/1.1
Host: zfs-storage.example.com
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表示されます。

レスポンスの例:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: /api/storage/v1/pools/gold/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。

表 69 URI パラメータ

パラメータ	説明
pool	ストレージプール名
project	プロジェクト名

リクエストの例:

```
DELETE /api/storage/v1/pools/gold/projects/proj-01 HTTP/1.1
Host: zfs-storage.example.com
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 70 ファイルシステムのコマンド

リクエスト	パス /api/storage/v1	説明
GET	/filesystems	すべてのファイルシステムを一覧表示します
GET	/pools/<pool>/projects /<project>/filesystems	ファイルシステムを一覧表示します
GET	/pools/<pool>/projects /<project>/filesystems /<filesystem>	ファイルシステムの詳細を取得します
POST	/pools/<pool>/projects /<project>/filesystems	ファイルシステムを作成します
PUT	/pools/<pool>/projects /<project>/filesystems /<filesystem>	ファイルシステムを変更します
DELETE	/pools/<pool>/projects /<project>/filesystems /<filesystem>	ファイルシステムを破棄します
GET	/pools/<pool>/projects /<project>/filesystems /<filesystem>/usage/groups	ファイルシステムグループの使用状況を取得します

リクエスト	パス <code>/api/storage/v1</code>	説明
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

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 71 ファイルシステムのプロパティー

型	名前	説明
string	<code>casesensitivity</code>	大/小文字の区別の設定 (「mixed」、「sensitive」または「insensitive」)
string	<code>group</code>	グループ名
string	<code>normalization</code>	正規化
string	<code>permissions</code>	ファイルシステムの権限
string	<code>project</code>	プロジェクト名
boolean	<code>quota_snap</code>	割り当て制限にスナップショットを含めるフラグ
boolean	<code>reservation_snap</code>	予約にスナップショットを含めるフラグ
string	<code>shadow</code>	データ移行ソース
string	<code>sharesmb_name</code>	SMB シェアの名前
オブジェクト	<code>source</code>	プロジェクト継承プロパティー
オブジェクト	<code>usage</code>	ファイルシステムの使用状況情報
string	<code>user</code>	シェアを所有するユーザー名
boolean	<code>utf8only</code>	UTF-8 以外を拒否するフラグ

ファイルシステムの一覧表示

ファイルシステム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のファイルシステムがすべて一覧表示されます。

問合せパラメータ - フィルタ – 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。

表 72 URI パラメータ

パラメータ	説明
pool	ストレージプール名
project	プロジェクト名

リクエストの例:

```
GET /api/storage/v1/pools/gold/projects/proj-01/filesystems HTTP/1.1
Host: zfs-storage.example.com
Accept: application/json
```

リクエストに成功すると、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の配列とともに返されます。

結果例:

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

{
  "filesystems": [{
    "name": "filesystem-01",
    "project": "proj-01",
    "pool": "gold",
    ...
  }, {
    "name": "filesystem-02",
    "project": "proj-01",
    "pool": "gold",
    ...
  }]
}
```

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/v1/filesystems となります

リクエストの例 (プロパティの一部として「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):

```
GET /api/storage/v1/filesystems?filter=abcd HTTP/1.1
Host: zfs-storage.example.com
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が返されます。

表 73 URI パラメータ

パラメータ	説明
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

リクエストの例 (「proj-01」という名前のプロジェクトを一覧表示する):

```
GET /api/storage/v1/pools/gold/projects/proj-01 HTTP/1.1
Host: zfs-storage.example.com
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とともに返されます。

レスポンスの例:

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

```
{
  "filesystem": {
    "logbias": "latency",
    "creation": "20130423T21:30:34",
    "nodelay": false,
    "dedup": false,
    "sharenfs": "on",
    "sharesmb": "off",
    "mountpoint": "/export/grape",
    "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",
    "sharesftp": "",
    "canonical_name": "platinum/local/default/grape",
    "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": "platinum",
    "quota_snap": true,
    "name": "grape",
    "checksum": "fletcher4",
    "project": "default",
    "sharesftp": "",
    "nbmand": false,
    "reservation_snap": true,
    "sharedav": "",
    "rstchown": true
}
}

```

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。

表 74 URI パラメータ

パラメータ	説明
pool	ストレージプール名

パラメータ	説明
project	プロジェクト名
filesystem	ファイルシステム名

リクエストのパラメータ:

- **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** – ファイルシステム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 (「share-01」という名前で、ユーザー「joe」が所有するファイルシステムを作成する):

```
POST /api/storage/v1/pools/gold/projects/proj-01/filesystems HTTP/1.1
Host: zfs-storage.example.com
Content-Type: application/json
Accept: application/json

{
  "name": "share-01",
  "owner": "joe"
}
```

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がすべて含まれます。

レスポンスの例:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: /api/storage/v1/pools/gold/projects/proj-01/filesystems/share-01

{
  "filesystem": {
    "name": "share-01",
    "pool": "gold",
    "collection": "local",
    "project": "proj-01",
    "owner": "joe"
    ...
  }
}
```

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が一覧表示されます。

リクエストパラメータ - ファイルシステムプロパティ - 任意のファイルシステムプロパティまたはプロジェクトプロパティを変更できます。

表 75 URI パラメータ

パラメータ	説明
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/gold/projects/proj-01/filesystems/share-01
HTTP/1.1
Host: zfs-storage.example.com
Content-Type: application/json
Accept: application/json
```

```
{
  "name": "new-name",
  "owner": "nobody",
}
```

レスポンスの例:

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Location: http://zfs-storage.example.com:215
/pools/gold/projects/proj-01/filesystems/share-01
```

```
{
  "filesystem": {
    "name": "new-name",
    "pool": "gold",
    "collection": "local",
    "project": "proj-01",
    "owner": "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。

表 76 URI パラメータ

パラメータ	説明
pool	ストレージプール名

パラメータ	説明
project	プロジェクト名
filesystem	ファイルシステム名

リクエストの例:

```
DELETE /api/storage/v1/pools/gold/projects/proj-01/filesystems/share-01
HTTP/1.1
Host: zfs-storage.example.com
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。

レスポンスの例:

```
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 77 ボリュームのコマンド

リクエスト	パス /api/storage/v1	説明
GET	/luns	すべての LUN を一覧表示します
GET	/pools/<pool>/projects /<project>/luns	LUN を一覧表示します
GET	/pools/<pool>/projects /<project>/luns/<lun>	LUN の詳細を取得します
POST	/pools/<pool>/projects /<project>/luns	LUN を作成します
PUT	/pools/<pool>/projects /<project>/luns/<lun>	LUN を変更します

リクエスト	パス /api/storage/v1	説明
DELETE	/pools/<pool>/projects /<project>/luns/<lun>	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 78 ボリュームのプロパティ

型	名前	説明
string	assignednumber	割り当てられた LU 番号。
boolean	fixednumber	現在の値で LU 番号を修正するフラグ
string	initiatorgroup	イニシエータグループ
string	lunguid	STMF の GUID
string	lunnumber	LU 番号。数字または「auto」のいずれか
string	project	プロジェクト名 (変更不可)
オブジェクト	source	プロパティのソース (「local」、「inherited」) を一覧表示します
boolean	sparse	シンプロビジョニングを有効にするフラグ
string	status	論理ユニットのステータス (「online」、「offline」)
string	targetgroup	ターゲットグループ
オブジェクト	usage	LUN 使用状況の統計を一覧表示します
number	volblocksize	ボリュームのブロックサイズ
number	volsize	ボリュームのサイズ
boolean	writocache	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「local」であるか、プロジェクトから「inherited」されたかどうかを特定します。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。ソースを変更して継承に戻すために、プロパティを「unset」にできます。

圧縮を設定解除する JSON リクエストの例:

```
{"unset": ["compression"]}
```

LUN の一覧表示

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。

表 79 URI パラメータ

パラメータ	説明
pool	ストレージプール名
project	プロジェクト名

リクエストの例 (プロジェクト「proj-01」内の LUN を一覧表示する):

```
GET /api/storage/v1/pools/gold/projects/proj-01/luns HTTP/1.1
Host: zfs-storage.example.com
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。

レスポンスの例:

```
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。

表 80 URI パラメータ

パラメータ	説明
pool	ストレージプール名
project	プロジェクト名

パラメータ	説明
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/gold/projects/proj-01/lun/vol-01 HTTP/1.1
Host: zfs-storage.example.com
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。

レスポンスの例:

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

{
  "lun": {
    "logbias": "latency",
    "creation": "20130423T21:31:17",
    "nodestroy": 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": "platinum/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": "platinum",
    "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。

表 81 URI パラメータ

パラメータ	説明
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/gold/projects/proj-01/luns HTTP/1.1
Host: zfs-storage.example.com
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
  lunnumber : 'auto',        // Volume LUN 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 プロパティがすべて含まれます。

結果の例:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://zfs-storage.example.com:215
```

```

        /pools/gold/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。

表 82 URI パラメータ

パラメータ	説明
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 – 任意の LUN プロパティまたはプロジェクトプロパティを変更できます。

リクエストの例 (LUN 名を「vol-01」から「new-name」に変更する):

```

POST /api/storage/v1/pools/gold/projects/proj-01/luns/vol-01 HTTP/1.1
Host: zfs-storage.example.com
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 プロパティが一覧表示されます。

レスポンスの例:

```

HTTP/1.1 201 Created
Content-Type: application/json
Location: /api/storage/v1/pools/gold/projects/proj-01/luns/new-name
  
```

```

{
  "lun": {
    "name": "new-name",
    "pool": "gold",
    "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。

表 83 URI パラメータ

パラメータ	説明
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:

```
DELETE /pools/gold/projects/proj-01/luns/lun-01 HTTP/1.1
Host: zfs-storage.example.com
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。

レスポンスの例:

```
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/v1/pools/{pool}/projects/{project}` で始まります。
- すべてのファイルベースのスナップショット操作の URI は `/api/storage/v1/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/v1/pools/{pool}/projects/{project}/luns/{lun}` で始まります。

表 84 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/v1</code>	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 /<project>/snapshots</code>	すべてのプロジェクトスナップショットを一覧表示します

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/v1</i>	説明
GET	<i>/pools/<pool>/projects /<project>/filesystems /<filesystem>/snapshots</i>	すべてのファイルシステムスナップショットを一覧表示します
GET	<i>/pools/<pool>/projects /<project>/luns/<lun> /snapshots</i>	すべての LUN スナップショットを一覧表示します
GET	<i>/pools/<pool>/projects /<project>/snapshots/<snapshot></i>	プロジェクトスナップショットの詳細を取得します
GET	<i>/pools/<pool>/projects /<project>/filesystems /<filesystem>/snapshots /<snapshot></i>	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	<i>/pools/<pool>/projects /<project>/luns/<lun> /snapshots/<snapshot></i>	LUN スナップショットの詳細を取得します
POST	<i>/pools/<pool>/projects /<project>/snapshots</i>	プロジェクトスナップショットを作成します
POST	<i>/pools/<pool>/projects /<project>/filesystems /<filesystem>/snapshots</i>	ファイルシステムスナップショットを作成します
POST	<i>/pools/<pool>/projects /<project>/luns/<lun> /snapshots</i>	LUN スナップショットを作成します
PUT	<i>/pools/<pool>/projects /<project>/snapshots/<snapshot></i>	プロジェクトスナップショットを変更します
PUT	<i>/pools/<pool>/projects /<project>/filesystems /<filesystem>/snapshots /<snapshot></i>	ファイルシステムスナップショットを変更します
PUT	<i>/pools/<pool>/projects /<project>/luns/<lun> /snapshots/<snapshot></i>	LUN スナップショットを変更します
PUT	<i>/pools/<pool>/projects /<project>/filesystems /<filesystem>/snapshots /<snapshot>/clone</i>	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	<i>/pools/<pool>/projects /<project>/luns/<lun> /snapshots/<snapshot>/clone</i>	LUN スナップショットをクローン作成します
PUT	<i>/pools/<pool>/projects /<project>/filesystems /<filesystem>/snapshots /<snapshot>/rollback</i>	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	<i>/pools/<pool>/projects /<project>/luns/<lun> /snapshots/<snapshot>/rollback</i>	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	<i>/pools/<pool>/projects /<project>/snapshots/<snapshot></i>	プロジェクトスナップショットを破棄します
DELETE	<i>/pools/<pool>/projects /<project>/filesystems /<filesystem>/snapshots /<snapshot></i>	ファイルシステムスナップショットを破棄します
DELETE	<i>/pools/<pool>/projects /<project>/luns/<lun> /snapshots/<snapshot></i>	LUN スナップショットを破棄します
GET	<i>/pools/<pool>/projects /<project>/snapshots/<snapshot> /dependents</i>	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

リクエスト	パス <i>/api/storage/v1</i>	説明
GET	<i>/pools/<pool>/projects /<project>/filesystems /<filesystem>/snapshots /<snapshot>/dependents</i>	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	<i>/pools/<pool>/projects /<project>/lun/<lun> /snapshots/<snapshot> /dependents</i>	LUN スナップショットの依存を一覧表示します
POST	<i>/pools/<pool>/projects /<project>/automatic</i>	新しいプロジェクトの自動スナップショットを作成します
GET	<i>/pools/<pool>/projects /<project>/automatic /<automatic></i>	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	<i>/pools/<pool>/projects /<project>/automatic</i>	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	<i>/pools/<pool>/projects /<project>/automatic /<automatic></i>	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
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
POST	<i>/pools/<pool>/projects /<project>/filesystems /<filesystem>/automatic</i>	新しいファイルシステムの自動スナップショットを作成します
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
DELETE	<i>/pools/<pool>/projects /<project>/filesystems /<filesystem>/automatic /<automatic></i>	指定された自動オブジェクトを破棄します
POST	<i>/pools/<pool>/projects /<project>/luns/<lun>/automatic</i>	新しい LUN の自動スナップショットを作成します
GET	<i>/pools/<pool>/projects /<project>/luns/<lun>/automatic /<automatic></i>	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	<i>/pools/<pool>/projects /<project>/luns/<lun>/automatic</i>	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	<i>/pools/<pool>/projects /<project>/luns/<lun>/automatic /<automatic></i>	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
DELETE	<i>/pools/<pool>/projects /<project>/luns/<lun>/automatic /<automatic></i>	指定された自動オブジェクトを破棄します

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 85 スナップショットの一覧表示コマンドの形式

コマンド	/api/storage/v1/pools/{pool}/projects/{project}
プロジェクトスナップショットの一覧表示	/snapshots
ファイルシステムスナップショットの一覧表示	/filesystems/{share}/snapshots
LUN スナップショットの一覧表示	/lun/{share}/snapshots

リクエストの例:

```
GET /api/storage/v1/pools/gold/projects/default/snapshots
Accept: application/json
```

レスポンスの例:

```
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/gold/projects/default/snapshots/snap-001
Accept: application/json
```

レスポンスの例:

```
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/gold/projects/default/snapshots
Content-Type: application/json
```

```
{"name": "initial-backup"}
```

レスポンスの例:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: /pools/gold/projects/default/snapshot/initial-backup

{
  "snapshot": {
    "name": "initial-backup",
    "numclones": 0,
    "creation": "20130610T21:00:49",
    "collection": "local",
    "project": "default",
    "canonical_name": "gold/local/default@initial-backup",
    "usage": {
```

```

        "unique": 0.0,
        "loading": false,
        "data": 145408.0
    },
    "type": "snapshot",
    "id": "a26abd24-e22b-62b2-0000-000000000000",
    "pool": "gold"
}
}

```

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/gold/projects/default/snapshots/initial-snapshot
Content-Type: application/json
Accept: application/json

```

```
{"name": "old-snapshot"}
```

レスポンスの例:

```

HTTP/1.1 202 Accepted
Content-Type: application/json
Location: /pools/gold/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。

ファイルシステムのクローン作成のリクエスト URI:

```
PUT /pools/{pool}/projects/{project}/filesystems/{share}/snapshots/{snap}/clone
```

クローンボリューム:

```
PUT /pools/{pool}/projects/{project}/luns/{lun}/snapshots/{snapshot}/clone
```

表 86 URI パラメータ

パラメータ	説明
pool	ソースプール名

パラメータ	説明
project	ソースプロジェクト名
filesystem	ソースシェア名 (ファイルシステムスナップショットの場合)
lun	ソースシェア名 (LUN スナップショットの場合)
snapshot	ソーススナップショット名

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 87 スナップショットのクローン作成のプロパティ

型	名前	説明
string	pool	宛先クローンプール名
string	project	宛先クローンプロジェクト名
string	lun	宛先 LUN 名 (LUN スナップショットの場合)

リクエストの例:

```
PUT /api/storage/v1/pools/gold/projects/default/filesystems/fs01/
    snapshots/snap01/clone
{"project":"rest", "share":"snap01clone01", "compression": "gzip-9"}
```

レスポンスの例:

```
HTTP/1.1 201 Created
Content-Length: 2035
X-Zfssa-Storage-API: 1.0
Location: /api/storage/v1/pools/gold/projects/rest/filesystem/snap01clone01
Content-Type: application/json; charset=utf-8

{
  "filesystem": {
    "origin": {
      "project": "default",
      "share": "fs01",
      "snapshot": "snap01",
      "pool": "gold",
      "collection": "local"
    },
    "href": "/api/storage/v1/pools/gold/projects/rest/filesystems/snap01clone01",
    "mountpoint": "/export/snap01clone01",
    "compression": "gzip-9",
    "source": {
      "compression": "local",
      ...
    },
    ...
  },
  "canonical_name": "gold/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。

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/{share}/snapshots/{snap}/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
```

表 88 URI パラメータ

パラメータ	説明
pool	ソースプール名
project	ソースプロジェクト名
filesystem	ソースファイルシステム名 (ファイルシステムスナップショットの場合)
lun	ソース LUN 名 (LUN スナップショットの場合)
snapshot	ソーススナップショット名

リクエストの例:

```
PUT /api/storage/v1/pools/gold/projects/default/filesystems/fs-01/snapshots/initial-backup/rollback
```

レスポンスの例:

```
HTTP/1.1 202 Accepted
Location: /pools/gold/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": "gold/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": "gold"
  }
}

```

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。

表 89 スナップショットの削除コマンドの形式

コマンド	DELETE /api/storage/v1/pools/{pool}/projects/{project}
指定されたプール名、プロジェクト名、およびスナップショット名のプロジェクトスナップショットを削除	/snapshots/{snapshot_name}
指定されたプール名、プロジェクト名、ファイルシステム名、およびスナップショット名のファイルシステムスナップショットを削除	/filesystems/{share_name}/snapshots/{snapshot_name}
指定されたプール名、プロジェクト名、ファイルシステム名、およびスナップショット名のファイルシステム LUN を削除	/luns/{lun_name}/snapshots/{snapshot_name}

表 90 URI パラメータ

パラメータ	説明
pool	ソースプール名
project	ソースプロジェクト名
filesystem	ソースファイルシステム名
lun	LUN の名前
snapshot	ソーススナップショット名

スナップショットに NDMP ホールドがある場合、`?confirm=true` を DELETE コマンドに追加する必要があります。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 OS8.6.x](#)』の「[NDMP 構成](#)」を参照してください。

例

```
DELETE /pools/gold/projects/default/filesystems/fs-01/snapshots/initial-backup?confirm=true
```

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、コマンドは次の出力とともに失敗します。

```
HTTP/1.1 409 Conflict
```

```
{"fault": {"message": "request requires confirm=true to complete (confirmation needed for scripted command(scripted commands must be prefixed with \"confirm\" to automatically
```

```
confirm or \"deny\" to automatically deny) (encountered while attempting to run command \"confirm destroy snap\"))", "code": 409, "name": "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。

表 91 スナップショット依存の一覧表示コマンドの形式

コマンド	/api/storage/v1/pools/{pool}/projects/{project}
ファイルシステムの依存の一覧表示	/filesystems/{share}/snapshots/{snapshot}/dependents
ボリューム依存の一覧表示	lun/{lun}/snapshots/{snapshot}/dependents

表 92 URI パラメータ

パラメータ	説明
pool	システムのストレージプールの名前
project	プロジェクト名
filesystem	ファイルシステム名
lun	LUN の名前

リクエストの例:

```
GET /api/storage/v1/pools/gold/projects/default/filesystems/fs01/snapshots/snap01/dependents
Accept: application/json
```

レスポンスの例:

```
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/gold/projects/rest/filesystems/snap01clone01",
      "share": "snap01clone01"
    },
    {
      "project": "rest",
      "href": "/api/storage/v1/pools/gold/projects/rest/filesystems/snap01clone02",
      "share": "snap01clone02"
    },
    {
      "project": "rest",
      "href": "/api/storage/v1/pools/gold/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 93 スキーマプロパティ

リクエスト	パス /api/storage/v1	説明
GET	/schema	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	/schema/<property>	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	/schema	新しい NAS スキーマプロパティを作成します
PUT	/schema/<property>	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	/schema/<property>	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 int プロパティを変更します。

```
{"custom:priority": 5}
```

表 94 スキーマパラメータ

パラメータ	説明
プロパティ	プロパティの名前 (変更不可)
説明	プロパティの説明 (ブラウザインタフェースの場合)
type	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: zfssa.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://host.example.com:215/api/storage/v1/replication` から使用できます。

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 95 レプリケーションサービスコマンド

リクエスト	パス <code>/api/service/v1/services</code>	説明
GET	<code>/replication</code>	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	<code>/replication/enable</code>	レプリケーションサービスを有効にします
PUT	<code>/replication/disable</code>	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取得

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/v1/services/replication HTTP/1.1
Host: zfssa.example.com:215
Authorization: Basic ab6rt4psMWE=
```

Accept: application/json

結果の例:

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

```
{
  "service": {
    "<status>": "online",
    "href": "/service/v1/services/replication",
    "sources": [],
    "targets": []
  }
}
```

レプリケーションサービスの状態の変更

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 96 レプリケーションターゲットコマンド

リクエスト	パス /api/storage/v1	説明
POST	/replication/targets	新しいレプリケーションターゲットを作成します
GET	/replication/targets/<target>	指定されたレプリケーションターゲットプロパティを取得します
GET	/replication/targets	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	/replication/targets/<target>	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	/replication/targets/<target>	指定されたターゲットオブジェクトを破棄します

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/v1/replication/targets HTTP/1.1
Host: zfssa-storage.example.com:215
Authorization: Basic ab6rt4psMWE=
Accept: application/json
```

レスポンスの例:

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

{
  "targets": [{
    "address": "10.133.64.44:216",
    "label": "chum",
    "hostname": "10.133.64.44",
    "asn": "9d7a7543-ca83-68f5-a8fc-f818f65e1cfc",
    "actions": 0,
    "target": "target-000",
    "href": "/api/storage/v1/replication/targets/chum"
  }, {
    "address": "10.153.34.165:216",
    "label": "oakmeal-7320-165",
    "hostname": "10.153.34.165",
    "asn": "16a4c82c-26c1-4a50-e317-ac53181f2e86",
    "actions": 0,
    "target": "target-001",
    "href": "/api/storage/v1/replication/targets/oakmeal-7320-165"
  }
]
```

レプリケーションターゲットの取得

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/v1/replication/targets/chum HTTP/1.1
Authorization: Basic ab6rt4psMWE=
Host: zfs-storage.example.com:215
Accept: application/json
```

レスポンスの例:

```
HTTP/1.1 200 OK
X-Zfssa-Replication-API: 1.0
Content-Type: application/json
Content-Length: 337

{
  "target": {
    "href": "/api/storage/v1/replication/targets/chum",
    "address": "10.133.64.44:216",
    "label": "chum",
    "hostname": "10.133.64.44",
```

```

        "asn": "9d7a7543-ca83-68f5-a8fc-f818f65e1cfc",
        "actions": 0
    }
}

```

レプリケーションターゲットの作成

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。

リクエストの例:

```

POST /api/replication/v1/targets HTTP/1.1
Authorization: Basic ab6rt4psMWE=
Host: example.zfssa.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 54

{"hostname":"example", "root_password":"letmein", "label":"east"}

```

レスポンスの例:

```

HTTP/1.1 201 Created
Content-Length: 135
Content-Type: application/json
Location: /service/v1/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",
    "hostname": "example",
    "href": "/service/v1/services/replication/targets/target-000",
    "label": "east"
  }
}

```

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/v1/services/replication/targets/target-000 HTTP/1.1
Host: zfs-storage.example.com
Authorization: Basic ab6rt4psMWE=

```

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

レスポンスの例:

```

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

```

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。

表 97 レプリケーションアクションコマンド

リクエスト	/api/storage/v1	説明
GET	/replication/actions	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	/replication/actions/<ra_id>	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	/replication/actions	新しいレプリケーションアクションを作成します
PUT	/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変更します
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
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
GET	/pools/<pool>/projects /<project>/replication/actions	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	/pools/<pool>/projects /<project>/replication/actions /<ra_id>	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	/pools/<pool>/projects /<project>/replication/actions	新しいレプリケーションアクションを作成します

リクエスト	/api/storage/v1	説明
PUT	/pools/<pool>/projects /<project>/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変更します
DELETE	/pools/<pool>/projects /<project>/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	/pools/<pool>/projects /<project>/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	/pools/<pool>/projects /<project>/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
GET	/pools/<pool>/projects /<project>/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	/pools/<pool>/projects /<project>/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	/pools/<pool>/projects /<project>/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	/pools/<pool>/projects /<project>/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	/pools/<pool>/projects /<project>/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
GET	/pools/<pool>/projects /<project>/filesystems /<filesystem>/replication /actions	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	/pools/<pool>/projects /<project>/filesystems /<filesystem>/replication /actions /<ra_id>	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	/pools/<pool>/projects /<project>/filesystems /<filesystem>/replication /actions	新しいレプリケーションアクションを作成します
PUT	/pools/<pool>/projects /<project>/filesystems /<filesystem>/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変更します
DELETE	/pools/<pool>/projects /<project>/filesystems /<filesystem>/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
GET	/pools/<pool>/projects /<project>/filesystems /<filesystem>/replication	ファイルシステムのレプリケーションアクション設定を取得します
PUT	/pools/<pool>/projects /<project>/filesystems /<filesystem>/replication	ファイルシステムのレプリケーションアクション設定を変更します

レプリケーションアクション

リクエスト	/api/storage/v1	説明
PUT	/pools/<pool>/projects /<project>/ filesystems /<filesystem>/ 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	/pools/<pool>/projects /<project>/ filesystems /<filesystem>/ 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
GET	/pools/<pool>/projects /<project>/ filesystems /<filesystem>/ 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	/pools/<pool>/projects /<project>/ filesystems /<filesystem>/ 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	/pools/<pool>/projects /<project>/ filesystems /<filesystem>/ 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	/pools/<pool>/projects /<project>/ filesystems /<filesystem>/ 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	/pools/<pool>/projects /<project>/ filesystems /<filesystem>/ 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
GET	/pools/<pool>/projects /<project>/ luns/<lun> /replication/actions	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	/pools/<pool>/projects /<project>/ luns/<lun> /replication/actions/ <ra_id>	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	/pools/<pool>/projects /<project>/ luns/<lun> /replication/actions	新しいレプリケーションアクションを作成します
PUT	/pools/<pool>/projects /<project>/ luns/<lun> /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変更します
DELETE	/pools/<pool>/projects /<project>/ luns/<lun> /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
GET	/pools/<pool>/projects /<project>/ luns/<lun> /replication	LUN のレプリケーションアクション設定を取得します
PUT	/pools/<pool>/projects /<project>/ luns/<lun> /replication	LUN のレプリケーションアクション設定を変更します

リクエスト	/api/storage/v1	説明
PUT	/pools/<pool>/projects /<project>/luns/<lun> /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	/pools/<pool>/projects /<project>/luns/<lun> /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
GET	/pools/<pool>/projects /<project>/luns/<lun> /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	/pools/<pool>/projects /<project>/luns/<lun> /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	/pools/<pool>/projects /<project>/luns/<lun> /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	/pools/<pool>/projects /<project>/luns/<lun> /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	/pools/<pool>/projects /<project>/luns/<lun> /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

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/v1/replication/actions HTTP/1.1
Authorization: Basic ab6rt4psMWE=
Accept: application/json
```

レスポンスの例:

```
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/v1/replication/actions/1438ed7f-aad3-c631-d869-9e85cd7f15b4 HTTP/1.1
Authorization: Basic ab6rt4psMWE=
Accept: application/json
```

レスポンスの例:

```
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/v1/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": "gold",
    "project": "blah1",
    "retain_user_snaps_on_target": false,
    "share": "fs1",
    "state": "sending",
    "target": "38094753-6c90-49ed-aa92-995a296d432a",
    "use_ssl": true
  }
}
```

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 = (unset)
    pool = (unset)
    enabled = true
```

```
        continuous = false
        include_snaps = true
        max_bandwidth = unlimited
        bytes_sent = 0
        estimated_size = 0
        estimated_time_left = 0
        average_throughput = 0
        use_ssl = true
        compression = true
        retain_user_snaps_on_target = false
        include_clone_origin_as_data = false
```

リクエストの例:

```
POST /api/storage/v1/replication/actions HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic ab6rt4psMWE=
Content-Type: application/json
Content-Length: 121
Accept: application/json
```

```
{
  "pool": "gold",
  "project": "blue1",
  "share": "fs1",
  "target_pool": "pool1",
  "target": "38094753-6c90-49ed-aa92-995a296d432a"
}
```

レスポンスの例:

```
HTTP/1.1 201 Created
Content-Length: 506
Content-Type: application/json
Location: /api/storage/v1/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,
    "max_bandwidth": 0,
    "collection": "local",
    "estimated_size": 0.0,
    "state": "idle",
    "href": "/api/storage/v1/replication/pools/gold/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": "gold",
    "include_clone_origin_as_data": false,
    "include_snaps": true
  }
}
```

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/v1/replication/actions/c141d88d-ffd2-6730-d489-b71905f340cc HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic ab6rt4psMWE=
Content-Type: application/json

{"use_ssl": false}
```

レスポンスの例:

```
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
  }
}
```

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/v1/replication/actions/c141d88d-ffd2-6730-d489-b71905f340cc/cancelupdate
HTTP/1.1
Host: zfs-storage.example.com
Authorization: Basic ab6rt4psMWE=
```

レスポンスの例:

```
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に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/v1/replication/actions/c141d88d-ffd2-6730-d489-b71905f340cc/sendupdate
HTTP/1.1
Authorization: Basic ab6rt4psMWE=
```

レスポンスの例:

```
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/v1/replication/actions/e7e688b1-ff07-474f-d5cd-cac08293506e
HTTP/1.1
Host: zfs-storage.example.com
Authorization: Basic ab6rt4psMWE=
```

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

レスポンスの例:

```
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

レプリケーションソースおよびパッケージのコマンド。

表 98 レプリケーションソースおよびパッケージのコマンド

リクエスト	/api/storage/v1	説明
GET	/replication/sources	レプリケーションソースを一覧表示します
GET	/replication/sources/<source>	レプリケーションソースの詳細を一覧表示します
GET	/replication/sources/<source>/packages/<package>	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	/replication/sources/<source>/packages/<package>	指定されたレプリケーションパッケージを変更します

レプリケーションパッケージ

リクエスト	/api/storage/v1	説明
DELETE	/replication/sources/ <source> /packages/<package>	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	/replication/sources/ <source> /packages/ <package> /cancelupdate	指定されたパッケージでcancelupdateを実行します
PUT	/replication/sources/ <source> /packages/<package>/ sever	指定されたパッケージで切断を実行します
PUT	/replication/sources/ <source> /packages/<package>/ reverse	指定されたパッケージでの逆方向化を実行します
PUT	/replication/sources/ <source> /packages/<package>/ clone	指定されたパッケージのクローンを作成します
GET	/replication/sources/ <source> /packages/<package>/ clone /conflicts	シェアプロパティの競合を一覧表示します
GET	/replication/sources/ <source> /packages/<package>/ projects	パッケージプロジェクトを一覧表示します
GET	/replication/sources/ <source> /packages/<package>/ projects /<project>	パッケージプロジェクトを取得します
PUT	/replication/sources/ <source> /packages/<package>/ projects /<project>	パッケージプロジェクトを変更します
GET	/replication/sources/ <source> /packages/<package>/ projects /<project>/usage/groups	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	/replication/sources/ <source> /packages/<package>/ projects /<project>/usage/users	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
POST	/replication/sources/ <source> /packages/<package>/ projects /<project>/snapshots	新しいスナップショットを作成します
GET	/replication/sources/ <source> /packages/<package>/ projects /<project>/snapshots/ <snapshot>	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	/replication/sources/ <source> /packages/<package>/ projects /<project>/snapshots	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	/replication/sources/ <source> /packages/<package>/ projects /<project>/snapshots/ <snapshot>	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	<code>/api/storage/v1</code>	説明
PUT	<code>/replication/sources/ <source> /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
POST	<code>/replication/sources/ <source> /packages/<package>/ projects /<project>/automatic</code>	新しいパッケージプロジェクトの自動スナップショットを作成します
GET	<code>/replication/sources/ <source> /packages/<package>/ projects /<project>/automatic / <automatic></code>	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	<code>/replication/sources/ <source> /packages/<package>/ projects /<project>/automatic</code>	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
PUT	<code>/replication/sources/ <source> /packages/<package>/ projects /<project>/automatic / <automatic></code>	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オブジェクトを変更します
DELETE	<code>/replication/sources/ <source> /packages/<package>/ projects /<project>/automatic / <automatic></code>	指定された自動オブジェクトを破棄します
GET	<code>/replication/sources/ <source> /packages/<package>/ projects /<project>/filesystems</code>	パッケージファイルシステムを一覧表示します
GET	<code>/replication/sources/ <source> /packages/<package>/ projects /<project>/filesystems / <filesystem></code>	パッケージファイルシステムを取得します
PUT	<code>/replication/sources/ <source> /packages/<package>/ projects /<project>/filesystems / <filesystem></code>	パッケージファイルシステムを変更します
GET	<code>/replication/sources/ <source> /packages/<package>/ projects /<project>/filesystems / <filesystem>/usage/groups</code>	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	<code>/replication/sources/ <source> /packages/<package>/ projects /<project>/filesystems / <filesystem>/usage/users</code>	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
POST	<code>/replication/sources/ <source> /packages/<package>/ projects /<project>/filesystems / <filesystem>/snapshots</code>	新しいスナップショットを作成します
GET	<code>/replication/sources/ <source> /packages/<package>/ projects /<project>/filesystems / <filesystem>/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

リクエスト	/api/storage/v1	説明
GET	/replication/sources/ <source> /packages/<package>/ projects /<project>/filesystems / <filesystem>/snapshots	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	/replication/sources/ <source> /packages/<package>/ projects /<project>/filesystems / <filesystem>/snapshots /<snapshot>	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	/replication/sources/ <source> /packages/<package>/ projects /<project>/filesystems / <filesystem>/snapshots /<snapshot>	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
POST	/replication/sources/ <source> /packages/<package>/ projects /<project>/filesystems / <filesystem>/automatic	新しいパッケージファイルシステムの自動スナップショットを作成します
GET	/replication/sources/ <source> /packages/<package>/ projects /<project>/filesystems / <filesystem>/automatic /<automatic>	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	/replication/sources/ <source> /packages/<package>/ projects /<project>/filesystems / <filesystem>/automatic	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
PUT	/replication/sources/ <source> /packages/<package>/ projects /<project>/filesystems / <filesystem>/automatic /<automatic>	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オブジェクトを変更します
DELETE	/replication/sources/ <source> /packages/<package>/ projects /<project>/filesystems / <filesystem>/automatic /<automatic>	指定された自動オブジェクトを破棄します
GET	/replication/sources/ <source> /packages/<package>/ projects /<project>/luns	パッケージ LUN を一覧表示します
GET	/replication/sources/ <source> /packages/<package>/ projects /<project>/luns/<lun>	パッケージ LUN を取得します
PUT	/replication/sources/ <source> /packages/<package>/ projects /<project>/luns/<lun>	パッケージ LUN を変更します
GET	/replication/sources/ <source> /packages/<package>/ projects /<project>/luns/<lun>/ usage /groups	パッケージ LUN グループの使用状況を取得します
GET	/replication/sources/ <source> /packages/<package>/ projects /<project>/luns/<lun>/ usage /users	パッケージ LUN ユーザーの使用状況を取得します

リクエスト	/api/storage/v1	説明
POST	/replication/sources/ <source> /packages/<package>/ projects /<project>/luns/<lun>/ snapshots	新しいスナップショットを作成します
GET	/replication/sources/ <source> /packages/<package>/ projects /<project>/luns/<lun>/ snapshots /<snapshot>	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	/replication/sources/ <source> /packages/<package>/ projects /<project>/luns/<lun>/ snapshots	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	/replication/sources/ <source> /packages/<package>/ projects /<project>/luns/<lun>/ snapshots /<snapshot>	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	/replication/sources/ <source> /packages/<package>/ projects /<project>/luns/<lun>/ snapshots /<snapshot>	パッケージ LUN スナップショットの名前を変更します
POST	/replication/sources/ <source> /packages/<package>/ projects /<project>/luns/<lun>/ automatic	新しいパッケージ LUN の自動スナップショットを作成します
GET	/replication/sources/ <source> /packages/<package>/ projects /<project>/luns/<lun>/ automatic /<automatic>	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
GET	/replication/sources/ <source> /packages/<package>/ projects /<project>/luns/<lun>/ automatic	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
PUT	/replication/sources/ <source> /packages/<package>/ projects /<project>/luns/<lun>/ automatic /<automatic>	指定されたパッケージ LUN 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/sources/ <source> /packages/<package>/ projects /<project>/luns/<lun>/ automatic /<automatic>	指定された自動オブジェクトを破棄します

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/v1/replication/sources HTTP/1.1
```

```
Host: zfs-storage.example.com
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/v1/replication/sources/zfssa-repl-host",
    "ip_address": "10.80.231.58:216",
    "name": "zfssa-repl-host",
    "source": "source-000"
  }]
}
```

レプリケーションパッケージの一覧表示

指定されたレプリケーションソースから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/v1/replication/sources/zfssa-repl/packages HTTP/1.1
Host: zfs-storage.example.com
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": [{
    "enabled": true,
    "href": "/api/v1/storage/replication/sources/zfssa-repl/packages/package-008",
    "id": "b2d8b35a-a5a0-6c74-f7e9-b75c357e841f",
    "last_result": "unknown",
    "last_sync": "unknown",
    "last_try": "unknown",
    "state": "idle",
    "state_description": "Idle (no update in progress)"
  }, {
    "enabled": true,
    "href": "/api/storage/v1/replication/sources/zfssa-repl/packages/package-009",
    "id": "2643a0eb-648d-6ad7-d405-b690d06f6cf6",
    "last_result": "success",
    "last_sync": "Wed Jul 31 2013 21:58:02 GMT+0000 (UTC)",
    "last_try": "Wed Jul 31 2013 21:58:02 GMT+0000 (UTC)",
    "state": "idle",
    "state_description": "Idle (no update in progress)",
    "project": "gold/nas-rr-2643a0eb-648d-6ad7-d405-b690d06f6cf6/default",
  }
]
```

```
  ]}
```

パッケージの変更

パッケージプロパティを変更します。

表 99 パッケージプロパティの変更

型	名前	説明
boolean	enabled	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/v1/replication/sources/zfssa-repl/packages/
    8373d331-de60-e590-90e8-9ad69fcb4aec HTTP/1.1
Host: zfs-storage.example.com
Authorization: Basic ab6rt4psMWE=
Content-Type: application/json
```

```
{"enabled": false}
```

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/v1/replication/sources/zfssa-repl/packages/
    /8373d331-de60-e590-90e8-9ad69fcb4aec HTTP/1.1
Host: zfs-storage.example.com
Authorization: Basic ab6rt4psMWE=
```

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

レスポンスの例:

```
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/v1/replication/sources/zfssa-repl/packages/
8373d331-de60-e590-90e8-9ad69fcb4aec/cancelupdate HTTP/1.1
Host: zfs-storage.example.com
Authorization: Basic ab6rt4psMWE=
```

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。

レスポンスの例:

```
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"
  }
}
```

レスポンスの例:

```
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/v1/storage/replication/sources/zfssa-repl/packages/
8373d331-de60-e590-90e8-9ad69fcb4aec/clone HTTP/1.1
Host: zfs-storage.example.com
Authorization: Basic ab6rt4psMWE=
```

レスポンスの例:

```
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して、クローン操作との競合が存在するかどうかを判断できます。

クローンの競合リクエストの例:

```
GET /api/storage/v1/replication/sources/zfssa-repl/packages/
8373d331-de60-e590-90e8-9ad69fcb4aec/clone/conflicts HTTP/1.1
Host: zfs-storage.example.com
Authorization: Basic ab6rt4psMWE=
```

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.\n"
}
```

プロパティ:

```
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/v1/replication/sources/zfssa-repl/packages/
    8373d331-de60-e590-90e8-9ad69fcb4aec/sever HTTP/1.1
Host: zfs-storage.example.com
Authorization: Basic ab6rt4psMWE=

{"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/v1/replication/sources/zfssa-repl/packages/
    8373d331-de60-e590-90e8-9ad69fcb4aec/reverse HTTP/1.1
```

```
Host: zfs-storage.example.com
Authorization: Basic ab6rt4psMWE=
```

```
{"projname":"restrev"}
```

成功のレスポンス:

```
HTTP/1.1 202 Accepted
X-Zfssa-Replication-API: 1.0
```

暗号化

ライセンスについて: 暗号化は無償で評価することが許可されていますが、本稼働環境で使用するには、機能のライセンスを個別に購入する必要があります。評価期間を過ぎたら、この機能のライセンスを購入するか、機能を非アクティブ化する必要があります。オラクル社は、ライセンスが遵守されているかどうかをいつでも監査する権利を保持しています。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」を参照してください。

Oracle ZFS Storage Appliance は、プロジェクトレベルおよび個々のシェア (ファイルシステムと LUN) レベルで透過的なデータ暗号化を提供します。アプライアンスは、組み込みのローカルキーストアを含んでおり、Oracle Key Manager (OKM) システムにも接続できます。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または OKM 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、ローカルまたは OKM キーストアからの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。

次の表では、ローカルおよび OKM 暗号化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 100 ローカル暗号化

リクエスト	パス	説明
	/api/storage/v1	
GET	/encryption/local	ローカルキーストアのプロパティを取得します
PUT	/encryption/local	ローカルキーストアのプロパティを変更します
GET	/encryption/local/keys	ローカル鍵を取得します
GET	/encryption/local/keys/<key>	ローカル鍵の詳細を取得します
POST	/encryption/local/keys	ローカル鍵を作成します
DELETE	/encryption/local/keys/<key>	ローカル鍵を破棄します

リクエスト	パス	説明
	/api/storage/v1	
GET	/encryption/local/keys/ <key>/dependents	この鍵に依存している シェアを一覧表示します

表 101 OKM 暗号化

リクエスト	パス	説明
	/api/storage/v1	
GET	/encryption/okm	OKM キーストアのプロ パティアーを取得します
PUT	/encryption/okm	OKM キーストアのプロ パティアーを変更します
GET	/encryption/okm/keys	OKM 鍵を取得します
GET	/encryption/okm/keys/ <key>	OKM 鍵の詳細を取得し ます
POST	/encryption/okm/keys	OKM 鍵を作成します
DELETE	/encryption/okm/keys/ <key>	OKM 鍵を破棄します
GET	/encryption/okm/keys/ <key>/dependents	この鍵に依存している シェアを一覧表示します

すべてのローカル鍵の一覧表示

出力:

```
{
  "keys": [{
    "cipher": "AES",
    "keyname": "key-1",
    "href": "/api/storage/v1/encryption/local/keys/key-000"
  }, {
    "cipher": "AES",
    "keyname": "key-2",
    "href": "/api/storage/v1/encryption/local/keys/key-001"
  }, {
    "cipher": "AES",
    "keyname": "key-3",
    "href": "/api/storage/v1/encryption/local/keys/key-002"
  }
]}
```

ローカル鍵の一覧表示

出力:

```
{
```

```
"key": {  
  "href": "/api/storage/v1/encryption/local/keys/key-000",  
  "cipher": "AES",  
  "keyname": "key-1"  
}
```

すべての OKM 鍵の一覧表示

出力:

```
{  
  "keys": [{  
    "cipher": "AES",  
    "keyname": "okm-key-1",  
    "href": "/api/storage/v1/encryption/local/keys/key-000"  
  }, {  
    "cipher": "AES",  
    "keyname": "okm-key-2",  
    "href": "/api/storage/v1/encryption/local/keys/key-001"  
  }, {  
    "cipher": "AES",  
    "keyname": "okm-key-3",  
    "href": "/api/storage/v1/encryption/local/keys/key-002"  
  }  
}]  
}
```

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 102 アプライアンスシステムコマンド

リクエスト	パス /api/system/v1	説明
GET	/version	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	/diagreboot	アプライアンスをリブートし、プロセスにおいて追加の診断情報を収集します。
PUT	/reboot	アプライアンスをリブートします
PUT	/poweroff	アプライアンスをオフにします
PUT	/restart	管理インターフェースをリブートし、診断情報を収集します
PUT	/factoryreset	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	/disks	すべてのシステムディスクを一覧表示します
GET	/disks/<disk>	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	/memory	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
Accept: application/json
```

レスポンスの例:

```
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": "tanana",
    "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
```

システムのリポート

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。

リクエストの例:

```
PUT /api/system/v1/reboot HTTP/1.1
Host: zfs-storage.example.com
```

システム管理の再起動

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
```

診断リポート

アプライアンスをリポートし、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。

リクエストの例:

```
PUT /api/system/v1/diagreboot HTTP/1.1
Host: zfs-storage.example.com
```

出荷時リセット

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、問合せパラメータ「confirm=true」に設定しないと、コマンドは失敗します。

リクエストの例:

```
PUT /api/system/v1/factoryreset?confirm=true HTTP/1.1
Host: zfs-storage.example.com
```

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 103 サポートバンドルコマンド

リクエスト	パス <i>/api/system/v1</i>	説明
GET	<i>/bundles</i>	すべてのサポートバンドルを一覧表示します
GET	<i>/bundles/<bundle></i>	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	<i>/bundles</i>	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	<i>/bundles/<bundle>/retry</i>	指定されたバンドルのアップロードを再試行します
PUT	<i>/bundles/<bundle>/cancel</i>	指定されたバンドルのアップロードを取り消します
PUT	<i>/bundles/<bundle>/send</i>	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	<i>/bundles/<bundle></i>	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 番号は「3-*nnnnnnnnnn*」の形式でなければいけません。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 abhaddbfSMWE=
Host: zfssa.example.com:215
Accept: application/json
Content-Type: application/json
```

Content-Length: 23

```
{"srn": "3-0123456789"}
```

レスポンスの例:

```
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 abhadbfsMWE=
Host: zfssa.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 23
```

レスポンスの例:

```
{
  "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 abhadbfsMWE=
Host: zfssa.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 abhadbfsmWE=
Host: zfssa.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 abhadbfsmWE=
```

```
Host: zfssa.example.com:215
```

レスポンスの例:

```
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 abhadbfsMWE=
Host: zfssa.example.com:215
```

レスポンスの例:

```
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 abhadbfsMWE=
Host: zfssa.example.com:215
```

```
{"srn": "3-0123456789"}
```

レスポンスの例:

```
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 abhaddbfsMWE=
Host: zfssa.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 abhaddbfsMWE=
Host: zfssa.example.com:215
```

レスポンスの例:

```
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 104 更新コマンド

リクエスト	パス <code>/api/system/v1</code>	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
GET	<code>/updates-firmware</code>	アップグレードされる残りのコンポーネントを、最後の試行の時間および現在のステータスとともに一覧表示します
PUT	<code>/updates/<update></code>	更新設定を変更します

リクエスト	パス <code>/api/system/v1</code>	説明
PUT	<code>/updates/<update>/upgrade</code>	指定された更新イメージへアップグレードします
PUT	<code>/updates/<update>/check</code>	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	<code>/updates/<update>/rollback</code>	指定された更新イメージにロールバックします
PUT	<code>/updates-apply</code>	互換性のない遅延更新を適用します
DELETE	<code>/updates/<update></code>	指定されたシステム更新を破棄します
POST	<code>/updates</code>	更新イメージをアプライアンスにロードします

表 105 システム更新プロパティ

名前	型	説明
<code>version</code>	String	媒体バージョンを更新します
<code>date</code>	DateTime	リリース日を更新します
<code>status</code>	String	媒体ステータスを更新します (変更不可)
<code>update_deferred</code>	ChooseOne	遅延設定 [「onreboot」, 「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 abcdefgMWE=
Host: zfssa.example.com:215
Accept: application/json
```

レスポンスの例:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Length: 541
Content-Type: application/json
```

```
{
  "updates": [
    {
      "date": "Tue Aug 13 2013 17:47:32 GMT+0000 (UTC)",
      "href": "/api/system/v1/updates/nas@2013.08.13,1-0",
      "status": "previous",
      "version": "2013.08.13,1-0"
    },
    {
      "date": "Sat Aug 24 2013 17:54:23 GMT+0000 (UTC)",
      "href": "/api/system/v1/updates/nas@2013.08.24,1-0",
      "status": "current",
      "version": "2013.08.24,1-0"
    },
    {
      "date": "Sun Aug 25 2013 12:56:57 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 abcefgMWE=
Host: zfssa.example.com:215
Accept: application/json
```

レスポンスの例:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Length: 541
Content-Type: application/json
```

```
{
  "update": {
    "date": "Sat Aug 24 2013 17:54:23 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"
  }
}
```

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 使用したアップロードコマンドの例:

```
curl --user root:letmein -k --data-binary @nas@2013.08.24,1-0.pkg.gz \  
--header "Content-Type: application/octet-stream" \  
https://zfssa.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。

結果の例:

```
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": {  
    "date": "Sat Aug 24 2013 17:54:23 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: zfssa.example.com:215  
Authorization: Basic abcdefgMWE=  
Content-Length: 0
```

レスポンスの例:

```
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: zfssa.example.com:215
Authorization: Basic abcefgMWE=
Content-Length: 0
```

レスポンスの例:

```
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: zfssa.example.com:215
Authorization: Basic abcefgMWE=
```

レスポンスの例:

```
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、アプライアンスでのローカル管理ユーザーおよびユーザー設定の構成に使用されます。

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 106 ユーザーサービスのコマンド

リクエスト	パス <i>/api/user/v1</i>	説明
GET		ユーザーサービスコマンドを一覧表示します
GET	<i>/users</i>	すべてのユーザーについてのサマリー情報を一覧表示します
GET	<i>/users/<user></i>	特定のユーザーに関する詳細情報を取得します
DELETE	<i>/users/<user></i>	システムからローカルユーザーを削除します
POST	<i>/users</i>	新しいローカルユーザーを作成するか、既存のユーザーを新しいユーザーとしてクローン作成するか、ネットワークディレクトリから管理者を追加します
PUT	<i>/users/<user></i>	ユーザープロパティを変更します
PUT	<i>/users/<user>/preferences</i>	ユーザープリファレンスを変更します
GET	<i>/users/<user>/preferences</i>	ユーザーのプリファレンスを取得します
POST	<i>/users/<user>/exceptions</i>	新しいユーザー承認の例外を作成します
GET	<i>/users/<user>/exceptions/<auth></i>	指定されたユーザー承認の例外プロパティを取得します
GET	<i>/users/<user>/exceptions</i>	すべてのユーザー承認の例外オブジェクトを一覧表示します

リクエスト	パス /api/user/v1	説明
PUT	/users/<user>/exceptions/<auth>	指定されたユーザー承認の例外オブジェクトを変更します
DELETE	/users/<user>/exceptions/<auth>	指定された承認オブジェクトを破棄します
POST	/users/<user>/preferences/keys	新しいユーザーの ssh 鍵を作成します
GET	/users/<user>/preferences/keys /<key>	指定されたユーザーの ssh 鍵プロパティを取得します
GET	/users/<user>/preferences/keys	すべてのユーザーの ssh 鍵オブジェクトを一覧表示します
PUT	/users/<user>/preferences/keys /<key>	特定のユーザーの指定された ssh 鍵を変更します
DELETE	/users/<user>/preferences/keys /<key>	指定された鍵オブジェクトを破棄します

ユーザーの一覧表示

各ユーザーには、次の使用可能なサマリープロパティがあります。

表 107 ユーザープロパティ

型	プロパティ名	説明
string	logname	ユーザー名 (作成後は変更不可)
number	uid	ユーザー ID、ディレクトリユーザーに対して有効になっていません
string	fullname	フルネーム
string	initial_password	パスワード
boolean	require_annotation	セッションの注釈を必要とするフラグです
string	roles	このユーザーのロールです
boolean	kiosk_mode	キオスクユーザー
string	kiosk_screen	キオスク画面

リクエストの例:

```
GET /api/user/v1/users HTTP/1.1
Authorization: Basic abcdefgMWE=
Host: zfssa.example.com:215
Accept: application/json
```

レスポンスの例:

```

{
  "user":
  {
    "href": "/api/user/v1/users/john",
    "logname": "john",
    "type": "local",
    "uid": 2000000000,
    "fullname": "John Doe",
    "initial_password": "DummyPassword",
    "require_annotation": false,
    "roles": [
      "basic"
    ],
    "kiosk_mode": false,
    "kiosk_screen": "status/dashboard",
    "exceptions": [
    ],
    "preferences": {
      "href": "/api/user/v1/users/john/preferences",
      "locale": "C",
      "login_screen": "status/dashboard",
      "session_timeout": 15,
      "advanced_analytics": false,
      "keys": [
      ]
    }
  }
}

```

ユーザーの取得

ユーザーに関する詳細な情報を取得し、ユーザー設定および承認の例外を含めます。各承認例外タイプは、独自のプロパティを定義します。ユーザー設定プロパティが示されます。

表 108 ユーザー設定

型	プロパティ名	説明
string	locale	場所
string	login_screen	初期ログイン画面
string	session_timeout	セッションタイムアウト (分)
string	advanced_analytics	使用可能な高度な分析統計を作成します

各ユーザーは、定義済み設定の一部として指定された ssh 鍵を持つことができます。

表 109 SSH 鍵プロパティ

型	プロパティ名	説明
string	type	SSH 鍵のタイプです。RSA または DSA

型	プロパティ名	説明
string	key	SSH 鍵の内容
string	comment	SSH 鍵に関連付けられたコメント

リクエストの例:

```
GET /api/user/v1/users/joe HTTP/1.1
Authorization: Basic abcdefgMWE=
Host: zfssa.example.com:215
Accept: application/json
```

レスポンスの例:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 390

{
  "user": {
    "fullname": "John Doe",
    "href": "/api/user/v1/users/john",
    "initial_password": "DummyPassword",
    "kiosk_mode": false,
    "kiosk_screen": "status/dashboard",
    "logname": "john",
    "require_annotation": false,
    "roles": ["basic"]
  }
}
```

ユーザーの作成

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 OS8.6.x](#)』の「[ユーザーとロールについて](#)」を参照してください。

このコマンドは、次の3つの形式を使用します。

- **新規ユーザーの作成** - 新しいユーザーを作成します
- **既存のユーザーのクローン作成** - 既存のユーザーから新しいユーザーをクローン作成します
- **管理者の追加** - 企業ディレクトリ内で定義されているユーザーがアプライアンスを管理できるようにします。

3つすべての場合において、本文でのJSON形式のプロパティとともにユーザーへのPOSTリクエストが送信されます。

新規ユーザーの作成のプロパティ

表 110 新規ユーザーの作成のプロパティ

型	プロパティ名	説明
string	logname	新規ユーザーのログイン名 (必須)
number	uid	オプションユーザー ID
string	fullname	新規ユーザーのフルネーム (必須)
string	type	「ローカル」、「データ」、 「ログインなし」(デフォルトは 「ローカル」)
string	initial_password	初期のユーザーパスワード (「ローカル」および「データ」 のみ)
boolean	require_annotation	セッションの注釈を必要とする オプションフラグです(「ローカ ル」のみ)

ユーザーのクローン作成プロパティ

表 111 ユーザーのクローン作成プロパティ

型	プロパティ名	説明
string	user	ソースユーザー名
number	uid	ユーザー ID、ディレクトリユー ザーに対して有効になっていま せん
string	clonename	新規クローンのログイン名
string	fullname	新規のクローンユーザーのフル ネーム (ディレクトリユーザーの ものではない)
string	password	新規のクローンユーザーのパス ワード (ディレクトリユーザーま たはログインなしユーザーのも のではない)

管理者の追加プロパティ

表 112 管理者の追加プロパティ

型	プロパティ名	説明
string	type	ディレクトリユーザー
string	logname	ディレクトリユーザーログイン名

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
Host: zfssa.example.com:215
Authorization: Basic abcdefghijklmnop
Content-type: application/json
```

```
{
  "type": "local",
  "logname": "john",
  "initial_password": "DummyPassword",
  "fullname": "John Doe"
}
```

結果例:

```
{
  "user": {
    "href": "/api/user/v1/users/john",
    "logname": "john",
    "type": "local",
    "uid": 2000000002,
    "fullname": "John Doe",
    "initial_password": "DummyPassword",
    "require_annotation": false,
    "roles": [
      "basic"
    ],
    "kiosk_mode": false,
    "kiosk_screen": "status/dashboard",
    "exceptions": [
    ],
    "preferences": {
      "href": "/api/user/v1/users/john/preferences",
      "locale": "C",
      "login_screen": "status/dashboard",
      "session_timeout": 15,
      "advanced_analytics": false,
      "keys": [
      ]
    }
  }
}
```

例 2 ディレクトリユーザーの作成

リクエストの例:

```
POST /api/user/v1/users
{
  "type": "directory",
  "logname": "john"
}
```

結果例:

```
{
  "user": {
    {
```

```

    "href": "/api/user/v1/users/john",
    "logname": "john",
    "type": "directory",
    "uid": 26718,
    "fullname": "John Doe",
    "require_annotation": false,
    "roles": [
      "basic"
    ],
    "kiosk_mode": false,
    "kiosk_screen": "status/dashboard",
    "exceptions": [
    ],
    "preferences": {
      "href": "/api/user/v1/users/john/preferences",
      "locale": "C",
      "login_screen": "status/dashboard",
      "session_timeout": 15,
      "advanced_analytics": false,
      "keys": [
      ]
    }
  }
}

```

例 3 データ専用ユーザーの作成

リクエストの例:

```

POST /api/user/v1/users
{
  "type": "data",
  "logname": "john",
  "initial_password": "password",
  "fullname": "John Doe",
  "uid": 5000000
}

```

結果例:

```

{
  "user":
  {
    "href": "/api/user/v1/users/data",
    "logname": "john",
    "type": "data",
    "uid": 5000000,
    "fullname": "John Doe",
    "initial_password": "DummyPassword"
  }
}

```

例 4 ログインなしユーザーの作成

リクエストの例:

```

POST /api/user/v1/users

```

```
{
  "type": "nologin",
  "logname": "john",
  "fullname": "John Doe",
  "uid": 5000001
}
```

結果例:

```
{
  "user": {
    "href": "/api/user/v1/users/john",
    "logname": "john",
    "type": "nologin",
    "uid": 5000001,
    "fullname": "John Doe"
  }
}
```

ユーザーの変更

ユーザーのプロパティを直接変更します。ユーザーリソース: 例外、プリファレンス、および ssh 鍵を追加、変更または削除できます。一度作成されると、「UID」および「タイプ」は変更できません。

リクエストの例:

```
PUT /api/user/v1/users/joe HTTP/1.1
Authorization: Basic abcdefgMWE=
Host: zfssa.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 24

{"require_annotation": true}
```

結果例:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 236

{
  "user": {
    "href": "/api/user/v1/users/john",
    "logname": "john",
    "type": "local",
    "uid": 2000000000,
    "fullname": "John Doe",
    "initial_password": "DummyPassword",
    "require_annotation": true,
    "kiosk_mode": false,
    "kiosk_screen": "status/dashboard",
    "roles": ["basic"]
  }
}
```

```
}
```

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/joe HTTP/1.1
Authorization: Basic abcefgMWE=
Host: zfssa.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 113 ワークフローサービスコマンド

リクエスト	パス <code>/api/workflow/v1</code>	説明
GET		ワークフローサービスコマンドを一覧表示します。
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>/workflows</code>	新しいワークフローをアプライアンスにロードします

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 abcdefgMWE=
Host: zfssa.example.com:215
Accept: application/json
```

レスポンスの例:

```
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-cfbe527df92d",
    ...
  },
  {
    "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。

リクエストの例 - Accept に application/javascript を指定した場合:

```
GET /api/workflow/v1/workflows/cc574599-4763-4523-9e72-b74e1246d448 HTTP/1.1
Authorization: Basic cm9vdDpsMWE=
Host: zfssa.example.com:215
Accept: application/javascript
```

レスポンスの例:

```
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'
    },
    ipaddr: {
      label: 'Client IP address',
      type: 'String'
    }
  },
  validate: function (params) {
    if (params.hostname == '') {
      return ({ hostname: 'Hostname cannot be empty.' });
    }

    if (params.ipaddr == '') {
      return ({ ipaddr: 'IP address cannot be empty.' });
    }
  },
  execute: function (params) {
    try {
      nas.clearLocks(params.hostname, params.ipaddr);
    } catch (err) {
      return ('Failed to clear NFS locks: ' + err.message);
    }

    return ('Clear of locks held for ' + params.hostname +
      ' returned success. ');
  }
};
```

リクエストの例 - Accept を指定しなかった場合や application/json を指定した場合:

```
GET /api/workflow/v1/workflows/cc574599-4763-4523-9e72-b74e1246d448 HTTP/1.1
Authorization: Basic cm9vdDpsMWE=
Host: zfssa.example.com:215
Accept: application/json
```

レスポンスの例:

```
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 abcefgMWE=
Host: zfssa.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 28
```

```
{"setid": false}
```

レスポンスの例:

```
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/execute HTTP/1.1
Authorization: Basic abcefgMWE=
Host: zfssa.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 abcefgMWE=
Host: zfssa.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 abcefgMWE=
Host: zfssa.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: jkremer/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
  }
}
```

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。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

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://zfssa.example.com:215/api/nas/v1/pools/gold

HTTP/1.1 200 OK
Date: Tue, 23 Jul 2013 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": "gold",
    "usage": {
      "available": 895468984832.0,
      "total": 895500681216.0,
      "dedupratio": 100,
      "used": 31696384.0
    }
  },
}
```

```
    "peer": "00000000-0000-0000-0000-000000000000",
    "state": "online",
    "owner": "tanana",
    "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://zfssa-host.example.com:215/api/user/v1/users <<JSON
> {"logname": "rest_user",
>  "fullname": "REST User",
>  "initial_password": "letmein"}
> JSON
```

```
HTTP/1.1 201 Created
Date: Tue, 23 Jul 2013 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": "DummyPassword",
    "require_annotation": false,
    "kiosk_mode": false,
    "kiosk_screen": "status/dashboard",
    "roles": ["basic"],
    "exceptions": {},
    "preferences": {
      "href": "/api/user/v1/users/larry/preferences",
      "locale": "C",
      "login_screen": "status/dashboard",
      "session_timeout": 15,
      "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 larry:letmein -3 -s -k -i -X PUT \
-H "Content-Type: application/json" -d @- \
https://tanana:215/api/appliance/v1/users/larry/preferences <<JSON
```

```
> {"session_timeout":60}
> JSON
HTTP/1.1 202 Accepted
Date: Wed, 24 Jul 2013 05:43:17 GMT
X-Zfs-Sa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 0

{
  "preferences": {
    "href": "appliance/v1/users/larry/preferences",
    "locale": "C",
    "login_screen": "status/dashboard",
    "session_timeout": 60,
    "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://tanana:215/api/appliance/v1/users/jschwartz
```

```
HTTP/1.1 204 No Content
Date: Tue, 23 Jul 2013 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", "letmein")
>>> 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
できている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://zfssa: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": "0.0.0.0",
      "family": "IPv4",
      "gateway": "10.80.231.1",
      "href":
"/api/network/v1/routes/ixgbe0,0.0.0.0,10.80.231.1",
      "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://zfssa: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://zfssa: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://zfssa:215/api/alert/v1/actions/actions-001")
>>> request.get_method = lambda: 'DELETE'
>>> response = opener.open(request)
>>> response.getcode()
204
```