



Red Hat Satellite 6.4

API ガイド

Red Hat Satellite の Representational State Transfer (REST) API 使用ガイド

Red Hat Satellite 6.4 API ガイド

Red Hat Satellite の Representational State Transfer (REST) API 使用ガイド

Red Hat Satellite Documentation Team
satellite-doc-list@redhat.com

法律上の通知

Copyright © 2017 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

Red Hat Satellite 6.4 ベータ版 Representational State Transfer (REST) API ガイドでは、REST API のコンセプトの説明と、さまざまな種類の要求に関する使用例を紹介します。本書では、管理者および開発者がカスタムのスクリプトを記述して Red Hat Satellite とサードパーティのアプリケーションを統合するための基本事項を説明します。

目次

第1章 RED HAT SATELLITE API について	3
1.1. RED HAT SATELLITE API	3
1.2. REPRESENTATIONAL STATE TRANSFER	3
1.3. HAMMER CLI ツールと SATELLITE API の比較	4
第2章 API リファレンス	5
2.1. API 構文の理解	5
HTTP 動詞 (GET) の使用	5
HTTP 動詞 (POST) の使用	6
HTTP 動詞 (PUT) の使用	8
HTTP 動詞 (DELETE) の使用	9
API エラーメッセージと API リファレンスの関連付け	9
2.2. JSON 応答形式の理解	9
コレクション用の JSON 応答形式	10
単一オブジェクトの JSON 応答形式	11
第3章 API 呼び出しの認証	12
3.1. SSL 認証の使用	12
3.2. HTTP 認証の使用	13
3.3. OAUTH 認証の使用	13
3.3.1. OAuth の認証	14
3.3.2. OAuth 要求の作成	14
第4章 RED HAT SATELLITE API の使用開始	15
4.1. CURL を使用した API の例	15
4.1.1. シンプルなクエリーの実行	15
4.1.2. リソースの作成および変更	18
4.1.2.1. Satellite サーバーへのコンテンツのアップロード	20
4.1.3. スマートクラスのオーバーライド	23
4.1.3.1. 外部ファイルを使用したスマートクラスパラメーターの変更	24
4.1.4. エラータのホストまたはホストコレクションへの適用	28
4.2. RUBY を使用した API の例	29
4.2.1. Ruby を使用したオブジェクトの作成	29
4.2.2. Apipie バインディングの使用	30
4.3. PYTHON を使用した API の例	32
4.3.1. Python を使用したオブジェクトの作成	32
4.3.2. Python を使用したクエリーの実行	34
4.4. 詳細検索の使用	36
4.5. ページネーション制御のある検索の使用	36
4.6. ライフサイクル環境との作業	37
ライフサイクル環境の更新	40
ライフサイクル環境の削除	40
付録A API レスポンスコード	41
付録B API パーMISSIONの表	42

第1章 RED HAT SATELLITE API について

Red Hat Satellite は、物理環境、仮想環境、およびクラウド環境でシステムをデプロイ、設定、および保守することを可能にするシステム管理ソリューションです。Satellite は、一元化された単一のツールにより、プロビジョニング、リモート管理、および複数の Red Hat Enterprise Linux デプロイメントの監視を可能にします。Red Hat Satellite サーバーは、Red Hat カスタマーポータルからのコンテンツを同期し、詳細なライフサイクル管理、ユーザーおよびグループロールベースアクセス制御、統合サブスクリプション管理、高度な GUI、CLI、または API アクセスなどの機能を提供します。

1.1. RED HAT SATELLITE API

Red Hat Satellite は *Representational State Transfer (REST)* API を提供します。API により、開発者やシステム管理者は、標準の Web インターフェース以外で Red Hat Satellite 環境を制御できるようになります。REST API は、Red Hat Satellite の機能と、HTTP 経由で API にアクセスする外部アプリケーションやカスタムスクリプトと統合することを目的とする開発者や管理者には便利です。



注記

完全な API リファレンスは <https://satellite6.example.com/apidoc/v2.html> (satellite6.example.com は Satellite サーバーのホスト名に置き換えてください) の Satellite サーバーからご利用いただけます。提供されている Satellite 6 API のバージョンは 1 と 2 ですが、Red Hat がサポートしているのはバージョン 2 のみである点に注意してください。

REST API には、以下のようなメリットがあります。

- **幅広いクライアントサポート:** HTTP 対応のあらゆるプログラミング言語/フレームワーク/システムで使用することが可能です。
- **自己記述型:** 詳しい情報の多くは実行時に確認できるため、クライアントアプリケーションに必要な仮想化インフラストラクチャーの知識は最小限で済みます。
- **リソースベースのモデル:** リソースベースの REST モデルにより仮想化プラットフォームを自然な形で管理することが可能です。

これにより、開発者および管理者は以下のような作業を行うことができます。

- エンタープライズ IT システムとの統合
- サードパーティー製のアプリケーションとの統合
- 自動メンテナンスやエラーチェックなどのタスクの実行
- スクリプトによる反復タスクの自動化

1.2. REPRESENTATIONAL STATE TRANSFER

Representational State Transfer (REST) は、特定のサービスとその表現にフォーカスした設計アーキテクチャーです。リソース表現は、サーバー上にある 1 つの特定管理対象要素に対応した情報の主要な抽象概念です。クライアントは Uniform Resource Identifier (URI) にあるサーバー要素に対して要求を送信し、**GET**、**POST**、**PUT**、**DELETE** などの HTTP の標準メソッドで操作を実行します。これにより、クライアントとサーバー間のステートレスな通信が提供され、各要求は他の要求に依存せずに機能し、その要求を完了するために必要な情報をすべて含みます。

1.3. HAMMER CLI ツールと SATELLITE API の比較

多くのタスクで、Hammer も Satellite API も同等に利用可能です。Hammer は、スクリプトに適用する前に API の呼び出しの応答をテストするなど、Satellite API よりも使いやすいインターフェースとして利用できます (`hammer -d organization list` など、Hammer で発行した API の呼び出しを検査するには `-d` オプションを使用します)。API での変更は自動的に Hammer に適用されますが、API を直接使用するスクリプトは手動で更新する必要があります。

バックグラウンドで、各 Hammer コマンドは最初に API へのバインドを確立し、要求を送信します。大量の Hammer コマンドを順番に実行する場合には、パフォーマンスに影響が出る可能性があります。反対に、API で直接通信するスクリプトではバインドを確立するのは 1 度だけです。詳しい情報は、『[Hammer CLI ガイド](#)』を参照してください。

第2章 API リファレンス

完全な API リファレンスは <https://satellite6.example.com/apidoc/v2.html> (satellite6.example.com は Satellite サーバーのホスト名に置き換えてください) の Satellite サーバーからご利用いただけます。提供されている Satellite 6 API のバージョンは 1 と 2 ですが、Red Hat がサポートしているのはバージョン 2 のみである点に注意してください。

2.1. API 構文の理解

同梱の API リファレンスでは、

```
HTTP_VERB API_ROUTE
```

などのように、HTTP の動詞の後に API ルートまたはパスが続く形式となっています。API が使用する HTTP の動詞は、GET、POST、PUT、DELETE です。例については、<http://satellite6.example.com/apidoc/v2/hosts.html> の API リファレンスドキュメントの HOST セクションを参照してください。API 構文および `curl` コマンドに精通している場合にはこの章を飛ばして次に進んでください。

API を利用するには、参照ドキュメントからの API ルートと、コマンドのドキュメントからのコマンド構文を使用してコマンドを構築します。たとえば `curl` の man ページでは

```
curl [options] [URL...]
```

などの基本構文が紹介されています。このガイドで使用するオプションには `-X`、`--request command` などが含まれます。`command` は HTTP の動詞に置き換えてください。

HTTP 動詞 (GET) の使用

HTTP 動詞の GET は、既存のエントリまたはリソースに関して API からデータを取得するのに使用します。

`GET /api/hosts` など、API HOST セクションの例と、`curl` 構文を組み合わせると、

```
curl -X GET https://satellite6.example.com/api/hosts
```

といったコマンドになります。Satellite は、API への接続には HTTPS のみをサポートし、一定の形式の認証が必要です。

使用可能な例として、最低でも `-u` オプションで名前を、`-k` オプションで SSL のピア証明書検証を省略する必要があります。

```
$ curl -X GET -k -u sat_username https://satellite6.example.com/api/hosts
Enter host password for user 'sat_username':
{
  "total": 2,
  "subtotal": 2,
  "page": 1,
  "per_page": 20,
  "search": null,
  "sort": {
    "by": null,
    "order": null
  },
  "results":
出力省略
```

API からの上記の応答により合計の結果数が 2 つで、以下のような 2 つの結果が返されたこと、またこれは結果の 1 ページ目で、ページごとの最大結果数は 20 に設定されていることが分かります。これについては「[JSON 応答形式の理解](#)」でさらに詳しく説明します。

API リファレンスの例には、**:parameter** などのように、コロンが先頭にきてその後に用語が含まれるというものがあります。以下に例を示します。

```
GET /api/hosts/:id
```

これらは API ルートパラメーターで、適切な値に置き換える必要があります。ルートパラメーターは、コロンで始まり **id** で終わります。



注記

Satellite 6 では、API のバージョン 2 がデフォルトであるため、API 呼び出しの URL に **v2** を使用する必要はありません。

HTTP 動詞 (POST) の使用

HTTP 動詞の POST は、API にデータを送信して新規エントリーまたはリソースを作成するのに使用します。データは JSON 形式で記述し、**-d**, **--data** オプションの後に、**{}** カッコで閉じた JSON 形式の引用データを指定します。引用なしの JSON 形式のデータをファイルに指定することもできますが、その際は **curl** コマンドの **@** オプションで指定します (例: **-d @file.json**)。

JSON 形式のデータに外部ファイルを使用する利点として、引用やエスケープの問題が減り、構文チェッカーの付いた任意のエディターを使用してエラーを特定し、回避することができ、JSON データの妥当性を確認して再フォーマットする外部ツールを使用できる点が挙げられます。たとえば、yajl パッケージでは **json_verify** ツールと **json_reformat** ツールが提供されています。

json_verify ツールを使用すると、以下のように JSON ファイルの妥当性を確認することができます。

```
$ json_verify < test_file.json
```

API 呼び出しで返された非構造化 JSON データは、Python モジュール **json.tool** で、

```
curl API_call | python -m json.tool
```

のようにパイプすることができます。または、

```
curl API_call | json_reformat
```

のように、**json_reformat** ツールを使用してください。出力形式は「[JSON 応答形式の理解](#)」で説明しています。

API リファレンスには、**Activation keys** セクションの

```
POST /katello/api/activation_keys
```

が含まれます。

これは、**POST /katello/api/activation_keys** コマンドの形式の一例です。

```
curl -X POST -k -u sat_username:sat_password \  
-d @file_of_json-formatted_data \  
https://satellite6.example.com/katello/api/activation_keys
```

HTTP 動詞の POST がどのように機能するかを確認するには、以下のような内容が含まれる **activation-key.json** などのテストファイルを作成します。

```
{"organization_id":1, "name":"TestKey", "description":"Just for testing"}
```

以下の例では、先程作成したファイルのデータを適用して、新規アクティベーションキーを作成します。

```
$ curl -H "Accept:application/json,version=2" \  
-H "Content-Type:application/json" -X POST \  
-u sat_username:sat_password -k \  
-d @activation-key.json \  
https://satellite6.example.com/katello/api/activation_keys | json_reformat  
{  
  "id": 2,  
  "name": "TestKey",  
  "description": "Just for testing",  
  "unlimited_hosts": true,  
  "auto_attach": true,  
  "content_view_id": null,  
  "environment_id": null,  
  "usage_count": 0,  
  "user_id": 3,  
  "max_hosts": null,  
  "release_version": null,  
  "service_level": null,  
  "content_overrides": [  
  
  ],  
  "organization": {  
    "name": "Default Organization",  
    "label": "Default_Organization",  
    "id": 1  
  },  
  "created_at": "2017-02-16 12:37:47 UTC",  
  "updated_at": "2017-02-16 12:37:48 UTC",  
  "content_view": null,  
  "environment": null,  
  "products": null,  
  "host_collections": [  
  
  ],  
  "permissions": {  
    "view_activation_keys": true,  
    "edit_activation_keys": true,  
    "destroy_activation_keys": true  
  }  
}
```

Web UI でこのエントリーを表示するには、**コンテンツ → アクティベーションキー** の順に移動します。変更を加えた後は、ページを再読込するのを忘れないでください。

HTTP 動詞 (PUT) の使用

HTTP 動詞の PUT は、API にデータを送信して既存のエントリまたはリソースを更新するのに使用します。POST API 呼び出しと同様に、データは JSON 形式で記述し、**-d**, **--data** オプションの後に、**{ }** カッコで閉じた JSON 形式の引用データを指定します。引用なしの JSON 形式のデータをファイルに指定することもできますが、その際は **curl** コマンドの **@** オプションで指定します (例: **-d @file.json**)。

既存の値を変更するか、既存のリソースに値を追加するには、HTTP 動詞の PUT を使用します。API リファレンスには、

```
PUT /katello/api/activation_keys/:id
```

のように、アクティベーションキーを更新するエントリがあります。

既存のアクティベーションキーを更新するには、以下の形式のコマンドを使用します。

```
curl -X PUT -k -u sat_username:sat_password \  
-d @file_of_json-formatted_data \  
https://satellite6.example.com/katello/api/activation_keys/:id
```

:id は、更新するアクティベーションキーの ID に置き換えます。同じ値で PUT コマンドを複数回使用しても、エントリは複数 **作成されません**。

たとえば、以前の例で作成したテスト用のアクティベーションキーを更新するには、以下のように以前作成したファイルを編集します。

```
{"organization_id":1, "name":"TestKey", "description":"Just for testing", "max_hosts":"10" }
```

以下のコマンドを使用して、JSON ファイルに変更を適用します。

```
$ curl -H "Accept:application/json,version=2" \  
-H "Content-Type:application/json" -X PUT \  
-u sat_username:sat_password -k \  
-d @activation-key.json \  
https://satellite6.example.com/katello/api/activation_keys/2  
{  
  "id": 2,  
  "name": "TestKey",  
  "description": "Just for testing",  
  "unlimited_hosts": false,  
  "auto_attach": true,  
  "content_view_id": null,  
  "environment_id": null,  
  "usage_count": 0,  
  "user_id": 3,  
  "max_hosts": 10,  
  "release_version": null,  
  "service_level": null,  
  "content_overrides": [  
  
  ],  
  "organization": {  
    "name": "Default Organization",  
    "label": "Default_Organization",
```

```

      "id": 1
    },
    "created_at": "2017-02-16 12:37:47 UTC",
    "updated_at": "2017-02-16 12:46:17 UTC",
    "content_view": null,
    "environment": null,
    "products": null,
    "host_collections": [

  ],
  "permissions": {
    "view_activation_keys": true,
    "edit_activation_keys": true,
    "destroy_activation_keys": true
  }
}

```

HTTP 動詞 (DELETE) の使用

リソースを削除するには、削除するリソース ID を含む API ルートと DELETE 動詞を使用します。

既存のアクティベーションキーを削除するには、以下の形式のコマンドを使用します。

```

curl -X DELETE -k -u sat_username:sat_password \
https://satellite6.example.com/katello/api/activation_keys/:id

```

:id は削除するアクティベーションキーの ID に置き換えます。以下に例を示します。

```

$ curl -H "Accept:application/json,version=2" \
-H "Content-Type:application/json" -X DELETE \
-u admin:RedHat1! -k \
https://satellite6.example.com/katello/api/activation_keys/2 |
json_reformat出力省略
  "started_at": "2017-02-16 12:58:17 UTC",
  "ended_at": "2017-02-16 12:58:18 UTC",
  "state": "stopped",
  "result": "success",
  "progress": 1.0,
  "input": {
    "activation_key": {
      "id": 2,
      "name": "TestKey"出力省略
    }
  }

```

API エラーメッセージと API リファレンスの関連付け

API はエラーの表示に RAILS 形式を使用します。

```

Nested_Resource.Attribute_Name

```

これは、API リファレンスで使用する以下の形式に変換されます。

```

Resource[Nested_Resource_attributes][Attribute_Name_id]

```

2.2. JSON 応答形式の理解

GET を使用して API に呼び出しを行うと、JSON 形式で結果が返されます。Python `json.tool` モジュールにかけると、人間がより判読しやすい形式で出力されます。

コレクション用の JSON 応答形式

コレクションとは、ホストやドメインなどのオブジェクトの一覧のことです。コレクションの JSON 応答は、メタデータのフィールドの後に結果セクションが表示される形式です。以下の例は、API ルート `GET /api/domains` を使用した場合のドメイン一覧のコレクション JSON 応答形式です。出力は、結果セクションがより簡単に確認できるように `json.tool` でパイプしています。

```
$ curl -X GET -k -u admin:password
https://satellite6.example.com/api/domains | python -m json.tool
{
  "total": 3,
  "subtotal": 3,
  "page": 1,
  "per_page": 20,
  "search": null,
  "sort": {
    "by": null,
    "order": null
  },
  "results": [
    {
      "id": 23,
      "name": "qa.lab.example.com",
      "fullname": "QA",
      "dns_id": 10,
      "created_at": "2013-08-13T09:02:31Z",
      "updated_at": "2013-08-13T09:02:31Z"
    },
    {
      "id": 25,
      "name": "sat.lab.example.com",
      "fullname": "SATLAB",
      "dns_id": 8,
      "created_at": "2013-08-13T08:32:48Z",
      "updated_at": "2013-08-14T07:04:03Z"
    },
    {
      "id": 32,
      "name": "hr.lab.example.com",
      "fullname": "HR",
      "dns_id": 8,
      "created_at": "2013-08-16T08:32:48Z",
      "updated_at": "2013-08-16T07:04:03Z"
    }
  ]
}
```

応答のメタデータのフィールドの説明は以下のとおりです。

- **total** — 検索パラメーターなしのオブジェクトの合計数
- **subtotal** — 検索パラメーターを指定して返されたオブジェクト数 (検索がない場合には、累計は合計と同じになります)

- **page** — ページ数
- **per_page** — ページごとに返す最大数
- **limit** — コレクションの応答で返すオブジェクトの指定数
- **offset** — コレクションを返す前に省略するオブジェクト数
- **search** — **scoped_scoped** の構文をベースにした検索文字列
- **sort**
 - **by** — コレクションをソートするためのフィールド
 - **order** — ソート順 (ASC は昇順、DESC は降順)
- **results** — オブジェクトのコレクション

単一オブジェクトの JSON 応答形式

単一オブジェクトの JSON 応答を使用して、単一オブジェクトを表示します。GET 要求には、オブジェクトの一意識別子 **:id** または **:name** が必要です。**:name** は一意識別子として常に使用できるわけではありませんが、**:id** は常に使用できる点に注意してください。単一オブジェクトの JSON 応答形式には、オブジェクトの属性のみが含まれます。

以下の例は、API ルートの **GET /api/domains/23** または **GET /api/domains/qa.lab.example.com** を使用時の単一オブジェクト JSON 応答形式です。

```
$ curl -X GET -k -u admin:password
https://satellite6.example.com/api/domains/23 | python -m json.tool
{
  "id": 23,
  "name": "qa.lab.example.com",
  "fullname": "QA",
  "dns_id": 10,
  "created_at": "2013-08-13T09:02:31Z",
  "updated_at": "2013-08-13T09:02:31Z"
}
```

第3章 API 呼び出しの認証

Satellite API と対話するには、認証が必要です。各 API 要求で使用するためにローカルホストに Satellite サーバー CA 証明書をダウンロードして、SSL 認証を提供することができます。各 API 要求には、有効なユーザー名とパスワードが必要です。以下のセクションで、これらの点について説明します。

3.1. SSL 認証の使用

Red Hat Satellite は HTTPS を使用して、Red Hat Satellite サーバーと通信する際に一定の暗号化レベルを提供し、ID の確認を行います。SSL 以外の通信は、Satellite 6 ではサポートしていません。

Red Hat Satellite サーバーはそれぞれ、自己署名の証明書を使用します。この証明書は、暗号化キーを検証するサーバーの証明書と、Satellite サーバーの ID を信頼する証明局 (CA) の両方の役割を果たします。以下の手順は、Satellite サーバーの SSL 認証を設定する方法です (以下の例では `satellite6.example.com`)。

1. 以下のオプションの 1 つを使用して、通信する Satellite サーバー (`satellite6.example.com`) からの証明書を取得します。

- a. SSH を使用して証明書を取得するには、以下のコマンドを実行します。

```
$ scp root@satellite6.example.com:/var/www/html/pub/katello-server-ca.crt ./
```

- b. Satellite サーバーで直接コマンドを実行する場合は、以下のコマンドを実行してローカルで利用可能なコピーから証明書を取得します。

```
$ cp /var/www/html/pub/katello-server-ca.crt ./
```

- c. HTTP を使用して証明書を取得するには、以下のコマンドを実行します。

```
$ curl -O http://satellite6.example.com/pub/katello-server-ca.crt
```



警告

暗号化されていない HTTP 接続を使用して証明書を取得すると、セキュリティリスクにさらされる可能性があります。

2. クライアント上の証明書を証明局として使用して、Satellite サーバーの ID を検証します。

```
$ curl -X GET -u sat_username:sat_password \  
-H "Accept:application/json" --cacert katello-server-ca.crt \  
https://satellite6.example.com/katello/api/organizations
```

GET はデフォルトのアクションであるため、**-X GET** 属性はここでは省略可能です。

3. Network Security Services (NSS) データベースを作成して証明書を保存します。


```
$ certutil -N -d sql:$HOME/.pki/nssdb
Enter a password which will be used to encrypt your keys.
The password should be at least 8 characters long,
and should contain at least one non-alphabetic character.

Enter new password:
Re-enter password:
```

NSS データベースがすでに存在する場合には、以下のようにパスワードが求められます。

```
Enter Password or Pin for "NSS Certificate DB":
```

4. 以下のコマンドを使用して、NSS データベースに永続的に証明書を追加します。

```
$ certutil -d sql:$HOME/.pki/nssdb -A -t TC -n "Red Hat Satellite" \
-i katello-server-ca.crt
```

これで、NSS データベースに証明書がインポートされ、要求ごとに **--cacert** オプションを指定する必要がなくなります。これは、以下のようにテストできます。

```
$ curl -X GET -u sat_username:sat_password
https://satellite6.example.com/api/v2/hosts
{
  "total": 2,
  ...,
  "results": [
    ...
  ]
} 出力省略
```

3.2. HTTP 認証の使用

Satellite API への要求にはすべて、適切なユーザー名とパスワードが必要です。API は HTTP Basic 認証 (基本認証) ^[1] を使用してこれらの認証情報を暗号化して、認証ヘッダーに追加します。要求に適切な認証ヘッダーが含まれない場合には、API は 401 Authorization Required エラーを返します。

重要

Basic 認証 (基本認証) では、パスワードなどの潜在的な機密情報がプレーンテキストで送信されてしまいます。REST API には、プレーンテキストの要求をトランスポートレベルで暗号化する HTTPS が必要です。

base64 ライブラリーによっては、暗号化された認証情報を複数行に分け、行末には改行文字を付けることがあります。そのような場合には、ヘッダーが破損し、要求に問題が発生します。認証ヘッダーには、エンコードされた認証情報がヘッダー内に一行で記載されている必要があります。

3.3. OAUTH 認証の使用

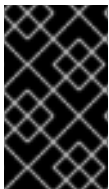
Basic 認証の代わりに、制約はありますが API で OAuth 1.0 認証をサポートしています (プロトコルのバージョン 1.0a の 1-legged OAuth と呼ばれることもあります)。

3.3.1. OAuth の認証

Satellite 6.4 ベータでは、OAuth は デフォルトで有効になっています。構成設定は `/etc/foreman/settings.yaml` の設定ファイルに保存されており、**管理** → **設定** → **Auth** の順に移動して Web UI から表示できます。**OAuth コンシューマーキー** とは、すべての OAuth クライアントが使用するトークンのことです。Web UI の特定の値は変更できません。これらの値は **satellite-installer** スクリプトに新しいオプションを指定しなおして実行しなければ変更できません。ファイルを手動で変更した内容はアップグレード時に失われてしまう点に注意してください。以下のコマンドを実行して、OAuth 関連のインストーラーオプションをすべて表示します。

```
# satellite-installer --full-help | grep oauth
```

OAuth を使用して行ったすべての API 要求を組み込みの匿名 API 管理者アカウントで認証する場合には、`/etc/foreman/settings.yaml` の **OAuth マップユーザー** を **false** に設定します。要求を行ったユーザーを指定するには、この設定オプションを **true** に設定します。これにより、クライアントは FOREMAN-USER ヘッダーに既存の Foreman ユーザーのログインを指定して送信することができます。



重要

ヘッダーは OAuth 要求で署名されないため、偽造されてしまう可能性があります。有効なコンシューマーキーのあるユーザーは、Foreman ユーザーになります。

3.3.2. OAuth 要求の作成

通常 OAuth クライアントライブラリーは要求の生成に使用します。ここでは **curl** を使用した OAuth 要求がどのように機能するか理解いただけるように一例を紹介しています。

例3.1 curl を使用した OAuth 要求の例

```
$ curl 'https://satellite6.example.com/api/architectures' \
-H 'Content-Type: application/json' \
-H 'Accept: application/json,version=2' \
-H 'FOREMAN-USER: User1' \
-H 'Authorization: OAuth
oauth_version="1.0",oauth_consumer_key="secretkey",oauth_signature_method="hmac-sha1",oauth_timestamp=1321473112,oauth_signature=Il8hR8/ogj/XVu0qMPB9qNjSy6E='
```

上記の例では、アーキテクチャーが OAuth を使用してリストされ、認証されます。この要求は、*User1* のユーザーとしてログインを試行します。マッピングが Foreman の設定で有効な場合、結果にはユーザー *User1* のみに表示可能なアーキテクチャーのみが含まれます。署名が手動で作成された場合には、**oauth_timestamp** を変更すると変更されるはずですが、また、署名はすべてのパラメーター、HTTP メソッド、URI の変更にも反映されます。そのため、OAuth クライアントライブラリーを使用して OAuth パラメーターを構築することを推奨します。

[1] Basic 認証については [RFC 2617 HTTP Authentication: Basic and Digest Access Authentication](#) に説明されています。

第4章 RED HAT SATELLITE API の使用開始

本章では、Red Hat Satellite API を使用して異なるタスクを実行する方法を例をあげて紹介します。以下の例では、ポート 43 の HTTPS を使用した Satellite サーバーにフォーカスします。Satellite Capsule 経由で API にアクセスすることもできますが、ポート 8443 を使用する必要があり、使用しない場合には API の呼び出しに失敗します。

スクリプト自体で、異なるポートの要件に対応することができます。たとえば、Ruby では Satellite および Capsule の URL を以下のように指定することができます。

```
url = 'https://satellite6.example.com/api/v2/'
capsule_url = 'https://capsule.example.com:8443/api/v2/'
katello_url = 'https://satellite6.example.com/katello/api/v2/'
```

ホストが Satellite サーバーまたは Capsule サーバーにサブスクライブしている場合には、`/etc/rhsm/rhsm.conf` ファイルにある `[server]` セクションの `port` エントリーをもとに、API にアクセスする際に必要な、正しいポートを判断することができます。これらの値を使用して、スクリプトを完全に自動化して、使用するポートを検証する必要性をなくすことができます。

4.1. CURL を使用した API の例

以下のセクションでは `curl` を使用して Satellite API でさまざまなタスクを実行する方法について説明します。

4.1.1. シンプルなクエリーの実行

以下の例では `curl` を使用して Satellite デプロイメントの情報を検索する方法を説明します。以下の例には、実際のコマンドと、サンプルの出力、ユーザー名やパスワードの値の例などが含まれます。デプロイメントごとに結果は異なります。また、以下の例では `python -m json.tool` コマンドを使用して出力をフォーマットしています。

注記

Red Hat Satellite は HTTPS と、デフォルトではホストの識別用に証明書を使用する必要があります。「[SSL 認証の使用](#)」の記載どおりに Satellite サーバーの証明書を追加していない場合には、`-k` (セキュアではありません) オプションを使用して証明書チェックを省略することができます。

ユーザー認証に、`-u username:password` の形式を使用するか、パスワードを追加していない場合にはコマンドでパスワードの入力が求められます。コマンドの一部としてパスワードを追加すると shell の履歴に残り、セキュリティリスクとなる可能性があるため、Red Hat ではこれは推奨していません。以下の例では、単純化の目的としてのみ、パスワードをコマンドに含めています。

`curl` で `-s` (サイレント) オプションを使用する場合は、進捗バーやエラーメッセージが表示されないので注意してください。

リソース一覧の取得

以下は、リソースの一覧を返す基本的なクエリーです。このような要求は、メタデータでラップされたデータ一覧を返しますが、他の要求タイプでは実際のオブジェクトのみを返します。

```
$ curl -X GET -s -k -u sat_username:sat_password
https://satellite6.example.com/api/v2/hosts | python -m json.tool
```

```
{
  "total" => 2,
  "subtotal" => 2,
  "page" => 1,
  "per_page" => 1000,
  "search" => nil,
  "sort" => {
    "by" => nil,
    "order" => nil
  },
  "results" => [
    ...
  ]
}
```

例4.1 ユーザーの一覧表示

```
$ curl -X GET -s -k -u sat_username:sat_password
https://satellite6.example.com/api/users
{
  "total": 1,
  "subtotal": 1,
  "page": 1,
  "per_page": 20,
  "search": null,
  "sort": {
    "by": null,
    "order": null
  },
  "results":
  [{"firstname":"Admin","lastname":"User","mail":"root@example.com","admin
  ":true,"auth_source_id":1,"auth_source_name":"Internal","timezone":null,
  "locale":null,"last_login_on":"2017-02-08 23:25:51
  UTC","created_at":"2017-01-09 12:10:02 UTC","updated_at":"2017-02-08
  23:25:51
  UTC","id":3,"login":"admin","default_location":null,"locations":
  [],"default_organization":{"id":1,"name":"Default
  Organization","title":"Default
  Organization","description":null},"organizations":[]}]
}
```

一般的なホストクエリーの実行

以下のクエリーは、ホスト **satellite6.example.com** の情報を返します。

```
$ curl -X GET -s -k -u sat_username:sat_password
https://satellite6.example.com/api/v2/hosts/satellite6.example.com |
python -m json.tool
{
  "all_puppetclasses": [],
  "architecture_id": 1,
  "architecture_name": "x86_64",
  "build": false,
  "capabilities": [
```

```

        "build"
    ],
    "certname": "satellite6.example.com",
    "comment": null,
    "compute_profile_id": null,
    ...
}

```

特定のホストのファクト検索

以下のクエリーは、ホスト **satellite6.example.com** の全ファクトを返します。

```

$ curl -X GET -s -k -u sat_username:sat_password
https://satellite6.example.com/api/v2/hosts/satellite6.example.com/facts |
python -m json.tool
{
  ...
  "results": {
    "satellite6.example.com": {
      "augeasversion": "1.0.0",
      "bios_release_date": "01/01/2007",
      "bios_version": "0.5.1",
      "blockdevice_sr0_size": "1073741312",
      "facterversion": "1.7.6",
      ...
    }
  }
}

```

一致するパターンがあるか全ホストの検索

以下のクエリーは、「example」というパターンと一致するホストをすべて返します。

```

$ curl -X GET -s -k -u sat_username:sat_password
https://satellite6.example.com/api/v2/hosts?search=example | python -m
json.tool
{
  ...
  "results": [
    {
      "name": "satellite6.example.com",
      ...
    }
  ],
  "search": "example",
  ...
}

```

特定の環境内の全ホスト検索

以下のクエリーは、「production」環境内の全ホストを返します。

```

$ curl -X GET -s -k -u sat_username:sat_password
https://satellite6.example.com/api/v2/hosts?search=environment=production
| python -m json.tool
{
  ...
  "results": [

```

```
{
  {
    "environment_name": "production",
    "name": "satellite6.example.com",
    ...
  }
],
"search": "environment=production",
...
}
```

特定のファクト値を持つ全ホストの検索

以下のクエリーでは、「RHEV Hypervisor」というモデル名を持つホストがすべて返されます。

```
$ curl -X GET -s -k -u sat_username:sat_password
https://satellite6.example.com/api/v2/hosts?
search=model=\"RHEV+Hypervisor\" | python -m json.tool
{
  ...
  "results": [
    {
      "model_id": 1,
      "model_name": "RHEV Hypervisor",
      "name": "satellite6.example.com",
      ...
    }
  ],
  "search": "model=\"RHEV Hypervisor\"",
  ...
}
```

ホストの削除

以下のクエリーでは、*host1.example.com* という名前のホストが削除されます。

```
curl -k -u sat_username:sat_password -X DELETE
https://satellite6.example.com/api/v2/hosts/host1.example.com
```

4.1.2. リソースの作成および変更

Satellite API を使用して Satellite サーバーのリソースを操作することができます。これらの API 呼び出しでは、ユーザー名、パスワード、URI 以外のさまざまなクエリーパラメーターを渡す必要があります。たとえば、Satellite サーバーにコンテンツをアップロードしたり、Satellite リソースを変更したりするには、要求の作成時にヘッダーに追加情報を含める必要があります。

以下の例のように API のバージョンをヘッダーに指定するか、URL の一部として指定します。たとえば、*https://satellite6.example.com/api/v2/architectures* は要求ヘッダーで **Accept: version=2** を使用するのと同じです。URL の指定が優先されます。

以下は、POST 要求の基本構文です。

```
$ curl -H "Accept:application/json,version=2" \
-H "Content-Type:application/json" -X POST \
-u username:password -k \
-d json-formatted-data https://satellite6.example.com
```

たとえば、新規アーキテクチャーを作成するには、以下の例のような要求を使用することができます。

```
$ curl -H "Accept:application/json,version=2" \
      -H "Content-Type:application/json" -X POST -u
sat_username:sat_password \
      -k -d "{\"architecture\":{\"name\":\"i686\"}}\" \
      https://satellite6.example.com/api/architectures
```

これにより、以下のような出力が表示されます。

```
{"name":"i686","id":3,"created_at":"2015-10-
29T13:21:09Z","updated_at":"2015-10-29T13:21:09Z","operatingsystems":
[],"images":[]}
```

以下のコマンドを使用して、作成したアーキテクチャーを検証します。

```
$ curl -X GET -u sat_username:sat_password -k
https://satellite6.example.com/api/v2/architectures | python -m json.tool
{
  "page": 1,
  "per_page": 20,
  "results": [
    {
      "created_at": "2015-04-02T05:29:46Z",
      "id": 2,
      "name": "i386",
      "updated_at": "2015-04-02T05:29:46Z"
    },
    {
      "created_at": "2015-04-02T05:29:46Z",
      "id": 1,
      "name": "x86_64",
      "updated_at": "2015-04-02T05:29:46Z"
    },
    {
      "created_at": "2015-11-04T19:40:15Z",
      "id": 3,
      "name": "i686",
      "updated_at": "2015-11-04T19:40:15Z"
    }
  ],
  "search": null,
  "sort": {
    "by": null,
    "order": null
  },
  "subtotal": 3,
  "total": 3
}
```

Satellite サーバーの **hammer** を使用して結果を検証することも可能です。

```
$ hammer -u sat_username -p sat_password architecture list
---|-----
ID | NAME
```

```

---|-----
2 | i386
1 | x86_64
3 | i686
---|-----

```

例4.2 新しいユーザーの作成

```

$ curl -H "Accept:application/json,version=2" \
-H "Content-Type:application/json" -X POST \
-u sat_username:sat_password -k \
-d '{"firstname":"Test","lastname":"API-
User","mail":"test@example.com","login":"test_api","password"
:"123456","auth_source_id":1}' \
https://satellite6.example.com/api/users

```

4.1.2.1. Satellite サーバーへのコンテンツのアップロード

以下のセクションでは、Satellite 6 API で `curl` を使用して、Satellite サーバーに大容量ファイルをアップロードしてインポートする方法を説明します。このプロセスには 4 つの手順が含まれます。

1. アップロード要求を作成します。
2. コンテンツをアップロードします。
3. コンテンツをインポートします。
4. アップロード要求を削除します。

アップロード可能な最大ファイルサイズは 30 MB です。それよりも大きいサイズのコンテンツをアップロードするには、[例4.3「30 MB よりも大きいコンテンツのアップロード」](#)を参照してください。

手順4.1 Satellite サーバーへのコンテンツのアップロード

1. アップロード要求を作成します。デプロイメントに適したサンプルパラメーターを変更するようにしてください。

```

$ curl -H "Accept:application/json,version=2" \
-H "Content-Type:application/json" \
-X POST \
-u sat_username:sat_password -k -d "{}" \

https://satellite6.example.com/katello/api/repositories/3/content_uploads

```

このコマンドは、以下のような `upload_id` を返します。

```

{"upload_id":"0be156b1-f373-4cad-89d0-924f8f4491d2",
 "_href":"/pulp/api/v2/content/uploads/0be156b1-f373-4cad-89d0-924f8f4491d2/"}

```

2. コンテンツをアップロードします。データのアップロード時には、正しい MIME タイプを使用していることを確認します。Satellite 6 に対する要求にはほぼ、「application/json」の

MIME タイプが使用されます。**upload_id** と MIME タイプ、他のパラメーターを組み合わせ、コンテンツをアップロードします。

```
$ curl -H "Accept:application/json,version=2" \
      -H "Content-Type:multipart/form-data" \
      -X PUT \
      -u sat_username:sat_password \
      -k --data-urlencode
"content@/home/sat6user/rpmbuild/RPMS/noarch/python-scripttest-
1.1.1-1.fc21.noarch.rpm" \
      --data-urlencode offset=0 \

https://satellite6.example.com/katello/api/repositories/3/content_uploads/0be156b1-f373-4cad-89d0-924f8f4491d2
```

3. Satellite サーバーにコンテンツをアップロードした後に、適切なりポジトリにそのコンテンツをインポートする必要があります。この手順を完了するまで、Satellite サーバーではこの新しいコンテンツは認識されません。

```
$ curl -H "Accept:application/json,version=2" \
      -H "Content-Type:application/json" \
      -X PUT \
      -u sat_username:sat_password \
      -k -d "{\"upload_ids\": [\"0be156b1-f373-4cad-89d0-924f8f4491d2\"]}" \

https://satellite6.example.com/katello/api/repositories/3/import_uploads
```

4. コンテンツのアップロードおよびインポートが正常に完了したら、アップロード要求を削除することができます。削除することで、アップロード中に使用した一時的なディスク領域を解放することができます。

```
$ curl -H "Accept:application/json,version=2" \
      -H "Content-Type:application/json" \
      -X DELETE -d "{}" \
      -u sat_username:sat_password \
      -k

https://satellite6.example.com/katello/api/repositories/3/content_uploads/0be156b1-f373-4cad-89d0-924f8f4491d2
```

例4.3 30 MB よりも大きいコンテンツのアップロード

以下の例では、大容量のファイルを複数のチャンクに分割して、アップロード要求の作成、個別ファイルのアップロード、Satellite へのインポートを行ってから、アップロード要求を削除する方法を詳しく説明しています。この例は、サンプルのコンテンツ、ホスト名、ユーザー名、ファイル名を使用している点に注意してください。

1. サンプルモジュールをダウンロードします。

```
$ wget https://forgeapi.puppetlabs.com/v3/files/theforeman-foreman-5.0.1.tar.gz?_ga=1.267255502.1792403825.1430297670 -O theforeman-foreman-5.0.1.tar.gz
```

50,000 バイトのチャンクにモジュールを分割します。

```
$ split --bytes 50000 --numeric-suffixes --suffix-length=1
theforeman-foreman-5.0.1.tar.gz foreman_module.
```

結果ファイルを表示します。

```
$ ls -la theforeman-foreman-5.0.1.tar.gz foreman_module.*
-rw-r--r--. 1 root root 50000 Nov  4 04:42 foreman_module.0
-rw-r--r--. 1 root root 32928 Nov  4 04:42 foreman_module.1
-rw-r--r--. 1 root root 82928 Nov  4 04:41 theforeman-foreman-
5.0.1.tar.gz
```

2. 新規アップロード要求を作成します (これは Satellite サーバーでの **cat** と同じです)。

```
$ curl -H "Accept:application/json,version=2" \
-H "Content-Type:application/json" \
-X POST \
-u sat_username:sat_password -k -d "{}" \
https://ibm-
vm01.example.com/katello/api/repositories/2/content_uploads
```

上記のコマンドはアップロード ID を返します。

```
{"upload_id":"9585528f-07ad-4bb1-9c80-
ccece249b2b7","_href":"/pulp/api/v2/content/uploads/9585528f-07ad-
4bb1-9c80-ccece249b2b7/"}
```

3. 手順 1 で作成したファイルのチャンクをアップロードします。以下の例で **offset** パラメーターを使用して、ファイルサイズと関連付けている点に注意してください。

```
$ curl -H "Accept:application/json,version=2" \
-H "Content-Type:multipart/form-data" \
-X PUT \
-u sat_username:sat_password \
-k --data-urlencode "content@foreman_module.0" \
--data-urlencode offset=0 \
https://ibm-
vm01.example.com/katello/api/repositories/2/content_uploads/958552
8f-07ad-4bb1-9c80-ccece249b2b7
```

```
$ curl -H "Accept:application/json,version=2" \
-H "Content-Type:multipart/form-data" \
-X PUT \
-u sat_username:sat_password \
-k --data-urlencode "content@foreman_module.1" \
--data-urlencode offset=50000 \
https://ibm-
vm01.example.com/katello/api/repositories/2/content_uploads/958552
8f-07ad-4bb1-9c80-ccece249b2b7
```

4. 完全なアップロードをリポジトリにインポートします。

```
$ curl -H "Accept:application/json,version=2" \
-H "Content-Type:application/json" \
-X PUT \
-u sat_username:sat_password \
-k -d "{\"upload_ids\":[\"9585528f-07ad-4bb1-9c80-
ccece249b2b7\"]}" \
https://ibm-
vm01.example.com/katello/api/repositories/2/import_uploads
```

5. アップロード要求を削除します。

```
$ curl -H "Accept:application/json,version=2" \
-H "Content-Type:application/json" \
-X DELETE -d "{}" \
-u sat_username:sat_password \
-k https://ibm-
vm01.example.com/katello/api/repositories/2/content_uploads/958552
8f-07ad-4bb1-9c80-ccece249b2b7
```

6. Satellite サーバーにログインして、ファイルが正しく転送されたかどうかを確認します。

```
$ ls -la /var/lib/pulp/content/puppet_module/foreman-foreman-
5.0.1.tar.gz
-rw-r--r--. 1 apache apache 82928 Nov  4 04:55
/var/lib/pulp/content/puppet_module/foreman-foreman-
5.0.1.tar.gz
```

ファイルを比較します。

```
$ cmp /var/lib/pulp/content/puppet_module/foreman-foreman-
5.0.1.tar.gz foreman-foreman-5.0.1.tar.gz
```

```
$ echo $?
0
```

4.1.3. スマートクラスのオーバーライド

API を使用してスマートパラメーターを検索し、値を指定してクラスのスマートパラメーターをオーバーライドすることができます。変更可能な属性の完全一覧

は、https://satellite6.example.com/apidoc/v2/smart_class_parameters/update.html にある同梱の API リファレンスで確認できます。

たとえば、スマートクラスのパラメーターをすべて表示する場合、API ルートは **GET /api/smart_class_parameters** のようになります。**curl** を使用する場合は以下のようになります。

```
$ curl -X GET -s -k -u sat_username:sat_password \
https://satellite6.example.com/api/smart_class_parameters
```

5 など、Puppet クラス ID が分かる場合には、以下のように範囲を絞り込むことができます。

```
$ curl -X GET -s -k -u sat_username:sat_password
```

```
https://satellite6.example.com/api/puppetclasses/5/smart_class_parameters
```

いずれの呼び出しも検索パラメーターに対応しています。検索可能なフィールドの完全一覧は、Web UI の検索入力ボックスで確認できます。設定 → スマート変数 の順に移動して、検索クエリーボックスをクリックしてフィールドの一覧を表示します。

特に検索パラメーターで便利なのは、**puppetclass_name** と **key** の 2 つで、特定のパラメーターの検索が可能になります。たとえば、**-d** オプションと **--data** オプションを使用して URL のエンコードデータを渡すことができます。

```
$ curl -X GET -s -k -u sat_username:sat_password \
https://satellite6.example.com/api/smart_class_parameters -d \
'search=puppetclass_name = access_insights_client and key = authmethod'
```

標準のスコープ指定の検索構文はサポートされています。

パラメーターの ID を特定したら、現在のオーバーライド値など完全な詳細の表示に進むことができます。たとえば、ID が 63 の場合、API ルートは **GET /api/smart_class_parameters/63** です。**curl** を使用する場合は以下ようになります。

```
$ curl -X GET -s -k -u sat_username:sat_password \
https://satellite6.example.com/api/smart_class_parameters/63
```

これで、PUT 呼び出しでパラメーターの値をオーバーライドすることができます。

```
$ curl -H "Accept:application/json,version=2" \
-H "Content-Type:application/json" -X PUT \
-s -k -u sat_username:sat_password \
-d '{"smart_class_parameter":{"override":true}}' \
https://satellite6.example.com/api/smart_class_parameters/63
```

パラメーターを手動で作成または削除する方法はありません。属性の変更のみが可能です。パラメーターは、プロキシからクラスをインポートすることでのみ作成、削除されます。

オーバーライドが有効な場合には、カスタムの Override Matcher を追加できます。

```
$ curl -H "Accept:application/json,version=2" \
-H "Content-Type:application/json" -X PUT \
-s -k -u sat_username:sat_password \
-d '{"smart_class_parameter":{"override_value": \
{"match":"hostgroup=Test","value":"2.4.6"}}}' \
https://satellite6.example.com/api/smart_class_parameters/63
```

API 呼び出しの全パラメーターに関する情報は

https://satellite6.example.com/apidoc/v2/override_values.html で参照してください。

オーバーライド値を削除するには、以下のようなコマンドを使用します。

```
$ curl -X DELETE -s -u sat_username:sat_password \
https://satellite6.example.com/api/smart_class_parameters/63/override_valu
es/3
```

4.1.3.1. 外部ファイルを使用したスマートクラスパラメーターの変更

外部ファイルを使用すると、JSON データでの作業が簡素化されます。構文が強調されるエディターを使用するので、間違いを回避し、特定しやすくなります。

手順4.2 外部ファイルを使用したスマートクラスパラメーターの変更

以下の例では、MOTD Puppet マニフェストを使用します。

1. 今回は **motd** という名前で Puppet クラスを検索します。

```
$ curl -H "Accept:application/json,version=2" \
-H "Content-Type:application/json" -X GET \
-u sat_user:sat_passwd -k \
"https://satellite6.example.com/api/smart_class_parameters?
search=puppetclass_name=motd" \
| python -m json.tool
{
  "page": 1,
  "per_page": 20,
  "results": [
    {
      "avoid_duplicates": false,
      "created_at": "2017-02-06 12:37:48 UTC",
      "default_value": "",
      "description": "",
      "hidden_value": "*****",
      "hidden_value?": false,
      "id": 3,
      "merge_default": false,
      "merge_overrides": false,
      "override": false,
      "override_value_order": "fqdn\nnhostgroup\nnos\ndomain",
      "override_values_count": 0,
      "parameter": "content",
      "parameter_type": "string",
      "puppetclass_id": 3,
      "puppetclass_name": "motd",
      "required": false,
      "updated_at": "2017-02-07 13:08:42 UTC",
      "use_puppet_default": false,
      "validator_rule": null,
      "validator_type": ""
    },
    {
      "avoid_duplicates": false,
      "created_at": "2017-02-06 12:37:48 UTC",
      "default_value": true,
      "description": "",
      "hidden_value": "*****",
      "hidden_value?": false,
      "id": 1,
      "merge_default": false,
      "merge_overrides": false,
      "override": false,
      "override_value_order": "fqdn\nnhostgroup\nnos\ndomain",
      "override_values_count": 0,
      "parameter": "dynamic_motd",
      "parameter_type": "boolean",
```

```

"puppetclass_id": 3,
"puppetclass_name": "motd",
"required": false,
"updated_at": "2017-02-06 15:21:06 UTC",
"use_puppet_default": null,
"validator_rule": null,
"validator_type": null
},
{
"avoid_duplicates": false,
"created_at": "2017-02-06 12:37:48 UTC",
"default_value": "",
"description": "",
"hidden_value": "*****",
"hidden_value?": false,
"id": 2,
"merge_default": false,
"merge_overrides": false,
"override": false,
"override_value_order": "fqdn\ngroup\nnos\ndomain",
"override_values_count": 0,
"parameter": "template",
"parameter_type": "string",
"puppetclass_id": 3,
"puppetclass_name": "motd",
"required": false,
"updated_at": "2017-02-06 15:21:06 UTC",
"use_puppet_default": null,
"validator_rule": null,
"validator_type": null
}
],
"search": "puppetclass_name=motd",
"sort": {
"by": null,
"order": null
},
"subtotal": 3,
"total": 66
}

```

各スマートクラスパラメーターは、同じ Satellite インスタンスにおいてグローバルな ID となっています。**motd** クラスの **content** パラメーターは、この Satellite サーバーでは **id=3** です。この ID は、Puppet クラス名の前に表示される Puppet クラス ID と間違わないようにしてください。

2. パラメーター ID **3** を使用して、**motd** パラメーター固有の情報を取得して、出力を **output_file.json** などのファイルにリダイレクトします。

```

$ curl -H "Accept:application/json,version=2" \
-H "Content-Type:application/json" -X GET \
-u sat_user:sat_passwd -k \
"https://satellite6.example.com/api/smart_class_parameters/3" \
| python -m json.tool > output_file.json

```

3. 以前の手順で作成したファイルを **changed_file.json** という新しいファイルにコピーして

編集します。エディターでファイルを開き、任意の値に変更します。以下の例では、**motd** モジュールのコンテンツパラメーターを変更します。この際、**override** オプションを **false** から **true** に変更する必要があります。

```
{
  "avoid_duplicates": false,
  "created_at": "2017-02-06 12:37:48 UTC", # This line must be
  removed.
  "default_value": "", # A new value should be supplied here.
  "description": "",
  "hidden_value": "*****",
  "hidden_value?": false,
  "id": 3,
  "merge_default": false,
  "merge_overrides": false,
  "override": false, # The override value must be set to true.
  "override_value_order": "fqdn\ngroup\nnos\ndomain",
  "override_values": [], # This line must be removed.
  "override_values_count": 0,
  "parameter": "content",
  "parameter_type": "string",
  "puppetclass_id": 3,
  "puppetclass_name": "motd",
  "required": false,
  "updated_at": "2017-02-07 11:56:55 UTC", # This line must be
  removed.
  "use_puppet_default": false,
  "validator_rule": null,
  "validator_type": ""
}
```

4. ファイルの編集後に、以下のようになっていることを確認して、変更を保存します。

```
{
  "avoid_duplicates": false,
  "default_value": "No Unauthorized Access Allowed",
  "description": "",
  "hidden_value": "*****",
  "hidden_value?": false,
  "id": 3,
  "merge_default": false,
  "merge_overrides": false,
  "override": true,
  "override_value_order": "fqdn\ngroup\nnos\ndomain",
  "override_values_count": 0,
  "parameter": "content",
  "parameter_type": "string",
  "puppetclass_id": 3,
  "puppetclass_name": "motd",
  "required": false,
  "use_puppet_default": false,
  "validator_rule": null,
  "validator_type": ""
}
```

5. 以下のように PUT コマンドを使用して、Satellite サーバーに変更を適用します。

```
$ curl -H "Accept:application/json,version=2" \
-H "Content-Type:application/json" \
-X PUT -u $user:$passwd \
-d @changed_file.json \
-k "https://satellite6.example.com/api/smart_class_parameters/3"
```

4.1.4. エラータのホストまたはホストコレクションへの適用

PUT コマンドで `curl` を使用して、エラータをホスト、ホストグループ、またはホストコレクションに適用することができます。以下は、PUT 要求の基本的な構文です。

```
$ curl -H "Accept:application/json,version=2" \
-H "Content-Type:application/json" -X PUT \
-u sat_username:sat_password -k \
-d json-formatted-data https://satellite6.example.com
```

同梱の API ドキュメント (<https://satellite6.example.com/apidoc/v2.html>) を参照してエラータの適用に使用する URL を検索します。Satellite Web UI を使用すると、検索クエリー形式の検索に役立ちます。ホスト → ホストコレクション の順に移動して、ホストコレクションを選択します。コレクションの各種アクション → エラータのインストール に移動して、検索クエリーボックスに注目します。たとえば `my-collection` と呼ばれるホストコレクションでは、検索ボックスに `host_collection="my-collection"` が含まれます。以下のホストコレクションの例では、これを使用します。

例4.4 ホストへのエラータの適用

以下の例では、一括アクションの API URL `/katello/api/hosts/bulk/install_content` を使用して単純な検索に必要な形式を表示します。

```
$ curl -H "Accept:application/json,version=2" \
-H "Content-Type:application/json" -X PUT \
-u sat_username:sat_password -k \
-d "{\"organization_id\":1,\"included\":{\"search\":\"my-host\"},\"content_type\":\"errata\",\"content\":[\"RHBA-2016:1981\"]}"
https://satellite6.example.com/api/v2/hosts/bulk/install_content
```

例4.5 ホストコレクションへのエラータの適用

以下の例では、Satellite Web UI に表示されているように、検索文字列 `host_collection="my-collection"` を渡すのに必要なエスケープレベルに注目してください。

```
$ curl -H "Accept:application/json,version=2" \
-H "Content-Type:application/json" -X PUT \
-u sat_username:sat_password -k \
-d "{\"organization_id\":1,\"included\":{\"search\":\"host_collection=\\\\"my-collection\\\\\\\",\"content_type\":\"errata\",\"content\":[\"RHBA-2016:1981\"]}"
https://satellite6.example.com/api/v2/hosts/bulk/install_content
```


4.2. RUBY を使用した API の例

以下の例では、Ruby を使用してさまざまなタスクを実行して Satellite API と通信する方法について説明します。



重要

以下は、スクリプトおよびコマンドの例です。以下のスクリプトを慎重にレビューしてから使用するようにしてください。変数、ユーザー名、パスワード、その他の情報は、お使いのデプロイメントに適した値に置き換えてください。

4.2.1. Ruby を使用したオブジェクトの作成

以下のスクリプトは Red Hat Satellite 6 API と接続して、新しい組織を作成し、その新規組織内に 3 つの環境を作成します。組織がすでに存在する場合、スクリプトはその組織を使用します。組織内に環境が 1 つでも存在する場合は、スクリプトによりエラーが送出されて、スクリプトは終了します。

```
#!/usr/bin/ruby

require 'rest-client'
require 'json'

url = 'https://satellite6.example.com/api/v2/'
katello_url = "#{url}/katello/api/v2/"

$username = 'admin'
$password = 'changeme'

org_name = "MyOrg"
environments = [ "Development", "Testing", "Production" ]

# Performs a GET using the passed URL location
def get_json(location)
  response = RestClient::Request.new(
    :method => :get,
    :url => location,
    :user => $username,
    :password => $password,
    :headers => { :accept => :json,
                  :content_type => :json }
  ).execute
  JSON.parse(response.to_str)
end

# Performs a POST and passes the data to the URL location
def post_json(location, json_data)
  response = RestClient::Request.new(
    :method => :post,
    :url => location,
    :user => $username,
    :password => $password,
    :headers => { :accept => :json,
                  :content_type => :json},
    :payload => json_data
  ).execute
```

```

    JSON.parse(response.to_str)
  end

  # Creates a hash with ids mapping to names for an array of records
  def id_name_map(records)
    records.inject({}) do |map, record|
      map.update(record['id'] => record['name'])
    end
  end

  # Get list of existing organizations
  orgs = get_json("#{katello_url}/organizations")
  org_list = id_name_map(orgs['results'])

  if !org_list.has_value?(org_name)
    # If our organization is not found, create it
    puts "Creating organization: \t#{org_name}"
    org_id = post_json("#{katello_url}/organizations",
JSON.generate({"name"=> org_name}))["id"]
  else
    # Our organization exists, so let's grab it
    org_id = org_list.key(org_name)
    puts "Organization \"#{org_name}\" exists"
  end

  # Get list of organization's lifecycle environments
  envs = get_json("#{katello_url}/organizations/#{org_id}/environments")
  env_list = id_name_map(envs['results'])
  prior_env_id = env_list.key("Library")

  # Exit the script if at least one life cycle environment already exists
  environments.each do |e|
    if env_list.has_value?(e)
      puts "ERROR: One of the Environments is not unique to organization"
      exit
    end
  end

  # Create life cycle environments
  environments.each do |environment|
    puts "Creating environment: \t#{environment}"
    prior_env_id = post_json("#{katello_url}/organizations/#{
org_id}/environments", JSON.generate({"name" => environment,
"organization_id" => org_id, "prior_id" => prior_env_id}))["id"]
  end
end

```

4.2.2. Apipie バインディングの使用

Apipie バインディングは、apipie で記述した API 向けの Ruby のバインディングで、Satellite から API の定義をフェッチして、キャッシュし、要求に応じて API 呼び出しを生成します。apipie バインディングを使用すると、Ruby API のクエリーが簡素化できます。Apipie は通常、h を発音せずに、「happy」のリズムで「アピーパイ」と発音します。

以下の例では、新しい組織を作成してから、その新規組織に 3 つの環境を作成します。組織がすでに存在する場合、スクリプトはその組織を使用します。組織内に環境が 1 つでも存在する場合は、スクリプトによりエラーが送出されて、スクリプトは終了します。

```
#!/usr/bin/tfm-ruby

require 'apipie-bindings'

org_name = "MyOrg"
environments = [ "Development", "Testing", "Production" ]

# Create an instance of apipie bindings
@api = ApipieBindings::API.new({
  :uri => 'https://satellite6.example.com/',
  :username => 'admin',
  :password => 'changeme',
  :api_version => 2
})

# Performs an API call with default options
def call_api(resource_name, action_name, params = {})
  http_headers = {}
  apipie_options = { :skip_validation => true }
  @api.resource(resource_name).call(action_name, params, http_headers,
  apipie_options)
end

# Creates a hash with IDs mapping to names for an array of records
def id_name_map(records)
  records.inject({}) do |map, record|
    map.update(record['id'] => record['name'])
  end
end

# Get list of existing organizations
orgs = call_api(:organizations, :index)
org_list = id_name_map(orgs['results'])

if !org_list.has_value?(org_name)
  # If our organization is not found, create it
  puts "Creating organization: \t#{org_name}"
  org_id = call_api(:organizations, :create, {'organization' => { :name =>
  org_name }})['id']
else
  # Our organization exists, so let's grab it
  org_id = org_list.key(org_name)
  puts "Organization \"#{org_name}\" exists"
end

# Get list of organization's life cycle environments
envs = call_api(:lifecycle_environments, :index, {'organization_id' =>
  org_id})
env_list = id_name_map(envs['results'])
prior_env_id = env_list.key("Library")

# Exit the script if at least one life cycle environment already exists
environments.each do |e|
  if env_list.has_value?(e)
    puts "ERROR: One of the Environments is not unique to organization"
    exit
  end
end
```

```

    end
end

# Create life cycle environments
environments.each do |environment|
  puts "Creating environment: \t#{environment}"
  prior_env_id = call_api(:lifecycle_environments, :create, {"name" =>
environment, "organization_id" => org_id, "prior_id" => prior_env_id })
  ['id']
end

```

4.3. PYTHON を使用した API の例

以下の例では、Python を使用してさまざまなタスクを実行して Satellite API と通信する方法について説明します。



重要

以下は、スクリプトおよびコマンドの例です。以下のスクリプトを慎重にレビューしてから使用するようにしてください。変数、ユーザー名、パスワード、その他の情報は、お使いのデプロイメントに適した値に置き換えてください。

以下のスクリプトは、REST API との対話に SSL 検証を使用しません。ここで紹介しているスクリプトはデモのみを目的としています。

4.3.1. Python を使用したオブジェクトの作成

以下のスクリプトは Red Hat Satellite 6 API と接続して、新しい組織を作成し、その新規組織内に 3 つの環境を作成します。組織がすでに存在する場合、スクリプトはその組織を使用します。組織内に環境が 1 つでも存在する場合は、スクリプトによりエラーが送出されて、スクリプトは終了します。

```

#!/usr/bin/python

import json
import sys

try:
    import requests
except ImportError:
    print "Please install the python-requests module."
    sys.exit(-1)

# URL to your Satellite 6 server
URL = "https://satellite6.example.com"
# URL for the API to your deployed Satellite 6 server
SAT_API = "%s/katello/api/v2/" % URL
# Katello-specific API
KATELLO_API = "%s/katello/api/" % URL
POST_HEADERS = {'content-type': 'application/json'}
# Default credentials to login to Satellite 6
USERNAME = "admin"
PASSWORD = "changeme"
# Ignore SSL for now
SSL_VERIFY = False

```

```
# Name of the organization to be either created or used
ORG_NAME = "MyOrg"
# Name for life cycle environments to be either created or used
ENVIRONMENTS = ["Development", "Testing", "Production"]

def get_json(location):
    """
    Performs a GET using the passed URL location
    """

    r = requests.get(location, auth=(USERNAME, PASSWORD),
verify=SSL_VERIFY)

    return r.json()

def post_json(location, json_data):
    """
    Performs a POST and passes the data to the URL location
    """

    result = requests.post(
        location,
        data=json_data,
        auth=(USERNAME, PASSWORD),
        verify=SSL_VERIFY,
        headers=POST_HEADERS)

    return result.json()

def main():
    """
    Main routine that creates or re-uses an organization and
    life cycle environments. If life cycle environments already
    exist, exit out.
    """

    # Check if our organization already exists
    org = get_json(SAT_API + "organizations/" + ORG_NAME)

    # If our organization is not found, create it
    if org.get('error', None):
        org_id = post_json(
            SAT_API + "organizations/",
            json.dumps({"name": ORG_NAME}))["id"]
        print "Creating organization: \t" + ORG_NAME
    else:
        # Our organization exists, so let's grab it
        org_id = org['id']
        print "Organization '%s' exists." % ORG_NAME

    # Now, let's fetch all available life cycle environments for this
    org...
```

```

envs = get_json(
    SAT_API + "organizations/" + str(org_id) + "/environments/")

# ... and add them to a dictionary, with respective 'Prior'
environment
prior_env_id = 0
env_list = {}
for env in envs['results']:
    env_list[env['id']] = env['name']
    prior_env_id = env['id'] if env['name'] == "Library" else
prior_env_id

# Exit the script if at least one life cycle environment already
exists
if all(environment in env_list.values() for environment in
ENVIRONMENTS):
    print "ERROR: One of the Environments is not unique to
organization"
    sys.exit(-1)

# Create life cycle environments
for environment in ENVIRONMENTS:
    new_env_id = post_json(
        SAT_API + "organizations/" + str(org_id) + "/environments/",
        json.dumps(
            {
                "name": environment,
                "organization_id": org_id,
                "prior": prior_env_id}
        ))["id"]

    print "Creating environment: \t" + environment
    prior_env_id = new_env_id

if __name__ == "__main__":
    main()

```

4.3.2. Python を使用したクエリーの実行

Python スクリプトを作成して、実行すると、「[curl を使用した API の例](#)」に記載のものと同じ結果を獲得することができます。まず、**sat6api.py** という実行可能なファイルを作成して、以下の内容を追加します。

```

#!/usr/bin/python
import json
import sys
try:
    import requests
except ImportError:
    print "Please install the python-requests module."
    sys.exit(-1)

SAT_API = 'https://satellite6.example.com/api/v2/'
USERNAME = "admin"

```

```
PASSWORD = "password"
SSL_VERIFY = False # Ignore SSL for now

def get_json(url):
    # Performs a GET using the passed URL location
    r = requests.get(url, auth=(USERNAME, PASSWORD), verify=SSL_VERIFY)
    return r.json()

def get_results(url):
    jsn = get_json(url)
    if jsn.get('error'):
        print "Error: " + jsn['error']['message']
    else:
        if jsn.get('results'):
            return jsn['results']
        elif 'results' not in jsn:
            return jsn
        else:
            print "No results found"
    return None

def display_all_results(url):
    results = get_results(url)
    if results:
        print json.dumps(results, indent=4, sort_keys=True)

def display_info_for_hosts(url):
    hosts = get_results(url)
    if hosts:
        for host in hosts:
            print "ID: %-10d Name: %-30s IP: %-20s OS: %-30s" %
(host['id'], host['name'], host['ip'], host['operatingsystem_name'])

def main():
    host = 'satellite6.example.com'
    print "Displaying all info for host %s ..." % host
    display_all_results(SAT_API + 'hosts/' + host)

    print "Displaying all facts for host %s ..." % host
    display_all_results(SAT_API + 'hosts/%s/facts' % host)

    host_pattern = 'example'
    print "Displaying basic info for hosts matching pattern '%s'..." %
host_pattern
    display_info_for_hosts(SAT_API + 'hosts?search=' + host_pattern)

    environment = 'production'
    print "Displaying basic info for hosts in environment %s..." %
environment
    display_info_for_hosts(SAT_API + 'hosts?search=environment=' +
environment)

    model = 'RHEV Hypervisor'
    print "Displaying basic info for hosts with model name %s..." % model
    display_info_for_hosts(SAT_API + 'hosts?search=model="' + model + '"')
```

```
if __name__ == "__main__":  
    main()
```

次にコマンドラインから `./sat6api.py` を実行して、結果を表示することができます。

4.4. 詳細検索の使用

Web UI を使用して、クエリーの構築に使用可能な検索用語を判断することができます。Satellite 6 は、範囲内の検索や自動補完機能をサポートしており、このタスクを簡素化します。

たとえば、オペレーティングシステム別にホストを検索するには **ホスト →すべてのホスト** の順に移動して、**検索** テキストボックスをクリックして検索用語の一覧を表示します。オペレーティングシステムの検索用語の一例は **os_description** で、以下のように API クエリーで使用することができます。

```
$ curl -s -k -u sat_username:sat_password  
https://satellite6.example.com/api/v2/hosts?  
search=os_description=\"RHEL+Server+6.6\" | python -m json.tool  
{  
  ...  
  "results": [  
    {  
      "name": "satellite6.example.com",  
      "operatingsystem_id": 1,  
      "operatingsystem_name": "RHEL Server 6.6",  
      ...  
    }  
  ],  
  "search": "os_description=\"RHEL Server 6.6\"",  
}
```

4.5. ページネーション制御のある検索の使用

ページネーションパラメーター **per_page** および **page** を使用して、API 検索クエリーが返す検索結果を絞り込むことができます。**per_page** パラメーターは、ページごとの項目数を指定し、**page** パラメーターは、**per_page** パラメーターで算出された数値をもとにどのページを返すかを指定します。

ページネーションパラメーターを指定しない場合に、返す項目数のデフォルト値を 1000 に設定しますが、**page** パラメーターが指定されている場合には **per_page** のデフォルト値は 20 が適用されます。

例4.6 コンテンツビューの表示

以下の例では、返された結果の 3 ページ目の 10 件分をリストするコンテンツビューが表示されません。

```
$ curl -X GET --user sat_username:sat_password \  
"https://satellite6.example.com/katello/api/content_views?  
per_page=10&page=3"
```

例4.7 アクティベーションキーの表示

以下の例では、ID が 1 の組織のアクティベーションキー 30 個を 1 ページにリストする検索の 2 ページ目を表示します。

```
$ curl -X GET --user sat_username:sat_password \
"https://satellite6.example.com/katello/api/activation_keys?
organization_id=1&per_page=30&page=2"
```

複数ページを表示する結果を取得するには、*for loop* の構造を使用します。

例4.8 複数ページを返す設定

以下の例では、ページごとに 5 件ずつ表示するコンテンツビュー 3 ページ分の 1 ページを返します。

```
$ for i in `seq 1 3`; do curl -X GET --user sat_username:sat_password \
"https://satellite6.example.com/katello/api/content_views?
per_page=5&page=$i"; done
```

4.6. ライフサイクル環境との作業

『Planning for Red Hat Satellite 6』の [Life Cycle Environments](#) セクションの説明にあるように、アプリケーションのライフサイクルは複数のライフサイクル環境に分けられ、各ライフサイクル環境はアプリケーションライフサイクルの各ステージを表します。複数のライフサイクル環境が結び付けられ、環境パスを形成します。API を使用してライフサイクル環境を結び付けるには、**prior_id** パラメーターを使用します。

https://satellite6.example.com/apidoc/v2/lifecycle_environments.html でライフサイクル環境に関する同梱の API リファレンスを参照できます。API ルートには、**/katello/api/environments** および **/katello/api/organizations/:organization_id/environments** が含まれます。

Satellite で、デフォルトの組織 **1** に対する現在の全ライフサイクル環境を、以下のように表示することができます。

```
$ curl -H "Accept:application/json,version=2" \
-H "Content-Type:application/json" -X GET \
-u sat_user:sat_password -k \
https://satellite6.example.com/katello/api/organizations/1/environments |
python -m json.tool
```

新規インストールした Satellite の出力では、以下のようなセクションが表示されるはずですが、

```
出力省略
"description": null,
"id": 1,
"label": "Library",
"library": true,
"name": "Library",
"organization": {
  "id": 1,
  "label": "Default_Organization",
```

```

    "name": "Default Organization"
  },
  "permissions": {
    "destroy_lifecycle_environments": false,
    "edit_lifecycle_environments": true,
    "promote_or_remove_content_views_to_environments": true,
    "view_lifecycle_environments": true
  },
  "prior": null,
  "successor": null,
  出力省略

```

以下の手順では、ID が **1** のデフォルトのライブラリー環境が、ライフサイクル環境作成の開始点として使用されています。

手順4.3 ライフサイクル環境のリンク作成

1. 開始点として使用する既存のライフサイクル環境を選択します。その ID を使用して環境を表示します。今回の例では、ID が **1** の環境を使用します。

```

$ curl -X GET -s -k -u sat_user:sat_password \
https://satellite6.example.com/katello/api/environments/1 | python -
m json.tool
出力省略
  "id": 1,
  "label": "Library",
出力省略
  "prior": null,
  "successor": null,
出力省略

```

2. **1** に設定した **prior** オプションを使用して、新しいライフサイクル環境を作成します。

- a. 内容が **{"organization_id":1,"label":"api-dev","name":"API Development","prior":1}** の **life-cycle.json** などの JSON ファイルを作成します。
- b. 以下のようなコマンドを入力します。

```

$ curl -H "Accept:application/json,version=2" \
-H "Content-Type:application/json" -X POST \
-u sat_user:sat_password -k \
-d @life-cycle.json \
https://satellite6.example.com/katello/api/environments \
| python -m json.tool
出力省略
  "description": null,
  "id": 2,
  "label": "api-dev",
  "library": false,
  "name": "API Development",
  "organization": {
    "id": 1,
    "label": "Default Organization",
    "name": "Default Organization"
  }

```

```

    },
    "permissions": {
      "destroy_lifecycle_environments": true,
      "edit_lifecycle_environments": true,
      "promote_or_remove_content_views_to_environments": true,
      "view_lifecycle_environments": true
    },
    "prior": {
      "id": 1,
      "name": "Library"
    },
  },
  出力省略

```

以下のコマンドの出力では、ライフサイクル環境の ID が 2 で、その 1 つ前のライフサイクル環境は 1 であると分かります。これは、1 と 2 がリンクされていることを示します。この環境の後継を作成する場合に、ライフサイクル環境 ID 2 が使用されます。

3. 2 に設定した **prior** オプションを使用して、別のライフサイクル環境を作成します。
 - a. 以前に作成した **life-cycle.json** を編集して、**label**、**name**、**prior** の値を **{"organization_id":1, "label":"api-qa", "name":"API QA","prior":2}** のように更新します。
 - b. 以下のようなコマンドを入力します。

```

$ curl -H "Accept:application/json,version=2" \
-H "Content-Type:application/json" -X POST \
-u sat_user:sat_password -k \
-d @life-cycle.json \
https://satellite6.example.com/katello/api/environments \
| python -m json.tool
  出力省略
  "description": null,
  "id": 3,
  "label": "api-qa",
  "library": false,
  "name": "API QA",
  "organization": {
    "id": 1,
    "label": "Default Organization",
    "name": "Default Organization"
  },
  "permissions": {
    "destroy_lifecycle_environments": true,
    "edit_lifecycle_environments": true,
    "promote_or_remove_content_views_to_environments": true,
    "view_lifecycle_environments": true
  },
  "prior": {
    "id": 2,
    "name": "API Development"
  },
  "successor": null,
  出力省略

```

以下のコマンドの出力では、ライフサイクル環境の ID が **3** で、その 1 つ前のライフサイクル環境は **1** であると分かります。これは、**2** と **3** がリンクされていることを示します。

ライフサイクル環境の更新

以下のように、ライフサイクル環境は PUT コマンドを使用して更新できます。

```
$ curl -H "Accept:application/json,version=2" \  
-H "Content-Type:application/json" -X POST \  
-u sat_user:sat_password -k \  
-d '{"description":"Quality Acceptance Testing"}' \  
https://satellite6.example.com/katello/api/environments/3 \  
| python -m json.tool  
出力省略  
{  
  "description": "Quality Acceptance Testing",  
  "id": 3,  
  "label": "api-qa",  
  "library": false,  
  "name": "API QA",  
  "organization": {  
    "id": 1,  
    "label": "Default_Organization",  
    "name": "Default Organization"  
  },  
  "permissions": {  
    "destroy_lifecycle_environments": true,  
    "edit_lifecycle_environments": true,  
    "promote_or_remove_content_views_to_environments": true,  
    "view_lifecycle_environments": true  
  },  
  "prior": {  
    "id": 2,  
    "name": "API Development"  
  },  
  "organization": {  
    "id": 1,  
    "label": "Default_Organization",  
    "name": "Default Organization"  
  }  
}
```

ライフサイクル環境の削除

後継する環境がない場合には、ライフサイクル環境を削除できます。そのため、

```
curl -X DELETE -s -k -u sat_user:sat_password  
https://satellite6.example.com/katello/api/environments/:id
```

の形式のコマンドを使用して、逆の順番で環境を削除します。

付録A API レスポンスコード

Red Hat Satellite 6 API は、API 呼び出しに対して HTTP のレスポンスステータスコードが表示されます。以下のコードは、Satellite API の全リソースに共通します。

表A.1 API レスポンスコード

レスポンス	説明
200 OK	show、index、update、delete (GET、PUT、DELETE 要求) などの要求アクションに成功した場合
201 Created	create アクション (POST 要求) に成功した場合
301 Moved Permanently	Satellite が HTTPS しか使用できないように制限されているにも拘らず、HTTP が試行された時にリダイレクトされる場合
400 Bad Request	必要なパラメーターがないか、検索クエリーに無効な構文が含まれている場合
401 Unauthorized	ユーザー認証に失敗した場合 (認証情報が不正な場合など)
403 Forbidden	ユーザーにアクションの実行、リソースの読み込みなど、十分なパーミッションがないか、アクション自体がサポートされていない場合
404 Not Found	指定の ID のレコードが存在しない場合。要求のレコードが存在しない場合には show または delete アクションに、関連のレコードの 1 つが存在しない場合には create、update、delete アクションに表示されることがあります。
409 Conflict	既存の依存関係が原因でレコードを削除できない場合 (例: ホストのあるホストグループなど)
415 Unsupported Media Type	HTTP 要求のコンテンツタイプが JSON でない場合
422 Unprocessable Entity	バリデーションエラーが原因でエンティティの作成に失敗した場合。アクションの作成または更新にのみ適用されます。
500 Internal Server Error	予期しないサーバーエラーが発生した場合
503 Service Unavailable	サーバーが実行されていない場合

付録B API パーミッションの表

Red Hat Satellite 6 API は、さまざまなアクションをサポートしますが、それぞれのアクションには固有のパーミッションが必要です。以下の表では、API パーミッション名、そのパーミッションに関連付けられたアクションとリソースタイプを紹介しています。

表B.1 API パーミッションの表

パーミッション名	アクション	リソースタイプ
view_activation_keys	<ul style="list-style-type: none"> katello/activation_keys/all katello/activation_keys/index katello/activation_keys/auto_complete_search katello/api/v2/activation_keys/index katello/api/v2/activation_keys/show katello/api/v2/activation_keys/available_host_collections katello/api/v2/activation_keys/available_releases katello/api/v2/activation_keys/product_content 	Katello::ActivationKey
create_activation_keys	<ul style="list-style-type: none"> katello/api/v2/activation_keys/create katello/api/v2/activation_keys/copy 	Katello::ActivationKey
edit_activation_keys	<ul style="list-style-type: none"> katello/api/v2/activation_keys/update katello/api/v2/activation_keys/content_override katello/api/v2/activation_keys/add_subscriptions katello/api/v2/activation_keys/remove_subscriptions 	Katello::ActivationKey
destroy_activation_keys	<ul style="list-style-type: none"> katello/api/v2/activation_keys/destroy 	Katello::ActivationKey
logout	<ul style="list-style-type: none"> users/logout 	

パーMISSION名	アクション	リソースタイプ
view_architectures	<ul style="list-style-type: none">• architectures/index• architectures/show• architectures/auto_complete_search• api/v1/architectures/index• api/v1/architectures/show• api/v2/architectures/index• api/v2/architectures/show	
create_architectures	<ul style="list-style-type: none">• architectures/new• architectures/create• api/v1/architectures/create• api/v2/architectures/create	
edit_architectures	<ul style="list-style-type: none">• architectures/edit• architectures/update• api/v1/architectures/update• api/v2/architectures/update	
destroy_architectures	<ul style="list-style-type: none">• architectures/destroy• api/v1/architectures/destroy• api/v2/architectures/destroy	
view_audit_logs	<ul style="list-style-type: none">• audits/index• audits/show• audits/auto_complete_search• api/v1/audits/index• api/v1/audits/show• api/v2/audits/index• api/v2/audits/show	

パーミッション名	アクション	リソースタイプ
view_authenticators	<ul style="list-style-type: none">auth_source_ldaps/indexauth_source_ldaps/showapi/v1/auth_source_ldaps/indexapi/v1/auth_source_ldaps/showapi/v2/auth_source_ldaps/indexapi/v2/auth_source_ldaps/show	
create_authenticators	<ul style="list-style-type: none">auth_source_ldaps/newauth_source_ldaps/createapi/v1/auth_source_ldaps/createapi/v2/auth_source_ldaps/create	
edit_authenticators	<ul style="list-style-type: none">auth_source_ldaps/editauth_source_ldaps/updateapi/v1/auth_source_ldaps/updateapi/v2/auth_source_ldaps/update	
destroy_authenticators	<ul style="list-style-type: none">auth_source_ldaps/destroyapi/v1/auth_source_ldaps/destroyapi/v2/auth_source_ldaps/destroy	
view_bookmarks	<ul style="list-style-type: none">bookmarks/indexbookmarks/showapi/v1/bookmarks/indexapi/v1/bookmarks/showapi/v2/bookmarks/indexapi/v2/bookmarks/show	

パーMISSION名	アクション	リソースタイプ
create_bookmarks	<ul style="list-style-type: none">• bookmarks/new• bookmarks/create• api/v1/bookmarks/new• api/v1/bookmarks/create• api/v2/bookmarks/new• api/v2/bookmarks/create	
edit_bookmarks	<ul style="list-style-type: none">• bookmarks/edit• bookmarks/update• api/v1/bookmarks/edit• api/v1/bookmarks/update• api/v2/bookmarks/edit• api/v2/bookmarks/update	
destroy_bookmarks	<ul style="list-style-type: none">• bookmarks/destroy• api/v1/bookmarks/destroy• api/v2/bookmarks/destroy	
download_bootdisk	<ul style="list-style-type: none">• foreman_bootdisk/disks/generic• foreman_bootdisk/disks/host• foreman_bootdisk/disks/full_host• foreman_bootdisk/disks/subnet• foreman_bootdisk/disks/help• foreman_bootdisk/api/v2/disks/generic• foreman_bootdisk/api/v2/disks/host	

パーミッション名	アクション	リソースタイプ
manage_capsule_content	<ul style="list-style-type: none"> katello/api/v2/capsule_content/lifecycle_environments katello/api/v2/capsule_content/available_lifecycle_environments katello/api/v2/capsule_content/add_lifecycle_environment katello/api/v2/capsule_content/remove_lifecycle_environment katello/api/v2/capsule_content/sync katello/api/v2/capsule_content/sync_status katello/api/v2/capsule_content/cancel_sync 	SmartProxy
view_capsule_content	<ul style="list-style-type: none"> smart_proxies/pulp_storage smart_proxies/pulp_status smart_proxies/show_with_content 	SmartProxy
view_compute_profiles	<ul style="list-style-type: none"> compute_profiles/index compute_profiles/show compute_profiles/auto_complete_search api/v2/compute_profiles/index api/v2/compute_profiles/show 	
create_compute_profiles	<ul style="list-style-type: none"> compute_profiles/new compute_profiles/create api/v2/compute_profiles/create 	
edit_compute_profiles	<ul style="list-style-type: none"> compute_profiles/edit compute_profiles/update api/v2/compute_profiles/update 	

パーMISSION名	アクション	リソースタイプ
destroy_compute_profiles	<ul style="list-style-type: none"> • compute_profiles/destroy • api/v2/compute_profiles/destroy 	
view_compute_resources	<ul style="list-style-type: none"> • compute_resources/index • compute_resources/show • compute_resources/auto_complete_search • compute_resources/ping • compute_resources/available_images • api/v1/compute_resources/index • api/v1/compute_resources/show • api/v2/compute_resources/index • api/v2/compute_resources/show • api/v2/compute_resources/available_images • api/v2/compute_resources/available_clusters • api/v2/compute_resources/available_folders • api/v2/compute_resources/available flavors • api/v2/compute_resources/available_networks • api/v2/compute_resources/available_resource_pools • api/v2/compute_resources/available_security_groups • api/v2/compute_resources/available_storage_domains • api/v2/compute_resources/available_zones • api/v2/compute_resources/available_storage_pods 	

パーミッション名	アクション	リソースタイプ
create_compute_resources	<ul style="list-style-type: none">• compute_resources/new• compute_resources/create• compute_resources/test_connection• api/v1/compute_resources/create• api/v2/compute_resources/create	
edit_compute_resources	<ul style="list-style-type: none">• compute_resources/edit• compute_resources/update• compute_resources/test_connection• compute_attributes/new• compute_attributes/create• compute_attributes/edit• compute_attributes/update• api/v1/compute_resources/update• api/v2/compute_resources/update• api/v2/compute_attributes/create• api/v2/compute_attributes/update	
destroy_compute_resources	<ul style="list-style-type: none">• compute_resources/destroy• api/v1/compute_resources/destroy• api/v2/compute_resources/destroy	
view_compute_resources_vms	<ul style="list-style-type: none">• compute_resources_vms/index• compute_resources_vms/show	
create_compute_resources_vms	<ul style="list-style-type: none">• compute_resources_vms/new• compute_resources_vms/create	

パーMISSION名	アクション	リソースタイプ
edit_compute_resources_vms	<ul style="list-style-type: none"> • compute_resources_vms/edit • compute_resources_vms/update 	
destroy_compute_resources_vms	<ul style="list-style-type: none"> • compute_resources_vms/destroy 	
power_compute_resources_vms	<ul style="list-style-type: none"> • compute_resources_vms/power • compute_resources_vms/pause 	
console_compute_resources_vms	<ul style="list-style-type: none"> • compute_resources_vms/console 	
view_config_groups	<ul style="list-style-type: none"> • config_groups/index • config_groups/auto_complete_search • api/v2/config_groups/index • api/v2/config_groups/show 	
create_config_groups	<ul style="list-style-type: none"> • config_groups/new • config_groups/create • api/v2/config_groups/create 	
edit_config_groups	<ul style="list-style-type: none"> • config_groups/edit • config_groups/update • api/v2/config_groups/update 	
destroy_config_groups	<ul style="list-style-type: none"> • config_groups/destroy • api/v2/config_groups/destroy 	

パーミッション名	アクション	リソースタイプ
view_config_reports	<ul style="list-style-type: none"> • api/v1/reports/index • api/v1/reports/show • api/v1/reports/last • api/v2/reports/index • api/v2/reports/show • api/v2/reports/last • config_reports/index • config_reports/show • config_reports/auto_complete_search • api/v2/config_reports/index • api/v2/config_reports/show • api/v2/config_reports/last 	
destroy_config_reports	<ul style="list-style-type: none"> • config_reports/destroy • api/v1/reports/destroy • api/v2/reports/destroy • api/v2/config_reports/destroy 	
upload_config_reports	<ul style="list-style-type: none"> • api/v2/reports/create • api/v2/config_reports/create 	
view_containers	<ul style="list-style-type: none"> • containers/index • containers/show • api/v2/containers/index • api/v2/containers/show • api/v2/containers/logs 	Container
commit_containers	<ul style="list-style-type: none"> • containers/commit 	Container

パーミッション名	アクション	リソースタイプ
create_containers	<ul style="list-style-type: none"> containers/steps/show containers/steps/update containers/new api/v2/containers/create api/v2/containers/power 	Container
destroy_containers	<ul style="list-style-type: none"> containers/destroy api/v2/containers/destroy 	Container
power_compute_resources_vms	<ul style="list-style-type: none"> containers/power api/v2/containers/create api/v2/containers/power 	ComputeResource
view_content_hosts	<ul style="list-style-type: none"> katello/content_hosts/auto_complete_search katello/api/v2/systems/index katello/api/v2/systems/show katello/api/v2/systems/errata katello/api/v2/systems/package_profile katello/api/v2/systems/product_content katello/api/v2/systems/report katello/api/v2/systems/releases katello/api/v2/systems/available_host_collections katello/api/v2/host_collections/systems 	Katello::System
create_content_hosts	<ul style="list-style-type: none"> katello/api/v2/systems/create katello/api/rhsm/candlepin_proxies/consumer_create katello/api/rhsm/candlepin_proxies/consumer_show 	Katello::System

パーミッション名	アクション	リソースタイプ
edit_content_hosts	<ul style="list-style-type: none"> katello/api/v2/systems/update katello/api/v2/systems/content_override katello/api/rhsm/candlepin_proxies/upload_package_profile katello/api/rhsm/candlepin_proxies/regenerate_identity_certificates katello/api/rhsm/candlepin_proxies/hypervisors_update 	Katello::System
destroy_content_hosts	<ul style="list-style-type: none"> katello/api/v2/systems/destroy 	Katello::System
view_content_views	<ul style="list-style-type: none"> katello/api/v2/content_views/index katello/api/v2/content_views/show katello/api/v2/content_views/available_puppet_modules katello/api/v2/content_views/available_puppet_module_names katello/api/v2/content_view_filters/index katello/api/v2/content_view_filters/show katello/api/v2/content_view_filter_rules/index katello/api/v2/content_view_filter_rules/show katello/api/v2/content_view_histories/index katello/api/v2/content_view_puppet_modules/index katello/api/v2/content_view_puppet_modules/show katello/api/v2/content_view_versions/index katello/api/v2/content_view_versions/show katello/api/v2/package_groups/index katello/api/v2/package_groups/show 	Katello::ContentView

パーMISSION名	アクション	リソースタイプ
	<ul style="list-style-type: none"> • katello/api/v2/errata/index • katello/api/v2/errata/show • katello/api/v2/puppet_modules/index • katello/api/v2/puppet_modules/show • katello/content_views/auto_complete • katello/content_views/auto_complete_search • katello/errata/short_details • katello/errata/auto_complete • katello/packages/details • katello/packages/auto_complete • katello/products/auto_complete • katello/repositories/auto_complete_library • katello/content_search/index • katello/content_search/products • katello/content_search/repos • katello/content_search/packages • katello/content_search/errata • katello/content_search/puppet_modules • katello/content_search/packages_items • katello/content_search/errata_items • katello/content_search/puppet_modules_items • katello/content_search/view_packages • katello/content_search/view_puppet_modules • katello/content_search/repo_packages • katello/content_search/repo_errata • katello/content_search/repo_puppet_modules • katello/content_search/repo_compare_errata • katello/content_search/repo_compare_packages • katello/content_search/repo_compare_puppet_modules 	

パーミッション名	アクション <ul style="list-style-type: none"> katello/content_search/view_compare_rata 	リソースタイプ
	<ul style="list-style-type: none"> katello/content_search/view_compare_packages katello/content_search/view_compare_puppet_modules katello/content_search/views 	
create_content_views	<ul style="list-style-type: none"> katello/api/v2/content_views/create katello/api/v2/content_views/copy 	Katello::ContentView
edit_content_views	<ul style="list-style-type: none"> katello/api/v2/content_views/update katello/api/v2/content_view_filters/create katello/api/v2/content_view_filters/update katello/api/v2/content_view_filters/destroy katello/api/v2/content_view_filter_rules/create katello/api/v2/content_view_filter_rules/update katello/api/v2/content_view_filter_rules/destroy katello/api/v2/content_view_puppet_modules/create katello/api/v2/content_view_puppet_modules/update katello/api/v2/content_view_puppet_modules/destroy 	Katello::ContentView
destroy_content_views	<ul style="list-style-type: none"> katello/api/v2/content_views/destroy katello/api/v2/content_views/remove katello/api/v2/content_view_versions/destroy 	Katello::ContentView

パーミッション名	アクション	リソースタイプ
publish_content_views	<ul style="list-style-type: none"> katello/api/v2/content_views/publish katello/api/v2/content_view_versions/incremental_update 	Katello::ContentView
promote_or_remove_content_views	<ul style="list-style-type: none"> katello/api/v2/content_view_versions/promote katello/api/v2/content_views/remove_from_environment katello/api/v2/content_views/remove 	Katello::ContentView
export_content_views	<ul style="list-style-type: none"> katello/api/v2/content_view_versions/export 	Katello::ContentView
access_dashboard	<ul style="list-style-type: none"> dashboard/index dashboard/save_positions dashboard/reset_default dashboard/create dashboard/destroy api/v1/dashboard/index api/v2/dashboard/index 	
view_discovered_hosts	<ul style="list-style-type: none"> discovered_hosts/index discovered_hosts/show discovered_hosts/auto_complete_search api/v2/discovered_hosts/index api/v2/discovered_hosts/show 	Host
submit_discovered_hosts	<ul style="list-style-type: none"> api/v2/discovered_hosts/facts api/v2/discovered_hosts/create 	Host

パーミッション名	アクション	リソースタイプ
auto_provision_discovered_hosts	<ul style="list-style-type: none"> discovered_hosts/auto_provision discovered_hosts/auto_provision_all api/v2/discovered_hosts/auto_provision api/v2/discovered_hosts/auto_provision_all 	Host
provision_discovered_hosts	<ul style="list-style-type: none"> discovered_hosts/edit discovered_hosts/update api/v2/discovered_hosts/update 	Host
edit_discovered_hosts	<ul style="list-style-type: none"> discovered_hosts/update_multiple_location discovered_hosts/select_multiple_organization discovered_hosts/update_multiple_organization discovered_hosts/select_multiple_location discovered_hosts/refresh_facts discovered_hosts/reboot discovered_hosts/reboot_all api/v2/discovered_hosts/refresh_facts api/v2/discovered_hosts/reboot api/v2/discovered_hosts/reboot_all 	Host
destroy_discovered_hosts	<ul style="list-style-type: none"> discovered_hosts/destroy discovered_hosts/submit_multiple_destroy discovered_hosts/multiple_destroy api/v2/discovered_hosts/destroy 	Host

パーMISSION名	アクション	リソースタイプ
view_discovery_rules	<ul style="list-style-type: none"> • discovery_rules/index • discovery_rules/show • discovery_rules/auto_complete_search • api/v2/discovery_rules/index • api/v2/discovery_rules/show 	DiscoveryRule
create_discovery_rules	<ul style="list-style-type: none"> • discovery_rules/new • discovery_rules/create • api/v2/discovery_rules/create 	DiscoveryRule
edit_discovery_rules	<ul style="list-style-type: none"> • discovery_rules/edit • discovery_rules/update • discovery_rules/enable • discovery_rules/disable • api/v2/discovery_rules/create • api/v2/discovery_rules/update 	DiscoveryRule
execute_discovery_rules	<ul style="list-style-type: none"> • discovery_rules/auto_provision • discovery_rules/auto_provision_all • api/v2/discovery_rules/auto_provision • api/v2/discovery_rules/auto_provision_all 	DiscoveryRule
destroy_discovery_rules	<ul style="list-style-type: none"> • discovery_rules/destroy • api/v2/discovery_rules/destroy 	DiscoveryRule

パーミッション名	アクション	リソースタイプ
view_domains	<ul style="list-style-type: none">domains/indexdomains/showdomains/auto_complete_searchapi/v1/domains/indexapi/v1/domains/showapi/v2/domains/indexapi/v2/domains/showapi/v2/parameters/indexapi/v2/parameters/show	
create_domains	<ul style="list-style-type: none">domains/newdomains/createapi/v1/domains/createapi/v2/domains/create	
edit_domains	<ul style="list-style-type: none">domains/editdomains/updateapi/v1/domains/updateapi/v2/domains/updateapi/v2/parameters/createapi/v2/parameters/updateapi/v2/parameters/destroyapi/v2/parameters/reset	
destroy_domains	<ul style="list-style-type: none">domains/destroyapi/v1/domains/destroyapi/v2/domains/destroy	

パーMISSION名	アクション	リソースタイプ
view_environments	<ul style="list-style-type: none">• environments/index• environments/show• environments/auto_complete_search• api/v1/environments/index• api/v1/environments/show• api/v2/environments/index• api/v2/environments/show	
create_environments	<ul style="list-style-type: none">• environments/new• environments/create• api/v1/environments/create• api/v2/environments/create	
edit_environments	<ul style="list-style-type: none">• environments/edit• environments/update• api/v1/environments/update• api/v2/environments/update	
destroy_environments	<ul style="list-style-type: none">• environments/destroy• api/v1/environments/destroy• api/v2/environments/destroy	

パーミッション名	アクション	リソースタイプ
import_environments	<ul style="list-style-type: none"> environments/import_environments environments/obsolete_and_new api/v1/environments/import_puppetclasses api/v2/environments/import_puppetclasses api/v1/smart_proxies/import_puppetclasses api/v2/smart_proxies/import_puppetclasses 	
view_external_usergroups	<ul style="list-style-type: none"> external_usergroups/index external_usergroups/show api/v2/external_usergroups/index api/v2/external_usergroups/show 	
create_external_usergroups	<ul style="list-style-type: none"> external_usergroups/new external_usergroups/create api/v2/external_usergroups/new api/v2/external_usergroups/create 	
edit_external_usergroups	<ul style="list-style-type: none"> external_usergroups/edit external_usergroups/update external_usergroups/refresh api/v2/external_usergroups/update api/v2/external_usergroups/refresh 	
destroy_external_usergroups	<ul style="list-style-type: none"> external_usergroups/destroy api/v2/external_usergroups/destroy 	

パーMISSION名	アクション	リソースタイプ
view_external_variables	<ul style="list-style-type: none">• lookup_keys/index• lookup_keys/show• lookup_keys/auto_complete_search• puppetclass_lookup_keys/index• puppetclass_lookup_keys/show• puppetclass_lookup_keys/auto_complete_search• variable_lookup_keys/index• variable_lookup_keys/show• variable_lookup_keys/auto_complete_search• lookup_values/index• api/v1/lookup_keys/index• api/v1/lookup_keys/show• api/v2/smart_variables/index• api/v2/smart_variables/show• api/v2/smart_class_parameters/index• api/v2/smart_class_parameters/show• api/v2/override_values/index• api/v2/override_values/show	

パーミッション名	アクション	リソースタイプ
create_external_variables	<ul style="list-style-type: none"> • lookup_keys/new • lookup_keys/create • puppetclass_lookup_keys/new • puppetclass_lookup_keys/create • variable_lookup_keys/new • variable_lookup_keys/create • lookup_values/create • api/v1/lookup_keys/create • api/v2/smart_variables/create • api/v2/smart_class_parameters/create • api/v2/override_values/create 	
edit_external_variables	<ul style="list-style-type: none"> • lookup_keys/edit • lookup_keys/update • puppetclass_lookup_keys/edit • puppetclass_lookup_keys/update • variable_lookup_keys/edit • variable_lookup_keys/update • lookup_values/create • lookup_values/update • lookup_values/destroy • api/v1/lookup_keys/update • api/v2/smart_variables/update • api/v2/smart_class_parameters/update • api/v2/override_values/create • api/v2/override_values/update • api/v2/override_values/destroy 	

パーMISSION名	アクション	リソースタイプ
destroy_external_variables	<ul style="list-style-type: none"> • lookup_keys/destroy • puppetclass_lookup_keys/destroy • variable_lookup_keys/destroy • lookup_values/destroy • api/v1/lookup_keys/destroy • api/v2/smart_variables/destroy • api/v2/smart_class_parameters/destroy • api/v2/override_values/create • api/v2/override_values/update • api/v2/override_values/destroy 	
view_facts	<ul style="list-style-type: none"> • facts/index • facts/show • fact_values/index • fact_values/show • fact_values/auto_complete_search • api/v1/fact_values/index • api/v1/fact_values/show • api/v2/fact_values/index • api/v2/fact_values/show 	
upload_facts	<ul style="list-style-type: none"> • api/v2/hosts/facts 	
view_filters	<ul style="list-style-type: none"> • filters/index • filters/auto_complete_search • api/v2/filters/index • api/v2/filters/show 	

パーミッション名	アクション	リソースタイプ
create_filters	<ul style="list-style-type: none"> filters/new filters/create api/v2/filters/create 	
edit_filters	<ul style="list-style-type: none"> filters/edit filters/update permissions/index api/v2/filters/update api/v2/permissions/index api/v2/permissions/show api/v2/permissions/resource_types 	
destroy_filters	<ul style="list-style-type: none"> filters/destroy api/v2/filters/destroy 	
view_arf_reports	<ul style="list-style-type: none"> arf_reports/index arf_reports/show arf_reports/parse_html arf_reports/show_html arf_reports/parse_bzip arf_reports/auto_complete_search api/v2/compliance/arf_reports/index api/v2/compliance/arf_reports/show compliance_hosts/show 	
destroy_arf_reports	<ul style="list-style-type: none"> arf_reports/destroy arf_reports/delete_multiple arf_reports/submit_delete_multiple api/v2/compliance/arf_reports/destroy 	

パーMISSION名	アクション	リソースタイプ
create_arf_reports	<ul style="list-style-type: none"> • api/v2/compliance/arf_reports/create 	
view_policies	<ul style="list-style-type: none"> • policies/index • policies/show • policies/parse • policies/auto_complete_search • policy_dashboard/index • compliance_dashboard/index • api/v2/compliance/policies/index • api/v2/compliance/policies/show • api/v2/compliance/policies/content 	ForemanOpenscap::Policy
edit_policies	<ul style="list-style-type: none"> • policies/edit • policies/update • policies/scap_content_selected • api/v2/compliance/policies/update 	ForemanOpenscap::Policy
create_policies	<ul style="list-style-type: none"> • policies/new • policies/create • api/v2/compliance/policies/create 	ForemanOpenscap::Policy
destroy_policies	<ul style="list-style-type: none"> • policies/destroy • api/v2/compliance/policies/destroy 	ForemanOpenscap::Policy
assign_policies	<ul style="list-style-type: none"> • policies/select_multiple_hosts • policies/update_multiple_hosts • policies/disassociate_multiple_hosts • policies/remove_policy_from_multiple_hosts 	ForemanOpenscap::Policy

パーミッション名	アクション	リソースタイプ
view_scap_contents	<ul style="list-style-type: none"> scap_contents/index scap_contents/show scap_contents/auto_complete_search api/v2/compliance/scap_contents/index api/v2/compliance/scap_contents/show 	ForemanOpenscap::ScapContent
view_scap_contents	<ul style="list-style-type: none"> scap_contents/index scap_contents/show scap_contents/auto_complete_search api/v2/compliance/scap_contents/index api/v2/compliance/scap_contents/show 	ForemanOpenscap::ScapContent
edit_scap_contents	<ul style="list-style-type: none"> scap_contents/edit scap_contents/update api/v2/compliance/scap_contents/update 	ForemanOpenscap::ScapContent
create_scap_contents	<ul style="list-style-type: none"> scap_contents/new scap_contents/create api/v2/compliance/scap_contents/create 	ForemanOpenscap::ScapContent
destroy_scap_contents	<ul style="list-style-type: none"> scap_contents/destroy api/v2/compliance/scap_contents/destroy 	ForemanOpenscap::ScapContent
edit_hosts	<ul style="list-style-type: none"> hosts/openscap_proxy_changed 	Host
edit_hostgroups	<ul style="list-style-type: none"> hostgroups/openscap_proxy_changed 	Host

パーMISSION名	アクション	リソースタイプ
view_job_templates	<ul style="list-style-type: none"> • job_templates/index • job_templates/show • job_templates/revision • job_templates/auto_complete_search • job_templates/auto_complete_job_category • job_templates/preview • job_templates/export • api/v2/job_templates/index • api/v2/job_templates/show • api/v2/job_templates/revision • api/v2/job_templates/export • api/v2/template_inputs/index • api/v2/template_inputs/show • api/v2/foreign_input_sets/index • api/v2/foreign_input_sets/show 	JobTemplate
create_job_templates	<ul style="list-style-type: none"> • job_templates/new • job_templates/create • job_templates/clone_template • job_templates/import • api/v2/job_templates/create • api/v2/job_templates/clone • api/v2/job_templates/import 	JobTemplate

パーミッション名	アクション	リソースタイプ
edit_job_templates	<ul style="list-style-type: none"> • job_templates/edit • job_templates/update • api/v2/job_templates/update • api/v2/template_inputs/create • api/v2/template_inputs/update • api/v2/template_inputs/destroy • api/v2/foreign_input_sets/create • api/v2/foreign_input_sets/update • api/v2/foreign_input_sets/destroy 	
edit_job_templates	<ul style="list-style-type: none"> • job_templates/edit • job_templates/update • api/v2/job_templates/update • api/v2/template_inputs/create • api/v2/template_inputs/update • api/v2/template_inputs/destroy • api/v2/foreign_input_sets/create • api/v2/foreign_input_sets/update • api/v2/foreign_input_sets/destroy 	
edit_remote_execution_features	<ul style="list-style-type: none"> • remote_execution_features/index • remote_execution_features/show • remote_execution_features/update • api/v2/remote_execution_features/index • api/v2/remote_execution_features/show • api/v2/remote_execution_features/update 	RemoteExecutionFeature

パーミッション名	アクション	リソースタイプ
destroy_job_templates	<ul style="list-style-type: none"> job_templates/destroy api/v2/job_templates/destroy 	JobTemplate
lock_job_templates	<ul style="list-style-type: none"> job_templates/lock job_templates/unlock 	JobTemplate
create_job_invocations	<ul style="list-style-type: none"> job_invocations/new job_invocations/create job_invocations/refresh job_invocations/rerun job_invocations/preview_hosts api/v2/job_invocations/create 	JobInvocation
view_job_invocations	<ul style="list-style-type: none"> job_invocations/index job_invocations/show template_invocations/show api/v2/job_invocations/index api/v2/job_invocations/show api/v2/job_invocations/output 	JobInvocation
execute_template_invocation		TemplateInvocation
filter_autocompletion_for_template_invocation	<ul style="list-style-type: none"> template_invocations/auto_complete_search job_invocations/show template_invocations/index 	TemplateInvocation

パーミッション名	アクション	リソースタイプ
view_foreman_tasks	<ul style="list-style-type: none"> • foreman_tasks/tasks/auto_complete_search • foreman_tasks/tasks/sub_tasks • foreman_tasks/tasks/index • foreman_tasks/tasks/show • foreman_tasks/api/tasks/bulk_search • foreman_tasks/api/tasks/show • foreman_tasks/api/tasks/index • foreman_tasks/api/tasks/summary 	ForemanTasks::Task
edit_foreman_tasks	<ul style="list-style-type: none"> • foreman_tasks/tasks/resume • foreman_tasks/tasks/unlock • foreman_tasks/tasks/force_unlock • foreman_tasks/tasks/cancel_step • foreman_tasks/api/tasks/bulk_resume 	ForemanTasks::Task
create_recurring_logics		ForemanTasks::RecurringLogic
view_recurring_logics	<ul style="list-style-type: none"> • foreman_tasks/recurring_logics/index • foreman_tasks/recurring_logics/show • foreman_tasks/api/recurring_logics/index • foreman_tasks/api/recurring_logics/show 	ForemanTasks::RecurringLogic
edit_recurring_logics	<ul style="list-style-type: none"> • foreman_tasks/recurring_logics/cancel • foreman_tasks/api/recurring_logics/cancel 	ForemanTasks::RecurringLogic

パーMISSION名	アクション	リソースタイプ
view_globals	<ul style="list-style-type: none"> • common_parameters/index • common_parameters/show • common_parameters/auto_complete_search • api/v1/common_parameters/index • api/v1/common_parameters/show • api/v2/common_parameters/index • api/v2/common_parameters/show 	
create_globals	<ul style="list-style-type: none"> • common_parameters/new • common_parameters/create • api/v1/common_parameters/create • api/v2/common_parameters/create 	
edit_globals	<ul style="list-style-type: none"> • common_parameters/edit • common_parameters/update • api/v1/common_parameters/update • api/v2/common_parameters/update 	
destroy_globals	<ul style="list-style-type: none"> • common_parameters/destroy • api/v1/common_parameters/destroy • api/v2/common_parameters/destroy 	
view_gpg_keys	<ul style="list-style-type: none"> • katello/gpg_keys/all • katello/gpg_keys/index • katello/gpg_keys/auto_complete_search • katello/api/v2/gpg_keys/index • katello/api/v2/gpg_keys/show 	Katello::GpgKey

パーミッション名	アクション	リソースタイプ
create_gpg_keys	<ul style="list-style-type: none"> katello/api/v2/gpg_keys/create 	Katello::GpgKey
edit_gpg_keys	<ul style="list-style-type: none"> katello/api/v2/gpg_keys/update katello/api/v2/gpg_keys/content 	Katello::GpgKey
destroy_gpg_keys	<ul style="list-style-type: none"> katello/api/v2/gpg_keys/destroy 	Katello::GpgKey
view_host_collections	<ul style="list-style-type: none"> katello/api/v2/host_collections/index katello/api/v2/host_collections/show katello/host_collections/auto_complete_search 	Katello::HostCollection
create_host_collections	<ul style="list-style-type: none"> katello/api/v2/host_collections/create katello/api/v2/host_collections/copy 	Katello::HostCollection
edit_host_collections	<ul style="list-style-type: none"> katello/api/v2/host_collections/update katello/api/v2/host_collections/add_systems katello/api/v2/host_collections/remove_systems 	Katello::HostCollection
destroy_host_collections	<ul style="list-style-type: none"> katello/api/v2/host_collections/destroy 	Katello::HostCollection
edit_classes	<ul style="list-style-type: none"> host_editing/edit_classes api/v2/host_classes/index api/v2/host_classes/create api/v2/host_classes/destroy 	
create_params	<ul style="list-style-type: none"> host_editing/create_params api/v2/parameters/create 	

パーMISSION名	アクション	リソースタイプ
edit_params	<ul style="list-style-type: none">• host_editing/edit_params• api/v2/parameters/update	
destroy_params	<ul style="list-style-type: none">• host_editing/destroy_params• api/v2/parameters/destroy• api/v2/parameters/reset	
view_hostgroups	<ul style="list-style-type: none">• hostgroups/index• hostgroups/show• hostgroups/auto_complete_search• api/v1/hostgroups/index• api/v1/hostgroups/show• api/v2/hostgroups/index• api/v2/hostgroups/show	

パーミッション名	アクション	リソースタイプ
create_hostgroups	<ul style="list-style-type: none">• hostgroups/new• hostgroups/create• hostgroups/clone• hostgroups/nest• hostgroups/process_hostgroup• hostgroups/architecture_selected• hostgroups/domain_selected• hostgroups/environment_selected• hostgroups/medium_selected• hostgroups/os_selected• hostgroups/use_image_selected• hostgroups/process_hostgroup• hostgroups/puppetclass_parameters• host/process_hostgroup• puppetclasses/parameters• api/v1/hostgroups/create• api/v1/hostgroups/clone• api/v2/hostgroups/create• api/v2/hostgroups/clone	

パーMISSION名	アクション	リソースタイプ
edit_hostgroups	<ul style="list-style-type: none"> • hostgroups/edit • hostgroups/update • hostgroups/architecture_selected • hostgroups/process_hostgroup • hostgroups/architecture_selected • hostgroups/domain_selected • hostgroups/environment_selected • hostgroups/medium_selected • hostgroups/os_selected • hostgroups/use_image_selected • hostgroups/process_hostgroup • hostgroups/puppetclass_parameters • host/process_hostgroup • puppetclasses/parameters • api/v1/hostgroups/update • api/v2/hostgroups/update • api/v2/parameters/create • api/v2/parameters/update • api/v2/parameters/destroy • api/v2/parameters/reset • api/v2/hostgroup_classes/index • api/v2/hostgroup_classes/create • api/v2/hostgroup_classes/destroy 	
destroy_hostgroups	<ul style="list-style-type: none"> • hostgroups/destroy • api/v1/hostgroups/destroy • api/v2/hostgroups/destroy 	
view_hosts	<ul style="list-style-type: none"> • hosts/index • hosts/show • hosts/errors 	

パーミッション名	アクション	リソースタイプ
	<ul style="list-style-type: none"> • hosts/active • hosts/out_of_sync • hosts/disabled • hosts/pending • hosts/vm • hosts/externalNodes • hosts/pxe_config • hosts/storeconfig_klasses • hosts/auto_complete_search • hosts/bmc • hosts/runtime • hosts/resources • hosts/templates • hosts/overview • hosts/nics • dashboard/OutOfSync • dashboard/errors • dashboard/active • unattended/host_template • unattended/hostgroup_template • api/v1/hosts/index • api/v1/hosts/show • api/v1/hosts/status • api/v2/hosts/index • api/v2/hosts/show • api/v2/hosts/status • api/v2/hosts/get_status • api/v2/hosts/vm_compute_attributes • api/v2/hosts/template • api/v2/interfaces/index • api/v2/interfaces/show • locations/mismatches • organizations/mismatches 	

パーMISSION名	アクション • hosts/puppet_environment_for_content view	リソースタイプ
	<ul style="list-style-type: none">• katello/api/v2/host_autocomplete/auto_complete_search• katello/api/v2/host_errata/index• katello/api/v2/host_errata/show• katello/api/v2/host_errata/auto_complete_search• katello/api/v2/host_subscriptions/index• katello/api/v2/host_subscriptions/events• katello/api/v2/host_subscriptions/product_content• katello/api/v2/hosts_bulk_actions/installable_errata• katello/api/v2/hosts_bulk_actions/available_incremental_updates• katello/api/v2/host_packages/index	

パーミッション名	アクション	リソースタイプ
create_hosts	<ul style="list-style-type: none"> • hosts/new • hosts/create • hosts/clone • hosts/architecture_selected • hosts/compute_resource_selected • hosts/domain_selected • hosts/environment_selected • hosts/hostgroup_or_environment_selected • hosts/medium_selected • hosts/os_selected • hosts/use_image_selected • hosts/process_hostgroup • hosts/process_taxonomy • hosts/current_parameters • hosts/puppetclass_parameters • hosts/template_used • hosts/interfaces • compute_resources/cluster_selected • compute_resources/template_selected • compute_resources/provider_selected • compute_resources/resource_pools • puppetclasses/parameters • subnets/freeip • interfaces/new • api/v1/hosts/create • api/v2/hosts/create • api/v2/interfaces/create • api/v2/tasks/index 	
edit_hosts	<ul style="list-style-type: none"> • hosts/edit • hosts/update 	

パーミッション名	アクション	リソースタイプ
	<ul style="list-style-type: none"> • hosts/multiple_actions • hosts/reset_multiple • hosts/submit_multiple_enable • hosts/select_multiple_hostgroup • hosts/select_multiple_environment • hosts/submit_multiple_disable • hosts/multiple_parameters • hosts/multiple_disable • hosts/multiple_enable • hosts/update_multiple_environment • hosts/update_multiple_hostgroup • hosts/update_multiple_parameters • hosts/toggle_manage • hosts/select_multiple_organization • hosts/update_multiple_organization • hosts/disassociate • hosts/multiple_disassociate • hosts/update_multiple_disassociate • hosts/select_multiple_owner • hosts/update_multiple_owner • hosts/select_multiple_power_state • hosts/update_multiple_power_state • hosts/select_multiple_puppet_proxy • hosts/update_multiple_puppet_proxy • hosts/select_multiple_puppet_ca_proxy • hosts/update_multiple_puppet_ca_proxy • hosts/select_multiple_location • hosts/update_multiple_location • hosts/architecture_selected • hosts/compute_resource_selected • hosts/domain_selected • hosts/environment_selected 	

パーミッション名	アクション • hosts/hostgroup_or_environment_selected	リソースタイプ
	<ul style="list-style-type: none"> • hosts/medium_selected • hosts/os_selected • hosts/use_image_selected • hosts/process_hostgroup • hosts/process_taxonomy • hosts/current_parameters • hosts/puppetclass_parameters • hosts/template_used • hosts/interfaces • compute_resources/associate • compute_resources/[:cluster_selected, :template_selected, :provider_selected, :resource_pools] • compute_resources_vms/associate • puppetclasses/parameters • subnets/freeip • interfaces/new • api/v1/hosts/update • api/v2/hosts/update • api/v2/hosts/disassociate • api/v2/interfaces/create • api/v2/interfaces/update • api/v2/interfaces/destroy • api/v2/compute_resources/associate • api/v2/hosts/host_collections • katello/api/v2/host_errata/apply • katello/api/v2/host_packages/install • katello/api/v2/host_packages/upgrade • katello/api/v2/host_packages/upgrade_all • katello/api/v2/host_packages/remove • katello/api/v2/host_subscriptions/auto_attach 	

パーミッション名	アクション	リソースタイプ
	<ul style="list-style-type: none"> • katello/api/v2/host_subscriptions/add_subscriptions • katello/api/v2/host_subscriptions/remove_subscriptions • katello/api/v2/host_subscriptions/content_override • katello/api/v2/hosts_bulk_actions/bulk_add_host_collections • katello/api/v2/hosts_bulk_actions/bulk_remove_host_collections • katello/api/v2/hosts_bulk_actions/install_content • katello/api/v2/hosts_bulk_actions/update_content • katello/api/v2/hosts_bulk_actions/remove_content • katello/api/v2/hosts_bulk_actions/environment_content_view 	
destroy_hosts	<ul style="list-style-type: none"> • hosts/destroy • hosts/multiple_actions • hosts/reset_multiple • hosts/multiple_destroy • hosts/submit_multiple_destroy • api/v1/hosts/destroy • api/v2/hosts/destroy • api/v2/interfaces/destroy • katello/api/v2/hosts_bulk_actions/destroy_hosts 	

パーミッション名	アクション	リソースタイプ
build_hosts	<ul style="list-style-type: none"> • hosts/setBuild • hosts/cancelBuild • hosts/multiple_build • hosts/submit_multiple_build • hosts/review_before_build • hosts/rebuild_config • hosts/submit_rebuild_config • tasks/show • api/v2/tasks/index • api/v2/hosts/rebuild_config 	
power_hosts	<ul style="list-style-type: none"> • hosts/power • api/v2/hosts/power 	
console_hosts	<ul style="list-style-type: none"> • hosts/console 	
ipmi_boot	<ul style="list-style-type: none"> • hosts/ipmi_boot • api/v2/hosts/boot 	
puppetrun_hosts	<ul style="list-style-type: none"> • hosts/puppetrun • hosts/multiple_puppetrun • hosts/update_multiple_puppetrun • api/v2/hosts/puppetrun 	
search_repository_image_search	<ul style="list-style-type: none"> • image_search/auto_complete_repository_name • image_search/auto_complete_image_tag • image_search/search_repository 	Docker/ImageSearch

パーMISSION名	アクション	リソースタイプ
view_images	<ul style="list-style-type: none"> images/index images/show images/auto_complete_search api/v1/images/index api/v1/images/show api/v2/images/index api/v2/images/show 	
create_images	<ul style="list-style-type: none"> images/new images/create api/v1/images/create api/v2/images/create 	
edit_images	<ul style="list-style-type: none"> images/edit images/update api/v1/images/update api/v2/images/update 	
destroy_images	<ul style="list-style-type: none"> images/destroy api/v1/images/destroy api/v2/images/destroy 	
view_lifecycle_environments	<ul style="list-style-type: none"> katello/api/v2/environments/index katello/api/v2/environments/show katello/api/v2/environments/paths katello/api/v2/environments/repositories katello/api/rhsm/candlepin_proxies/rhsm_index katello/environments/auto_complete_search 	Katello::KTEEnvironment

パーミッション名	アクション	リソースタイプ
create_lifecycle_environments	<ul style="list-style-type: none"> katello/api/v2/environments/create 	Katello::KTEEnvironment
edit_lifecycle_environments	<ul style="list-style-type: none"> katello/api/v2/environments/update 	Katello::KTEEnvironment
destroy_lifecycle_environments	<ul style="list-style-type: none"> katello/api/v2/environments/destroy 	Katello::KTEEnvironment
promote_or_remove_content_views_to_environments		Katello::KTEEnvironment
view_locations	<ul style="list-style-type: none"> locations/index locations/show locations/auto_complete_search api/v1/locations/index api/v1/locations/show api/v2/locations/index api/v2/locations/show 	
create_locations	<ul style="list-style-type: none"> locations/new locations/create locations/clone_taxonomy locations/step2 locations/nest api/v1/locations/create api/v2/locations/create 	

パーMISSION名	アクション	リソースタイプ
edit_locations	<ul style="list-style-type: none"> • locations/edit • locations/update • locations/import_mismatches • locations/parent_taxonomy_selected • api/v1/locations/update • api/v2/locations/update 	
destroy_locations	<ul style="list-style-type: none"> • locations/destroy • api/v1/locations/destroy • api/v2/locations/destroy 	
assign_locations	<ul style="list-style-type: none"> • locations/assign_all_hosts • locations/assign_hosts • locations/assign_selected_hosts 	
view_mail_notifications	<ul style="list-style-type: none"> • mail_notifications/index • mail_notifications/auto_complete_search • mail_notifications/show • api/v2/mail_notifications/index • api/v2/mail_notifications/show 	
view_media	<ul style="list-style-type: none"> • media/index • media/show • media/auto_complete_search • api/v1/media/index • api/v1/media/show • api/v2/media/index • api/v2/media/show 	

パーミッション名	アクション	リソースタイプ
create_media	<ul style="list-style-type: none">• media/new• media/create• api/v1/media/create• api/v2/media/create	
edit_media	<ul style="list-style-type: none">• media/edit• media/update• api/v1/media/update• api/v2/media/update	
destroy_media	<ul style="list-style-type: none">• media/destroy• api/v1/media/destroy• api/v2/media/destroy	
view_models	<ul style="list-style-type: none">• models/index• models/show• models/auto_complete_search• api/v1/models/index• api/v1/models/show• api/v2/models/index• api/v2/models/show	
create_models	<ul style="list-style-type: none">• models/new• models/create• api/v1/models/create• api/v2/models/create	

パーMISSION名	アクション	リソースタイプ
edit_models	<ul style="list-style-type: none"> models/edit models/update api/v1/models/update api/v2/models/update 	
destroy_models	<ul style="list-style-type: none"> models/destroy api/v1/models/destroy api/v2/models/destroy 	
view_operatingsystems	<ul style="list-style-type: none"> operatingsystems/index operatingsystems/show operatingsystems/bootfiles operatingsystems/auto_complete_search api/v1/operatingsystems/index api/v1/operatingsystems/show api/v1/operatingsystems/bootfiles api/v2/operatingsystems/index api/v2/operatingsystems/show api/v2/operatingsystems/bootfiles api/v2/os_default_templates/index api/v2/os_default_templates/show 	
create_operatingsystems	<ul style="list-style-type: none"> operatingsystems/new operatingsystems/create api/v1/operatingsystems/create api/v2/operatingsystems/create api/v2/os_default_templates/create 	

パーミッション名	アクション	リソースタイプ
edit_operatingsystems	<ul style="list-style-type: none"> operatingsystems/edit operatingsystems/update api/v1/operatingsystems/update api/v2/operatingsystems/update api/v2/parameters/create api/v2/parameters/update api/v2/parameters/destroy api/v2/parameters/reset api/v2/os_default_templates/create api/v2/os_default_templates/update api/v2/os_default_templates/destroy 	
destroy_operatingsystems	<ul style="list-style-type: none"> operatingsystems/destroy api/v1/operatingsystems/destroy api/v2/operatingsystems/destroy api/v2/os_default_templates/create 	
view_organizations	<ul style="list-style-type: none"> organizations/index organizations/show organizations/auto_complete_search api/v1/organizations/index api/v1/organizations/show api/v2/organizations/index api/v2/organizations/show katello/api/v2/organizations/index katello/api/v2/organizations/show katello/api/v2/organizations/redhat_provider katello/api/v2/organizations/download_debug_certificate katello/api/v2/tasks/index 	

パーMISSION名	アクション	リソースタイプ
create_organizations	<ul style="list-style-type: none"> • organizations/new • organizations/create • organizations/clone_taxonomy • organizations/step2 • organizations/nest • api/v1/organizations/create • api/v2/organizations/create • katello/api/v2/organizations/create 	
edit_organizations	<ul style="list-style-type: none"> • organizations/edit • organizations/update • organizations/import_mismatches • organizations/parent_taxonomy_selected • api/v1/organizations/update • api/v2/organizations/update • katello/api/v2/organizations/update • katello/api/v2/organizations/autoattach_subscriptions 	
destroy_organizations	<ul style="list-style-type: none"> • organizations/destroy • api/v1/organizations/destroy • api/v2/organizations/destroy • katello/api/v2/organizations/destroy 	
assign_organizations	<ul style="list-style-type: none"> • organizations/assign_all_hosts • organizations/assign_hosts • organizations/assign_selected_hosts 	

パーミッション名	アクション	リソースタイプ
view_ptables	<ul style="list-style-type: none">• ptables/index• ptables/show• ptables/auto_complete_search• ptables/revision• ptables/preview• api/v1/ptables/index• api/v1/ptables/show• api/v2/ptables/index• api/v2/ptables/show• api/v2/ptables/revision	
create_ptables	<ul style="list-style-type: none">• ptables/new• ptables/create• ptables/clone_template• api/v1/ptables/create• api/v2/ptables/create• api/v2/ptables/clone	
edit_ptables	<ul style="list-style-type: none">• ptables/edit• ptables/update• api/v1/ptables/update• api/v2/ptables/update	
destroy_ptables	<ul style="list-style-type: none">• ptables/destroy• api/v1/ptables/destroy• api/v2/ptables/destroy	

パーMISSION名	アクション	リソースタイプ
lock_ptables	<ul style="list-style-type: none"> • ptables/lock • ptables/unlock • api/v2/ptables/lock • api/v2/ptables/unlock 	
view_plugins	<ul style="list-style-type: none"> • plugins/index • api/v2/plugins/index 	
view_products	<ul style="list-style-type: none"> • katello/products/auto_complete • katello/products/auto_complete_search • katello/api/v2/products/index • katello/api/v2/products/show • katello/api/v2/repositories/index • katello/api/v2/repositories/show • katello/api/v2/packages/index • katello/api/v2/packages/show • katello/api/v2/distributions/index • katello/api/v2/distributions/show • katello/api/v2/package_groups/index • katello/api/v2/package_groups/show • katello/api/v2/errata/index • katello/api/v2/errata/show • katello/api/v2/puppet_modules/index • katello/api/v2/puppet_modules/show • katello/errata/short_details • katello/errata/auto_complete • katello/packages/details • katello/packages/auto_complete • katello/puppet_modules/show • katello/repositories/auto_complete_library 	Katello::Product

パーミッション名	アクション	リソースタイプ
	<ul style="list-style-type: none"> • katello/repositories/repository_types • katello/content_search/index • katello/content_search/products • katello/content_search/repos • katello/content_search/packages • katello/content_search/errata • katello/content_search/puppet_modules • katello/content_search/packages_items • katello/content_search/errata_items • katello/content_search/puppet_modules_items • katello/content_search/repo_packages • katello/content_search/repo_errata • katello/content_search/repo_puppet_modules • katello/content_search/repo_compare_errata • katello/content_search/repo_compare_packages • katello/content_search/repo_compare_puppet_modules 	
create_products	<ul style="list-style-type: none"> • katello/api/v2/products/create • katello/api/v2/repositories/create 	Katello::Product

パーMISSION名	アクション	リソースタイプ
edit_products	<ul style="list-style-type: none"> katello/api/v2/products/update katello/api/v2/repositories/update katello/api/v2/repositories/remove_content katello/api/v2/repositories/import_uploads katello/api/v2/repositories/upload_content katello/api/v2/products_bulk_actions/update_sync_plans katello/api/v2/content_uploads/create katello/api/v2/content_uploads/update katello/api/v2/content_uploads/destroy katello/api/v2/organizations/repo_discover katello/api/v2/organizations/cancel_repo_discover 	Katello::Product
destroy_products	<ul style="list-style-type: none"> katello/api/v2/products/destroy katello/api/v2/repositories/destroy katello/api/v2/products_bulk_actions/destroy_products katello/api/v2/repositories_bulk_actions/destroy_repositories 	Katello::Product

パーミッション名	アクション	リソースタイプ
sync_products	<ul style="list-style-type: none">katello/api/v2/products/synckatello/api/v2/repositories/synckatello/api/v2/products_bulk_actions/sync_productskatello/api/v2/repositories_bulk_actions/sync_repositorieskatello/api/v2/sync/indexkatello/api/v2/sync_plans/synckatello/sync_management/indexkatello/sync_management/sync_statuskatello/sync_management/product_statuskatello/sync_management/synckatello/sync_management/destroy	Katello::Product
export_products	<ul style="list-style-type: none">katello/api/v2/repositories/export	Katello::Product

パーMISSION名	アクション	リソースタイプ
view_provisioning_templates	<ul style="list-style-type: none"> • provisioning_templates/index • provisioning_templates/show • provisioning_templates/revision • provisioning_templates/auto_complete_search • provisioning_templates/preview • api/v1/config_templates/index • api/v1/config_templates/show • api/v1/config_templates/revision • api/v2/config_templates/index • api/v2/config_templates/show • api/v2/config_templates/revision • api/v2/provisioning_templates/index • api/v2/provisioning_templates/show • api/v2/provisioning_templates/revision • api/v2/template_combinations/index • api/v2/template_combinations/show • api/v1/template_kinds/index • api/v2/template_kinds/index 	
create_provisioning_templates	<ul style="list-style-type: none"> • provisioning_templates/new • provisioning_templates/create • provisioning_templates/clone_template • api/v1/config_templates/create • api/v2/config_templates/create • api/v2/config_templates/clone • api/v2/provisioning_templates/create • api/v2/provisioning_templates/clone • api/v2/template_combinations/create 	

パーミッション名	アクション	リソースタイプ
edit_provisioning_templates	<ul style="list-style-type: none"> provisioning_templates/edit provisioning_templates/update api/v1/config_templates/update api/v2/config_templates/update api/v2/provisioning_templates/update api/v2/template_combinations/update 	
destroy_provisioning_templates	<ul style="list-style-type: none"> provisioning_templates/destroy api/v1/config_templates/destroy api/v2/config_templates/destroy api/v2/provisioning_templates/destory api/v2/template_combinations/destory 	
deploy_provisioning_templates	<ul style="list-style-type: none"> provisioning_templates/build_pxe_default api/v1/config_templates/build_pxe_default api/v2/config_templates/build_pxe_default api/v2/provisioning_templates/build_pxe_default 	
lock_provisioning_templates	<ul style="list-style-type: none"> provisioning_templates/lock provisioning_templates/unlock api/v2/config_templates/lock api/v2/config_templates/unlock api/v2/provisioning_templates/lock api/v2/provisioning_templates/unlock 	
user_logout	<ul style="list-style-type: none"> users/logout 	

パーMISSION名	アクション	リソースタイプ
my_account	<ul style="list-style-type: none"> • users/edit • katello/api/v2/tasks/show 	
api_status	<ul style="list-style-type: none"> • api/v1/home/status • api/v2/home/status 	
view_puppetclasses	<ul style="list-style-type: none"> • puppetclasses/index • puppetclasses/show • puppetclasses/auto_complete_search • api/v1/puppetclasses/index • api/v1/puppetclasses/show • api/v2/puppetclasses/index • api/v2/puppetclasses/show • api/v1/lookup_keys/index • api/v1/lookup_keys/show • api/v2/smart_variables/index • api/v2/smart_variables/show • api/v2/smart_class_parameters/index • api/v2/smart_class_parameters/show 	
create_puppetclasses	<ul style="list-style-type: none"> • puppetclasses/new • puppetclasses/create • api/v1/puppetclasses/create • api/v2/puppetclasses/create 	

パーミッション名	アクション	リソースタイプ
edit_puppetclasses	<ul style="list-style-type: none"> • puppetclasses/edit • puppetclasses/update • puppetclasses/override • api/v1/puppetclasses/update • api/v2/puppetclasses/update • api/v1/lookup_keys/create • api/v1/lookup_keys/update • api/v1/lookup_keys/destroy • api/v2/smart_variables/create • api/v2/smart_variables/update • api/v2/smart_variables/destroy • api/v2/smart_class_parameters/create • api/v2/smart_class_parameters/update • api/v2/smart_class_parameters/destroy 	
destroy_puppetclasses	<ul style="list-style-type: none"> • puppetclasses/destroy • api/v1/puppetclasses/destroy • api/v2/puppetclasses/destroy 	
import_puppetclasses	<ul style="list-style-type: none"> • puppetclasses/import_environments • puppetclasses/obsolete_and_new • api/v1/environments/import_puppetclasses • api/v2/environments/import_puppetclasses • api/v1/smart_proxies/import_puppetclasses • api/v2/smart_proxies/import_puppetclasses 	

パーMISSION名	アクション	リソースタイプ
view_realms	<ul style="list-style-type: none"> • realms/index • realms/show • realms/auto_complete_search • api/v2/realms/index • api/v2/realms/show 	
create_realms	<ul style="list-style-type: none"> • realms/new • realms/create • api/v2/realms/create 	
edit_realms	<ul style="list-style-type: none"> • realms/edit • realms/update • api/v2/realms/update 	
destroy_realms	<ul style="list-style-type: none"> • realms/destroy • api/v2/realms/destroy 	
view_search	<ul style="list-style-type: none"> • redhat_access/search/index 	
view_cases	<ul style="list-style-type: none"> • redhat_access/cases/index • redhat_access/cases/create 	
attachments	<ul style="list-style-type: none"> • redhat_access/attachments/index • redhat_access/attachments/create 	
設定	<ul style="list-style-type: none"> • redhat_access/configuration/index 	
app_root	<ul style="list-style-type: none"> • redhat_access/redhat_access/index 	

パーミッション名	アクション	リソースタイプ
view_log_viewer	<ul style="list-style-type: none"> redhat_access/logviewer/index 	
ログ	<ul style="list-style-type: none"> redhat_access/logs/index 	
rh_telemetry_api	<ul style="list-style-type: none"> redhat_access/api/telemetry_api/proxy redhat_access/api/telemetry_api/connection_status 	
rh_telemetry_view	<ul style="list-style-type: none"> redhat_access/analytics_dashboard/index 	
rh_telemetry_configurations	<ul style="list-style-type: none"> redhat_access/telemetry_configurations/show redhat_access/telemetry_configurations/update 	
view_registries	<ul style="list-style-type: none"> registries/index registries/show api/v2/registries/index api/v2/registries/show 	DockerRegistry
view_registries	<ul style="list-style-type: none"> registries/index registries/show api/v2/registries/index api/v2/registries/show 	DockerRegistry

パーMISSION名	アクション	リソースタイプ
create_registries	<ul style="list-style-type: none"> registries/new registries/create registries/update registries/edit api/v2/registries/create api/v2/registries/update 	DockerRegistry
destroy_registries	<ul style="list-style-type: none"> registries/destroy api/v2/registries/destroy 	DockerRegistry
view_roles	<ul style="list-style-type: none"> roles/index roles/auto_complete_search api/v2/roles/index api/v2/roles/show 	
create_roles	<ul style="list-style-type: none"> roles/new roles/create roles/clone api/v2/roles/create 	
edit_roles	<ul style="list-style-type: none"> roles/edit roles/update api/v2/roles/update 	
destroy_roles	<ul style="list-style-type: none"> roles/destroy api/v2/roles/destroy 	
access_settings	<ul style="list-style-type: none"> home/settings 	

パーミッション名	アクション	リソースタイプ
view_smart_proxies	<ul style="list-style-type: none">• smart_proxies/index• smart_proxies/ping• smart_proxies/auto_complete_search• smart_proxies/version• smart_proxies/show• smart_proxies/plugin_version• smart_proxies/tftp_server• smart_proxies/puppet_environments• smart_proxies/puppet_dashboard• smart_proxies/log_pane• smart_proxies/failed_modules• smart_proxies/errors_card• smart_proxies/modules_card• api/v1/smart_proxies/index• api/v1/smart_proxies/show• api/v2/smart_proxies/index• api/v2/smart_proxies/show• api/v2/smart_proxies/version• api/v2/smart_proxies/log	
create_smart_proxies	<ul style="list-style-type: none">• smart_proxies/new• smart_proxies/create• api/v1/smart_proxies/create• api/v2/smart_proxies/create	

パーMISSION名	アクション	リソースタイプ
edit_smart_proxies	<ul style="list-style-type: none"> • smart_proxies/edit • smart_proxies/update • smart_proxies/refresh • smart_proxies/expire_logs • api/v1/smart_proxies/update • api/v1/smart_proxies/refresh • api/v2/smart_proxies/update • api/v2/smart_proxies/refresh 	
destroy_smart_proxies	<ul style="list-style-type: none"> • smart_proxies/destroy • api/v1/smart_proxies/destroy • api/v2/smart_proxies/destroy 	
view_smart_proxies_autosign	<ul style="list-style-type: none"> • autosign/index • autosign/show • autosign/counts • api/v1/autosign/index • api/v2/autosign/index 	
create_smart_proxies_autosign	<ul style="list-style-type: none"> • autosign/new • autosign/create 	
destroy_smart_proxies_autosign	<ul style="list-style-type: none"> • autosign/destroy 	
view_smart_proxies_puppetca	<ul style="list-style-type: none"> • puppetca/index • puppetca/counts • puppetca/expiry 	
edit_smart_proxies_puppetca	<ul style="list-style-type: none"> • puppetca/update 	

パーミッション名	アクション	リソースタイプ
destroy_smart_proxies_puppetca	<ul style="list-style-type: none">• puppetca/destroy	
view_statistics	<ul style="list-style-type: none">• statistics/index• api/v1/statistics/index• api/v2/statistics/index	
view_subnets	<ul style="list-style-type: none">• subnets/index• subnets/show• subnets/auto_complete_search• api/v1/subnets/index• api/v1/subnets/show• api/v2/subnets/index• api/v2/subnets/show	
create_subnets	<ul style="list-style-type: none">• subnets/new• subnets/create• api/v1/subnets/create• api/v2/subnets/create	
edit_subnets	<ul style="list-style-type: none">• subnets/edit• subnets/update• api/v1/subnets/update• api/v2/subnets/update	
destroy_subnets	<ul style="list-style-type: none">• subnets/destroy• api/v1/subnets/destroy• api/v2/subnets/destroy	

パーMISSION名	アクション	リソースタイプ
import_subnets	<ul style="list-style-type: none"> • subnets/import • subnets/create_multiple 	
view_subscriptions	<ul style="list-style-type: none"> • katello/api/v2/subscriptions/index • katello/api/v2/subscriptions/show • katello/api/v2/subscriptions/available • katello/api/v2/subscriptions/manifest_history • katello/api/v2/subscriptions/auto_complete_search • katello/api/v2/repository_sets/index • katello/api/v2/repository_sets/show • katello/api/v2/repository_sets/available_repositories 	Organization
attach_subscriptions	<ul style="list-style-type: none"> • katello/api/v2/subscriptions/create 	Organization
unattach_subscriptions	<ul style="list-style-type: none"> • katello/api/v2/subscriptions/destroy 	Organization
import_manifest	<ul style="list-style-type: none"> • katello/products/available_repositories • katello/products/toggle_repository • katello/providers/redhat_provider • katello/providers/redhat_provider_tab • katello/api/v2/subscriptions/upload • katello/api/v2/subscriptions/refresh_manifest • katello/api/v2/repository_sets/enable • katello/api/v2/repository_sets/disable 	Organization
delete_manifest	<ul style="list-style-type: none"> • katello/api/v2/subscriptions/delete_manifest 	Organization

パーミッション名	アクション	リソースタイプ
view_sync_plans	<ul style="list-style-type: none"> katello/sync_plans/all katello/sync_plans/index katello/sync_plans/auto_complete_search katello/api/v2/sync_plans/index katello/api/v2/sync_plans/show katello/api/v2/sync_plans/add_products katello/api/v2/sync_plans/remove_products katello/api/v2/sync_plans/available_products katello/api/v2/products/index 	Katello::SyncPlan
create_sync_plans	<ul style="list-style-type: none"> katello/api/v2/sync_plans/create 	Katello::SyncPlan
edit_sync_plans	<ul style="list-style-type: none"> katello/api/v2/sync_plans/update 	Katello::SyncPlan
destroy_sync_plans	<ul style="list-style-type: none"> katello/api/v2/sync_plans/destroy 	Katello::SyncPlan
view_tasks	<ul style="list-style-type: none"> trends/show 	
view_trends	<ul style="list-style-type: none"> trends/index trends/show 	
create_trends	<ul style="list-style-type: none"> trends/new trends/create 	
edit_trends	<ul style="list-style-type: none"> trends/edit trends/update 	

パーMISSION名	アクション	リソースタイプ
destroy_trends	<ul style="list-style-type: none"> trends/destroy 	
update_trends	<ul style="list-style-type: none"> trends/count 	
my_organizations	<ul style="list-style-type: none"> katello/api/rhsm/candlepin_proxies/list_owners 	
view_usergroups	<ul style="list-style-type: none"> usergroups/index usergroups/show usergroups/auto_complete_search api/v1/usergroups/index api/v1/usergroups/show api/v2/usergroups/index api/v2/usergroups/show 	
create_usergroups	<ul style="list-style-type: none"> usergroups/new usergroups/create api/v1/usergroups/create api/v2/usergroups/create 	
edit_usergroups	<ul style="list-style-type: none"> usergroups/edit usergroups/update api/v1/usergroups/update api/v2/usergroups/update 	
destroy_usergroups	<ul style="list-style-type: none"> usergroups/destroy api/v1/usergroups/destroy api/v2/usergroups/destroy 	

パーミッション名	アクション	リソースタイプ
view_users	<ul style="list-style-type: none">• users/index• users/show• users/auto_complete_search• api/v1/users/index• api/v1/users/show• api/v2/users/index• api/v2/users/show	
create_users	<ul style="list-style-type: none">• users/new• users/create• users/auth_source_selected• api/v1/users/create• api/v2/users/create	
edit_users	<ul style="list-style-type: none">• users/edit• users/update• users/auth_source_selected• users/test_mail• api/v1/users/update• api/v2/users/update	
destroy_users	<ul style="list-style-type: none">• users/destroy• api/v1/users/destroy• api/v2/users/destroy	