



Red Hat Satellite 6.14

API ガイド

Satellite REST API を使用したカスタムアプリケーションや統合の開発

Red Hat Satellite 6.14 API ガイド

Satellite REST API を使用したカスタムアプリケーションや統合の開発

Red Hat Satellite Documentation Team

satellite-doc-list@redhat.com

法律上の通知

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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, the Red Hat 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 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 Representational State Transfer (REST) API ガイドでは、REST API のコンセプトを説明し、さまざまな要求タイプの使用例を紹介します。本書は、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、サードパーティーのアプリケーションと Red Hat Satellite を統合するための基本事項を説明します。

目次

RED HAT ドキュメントへのフィードバック (英語のみ)	3
第1章 概要	4
1.1. RED HAT SATELLITE API の概要	4
1.2. 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
2.2. JSON 応答形式の理解	9
第3章 API 呼び出しの認証	12
3.1. SSL 認証の概要	12
3.2. HTTP 認証の概要	12
3.3. PERSONAL ACCESS TOKEN 認証の概要	13
3.4. OAUTH 認証の概要	14
第4章 異なる言語での API 要求	16
4.1. CURL を使用した API 要求	16
4.2. RUBY を使用した API 要求	19
4.3. PYTHON を使用した API 要求	22
第5章 RED HAT SATELLITE API の使用	28
5.1. ホストの使用方法	28
5.2. ライフサイクル環境の使用	31
5.3. SATELLITE SERVER へのコンテンツのアップロード	35
5.4. エラータのホストまたはホストコレクションへの適用	39
5.5. 詳細検索の使用	40
5.6. ページネーション制御のある検索の使用	41
5.7. スマートクラスのオーバーライド	42
5.8. 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	43
5.9. OPENS CAP レポートの削除	46
5.10. SATELLITE API での PULP の使用	48
付録A API 応答コード	51
付録B API パーMISSIONの表	52

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに対するご意見をお聞かせください。ドキュメントの改善点があればお知らせください。

Bugzilla でチケットを作成することでフィードバックを送信できます。

1. [Bugzilla](#) のWeb サイトに移動します。
2. **Component** フィールドで、**Documentation** を使用します。
3. **Description** フィールドに、ドキュメントの改善に関するご意見を記入してください。ドキュメントの該当部分へのリンクも追加してください。
4. **Submit Bug** をクリックします。

第1章 概要

Red Hat Satellite には Representational State Transfer (REST) API が含まれています。ソフトウェア開発者やシステム管理者は、この API を使用することで、標準の Web インターフェイス以外で 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を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。

1.1. RED HAT SATELLITE API の概要

REST API の使用すると、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 プロトコルをサポートする各種プログラミング言語、フレームワークまたはシステムで API を使用できます。
- 自己記述型: 詳しい情報の多くはランタイム時に確認できるので、クライアントアプリケーションでは、必要な Red Hat Satellite インフラストラクチャーの情報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。

REST API を使用して、以下のタスクを実行できます。

- 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

Satellite Server のアップグレードを準備する時に、使用するスクリプトに含まれる Satellite API コマンドが最新であることを確認してください。API コマンドは、Satellite のバージョンごとに異なります。

1.2. HAMMER CLI ツールと SATELLITE API の比較

多くのタスクで、Hammer および Satellite API の両方を使用できます。Satellite API を人間でも簡単に使用できるように、Hammer をインターフェイスとして使用できます。Satellite API への使いやすいインターフェイスとして使用できます。たとえば、API 呼び出しへの応答をテストしてからスクリプトに適用するには、**--debug** オプションを使用して対象の Hammer の問題 (**hammer --debug organization list**) が含まれる API 呼び出しを検出します。

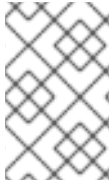
各 Hammer コマンドはバックグラウンドで、先に API へのバインディングを確立してから要求を送信します。この動作は、大量の Hammer コマンドを順に実行する場合に、パフォーマンスに影響を与える可能性があります。一方、API コマンドを使用するスクリプトは、Satellite API と直接通信します。

Hammer コマンドは API の変更を自動的に反映しますが、API コマンドを使用するスクリプトは手動で更新する必要がある点に注意してください。詳細は、[Hammer CLI ガイド](#) を参照してください。

第2章 API リファレンス

包括的な API リファレンスは、お使いの Satellite Server でご確認いただけます (<https://satellite.example.com/apidoc/v2.html>)。Satellite 6 API のバージョン 1 と 2 が利用できますが、Red Hat ではバージョン 2 のみをサポートする点に注意してください。

2.1. API 構文の理解



注記

以下の要求例では、**python3** を使用して Satellite Server からの応答をフォーマットしています。RHEL 7 およびそれ以前の一部のシステムでは、**python3** の代わりに **python** を使用する必要があります。

Satellite に含まれている API リファレンスは、HTTP 動詞の後に API ルートまたはパスを記載します。

HTTP_VERB API_ROUTE

API を使用するには、**curl** コマンド構文と当リファレンスドキュメントの API ルートを使用して、コマンドを構築します。

```
$ curl --request HTTP_VERB \
--insecure \
--user sat_username:sat_password \
--data @file.json \
--header "Accept:application/json" \
--header "Content-Type:application/json" \
--output file
API_ROUTE \
| python3 -m json.tool
```

- 1 API 呼び出しに **curl** を使用するには、HTTP 動詞に **--request** オプションを指定します。(例: **--request POST**)。
- 2 **--insecure** オプションを追加して、SSL ピア証明書の検証確認をスキップします。
- 3 **--user** オプションを使用してユーザー認証を指定します。
- 4 **POST** および **PUT** 要求の場合は、**--data** オプションを使用して、JSON 形式のデータを渡します。詳細は、「[API 要求への JSON データの指定](#)」を参照してください。
- 5 6 **--data** オプションを指定して JSON データを渡す場合は、**--header** オプションを指定して以下のヘッダーを指定する必要があります。詳細は、「[API 要求への JSON データの指定](#)」を参照してください。
- 7 Satellite Server からコンテンツをダウンロードする場合に、**--output** オプションを使用して、出力ファイルを指定します。
- 8 https://satellite.example.com/katello/api/activation_keys 形式の API ルートを使用します。Satellite 6 では、API のバージョン 2 がデフォルトです。したがって、API 呼び出しの URL に **v2** を指定する必要はありません。

- 9 Python **json.tool** モジュールに出力をリダイレクトして、出力の可読性を向上します。

2.1.1. HTTP 動詞 (GET) の使用

HTTP 動詞 (GET) を使用して、既存のエントリまたはリソースに関するデータを API から取得します。

例

この例では、Satellite ホストの数を要求します。

要求例:

```
$ curl --request GET --insecure --user sat_username:sat_password \  
https://satellite.example.com/api/hosts | python3 -m json.tool
```

応答例:

```
{  
  "total": 2,  
  "subtotal": 2,  
  "page": 1,  
  "per_page": 20,  
  "search": null,  
  "sort": {  
    "by": null,  
    "order": null  
  },  
  "results":  
output truncated
```

API からの応答により、結果数が合計 2 つ、これらは結果の 1 ページ目に表示され、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。詳細は、「[JSON 応答形式の理解](#)」を参照してください。

2.1.2. HTTP 動詞 (POST) の使用

HTTP 動詞 (POST) を使用して、データを API に送信してエントリまたはリソースを作成します。JSON 形式でデータを送信する必要があります。詳細は、「[API 要求への JSON データの指定](#)」を参照してください。

例

この例では、アクティベーションキーを作成します。

1. 以下のコンテンツを含めて、**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"  
}
```

2. **activation-key.json** ファイルにデータを適用してアクティベーションキーを作成します。
要求例:

```
$ curl --header "Accept:application/json" \  
--header "Content-Type:application/json" --request POST \  
https://satellite.example.com/api/activation_keys
```

```
--user sat_username:sat_password --insecure \  
--data @activation-key.json \  
https://satellite.example.com/katello/api/activation_keys \  
| python3 -m json.tool
```

応答例:

```
{  
  "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  
  }  
}
```

3. 新しいアクティベーションキーが存在することを確認します。Satellite Web UI で、**コンテンツ** > **アクティベーションキー** に移動して、アクティベーションキーを表示します。

2.1.3. HTTP 動詞 (PUT) の使用

HTTP 動詞 (PUT) を使用して、既存の値を変更するか、既存のリソースに追加します。JSON 形式でデータを送信する必要があります。詳細は、「[API 要求への JSON データの指定](#)」を参照してください。

例

この例では、1つ前の例で作成した **TestKey** アクティベーションキーを更新します。

1. 先ほど作成した **activation-key.json** ファイルを以下のように編集します。

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

2. JSON ファイルの変更を適用します。

要求例:

```
$ curl --header "Accept:application/json" \  
--header "Content-Type:application/json" --request PUT \  
--user sat_username:sat_password --insecure \  
--data @activation-key.json \  
https://satellite.example.com/katello/api/activation_keys/2 \  
| python3 -m json.tool
```

応答例:

```
{  
  "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  
  }  
}
```

3. Satellite Web UI で、**コンテンツ > アクティベーションキー** に移動して、変更を確認します。

2.1.4. 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。

例

この例では、ID が 2 の **TestKey** アクティベーションキーを削除します。

要求例:

```
$ curl --header "Accept:application/json" \
--header "Content-Type:application/json" --request DELETE \
--user sat_username:sat_password --insecure \
https://satellite.example.com/katello/api/activation_keys/2 \
| python3 -m json.tool
```

応答例:

```
output omitted
"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"
  }
}
output truncated
```

2.1.5. 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、以下のように、API リファレンスで使用する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

API への呼び出しは、JSON 形式の結果を返します。API 呼び出しは、単一オプションの応答または、応答コレクションの結果を返します。



注記

以下の要求例では、**python3** を使用して Satellite Server からの応答をフォーマットしています。RHEL 7 およびそれ以前の一部のシステムでは、**python3** の代わりに **python** を使用する必要があります。

単一オブジェクトの JSON 応答形式

単一オブジェクトで作業するには、単一オブジェクトの JSON 応答を使用します。単一のオブジェクトに対する API 要求では、オブジェクトの一位識別子 **:id** が必要です。

これは、ID が 23 の Satellite ドメインに対する単一オブジェクトの形式例です。

要求例:

```
$ curl --request GET --insecure --user sat_username:sat_password \  
https://satellite.example.com/api/domains/23 | python3 -m json.tool
```

応答例:

```
{  
  "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"  
}
```

コレクション用の JSON 応答形式

コレクションは、ホストとドメインなどのオブジェクトリストです。JSON 応答コレクションの形式は、メタデータのフィールドセクションと結果セクションで設定されます。

以下は、Satellite ドメインリストのコレクション要求の形式例です。

要求例:

```
$ curl --request GET --insecure --user sat_username:sat_password \  
https://satellite.example.com/api/domains | python3 -m json.tool
```

応答例:

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

応答メタデータフィールド

API 応答は、以下のメタデータフィールドを使用します。

- **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オブジェクト数
- **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** 構文をもとにした検索文字列
- **sort**
 - **by**: API がコレクションをソートするフィールド別に指定
 - **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

第3章 API 呼び出しの認証

Satellite API との対話には、Satellite Server の CA 証明書で SSL 認証するか、有効な Satellite ユーザー認証情報で認証する必要があります。本章では、使用可能な認証方法について説明します。

3.1. SSL 認証の概要

Red Hat Satellite は HTTPS を使用して、Red Hat Satellite Server との通信時に、一定の暗号化およびアイデンティティの検証を行います。Satellite 6.14 では、SSL 以外の通信はサポートしません。

Red Hat Satellite Server はそれぞれ、自己署名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証明書および Satellite Server のアイデンティティを信頼する証明局のルールを果たします。

3.1.1. SSL 認証の設定

以下の手順を使用して、Satellite Server に対する API 要求の SSL 認証を設定します。

手順

1. 以下のオプションのいずれかを使用して、通信に使用する Satellite Server から証明書を取得します。

- リモートサーバーからコマンドを実行する場合は、SSH を使用して証明書を取得します。

```
$ scp root@satellite.example.com:/var/www/html/pub/katello-server-ca.crt /etc/pki/ca-trust/source/anchors/satellite.example.com-katello-server-ca.crt
```

- コマンドを Satellite Server で直接実行する場合は、証明書を **/etc/pki/ca-trust/source/anchors** ディレクトリーにコピーします。

```
$ cp /var/www/html/pub/katello-server-ca.crt /etc/pki/ca-trust/source/anchors/satellite.example.com-katello-server-ca.crt
```

2. 信頼できる CA のリストに証明書を追加します。

```
update-ca-trust extract
```

検証

- **--cacert** オプションを指定せずに API 要求を入力して、証明書が NSS データベースに存在することを確認します。

```
$ curl --request GET \  
--user sat_username:sat_password \  
https://satellite.example.com/api/v2/hosts
```

3.2. HTTP 認証の概要

Satellite API への要求には必ず、有効な Satellite ユーザー名とパスワードが必要です。API は HTTP の Basic 認証を使用して、これらの認証情報を暗号化して、**Authorization** ヘッダーに追加します。Basic 認証に関する詳細情報は、[RFC 2617 HTTP Authentication: Basic and Digest Access Authentication](#) を

参照してください。要求に適切な **Authorization** ヘッダーが含まれていない場合には、API は **401 Authorization Required** エラーを返します。



重要

Basic 認証では、パスワードをプレーンテキストで送信するなど、機密情報が含まれている可能性のある情報を扱います。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. PERSONAL ACCESS TOKEN 認証の概要

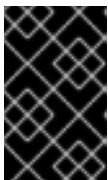
Red Hat Satellite は、パスワードを使用する代わりに API リクエストを認証するために使用できる Personal Access Tokens をサポートしています。Personal Access Token の有効期限を設定できます。有効期限が切れる前に、有効期限が切れると判断した場合は、トークンを取り消すことができます。

3.3.1. Personal Access Token の作成

この手順を使用して、Personal Access Token を作成します。

手順

1. Satellite Web UI で、**Administer** > **Users** に移動します。
2. Personal Access Token を作成するユーザーを選択します。
3. **Personal Access Tokens** タブで、**Add Personal Access Token** をクリックします。
4. Personal Access Token の **Name** を入力します。
5. オプション: **Expires** 日付を選択して、有効期限を設定します。有効期限を設定しない場合、Personal Access Token は、取り消されないかぎり、期限切れになりません。
6. **Submit** をクリックします。**Personal Access Tokens** タブで、Personal Access Token を使用できるようになりました。



重要

ページを離れるか、新しい Personal Access Token を作成すると、再びアクセスできなくなるため、Personal Access Token を必ず保存してください。**Copy to clipboard** をクリックして、Personal Access Token をコピーできます。

検証

1. Satellite Server に API リクエストを送信し、Personal Access Token で認証します。

```
# curl https://satellite.example.com/api/status --user
My_Username:My_Personal_Access-Token
```

2. ステータス **200** のレスポンスを受け取る必要があります。次に例を示します。

■

```
 {"satellite_version":"6.14.0","result":"ok","status":200,"version":"3.5.1.10","api_version":2}
```

Personal Access Tokens タブに戻ると、Personal Access Token の横に更新された **Last Used** 時刻が表示されます。

3.3.2. Personal Access Token の取り消し

この手順を使用して、有効期限が切れる前に、Personal Access Token を取り消します。

手順

1. Satellite Web UI で、**Administer** > **Users** に移動します。
2. Personal Access Token を取り消すユーザーを選択します。
3. **Personal Access Tokens** タブで、取り消す Personal Access Token を見つけます。
4. 取り消す Personal Access Token の横にある **Actions** 列の **取り消し** をクリックします。

検証

1. Satellite Server に API リクエストを送信し、失効した Personal Access Token で認証を試みません。

```
# curl https://satellite.example.com/api/status --user  
My_Username:My_Personal_Access_Token
```

2. 次のエラーメッセージが表示されます。

```
{  
  "error": {"message": "Unable to authenticate user My_Username"}  
}
```

3.4. OAUTH 認証の概要

Basic 認証の代わりに、制限のある OAuth 1.0 認証をサポートしています。これは、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。

OAuth の設定を確認するには、Satellite Web UI で **管理** > **設定** > **認証** に移動します。OAuth コンシューマーキー は全 OAuth クライアントが使用するトークンです。

Satellite は、OAuth 設定を `/etc/foreman/settings.yaml` ファイルに保存します。Satellite はアップグレード時にこのファイルへの手動の変更を上書きするので、これらの設定を行うには、**satellite-installer** スクリプトを使用します。

3.4.1. OAuth の認証

OAuth 設定を変更するには、必要なオプションを指定して **satellite-installer** を入力します。以下のコマンドを入力して、OAuth 関連のインストーラーオプションすべてを表示します。

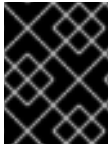
```
# satellite-installer --full-help | grep oauth
```

OAuth のマッピングの有効化

デフォルトでは、Satellite は、ビルトインの匿名 API 管理者アカウントで、全 OAuth API 要求を認可します。そのため、API 応答には全 Satellite データが含まれます。ただし、要求を行う Foreman ユーザーを指定して、そのユーザーに対するデータへのアクセスを制限することも可能です。

OAuth ユーザーマッピングを有効にするには、以下のコマンドを入力します。

```
# satellite-installer --foreman-oauth-map-users true
```



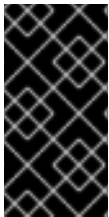
重要

Satellite は OAuth 要求のヘッダーを署名しません。有効なコンシューマーキーのあるユーザーなら誰でも、Foreman ユーザーになりすますことができます。

3.4.2. OAuth 要求の形式

全 OAuth API 要求には、以下の形式で、既存の Foreman ユーザーのログインを指定した **FOREMAN-USER** ヘッダーと、**Authorization** ヘッダーが必要です。

```
--header 'FOREMAN-USER: sat_username' \  
--header 'Authorization: OAuth  
oauth_version="1.0",oauth_consumer_key="secretkey",oauth_signature_method="hmac-  
sha1",oauth_timestamp=timestamp,oauth_signature=signature'
```



重要

OAuth クライアントライブラリーを使用して、全 OAuth パラメーターを構築します。Python モジュール `request_oauthlib` を使用する例については、Red Hat ナレッジベースの [How to execute an API call using the OAuth authentication method via python script in Red Hat Satellite 6?](#) を参照してください。

例

以下の例では、認証に OAuth を使用したアーキテクチャーが表示されます。この要求では、**FOREMAN-USER** ヘッダーに `sat_username` ユーザー名を使用します。`--foreman-oauth-map-users` を `true` に設定すると、要求には、そのユーザーに対して表示権限が割り当てられたアーキテクチャーのみが含まれます。署名は、全パラメーター、HTTP メソッドおよび URI 変更を反映します。

要求例:

```
$ curl 'https://satellite.example.com/api/architectures' \  
--header 'Content-Type: application/json' \  
--header 'Accept:application/json' \  
--header 'FOREMAN-USER: sat_username' \  
--header 'Authorization: OAuth  
oauth_version="1.0",oauth_consumer_key="secretkey",oauth_signature_method="hmac-  
sha1",oauth_timestamp=1321473112,oauth_signature=ll8hR8/ogj/XVuOqMPB9qNjSy6E='
```

関連情報

- [OAuth Core 1.0 Revision A のドキュメント](#)

第4章 異なる言語での API 要求

本章では、curl、Ruby および Python を使用して Red Hat Satellite に API 要求を送信する方法を、例と併せて説明します。

4.1. CURL を使用した API 要求

本セクションでは、Satellite API で **curl** を使用して、さまざまなタスクを実行する方法を説明します。

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 認証の概要](#)」の記載どおりに Satellite Server の証明書を追加していない場合には、**--insecure** 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。

ユーザー認証の場合は、**--user** オプションを指定して、**--user username:password** 形式で、Satellite ユーザー認証情報を指定します。または、パスワードを含めない場合には、コマンドでパスワードを入力するようにプロンプトが表示されます。セキュリティリスクを軽減するには、コマンドの一部としてパスワードを入力すると、シェルの履歴の一部として残ってしまうので、入力しないようにしてください。本セクションの例では、簡素化の目的でのみ、パスワードを含めています。

--silent オプションを使用する場合は、**curl** で進捗メーターやエラーメッセージは表示されない点に注意してください。

本章の例では、Python **json.tool** モジュールを使用して出力をフォーマットしています。

4.1.1. API 要求への JSON データの指定

データを API 要求で Satellite Server に渡すことができます。このデータは、JSON 形式でなければなりません。**--data** オプションで、JSON データを指定する場合には、**--header** オプションを使用して、以下の HTTP ヘッダーを設定する必要があります。

```
--header "Accept:application/json" \  
--header "Content-Type:application/json"
```

以下のオプションのいずれかを使用して、**--data** オプションでデータを追加します。

1. 引用符の付いた JSON 形式のデータを中括弧 {} 囲みます。JSON タイプのパラメーターの値を渡す場合には、バックスラッシュ \ を使用して、引用符 " をエスケープする必要があります。たとえば、中括弧内では **"Example JSON Variable"** は **"\Example JSON Variable\"** のような形式にする必要があります。

```
--data {"id":44, "smart_class_parameter":{"override":"true", "parameter_type":"json",  
"default_value":{"GRUB_CMDLINE_LINUX":{"audit":"1","crashkernel":"true"}}}}
```

2. 引用符なしの JSON 形式のデータは、@ マークとファイル名を指定します。以下に例を示します。

```
--data @file.json
```

JSON 形式データの外部ファイルを使用すると、以下のような利点があります。

- 任意のテキストエディターを使用できる
- 構文チェッカーを使用してミスを特定し、回避できる

- ツールを使用して JSON データの妥当性を確認して、再フォーマットができる

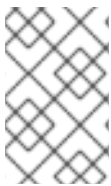
JSON ファイルの検証

`json_verify` ツールを使用して、JSON ファイルの妥当性を確認します。

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

4.1.2. リソースリストの取得

本セクションは、Satellite 6 API で `curl` を使用して、Satellite デプロイメントから情報を要求する方法を説明します。これらの例には、要求と応答の両方が含まれます。デプロイメントごとに、異なる結果が得られることが想定されます。



注記

以下の要求例では、`python3` を使用して Satellite Server からの応答をフォーマットしています。RHEL 7 およびそれ以前の一部のシステムでは、`python3` の代わりに `python` を使用する必要があります。

ユーザーのリスト表示

以下の例は、Satellite リソース (今回の例では、Satellite ユーザー) のリストを返す基本的な要求です。このような要求では、メタデータでラップされたデータリストを返しますが、他の要求タイプでは実際のオブジェクトが返されます。

要求例:

```
$ curl --request GET --insecure --user sat_username:sat_password \
https://satellite.example.com/api/users | python3 -m json.tool
```

応答例:

```
{
  "page": 1,
  "per_page": 20,
  "results": [
    {
      "admin": false,
      "auth_source_id": 1,
      "auth_source_name": "Internal",
      "created_at": "2018-09-21 08:59:22 UTC",
      "default_location": null,
      "default_organization": null,
      "description": "",
      "effective_admin": false,
      "firstname": "",
      "id": 5,
      "last_login_on": "2018-09-21 09:03:25 UTC",
      "lastname": "",
      "locale": null,
      "locations": [],
      "login": "test",
      "mail": "example@domain.com",
```

```
    "organizations": [
      {
        "id": 1,
        "name": "Default Organization"
      }
    ],
    "ssh_keys": [],
    "timezone": null,
    "updated_at": "2018-09-21 09:04:45 UTC"
  },
  {
    "admin": true,
    "auth_source_id": 1,
    "auth_source_name": "Internal",
    "created_at": "2018-09-20 07:09:41 UTC",
    "default_location": null,
    "default_organization": {
      "description": null,
      "id": 1,
      "name": "Default Organization",
      "title": "Default Organization"
    },
    "description": "",
    "effective_admin": true,
    "firstname": "Admin",
    "id": 4,
    "last_login_on": "2018-12-07 07:31:09 UTC",
    "lastname": "User",
    "locale": null,
    "locations": [
      {
        "id": 2,
        "name": "Default Location"
      }
    ],
    "login": "admin",
    "mail": "root@example.com",
    "organizations": [
      {
        "id": 1,
        "name": "Default Organization"
      }
    ],
    "ssh_keys": [],
    "timezone": null,
    "updated_at": "2018-11-14 08:19:46 UTC"
  }
],
"search": null,
"sort": {
  "by": null,
  "order": null
},
"subtotal": 2,
"total": 2
}
```

4.1.3. リソースの作成および変更

本章では、Satellite 6 API で **curl** を使用して Satellite Server のリソースを操作する方法を説明します。これらの API 呼び出しでは、API 呼び出しの **json** 形式でデータを渡す必要があります。詳細は、「[API 要求への JSON データの指定](#)」を参照してください。

ユーザーの作成

以下の例では、**--data** オプションを使用してユーザーを作成し、必要な情報を指定します。

要求例:

```
$ curl --header "Accept:application/json" \
--header "Content-Type:application/json" --request POST \
--user sat_username:sat_password --insecure \
--data "{\"firstname\":\"Test
Name\",\"mail\":\"test@example.com\", \"login\":\"test_user\", \"password\":\"password123\", \"auth_s
ource_id\":1}" \
https://satellite.example.com/api/users | python3 -m json.tool
```

ユーザーの変更

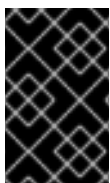
以下の例では、[ユーザーの作成](#) で作成した **test_user** のログインと名前を変更します。

要求例:

```
$ curl --header "Accept:application/json" \
--header "Content-Type:application/json" --request PUT \
--user sat_username:sat_password --insecure \
--data "{\"firstname\":\"New Test
Name\",\"mail\":\"test@example.com\", \"login\":\"new_test_user\", \"password\":\"password123\", \"a
uth_source_id\":1}" \
https://satellite.example.com/api/users/8 | python3 -m json.tool
```

4.2. RUBY を使用した API 要求

本セクションでは、Satellite API で Ruby を使用して、さまざまなタスクを実行する方法を説明します。



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、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には、この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://satellite.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)
end
```



```

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

```

4.2.2. Ruby での Apipie バインディングの使用

Apipie バインディングは、apipie で記述された API 呼び出しの Ruby バインディングです。このバインディングは、Satellite から API 定義を取得してキャッシュし、オンデマンドで API 呼び出しを生成します。以下の例では、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には、この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://satellite.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: \#{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

# Create life cycle environments
environments.each do |environment|
  puts "Creating environment: \#{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 要求

本セクションでは、Satellite API で Python を使用して、さまざまなタスクを実行する方法を説明します。



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なしてください。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。

本セクションのスクリプト例では、REST API との対話に SSL 検証を使用しません。

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、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には、この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。

Python 2 の例

```
#!/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://satellite.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 API からの情報要求

以下は、さまざまな API 要求に Python を使用するスクリプト例です。

Python 2 の例

```
#!/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://satellite.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 = 'satellite.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()

```

Python 3 の例

```

#!/usr/bin/env python3

import json
import sys

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

SAT = "satellite.example.com"
# URL for the API to your deployed Satellite 6 server
SAT_API = f"https://{SAT}/api/"
KATELLO_API = f"https://{SAT}/katello/api/v2/"

POST_HEADERS = {'content-type': 'application/json'}
# Default credentials to login to Satellite 6
USERNAME = "admin"
PASSWORD = "password"
# Ignore SSL for now
SSL_VERIFY = False
#SSL_VERIFY = "./path/to/CA-certificate.crt" # Put the path to your CA certificate here to allow
SSL_VERIFY

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:
        print(f"{'ID':10}{'Name':40}{'IP':30}{'Operating System':30}")
        for host in hosts:
            print(f"{str(host['id']):10}{host['name']:40}{str(host['ip']):30}
{str(host['operatingsystem_name']):30}")

def display_info_for_subs(url):
    subs = get_results(url)
    if subs:
        print(f"{'ID':10}{'Name':90}{'Start Date':30}")
        for sub in subs:
            print(f"{str(sub['id']):10}{sub['name']:90}{str(sub['start_date']):30}")

def main():
    host = SAT
    print(f"Displaying all info for host {host} ...")
    display_all_results(SAT_API + 'hosts/' + host)

    print(f"Displaying all facts for host {host} ...")
    display_all_results(SAT_API + f'hosts/{host}/facts')

    host_pattern = 'example'
    print(f"Displaying basic info for hosts matching pattern '{host_pattern}'...")
    display_info_for_hosts(SAT_API + 'hosts?per_page=1&search=name~' + host_pattern)

    print(f"Displaying basic info for subscriptions")
    display_info_for_subs(KATELLO_API + 'subscriptions')

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

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

第5章 RED HAT SATELLITE API の使用

本章では、Red Hat Satellite API を使用して各種タスクを実行する方法をさまざまな例を挙げて説明します。Satellite Server で API を使用するには、ポート 443 の HTTPS を、Capsule Server で API を使用するには、ポート 8443 の HTTPS を使用します。

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

ホストが Satellite Server または Capsule Server にサブスクライブしている場合は、`/etc/rhsm/rhsm.conf` の **[server]** セクションのポートエントリをもとに、API へのアクセスに必要な、正しいポートを判断できます。これらの値を使用してスクリプトを完全に自動化し、使用するポートを検証する必要性をなくします。

本章では、**curl** を使用して API 要求を送信します。詳細は、「[curl を使用した API 要求](#)」を参照してください。

本章の例では、Python **json.tool** モジュールを使用して出力をフォーマットしています。

5.1. ホストの使用方法



注記

以下の要求例では、**python3** を使用して Satellite Server からの応答をフォーマットしています。RHEL 7 およびそれ以前の一部のシステムでは、**python3** の代わりに **python** を使用する必要があります。

ホストのリスト表示

以下の例では、Satellite ホストのリストを返します。

要求例:

```
$ curl --request GET --insecure --user sat_username:sat_password \
https://satellite.example.com/api/v2/hosts | python3 -m json.tool
```

応答例:

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



```
"results" => [  
  ...  
]
```

ホストの情報要求

この要求は、**satellite.example.com** ホストの情報を返します。

要求例:

```
$ curl --request GET --insecure --user sat_username:sat_password \  
https://satellite.example.com/api/v2/hosts/satellite.example.com \  
| python3 -m json.tool
```

応答例:

```
{  
  "all_puppetclasses": [],  
  "architecture_id": 1,  
  "architecture_name": "x86_64",  
  "build": false,  
  "capabilities": [  
    "build"  
  ],  
  "certname": "satellite.example.com",  
  "comment": null,  
  "compute_profile_id": null,  
  ...  
}
```

ホストのファクト表示

この要求は、**satellite.example.com** ホストの全ファクトを返します。

要求例:

```
$ curl --request GET --insecure --user sat_username:sat_password \  
https://satellite.example.com/api/v2/hosts/satellite.example.com/facts \  
| python3 -m json.tool
```

応答例:

```
{  
  ...  
  "results": {  
    "satellite.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",  
      ...  
    }  
  }  
}
```

パターンが一致するホストの検索

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

要求例:

```
$ curl --request GET --insecure --user sat_username:sat_password \  
https://satellite.example.com/api/v2/hosts?search=example \  
| python3 -m json.tool
```

応答例:

```
{  
  ...  
  "results": [  
    {  
      "name": "satellite.example.com",  
      ...  
    }  
  ],  
  "search": "example",  
  ...  
}
```

環境でのホストの検索

以下のクエリは、**production** 環境内の全ホストを返します。

要求例:

```
$ curl --request GET --insecure --user sat_username:sat_password \  
https://satellite.example.com/api/v2/hosts?search=environment=production \  
| python3 -m json.tool
```

応答例:

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

特定のファクト値を含むホストの検索

以下のクエリでは、**RHEV Hypervisor** というモデル名のホストがすべて返されます。

要求例:

```
$ curl --request GET --insecure --user sat_username:sat_password \
https://satellite.example.com/api/v2/hosts?search=model=\"RHEV+Hypervisor\" \
| python3 -m json.tool
```

応答例:

```
{
  ...
  "results": [
    {
      "model_id": 1,
      "model_name": "RHEV Hypervisor",
      "name": "satellite.example.com",
      ...
    }
  ],
  "search": "model=\"RHEV Hypervisor\"",
  ...
}
```

ホストの削除

この要求は、名前が `host1.example.com` のホストを削除します。

要求例:

```
$ curl --request DELETE --insecure --user sat_username:sat_password \
https://satellite.example.com/api/v2/hosts/host1.example.com \
| python3 -m json.tool
```

完全な起動ディスクイメージのダウンロード

以下の要求では、ホストの完全な起動ディスクイメージを ID を使用してダウンロードします。

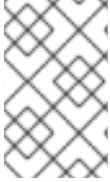
要求例:

```
$ curl --request GET --insecure --user sat_username:sat_password \
https://satellite.example.com/bootdisk/api/hosts/host_ID?full=true \
--output image.iso
```

5.2. ライフサイクル環境の使用

Satellite は、アプリケーションのライフサイクルを、ライフサイクル環境に分割します。ライフサイクル環境は、アプリケーションのライフサイクルの各ステージを表します。ライフサイクル環境は、環境パスからリンクされます。リンクされたライフサイクル環境を API で作成するには、`prior_id` パラメーターを使用します。

https://satellite.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 リファレンスを確認できます。API ルートには `/katello/api/environments` および `/katello/api/organizations/:organization_id/environments` が含まれます。



注記

以下の要求例では、**python3** を使用して Satellite Server からの応答をフォーマットしています。RHEL 7 およびそれ以前の一部のシステムでは、**python3** の代わりに **python** を使用する必要があります。

ライフサイクル環境の表示

以下の API 呼び出しを使用して、Satellite にある ID が 1 のデフォルトの組織に対する現在の全ライフサイクル環境を表示します。

要求例:

```
$ curl --header "Accept:application/json" \
--header "Content-Type:application/json" \
--request GET --user sat_username:sat_password --insecure \
https://satellite.example.com/katello/api/organizations/1/environments \
| python3 -m json.tool`
```

応答例:

```
output omitted
"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,
output truncated
```

ライフサイクル環境のリンク作成

以下の例を使用して、ライフサイクル環境のパスを作成します。

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。

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。

要求例:

```
$ curl --request GET --user sat_username:sat_password --insecure \
https://satellite.example.com/katello/api/environments/1 \
| python3 -m json.tool
```

応答例:

output omitted

```
"id": 1,
"label": "Library",
```

output omitted

```
"prior": null,
"successor": null,
```

output truncated

- 以下のコンテンツを含めて、**life-cycle.json** などの JSON ファイルを作成します。

```
{"organization_id":1,"label":"api-dev","name":"API Development","prior":1}
```

- prior** オプションを **1** に設定して、ライフサイクル環境を作成します。

要求例:

```
$ curl --header "Accept:application/json" \
--header "Content-Type:application/json" \
--request POST --user sat_username:sat_password --insecure \
--data @life-cycle.json \
https://satellite.example.com/katello/api/environments \
| python3 -m json.tool
```

応答例:

output omitted

```
"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"
```

```
},
```

output truncated

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。ID が **2** のライフサイクル環境を使用して、この環境の後継を作成します。

4. 以前作成した **life-cycle.json** ファイルを編集して、**label**、**name**、**prior** の値を更新します。

```
{ "organization_id":1,"label":"api-qa","name":"API QA","prior":2}
```

5. **prior** オプションを **2** に設定して、ライフサイクル環境を作成します。
要求例:

```
$ curl --header "Accept:application/json" \
--header "Content-Type:application/json" \
--request POST --user sat_username:sat_password --insecure \
--data @life-cycle.json \
https://satellite.example.com/katello/api/environments \
| python3 -m json.tool
```

応答例:

```
output omitted
"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,
output truncated
```

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 **2** であると分かります。

ライフサイクル環境の更新

PUT コマンドを使用して、ライフサイクル環境を更新できます。

以下の要求例では、ID が **3** のライフサイクル環境の説明を更新します。

要求例:

```
$ curl --header "Accept:application/json" \
--header "Content-Type:application/json" \
--request POST --user sat_username:sat_password --insecure \
--data '{"description":"Quality Acceptance Testing"}' \
https://satellite.example.com/katello/api/environments/3 \
| python3 -m json.tool
```

応答例:

```
output omitted
"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"
},
output truncated
```

ライフサイクル環境の削除

後継がない前提でライフサイクル環境を削除できます。このような理由から、以下の形式のコマンドを使用して、逆順にライフサイクル環境を削除します。

要求例:

```
$ curl --request DELETE --user sat_username:sat_password --insecure \
https://satellite.example.com/katello/api/environments/:id
```

5.3. SATELLITE SERVER 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 を使用して Satellite Server に大容量のファイルをアップグレードしてインポートする方法を説明します。このプロセスでは、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、約 2 MB です。大容量のコンテンツのアップロードに関する情報は、[2 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

1. パッケージ名を変数 **名** に割り当てます。

要求例:

```
$ export name=jq-1.6-2.el7.x86_64.rpm
```

2. ファイルのチェックサムを変数 **チェックサム** に割り当てます。

要求例:

```
$ export checksum=$(sha256sum $name|cut -c 1-65)
```

3. ファイルサイズを変数 **サイズ** に割り当てます。

要求例:

```
$ export size=$(du -bs $name|cut -f 1)
```

4. 以下のコマンドは、新しいアップロード要求を作成し、**サイズ** と **チェックサム** を使用して要求のアップロード ID を返します。

要求例:

```
$ curl -H 'Content-Type: application/json' -X POST -k \  
-u sat_username:sat_password -d '{"size": "$size", \  
"checksum": "$checksum"}' \  
https://$(hostname -f)/katello/api/v2/repositories/76/content_uploads
```

今回の場合は、76 はリポジトリ ID の例です。

要求例:

```
{"upload_id": "37eb5900-597e-4ac3-9bc5-2250c302fdc4"}
```

5. 変数 **upload_id** にアップロード ID を割り当てます。

要求例:

```
$ export upload_id=37eb5900-597e-4ac3-9bc5-2250c302fdc4
```

6. アップロードするパッケージのパスを変数 **パス** に割り当てます。

```
$ export path=/root/jq/jq-1.6-2.el7.x86_64.rpm
```

7. 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。この API では、Satellite 6 に対する要求にほぼ、application/json の MIME タイプが使用します。upload_id と MIME タイプ、他のパラメーターを組み合わせるとコンテンツをアップロードします。

要求例:

```
$ curl -u sat_username:sat_password -H Accept:application/json -H \  
Content-Type:multipart/form-data -X PUT --data-urlencode size=$size --data-urlencode
```



```
offset=0 \
--data-urlencode content@${path} \
https://$(hostname -f)/katello/api/v2/repositories/76/content_uploads/$upload_id
```

- Satellite Server 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 Server ではこの新しいコンテンツは認識されません。

要求例:

```
$ curl -H "Content-Type:application/json" -X PUT -u \
sat_username:sat_password -k -d \
"{\"uploads\":{\"id\": \"$upload_id\", \"name\": \"$name\", \
\"checksum\": \"$checksum\" }}" \
https://$(hostname -f)/katello/api/v2/repositories/76/import_uploads
```

- 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、アップロード中にデータが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。

要求例:

```
$ curl -H 'Content-Type: application/json' -X DELETE -k \
-u sat_username:sat_password -d "{}" \
https://$(hostname -f)/katello/api/v2/repositories/76/content_uploads/$upload_id
```

2 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、リポジトリ ID、ファイル名を使用している点に注意してください。

- パッケージ名を変数 **名** に割り当てます。

```
$ export name=bpftool-3.10.0-1160.2.1.el7.centos.plus.x86_64.rpm
```

- ファイルのチェックサムを変数 **チェックサム** に割り当てます。

```
$ export checksum=$(sha256sum $name|cut -c 1-65)
```

- ファイルサイズを変数 **サイズ** に割り当てます。

```
$ export size=$(du -bs $name|cut -f 1)
```

- 以下のコマンドは、新しいアップロード要求を作成し、**サイズ** と **チェックサム** を使用して要求のアップロード ID を返します。

要求例:

```
$ curl -H 'Content-Type: application/json' -X POST -k \
-u sat_username:sat_password -d "{\"size\": \"$size\", \
\"checksum\": \"$checksum\"}" \
https://$(hostname -f)/katello/api/v2/repositories/76/content_uploads
```

今回の場合は、76 はリポジトリ ID の例です。

出力例

```
 {"upload_id":"37eb5900-597e-4ac3-9bc5-2250c302fdc4"}
```

- 変数 **upload_id** にアップロード ID を割り当てます。

```
 $ export upload_id=37eb5900-597e-4ac3-9bc5-2250c302fdc4
```

- ファイルを 2MB のチャンクに分割します。

```
 $ split --bytes 2MB --numeric-suffixes \
  --suffix-length=1 bpftool-3.10.0-1160.2.1.el7.centos.plus.x86_64.rpm bpftool
```

出力例

```
 $ ls bpftool[0-9] -l
-rw-r--r--. 1 root root 2000000 Mar 31 14:15 bpftool0
-rw-r--r--. 1 root root 2000000 Mar 31 14:15 bpftool1
-rw-r--r--. 1 root root 2000000 Mar 31 14:15 bpftool2
-rw-r--r--. 1 root root 2000000 Mar 31 14:15 bpftool3
-rw-r--r--. 1 root root 868648 Mar 31 14:15 bpftool4
```

- 分割ファイルの接頭辞を変数パスに割り当てます。

```
 $ export path=/root/tmp/bpftool
```

- ファイルのチャンクをアップロードします。オフセットは、最初のチャンクの 0 から始まり、ファイルごとに 2000000 ずつ増加します。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。また、インデックスはパス変数の後に使用されます (例: `${path}0`、`${path}1`)。

要求例:

```
 $ curl -u sat_username:sat_password -H Accept:application/json -H \
  Content-Type:multipart/form-data \
  -X PUT --data-urlencode size=$size --data-urlencode offset=0 \
  --data-urlencode content@${path}0 https://$(hostname -
  f)/katello/api/v2/repositories/76/content_uploads/$upload_id
```

```
 $ curl -u sat_username:sat_password -H Accept:application/json -H \
  Content-Type:multipart/form-data \
  -X PUT --data-urlencode size=$size --data-urlencode offset=2000000 \
  --data-urlencode content@${path}1 https://$(hostname -
  f)/katello/api/v2/repositories/76/content_uploads/$upload_id
```

```
 $ curl -u sat_username:sat_password -H Accept:application/json -H \
  Content-Type:multipart/form-data \
  -X PUT --data-urlencode size=$size --data-urlencode offset=4000000 \
  --data-urlencode content@${path}2 https://$(hostname -
  f)/katello/api/v2/repositories/76/content_uploads/$upload_id
```

```
 $ curl -u sat_username:sat_password -H Accept:application/json -H \
  Content-Type:multipart/form-data \
  -X PUT --data-urlencode size=$size --data-urlencode offset=6000000
```

```
--data-urlencode content@${path}3 https://$(hostname -
f)/katello/api/v2/repositories/76/content_uploads/$upload_id

$ curl -u sat_username:sat_password -H Accept:application/json -H \
Content-Type:multipart/form-data \
-X PUT --data-urlencode size=$size --data-urlencode offset=8000000 \
--data-urlencode content@${path}4 https://$(hostname -
f)/katello/api/v2/repositories/76/content_uploads/$upload_id
```

9. 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 "Content-Type:application/json" -X PUT -u \
sat_username:sat_password -k -d \
{"uploads":[{"id": "$upload_id", \
"name": "$name", "checksum": "$checksum"}]} \
https://$(hostname -f)/katello/api/v2/repositories/76/import_uploads
```

10. アップロード要求を削除します。

```
$ curl -H 'Content-Type: application/json' -X DELETE -k \
-u sat_username:sat_password -d "{}" \
https://$(hostname -f)/katello/api/v2/repositories/76/content_uploads/$upload_id
```

重複コンテンツのアップロード

以下を使用して重複したコンテンツをアップロードしようとする、以下の点に留意してください。

要求例:

```
$ curl -H 'Content-Type: application/json' -X POST -k \
-u sat_username:sat_password -d '{"size": "$size", "checksum": "$checksum"}' \
https://$(hostname -f)/katello/api/v2/repositories/76/content_uploads
```

この呼び出しは、以下のようなアップロード ID ではなくコンテンツユニット ID を返します。

```
{"content_unit_href": "/pulp/api/v3/content/file/files/c1bcdfb8-d840-4604-845e-86e82454c747/"}
```

この出力をコピーし、インポートアップロードを直接呼び出して、コンテンツをリポジトリに追加できます。

要求例:

```
$ curl -H "Content-Type:application/json" -X PUT -u \
sat_username:sat_password -k -d \
{"uploads":[{"content_unit_id": "/pulp/api/v3/content/file/files/c1bcdfb8-d840-4604-845e-86e82454c747/", \
"name": "$name", "checksum": "$checksum"}]} https://$(hostname -
f)/katello/api/v2/repositories/76/import_uploads
```

呼び出しが **upload_id** の使用から **content_unit_id** の使用に変わることにご注意してください。

5.4. エラータのホストまたはホストコレクションへの適用

API 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 --header "Accept:application/json" \
--header "Content-Type:application/json" --request PUT \
--user sat_username:sat_password --insecure \
--data json-formatted-data https://satellite7.example.com
```

同梱の API ドキュメントを参照して、エラータ適用に使用する URL を検索します。Satellite Web UI で、検索クエリーの形式を確認できます。ホスト > ホストコレクション 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の内容を確認します。たとえば、**my-collection** と呼ばれるホストコレクションの場合は、検索ボックスに **host_collection="my-collection"** が含まれます。

ホストへのエラータの適用

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

要求例:

```
$ curl --header "Accept:application/json" \
--header "Content-Type:application/json" --request PUT \
--user sat_username:sat_password --insecure \
--data "{\"organization_id\":1,\"included\":{\"search\":\"my-host\"},\"content_type\":\"errata\",\"content\":[\"RHBA-2016:1981\"]}" \
https://satellite.example.com/api/v2/hosts/bulk/install_content
```

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。

要求例:

```
$ curl --header "Accept:application/json" \
--header "Content-Type:application/json" --request PUT \
--user sat_username:sat_password --insecure \
--data "{\"organization_id\":1,\"included\":{\"search\":\"host_collection=\\\"my-collection\\\"\"},\"content_type\":\"errata\",\"content\":[\"RHBA-2016:1981\"]}" \
https://satellite.example.com/api/v2/hosts/bulk/install_content
```

5.5. 詳細検索の使用

Web UI で検索クエリーを構築するのに使用可能な検索パラメーターを検索できます。詳細は、[Red Hat Satellite の管理](#) の [検索クエリーの構築](#) を参照してください。

たとえば、ホストを検索するには、以下の手順を実行します。

1. Satellite Web UI で、**ホスト > すべてのホスト** に移動して、**検索** フィールドをクリックして、検索パラメーターのリストを表示します。
2. 使用する検索パラメーターの場所を特定します。以下の例では、**os_title** と **model** を特定します。

3. 以下のように、API クエリーで検索パラメーターを組み合わせます。

要求例:

```
$ curl --insecure --user sat_username:sat_password \
https://satellite.example.com/api/v2/hosts?
search=os_title=\"RedHat+7.7\",model=\"PowerEdge+R330\" \
| python3 -m json.tool
```



注記

この要求例では、**python3** を使用して Satellite Server からの応答をフォーマットしています。RHEL 7 およびそれ以前の一部のシステムでは、**python3** の代わりに **python** を使用する必要があります。

応答例:

```
{
  ...
  "results": [
    {
      "model_id": 1,
      "model_name": "PowerEdge R330",
      "name": "satellite.example.com",
      "operatingsystem_id": 1,
      "operatingsystem_name": "RedHat 7.7",
      ...
    }
  ],
  "search": "os_title=\"RedHat 7.7\",model=\"PowerEdge R330\"",
  "subtotal": 1,
  "total": 11
}
```

5.6. ページネーション制御のある検索の使用

per_page および **page** ページネーションパラメーターを使用して、API 検索クエリーが返した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、ページごとに表示する結果数を指定し、**page** パラメーターは **per_page** パラメーターの計算に合わせて、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。

コンテンツビューの表示

以下の例では、複数ページにわたるコンテンツビューのリストを返します。このリストでは、ページごとにキー 10 個が含まれており、3 ページ目を返します。

要求例:

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

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リストを複数ページで返します。このリストには、ページごとにキーが 30 個含まれており、2 ページ目が返されます。

要求例:

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

複数ページを返す設定

for loop 構造を使用して、複数ページにわたる結果を取得します。

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 --request GET --user sat_username:sat_password \
https://satellite.example.com/katello/api/content_views?per_page=5&page=$i; \
done
```

5.7. 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上書きすることができます。変更可能な属性の完全リスト

は、https://satellite.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。

1. 変更するスマートクラスパラメーターの ID を検索します。

- スマートクラスパラメーターすべてを表示します。

要求例:

```
$ curl --request GET --insecure --user sat_username:sat_password \
https://satellite.example.com/api/smart_class_parameters
```

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

要求例:

```
$ curl --request GET --insecure --user sat_username:sat_password \
https://satellite.example.com/api/puppetclasses/5/smart_class_parameters
```

どちらの呼び出しも検索パラメーターに対応します。Satellite Web UI で検索可能なフィールドの全リストを表示できます。**設定 > スマート変数** に移動して、検索クエリーボックスをクリックし、フィールドのリストを表示します。

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、**--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。

要求例:

```
$ curl --request GET --insecure --user sat_username:sat_password \
--data 'search=puppetclass_name = access_insights_client and key = authmethod' \
https://satellite.example.com/api/smart_class_parameters
```

Satellite は、標準のスコープ指定の検索構文をサポートします。

2. パラメーターの ID が見つかり、現在の上書き値など、全詳細が表示されます。
要求例:

```
$ curl --request GET --insecure --user sat_username:sat_password \
https://satellite.example.com/api/smart_class_parameters/63
```

3. パラメーターの値のオーバーライドを有効化します。
要求例:

```
$ curl --header "Accept:application/json" \
--header "Content-Type:application/json" \
--request PUT --insecure --user sat_username:sat_password \
--data '{"smart_class_parameter":{"override":true}}' \
https://satellite.example.com/api/smart_class_parameters/63
```

パラメーターを手動で作成または削除できない点に注意してください。属性の変更のみが可能です。Satellite は、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。

4. カスタムの override matcher を追加します。
要求例:

```
$ curl --header "Accept:application/json" \
--header "Content-Type:application/json" \
--request PUT --insecure --user sat_username:sat_password \
--data '{"smart_class_parameter":{"override_value":
{"match":"hostgroup=Test","value":"2.4.6"}}}' \
https://satellite.example.com/api/smart_class_parameters/63
```

オーバーライドの値に関する情報

は、https://satellite.example.com/apidoc/v2/override_values.html を参照してください。

5. 上書きの値を削除します。
要求例:

```
$ curl --request DELETE --user sat_username:sat_password \
https://satellite.example.com/api/smart_class_parameters/63/override_values/3
```

5.8. 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。



注記

以下の要求例では、**python3** を使用して Satellite Server からの応答をフォーマットしています。RHEL 7 およびそれ以前の一部のシステムでは、**python3** の代わりに **python** を使用する必要があります。

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. 今回の例では、**motd** という名前で Puppet クラスを検索します。
要求例:

```
$ curl --header "Accept:application/json" \
--header "Content-Type:application/json" \
--request GET --user sat_user:sat_password --insecure \
https://satellite.example.com/api/smart_class_parameters?
search=puppetclass_name=motd \
| python3 -m json.tool
```

2. 以下の出力を検証します。スマートクラスのパラメーターにはそれぞれ、同じ Satellite インスタンスでグローバルとなる ID が割り当てられています。Satellite Server では、**motd** クラスの **content** パラメーターは **id=3** となっています。Puppet クラス名の前に表示される Puppet クラス ID と混同しないようにしてください。

応答例:

```
{
  "avoid_duplicates": false,
  "created_at": "2017-02-06 12:37:48 UTC", # Remove this line.
  "default_value": "", # Add a new value here.
  "description": "",
  "hidden_value": "",
  "hidden_value?": false,
  "id": 3,
  "merge_default": false,
  "merge_overrides": false,
  "override": false, # Set the override value to true.
  "override_value_order": "fqdn\nhostgroup\nnos\ndomain",
  "override_values": [], # Remove this line.
  "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", # Remove this line.
  "use_puppet_default": false,
  "validator_rule": null,
  "validator_type": ""
}
```

3. 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。
要求例:


```
$ curl --header "Accept:application/json" \
--header "Content-Type:application/json" --request GET \
--user sat_user:sat_password --insecure \
https://satellite.example.com/api/smart_class_parameters/3 \
| python3 -m json.tool > output_file.json
```

- 1つ前の手順で作成したファイルを、新しいファイル (例: **changed_file.json**) にコピーして編集します。

```
$ cp output_file.json changed_file.json
```

5. ファイルで、必要な値を変更します。以下の例では、**motd** モジュールのコンテンツパラメータを変更しますが、これには、**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", # Remove this line.
  "default_value": "", # Add a new value here.
  "description": "",
  "hidden_value": "",
  "hidden_value?": false,
  "id": 3,
  "merge_default": false,
  "merge_overrides": false,
  "override": false, # Set the override value to true.
  "override_value_order": "fqdn\nhostgroup\nos\ndomain",
  "override_values": [], # Remove this line.
  "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", # Remove this line.
  "use_puppet_default": false,
  "validator_rule": null,
  "validator_type": ""
}
```

6. 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\nhostgroup\nos\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": ""
}

```

7. Satellite Server への変更を適用します。

```

$ curl --header "Accept:application/json" \
--header "Content-Type:application/json" \
--request PUT --user sat_username:sat_password --insecure \
--data @changed_file.json \
https://satellite.example.com/api/smart_class_parameters/3

```

5.9. OPENSAP レポートの削除

Satellite Server で、1つまたは複数の OpenSCAP レポートを削除できます。ただし、レポートは、一度に1ページずつしか削除できません。すべての Openscap レポートを削除する場合には、以下のバッチスクリプトを使用してください。



注記

以下の要求例とスクリプト例では、**python3** を使用して Satellite Server からの応答をフォーマットしています。RHEL 7 およびそれ以前の一部のシステムでは、**python3** の代わりに **python** を使用する必要があります。

OpenSCAP レポートの削除

OpenSCAP レポートを削除するには、次の手順を実行します。

1. 全 OpenSCAP レポートを表示します。削除するレポートの ID をメモしてください。
要求例:

```

curl --insecure --user username:_password_ \
https://satellite.example.com/api/v2/compliance/arf_reports/ | python3 -m json.tool

```

応答例:

```

% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left  Speed
100 3252    0 3252    0    0  4319    0  ---:--:  ---:--:  ---:--:  4318
{
  "page": 1,
  "per_page": 20,
  "results": [
    {
      "created_at": "2017-05-16 13:27:09 UTC",
      "failed": 0,
      "host": "host1.example.com",
      "id": 404,
      "othered": 0,

```

```

    "passed": 0,
    "updated_at": "2017-05-16 13:27:09 UTC"
  },
  {
    "created_at": "2017-05-16 13:26:07 UTC",
    "failed": 0,
    "host": "host2.example.com",
    "id": 405,
    "othered": 0,
    "passed": 0,
    "updated_at": "2017-05-16 13:26:07 UTC"
  },
  {
    "created_at": "2017-05-16 13:25:07 UTC",
    "failed": 0,
    "host": "host3.example.com",
    "id": 406,
    "othered": 0,
    "passed": 0,
    "updated_at": "2017-05-16 13:25:07 UTC"
  },
  {
    "created_at": "2017-05-16 13:24:07 UTC",
    "failed": 0,
    "host": "host4.example.com",
    "id": 407,
    "othered": 0,
    "passed": 0,
    "updated_at": "2017-05-16 13:24:07 UTC"
  },
],
"search": null,
"sort": {
  "by": null,
  "order": null
},
"subtotal": 29,
"total": 29

```

- 1つ前の手順で取得した ID を使用して、OpenSCAP レポートを削除します。削除する ID ごとに同様の手順を繰り返します。

要求例:

```

# curl --insecure --user username:_password_ \
--header "Content-Type: application/json" \
--request DELETE https://satellite.example.com/api/v2/compliance/arf_reports/405

```

応答例:

```

HTTP/1.1 200 OK
Date: Thu, 18 May 2017 07:14:36 GMT
Server: Apache/2.4.6 (Red Hat Enterprise Linux)
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff

```

```

Foreman_version: 1.11.0.76
Foreman_api_version: 2
Apiipie-Checksum: 2d39dc59aed19120d2359f7515e10d76
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: f47eb877-35c7-41fe-b866-34274b56c506
X-Runtime: 0.661831
X-Powered-By: Phusion Passenger 4.0.18
Set-Cookie: request_method=DELETE; path=/
Set-Cookie: _session_id=d58fe2649e6788b87f46eabf8a461edd; path=/; secure; HttpOnly
ETag: "2574955fc0afc47cb5394ce95553f428"
Status: 200 OK
Vary: Accept-Encoding
Transfer-Encoding: chunked
Content-Type: application/json; charset=utf-8

```

全 OpenSCAP レポートを削除する Bash スクリプトの例

以下の Bash スクリプトを使用して、全 OpenSCAP レポートを削除します。

```

#!/bin/bash

#this script removes all the arf reports from the satellite server

#settings
USER=username
PASS=password
URI=https://satellite.example.com

#check amount of reports
while [ $(curl --insecure --user $USER:$PASS $URI/api/v2/compliance/arf_reports/ | python3 -m
json.tool | grep "\"total\": | cut --fields=2 --delimiter\":\" | cut --fields=1 --delimiter\"," | sed "s/ //g") -gt 0 ];
do

#fetch reports
for i in $(curl --insecure --user $USER:$PASS $URI/api/v2/compliance/arf_reports/ | python3 -m
json.tool | grep "\"id\": | cut --fields=2 --delimiter\":\" | cut --fields=1 --delimiter\"," | sed "s/ //g");

#delete reports
do
curl --insecure --user $USER:$PASS --header "Content-Type: application/json" --request DELETE
$URI/api/v2/compliance/arf_reports/$i
done
done

```

5.10. SATELLITE API での PULP の使用

Satellite と統合した Pulp に API 要求を送信する場合は、証明書ベースの認証を使用します。

以下の Pulp API 要求の例には、代わりに Pulp CLI を使用する方法の例が含まれます。root として **pulp** コマンドを実行すると、Pulp CLI は **/root/.config/pulp/cli.toml** で設定されたシステム証明書を使用します。

リポジトリのリスト表示

すべてのリポジトリを一覧表示するエンドポイントは `/pulp/api/v3/repositories/` です。次のクエリーは、Satellite Server から要求を発行するために必要な証明書を提供し、`satellite.example.com` からリポジトリのリストを取得します。

要求例:

```
curl --cacert /etc/pki/katello/certs/katello-server-ca.crt \  
--cert /etc/foreman/client_cert.pem --key /etc/foreman/client_key.pem \  
https://<satellite.example.com>/pulp/api/v3/repositories/ \  
| python3 -m json.tool
```

応答例:

```
{  
  "count": 1,  
  "next": null,  
  "previous": null,  
  "results": [  
    {  
      "pulp_href": "/pulp/api/v3/repositories/rpm/rpm/018cd05a-4b83-73db-b71c-587c6181d89b/",  
      "pulp_created": "2024-01-03T17:23:47.715882Z",  
      "versions_href": "/pulp/api/v3/repositories/rpm/rpm/018cd05a-4b83-73db-b71c-587c6181d89b/versions/",  
      "pulp_labels": {},  
      "latest_version_href": "/pulp/api/v3/repositories/rpm/rpm/018cd05a-4b83-73db-b71c-587c6181d89b/versions/1/",  
      "name": "Red_Hat_Enterprise_Linux_8_for_x86_64_-_BaseOS_Kickstart_8_9-49838",  
      "description": null,  
      "retain_repo_versions": null,  
      "remote": null  
    }  
  ]  
}
```

または、Pulp CLI を使用してリポジトリを一覧表示します。

```
# pulp repository list  
[  
  {  
    "pulp_href": "/pulp/api/v3/repositories/rpm/rpm/018cd025-c6ef-7237-a99e-70bab3d30941/",  
    "pulp_created": "2024-01-03T16:26:25.904682Z",  
    "versions_href": "/pulp/api/v3/repositories/rpm/rpm/018cd025-c6ef-7237-a99e-70bab3d30941/versions/",  
    "pulp_labels": {},  
    "latest_version_href": "/pulp/api/v3/repositories/rpm/rpm/018cd025-c6ef-7237-a99e-70bab3d30941/versions/1/",  
    "name": "Red_Hat_Enterprise_Linux_8_for_x86_64_-_AppStream_RPMs_8-2875",  
    "description": null,  
    "retain_repo_versions": null,  
    "remote": null  
  }  
]
```

Pulp ステータスの確認

Pulp のステータス情報を返すエンドポイントは **/pulp/api/v3/status/** です。Pulp ステータスの要求は認証を必要としません。

要求例:

```
curl https://<satellite.example.com>/pulp/api/v3/status/ \  
| python3 -m json.tool
```

応答例:

```
{  
  "versions": [  
    {  
      "component": "core",  
      "version": "3.39.4",  
      "package": "pulpcore",  
      "domain_compatible": true  
    },  
    {  
      "component": "rpm",  
      "version": "3.23.0",  
      "package": "pulp-rpm",  
      "domain_compatible": true  
    },  
    ...  
  ]  
}
```

または、Pulp CLI を使用して Pulp ステータスを取得します。

```
# pulp status  
{  
  "versions": [  
    {  
      "component": "core",  
      "version": "3.39.4",  
      "package": "pulpcore",  
      "domain_compatible": true  
    },  
    {  
      "component": "rpm",  
      "version": "3.23.0",  
      "package": "pulp-rpm",  
      "domain_compatible": true  
    },  
    ...  
  ]  
}
```

関連情報

- Pulp CLI の使用方法は、**pulp --help** を実行してください。
- Pulp の包括的な API リファレンスは、Satellite Server (<https://<satellite.example.com>/pulp/api/v3/docs/>) で確認できます。

付録A API 応答コード

Red Hat Satellite 6 API には、API 呼び出しの HTTP 応答ステータスコードがあります。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 応答コード

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

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

パーミッション名	アクション	リソースタイプ
view_audit_logs	<ul style="list-style-type: none"> audits/index audits/show audits/auto_complete_search api/v2/audits/index api/v2/audits/show 	
view_authenticators	<ul style="list-style-type: none"> auth_source_ldaps/index auth_source_ldaps/show api/v2/auth_source_ldaps/index api/v2/auth_source_ldaps/show 	
create_authenticators	<ul style="list-style-type: none"> auth_source_ldaps/new auth_source_ldaps/create api/v2/auth_source_ldaps/create 	
edit_authenticators	<ul style="list-style-type: none"> auth_source_ldaps/edit auth_source_ldaps/update api/v2/auth_source_ldaps/update 	
destroy_authenticators	<ul style="list-style-type: none"> auth_source_ldaps/destroy api/v2/auth_source_ldaps/destroy 	

パーMISSION名	アクション	リソースタイプ
view_bookmarks	<ul style="list-style-type: none">● bookmarks/index● bookmarks/show● api/v2/bookmarks/index● api/v2/bookmarks/show	
create_bookmarks	<ul style="list-style-type: none">● bookmarks/new● bookmarks/create● api/v2/bookmarks/new● api/v2/bookmarks/create	
edit_bookmarks	<ul style="list-style-type: none">● bookmarks/edit● bookmarks/update● api/v2/bookmarks/edit● api/v2/bookmarks/update	
destroy_bookmarks	<ul style="list-style-type: none">● bookmarks/destroy● api/v2/bookmarks/destroy	

パーミッション名	アクション	リソースタイプ
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 	
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

パーMISSION名	アクション	リソースタイプ
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	
destroy_compute_profiles	<ul style="list-style-type: none">● compute_profiles/destroy● api/v2/compute_profiles/destroy	

パーミッション名	アクション	リソースタイプ
view_compute_resources	<ul style="list-style-type: none"> ● compute_resources/index ● compute_resources/show ● compute_resources/automatic_complete_search ● compute_resources/ping ● compute_resources/available_images ● 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/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/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/v2/compute_resources/destroy 	
view_compute_resources_vms	<ul style="list-style-type: none"> ● compute_resources_vms/index ● compute_resources_vms/show 	

パーミッション名	アクション	リソースタイプ
create_compute_resources_vms	<ul style="list-style-type: none"> ● compute_resources_vms/new ● compute_resources_vms/create 	
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 	

パーMISSION名	アクション	リソースタイプ
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 	
view_config_reports	<ul style="list-style-type: none"> ● 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/latest 	
destroy_config_reports	<ul style="list-style-type: none"> ● config_reports/destroy ● api/v2/config_reports/destroy 	
upload_config_reports	<ul style="list-style-type: none"> ● 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

パーミッション名	アクション	リソースタイプ
commit_containers	<ul style="list-style-type: none"> containers/commit 	Container
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_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::ContentView

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/content_views_histories/index ● katello/api/v2/content_views_puppet_modules/index ● katello/api/v2/content_views_puppet_modules/show ● katello/api/v2/content_views_versions/index ● katello/api/v2/content_views_versions/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/content_views/automatic_complete ● katello/content_views/automatic_complete_search ● katello/errata/short_details ● katello/errata/automatic_complete ● katello/packages/details ● katello/packages/automatic_complete ● katello/products/automatic_complete ● katello/repositories/automatic_complete_library ● katello/content_search/index ● katello/content_search/products ● katello/content_search/repos 	

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/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/rpo_packages ● katello/content_search/rpo_errata ● katello/content_search/rpo_puppet_modules ● katello/content_search/rpo_compare_errata ● katello/content_search/rpo_compare_packages ● katello/content_search/rpo_compare_puppet_modules ● katello/content_search/view_compare_errata ● 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

パーMISSION名	アクション	リソースタイプ
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/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/show 	Host

パーMISSION名	アクション	リソースタイプ
submit_discovered_hosts	<ul style="list-style-type: none">● api/v2/discovered_hosts/facts● api/v2/discovered_hosts/create	Host
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

パーミッション名	アクション	リソースタイプ
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
view_discovery_rules	<ul style="list-style-type: none"> ● discovery_rules/index ● discovery_rules/show ● discovery_rules/autocomplete_search ● api/v2/discovery_rules/index ● api/v2/discovery_rules/show 	DiscoveryRule

パーMISSION名	アクション	リソースタイプ
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/v2/domains/indexapi/v2/domains/showapi/v2/parameters/indexapi/v2/parameters/show	
create_domains	<ul style="list-style-type: none">domains/newdomains/createapi/v2/domains/create	
edit_domains	<ul style="list-style-type: none">domains/editdomains/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/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/v2/environments/index ● api/v2/environments/show 	
create_environments	<ul style="list-style-type: none"> ● environments/new ● environments/create ● api/v2/environments/create 	
edit_environments	<ul style="list-style-type: none"> ● environments/edit ● environments/update ● api/v2/environments/update 	
destroy_environments	<ul style="list-style-type: none"> ● environments/destroy ● api/v2/environments/destroy 	
import_environments	<ul style="list-style-type: none"> ● environments/import_environments ● environments/obsolete_and_new ● api/v2/environments/import_puppetclasses ● api/v2/smart_proxies/import_puppetclasses 	

パーミッション名	アクション	リソースタイプ
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/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/v2/smart_variables/create● api/v2/smart_class_parameters/create● api/v2/override_values/create	

パーMISSION名	アクション	リソースタイプ
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/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	

パーミッション名	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/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/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MISSION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 	
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 	

パーミッション名	アクション	リソースタイプ
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 	
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

パーMISSION名	アクション	リソースタイプ
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
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

パーミッション名	アクション	リソースタイプ
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 	

パーMISSION名	アクション	リソースタイプ
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
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

パーミッション名	アクション	リソースタイプ
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/autocomplete_search ● job_invocations/show ● template_invocations/index 	TemplateInvocation
view_foreman_tasks	<ul style="list-style-type: none"> ● foreman_tasks/tasks/autocomplete_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

パーMISSION名	アクション	リソースタイプ
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
view_globals	<ul style="list-style-type: none"> ● common_parameters/index ● common_parameters/show ● common_parameters/autocomplete_search ● api/v2/common_parameters/index ● api/v2/common_parameters/show 	

パーミッション名	アクション	リソースタイプ
create_globals	<ul style="list-style-type: none"> ● common_parameters/new ● common_parameters/create ● api/v2/common_parameters/create 	
edit_globals	<ul style="list-style-type: none"> ● common_parameters/edit ● common_parameters/update ● api/v2/common_parameters/update 	
destroy_globals	<ul style="list-style-type: none"> ● 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
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

パーMISSION名	アクション	リソースタイプ
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 	

パーミッション名	アクション	リソースタイプ
create_params	<ul style="list-style-type: none">● host_editing/create_params● api/v2/parameters/create	
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/v2/hostgroups/index● api/v2/hostgroups/show	

パーMISSION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/v2/hostgroups/create● api/v2/hostgroups/clone	

パーミッション名	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/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 	

パーMISSION名	アクション	リソースタイプ
destroy_hostgroups	<ul style="list-style-type: none"> ● hostgroups/destroy ● api/v2/hostgroups/destroy 	
view_hosts	<ul style="list-style-type: none"> ● hosts/index ● hosts/show ● hosts/errors ● 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/v2/hosts/index ● api/v2/hosts/show 	

パーミッション名	アクション Configuration	リソースタイプ
	<ul style="list-style-type: none"> ● api/v2/hosts/status/configuration ● 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 ● hosts/puppet_environment_for_content_view ● katello/api/v2/host_auto_complete/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 	
create_hosts	<ul style="list-style-type: none"> ● hosts/new ● hosts/create ● hosts/clone 	

パーMISSION名	アクション	リソースタイプ
	<ul style="list-style-type: none"> ● 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/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 	

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/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 ● 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/associate ● compute_resources/[:cluster_selected, :template_selected, :provider_selected, :resource_pools] ● compute_resources_vms/associate ● puppetclasses/parameters ● subnets/freeip 	

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"> ● interfaces/new ● 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 ● 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 	

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/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/v2/hosts/destroy api/v2/interfaces/destroy katello/api/v2/hosts_bulk_actions/destroy_hosts 	
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 	

パーミッション名	アクション	リソースタイプ
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
view_images	<ul style="list-style-type: none"> ● images/index ● images/show ● images/auto_complete_search ● api/v2/images/index ● api/v2/images/show 	
create_images	<ul style="list-style-type: none"> ● images/new ● images/create ● api/v2/images/create 	

パーMISSION名	アクション	リソースタイプ
edit_images	<ul style="list-style-type: none"> ● images/edit ● images/update ● api/v2/images/update 	
destroy_images	<ul style="list-style-type: none"> ● 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/automatic_complete_search 	Katello::KTEEnvironment
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

パーミッション名	アクション	リソースタイプ
view_locations	<ul style="list-style-type: none"> ● locations/index ● locations/show ● locations/auto_complete_search ● 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/v2/locations/create 	
edit_locations	<ul style="list-style-type: none"> ● locations/edit ● locations/update ● locations/import_mismatches ● locations/parent_taxonomy_selected ● api/v2/locations/update 	
destroy_locations	<ul style="list-style-type: none"> ● 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 	

パーMISSION名	アクション	リソースタイプ
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/v2/media/index● api/v2/media/show	
create_media	<ul style="list-style-type: none">● media/new● media/create● api/v2/media/create	
edit_media	<ul style="list-style-type: none">● media/edit● media/update● api/v2/media/update	
destroy_media	<ul style="list-style-type: none">● media/destroy● api/v2/media/destroy	

パーミッション名	アクション	リソースタイプ
view_models	<ul style="list-style-type: none">models/indexmodels/showmodels/auto_complete_searchapi/v2/models/indexapi/v2/models/show	
create_models	<ul style="list-style-type: none">models/newmodels/createapi/v2/models/create	
edit_models	<ul style="list-style-type: none">models/editmodels/updateapi/v2/models/update	
destroy_models	<ul style="list-style-type: none">models/destroyapi/v2/models/destroy	

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

パーMISSION名	アクション	リソースタイプ
view_organizations	<ul style="list-style-type: none"> ● organizations/index ● organizations/show ● organizations/auto_complete_search ● 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 	
create_organizations	<ul style="list-style-type: none"> ● organizations/new ● organizations/create ● organizations/clone_taxonomy ● organizations/step2 ● organizations/nest ● api/v2/organizations/create ● katello/api/v2/organizations/create 	

パーミッション名	アクション	リソースタイプ
edit_organizations	<ul style="list-style-type: none"> ● organizations/edit ● organizations/update ● organizations/import_matches ● organizations/parent_taxonomy_selected ● 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/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 	
view_ptables	<ul style="list-style-type: none"> ● ptables/index ● ptables/show ● ptables/auto_complete_search ● ptables/revision ● ptables/preview ● api/v2/ptables/show ● api/v2/ptables/revision 	

パーMISSION名	アクション	リソースタイプ
create_ptables	<ul style="list-style-type: none"> ● ptables/new ● ptables/create ● ptables/clone_template ● api/v2/ptables/create ● api/v2/ptables/clone 	
edit_ptables	<ul style="list-style-type: none"> ● ptables/edit ● ptables/update ● api/v2/ptables/update 	
destroy_ptables	<ul style="list-style-type: none"> ● ptables/destroy ● api/v2/ptables/destroy 	
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::Product

パーミッション名	アクション ● katello/api/v2/packages/index	リソースタイプ
	<ul style="list-style-type: none"> ● 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/repositories/repository_types ● katello/content_search/index ● katello/content_search/products ● katello/content_search/repos ● katello/content_search/packages ● katello/content_search/errata 	

パーMISSION名	アクション	リソースタイプ
	<ul style="list-style-type: none"> ● katello/content_search/puppet_modules ● katello/content_search/packages_items ● katello/content_search/errata_items ● katello/content_search/puppet_modules_items ● katello/content_search/r epo_packages ● katello/content_search/r epo_errata ● katello/content_search/r epo_puppet_modules ● katello/content_search/r epo_compare_errata ● katello/content_search/r epo_compare_packages ● katello/content_search/r epo_compare_puppet_modules 	
create_products	<ul style="list-style-type: none"> ● katello/api/v2/products/create ● katello/api/v2/repositories/create 	Katello::Product

パーミッション名	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/sync● katello/api/v2/repositories/sync● katello/api/v2/products_bulk_actions/sync_products● katello/api/v2/repositories_bulk_actions/sync_repositories● katello/api/v2/sync/index● katello/api/v2/sync_plans/sync● katello/sync_management/index● katello/sync_management/sync_status● katello/sync_management/product_status● katello/sync_management/sync● katello/sync_management/destroy	Katello::Product
export_products	<ul style="list-style-type: none">● katello/api/v2/repositories/export	Katello::Product

パーミッション名	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/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/v2/template_kinds/index 	
create_provisioning_templates	<ul style="list-style-type: none"> ● provisioning_templates/new ● provisioning_templates/create ● provisioning_templates/clone_template ● 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/v2/provisioning_templates/update ● api/v2/template_combinations/update 	
destroy_provisioning_templates	<ul style="list-style-type: none"> ● provisioning_templates/destroy ● api/v2/provisioning_templates/destroy ● api/v2/template_combinations/destroy 	
deploy_provisioning_templates	<ul style="list-style-type: none"> ● provisioning_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/provisioning_templates/lock ● api/v2/provisioning_templates/unlock 	
user_logout	<ul style="list-style-type: none"> ● users/logout 	
my_account	<ul style="list-style-type: none"> ● users/edit ● katello/api/v2/tasks/show 	

パーミッション名	アクション	リソースタイプ
api_status	<ul style="list-style-type: none">● api/v2/home/status/	
view_puppetclasses	<ul style="list-style-type: none">● puppetclasses/index● puppetclasses/show● puppetclasses/auto_complete_search● api/v2/puppetclasses/index● api/v2/puppetclasses/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/v2/puppetclasses/create	

パーMISSION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/v2/puppetclasses/update ● 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/v2/puppetclasses/destroy 	
import_puppetclasses	<ul style="list-style-type: none"> ● puppetclasses/import_environments ● puppetclasses/obsolete_and_new ● api/v2/environments/import_puppetclasses ● api/v2/smart_proxies/import_puppetclasses 	

パーミッション名	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 	

パーMISSION名	アクション	リソースタイプ
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 	
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_roles	<ul style="list-style-type: none"> ● roles/index ● roles/auto_complete_search ● api/v2/roles/index ● api/v2/roles/show 	

パーミッション名	アクション	リソースタイプ
create_roles	<ul style="list-style-type: none">roles/newroles/createroles/cloneapi/v2/roles/create	
edit_roles	<ul style="list-style-type: none">roles/editroles/updateapi/v2/roles/update	
destroy_roles	<ul style="list-style-type: none">roles/destroyapi/v2/roles/destroy	
access_settings	<ul style="list-style-type: none">home/settings	

パーMISSION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/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/v2/smart_proxies/create 	

パーミッション名	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/v2/smart_proxies/update ● api/v2/smart_proxies/refresh 	
destroy_smart_proxies	<ul style="list-style-type: none"> ● 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/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 	
destroy_smart_proxies_puppetca	<ul style="list-style-type: none"> ● puppetca/destroy 	

パーMISSION名	アクション	リソースタイプ
view_subnets	<ul style="list-style-type: none">● subnets/index● subnets/show● subnets/auto_complete_search● api/v2/subnets/index● api/v2/subnets/show	
create_subnets	<ul style="list-style-type: none">● subnets/new● subnets/create● api/v2/subnets/create	
edit_subnets	<ul style="list-style-type: none">● subnets/edit● subnets/update● api/v2/subnets/update	
destroy_subnets	<ul style="list-style-type: none">● subnets/destroy● api/v2/subnets/destroy	
import_subnets	<ul style="list-style-type: none">● subnets/import● subnets/create_multiple	

パーミッション名	アクション	リソースタイプ
view_subscriptions	<ul style="list-style-type: none">katello/api/v2/subscriptions/indexkatello/api/v2/subscriptions/showkatello/api/v2/subscriptions/availablekatello/api/v2/subscriptions/manifest_historykatello/api/v2/subscriptions/auto_complete_searchkatello/api/v2/repository_sets/indexkatello/api/v2/repository_sets/showkatello/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

パーMISSION名	アクション	リソースタイプ
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
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

パーミッション名	アクション	リソースタイプ
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
	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/v2/usergroups/index api/v2/usergroups/show
	create_usergroups	<ul style="list-style-type: none"> usergroups/new usergroups/create api/v2/usergroups/create
	edit_usergroups	<ul style="list-style-type: none"> usergroups/edit usergroups/update api/v2/usergroups/update
	destroy_usergroups	<ul style="list-style-type: none"> usergroups/destroy api/v2/usergroups/destroy

パーMISSION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/v2/users/index● api/v2/users/show
	create_users	<ul style="list-style-type: none">● users/new● users/create● users/auth_source_selected● api/v2/users/create
	edit_users	<ul style="list-style-type: none">● users/edit● users/update● users/auth_source_selected● users/test_mail● api/v2/users/update
	destroy_users	<ul style="list-style-type: none">● users/destroy● api/v2/users/destroy