



Red Hat Satellite 6.12

API 指南

使用 Red Hat Satellite Representational State Transfer (REST) API 的指南

Red Hat Satellite 6.12 API 指南

使用 Red Hat Satellite Representational State Transfer (REST) API 的指南

Red Hat Satellite Documentation Team

satellite-doc-list@redhat.com

法律通告

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.

摘要

红帽卫星代表性状态转移(Rest)API 指南介绍了 REST API 背后的概念，并为各种请求提供示例使用情况。这为管理员和开发人员提供了编写自定义脚本并将 Red Hat Satellite 与第三方应用程序集成的基础。

目录

向红帽文档提供反馈	3
第 1 章 简介	4
1.1. RED HAT SATELLITE API 概述	4
1.2. SATELLITE API 与 HAMMER CLI 工具的比较	4
第 2 章 API 参考	5
2.1. 了解 API 语法	5
2.2. 了解 JSON 响应格式	9
第 3 章 API 调用身份验证	12
3.1. SSL 身份验证概述	12
3.2. HTTP 身份验证概述	12
3.3. 个人访问令牌身份验证概述	13
3.4. OAUTH 身份验证概述	14
第 4 章 不同语言的 API 请求	16
4.1. 使用 CURL 进行 API 请求	16
4.2. 使用 RUBY 进行 API 请求	19
4.3. 使用 PYTHON 的 API 请求	22
第 5 章 使用 RED HAT SATELLITE API	28
5.1. 使用主机	28
5.2. 使用生命周期环境	31
5.3. 将内容上传到卫星服务器	35
5.4. 将勘误应用到主机或主机集合	39
5.5. 使用扩展搜索	40
5.6. 使用带有 PAGINATION CONTROL 的搜索	41
5.7. 覆盖智能类参数	41
5.8. 使用外部文件修改智能类参数	43
5.9. 删除 OPENSAP 报告	45
附录 A. API 响应代码	48
附录 B. API 权限列表	49

向红帽文档提供反馈

我们感谢您对文档提供反馈信息。请让我们了解如何改进文档。

您可以通过在 Bugzilla 中记录一个 ticket 来提交反馈：

1. 导航到 [Bugzilla](#) 网站。
2. 在 **Component** 字段中，使用 **Documentation**。
3. 在 **Description** 字段中，输入您要改进的建议。包括文档相关部分的链接。
4. 点 **Submit Bug**。

第 1 章 简介

红帽卫星提供了一个具象状态传输(REST)API。API 可让软件开发人员和系统管理员在标准 Web 界面之外控制其红帽卫星环境。REST API 对于旨在将 Red Hat Satellite 的功能与通过 HTTP 访问 API 的自定义脚本或外部应用程序集成的开发人员和管理员很有用。

1.1. RED HAT SATELLITE API 概述

使用 REST API 的优点包括：

- 广泛的客户端支持 - 支持 HTTP 协议的任何编程语言、框架或系统都可以使用 API。
- 自助服务应用程序 - 客户端应用程序需要最少的 Red Hat Satellite 基础架构知识，因为用户在运行时发现多个详情。
- 基于资源的模式 - 基于资源的 REST 模型提供了管理虚拟化平台的自然方式。

您可以使用 REST API 执行以下任务：

- 与企业 IT 系统集成。
- 与第三方应用程序集成。
- 执行自动化维护或错误检查任务。
- 使用脚本自动执行重复任务。

当您准备升级卫星服务器时，请确保使用包含 Satellite API 命令的任何脚本都保持最新状态。API 命令因 Satellite 版本而异。

1.2. SATELLITE API 与 HAMMER CLI 工具的比较

对于许多任务，您可以使用 Hammer 和 Satellite API。您可以使用 Hammer 作为卫星 API 的人类可读界面。例如，要在脚本中应用对 API 调用的响应，请使用 `--debug` 选项检查 Hammer 问题的 API 调用：
hammer --debug 组织列表。

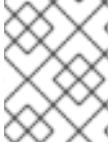
在背景中，每个 Hammer 命令首先建立到 API 的绑定，然后发送请求。这在按顺序执行大量 Hammer 命令时存在性能影响。与之相反，使用 API 命令的脚本直接与卫星 API 通信。

请注意，您必须手动更新使用 API 命令的脚本，而 Hammer 会自动反映 API 中的更改。如需更多信息，请参阅 [Hammer CLI 指南](#)。

第 2 章 API 参考

完整的 API 引用位于 <https://satellite.example.com/apidoc/v2.html> 的卫星服务器上。请注意，即使有 Satellite 6 API 的版本 1 和 2，红帽仅支持版本 2。

2.1. 了解 API 语法



注意

以下示例请求使用 **python3** 来格式化 Satellite 服务器中的 response。在 RHEL 7 和一些较旧的系统中，您必须使用 **python** 而不是 **python3**。

内置 API 引用显示 API 路由或路径，前面带有 HTTP 动词：

```
HTTP_VERB API_ROUTE
```

要使用 API，请使用 **curl** 命令语法和参考文档中的 API 路由来构造命令：

```
$ 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 要将 **curl** 用于 API 调用，请使用 **--request** 选项指定 HTTP 动词。例如：**--request POST**。
- 2 添加 **--insecure** 选项以跳过 SSL peer 证书验证检查。
- 3 使用 **--user** 选项提供用户凭据。
- 4 对于 **POST** 和 **PUT** 请求，请使用 **--data** 选项传递 JSON 格式的数据。更多信息请参阅 [第 4.1.1 节“将 JSON 数据传递给 API 请求”](#)。
- 5 6 使用 **--data** 选项传递 JSON 数据时，您必须使用 **--header** 选项指定以下标头。更多信息请参阅 [第 4.1.1 节“将 JSON 数据传递给 API 请求”](#)。
- 7 从卫星服务器下载内容时，请使用 **--output** 选项指定输出文件。
- 8 使用以下格式的 API 路由：**https://satellite.example.com/katello/api/activation_keys**。在 Satellite 6 中，API 的版本 2 是默认值。因此，不需要在 API 调用的 URL 中使用 **v2**。
- 9 将输出重定向到 Python **json.tool** 模块，以便更轻松地读取输出。

2.1.1. 使用 GET HTTP Verb

使用 GET HTTP 动词从 API 中获取现有条目或资源的数据。

示例

这个示例请求 Satellite 主机的数量：

请求示例：

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

响应示例：

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

来自 API 的响应表示总共有两个结果，这是结果的第一个页面，每个页面最大结果设置为 20。更多信息请参阅 [第 2.2 节“了解 JSON 响应格式”](#)。

2.1.2. 使用 POST HTTP Verb

使用 POST HTTP 动词将数据提交到 API 以创建条目或资源。您必须使用 JSON 格式提交数据。更多信息请参阅 [第 4.1.1 节“将 JSON 数据传递给 API 请求”](#)。

示例

这个示例创建了一个激活码。

1. 创建一个测试文件，如 **activation-key.json**，其内容如下：

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

2. 通过在 **activation-key.json** 文件中应用数据来创建激活码：
请求示例：

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

响应示例：

```
{
  "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. 验证新激活码是否存在。在 Satellite Web UI 中，导航到 **Content > Activation keys** 以查看您的激活码。

2.1.3. 使用 PUT HTTP Verb

使用 PUT HTTP 动词更改现有值或附加到现有资源。您必须使用 JSON 格式提交数据。更多信息请参阅第 4.1.1 节“将 JSON 数据传递给 API 请求”。

示例

本例更新了上例中创建的 **TestKey** 激活码。

1. 编辑之前创建的 **activation-key.json** 文件，如下所示：

```

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

```

2. 应用 JSON 文件中的更改：
请求示例：

```

$ 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
```

响应示例：

```
{
  "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. 在 Satellite Web UI 中，导航到 **Content > Activation keys** 来验证更改。

2.1.4. 使用 DELETE HTTP Verb

要删除资源，可将 DELETE 动词与 API 路由一起使用，该路由中包含您要删除的资源 ID。

示例

这个示例删除 ID 为 2 的 **TestKey** 激活码：

请求示例：

```
$ 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
```

响应示例：

```
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. 将 API 错误消息与 API 引用有关

API 使用 RAILS 格式来指示错误：

```
Nested_Resource.Attribute_Name
```

这会转换为 API 引用中使用的以下格式：

```
Resource[Nested_Resource_attributes][Attribute_Name_id]
```

2.2. 了解 JSON 响应格式

调用 API 返回的结果，采用 JSON 格式。API 调用返回单选响应或响应集合的结果。



注意

以下示例请求使用 **python3** 来格式化 Satellite 服务器中的 response。在 RHEL 7 和一些较旧的系统中，您必须使用 **python** 而不是 **python3**。

单一对象的 JSON 响应格式

您可以使用单一对象 JSON 响应来操作单个对象。对单个对象的 API 请求需要对象的唯一标识符 **:id**。

这是 Satellite 域的单对象请求格式的示例，ID 为 23：

请求示例：

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

响应示例：

```
{
```

```
"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"  
}
```

集合的 JSON 响应格式

集合是主机和域等对象列表。集合 JSON 响应的格式由 metadata 字段和 results 部分组成。

以下是 Satellite 域列表的集合请求格式的示例：

请求示例：

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

响应示例：

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

```
{  
  ]  
}
```

响应元数据字段

API 响应使用以下元数据字段：

- **total** - 没有搜索参数的对象总数。
- **subtotal** - 返回给定搜索参数的对象数量。如果没有搜索，则 subtotal 等于 total。
- **页面** - 页面号。
- **per_page** - 每个页面返回的最大对象数量。
- **limit** - 在集合响应中返回指定的对象数量。
- **offset** - 返回集合前跳过的对象数量。
- **search** - 基于 **scoped_scoped** 语法的搜索字符串。
- **sort**
 - **by** - 通过什么字段指定 API 对集合进行排序。
 - **order** - 排序顺序，即 ASC 升序，或 DESC 表示降序。
- **结果** - 对象的集合。

第 3 章 API 调用身份验证

与 Satellite API 交互需要与卫星服务器 CA 证书进行 SSL 身份验证，并与有效的 Satellite 用户凭证进行身份验证。本章概述了可以使用的验证方法。

3.1. SSL 身份验证概述

Red Hat Satellite 使用 HTTPS，在与 Red Hat Satellite Server 通信时提供一定程度的加密和身份验证。Satellite 6.12 不支持非 SSL 通信。

每个红帽卫星服务器都使用自签名证书。此证书同时充当服务器证书，以验证加密密钥和证书颁发机构 (CA) 以信任卫星服务器的身份。

3.1.1. 配置 SSL 身份验证

使用以下步骤为 API 请求配置到卫星服务器的 SSL 身份验证。

流程

1. 从卫星服务器获取您要使用以下选项之一进行通信的证书：

- 如果您从远程服务器执行命令，请使用 SSH 获取证书：

```
$ 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
```

- 如果您直接在 Satellite 服务器上执行命令，请将证书复制到 `/etc/pki/ca-trust/source/anchors` 目录中：

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

2. 将证书添加到可信 CA 列表中：

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

验证

- 输入没有 `--cacert` 选项的 API 请求来验证 NSS 数据库中是否存在证书：

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

3.2. HTTP 身份验证概述

所有对 Satellite API 的请求都需要有效的 Satellite 用户名和密码。API 使用 HTTP 基本身份验证对这些凭证进行编码，并添加到 **Authorization** 标头中。有关基本身份验证的更多信息，请参阅 [RFC 2617 HTTP Authentication: Basic](#) 和 [Digest Access Authentication](#)。如果请求没有包括适当的 **Authorization** 标头，API 会返回 **401 Authorization Required** 错误。



重要

基本身份验证涉及潜在的敏感信息，例如，它会以纯文本形式发送密码。REST API 需要 HTTPS 进行纯文本请求的传输级别加密。

有些 base64 库可将编码的凭证拆分为多行，并使用换行符终止每行。这会导致标头无效，并导致一个错误的请求。Authorization 标头要求编码的凭证位于标题中的一行中。

3.3. 个人访问令牌身份验证概述

Red Hat Satellite 支持可用于验证 API 请求的个人访问令牌，而不使用您的密码。您可以为个人访问令牌设置过期日期。如果您决定它应在过期日期前过期，可以撤销它。

3.3.1. 创建一个个人访问令牌

使用这个流程创建个人访问令牌。

流程

1. 在 Satellite Web UI 中，进入到 **Administer > Users**。
2. 选择您要为其创建个人访问令牌的用户。
3. 在 **Personal Access Tokens** 选项卡中，点 **Add Personal Access Token**。
4. 输入个人访问令牌的名称。
5. 可选：选择 **Expires** 日期来设置过期日期。如果您没有设置过期日期，您的个人访问令牌将永远不会过期，除非被撤销。
6. 点 Submit。到此，在 **Personal Access Tokens** 选项卡会包括您可用的个人访问令牌。



重要

确保存储您的个人访问令牌，因为您将无法在离开页面或创建新的个人访问令牌后再次访问它。您可以点 **Copy to clipboard** 复制个人访问令牌。

验证

1. 向 Satellite 服务器发出 API 请求，并使用您的个人访问令牌进行身份验证：

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

2. 您应该收到状态为 **200** 的响应，例如：

```
{"satellite_version":"6.12.0","result":"ok","status":200,"version":"3.5.1.10","api_version":2}
```

如果您再次返回 **Personal Access Tokens** 选项卡，您可以在个人访问令牌旁看到**最后使用时间**。

3.3.2. 撤销个人访问令牌

使用这个流程在过期日期前撤销个人访问令牌。

流程

1. 在 Satellite Web UI 中，进入到 **Administer > Users**。
2. 选择您要撤销个人访问令牌的用户。
3. 在 **Personal Access Tokens** 选项卡中，找到您要撤销的个人访问令牌。
4. 在您要撤销的个人访问令牌旁边的 **Actions** 列中点 **Revoke**。

验证

1. 向 Satellite 服务器发出 API 请求，并尝试使用撤销的个人访问令牌进行身份验证：

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

2. 您会收到以下出错信息：

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

3.4. OAUTH 身份验证概述

作为基本身份验证的替代选择，您可以使用有限的 OAuth 1.0 身份验证。这有时被称为协议 1.0a 中的 1 分支 OAuth。

要查看 Satellite Web UI 中的 OAuth 设置，请导航至 **Administer > Settings > Authentication**。OAuth 使用者密钥是供所有 OAuth 客户端使用的令牌。

Satellite 将 OAuth 设置存储在 `/etc/foreman/settings.yaml` 文件中。使用 `satellite-installer` 脚本配置这些设置，因为 Satellite 在升级时会覆盖对此文件的任何手动更改。

3.4.1. 配置 OAuth

要更改 OAuth 设置，请使用所需选项输入 `satellite-installer`。输入以下命令列出所有 OAuth 相关安装程序选项：

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

启用 OAuth 映射

默认情况下，Satellite 将所有 OAuth API 请求授权为内置的匿名 API 管理员帐户。因此，API 响应包括所有 Satellite 数据。但是，您也可以指定 Foreman 用户发出请求并限制对该用户的数据访问。

要启用 OAuth 用户映射，请输入以下命令：

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

**重要**

卫星没有为 OAuth 请求中的标头签名。具有有效使用者键的任何人都可以模拟任何 Foreman 用户。

3.4.2. OAuth 请求格式

每个 OAuth API 请求都需要一个带有现有 Foreman 用户的登录和 **Authorization** 标头的 **FOREMAN-USER** 标头，格式为：

```
--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**

使用 OAuth 客户端库构建所有 OAuth 参数。有关使用 `requests_oauthlib` Python 模块的示例，请参阅红帽知识库中的 [如何通过 Red Hat Satellite 6 中的 python 脚本使用 OAuth 身份验证方法执行 API 调用](#)。

示例

本例列出了使用 OAuth 进行身份验证的架构。请求在 **FOREMAN-USER** 标头中使用 `sat_username` 用户名。当 `--foreman-oauth-map-users` 被设置为 `true` 时，响应仅包含用户有权访问的构架。签名反映了每个参数、HTTP 方法和 URI 更改。

请求示例：

```
$ 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='
```

其他资源

- [OAuth Core 1.0 修订 A 的文档](#)

第 4 章 不同语言的 API 请求

本章概述了通过 curl、Ruby 和 Python 向 Red Hat Satellite 发送 API 请求，并提供示例。

4.1. 使用 CURL 进行 API 请求

本节概述了如何使用 **curl** 和 Satellite API 执行各种任务。

Red Hat Satellite 需要使用 HTTPS，默认证书进行主机识别。如果您还没有添加 Satellite 服务器证书，如 [第 3.1 节“SSL 身份验证概述”](#) 所述，您可以使用 **--insecure** 选项绕过证书检查。

对于用户身份验证，您可以使用 **--user** 选项以 **--user *username:password*** 格式提供卫星用户凭证，或者如果您未包含密码，该命令会提示您输入该密码。要降低安全风险，不要将密码包括在命令中，因为它随后成为您的 shell 历史记录的一部分。本节中的示例只包括用于简单起性的密码。

请注意，如果您使用 **--silent** 选项，**curl** 不会显示进度计量或任何错误消息。

本章示例使用 Python **json.tool** 模块来格式化输出。

4.1.1. 将 JSON 数据传递给 API 请求

您可以使用 API 请求将数据传递给卫星服务器。数据必须采用 JSON 格式。当使用 **--data** 选项指定 JSON 数据时，您必须使用 **--header** 选项设置以下 HTTP 标头：

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

使用以下选项之一使用 **--data** 选项包括数据：

1. 用大括号 **{}** 括起的带引号 JSON 格式的数据。当为 JSON 类型参数传递值时，您必须转义引号 **"** with backslashes ****。例如，在大括号中，您必须将 **"Example JSON Variable"** 格式化为 **\"Example JSON Variable\"**：

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

2. 未加引号的 JSON 格式数据包括在文件中，并由 **@** 符号和文件名指定。例如：

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

对 JSON 格式的数据使用外部文件有以下优点：

- 您可以使用您首选的文本编辑器。
- 您可以使用语法检查器来查找和避免错误。
- 您可以使用工具检查 JSON 数据的有效性或重新格式化它。

验证 JSON 文件

使用 **json_verify** 工具检查 JSON 文件的有效性：

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

4.1.2. 检索资源列表

本节概述了如何使用 Satellite 6 API 使用 **curl** 来从 Satellite 部署请求信息。这些示例包括 requests 和 响应。每个部署都有不同的结果。



注意

以下示例请求使用 **python3** 来格式化 Satellite 服务器中的 response。在 RHEL 7 和一些较旧的系统中，您必须使用 **python** 而不是 **python3**。

列出用户

这个示例是返回 Satellite 资源列表的基本请求，本例中为 Satellite 用户。这些请求会返回元数据中嵌套的数据列表，其他请求类型仅返回实际对象。

请求示例：

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

响应示例：

```
{
  "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. 创建和修改资源

本节概述了如何使用 **curl** 和 Satellite 6 API 来操作卫星服务器上的资源。这些 API 调用要求您通过 API 调用以 **json** 格式传递数据。更多信息请参阅 [第 4.1.1 节“将 JSON 数据传递给 API 请求”](#)。

创建用户

这个示例使用 **--data** 选项创建用户，以提供所需信息。

请求示例：

```
$ 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_sour
ce_id\":1}" \
https://satellite.example.com/api/users | python3 -m json.tool
```

修改用户

这个示例修改在 [创建用户](#) 中创建的 first name 并登录 `test_user`。

请求示例：

```
$ 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\",\"auth
_source_id\":1}" \
https://satellite.example.com/api/users/8 | python3 -m json.tool
```

4.2. 使用 RUBY 进行 API 请求

本节概述了如何使用 Ruby 与 Satellite API 执行各种任务。



重要

这些是示例脚本和命令。在使用前，请确保仔细检查这些脚本，并替换任何变量、用户名、密码和其他信息，以适应您自己的部署。

4.2.1. 使用 Ruby 创建对象

此脚本连接到 Red Hat Satellite 6 API，并创建组织，然后在组织中创建三个环境。如果该组织已存在，脚本将使用该组织。如果组织中已存在任何环境，该脚本会引发错误并退出。

```
#!/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,
```

```

:uri => 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,
    :uri => location,
    :user => $username,
    :password => $password,
    :headers => { :accept => :json,
:content_type => :json},
    :payload => json_data
  ).execute
  JSON.parse(response.to_str)
end

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

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

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

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

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

Apipie 绑定是 pipie 记录的 API 调用的 Ruby 绑定。它们从 Satellite 获取并缓存 API 定义，然后根据需要生成 API 调用。本例创建组织，然后在组织中创建三个环境。如果该组织已存在，脚本将使用该组织。如果组织中已存在任何环境，该脚本会引发错误并退出。

```

#!/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: \t#{org_name}"
  org_id = call_api(:organizations, :create, {'organization' => { :name => org_name }})['id']
else
  # Our organization exists, so let's grab it
  org_id = org_list.key(org_name)

```

```

puts "Organization \"#{org_name}\" exists"
end

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

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

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

```

4.3. 使用 PYTHON 的 API 请求

本节概述了如何使用 Python 和 Satellite API 执行各种任务。



重要

这些是示例脚本和命令。在使用前，请确保仔细检查这些脚本，并替换任何变量、用户名、密码和其他信息，以适应您自己的部署。

本节中的示例脚本不使用 SSL 验证与 REST API 交互。

4.3.1. 使用 Python 创建对象

此脚本连接到 Red Hat Satellite 6 API，并创建组织，然后在组织中创建三个环境。如果该组织已存在，脚本将使用该组织。如果组织中已存在任何环境，该脚本会引发错误并退出。

Python 2 示例

```

#!/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. 使用 Python 从 API 请求信息

这是一个示例脚本，它使用 Python 进行各种 API 请求。

Python 2 示例

```

#!/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 示例

```
#!/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 使用 RED HAT SATELLITE API

本章介绍了如何使用 Red Hat Satellite API 执行不同任务的示例。您可以通过 HTTPS 在端口 443 上使用 API，或者在端口 8443 上通过 HTTPS 在胶囊服务器中使用。

您可以在脚本本身中处理这些不同的端口要求。例如，在 Ruby 中，您可以指定 Satellite 和 Capsule URL，如下所示：

```
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/'
```

对于订阅到卫星服务器或胶囊服务器的主机，您可以在 **[server]** 部分的端口条目中确定从 `/etc/rhsm/rhsm.conf` 文件访问 API 所需的正确端口。您可以使用这些值来完全自动化脚本，删除任何需要验证要使用哪些端口。

本章使用 **curl** 来发送 API 请求。更多信息请参阅 [第 4.1 节“使用 curl 进行 API 请求”](#)。

本章示例使用 Python **json.tool** 模块来格式化输出。

5.1. 使用主机



注意

以下示例请求使用 **python3** 来格式化 Satellite 服务器中的 response。在 RHEL 7 和一些较旧的系统中，您必须使用 **python** 而不是 **python3**。

列出主机

本例返回 Satellite 主机列表。

请求示例：

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

响应示例：

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


为主机请求信息

此请求返回主机 **satellite.example.com** 的信息。

请求示例：

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

响应示例：

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

列出主机事实

此请求返回主机 **satellite.example.com** 的所有事实。

请求示例：

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

响应示例：

```
{
  ...
  "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",
      ...
    }
  }
}
```

使用匹配模式搜索主机

此查询会返回与模式"example"匹配的所有主机。

请求示例：

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

响应示例：

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

在环境中搜索主机

此查询会返回生产环境中的所有主机。

请求示例：

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

响应示例：

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

使用特定事实值搜索主机

此查询会返回所有带有模型名称 **RHEV Hypervisor** 的主机。

请求示例：

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

响应示例：

-

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

删除主机

此请求将删除名称为 *host1.example.com* 的主机。

请求示例：

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

下载完整引导磁盘镜像

此请求根据其 ID 为主机下载完整引导磁盘镜像。

请求示例：

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

5.2. 使用生命周期环境

卫星将应用程序生命周期划分为不同的生命周期环境，这些环境代表应用程序生命周期的每个阶段。生命周期环境从路径链接到。要使用 API 创建链接的生命周期环境，请使用 **prior_id** 参数。

您可以在 https://satellite.example.com/apidoc/v2/lifecycle_environments.html 找到用于生命周期环境的内置 API 参考。API 路由包括 `/katello/api/environments` 和 `/katello/api/organizations/:organization_id/environments`。



注意

以下示例请求使用 **python3** 来格式化 Satellite 服务器中的 response。在 RHEL 7 和一些较旧的系统中，您必须使用 **python** 而不是 **python3**。

列出生命周期环境

使用此 API 调用列出您 Satellite 上默认机构的所有当前生命周期环境，ID 为 **1**。

请求示例：

```
$ 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`
```

响应示例：

```
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
```

创建链接的生命周期环境

使用本例来创建生命周期环境路径。

此流程使用 ID 为 **1** 的默认库环境，作为创建生命周期环境的起点。

1. 选择您要用作起点的现有生命周期环境。使用其 ID 列出环境，在这个示例中，ID 为 **1** 的环境：
请求示例：

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

响应示例：

```
output omitted
"id": 1,
"label": "Library",
output omitted
"prior": null,
"successor": null,
output truncated
```

2. 创建一个 JSON 文件，如 **life-cycle.json**，其中包含以下内容：

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

3. 将 **prior** 选项设置为 **1** 来创建生命周期环境。

请求示例：

```
$ 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
```

响应示例：

```
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
```

在命令输出中，您可以看到此生命周期环境的 ID 为 **2**，且该生命周期环境是 **1**。使用 ID 为 **2** 的生命周期环境，为这个环境创建一个成功。

4. 编辑之前创建的 **生命周期.json** 文件，更新 **标签、名称** 和 **之前** 的值。

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

5. 将 **prior** 选项设置为 **2** 来创建生命周期环境。

请求示例：

```
$ 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
```

响应示例：

```

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

```

在命令输出中，您可以看到此生命周期环境的 ID 是 **3**，且该生命周期环境是 **2**。

更新生命周期环境

您可以使用 PUT 命令更新生命周期环境。

这个示例请求更新了生命周期环境的描述，其 ID 为 **3**。

请求示例：

```

$ 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

```

响应示例：

```

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
```

删除生命周期环境

您可以删除一个没有后续者的生命周期环境。因此，使用以下格式的命令以相反顺序删除它们：

请求示例：

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

5.3. 将内容上传到卫星服务器

本节概述了如何使用 Satellite 6 API 将大型文件上传并导入到 Satellite 服务器中。这个过程涉及四个步骤：

1. 创建上传请求。
2. 上传内容。
3. 导入内容。
4. 删除上传请求。

您可以上传的最大文件大小为 2MB。有关上传更大内容的详情请参考 [上传大于 2 MB 的内容](#)。

流程

1. 将软件包名称分配给 **变量名称**：

请求示例：

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

2. 将文件的 checksum 分配给变量 **checksum**：

请求示例：

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

3. 将文件大小分配给变量 **大小**：

请求示例：

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

4. 以下命令创建一个新的上传请求，并使用 **大小** 和 **checksum** 返回请求的上传 ID。

请求示例：

```
$ 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
```

在这种情况下，76 是一个存储库 ID 示例。

请求示例：

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

5. 将上传 ID 分配给变量 **upload_id**：

请求示例：

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

6. 分配您要上传到变量 **路径的软件包路径**：

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

7. 上传您的内容。确定在您上传数据时使用正确的 MIME 类型。API 使用 application/json MIME 类型，大多数请求都对 Satellite 6 使用。组合 upload_id、MIME 类型和其他参数来上传内容。

请求示例：

```
$ 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
```

8. 将内容上传到卫星服务器后，您需要将其导入到适当的存储库。在完成此步骤前，卫星服务器不会检测到新内容。

请求示例：

```
$ 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
```

9. 成功上传并导入内容后，您可以删除上传请求。这会释放在上传过程中使用的所有临时磁盘空间。

请求示例：

```
$ 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
```

上传大于 2 MB 的内容

以下示例演示了如何将大文件分成块，创建上传请求，上传单个文件，将其导入到 Satellite，然后删除上传请求。请注意，本示例使用示例内容、主机名、用户名、存储库 ID 和文件名。

1. 将软件包名称分配给 **变量名称**：

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

2. 将文件的 checksum 分配给变量 **checksum**：

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

3. 将文件大小分配给变量 **大小**：

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

4. 以下命令创建一个新的上传请求，并使用 **大小** 和 **checksum** 返回请求的上传 ID。
请求示例：

```
$ 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
```

在这种情况下，76 是一个存储库 ID 示例。

输出示例

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

5. 将上传 ID 分配给变量 **upload_id**：

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

6. 将文件分成 2MB 块：

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

输出示例

```
$ 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
```

7. 将拆分文件的前缀分配给变量路径。

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

8. 上传文件块。偏移时间从 0 开始，第一个块的块开始，为每个文件增加 2000000。注意使用偏移参数以及如何与文件大小相关。另请注意，索引在 path 变量后使用，如 `${path}0`、`${path}1`。请求示例：

```
$ 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. 将完整上传导入到存储库：

```
$ 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. 删除上传请求：

```
$ 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
```

上传重复内容

请注意，如果您试图使用以下内容上传重复的内容：

请求示例：

```
$ 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
```

这个调用会返回内容单元 ID 而不是上传 ID，如下所示：

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

您可以复制此输出并直接调用导入上传，将内容添加到存储库中：

请求示例：

```
$ 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
```

请注意，调用使用 **upload_id** 更改为使用 **content_unit_id**。

5.4. 将勘误应用到主机或主机集合

您可以使用 API 将勘误表应用到主机、主机组或主机集合。以下是 PUT 请求的基本语法：

```
$ 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
```

您可以浏览 API doc 内置的文件来查找用于应用勘误的 URL。您可以使用 Satellite Web UI 来帮助发现搜索查询的格式。进入 **Hosts > Host Collections** 并选择一个主机集合。进入 **Collection Actions > Errata Installation** 并注意搜索查询框内容。例如，对于名为 *my-collection* 的主机集合，搜索框包含 **host_collection="my-collection"**。

将勘误应用到主机

这个示例使用 API URL 用于批量操作 **/katello/api/hosts/bulk/install_content**，以显示简单搜索所需的格式。

请求示例：

```
$ 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
```

将勘误应用到主机集合

在本例中，请注意传递搜索字符串 **host_collection="my-collection"** 所需的级别，如 Satellite Web UI 所示。

请求示例：

```
$ 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. 使用扩展搜索

您可以在 Web UI 中查找用于构建搜索查询的搜索参数。如需更多信息，请参阅 [管理 Red Hat Satellite 中的构建搜索查询](#)。

例如，要搜索主机，请完成以下步骤：

1. 在 Satellite Web UI 中，导航到 **Hosts > All Hosts**，然后点击 **Search** 字段来显示搜索参数列表。
2. 找到您要使用的搜索参数。在本例中，找到 **os_title** 和 **模型**。
3. 在您的 API 查询中组合搜索参数，如下所示：
请求示例：

```
$ 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
```



注意

示例请求使用 **python3** 来格式化 Satellite 服务器中的 response。在 RHEL 7 和一些较旧的系统中，您必须使用 **python** 而不是 **python3**。

响应示例：

```
{
  ...
  "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. 使用带有 PAGINATION CONTROL 的搜索

您可以使用 `per_page` 和 `page` 分页参数来限制 API 搜索查询返回的搜索结果。`per_page` 参数指定每个页面的结果数，`page` 参数指定按 `per_page` 参数计算的页面，以便返回。

当您不指定任何分页参数时，要返回的默认项数量被设置为 1000，但 `per_page` 值的默认值为 20，在指定 `page` 参数时应用默认值。

列出内容视图

本例在页面中返回一个 Content Views 列表。列表中每个页面包含 10 个键并返回第三个页面。

请求示例：

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

列出激活码

这个示例返回带有 ID 1 在页中的机构的激活码列表。列表中每个页面包含 30 个密钥，并返回第二页。

请求示例：

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

返回多个页面

您可以使用 `for` 循环结构获取多个结果页面。

这个示例将页面 1 返回为 3 个内容视图，每个页面有 5 个结果：

```
$ 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. 覆盖智能类参数

您可以使用 API 搜索智能参数并提供一个值来覆盖类中的智能参数。您可以在 https://satellite.example.com/apidoc/v2/smart_class_parameters/update.html 的内置 API 参考中找到可以修改的属性的完整列表。

1. 查找您要更改的 Smart Class 参数的 ID：

- 列出所有智能类参数。

请求示例：

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

- 如果您知道 Puppet 类 ID，如 5，您可以限制范围：

请求示例：

■

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

这两个调用都接受 `search` 参数。您可以在 Satellite Web UI 中查看可搜索字段的完整列表。导航到 **Configure > Smart variables**，然后在搜索查询框中点击以显示字段列表。

两个特别有用的搜索参数是 `puppetclass_name` 和 `key`，可用于搜索特定参数。例如，使用 `--data` 选项传递 URL 编码数据。

请求示例：

```
$ 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 支持标准范围搜索语法。

- 找到参数的 ID 时，列出包括当前覆盖值的完整详情。

请求示例：

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

- 启用覆盖参数值。

请求示例：

```
$ 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
```

请注意，您无法手动创建或删除参数。您只能修改其属性。Satellite 仅在从代理导入的类时创建和删除参数。

- 添加自定义覆盖匹配器。

请求示例：

```
$ 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
```

有关覆盖值的更多信息，请参阅

https://satellite.example.com/apidoc/v2/override_values.html。

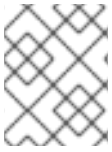
- 您可以删除覆盖值。

请求示例：

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

5.8. 使用外部文件修改智能类参数

使用外部文件简化了使用 JSON 数据的过程。使用语法高亮显示的编辑器可以帮助您避免和查找错误。



注意

以下示例请求使用 **python3** 来格式化 Satellite 服务器中的 response。在 RHEL 7 和一些较旧的系统中，您必须使用 **python** 而不是 **python3**。

使用外部文件修改智能类参数

本例使用 MOTD Puppet 清单。

1. 按名称搜索 Puppet 类 **motd**。
请求示例：

```
$ 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. 检查以下输出：每个智能类参数都有一个 ID，对于同一 Satellite 实例来说是全局的。**motd** 类的 **内容** 参数在这一卫星服务器中具有 **id=3**。不要将此与 Puppet Class 名称前面显示的 Puppet Class ID 混淆。
响应示例：

```
{
  "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\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": ""
}
```

3. 使用参数 ID **3** 获取特定于 **motd** 参数的信息，并将输出重定向到文件，如 **output_file.json**。

请求示例：

```
$ 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
```

4. 将上一步中创建的文件复制到要编辑的新文件，例如 **changed_file.json**：

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

5. 修改文件中所需的值。在本例中，将 **motd** 模块的 **content** 参数更改，这需要将 **override** 选项从 **false** 改为 **true**：

```
{
  "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. 编辑文件后，请验证其是否如下所示，然后保存更改：

```
{
  "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. 将更改应用到 Satellite 服务器：

```

$ 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. 删除 OPENSAP 报告

在卫星服务器中，您可以删除一个或多个 OpenSCAP 报告。但是，当您删除报告时，您必须一次删除一个页面。如果要删除所有 Openscap 报告，请使用如下 bash 脚本。



注意

以下示例请求和示例脚本使用 **python3** 来格式化来自 Satellite 服务器的响应。在 RHEL 7 和一些较旧的系统中，您必须使用 **python** 而不是 **python3**。

删除 OpenSCAP 报告

要删除 OpenSCAP 报告，请完成以下步骤：

1. 列出所有 OpenSCAP 报告。记录您要删除的报告 ID。

请求示例：

```

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

```

响应示例：

```

% 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

```

2. 使用上一步中的 ID，删除 OpenSCAP 报告。重复您要删除的每个 ID。
请求示例：

```

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

```

响应示例：

```

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
ApiPie-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
```

BASH 脚本删除所有 OpenSCAP 报告示例

使用以下 bash 脚本删除所有 OpenSCAP 报告：

```
#!/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
```

附录 A. API 响应代码

Red Hat Satellite 6 API 为 API 调用提供 HTTP 响应状态代码。以下代码适用于 Satellite API 中所有资源：

表 A.1. API 响应代码

响应	解释
200 OK	对于成功请求操作：显示、索引、更新或删除（GET、PUT、DELETE 请求）。
201 已创建	用于成功的创建操作（POST 请求）。
301 永久移动	当 Satellite 限制为使用 HTTPS 但尝试使用 HTTP 时，会进行重定向。
400 错误请求	缺少需要的参数，或者搜索查询具有无效的语法。
401 未授权	无法授权用户（例如，不正确的凭证）。
403 Forbidden	用户没有足够的权限来执行操作或读取资源，或一般不支持该操作。
404 not Found	带有给定 ID 的记录不存在。当请求的记录不存在时，它可能出现在显示和删除操作中；或者，当其中一个关联的记录不存在时，可以对其进行创建、更新和删除操作。
409 冲突	由于现有依赖项（例如，带有主机的主机组）无法删除记录。
415 不支持的 Media 类型	HTTP 请求的内容类型不是 JSON。
422 不可处理实体	因验证错误而创建实体失败。仅适用于创建或更新操作。
500 内部服务器错误	意外的内部服务器错误。
503 服务 Unavailable	服务器没有运行。

附录 B. API 权限列表

红帽卫星 6 API 支持大量需要特定权限的操作。下表列出了 API 权限名称、与这些权限关联的操作以及关联的资源类型。

表 B.1. API 权限列表

权限名称	Actions	资源类型
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

权限名称	Actions	资源类型
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 architecture/show architectures/auto_complete_search api/v2/architectures/index api/v2/architectures/show 	
create_architectures	<ul style="list-style-type: none"> architecture/new architectures/create api/v2/architectures/create 	
edit_architectures	<ul style="list-style-type: none"> architecture/edit architectures/update api/v2/architectures/update 	
destroy_architectures	<ul style="list-style-type: none"> architectures/destroy api/v2/architectures/destroy 	

权限名称	Actions	资源类型
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 	

权限名称	Actions	资源类型
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">● 书签/更新● 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	

权限名称	Actions	资源类型
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

权限名称	Actions	资源类型
view_capsule_content	<ul style="list-style-type: none"> ● smart_proxies/pulp_storage ● smart_proxies/pulp_status ● smart_proxies/show_with_content 	SmartProxy
view_compute_profiles	<ul style="list-style-type: none"> ● compute_profiles/index ● compute_profiles/show ● compute_profiles/auto_complete_search ● api/v2/compute_profiles/index ● api/v2/compute_profiles/show 	
create_compute_profiles	<ul style="list-style-type: none"> ● compute_profiles/new ● compute_profiles/create ● api/v2/compute_profiles/create 	
edit_compute_profiles	<ul style="list-style-type: none"> ● compute_profiles/edit ● compute_profiles/update ● api/v2/compute_profiles/update 	
destroy_compute_profiles	<ul style="list-style-type: none"> ● compute_profiles/destroy ● api/v2/compute_profiles/destroy 	

权限名称	Actions	资源类型
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_pools 	

权限名称	Actions	资源类型
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 	

权限名称	Actions	资源类型
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 	

权限名称	Actions	资源类型
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/autocomplete_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

权限名称	Actions	资源类型
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

权限名称	Actions	资源类型
	<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 	

权限名称	Actions	资源类型
	<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 	

权限名称	Actions	资源类型
create_content_views	<ul style="list-style-type: none"> katello/api/v2/content_views/create katello/api/v2/content_views/copy 	Katello::ContentView
edit_content_views	<ul style="list-style-type: none"> katello/api/v2/content_views/update katello/api/v2/content_view_filters/create katello/api/v2/content_view_filters/update katello/api/v2/content_view_filters/destroy katello/api/v2/content_view_filter_rules/create katello/api/v2/content_view_filter_rules/update katello/api/v2/content_view_filter_rules/destroy katello/api/v2/content_view_puppet_modules/create katello/api/v2/content_view_puppet_modules/update katello/api/v2/content_view_puppet_modules/destroy 	Katello::ContentView
destroy_content_views	<ul style="list-style-type: none"> katello/api/v2/content_views/destroy katello/api/v2/content_views/remove katello/api/v2/content_view_versions/destroy 	Katello::ContentView

权限名称	Actions	资源类型
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 	主机

权限名称	Actions	资源类型
submit_discovered_hosts	<ul style="list-style-type: none">● <code>api/v2/discovered_hosts/facts</code>● <code>api/v2/discovered_hosts/create</code>	主机
auto_provision_discovered_hosts	<ul style="list-style-type: none">● <code>discovered_hosts/auto_provision</code>● <code>discovered_hosts/auto_provision_all</code>● <code>api/v2/discovered_hosts/auto_provision</code>● <code>api/v2/discovered_hosts/auto_provision_all</code>	主机
provision_discovered_hosts	<ul style="list-style-type: none">● <code>discovered_hosts/edit</code>● <code>discovered_hosts/update</code>● <code>api/v2/discovered_hosts/update</code>	主机

权限名称	Actions	资源类型
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 	主机
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 	主机
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

权限名称	Actions	资源类型
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

权限名称	Actions	资源类型
view_domains	<ul style="list-style-type: none"> domains/index domains/show domains/auto_complete_search api/v2/domains/index api/v2/domains/show api/v2/parameters/index api/v2/parameters/show 	
create_domains	<ul style="list-style-type: none"> domains/new domains/create api/v2/domains/create 	
edit_domains	<ul style="list-style-type: none"> domains/edit domains/update api/v2/domains/update api/v2/parameters/create api/v2/parameters/update api/v2/parameters/destroy api/v2/parameters/reset 	
destroy_domains	<ul style="list-style-type: none"> domains/destroy api/v2/domains/destroy 	

权限名称	Actions	资源类型
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 	

权限名称	Actions	资源类型
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 	

权限名称	Actions	资源类型
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 	

权限名称	Actions	资源类型
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	

权限名称	Actions	资源类型
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	

权限名称	Actions	资源类型
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 	

权限名称	Actions	资源类型
create_filters	<ul style="list-style-type: none"> ● 过滤器/新 ● 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 	

权限名称	Actions	资源类型
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

权限名称	Actions	资源类型
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

权限名称	Actions	资源类型
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_hostgroups	<ul style="list-style-type: none">● hostgroups/openscap_proxy_changed	主机

权限名称	Actions	资源类型
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

权限名称	Actions	资源类型
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 	

权限名称	Actions	资源类型
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

权限名称	Actions	资源类型
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

权限名称	Actions	资源类型
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 	

权限名称	Actions	资源类型
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

权限名称	Actions	资源类型
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 	

权限名称	Actions	资源类型
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	

权限名称	Actions	资源类型
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 	

权限名称	Actions	资源类型
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 	

权限名称	Actions	资源类型
destroy_hostgroups	<ul style="list-style-type: none"> ● hostgroups/destroy 	
	<ul style="list-style-type: none"> ● api/v2/hostgroups/destroy 	
view_hosts	<ul style="list-style-type: none"> ● hosts/index ● 主机/显示 ● 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 ● api/v2/hosts/status/configuration 	

权限名称	Actions	资源类型
	<ul style="list-style-type: none"> ● 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/applicable_errata ● katello/api/v2/hosts/installable_errata ● katello/api/v2/hosts/bulk/available_incremental_updates ● katello/api/v2/host_packages/index 	
create_hosts	<ul style="list-style-type: none"> ● hosts/new ● hosts/create ● hosts/clone 	

权限名称	Actions	资源类型
	<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		

权限名称	Actions	资源类型
	<ul style="list-style-type: none"> ● hosts/openscap_proxy_changed ● hosts/edit ● hosts/update ● 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 	

权限名称	Actions	资源类型
	<ul style="list-style-type: none"> ● hosts/select_multiple_puppet_proxy ● hosts/update_multiple_puppet_proxy ● hosts/select_multiple_puppet_ca_proxy ● hosts/update_multiple_puppet_ca_proxy ● hosts/select_multiple_location ● hosts/update_multiple_location ● hosts/architecture_selected ● hosts/compute_resource_selected ● hosts/domain_selected ● hosts/environment_selected ● 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 	

权限名称	Actions	资源类型
	<ul style="list-style-type: none"> ● puppetclasses/parameters ● subnets/freeip ● 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/add_host_collections ● katello/api/v2/hosts/bulk/remove_host_collections ● katello/api/v2/hosts/bulk/install_content 	

权限名称	Actions	资源类型
	<ul style="list-style-type: none"> ● katello/api/v2/hosts/bulk/update_content ● katello/api/v2/hosts/bulk/remove_content ● katello/api/v2/hosts/bulk/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/destroy 	
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 	

权限名称	Actions	资源类型
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 	

权限名称	Actions	资源类型
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

权限名称	Actions	资源类型
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"> ● 位置/新 ● 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 	

权限名称	Actions	资源类型
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"> ● 媒体/新 ● 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 	

权限名称	Actions	资源类型
view_models	<ul style="list-style-type: none">● models/index● models/show● models/auto_complete_search● api/v2/models/index● api/v2/models/show	
create_models	<ul style="list-style-type: none">● models/new● models/create● api/v2/models/create	
edit_models	<ul style="list-style-type: none">● models/edit● models/update● api/v2/models/update	
destroy_models	<ul style="list-style-type: none">● models/destroy● api/v2/models/destroy	

权限名称	Actions	资源类型
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	

权限名称	Actions	资源类型
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	

权限名称	Actions	资源类型
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"> ● 机构/新功能 ● organizations/create ● organizations/clone_taxonomy ● organizations/step2 ● organizations/nest ● api/v2/organizations/create ● katello/api/v2/organizations/create 	

权限名称	Actions	资源类型
edit_organizations	<ul style="list-style-type: none"> ● organizations/edit ● organizations/update ● organizations/import_mismatches ● 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 	

权限名称	Actions	资源类型
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

权限名称	Actions	资源类型
	<ul style="list-style-type: none"> ● katello/api/v2/packages/index ● katello/api/v2/packages/show ● katello/api/v2/distributions/index ● katello/api/v2/distributions/show ● katello/api/v2/package_groups/index ● katello/api/v2/package_groups/show ● katello/api/v2/errata/index ● katello/api/v2/errata/show ● katello/api/v2/puppet_modules/index ● katello/api/v2/puppet_modules/show ● katello/errata/short_details ● katello/errata/auto_complete ● katello/packages/details ● katello/packages/auto_complete ● katello/puppet_modules/show ● katello/repositories/auto_complete_library ● katello/repositories/repository_types ● katello/content_search/index ● katello/content_search/products ● katello/content_search/repos ● katello/content_search/packages ● katello/content_search/errata 	

权限名称	Actions	资源类型
	<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

权限名称	Actions	资源类型
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

权限名称	Actions	资源类型
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

权限名称	Actions	资源类型
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 	

权限名称	Actions	资源类型
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 	

权限名称	Actions	资源类型
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 	

权限名称	Actions	资源类型
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 	

权限名称	Actions	资源类型
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 	

权限名称	Actions	资源类型
配置	<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 	
logs	<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 	

权限名称	Actions	资源类型
create_roles	<ul style="list-style-type: none">● roles/new● roles/create● roles/clone● api/v2/roles/create	
edit_roles	<ul style="list-style-type: none">● roles/edit● roles/update● api/v2/roles/update	
destroy_roles	<ul style="list-style-type: none">● roles/destroy● api/v2/roles/destroy	
access_settings	<ul style="list-style-type: none">● Home/settings	

权限名称	Actions	资源类型
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 	

权限名称	Actions	资源类型
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 	

权限名称	Actions	资源类型
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	

权限名称	Actions	资源类型
view_subscriptions	<ul style="list-style-type: none"> katello/api/v2/subscriptions/index katello/api/v2/subscriptions/show katello/api/v2/subscriptions/available katello/api/v2/subscriptions/manifest_history katello/api/v2/subscriptions/auto_complete_search katello/api/v2/repository_sets/index katello/api/v2/repository_sets/show katello/api/v2/repository_sets/available_repositories 	机构 (Organization)
attach_subscriptions	<ul style="list-style-type: none"> katello/api/v2/subscriptions/create 	机构 (Organization)
unattach_subscriptions	<ul style="list-style-type: none"> katello/api/v2/subscriptions/destroy 	机构 (Organization)

权限名称	Actions	资源类型
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 	机构 (Organization)
delete_manifest	<ul style="list-style-type: none"> katello/api/v2/subscriptions/delete_manifest 	机构 (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

权限名称	Actions	资源类型
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

权限名称	Actions	资源类型
	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">● 用户/新● 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