

IBM Multi-Cloud Data Encryption  
由 SPx<sup>®</sup> 提供技术支持  
V 2.3

*REST API* 规范



**注意**

在使用此信息及其支持的产品之前，请阅读第 179 页的『[声明](#)』中的信息。

本版本适用于 IBM Multi-Cloud Data Encryption V2.3（产品号 5737-C67）及所有后续发行版和修订版，直到在新版本中另有声明为止。

© Copyright IBM Corporation and others 2017, 2019

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

© **Copyright International Business Machines Corporation .**

# 目录

<b>第 1 章概述.....</b>	<b>1</b>
<b>第 2 章基本需求.....</b>	<b>3</b>
应用层协议.....	3
版本.....	3
支持的方法.....	3
连接安全性.....	3
加密.....	3
用户认证.....	3
跨站点请求伪造 (CSRF) 保护.....	4
有效内容格式.....	4
URL、URL 模式、资源方向和其他主题.....	4
资源方向.....	4
作业处理.....	5
<b>第 3 章基本资源类型.....</b>	<b>7</b>
原语.....	7
集合.....	7
模式.....	7
示例.....	7
<b>第 4 章响应状态码.....</b>	<b>9</b>
<b>第 5 章日志资源.....</b>	<b>11</b>
/logs.....	11
GET.....	11
/logs/events/.....	12
GET.....	12
/logs/events/{sequence_number}.....	14
GET.....	14
/logs/debug/.....	14
GET.....	15
/logs/debug/{log_id}.....	16
GET.....	16
/logs/recipients.....	16
GET.....	17
POST.....	18
/logs/recipients/{id}.....	18
GET.....	18
DELETE.....	19
PATCH.....	20
PUT.....	21
<b>第 6 章作业 .....</b>	<b>23</b>
/jobs.....	23
GET.....	23
/jobs/schedules.....	24
GET.....	24
POST.....	25
/jobs/schedules/{id}.....	30

GET.....	30
PATCH.....	31
PUT.....	32
/jobs/schedulers.....	33
GET.....	33
/jobs/schedulers/{id}.....	36
GET.....	36
PATCH.....	37
PUT.....	38
/jobs/schedulers/{id}/actions.....	39
GET.....	39
POST.....	40
/jobs/history[r].....	41
GET.....	41
/jobs/history/{id}.....	42
GET.....	42

## 第 7 章策略资源..... 45

/pathsets.....	45
GET.....	45
POST.....	46
/pathsets/{id}.....	47
GET.....	47
DELETE.....	48
PATCH.....	49
PUT.....	50
/processes.....	51
GET.....	51
POST.....	53
/processes/{id}.....	54
GET.....	54
DELETE.....	55
PATCH.....	55
PUT.....	56
/selectors.....	57
GET.....	57
POST.....	59
/selectors/{id}.....	60
GET.....	60
DELETE.....	61
PATCH.....	62
PUT.....	63
/keys.....	64
GET.....	64
POST.....	65
/keys/{id}.....	66
GET.....	66
PATCH.....	67
PUT.....	68
/groupscopes.....	69
GET.....	69
/groupscopes/{name}.....	70
GET.....	70
PATCH.....	71
PUT.....	71
/groupsets.....	72
GET.....	72

POST.....	74
/groupsets/{id}.....	74
GET.....	75
DELETE.....	75
/datatypes.....	76
GET.....	76
POST.....	79
/datatypes/{id}.....	80
GET.....	80
DELETE.....	81
PATCH.....	81
PUT.....	82

## 第 8 章受管代理程序.....85

/agents.....	85
GET.....	85
POST.....	90
/agents/{id}.....	91
GET.....	91
DELETE.....	93
PATCH.....	94
/agents/{id}/certificates.....	94
GET.....	95
POST.....	95
/agents/{id}/certificates/{certid}.....	96
GET.....	96
DELETE.....	97
PATCH.....	98
PUT.....	99
/agents/{id}/install_bundle.....	99
GET.....	100
/agents/{id}/install_bundle/authorized_users.....	101
GET.....	101
PUT.....	101
/agents/{id}/policy_snapshots.....	102
GET.....	102
POST.....	103
/agents/{id}/policy_snapshots/{snapshot_id}.....	104
GET.....	104
PUT.....	105
DELETE.....	106
/agents/{id}/policy.....	107
GET.....	107
POST.....	108
/agents/{id}/policy/{binding_id}.....	109
GET.....	109
PUT.....	110
PATCH.....	111
DELETE.....	112
/agents/{id}/policy/{binding_id}/storage.....	113
GET.....	113
PUT.....	114
PATCH.....	115
/agents/{id}/features.....	116
GET.....	116
/agents/{id}/features/{feature}.....	117
GET.....	117

PUT.....	117
PATCH.....	118
/agents/{id}/tools.....	119
GET.....	119
POST.....	120
<b>/agents/{id}/tools/{tool_id}.....</b>	<b>121</b>
GET.....	121
PUT.....	122
PATCH.....	124
DELETE.....	125
<b>/agents/{id}/object_store.....</b>	<b>125</b>
GET.....	125
PUT.....	127
PATCH.....	129
/agents/{id}/object_store/credentials.....	130
/agents/{id}/object_store/buckets.....	131
/agents/{id}/object_store/buckets/<bid>/policy.....	134
/agents/{id}/policy/<pid>.....	135
 <b>第 9 章用户帐户.....</b>	 <b>139</b>
/users.....	139
GET.....	139
POST.....	140
/users/{username}.....	141
GET.....	141
DELETE.....	142
PATCH.....	142
PUT.....	143
/roles.....	144
GET.....	144
/roles/{id}.....	145
GET.....	146
/sessions.....	146
GET.....	147
POST.....	148
/sessions/{id}.....	149
GET.....	149
DELETE.....	150
 <b>第 10 章服务器设置.....</b>	 <b>151</b>
/settings.....	151
GET.....	151
/settings/directories.....	152
GET.....	152
POST.....	153
/settings/directories/{id}.....	154
GET.....	154
DELETE.....	154
PUT.....	155
/settings/keystores.....	156
GET.....	156
POST.....	157
/settings/keystores/{id}.....	159
GET.....	159
PUT.....	160
PATCH.....	161
DELETE.....	162

<b>第 11 章归档并上载文件.....</b>	<b>163</b>
/files.....	163
GET.....	163
/files/upload.....	164
GET.....	164
POST.....	165
/files/upload/{id}.....	165
GET.....	165
DELETE.....	166
<b>第 12 章高级属性.....</b>	<b>167</b>
/properties.....	167
GET.....	167
/properties/{key}.....	168
GET.....	168
PATCH.....	169
<b>第 13 章设置和配置问题.....</b>	<b>171</b>
设置和配置问题.....	171
/issues.....	171
/issues/{id}.....	172
<b>第 14 章全球化资源.....</b>	<b>175</b>
/locales.....	175
GET.....	175
/locales/{id}.....	176
GET.....	176
/strings.....	177
GET.....	177
/strings/{id}.....	178
GET.....	178
<b>声明.....</b>	<b>179</b>
商标.....	180
产品文档的条款和条件.....	181
隐私策略注意事项.....	181





---

# 第 1 章 概述

此规范提出 MDE 产品的 REST API 需求和体系结构。

本文档假定您熟悉一般 REST API 概念，并试图阐明针对 MDE 做出的体系结构选择。



---

## 第 2 章 基本需求

### 应用层协议

---

#### 版本

MDE REST API 将设计为随附于 HTTP 1.1，与业界广泛采用的惯例保持一致。

#### 支持的方法

REST API 将支持下列请求方法：

- GET
- PUT
- PATCH（并非 HTTP 定义的 PATCH，请参阅 <https://tools.ietf.org/html/rfc5789>）
- POST
- DELETE

对于所有其他请求方法，将以 HTTP 405 (Not Implemented) 状态码进行响应。

### 连接安全性

---

#### 加密

REST API 将仅对 TLS 1.2+ 安全连接公开。密码套件、算法和其他参数限于 NIST sp800-131a 和 FIPS 140-2 准则所核准的各个项。

#### 用户认证

REST API 将需要登录。将支持两种认证方法

1. 将允许使用**会话认证**，该认证使用已对 Web UI 公开且在下列使用会话 Cookie 和其他标头（例如 CSRF 令牌）的请求中引用的表单认证方法。这样做主要是为了允许 Web UI 将 REST API 用于其自己的功能，并支持在开发期间进行基于浏览器的测试。没有已知的原因要在生产环境中对非浏览器客户机尝试并禁用此方法，但客户脚本编制不应使用此方法。

启动已认证的会话（登录）是通过向会话资源发送带有有效 CSRF 令牌的 POST 请求来完成的。同样地，结束该会话（注销）是通过向当前已登录会话的特定会话资源发送带有有效 CSRF 令牌的 DELETE 请求来完成的。有关更多信息，请参阅会话资源会话。

##### 1. HTTP 基本认证

正如 IETF RFC 1945 第 11.1 节中详细描述的那样，REST API 部分支持基本访问认证模式，但具有下列附带条件：

- 对未授权请求的响应将不提供认证挑战（WWW-Authenticate 标头）。
- 不在任何时间段内高速缓存凭证
- 不存储或启动 HTTP 会话

也就是说，REST API 将接受所规定格式的 Authorization 标头，这将允许在该单个请求的持续时间内进行已认证的访问。后续请求必须始终提供此标头，而那些只有在首先接收到认证挑战之后才应开始使用此标头的客户机将无法正常运行。

对于已进行基本认证的请求，跨站点请求伪造保护（在下一节中详细描述）将不会处于活动状态。

## 跨站点请求伪造 (CSRF) 保护

必须保护使用会话认证发出的 REST API 请求，使其免受 CSRF 攻击。具体而言，MDE 将实现下列要求，并且这些要求将应用于会话认证的请求：

- PATCH、PUT、POST 和 DELETE 请求被视为受 CSRF 保护的请求，并且必须随附有效的 CSRF 令牌
- 活动会话中将保留两个令牌。可使用任一令牌，但只要可能客户机应发送最新令牌（保留旧令牌只是为了避免在某些争用情况下失败）
- 将在每个受支持请求中以名为 Csrf-Token 的 Cookie 形式向客户机提供当前有效的令牌
- 有效令牌必须在受 CSRF 保护的请求中的 X-Csrf-Token 标头中传递
- 使用给定会话发出的每个受 CSRF 保护的请求将轮换会话 CSRF 令牌，而无论所传递的请求是否有效。将在服务器响应标头中设置最新令牌。
- 传递无效令牌的请求将被拒绝，在这种情况下，HTTP 状态码为 400，而且 Warning 标头中填充的文本会指出原因为 CSRF 令牌失败。
- CSRF 令牌是 256 位 Base64 编码值，由最安全的可用随机数字生成器随机生成。

## 有效内容格式

除非本规范中已定义的子资源另有指示，否则将使用 application/json 内容类型来表示所有资源。对于那些支持或其他内容类型的资源，REST API 将会期望设置 HTTP Accept 标头。任何其他请求的内容类型将以空响应主体和 415 Unsupported Media Type 状态码进行响应

## URL、URL 模式、资源方向和其他主题

### 资源方向

REST API 将高度面向资源。除非另有指示，否则将遵循下列原则：

- HTTP 方法 GET、PUT、POST 和 DELETE 与读取、更新、创建和删除资源相关
- 向同一 URL 发出的 GET 和 PUT 请求的主体在语法和结构上应完全相同，PUT 请求中请求的任何字段更新除外
- POST 请求经常仅对数组资源有效，并且这种请求的主体应该与数组资源的其他元素属于相同的类型（语言定义的类型或 API 定义的类型），除非特定 API 另有定义
- 产品配置中的所有更改都应遵循“GET-更新-PUT”模式，这意味着客户机应该能够对具有资源/设置的 URL 执行 GET 请求，更新 GET 响应文本中的所需值，然后使用更新后的响应文本作为向原始 URL 发出的相应 PUT 请求的主体。
- URL 与所返回对象内的层次结构应相互相符，这意味着：
  - 通过向 URL 追加斜杠 (/) 和字段名称，应该可通过该 URL 直接访问对象字段。  
例如，如果 /rest/resource 返回 {"id": 1, "name": "myResourceNumber1"}，那么 /rest/resource/name 应返回 "myResourceNumber1"
  - 通过向 URL 追加斜杠 (/) 以及所需元素的 id 字段的值，应该可通过该 URL 直接访问数组元素。  
例如，如果 /rest/resources 返回 [{"id": 1}, {"id": 2}]，那么 /rest/resources/1 应返回 {"id": 1}。  
**注：**id 字段已定义，不可将其命名为“id”
- 在 URL 中，数组资源应使用复数命名。  
例如，返回 users 的数组资源可响应 /rest/user
- 应避免 URL 分类，以防止模块实现泄漏到 REST API 资源层次结构中

例如, `/rest/users` 优于 `/rest/user/list`, 即使在 `/rest/user` 下面可能找到其他用户相关资源也是如此

## 作业处理

REST API 请求可长时间运行, 也可短时间运行。对 REST API 发出的任何可能长时间运行的 PUT、POST 或 DELETE 请求可选择性地衍生一个服务器端任务/线程, 然后按如下所示响应:

- Location 标头设置为将接受 GET 请求的 URL, 以用于状态检查
- 状态码为 202 (Accepted)
- 响应主体为可选项

Location 标头所指示的 URL 处定义的作业资源将不接受 PUT 或 POST 请求。DELETE 请求将解释为取消作业或删除运行后作业记录, 具体取决于作业状态。相对于应用程序而言, 所有作业资源都是易变的, 并且在应用程序重新启动之后将会丢失。



---

## 第 3 章 基本资源类型

### 原语

---

基本类型集合将与 JSON 语言规范中提供的基本类型集合一致。

### 集合

---

#### 模式

集合是其他类型的列表、集和数组。集合中的项将在响应中表示为数组，其排序（列表）或唯一性（集）约束由 REST 服务器强制实施，而不是通过语法显式指示。

```
{
  "type": "array",
  "minItems": 0,
  "items": { ... }
  "uniqueItems": true
}
```

#### 示例

集合将按如下所示出现在数组结构中：

```
[
  {
    "type": "user",
    "id": "4"
  },
  {
    "type": "user",
    "id": "5"
  },
  {
    "type": "user",
    "id": "8"
  }
]
```





## 第 4 章 响应状态码

代码	消息	描述
200	OK	请求已成功。所报告的内容将出现在响应主体中。
201	Created	已创建新的资源。 <b>Location</b> 标头将设置为所创建资源的 URL。
202	Accepted	如果以同步方式处理响应，响应将会不可接受地延迟。已创建作业来处理请求。 <b>Location</b> 标头将设置为该作业的 URL。
204	No Content	请求已成功，但响应中不需要任何内容。经常对 DELETE 请求返回此状态码，但此状态码也可用于退化的 PUT 和 PATCH 请求。
205	Reset Content	请求已成功。由于属性更改，应重置目标资源的客户端模型。经常对 PUT 和 PATCH 请求返回此状态码。
400	Bad Request	由于请求中存在错误而未能对其进行处理。
401	Unauthorized	请求未经过认证。
403	Forbidden	请求已经过认证，但不允许用户执行所请求的操作。
404	Not Found	所请求的资源不存在，或者服务器未保留任何有关该资源曾经存在的记录。
405	Method Not Allowed	该 HTTP 方法对于目标资源而言无效。
406	Not Acceptable	<b>Accept</b> 标头中请求的内容类型不受支持。目前，application/json 是唯一接受的类型。
409	Conflict	随请求方法不同而异： <ul style="list-style-type: none"><li>· POST - 具有相同身份的资源已存在。已锁定父资源以进行更改。</li><li>· PATCH/PUT - 正在更改身份字段，并且新身份字段与另一个已使用的身份相同。正在更新的资源已被另一个请求以使此请求失效的方式更新。已锁定资源以进行更改。</li><li>· DELETE - 已锁定资源以进行更改。</li></ul>

410	Gone	具有所请求身份的资源曾经存在，但已被删除。
424	Failed Dependency	所请求的更改取决于另一资源状态，但目前未满足该状态。
428	Precondition Required	要在发送一个或多个其他请求并且另一个相关资源的状态已更改之后，才能完成该请求。
500	Internal Server Error	由于内部服务器错误而回退，没有更具体的故障代码。 <b>Warning</b> 标头将包含在归档错误报告时应报告的信息。
503	Service Unavailable	执行该请求所需的内部服务不可用。 <b>Warning</b> 标头将包含在归档错误报告时应报告的信息。

# 第 5 章 日志资源

## /logs

所有已知日志的集合

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
200	响应主体将包含对子日志的引用的集合。这些引用将不包含任何日志条目本身。
401	当前请求未经过认证。
403	未向当前用户进行授权。
406	<b>Accept</b> 标头必须设置为受支持的内容类型。
500	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
503	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

```
{
  "type": "array",
  "minItems": 2,
  "uniqueItems": true,
  "items": [
    {
      "type": "object",
      "required": ["type","id", "href" ],
      "additionalProperties": true,
      "properties": {
        "type": { "enum": [ "ref" ] },
        "id": { "enum": [ "events", "debug", "recipients" ] },
        "href": "string"
      }
    }
  ]
}
```

#### 示例

```
Request: GET /logs

Response: 200 OK

[
  {
    "type": "ref",
    "id": "event",
```

```

    "href": "/rest/logs/events",
  },
  {
    "type": "ref",
    "id": "debug",
    "href": "/rest/logs/debug",
  },
  {
    "type": "ref",
    "id": "recipients",
    "href": "/rest/logs/recipients",
  }
]

```

## /logs/events/

系统事件和审计日志

### GET

#### 查询参数

参数	描述
<b>event_type</b>	设置为事件对象类型字段可具有的其中一个值，以便仅显示与该类型匹配的日志条目。无法识别的值将被忽略。示例： <b>event_type=audit&amp;event_type=system</b>
<b>order</b>	在设置为“descending”时将最新事件置于最前面的顺序值。示例： <b>order=descending</b>

#### 支持的内容类型

内容类型	描述
<b>application/json</b>	下面的模式所描述的对象
<b>text/csv</b>	下面所描述的日志资源的逗号分隔值 (CSV) 表示

#### 状态码

代码	描述
<b>200</b>	响应主体将包含集合形式的系统事件和审计日志条目。如果应用了查询过滤器，但没有任何日志匹配，那么将发送空集合。
<b>400</b>	当前请求的格式不正确。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

```
{
  "type": "array",
  "minItems": 0,
  "uniqueItems": true,
  "items": [
    {
      "type": "object",
      "required": [ "type", "sequence_number", "event_id", "source",
        "event_type", "severity", "timestamp", "message_args", "redacted" ],
      "additionalProperties": false,
      "properties": {
        "type": { "enum": [ "event" ] },
        "sequence_number": {
          "type": "number",
          "multipleOf": 1,
          "minimum": 1
        },
        "event_id": "string",
        "source": "string",
        "event_type": { "enum": [ "AUDIT", "SYSTEM" ] },
        "severity": { "enum": [ "DEBUG", "INFO",
          "WARN", "ERROR", "CRITICAL" ] },
        "timestamp": {
          "type": "number",
          "multipleOf": 1,
          "minimum": 0
        },
        "message": "string",
        "message_args": {
          "type": [ "array" ],
          "minItems": 0,
          "uniqueItems": false,
          "items": [ { "type": [ "number", "string",
            "boolean", "null" ] } ]
        },
        "redacted": "boolean"
      }
    }
  ]
}
```

## 示例

Request: GET /rest/logs/events?event\_type=audit&event\_type=system

Response: 200 OK

```
[
  {
    "type": "event",
    "sequence_number": 12,
    "event_id": "0X00020003",
    "source": "localhost",
    "event_type": "AUDIT",
    "severity": "INFO",
    "timestamp": 1442947787281,
    "message_args": [ "admin", "policy255" ],
    "redacted": false
  },
  {
    "type": "event",
    "sequence_number": 14,
    "event_id": "0x00220334",
    "source": "192.168.4.234",
    "event_type": "SYSTEM",
    "severity": "CRITICAL",
    "timestamp": 1442947787282,
    "severity": "CRITICAL",
    "message": "Policy %s could not be applied: RC %s",
    "message_args": [ "policy255", 23 ],
    "redacted": false
  }
]
```

## [/logs/events/{sequence\\_number}](#)

单个事件

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含与 URL 的 {sequence_number} 部分相匹配的事件对象。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {sequence_number} 的事件，且该事件之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>410</b>	找不到具有指定 {sequence_number} 的事件，但该事件之前曾存在
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

请参阅事件日志集合模式

#### 示例

```
Request: GET /rest/logs/events/12
Response: 200 OK
{
  "type": "event",
  "sequence_number": 12,
  "event_id": "0X00020003",
  "source": "localhost",
  "event_type": "AUDIT",
  "severity": "INFO",
  "timestamp": 1442947787281,
  "message_args": [ "admin", "policy255" ],
  "redacted": false
}
```

## [/logs/debug/](#)

系统调试日志

# GET

## 查询参数

参数	描述
<无>	

## 状态码

代码	描述
<b>200</b>	响应主体将包含对子日志的引用的集合。这些引用将不包含任何日志条目本身。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

```
{
  "type": "array",
  "minItems": 0,
  "uniqueItems": true,
  "items": [
    {
      "type": "object",
      "required": ["type", "id", "href"],
      "additionalProperties": true,
      "properties": {
        "type": { "enum": [ "ref" ] },
        "id": "string",
        "href": "string"
      }
    }
  ]
}
```

## 示例

```
Request: GET /rest/logs/debug
Response: 200 OK

[
  {
    "type": "ref",
    "id": "bundleAll.log",
    "href": "/rest/logs/debug/bundleAll.log",
  },
  {
    "type": "ref",
    "id": "bundleWarnPlus.log",
    "href": "/rest/logs/debug/bundleWarnPlus.log",
  }
]
```

## /logs/debug/{log\_id}

单个调试日志

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含字符串数组形式的所指定调试日志的内容，文件中的每一行都有一个相应的字符串。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {log_id} 的日志
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

```
{
  "type": "array",
  "minItems": 0,
  "uniqueItems": false,
  "items": [
    {
      "type": "string"
    }
  ]
}
```

#### 示例

Request: GET /rest/logs/debug/bundleAll.log

Response: 200 OK

```
[
  "r22 Sep 2015 13:09:38 [INFO ] c.s.a.f.l.LogConfigurator(LogConfigurator.java:42)
  Logs Configured",
  "22 Sep 2015 13:09:38 [INFO ] c.s.a.f.b.BundleValidator(BundleValidator.java:37)
  !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! ",
  "22 Sep 2015 13:09:38 [INFO ] c.s.a.f.l.Activator(Activator.java:35)
  Starting Loggerref"
]
```

## /logs/recipients

访问事件日志接收方



# GET

## 查询参数

参数	描述
<无>	

## 状态码

代码	描述
<b>200</b>	响应主体将包含所有已配置的事件日志接收方
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

```
{
  "type": "array",
  "minItems": 0,
  "uniqueItems": true,
  "items": [
    {
      "type": "object",
      "required": [ "type", "id", "transport", "user",
"password", "host", "port", "email", "security", "actions", "format" ],
      "additionalProperties": false,
      "properties": {
        "type": { "enum": [ "event-recipient" ] },
        "id": "number",
        "transport": { "enum": ["syslog"] },
        "user": "string",
        "password": "string",
        "host": "string",
        "port": "number",
        "email": "string",
        "format": { "enum": [ "LEEF", "CADF-CSV", "CADF-JSON", "CEF" ] },
        "security": { "enum": [ "none", "SSL", "TLS" ] },
      }
    }
  ]
}
```

## 示例

Request: GET /rest/logs/recipients

Response: 200 OK

```
[
  {
    "type": "event-recpient",
    "id": 4,
    "transport": "syslog",
    "host": "sfc.ibm.com",
    "port": 5687,
    "security": "none"
  }
]
```

## POST

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>201</b>	已创建接收方。 <b>Location</b> 标头将提供新接收方的 URL。
<b>400</b>	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户创建策略。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>409</b>	新接收方与现有接收方电子邮件或标识相冲突。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅事件接收方集合模式， *event-recipient* 对象。 **更改：** *id* 不是必需字段， 也不应设置这些字段。

### 示例

```
Request: POST /rest/logs/recipients

{
  "type": "event-recipient",
  "transport": "syslog",
  "host": "sfc.ibm.com",
  "port": 5687,
  "format": " LEEF",
  "security": "none"
}

Response: 201 Created | Location: /logs/recipients/4
```

## [/logs/recipients/{id}](#)

特定事件接收方

## GET

### 查询参数

参数	描述
<无>	

## 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的事件接收方
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 的事件接收方，且该接收方之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>410</b>	找不到具有指定 {id} 的事件接收方，但该接收方之前曾存在
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅事件接收方集合模式，*event-recipient* 对象

## 示例

```
Request: GET /rest/logs/recipients/4
```

```
Response: 200 OK
```

```
{
  "type": "event-recipient",
  "id": 4,
  "transport": "syslog",
  "host": "sfc.ibm.com",
  "port": 5687,
  "format": "LEEF",
  "security": "none"
}
```

## DELETE

### 查询参数

参数	描述
<无>	

## 状态码

代码	描述
<b>202</b>	已创建作业来处理删除。请查看 <b>Location</b> 标头以获取作业 URL。
<b>204</b>	删除已成功
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 的事件接收方，且该接收方之前未存在过

<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

*DELETE* 中不发送主体

## 示例

```
Request: DELETE /rest/logs/recipients/4
Response: 204 No content
```

## PATCH

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>204</b>	请求已成功，但未包含任何更改。不需要刷新客户端表示。
<b>205</b>	请求已成功，且资源已更新。
<b>400</b>	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户创建事件接收方。
<b>404</b>	找不到具有指定 {id} 的事件接收方，且该接收方之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>409</b>	对接收方所做的更改与另一个接收方相冲突。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅事件接收方集合模式，*event-recipient* 对象。 **更改：** 事件接收方对象仅需要 *PATCH* 中的 *type* 和 *id* 字段，所有其他字段都是可选的。

## 示例

```
Request: PATCH /rest/logs/recipients/4
{
  "id": 4,
```

```
    "email": "noreply-2@example.com"
  }

Response: 205 Reset content
```

## PUT

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>204</b>	请求已成功，但未包含任何更改。不需要刷新客户端表示。
<b>205</b>	请求已成功，且资源已更新。
<b>400</b>	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户创建事件接收方。
<b>404</b>	找不到具有指定 {id} 的事件接收方，且该接收方之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>409</b>	对接收方所做的更改与另一个接收方相冲突。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅事件接收方集合模式，*event-recipient* 对象。

### 示例

```
Request: PUT /rest/logs/recipients/4
```

```
{
  "type": "event-recipient",
  "id": 4,
  "transport": "email",
  "user": "admin",
  "password": "",
  "host": "",
  "port": null,
  "email": "noreply-3@example.com",
  "locale": "de",
  "security": "none"
}
```

```
Response: 205 Reset content
```



# 第 6 章 作业

## /jobs

作业子资源的列表。

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含对子资源的引用的集合。这些引用将不包含任何作业调度或作业条目本身。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

```
{
  "type": "array",
  "minItems": 3,
  "uniqueItems": true,
  "items": [
    {
      "type": "object",
      "required": ["type", "id", "href"],
      "additionalProperties": true,
      "properties": {
        "type": { "enum": [ "ref" ] },
        "id": { "enum": [ "schedules", "schedulers", "history" ] },
        "href": "string"
      }
    }
  ]
}
```

#### 示例

```
Request: GET /jobs

Response: 200 OK

[
  {
    "type": "ref",
    "id": "schedules",
```

```

    "href": "/jobs/schedules",
  },
  {
    "type": "ref",
    "id": "schedulers",
    "href": "/jobs/schedulers",
  },
  {
    "type": "ref",
    "id": "history",
    "href": "/jobs/history",
  }
]

```

## /jobs/schedules

系统中的所有作业调度。

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含所有已知作业调度。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户查看作业。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

```

{
  "type": "array",
  "minItems": 0,
  "uniqueItems": true,
  "items": [
    {
      "type": "object",
      "required": [ "type", "id", "job_id", "when_starts", "when_ends", "starts_unit",
        "ends_unit", "frequency", "multiplier", "state", "job_type",
        "job_notes", "job_properties" ],
      "additionalProperties": false,
      "properties": {
        "type": { "enum": [ "job-schedule" ] },
        "id": "number",
        "job_id": "number",
        "when_starts": "number",
        "when_ends": "number",
        "starts_unit": { "enum": [ "ABS_MS", "REL_MS", "REL_SECONDS", "REL_MINUTES",
          "REL_HOURS", "REL_DAYS", "REL_WEEKS", "REL_MONTHS", "REL_YEARS" ] },
        "ends_unit": { "enum": [ "ABS_MS", "REL_MS", "REL_SECONDS", "REL_MINUTES",
          "REL_HOURS", "REL_DAYS", "REL_WEEKS", "REL_MONTHS", "REL_YEARS" ] },
        "frequency": { "enum": [ "ONCE", "MINUTELY", "HOURLY", "DAILY", "WEEKLY",
          "MONTHLY", "YEARLY" ] },
        "multiplier": "number",

```



```

    "state": { "enum": [ "NOT_APPROVED", "WAITING", "RUNNING", "PAUSED",
                        "CANCELLED", "FINISHED" ] },
    "job_type": "string",
    "job_notes": "string",
    "job_properties": {
      "type": "object",
      "required": [],
      "additionalProperties": true,
      "properties": {}
    }
  }
}
]
}
```

示例

```
Request: GET /rest/jobs/schedules

Response: 200 OK

[
  {
    "id": 1,
    "job_id": 1,
    "when_starts": 0,
    "when_ends": 0,
    "starts_unit": "ABS_MS",
    "ends_unit": "ABS_MS",
    "frequency": "ONCE",
    "multiplier": 1,
    "state": "NOT_APPROVED",
    "job_properties": {
      "c.s.a.b.u.j.createUser": "secadm",
      "runnable.passwordSalt64": "oPgSfm+htrj13c73xgytESU2hiWJVfqkQB8CMcu7l+4\u003d",
      "job-submitter": "admin",
      "runnable.passwordDigest64": "tPV9p7WDjpkNv7czq1KUh+BCC3LcySTsUnZcNl0UYYQ\u003d",
      "c.s.a.b.u.j.createNotes": ""
    },
    "job_type": "c.s.a.b.u.j.CreateJob",
    "job_notes": "Creating a security administrator",
    "type": "job-schedule"
  }
]
```

POST

查询参数

参数	描述
<无>	

状态码

代码	描述
201	已创建作业调度。 <b>Location</b> 标头将提供新创建的调度程序作业的 URL，该作业一旦获核准就会激活实际作业调度。
400	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
401	当前请求未经过认证。
403	未授权当前用户为此作业调度创建所指示的作业类型。
406	<b>Accept</b> 标头必须设置为受支持的内容类型。

<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

```
{
  "type": "object",
  "required": [ "type", "when_starts", "when_ends", "frequency", "multiplier",
    "job_type", "job_notes", "job_properties" ],
  "additionalProperties": false,
  "properties": {
    "type": { "enum": [ "job-schedule" ] },
    "id": "number",
    "job_id": "number",
    "when_starts": "number",
    "when_ends": "number",
    "starts_unit": { "enum": [ "ABS_MS", "REL_MS", "REL_SECONDS", "REL_MINUTES",
      "REL_HOURS", "REL_DAYS", "REL_WEEKS", "REL_MONTHS", "REL_YEARS" ] },
    "ends_unit": { "enum": [ "ABS_MS", "REL_MS", "REL_SECONDS", "REL_MINUTES",
      "REL_HOURS", "REL_DAYS", "REL_WEEKS", "REL_MONTHS", "REL_YEARS" ] },
    "frequency": { "enum": [ "ONCE", "MINUTELY", "HOURLY", "DAILY", "WEEKLY",
      "MONTHLY", "YEARLY" ] },
    "multiplier": "number",
    "state": { "enum": [ "NOT_APPROVED", "WAITING", "RUNNING", "PAUSED",
      "CANCELLED", "FINISHED" ] },
    "job_type": "string",
    "job_notes": "string",
    "job_properties": {
      "type": "object",
      "required": [],
      "additionalProperties": true,
      "properties": {}
    }
  }
}
```

## 作业调度字段描述

字段	类型	允许值	描述
<b>id</b>	Integer	自动生成	这是调度的标识。
<b>job_id</b>	Integer	由系统设置	这是与该调度相关联的调度程序作业的标识。
<b>when_starts</b>	Long	0 到最大长整型值 0 表示获核准后立即开始。	绝对戳记毫秒数，或者相对于调度获核准时间的的时间单位数。 此值确定调度开始执行的时间。
<b>when_ends</b>	Long	0 到最大长整型值 0 表示永远运行（仅适用于定期调度）。	绝对戳记毫秒数，或者相对于调度开始执行的时间的时间单位数。 此值确定调度将执行多长时间。

<b>starts_unit</b>	Enum	<p>ABS_MS = 绝对戳记毫秒数</p> <p>REL_MS = 相对（从核准时间开始的）毫秒数</p> <p>REL_SECONDS = 相对（从核准时间开始的）秒数</p> <p>REL_MINUTES = 相对（从核准时间开始的）分钟数</p> <p>REL_HOURS = 相对（从核准时间开始的）小时数</p> <p>REL_DAYS = 相对（从核准时间开始的）天数</p> <p>REL_WEEKS = 相对（从核准时间开始的）周数</p> <p>REL_MONTHS = 相对（从核准时间开始的）月份数</p> <p>REL_YEARS = 相对（从核准时间开始的）年数</p>	<p>确定系统如何解释 when_starts 字段中的输入值。如果设置为 ABS_MS，那么 when_starts 具有戳记毫秒数；否则，when_starts 具有相对于核准时间的时间单位数。</p>
<b>ends_unit</b>	Enum	<p>ABS_MS = 绝对戳记毫秒数</p> <p>REL_MS = 相对（从调度开始时间开始的）毫秒数</p> <p>REL_SECONDS = 相对（从调度开始时间开始的）秒数</p> <p>REL_MINUTES = 相对（从调度开始时间开始的）分钟数</p> <p>REL_HOURS = 相对（从调度开始时间开始的）小时数</p> <p>REL_DAYS = 相对（从调度开始时间开始的）天数</p> <p>REL_WEEKS = 相对（从调度开始时间开始的）周数</p> <p>REL_MONTHS = 相对（从调度开始时间开始的）月份数</p> <p>REL_YEARS = 相对（从调度开始时间开始的）年数</p>	<p>确定系统如何解释 when_ends 字段中的输入值。如果设置为 ABS_MS，那么 when_ends 具有戳记毫秒数；否则，when_ends 具有相对于开始时间的相应时间单位数。换言之，这是调度的长度。</p>

<b>frequency</b>	Enum	<p>ONCE = 执行一次/不重复</p> <p>MINUTELY = 每 M 分钟重复一次</p> <p>HOURLY = 每 M 个小时重复一次</p> <p>DAILY = 每 M 天重复一次</p> <p>WEEKLY = 每 M 周重复一次</p> <p>MONTHLY = 每 M 个月重复一次</p> <p>YEARLY = 每 M 年重复一次</p> <p>注: M 是下面的 multiplier 字段。</p>	<p>确定定期调度频率。如果设置为 ONCE, 那么该调度将不重复 (非定期), 并且 when_ends 参数将被忽略。</p> <p>否则, 该值与 multiplier 字段一起将定义定期作业的频率, 并且 when_ends 参数确定调度将持续的时长。</p>
<b>multiplier</b>	Integer	0 到最大整型值	与 frequency 一起确定在定期执行之前要等待的时间间隔。例如, 如果 frequency 为 HOURLY, 并且 multiplier 为 3, 那么作业将每 3 个小时执行一次, 等等。
<b>state</b>	Enum	<p>NOT_APPROVED</p> <p>该调度未获核准, 未在运行, 并且将不执行作业。这是核准之前的初始状态。</p> <p>WAITING</p> <p>该调度正在运行, 作业已获核准, 但尚未首次执行任何动作。</p> <p>RUNNING</p> <p>该调度已执行一个或多个作业, 并且已调度其他执行。</p> <p>PAUSED</p> <p>该调度已临时暂停, 在它处于此状态期间, 未执行任何作业。</p> <p>CANCELLED</p> <p>该调度已被显式取消。不会再进行任何作业执行。</p> <p>FINISHED</p> <p>该调度已正常完成执行。不会再进行任何作业执行。</p>	这是该调度的状态。由系统进行管理。

<b>job_type</b>	String	请参阅关于作业类型的章节。	这是要提交的作业的类型。此字段确定在由调度程序触发时，将要执行的作业可运行项类型（实际工作负载）。
<b>job_notes</b>	String	任何字符串值。	这是缺省情况下要向作业添加的备注/注释。
<b>job_properties</b>	Object	JSON 对象。	这是用于定义要按调度运行的作业的作业属性对象。此处的值随作业类型不同而异。请参阅下一节以了解详细信息。

## 作业属性模式

```
{
  "type": "object",
  "required": [ "type", "job_type", "keys_to_update", "public_key" ],
  "additionalProperties": false,
  "properties": {
    "type": { "enum": [ "key-update-job" ] },
    "job_type": { "enum": [ "key-rotation", "key-revocation", "key-shred" ] },
    "keys_to_update": {
      "type": [ "array" ],
      "minItems": 0,
      "uniqueItems": true,
      "items": [
        {
          "type": "object",
          "required": [ "type", "id" ],
          "additionalProperties": false,
          "properties": {
            "id": "number"
          }
        }
      ]
    },
    "public_key": {
      "type": "object",
      "required": [ "name", "id", "type", "href" ],
      "additionalProperties": false,
      "properties": {
        "type": { "enum": [ "ref" ] },
        "name": "string",
        "id": "number",
        "href": "string"
      }
    }
  }
}
```

## 示例

Request: POST /rest/jobs/schedules

```
{
  "type": "job-schedule",
  "when_starts": 2,
  "when_ends": 0,
  "starts_unit": "REL_WEEKS",
  "frequency": "WEEKLY",
  "multiplier": 1,
  "job_type": "key-update-job",
  "job_notes": "",
  "job_properties": {
    "type": "key-update-job",
    "job_type": "key-rotation",
    "keys_to_update": [ {
      "id": 3
    } ],
    "id": 2
  }
}
```

```

    },
    {
      "id": 1
    }
  ],
  "notes": "",
  "public_key": {
    "name": "server1.cert.pem",
    "id": 1,
    "type": "client-file"
  }
}

```

Response: 201 Created | Location: /rest/jobs/schedulers/2

## /jobs/schedules/{id}

特定作业调度。

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的作业调度。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户查看与此作业调度相关联的作业或此作业类型。
<b>404</b>	找不到具有指定 {id} 的作业调度，且该作业调度之前未存在过。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

请参阅第 24 页的『模式』一节，数组内的对象类型 *job-schedule*。

#### 示例

来自上一个 POST 节的调度。请注意，在本示例中，该调度尚未获核准。另外，还要注意 *job\_properties* 对象将从外部对象格式更改为内部表示。

Request: GET /rest/jobs/schedules/2

Response: 200 OK

```

{
  "id": 2,
  "job_id": 2,
  "when_starts": 2,
  "when_ends": 0,

```

```

"starts_unit": "REL_WEEKS",
"ends_unit": "ABS_MS",
"frequency": "WEEKLY",
"multiplier": 1,
"state": "NOT_APPROVED",
"job_properties": {
  "com.securityfirstcorp.atlantis.bundles.keygen.keys.agentIndex": "",
  "com.securityfirstcorp.atlantis.bundles.keygen.keys.keyUpdateIndexes": "1 2 3",
  "runnable.updateJobType": "ROTATE",
  "runnable.keyIndexes2": "1",
  "runnable.fileBytes64":
    "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSU0tLS0tCk1JSUdIakNDQkFhZ0F3SUJBZ01DRUFFd0RRWUpLb1pJaH
... omitted ...
DNVYWPcC1VJbHN0a0Uwbk5MTjIyK2JoRGt0SHJNNmZSU1ZESEd3PT0
KLS0tLS1FTkQgQ0VSVElGSUNBVEUtLS0tLQo\u003d",
  "runnable.keyIndexes0": "3",
  "runnable.keyIndexes1": "2",
  "runnable.user": "admin",
  "runnable.notes": "",
  "job-submitter": "admin"
},
"job_type": "c.s.a.b.p.j.KeyRotationJob",
"job_notes": null,
"type": "job-schedule"
}

```

## PATCH

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>204</b>	请求已成功，但未包含任何更改。不需要刷新客户端表示。
<b>201</b>	已创建调度更新作业。 <b>Location</b> 标头将提供新创建的作业的 URL，该作业一旦获核准就会使用新参数刷新实际作业调度。
<b>400</b>	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改与此作业调度相关联的作业或特定作业类型。
<b>404</b>	找不到具有指定 {id} 的作业调度，且该作业调度之前未存在过。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅第 24 页的『模式』一节，数组内的对象类型 *job-schedule*。**更改：**只有 *type* 和 *id* 是必需字段，并且只有 *when\_starts*、*when\_ends*、*starts\_unit* 和 *ends\_unit* 字段可更改。

示例

对活动的作业调度进行修补，方法是覆盖其到过去的时间范围，这将导致它在调度更新作业获核准时立即结束。

```
Request: PATCH /rest/jobs/schedules/2

{
  "type": "job-schedule",
  "id": 2,
  "when_starts": 1,
  "when_ends": 2,
  "starts_unit": "ABS_MS",
  "ends_unit": "ABS_MS"
}

Response: 201 Created | Location: /rest/jobs/schedulers/3
```

PUT

查询参数

参数	描述
<无>	

状态码

代码	描述
204	请求已成功，但未包含任何更改。不需要刷新客户端表示。
201	已创建调度更新作业。 <b>Location</b> 标头将提供新创建的作业的 URL，该作业一旦获核准就会使用新参数刷新作业调度。
400	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
401	当前请求未经过认证。
403	未授权当前用户更改与此作业调度相关联的作业或特定作业类型。
404	找不到具有指定 {id} 的作业调度，且该作业调度之前未存在过
406	<b>Accept</b> 标头必须设置为受支持的内容类型。
500	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
503	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

模式

请参阅第 24 页的『模式』一节，数组内的对象类型 *job-schedule*。 **更改：**只有 *when\_starts*、*when\_ends*、*starts\_unit* 和 *ends\_unit* 字段可更改。

示例

除了需要发送完整调度资源之外，与上一个 PATCH 示例相似。

```
Request: PUT /rest/jobs/schedules/2
```



```

{
  "id": 2,
  "job_id": 2,
  "when_starts": 1,
  "when_ends": 2,
  "starts_unit": "ABS_MS",
  "ends_unit": "ABS_MS",
  "frequency": "WEEKLY",
  "multiplier": 1,
  "state": "NOT_APPROVED",
  "job_properties": {
    "com.securityfirstcorp.atlantis.bundles.keygen.keys.agentIndex": "",
    "com.securityfirstcorp.atlantis.bundles.keygen.keys.keyUpdateIndexes": "1 2 3",
    "runnable.updateJobType": "ROTATE",
    "runnable.keyIndexes2": "1",
    "runnable.fileBytes64":
"LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUdIakNDQkFhZ0F3SUJBZ01DRUFFd0RRWUpLb1pJaH
... omitted ...
DNVYWpCc1VJbHN0a0Uwbk5MTjIyK2JoRGt0SHJNNmZSU1ZESEd3PT0KLS0
tLS1FTkQgQ0VSVElGSUNBVEUtLS0tLQo\u003d",
    "runnable.keyIndexes0": "3",
    "runnable.keyIndexes1": "2",
    "runnable.user": "admin",
    "runnable.notes": "",
    "job-submitter": "admin"
  },
  "job_type": "c.s.a.b.p.j.KeyRotationJob",
  "job_notes": null,
  "type": "job-schedule"
}

```

Response: 201 Created | Location: /rest/jobs/schedulers/4

## /jobs/schedulers

所有调度程序作业。

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含所有已知调度程序作业
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户查看作业
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

```

{
  "type": "array",
  "minItems": 0,
  "uniqueItems": true,
  "items": [

```

```

{
  "type": "object",
  "required": [ "type", "id", "scheduler_id", "job_type",
"state", "substate", "when_created", "when_started", "when_done",
"required_approvals", "required_rejections", "notes", "properties",
"results", "schedules", "actions" ],
  "additionalProperties": false,
  "properties": {
    "type": { "enum": [ "job" ] },
    "id": "number",
    "scheduler_id": "number",
    "job_type": "string",
    "state": { "enum": [ "WAITING", "RUNNING", "DONE",
"REJECTED", "ERROR" ] },
    "substate": { "enum": [ "R_OK", "R_BLOCKED", "D_OK",
"D_REVIEW", "D_FAILED" ] },
    "when_created": "number",
    "when_started": "number",
    "when_done": "number",
    "required_approvals": "number",
    "required_rejections": "number",
    "notes": "string",
    "properties": {
      "type": "object",
      "required": [],
      "additionalProperties": true,
      "properties": {}
    },
    "results": {
      "type": "object",
      "required": [],
      "additionalProperties": true,
      "properties": {}
    },
    "schedules": (array of job-schedule, see 6.2.1.3)
    "actions": {
      "type": [ "array" ],
      "minItems": 0,
      "uniqueItems": true,
      "items": [
        {
          "type": "object",
          "required": [ "type", "id", "job_id", "user",
"action", "when_taken", "notes" ],
          "additionalProperties": false,
          "properties": {
            "type": { "enum": [ "job-action" ] },
            "id": "number",
            "job_id": "number",
            "user": "string",
            "action": { "enum": [ "APPROVE",
"REJECT", "ABSTAIN" ] },
            "when_taken": "number",
            "notes": "string"
          }
        }
      ]
    }
  }
}

```

## 示例

Request: GET /rest/jobs/schedulers

Response: 200 OK

```

[
  {
    "id": 1,
    "scheduler_id": 1,
    "job_type": "c.s.a.b.j.j.SchedulingJob",
    "state": "DONE",
    "substate": "D_OK",
    "when_created": 1503619618974,
    "when_started": 1503672302393,
    "when_done": 1503672302402,
    "required_approvals": 1,

```

```

    "required_rejections": 1,
    "notes": "Creating a security administrator",
    "properties": {
      "c.s.a.b.u.j.createUser": "secadm",
      "runnable.passwordSalt64": "oPgSfm+htrj13c73xgytESU2hiWJVfQkQB8CMcu7l+4\u003d",
      "job-submitter": "admin",
      "runnable.passwordDigest64": "tPV9p7WDjpkNv7czq1KUh+BCC3LcySTsUnZcN10UYyQ\u003d",
      "c.s.a.b.u.j.createNotes": ""
    },
    "results": null,
    "actions": [{
      "id": 2,
      "job_id": 1,
      "user": "admin",
      "action": "APPROVE",
      "when_taken": 1503672302385,
      "notes": "Approved on 16 April [JTK]",
      "type": "job-action"
    }],
    "schedules": [{
      "id": 1,
      "job_id": 1,
      "when_starts": 0,
      "when_ends": 0,
      "starts_unit": "ABS_MS",
      "ends_unit": "ABS_MS",
      "frequency": "ONCE",
      "multiplier": 1,
      "state": "FINISHED",
      "job_properties": {
        "c.s.a.b.u.j.createUser": "secadm",
        "runnable.passwordSalt64": "oPgSfm+htrj13c73xgytESU2hiWJVfQkQB8CMcu7l+4\u003d",
        "job-submitter": "admin",
        "runnable.passwordDigest64": "tPV9p7WDjpkNv7czq1KUh+BCC3LcySTsUnZcN10UYyQ\u003d",
        "c.s.a.b.u.j.createNotes": ""
      },
      "job_type": "c.s.a.b.u.j.CreateJob",
      "job_notes": "Creating a security administrator",
      "type": "job-schedule"
    }],
    "type": "job"
  }
]

```

## 子状态

只有某些状态值存在子状态。不适用时，子状态将为空。子状态值定义如下：

子状态	适用状态	描述
<b>R_OK</b>	<b>RUNNING</b>	作业已获核准，它正在排队等待线程时间以便运行，或者正在运行。目前为止未检测到问题。
<b>R_BLOCKED</b>	<b>RUNNING</b>	作业被完全或部分阻止，导致无法完成，将需要干预。可使用作业的 <b>results</b> 字段中的数据来确定正确的动作过程。
<b>D_OK</b>	<b>DONE</b>	作业已完成运行，并且未发现任何问题。
<b>D_REVIEW</b>	<b>DONE</b>	作业已完成运行，但以编程方式无法确定是成功还是失败。可使用作业的 <b>results</b> 字段中的数据以及对作业的预期效果进行手动验证所生成的数据来确定作业是否成功。此子状态应极少出现。

D_FAILED	DONE	作业已完成运行，并且失败。可使用 <b>results</b> 字段中的数据来识别故障
----------	------	---

## /jobs/schedulers/{id}

特定调度程序作业。

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的作业
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户查看作业或此作业类型
<b>404</b>	找不到具有指定 {id} 的作业，且该作业之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

请参阅第 33 页的『模式』一节，*job* 和 *job-action* 对象。

#### 示例

Request: GET /rest/jobs/schedulers/1

Response: 200 OK

```
{
  "id": 1,
  "scheduler_id": 1,
  "job_type": "c.s.a.b.j.j.SchedulingJob",
  "state": "DONE",
  "substate": "D_OK",
  "when_created": 1503619618974,
  "when_started": 1503672302393,
  "when_done": 1503672302402,
  "required_approvals": 1,
  "required_rejections": 1,
  "notes": "Creating a security administrator",
  "properties": {
    "c.s.a.b.u.j.createUser": "secadm",
    "runnable.passwordSalt64": "oPgSfm+htrj13c73xgytESU2hiWJVfqkQB8CMcu7l+4\u003d",
    "job-submitter": "admin",
    "runnable.passwordDigest64": "tPV9p7WDjpkNv7czq1KUh+BCC3LcySTsUnZcN10UYYQ\u003d",
    "c.s.a.b.u.j.createNotes": ""
  },
  "results": null,
  "actions": [
```

```

        {
          "id": 2,
          "job_id": 1,
          "user": "admin",
          "action": "APPROVE",
          "when_taken": 1503672302385,
          "notes": "Approved on 16 April [JTK]",
          "type": "job-action"
        }
      ],
      "schedules": [
        {
          "id": 1,
          "job_id": 1,
          "when_starts": 0,
          "when_ends": 0,
          "starts_unit": "ABS_MS",
          "ends_unit": "ABS_MS",
          "frequency": "ONCE",
          "multiplier": 1,
          "state": "FINISHED",
          "job_properties": {
            "c.s.a.b.u.j.createUser": "secadm",
            "runnable.passwordSalt64": "oPgSfm+htrj13c73xgytESU2hiWJVfqkQB8CMcu7l+4\u003d",
            "job-submitter": "admin",
            "runnable.passwordDigest64": "tPV9p7WDjpkNv7czq1KUh+BCC3LcySTsUnZcN10UYyQ\u003d",
            "c.s.a.b.u.j.createNotes": ""
          },
          "job_type": "c.s.a.b.u.j.CreateJob",
          "job_notes": "Creating a security administrator",
          "type": "job-schedule"
        }
      ],
      "type": "job"
    }
  }

```

## PATCH

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>204</b>	请求已成功，但未包含任何更改。不需要刷新客户端表示。
<b>205</b>	请求已成功，且资源已更新。
<b>400</b>	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改作业或特定作业类型。
<b>404</b>	找不到具有指定 {id} 的作业，且该作业之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅第 33 页的『模式』一节，*job* 对象。 **更改：** 只有 *type* 和 *id* 是必需字段，并且只有 *notes* 字段可更改。

## 示例

```
Request: PATCH /rest/jobs/schedulers/1

{
  "type": "job",
  "id": 1,
  "notes": "Creating a security administrator (delete after 1/1/2018)"
}

Response: 205 Reset content
```

## PUT

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>204</b>	请求已成功，但未包含任何更改。 不需要刷新客户端表示。
<b>205</b>	请求已成功，且资源已更新。
<b>400</b>	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改作业或特定作业类型
<b>404</b>	找不到具有指定 {id} 的作业，且该作业之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅第 33 页的『模式』一节，*job* 对象。 **更改：** 只有 *notes* 字段可更改。

## 示例

```
Request: PUT /rest/jobs/schedulers/1

{
  "id": 1,
  "scheduler_id": 1,
  "job_type": "c.s.a.b.j.j.SchedulingJob",
  "state": "DONE",
  "substate": "D_OK",
  "when_created": 1503619618974,
  "when_started": 1503672302393,
  "when_done": 1503672302402,
  "required_approvals": 1,
```

```

"required_rejections": 1,
"notes": "Creating a security administrator (delete after 1/1/2018)",
"properties": {
  "c.s.a.b.u.j.createUser": "secadm",
  "runnable.passwordSalt64": "oPgSfm+htrj13c73xgytESU2hiWJVfqkQB8CMcu7l+4\u003d",
  "job-submitter": "admin",
  "runnable.passwordDigest64": "tPV9p7WDjpkNv7czq1KUh+BCC3LcySTsUnZcN10UYYQ\u003d",
  "c.s.a.b.u.j.createNotes": ""
},
"results": null,
"actions": [
  {
    "id": 2,
    "job_id": 1,
    "user": "admin",
    "action": "APPROVE",
    "when_taken": 1503672302385,
    "notes": "Approved on 16 April [JTK]",
    "type": "job-action"
  }
],
"schedules": [
  {
    "id": 1,
    "job_id": 1,
    "when_starts": 0,
    "when_ends": 0,
    "starts_unit": "ABS_MS",
    "ends_unit": "ABS_MS",
    "frequency": "ONCE",
    "multiplier": 1,
    "state": "FINISHED",
    "job_properties": {
      "c.s.a.b.u.j.createUser": "secadm",
      "runnable.passwordSalt64": "oPgSfm+htrj13c73xgytESU2hiWJVfqkQB8CMcu7l+4\u003d",
      "job-submitter": "admin",
      "runnable.passwordDigest64": "tPV9p7WDjpkNv7czq1KUh+BCC3LcySTsUnZcN10UYYQ\u003d",
      "c.s.a.b.u.j.createNotes": ""
    },
    "job_type": "c.s.a.b.u.j.CreateJob",
    "job_notes": "Creating a security administrator",
    "type": "job-schedule"
  }
],
"type": "job"
}

```

Response: 205 Reset content

## /jobs/schedulers/{id}/actions

为特定作业执行的动作。

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含作业的动作数组
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户查看作业或此作业类型
<b>404</b>	找不到具有指定 {id} 的作业，且该作业之前未存在过

<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅第 33 页的『模式』一节，*job* 对象和 *job-action* 对象。

## 示例

Request: GET /rest/jobs/schedulers/1/actions

Response: 200 OK

```
[
  {
    "id": 2,
    "job_id": 1,
    "user": "admin",
    "action": "APPROVE",
    "when_taken": 1503672302385,
    "notes": "Approved on 16 April [JTK]",
    "type": "job-action"
  }
]
```

## POST

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>201</b>	已创建作业动作。 <b>Location</b> 标头将提供新作业动作的 URL。
<b>400</b>	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户查看作业或此作业类型，或者对此作业类型执行所指定的动作。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅第 33 页的『模式』一节，*job* 对象和 *job-action* 对象。 **更改：** 只有 *type*、*job\_id* 和 *action* 是必需字段。 *user*、*id* 和 *when\_taken* 字段是在 *POST* 成功时生成的。



## 示例

```
Request: POST /rest/jobs/schedulers/1/actions
{
  "type": "job-action",
  "job_id": 1,
  "action": "APPROVE",
  "notes": "Approved on 16 April [JTK]"
}

Response: 201 Created
```

## /jobs/history{r}

所有按调度执行的作业。

## GET

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>200</b>	响应主体将包含所有已知调度程序作业
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户查看作业
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅第 33 页的『模式』一节，*job* 对象。这类似于调度程序作业，除了 *id* 字段通常不同于 *scheduler\_id* 字段以外。*scheduler\_id* 字段使执行作业与相应的调度程序作业相关联。

## 示例

```
Request: GET /rest/jobs/history

Response: 200 OK

[
  {
    "id": 4,
    "scheduler_id": 1,
    "job_type": "c.s.a.b.u.j.CreateJob",
    "state": "DONE",
    "substate": "D_OK",
    "when_created": 1503672302432,
    "when_started": 1503672302455,
    "when_done": 1503672302509,
    "required_approvals": 1,
    "required_rejections": 1,
    "notes": "Creating a security administrator",
    "properties": {
```

```

        "c.s.a.b.u.j.createUser":"secadm",
        "runnable.passwordSalt64":"oPgSfm+htrj13c73xgytESU2hiWJVfqkQB8CMcu7l+4\u003d",
        "runnable.passwordDigest64":"tPV9p7WDjpkNv7czq1KUh+BCC3LcySTsUnZcN1OUYYQ\u003d",
        "job-submitter":"admin",
        "c.s.a.b.u.j.createNotes":""
    },
    "results":null,
    "actions":[
        {
            "id": 2,
            "job_id":1,
            "user":"admin",
            "action":"APPROVE",
            "when_taken":1503672302385,
            "notes":"Approved on 16 April [JTK]",
            "type":"job-action"
        }
    ],
    "schedules":[
        {
            "id": 1,
            "job_id":1,
            "when_starts":0,
            "when_ends":0,
            "starts_unit":"ABS_MS",
            "ends_unit":"ABS_MS",
            "frequency":"ONCE",
            "multiplier":1,
            "state":"FINISHED",
            "job_properties":{
                "c.s.a.b.u.j.createUser":"secadm",
                "runnable.passwordSalt64":"oPgSfm+htrj13c73xgytESU2hiWJVfqkQB8CMcu7l+4\u003d",
                "job-submitter":"admin",
                "runnable.passwordDigest64":"tPV9p7WDjpkNv7czq1KUh+BCC3LcySTsUnZcN1OUYYQ\u003d",
                "c.s.a.b.u.j.createNotes":""
            },
            "job_type":"c.s.a.b.u.j.CreateJob",
            "job_notes":"Creating a security administrator",
            "type":"job-schedule"
        }
    ],
    "type":"job"
}
]

```

## /jobs/history/{id}

特定已执行的作业。

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的已执行作业
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户查看作业或此作业类型
<b>404</b>	找不到具有指定 {id} 的作业，且该作业之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。

<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅第 33 页的『模式』一节，*job* 对象。这类似于调度程序作业，除了 *id* 字段通常不同于 *scheduler\_id* 字段以外。*scheduler\_id* 字段使执行作业与相应的调度程序作业相关联。

## 示例

Request: GET /rest/jobs/history/4

Response: 200 OK

```
{
  "id": 4,
  "scheduler_id": 1,
  "job_type": "c.s.a.b.u.j.CreateJob",
  "state": "DONE",
  "substate": "D_OK",
  "when_created": 1503672302432,
  "when_started": 1503672302455,
  "when_done": 1503672302509,
  "required_approvals": 1,
  "required_rejections": 1,
  "notes": "Creating a security administrator",
  "properties": {
    "c.s.a.b.u.j.createUser": "secadm",
    "runnable.passwordSalt64": "oPgSfm+htrj13c73xgytESU2hiWJVfQkQB8CMcu7l+4\u003d",
    "runnable.passwordDigest64": "tPV9p7WDjpkNv7czq1KUUh+BCC3LcySTsUnZcNl0UYyQ\u003d",
    "job-submitter": "admin",
    "c.s.a.b.u.j.createNotes": ""
  },
  "results": null,
  "actions": [
    {
      "id": 2,
      "job_id": 1,
      "user": "admin",
      "action": "APPROVE",
      "when_taken": 1503672302385,
      "notes": "Approved on 16 April [JTK]",
      "type": "job-action"
    }
  ],
  "schedules": [
    {
      "id": 1,
      "job_id": 1,
      "when_starts": 0,
      "when_ends": 0,
      "starts_unit": "ABS_MS",
      "ends_unit": "ABS_MS",
      "frequency": "ONCE",
      "multiplier": 1,
      "state": "FINISHED",
      "job_properties": {
        "c.s.a.b.u.j.createUser": "secadm",
        "runnable.passwordSalt64": "oPgSfm+htrj13c73xgytESU2hiWJVfQkQB8CMcu7l+4\u003d",
        "job-submitter": "admin",
        "runnable.passwordDigest64": "tPV9p7WDjpkNv7czq1KUUh+BCC3LcySTsUnZcNl0UYyQ\u003d",
        "c.s.a.b.u.j.createNotes": ""
      },
      "job_type": "c.s.a.b.u.j.CreateJob",
      "job_notes": "Creating a security administrator",
      "type": "job-schedule"
    }
  ],
  "type": "job"
}
```



# 第 7 章 策略资源

## /pathsets

所有可用的路径集。

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
200	响应主体将包含所有已配置的路径集
401	当前请求未经过认证。
403	未向当前用户进行授权。
406	<b>Accept</b> 标头必须设置为受支持的内容类型。
500	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
503	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

```
{
  "type": "array",
  "minItems": 0,
  "uniqueItems": true,
  "items": [
    {
      "type": "object",
      "required": [ "type", "id", "name", "notes",
"snapshot", "source", "paths", "href" ],
      "additionalProperties": false,
      "properties": {
        "type": { "enum": [ "pathset" ] },
        "id": "number",
        "name": "string",
        "notes": "string",
        "snapshot": "boolean",
        "source": "number",
        "paths": {
          "type": [ "array" ],
          "minItems": 0,
          "uniqueItems": true,
          "items": [
            {
              "type": "object",
              "required": [ "type", "id", "path_set_id",
"path", "notes", "source", "snapshot" ],
              "additionalProperties": false,
              "properties": {
                "type": { "enum": [ "path" ] },
                "id": "number",
                "path": "string",
```

```
{
  [
    {
      }
    ]
  ]
}
```

```
"path_set_id": "number",
"notes": "string",
"source": "number",
"snapshot": "boolean"
}
}
```

### 示例

Request: GET /rest/pathsets

Response: 200 OK

```
[
  {
    "id": 1,
    "name" : "newPathset",
    "notes": "",
    "source": 0,
    "snapshot": false,
    "paths" : [
      {
        "id": 1,
        "path_set_id" : 1,
        "path" : "/file/path/one",
        "notes": "",
        "source": 0,
        "snapshot": false,
        "type" : "path"
      },
      {
        "id": 2,
        "path_set_id" : 1,
        "path" : "/file/path/two",
        "notes": "",
        "source": 0,
        "snapshot": false,
        "type" : "path"
      }
    ],
    "href": null,
    "type": "pathset"
  }
]
```

## POST

### 查询参数

参数	描述
<无>	

状态码

代码	描述
<b>201</b>	已创建路径集。 <b>Location</b> 标头将提供新路径集的 URL。
<b>400</b>	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户创建路径集。

<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>409</b>	新路径集与现有路径集名称或标识相冲突。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅路径集集合模式，pathset 和 path 对象。 **更改：** *id* 不是必需字段，也不应为任一对象类型设置该字段。

### 示例

```
Request: POST /rest/pathsets
{
  "name" : "newPathset",
  "notes": "",
  "source": 0,
  "snapshot": false,
  "paths" : [
    {
      "path" : "/file/path/one",
      "notes": "",
      "source": 0,
      "snapshot": false,
      "type" : "path"
    },
    {
      "path" : "/file/path/two",
      "notes": "",
      "source": 0,
      "snapshot": false,
      "type" : "path"
    }
  ],
  "type": "pathset"
}

Response: 201 Created | Location: /rest/pathsets/1
```

## /pathsets/{id}

特定路径集

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的路径集
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。

<b>404</b>	找不到具有指定 {id} 的路径集，且该路径集之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅路径集集合模式，pathset 和 path 对象。

### 示例

```
Request: GET /rest/pathsets/1
Response: 200 OK
{
  "id": 1,
  "name": "newPathset",
  "notes": "",
  "source": 0,
  "snapshot": false,
  "paths": [
    {
      "id": 1,
      "path_set_id": 1,
      "path": "/file/path/one",
      "notes": "",
      "source": 0,
      "snapshot": false,
      "type": "path"
    },
    {
      "id": 2,
      "path_set_id": 1,
      "path": "/file/path/two",
      "notes": "",
      "source": 0,
      "snapshot": false,
      "type": "path"
    }
  ],
  "href": null,
  "type": "pathset"
}
```

## DELETE

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>204</b>	删除已成功
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。



<b>404</b>	找不到具有指定 {id} 的路径集，且该路径集之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

*DELETE* 中不发送主体

### 示例

Request: DELETE /rest/pathsets/1

Response: 204 No content

## PATCH

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>204</b>	请求已成功，但未包含任何更改。不需要刷新客户端表示。
<b>205</b>	请求已成功，且资源已更新。
<b>400</b>	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改进程集。
<b>404</b>	找不到具有指定 {id} 的路径集，且该路径集之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>409</b>	对路径集所做的更改与另一个路径集相冲突。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅路径集集合模式，pathset 和 path 对象。 **更改**：对于这两种对象类型，只有 *type* 和 *id* 是必需字段。

### 示例

Request: PATCH /rest/pathsets/1

```
{
  "type": "pathset",
  "id": 1,
  "name": "newPathSetName"
}
```

Response: 205 Reset content

## PUT

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>204</b>	请求已成功，但未包含任何更改。不需要刷新客户端表示。
<b>205</b>	请求已成功，且资源已更新。
<b>400</b>	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改进程集
<b>404</b>	找不到具有指定 {id} 的路径集，且该路径集之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>409</b>	对路径集所做的更改与另一个路径集相冲突。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅路径集集合模式，*pathset* 和 *path* 对象。

### 示例

Request: PUT /rest/pathsets/1

```
{
  "id": 1,
  "name": "newPathSetName",
  "notes": "Add some notes to the path set",
  "source": 0,
  "snapshot": false,
  "paths": [
    {
      "id": 1,
      "path_set_id": 1,
      "path": "/file/path/one",
      "notes": "",
      "source": 0,
      "snapshot": false,
      "type": "path"
    }
  ]
}
```

```

      "id": 2,
      "path_set_id" : 1,
      "path" : "/file/path/two",
      "notes": "",
      "source": 0,
      "snapshot": false,
      "type" : "path"
    }
  ],
  "href": null,
  "type": "pathset"
}

```

Response: 205 Reset content

## /processes

所有可用的进程

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含所有已配置的进程
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

```

{
  "type": "array",
  "minItems": 0,
  "uniqueItems": true,
  "items": [
    {
      "type": "object",
      "required": [ "type", "id", "name", "path",
"version", "os", "distribution", "snapshot", "source", "rows" ],
      "additionalProperties": false,
      "properties": {
        "type": { "enum": [ "process" ] },
        "id": "number",
        "name": "string",
        "path": "string",
        "version": "number",
        "os": "number",
        "distribution": "number",
        "snapshot": "boolean",
        "source": "number",
        "rows": {
          "type": [ "array" ],
          "minItems": 0,

```

```
        "uniqueItems": true,
        "items": [
            {
                "type": "object",
                "required": [ "type", "id", "hash", "source", "snapshot" ],
                "additionalProperties": false,
                "properties": {
                    "type": { "enum": [ "process-hash" ] },
                    "id": "number",
                    "hash": "string",
                    "source": "number",
                    "snapshot": "boolean"
                }
            }
        ]
    }
}
```

操作系统

“操作系统” 值表示为整数。 这些整数值按如下所示进行映射：

值	映射
0	无
1	Linux
2	Windows

分发版

“分发版” 值表示为整数。 这些整数值按如下所示进行映射：

值	映射
0	无
1	Red Hat
2	CentOS
3	Windows

示例

```
Request: GET /processes
Response: 200 OK
[
  {
    "id": 1,
    "name": "testProcess",
    "path": "/home/bin/processA",
    "version": "1.1",
    "os": 1,
    "distribution": 2,
    "source": 0,
    "snapshot": false,
    "rows": [
      {
        "id": 1,
        "hash": "01200102030405060708091011121314151617181920212223242526272829303132",
        "source": 0,
        "snapshot": false,
        "type": "process-hash"
      },
      {
        "id": 2,
        "hash":
```

```

"01200102030405060708091011121314151617181920212223242526272829303132",
    "source": 0,
    "snapshot": false,
    "type": "process-hash"
  },
  {
    "href": null,
    "type": "process"
  },
  {
    "id": 2,
    "name": "testProcess",
    "path": "/home/bin/processA",
    "version": "1.1",
    "os": 1,
    "distribution": 2,
    "source": 1,
    "snapshot": true,
    "rows": [],
    "href": null,
    "type": "process"
  }
]

```

## POST

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>201</b>	已创建进程。 <b>Location</b> 标头将提供新进程集的 URL。
<b>400</b>	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户创建进程。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>409</b>	新进程与现有进程集名称或标识相冲突。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅进程集合模式， *process* 和 *process-hash* 对象。 **更改：** *id* 不是必需字段，也不应为任一对象类型设置该字段。

### 示例

```

Request: POST /processes

{
  "name": "newProcess",
  "path": "/home/bin/newProcess",
  "version": "1.1",
  "os": 1,
  "distribution": 2,

```

```

    "source": 0,
    "snapshot": false,
    "rows": [
      {
        "hash":
"01200102030405060708091011121314151617181920212223242526272829303132",
        "source": 0,
        "snapshot": false,
        "type": "process-hash"
      },
      {
        "hash":
"01200102030405060708091011121314151617181920212223242526272829303132",
        "source": 0,
        "snapshot": false,
        "type": "process-hash"
      }
    ],
    "type": "process"
  }
}

Response: 201 Created | Location: /processes/2

```

## /processes/{id}

特定进程

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的进程
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 的进程，且该进程之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

请参阅进程集合模式，*process* 和 *process-hash* 对象。

#### 示例

```

Request: GET /rest/processes/1

Response: 200 OK

{
  "id": 1,
  "name": "testProcess",
  "path": "/home/bin/processA",

```

```

    "version": "1.1",
    "os": 1,
    "distribution": 2,
    "source": 0,
    "snapshot": false,
    "rows": [
      {
        "id": 1,
        "hash": "01200102030405060708091011121314151617181920212223242526272829303132",
        "source": 0,
        "snapshot": false,
        "type": "process-hash"
      },
      {
        "id": 2,
        "hash": "01200102030405060708091011121314151617181920212223242526272829303132",
        "source": 0,
        "snapshot": false,
        "type": "process-hash"
      }
    ],
    "href": null,
    "type": "process"
  },

```

## DELETE

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>204</b>	删除已成功
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 的进程，且该进程之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

DELETE 中不发送主体

### 示例

Request: DELETE /rest/processes/1  
Response: 204 No content

## PATCH

### 查询参数

参数	描述
----	----

<无>	
-----	--

## 状态码

代码	描述
<b>204</b>	请求已成功，但未包含任何更改。不需要刷新客户端表示。
<b>205</b>	请求已成功，且资源已更新。
<b>400</b>	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改进程集。
<b>404</b>	找不到具有指定 {id} 的进程，且该进程之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>409</b>	对进程所做的更改与另一个进程相冲突。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅进程集合模式，*process* 和 *process-hash* 对象。 **更改：**对于这两种对象类型，只有 *type* 和 *id* 是必需字段。

## 示例

Request: PATCH /rest/processes/1

```
{
  "type": "process",
  "id": 1,
  "name": "ProcessA-BC"
}
```

Response: 205 Reset content

## PUT

### 查询参数

参数	描述
<无>	

## 状态码

代码	描述
<b>204</b>	请求已成功，但未包含任何更改。不需要刷新客户端表示。
<b>205</b>	请求已成功，且资源已更新。
<b>400</b>	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。



401	当前请求未经过认证。
403	未授权当前用户更改进程集
404	找不到具有指定 {id} 的进程，且该进程之前未存在过
406	<b>Accept</b> 标头必须设置为受支持的内容类型。
409	对进程所做的更改与另一个进程相冲突。
500	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
503	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

模式

请参阅进程集合模式， *process* 和 *process-hash* 对象。

示例

```
Request: PUT /rest/processes/2
{
  "id": 2,
  "name": "newProcess",
  "path": "/home/bin/newProcess",
  "version": "1.1",
  "os": 1,
  "distribution": 2,
  "source": 0,
  "snapshot": false,
  "rows": [
    {
      "id": 4,
      "hash":
"01200102030405060708091011121314151617181920212223242526272829303132",
      "source": 0,
      "snapshot": false,
      "type": "process-hash"
    },
    {
      "id": 5,
      "hash":
"01200102030405060708091011121314151617181920212223242526272829303132",
      "source": 0,
      "snapshot": false,
      "type": "process-hash"
    }
  ],
  "type": "process"
}
Response: 205 Reset content
```

[/selectors](#)

所有可用的选择器

GET

查询参数

参数	描述
<无>	

## 状态码

代码	描述
<b>200</b>	响应主体将包含所有已配置的选择器
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

```
{
  "type": "array",
  "minItems": 0,
  "uniqueItems": true,
  "items": [
    {
      "type": "object",
      "required": [ "type", "id", "name", "notes", "snapshot", "rows", "href" ],
      "additionalProperties": false,
      "properties": {
        "type": { "enum": [ "selector" ] },
        "id": "number",
        "name": "string",
        "notes": "string",
        "snapshot": "boolean",
        "source": "number",
        "rows": {
          "type": [ "array" ],
          "minItems": 0,
          "uniqueItems": true,
          "items": [
            {
              "type": "object",
              "required": [ "type", "id", "order",
"user", "group", "external_group", "process_set", "snapshot" ],
              "additionalProperties": false,
              "properties": {
                "type": { "enum": [ "selector-row" ] },
                "id": "number",
                "order": "number",
                "user": "string",
                "group": "string",
                "external_group": "boolean",
                "process_set": {
                  "type": "object",
                  "required": [ "type", "id", "href" ],
                  "additionalProperties": true,
                  "properties": {
                    "type": { "enum": [ "ref" ] },
                    "id": "number",
                    "href": "string"
                  }
                }
              },
              "source": "number",
              "snapshot": "boolean"
            }
          ]
        }
      }
    }
  ]
}
```

示例

```
Request: GET /rest/selectors

Response: 200 OK

[
  {
    "id": 2,
    "name": "selector1",
    "notes": "",
    "snapshot": false,
    "source": 0,
    "rows": [
      {
        "id": 2,
        "order": 1,
        "user": "user1",
        "group": null,
        "external_group": false,
        "process_set": {
          "id": 0,
          "href": "/rest/processsets/0",
          "type": "ref"
        },
        "source": 0,
        "snapshot": false,
        "type": "selector-row"
      },
      {
        "id": 3,
        "type": "selector-row",
        "order": 2,
        "user": "sfc1",
        "group": null,
        "external_group": false,
        "process_set": null,
        "snapshot": false,
        "source": 0
      }
    ],
    "href": null,
    "type": "selector"
  }
]
```

POST

查询参数

参数	描述
<无>	

状态码

代码	描述
201	已创建选择器。 <b>Location</b> 标头将提供新选择器的 URL。
400	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
401	当前请求未经过认证。
403	未授权当前用户创建选择器。
406	<b>Accept</b> 标头必须设置为受支持的内容类型。
409	新选择器与现有选择器名称或标识相冲突。

<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

模式

请参阅选择器集合模式，*selector* 和 *selector-row* 对象。 **更改：** *id* 不是必需字段，也不应为任一对象类型设置该字段。

示例

```
Request: POST /rest/selectors

{
  "type": "selector",
  "name": "sfcUsers2",
  "notes": "Manual copy of selector for internal users",
  "snapshot": false,
  "source": 0,
  "rows": [
    {
      "type": "selector-row",
      "order": 1,
      "user": "sfc0",
      "group": null,
      "external_group": false,
      "process_set": null,
      "snapshot": false,
      "source": 0
    },
    {
      "type": "selector-row",
      "order": 2,
      "user": "sfc1",
      "group": null,
      "external_group": false,
      "process_set": null,
      "snapshot": false,
      "source": 0
    }
  ]
}

Response: 201 Created | Location: /rest/selectors/3
```

[/selectors/{id}](#)

特定选择器

GET

查询参数

参数	描述
<无>	

状态码

代码	描述
<b>200</b>	响应主体将包含所请求的选择器
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。

<b>404</b>	找不到具有指定 {id} 的选择器，且该选择器之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅选择器集合模式，*selector* 和 *selector-row* 对象。

### 示例

```
Request: GET /rest/selectors/3
Response: 200 OK
{
  "type": "selector",
  "id": 3,
  "name": "sfcUsers2",
  "notes": "Manual copy of selector for internal users",
  "snapshot": false,
  "source": 0,
  "rows": [
    {
      "type": "selector-row",
      "id": 5,
      "order": 1,
      "user": "sfc0",
      "group": null,
      "external_group": false,
      "process_set": null,
      "snapshot": false,
      "source": 0
    },
    {
      "type": "selector-row",
      "id": 6,
      "order": 2,
      "user": "sfc1",
      "group": null,
      "external_group": false,
      "process_set": null,
      "snapshot": false,
      "source": 0
    }
  ],
  "href": null
}
```

## DELETE

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>204</b>	删除已成功
<b>401</b>	当前请求未经过认证。

<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 的选择器，且该选择器之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

*DELETE* 中不发送主体

## 示例

Request: DELETE /rest/selectors/3

Response: 204 No content

## PATCH

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>204</b>	请求已成功，但未包含任何更改。不需要刷新客户端表示。
<b>205</b>	请求已成功，且资源已更新。
<b>400</b>	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改选择器。
<b>404</b>	找不到具有指定 {id} 的选择器，且该选择器之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>409</b>	对选择器所做的更改与另一个选择器相冲突。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅选择器集合模式，*selector* 和 *selector-row* 对象。 **更改：** 对于这两种对象类型，只有 *type* 和 *id* 是必需字段。

示例

```
Request: PATCH /rest/selectors/3

{
  "type": "selector",
  "id": 3,
  "notes": "Change to the notes field",
}

Response: 205 Reset content
```

PUT

查询参数

参数	描述
<无>	

状态码

代码	描述
204	请求已成功，但未包含任何更改。 不需要刷新客户端表示。
205	请求已成功，且资源已更新。
400	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
401	当前请求未经过认证。
403	未授权当前用户更改选择器
404	找不到具有指定 {id} 的选择器，且该选择器之前未存在过
406	<b>Accept</b> 标头必须设置为受支持的内容类型。
409	对选择器所做的更改与另一个选择器相冲突。
500	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
503	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

模式

请参阅选择器集合模式，*selector* 和 *selector-row* 对象。

示例

```
Request: PUT /rest/selectors/3

{
  "type": "selector",
  "id": 3,
  "name": "sfcUsers2",
  "notes": "Another change to the notes field",
  "snapshot": false,
  "source": 0,
  "rows": [
    {
      "type": "selector-row",
      "id": 5,
      "order": 1,
      "user": "sfc0",
    }
  ]
}
```

```

        "group": null,
        "external_group": false,
        "process_set": null,
        "snapshot": false,
        "source": 0
      },
      {
        "type": "selector-row",
        "id": 6,
        "order": 2,
        "user": "sfc1",
        "group": null,
        "external_group": false,
        "process_set": null,
        "snapshot": false,
        "source": 0
      }
    ]
  }
}

```

Response: 205 Reset content

## /keys

所有可用的已命名密钥

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含所有已配置的已命名密钥
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

```

{
  "type": "array",
  "minItems": 0,
  "uniqueItems": true,
  "items": [
    {
      "type": "object",
      "required": [ "type", "id", "name", "notes", "scope", "autogen", "versions" ],
      "additionalProperties": true,
      "properties": {
        "type": { "enum": [ "encryption-key" ] },
        "id": "number",
        "name": "string",
        "notes": "string",
        "scope": { "enum": [ "user", "internal" ] },
        "autogen": "boolean",

```



```

        "versions": {
          "type": [ "array" ],
          "minItems": 1,
          "uniqueItems": true,
          "items": [
            {
              "type": "object",
              "required": [ "id", "version", "current", "timestamp" ],
              "additionalProperties": false,
              "properties": {
                "id": "number",
                "version": "number",
                "current": "boolean",
                "timestamp": "number"
              }
            }
          ]
        }
      }
    ]
  }
}

```

### 示例

```

Request: GET /rest/keys
Response: 200 OK

[
  {
    "id": 1,
    "name": "keyA",
    "autogen": false,
    "versions": [
      {
        "id": 1,
        "version": 1,
        "current": true,
        "timestamp": 1478268524124
      }
    ],
    "scope": "user",
    "notes": "",
    "type": "encryption-key"
  },
  {
    "id": 2,
    "name": "keyB",
    "autogen": false,
    "versions": [
      {
        "id": 2,
        "version": 1,
        "current": true,
        "timestamp": 1478268526490
      }
    ],
    "scope": "user",
    "notes": "",
    "type": "encryption-key"
  }
]

```

### POST

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
----	----

<b>201</b>	已创建密钥。 <b>Location</b> 标头将提供新密钥的 URL。
<b>400</b>	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户创建密钥。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>409</b>	新密钥与现有密钥名称或标识相冲突。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅密钥集合模式， *encryption-key* 对象。 **更改：** *id* 不是必需字段， 也不应设置该字段。 *POST* 所创建的密钥只能使用 “*user*” 作用域。

## 示例

```
Request: POST /rest/keys
{
  "type": "encryption-key",
  "name": "testKey",
  "notes": "Key for users in the test org",
  "scope": "user",
  "autogen": false
}
Response: 201 Created | Location: /rest/keys/1
```

## /keys/{id}

特定已命名密钥

## GET

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的已命名密钥
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 的已命名密钥， 且该已命名密钥之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。

<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅密钥集合模式，*encryption-key* 对象。

## 示例

```
Request: GET /rest/keys/1
Response: 200 OK
{
  "id": 1,
  "name": "keyA",
  "autogen": false,
  "versions": [
    {
      "id": 1,
      "version": 1,
      "current": true,
      "timestamp": 1478268524124
    }
  ],
  "scope": "user",
  "notes": "",
  "type": "encryption-key"
}
```

## PATCH

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>204</b>	请求已成功，但未包含任何更改。不需要刷新客户端表示。
<b>205</b>	请求已成功，且资源已更新。
<b>400</b>	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改密钥。
<b>404</b>	找不到具有指定 {id} 的密钥，且该密钥之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>409</b>	对密钥所做的更改与另一个密钥相冲突。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅密钥集合模式，*encryption-key* 对象。 **更改：** 只有 *type* 和 *id* 是必需字段。

## 示例

```
Request: PATCH /rest/keys/1

{
  "type": "encryption-key",
  "id": 1,
  "name": "testKey-orig",
  "notes": "Original key for users in the test org"
}

Response: 205 Reset content
```

## PUT

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>204</b>	请求已成功，但未包含任何更改。 不需要刷新客户端表示。
<b>205</b>	请求已成功，且资源已更新。
<b>400</b>	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改密钥
<b>404</b>	找不到具有指定 {id} 的密钥，且该密钥之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>409</b>	对密钥所做的更改与另一个密钥相冲突。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅密钥集合模式，*encryption-key* 对象。

## 示例

```
Request: PUT /rest/keys/1

{
  "id": 1,
  "name": "testKey",
  "autogen": false,
  "versions": [
    {
      "id": 3,
      "version": 1,
    }
  ]
}
```

```

        "current": true,
        "timestamp": 1478547390464
      },
    ],
    "scope": "user",
    "notes": "Key for users in the test org",
    "type": "encryption-key"
  }
}

```

Response: 205 Reset content

## /groupscopes

外部组成员资格作用域。

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含所有已配置的组作用域
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

```

{
  "type": "array",
  "minItems": 0,
  "uniqueItems": true,
  "items": [
    {
      "type": "object",
      "required": [ "type", "name", "host", "port",
"secure", "bind_dn", "bind_password", "groups_dn", "users_dn" ],
      "additionalProperties": false,
      "properties": {
        "type": { "enum": [ "group-scope" ] },
        "name": "string",
        "host": "string",
        "port": "number",
        "secure": "boolean",
        "bind_dn": "string",
        "bind_password": "string",
        "groups_dn": "string",
        "users_dn": "string"
      }
    }
  ]
}

```

## 示例

Request: GET /rest/groupscores

Response: 200 OK

```
[
  {
    "type": "group-scope",
    "name": "ldap",
    "host": "192.168.4.223",
    "port": 389,
    "secure": false,
    "bind_dn": "cn=admin,ou=Users,dc=example,dc=com",
    "bind_password": null,
    "users_dn": "ou=Users,dc=example,dc=com",
    "groups_dn": "ou=Groups,dc=example,dc=com"
  }
]
```

## /groupscores/{name}

特定组作用域

## GET

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的组作用域
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {name} 的组作用域，且该组作用域之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅组作用域集合模式，*group-scope* 对象。

## 示例

Request: GET /rest/groupscores/ldap

Response: 200 OK

```
{
  "type": "group-scope",
  "name": "ldap",
  "host": "192.168.4.223",
  "port": 389,
```

```

"secure": false,
"bind_dn": "cn=admin,ou=Users,dc=example,dc=com",
"bind_password": null,
"users_dn": "ou=Users,dc=example,dc=com",
"groups_dn": "ou=Groups,dc=example,dc=com"
}

```

## PATCH

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>204</b>	请求已成功，但未包含任何更改。不需要刷新客户端表示。
<b>205</b>	请求已成功，且资源已更新。
<b>400</b>	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改组作用域。
<b>404</b>	找不到具有指定 {name} 的组作用域，且该组作用域之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅组作用域集合模式，*group-scope* 对象。 **更改**：只有 *type* 和 *name* 是必需字段。

### 示例

Request: PATCH /rest/groupsscopes/ldap

```

{
  "type": "group-scope",
  "name": "ldap",
  "port": 636,
  "secure": true,
}

```

Response: 205 Reset content

## PUT

### 查询参数

参数	描述
<无>	

状态码

代码	描述
204	请求已成功，但未包含任何更改。不需要刷新客户端表示。
205	请求已成功，且资源已更新。
400	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
401	当前请求未经过认证。
403	未授权当前用户更改组作用域
404	找不到具有指定 {name} 的组作用域，且该组作用域之前未存在过
406	<b>Accept</b> 标头必须设置为受支持的内容类型。
500	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
503	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

模式

请参阅组作用域集合模式，*group-scope* 对象。

示例

```
Request: PUT /rest/groupsscopes/ldap

{
  "type": "group-scope",
  "name": "ldap",
  "host": "192.168.4.223",
  "port": 636,
  "secure": true,
  "bind_dn": "cn=admin,ou=Users,dc=example,dc=com",
  "bind_password": null,
  "users_dn": "ou=Users,dc=example,dc=com",
  "groups_dn": "ou=Groups,dc=example,dc=com"
}

Response: 205 Reset content
```

/groupsets

所有可用的组集

.

GET

查询参数

参数	描述
<无>	

状态码

代码	描述
----	----



<b>200</b>	响应主体将包含所有已配置的组集
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

模式

```
{
  "type": "array",
  "minItems": 0,
  "uniqueItems": true,
  "items": [
    {
      "type": "object",
      "required": [ "type", "id", "policy_binding_id", "groups" ],
      "additionalProperties": false,
      "properties": {
        "type": { "enum": [ "group-set" ] },
        "id": "number",
        "policy_binding_id": "number",
        "groups": {
          "type": [ "array" ],
          "minItems": 0,
          "uniqueItems": true,
          "items": [
            {
              "type": "object",
              "required": [ "type", "name", "members" ],
              "additionalProperties": false,
              "properties": {
                "type": { "enum": [ "group " ] },
                "name": "string",
                "members": {
                  "type": [ "array" ],
                  "minItems": 0,
                  "uniqueItems": true,
                  "items": [
                    {
                      "type":

```

示例

```
Request: GET /rest/groupsets
Response: 200 OK
[
  {
    "type": "group-set",
    "id": 1,
    "policy_binding_id": 1,
    "groups": [
      {
        "type": "group",
        "name": "users",

```

```

    "members": [
      "user0",
      "anotheruser",
      "root",
      "someone"
    ]
  }
}
]
```

## POST

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>201</b>	已创建组集。 <b>Location</b> 标头将提供新组集的 URL。
<b>400</b>	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户创建组集。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅组集集合模式， *group-set* 对象。 **更改：** *POST* 中只应发送 *type* 和 *policy\_binding\_id*， 所有其他字段都是在服务器端生成的。

### 示例

```
Request: POST /rest/groupsets
{
  "type": "group-set",
  "policy_binding_id": 1
}

Response: 201 Created | Location: /rest/groupsets/3
```

## [/groupsets/{id}](#)

特定组集

## GET

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的组集
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 的组集，且该组集之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅组集集合模式，*group-set* 对象。

### 示例

Request: GET /rest/groupsets/3

Response: 200 OK

```
{
  "type": "group-set",
  "id": 3,
  "policy_binding_id": 1,
  "groups": [
    {
      "type": "group",
      "name": "users",
      "members": [
        "user1",
        "user2",
        "user3"
      ]
    }
  ]
}
```

## DELETE

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>204</b>	删除已成功

<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 的组集，且该组集之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

*DELETE* 中不发送主体

## 示例

Request: DELETE /groupsets/3

Response: 204 No content

## /datatypes

所有可用的数据类型

## GET

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>200</b>	响应主体将包含所有已配置的数据类型
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

```
{
  "type": "array",
  "minItems": 0,
  "uniqueItems": true,
  "items": [
    {
      "type": "object",
      "required": [ "type", "id", "name",
"notes", "snapshot", "source", "rows" ],
```

```

"additionalProperties": false,
"properties": {
  "type": { "enum": [ "data-type" ] },
  "id": "number",
  "name": "string",
  "notes": "string",
  "snapshot": "boolean",
  "source": "number",
  "rows": {
    "type": [ "array" ],
    "minItems": 0,
    "uniqueItems": true,
    "items": [
      {
        "type": "object",
        "required": [ "type", "id",
"datatype_id", "order", "selector", "ops", "actions", "notes",
"override_flags", "snapshot", "source"],
        "additionalProperties": true,
        "properties": {
          "type": { "enum": [ "data-type-row" ] },
          "id": "number",
          "datatype_id": "number",
          "order": "number",
          "selector": {
            "type": "object",
            "required": [ "type", "id", "href" ],
            "additionalProperties": true,
            "properties": {
              "type": { "enum": [ "ref" ] },
              "id": "number",
              "href": "string"
            }
          },
          "ops": { "enum": [ "read", "write", "read/write" ] },
          "actions": { "enum": [ "permit",
"permit, log", "deny", "deny, log" ] },
          "notes": "string",
          "override_flags": {
            "type": "object",
            "required": "required": [ "type"],
            "additionalProperties": true,
            "properties": {
              "type": { "enum": [ "override-flag-map" ] }
            }
          },
          "source": "number",
          "snapshot": "boolean"
        }
      }
    ]
  }
}

```

## 覆盖标志

数据类型中的每个字段都具有相应的覆盖标志值。对于允许值“may”、“may\_not”和“must”（在下面定义），这些值分别指示代理程序策略绑定是否可以、不可以或必须覆盖数据类型中的值。通过将数据类型标识、行标识和列标识编码到映射的字段名称中，使标志与数据类型中的特定字段相关联。

对于任何映射，如果没有在数据类型行 POST 或 PUT 中指定值，那么标志将存储为“may\_not”。缺少映射值将被认为在执行 PATCH 时保持不变。

## 覆盖标志值及含义

标志值	含义
<b>may</b>	如果代理程序策略绑定设置了另一个值，那么将使用绑定值。否则，将使用数据类型行所定义的值。
<b>may_not</b>	无论代理程序策略绑定是否设置了另一个值，都使用数据类型行所定义的值。

<b>must</b>	代理程序策略绑定必须将另一个值设置为有效。将不使用数据类型行本身中存储的任何值。
-------------	--

**覆盖标志字段名称编码**

映射字段名称的构建方式是，依次将带有各自前缀的数据类型标识、数据类型行标识和列标识（在下面定义）进行并置。也就是说，如果 x、y 和 z 分别表示数据类型标识、数据类型行标识和列标识，那么通过将 x、y 和 z 替换到字符串“dxrycz”（不带引号）中来指定覆盖标志的字段名称。

对数据类型行执行首个 POST 时，数据类型标识和数据类型行标识可能为 0，这意味着该标志适用于外层行和外层数据类型。

行字段	列标识
<b>selector</b>	1
<b>ops</b>	2
<b>actions</b>	3

**覆盖标志示例：**

下面为数据类型 1 和数据类型行 1 定义覆盖标志，这些标志将：

- 防止行选择器被覆盖
- 要求覆盖行操作
- 允许选择是否覆盖行动作

```
"override_flags": {
  "d1r1c1": "may_not",
  "d1r1c2": "must",
  "d1r1c3": "may",
  "type": "override-flag-map"
},
```

**示例**

```
Request: GET /rest/datatypes
Response: 200 OK

[
  {
    "id": 1,
    "name": "testDatatype",
    "notes": "",
    "source": 0,
    "snapshot": false,
    "rows": [
      {
        "id": 1,
        "datatype_id": 1,
        "order": 1,
        "selector": {
          "id": 3,
          "href": "/rest/selectors/3",
          "type": "ref"
        },
        "ops": "read/write",
        "actions": "permit",
        "override_flags": {
          "d1r1c1": "may_not",
          "d1r1c2": "may_not",
          "d1r1c3": "may_not",
          "type": "override-flag-map"
        },
        "notes": "",
        "source": 0,
        "snapshot": false,
        "type": "data-type-row"
      }
    ]
  }
]
```

```

    ],
    "type": "data-type"
  }
]

```

## POST

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>201</b>	已创建数据类型。 <b>Location</b> 标头将提供新策略的 URL。
<b>400</b>	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户创建数据类型。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅策略集合模式，*policy* 和 *policy-row* 对象。 **更改：** *id* 不是必需字段，也不应为任一对象类型设置该字段。

### 示例

Request: POST /rest/datatypes

```

{
  "name": "testDatatype",
  "notes": "",
  "source": 0,
  "snapshot": false,
  "rows": [
    {
      "order": 1,
      "selector": {
        "id": 3,
        "href": "/rest/selectors/3",
        "type": "ref"
      },
      "ops": "read/write",
      "actions": "permit",
      "override_flags": {
        "d0r0c1": "may_not",
        "d0r0c2": "may_not",
        "d0r0c3": "may_not",
        "type": "override-flag-map"
      },
      "notes": "",
      "source": 0,
      "snapshot": false,
      "type": "data-type-row"
    }
  ],
}

```

```
    "type": "data-type"
  }
Response: 201 Created | Location: / datatypes/1
```

## /datatypes/{id}

特定数据类型

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的数据类型
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 的数据类型，且该数据类型之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅策略集合模式，*policy* 和 *policy-row* 对象。

### 示例

```
Request: GET /rest/datatypes/1
Response: 200 OK
{
  "id": 1,
  "name": "testDatatype",
  "notes": "",
  "source": 0,
  "snapshot": false,
  "rows": [
    {
      "id": 1,
      "datatype_id": 1,
      "order": 1,
      "selector": {
        "id": 3,
        "href": "/rest/selectors/3",
        "type": "ref"
      },
      "ops": "read/write",
      "actions": "permit",
      "override_flags": {
```



```

        "d1r1c1": "may_not",
        "d1r1c2": "may_not",
        "d1r1c3": "may_not",
        "type": "override-flag-map"
      },
      "notes": "",
      "source": 0,
      "snapshot": false,
      "type": "data-type-row"
    },
    {
      "type": "data-type"
    }
  ]
}

```

## DELETE

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>204</b>	删除已成功
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 的数据类型，且该数据类型之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

DELETE 中不发送主体

### 示例

```

Request: DELETE /rest/datatypes/1
Response: 204 No content

```

## PATCH

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
----	----

<b>204</b>	请求已成功，但未包含任何更改。不需要刷新客户端表示。
<b>205</b>	请求已成功，且资源已更新。
<b>400</b>	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改数据类型。
<b>404</b>	找不到具有指定 {id} 的数据类型，且该数据类型之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>409</b>	对策略所做的更改与另一个策略相冲突。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅数据类型集合模式，*data-type* 和 *data-type-row* 对象。 **更改**：对于这两种对象类型，只有 *type* 和 *id* 是必需字段。

## 示例

```
Request: PATCH /rest/datatypes/1
{
  "id": 1,
  "name": "newDatatypeName",
  "notes": "new datatype notes",
  "type": "data-type"
}
Response: 205 Reset content
```

## PUT

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>204</b>	请求已成功，但未包含任何更改。不需要刷新客户端表示。
<b>205</b>	请求已成功，且资源已更新。
<b>400</b>	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改数据类型

<b>404</b>	找不到具有指定 {id} 的数据类型，且该数据类型之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>409</b>	对数据类型所做的更改与另一种数据类型相冲突。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅数据类型集合模式，*data-type* 和 *data-type-row* 对象。

## 示例

Request: PUT /rest/datatypes/1

Response: 205 Reset content

```
{
  "id": 1,
  "name": "testDatatype",
  "notes": "updated with new selector and overrides",
  "source": 0,
  "snapshot": false,
  "rows": [
    {
      "id": 1,
      "datatype_id": 1,
      "order": 1,
      "selector": {
        "id": 4,
        "href": "/rest/selectors/4",
        "type": "ref"
      },
      "ops": "read/write",
      "actions": "permit",
      "override_flags": {
        "d1r1c1": "may",
        "d1r1c2": "may",
        "d1r1c3": "may",
        "type": "override-flag-map"
      },
      "notes": "",
      "source": 0,
      "snapshot": false,
      "type": "data-type-row"
    }
  ],
  "type": "data-type"
}
```



# 第 8 章 受管代理程序

## /agents

所有受管代理程序

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
200	响应主体将包含所有已知受管代理程序
401	当前请求未经过认证。
403	未向当前用户进行授权。
406	<b>Accept</b> 标头必须设置为受支持的内容类型。
500	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
503	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

```
{
  "type": "array",
  "minItems": 0,
  "uniqueItems": true,
  "items": [
    {
      "type": "object",
      "required": [ "type", "id", "name", "agent_type", "cores", "host", "peer_host",
"uuid",
      "state", "notes", "certificates", "policy", "install_bundle", "snapshots"],
      "additionalProperties": false,
      "properties": {
        "type": { "enum": [ "agent" ] },
        "id": "number",
        "name": "string",
        "agent_type": { "enum": [ "volume", "file", "hybrid", "objstore" ] },
        "cores": "number",
        "host": "string",
        "peer_host": "string",
        "uuid": "string",
        "state": { "enum": [ "ACTIVE", "WAITING", "INACTIVE" ] },
        "notes": "string",
        "certificates": {
          "type": [ "array" ],
          "minItems": 0,
          "uniqueItems": true,
          "items": [
            {
              "type": "object",
              "required": [ "type", "id", "agent_id", "fingerprint", "state",
"notes" ],
```

```

        "additionalProperties": true,
        "properties": {
            "type": { "enum": [ "agent-certificate" ] },
            "id": "number",
            "agent_id": "number",
            "subject": "string",
            "fingerprint": "string",
            "expiry": "number",
            "has_privkey": "boolean",
            "state": { "enum": [ "ACTIVE", "WAITING", "INACTIVE" ] },
            "notes": "string"
        }
    }
}
],
},
"policy": {
    "type": "array",
    "minItems": 0,
    "uniqueItems": true,
    "items": [
        {
            "type": { "enum": [ "volume-policy" ] },
            "id": "number",
            "agent_id": "number",
            "device_label": "string",
            "mofn": { "enum": [ "1:1" ] },
            "key": "string"
        },
        {
            "type": { "enum": [ "file-policy" ] },
            "id": "number",
            "agent_id": "number",
            "path": "string",
            "key": "string",
            "data_type_id": "number",
            "path_set_id": "number",
            "group_set_id": "number",
            "variables": {
                "type": "object",
                "required": [],
                "additionalProperties": true
            }
        }
    ]
}
},
"install_bundle": {
    "type": { "enum": [ "install-bundle" ] },
    "id": "number",
    "agent_id": "number",
    "operating_system": { "enum": [ "centos-6x", "centos-7x", "windows" ] },
    "authorized_users": {
        "type": [ "array" ],
        "minItems": 0,
        "uniqueItems": true,
        "items": [
            { "type": "string" }
        ]
    },
    "supported_formats": {
        "type": [ "array" ],
        "minItems": 0,
        "uniqueItems": true,
        "items": [
            { "type": "string" }
        ]
    }
},
"object_store": {
    "mofn": "2:3", "1:1", "2:4"
    "certificates": {
        "type": [ "array" ],
        "minItems": 0,
        "uniqueItems": true,
        "items": [
            {
                "type": "object",
                "required": [ "type", "id", "agent_id", "fingerprint", "state",
                    "notes" ],
                "additionalProperties": true,
                "properties": {
                    "type": { "enum": [ "agent-certificate" ] },
                    "id": "number",

```

```

        "agent_id": "number",
        "subject": "string",
        "fingerprint": "string",
        "expiry": "number",
        "has_privkey": "boolean",
        "state": { "enum": [ "ACTIVE", "WAITING", "INACTIVE" ] },
        "notes": "string"
    }
}
    ],
    "backends": [{
        "url": "string",
        "key_id": "string",
        "api_key": "string",
        "auth_type": "S3_IBM4", "Se_AMZ4", "SWIFT"
        "share": "number",
    }
],
    "credentials": [{
        "key_id": "string",
        "access_key": "string",
        "auth_type": "S3_IBM4", "Se_AMZ4", "SWIFT"
    }
],
    "buckets": [{
        "name": "string",
        "log_denial": true,
        "policy": [{
            "cred_id": "string",
            "access": "string",
            "log": "boolean",
        }
    ]
}
],
    "policy_snapshots": {
        "type": [ "array" ],
        "minItems": 0,
        "uniqueItems": true,
        "items": [
            {
                "type": "object",
                "required": [ "type", "id", "policy", "state", "notes" ],
                "additionalProperties": true,
                "properties": {
                    "type": { "enum": [ "policy-snapshot" ] },
                    "id": "number",
                    "state": { "enum": [ "ACTIVE", "WAITING", "ACTIVATING", "INACTIVE" ] },
                    "notes": "string",
                    "policy": {
                        "type": "array",
                        "minItems": 0,
                        "uniqueItems": true,
                        "items": [
                            {
                                "type": { "enum": [ "volume-policy" ] },
                                "id": "number",
                                "agent_id": "number",
                                "device_label": "string",
                                "mofn": { "enum": [ "1:1" ] },
                                "key": "string"
                            }
                        ],
                    }
                },
            },
            {
                "type": { "enum": [ "file-policy" ] },
                "id": "number",
                "agent_id": "number",
                "path": "string",
                "key": "string",
                "data_type_id": "number",
                "path_set_id": "number",
                "group_set_id": "number"
            }
        ]
    }
}
    ],
    "storage": {

```

```

        "protocol": "smbv1, nfsv3, nfsv4" ,
        "host": "string",
        "share_path": "string",
        "share_user": "string",
        "share_pass": "string",
        "options": "string",
        "type": "network-share ",
        "features": {
            "type": { "enum": [ "agent_feature" ] },
            "policy": {
                "type": [ "agent_feature_status" ],
                "state": "number" {0, 1},
                "has_feature": "boolean",
            },
            "volume": {
                "type": [ "agent_feature_status" ],
                "state": "number" {0, 1},
                "has_feature": "boolean",
            },
            "su_block": {
                "type": [ "agent_feature_status" ],
                "state": "number" {0, 1},
                "has_feature": "boolean",
            },
            "objstore": {
                "type": [ "agent_feature_status" ],
                "state": "number" {0, 1},
                "has_feature": "boolean",
            }
        },
        "tools": {
            "type": [ "array" ],
            "minItems": 0,
            "uniqueItems": true,
            "items": [
                {
                    "type": { "enum": [ "agent_tool" ] },
                    "tool_id": "number",
                    "purpose": { "enum": [ "BACKUP", "RESTORE", "SEND", "RECEIVE",
                        "OBJSTORE" ] },
                    "key": "string",
                    "active": "boolean",
                },
            ],
        },
    },
},
}
]
}

```

## 示例

请求: **GET /rest/agents**

Response: 200 OK

```

[
  {
    "id": 1,
    "name": "fileAgent",
    "agent_type": "file",
    "cores": 0,
    "host": "1.1.1.1",
    "peer_host": "1.1.1.10",
    "uuid": "9faae641-347c-4f05-b47e-202b7430969d",
    "state": "ACTIVE",
    "notes": "",
    "certificates": [
      {
        "id": 1,
        "agent_id": 1,
        "subject": null,
        "fingerprint": "a15b027baaa1182b32e160dff4955bb2064d2d0b",
        "expiry": null,
        "has_privkey": null,
        "state": "ACTIVE",
        "notes": "",
        "type": "agent-certificate"
      }
    ],
  },
]

```



```

"policy": [
  {
    "id": 1,
    "agent_id": 1,
    "path": "/home/path",
    "key": "homekey",
    "data_type_id": 1,
    "group_set_id": 0,
    "path_set": null,
    "variables": {
      "d1r1c1": "2",
      "d1r1c3": "permit"
    },
    "type": "file-policy"
  },
  {
    "id": 2,
    "agent_id": 1,
    "path": "/data/path",
    "key": "datakey",
    "data_type_id": 1,
    "group_set_id": 0,
    "path_set": null,
    "variables": {
      "d1r1c1": "3",
      "d1r1c3": "deny, log"
    },
    "type": "file-policy"
  }
],
"install_bundle": {
  "id": 1,
  "agent_id": 1,
  "operating_system": "centos-6x",
  "authorized_users": [
    "admin"
  ],
  "supported_formats": [
    "application/x-tar",
    "application/zip",
    "application/json"
  ],
  "type": "install-bundle"
},
"policy_snapshots": [
  {
    "id": 1,
    "notes": "",
    "state": "ACTIVE",
    "policy": [
      {
        "id": 3,
        "agent_id": 1,
        "path": "/home/path",
        "key": "homekey",
        "data_type_id": 2,
        "group_set_id": 0,
        "path_set": null,
        "variables": {},
        "type": "file-policy"
      },
      {
        "id": 4,
        "agent_id": 1,
        "path": "/data/path",
        "key": "datakey",
        "data_type_id": 3,
        "group_set_id": 0,
        "path_set": null,
        "variables": {},
        "type": "file-policy"
      }
    ],
    "type": "policy-snapshot"
  }
],
"features": {
  "policy": {
    "has_feature": true,
    "state": 1,
    "type": "agent_feature_status"
  }
},

```

```

        "volume": {
          "has_feature": false,
          "state": 0,
          "type": "agent_feature_status"
        },
        "su_block": {
          "has_feature": true,
          "state": 1,
          "type": "agent_feature_status"
        },
        "objstore": {
          "has_feature": false,
          "state": 0,
          "type": "agent_feature_status"
        },
        "type": "agent_feature"
      },
      "type": "agent"
    },
    "tools": [
      {
        "tool_id": 1,
        "purpose": "BACKUP",
        "key": "toolKeyA",
        "active": true,
        "type": "agent_tool"
      },
      {
        "tool_id": 2,
        "purpose": "RESTORE",
        "key": "toolKeyA",
        "active": true,
        "type": "agent_tool"
      }
    ]
  }
]
```

POST

查询参数

参数	描述
<无>	

状态码

代码	描述
201	已创建代理程序。 <b>Location</b> 标头将提供新用户的 URL。
202	将在异步作业中创建代理程序。 <b>Location</b> 标头将提供该作业的 URL，以用于状态检查。
400	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
401	当前请求未经过认证。
403	未授权当前代理程序创建新用户
406	<b>Accept</b> 标头必须设置为受支持的内容类型。
409	新代理程序与现有代理程序相冲突。
500	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
503	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

模式

请参阅代理程序集合模式， *agent* 对象。

注意

**uuid** 字段必须只包含十六进制字符， 并采用以连字符分隔的 8-4-4-4-12 格式编写。但是， 不会针对变体和名称空间约束进行更详细的验证。

示例

请求： **POST /rest/agents**

```
{
  "name": "volumeAgent",
  "agent_type": "volume",
  "host": "1.1.1.1",
  "peer_host": "1.1.1.10",
  "uuid": "caca0d57-1f6a-4663-a2c7-e9b9591e48f5",
  "notes": "",
  "certificates": [
    {
      "has_privkey": true,
      "content_type": "application/x-pem-file",
      "subject": "CN=localhost,OU=Development,O=Test.,L=Test,ST=CA,C=US",
      "fingerprint": "a15b027baaa1182b32e160dff4955bb2064d2d0b",
      "name": "http.pem",
      "expiry": 1478733066000,
      "id": 3,
      "type": "client-file",
      "href": "/rest/files/upload/3"
    }
  ],
  "policy": [
    {
      "key": "webkey",
      "type": "volume-policy",
      "device_label": "webroot"
    },
    {
      "key": "nfskey",
      "type": "volume-policy",
      "device_label": "nfsshare"
    }
  ],
  "install_bundle": {
    "operating_system": "centos-7x",
    "authorized_users": ["admin"],
    "type": "install-bundle"
  },
  "policy_snapshots": [],
  "tools": [
    {
      "purpose": "BACKUP",
      "key": "toolKeyA",
      "active": true,
      "type": "agent_tool"
    }
  ]
}
```

Response: 202 Accepted | Location: /jobs/23

**/agents/{id}**

特定受管代理程序

GET

查询参数

参数	描述
----	----

<无>	
-----	--

## 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的代理程序
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 的代理程序，且该代理程序之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅代理程序集合模式，*agent* 对象。

## 示例

请求: **GET /rest/agents/4**

响应: **200 OK**

```
{
  "id": 4,
  "name": "volumeAgent",
  "agent_type": "volume",
  "host": "1.1.1.1",
  "peer_host": "1.1.1.10",
  "uuid": "caca0d57-1f6a-4663-a2c7-e9b9591e48f5",
  "notes": "",
  "certificates": [
    {
      "subject": "CN=localhost,OU=Development,O=Test,L=Test,ST=CA,C=US",
      "fingerprint": "a15b027baaa1182b32e160dff4955bb2064d2d0b",
      "expiry": 1478733066000,
      "id": 6,
      "type": "agent-certificate"
    }
  ],
  "policy": [
    {
      "id": 13,
      "agent_id": 4,
      "key": "webkey",
      "device_label": "webroot",
      "mofn": "1:1",
      "type": "volume-policy"
    },
    {
      "id": 14,
      "agent_id": 4,
      "key": "nfskey",
      "device_label": "nfsshare",
      "mofn": "1:1",
      "type": "volume-policy"
    }
  ],
  "install_bundle": {
    "id": 5,
    "agent_id": 4,
    "operating_system": "centos-7x",
    "authorized_users": ["admin"],
    "supported_formats": [
```

```

        "application/x-tar",
        "application/zip",
        "application/json"
    ],
    "type": "install-bundle"
},
"policy_snapshots": [],
"features": {
    "policy": {
        "has_feature": true,
        "state": 1,
        "type": "agent_feature_status"
    },
    "volume": {
        "has_feature": false,
        "state": 0,
        "type": "agent_feature_status"
    },
    "su_block": {
        "has_feature": true,
        "state": 1,
        "type": "agent_feature_status"
    },
    "objstore": {
        "has_feature": false,
        "state": 0,
        "type": "agent_feature_status"
    },
    "type": "agent_feature"
},
"tools": []
}

```

## DELETE

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>202</b>	已创建作业来处理删除。请查看 <b>Location</b> 标头以获取作业 URL。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 的代理程序，且该代理程序之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

DELETE 中不发送主体

示例

```
Request: DELETE /rest/agents/4
Response: 202 Accepted | Location: /rest/jobs/9
```

PATCH

查询参数

参数	描述
<无>	

状态码

代码	描述
204	请求已成功，但未包含任何更改。不需要刷新客户端表示。
205	请求已成功，且资源已更新。
400	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
401	当前请求未经过认证。
403	未授权当前用户更改代理程序。
404	找不到具有指定 {id} 的代理程序，且该代理程序之前未存在过
406	<b>Accept</b> 标头必须设置为受支持的内容类型。
409	对代理程序所做的更改与另一个代理程序相冲突。
424	未打开足够多的许可证槽，无法核准该代理程序
500	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
503	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

模式

请参阅代理程序集合模式，*agent* 对象。 **更改**：只有 *type* 和 *id* 是必需字段。

示例

```
Request: PATCH /rest/agents/4
{
  "type": "agent",
  "id": 4,
  "state": "ACTIVE"
}
Response: 205 Reset Content
```

[/agents/{id}/certificates](#)

特定受管代理程序的已知证书

## GET

### 查询参数

· 参数	· 描述
· <无>	·

### 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的代理程序证书
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 的代理程序，且该代理程序之前未存在过。对于现存的代理程序， <b>certificates</b> 字段将始终存在，即使该字段为空。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅代理程序集合模式，*agent-certificate* 对象。

### 示例

```
Request: GET /rest/agents/4/certificates
```

```
Response: 200 OK
```

```
[
  {
    "subject": "CN=localhost,OU=Development,O=Test.,L=Test,ST=CA,C=US",
    "fingerprint": "a15b027baaa1182b32e160dff4955bb2064d2d0b",
    "expiry": 1478733066000,
    "id": 6,
    "type": "agent-certificate"
  }
]
```

## POST

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>201</b>	已创建证书。 <b>Location</b> 标头将提供新证书的 URL。

<b>400</b>	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户创建证书。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>409</b>	新证书与现有证书相冲突。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅代理程序集合模式，*agent-certificate* 对象。更改：id 不是必需字段，也不应设置这些字段。

### 示例

```
Request: POST /rest/agents/4/certificates

{
  "id": 4,
  "type": "client-file",
  "href": "/rest/files/upload/4"
  "notes": "After expiration, contact person@example.com for a new one"
}

Response: 201 Created | Location: /rest/agents/4/certificates/7
```

## /agents/{id}/certificates/{certid}

特定受管代理程序证书

.

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的代理程序证书
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 的代理程序，且该代理程序之前未存在过；或者找不到具有指定 {certid} 的证书，且所指定代理程序的该证书之前未存在过。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。



<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅代理程序集合模式，*agent-certificate* 对象。

### 示例

```
Request: GET /rest/agents/4/certificates/7
Response: 200 OK
{
  "id": 7,
  "state": "ACTIVE",
  "subject": "CN=localhost2,OU=Development,O=Test.,L=Test,ST=CA,C=US",
  "fingerprint": "b88b327ba3a3182b32e160dff4955bb2064d2d1f",
  "expiry": 1478733066676,
  "has_privkey": true,
  "notes": "After expiration, contact person@example.com for a new one"
  "type": "agent-certificate"
}
```

## DELETE

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>202</b>	已创建作业来处理删除。请查看 <b>Location</b> 标头以获取作业 URL。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 的代理程序，且该代理程序之前未存在过；或者找不到具有指定 {certid} 的证书，且所指定代理程序的该证书之前未存在过。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

*DELETE* 中不发送主体

## 示例

```
Request: DELETE /rest/agents/4/certificates/7
Response: 202 Accepted | Location: /rest/jobs/33
```

## PATCH

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>204</b>	请求已成功，但未包含任何更改。不需要刷新客户端表示。
<b>205</b>	请求已成功，且资源已更新。
<b>400</b>	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户创建代理程序证书。
<b>404</b>	找不到具有指定 {id} 的代理程序，且该代理程序之前未存在过；或者找不到具有指定 {certid} 的证书，且所指定代理程序的该证书之前未存在过。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>409</b>	对代理程序所做的更改与另一个代理程序相冲突。
<b>424</b>	未打开足够多的许可证槽，无法核准该代理程序
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅代理程序集合模式，*agent* 对象。**更改**：只有 *type* 和 *id* 是必需字段。

### 示例

```
Request: PATCH /rest/agents/4/certificates/7

{
  "type": "agent-certificate",
  "id": 7,
  "state": "ACTIVE",
  "notes": "Expires 2017-04-28,
  contact jack@example.com for a new one. Activated 2016-04-15."
}

Response: 205 Reset Content
```

# PUT

## 查询参数

参数	描述
<无>	

## 状态码

代码	描述
<b>204</b>	请求已成功，但未包含任何更改。不需要刷新客户端表示。
<b>205</b>	请求已成功，且资源已更新。
<b>400</b>	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户创建策略。
<b>404</b>	找不到具有指定 {id} 的代理程序，且该代理程序之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>409</b>	对代理程序所做的更改与另一个代理程序相冲突。
<b>424</b>	未打开足够多的许可证槽，无法核准该代理程序
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅代理程序集合模式，*agent* 对象。

## 示例

```
Request: PUT /rest/agents/4/certificates/1

{
  "id": 7,
  "state": "ACTIVE",
  "subject": "CN=localhost2,OU=Development,O=Test.,L=Test,ST=CA,C=US",
  "fingerprint": "b88b327ba3a3182b32e160dff4955bb2064d2d1f",
  "expiry": 1478733066676,
  "has_privkey": true,
  "notes": " Expires 2017-04-28,
contact jill@example.com for a new one. Activated 2016-04-15."
  "type": "agent-certificate"
}

Response: 205 Reset Content
```

## [/agents/{id}/install\\_bundle](#)

代理程序的安装捆绑软件

代理程序安装捆绑软件是用户受限制的资源。只有该捆绑软件的 `authorized_users` 列表中的用户才能对其进行读取。

GET

查询参数

参数	描述
<无>	

状态码

代码	描述
200	响应主体将包含所请求的代理程序安装捆绑软件或安装捆绑软件元数据，具体取决于发送的 <b>Accept</b> 标头。
401	当前请求未经过认证。
403	未向当前用户进行授权。
404	找不到具有指定 {id} 的代理程序，且该代理程序之前未存在过。对于现存的代理程序， <code>install_bundle</code> 字段将始终存在。
406	<b>Accept</b> 标头必须设置为受支持的内容类型。请发送带有类型 <b>application/json</b> 的 <b>Accept</b> 标头，所返回的 JSON 数据将列出可接受的类型字符串。
500	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
503	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

模式

请参阅代理程序集合模式，`install-bundle` 对象。

示例

```
Request: GET /rest/agents/4/install_bundle
Response: 200 OK
{
  "id": 4,
  "agent_id": 4,
  "operating_system": "centos-7x",
  "authorized_users": ["admin"],
  "supported_formats": [
    "application/x-tar",
    "application/zip",
    "application/json"
  ],
  "expiry": 1478733066000,
  "type": "install-bundle"
}

Request: GET /rest/agents/4/install_bundle | Accept: application/x-tar
Response: 200 OK (binary content - tar file with agent installer and configuration)
```

## /agents/{id}/install\_bundle/authorized\_users

安装捆绑软件授权用户列表

.

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的代理程序安装捆绑软件授权用户列表。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 的代理程序，且该代理程序之前未存在过。对于现存的代理程序，install_bundle 字段以及 install_bundle/authorized_users 字段将始终存在。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

请参阅代理程序集合模式，*install-bundle* 对象的 *authorized\_users* 字段。

#### 示例

```
Request: GET /rest/agents/4/install_bundle/authorized_users
Response: 200 OK
[
  "admin"
]
```

### PUT

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
----	----

<b>204</b>	请求已成功，但未包含任何更改。不需要刷新客户端表示。
<b>205</b>	请求已成功，且资源已更新。
<b>400</b>	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改安装捆绑软件授权用户。
<b>404</b>	找不到具有指定 {id} 的代理程序，且该代理程序之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅代理程序集合模式，*install-bundle* 对象的 *authorized\_users* 字段。

### 示例

```
Request: PUT /rest/agents/4/install_bundle/authorized_users

[
  "admin",
  "wspoppleton",
  "zriggs"
]

Response: 205 Reset Content
```

## [/agents/{id}/policy\\_snapshots](#)

代理程序策略字段的快照。快照是不可变的，在创建快照时其效果将与策略完全相同（请参阅 POST）。快照对象具有新的标识，并且任何设置的变量也可能已直接应用于新对象，但尽管如此在语义上应相同。

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的代理程序证书
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。

<b>404</b>	找不到具有指定 {id} 的代理程序，且该代理程序之前未存在过。对于现存的代理程序， <b>certificates</b> 字段将始终存在，即使该字段为空。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

模式

请参阅代理程序集合模式，*policy-snapshot* 对象。

示例

```
Request: GET /rest/agents/4/policy_snapshots
Response: 200 OK

[
  {
    "id": 1,
    "notes": "This is a file agent for the CentOS7.x file system",
    "state": "ACTIVE",
    "policy": [
      {
        "id": 11,
        "agent_id": 4,
        "path": "/mrslate/misc",
        "key": "public",
        "data_type_id": 6,
        "group_set_id": 0,
        "path_set": null,
        "variables": {},
        "type": "file-policy"
      },
      {
        "id": 12,
        "agent_id": 4,
        "path": "/mrslate/company",
        "key": "company",
        "data_type_id": 7,
        "group_set_id": 0,
        "path_set": null,
        "variables": {},
        "type": "file-policy"
      }
    ],
    "type": "policy-snapshot"
  }
]
```

POST

生成新的策略快照

.

查询参数

参数	描述
<无>	

状态码

代码	描述
----	----

<b>201</b>	已创建快照。 <b>Location</b> 标头将提供新快照的 URL。
<b>400</b>	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户创建快照。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

空对象。

## 示例

```
Request: PUT /rest/agents/4/policy_snapshots/
{}
Response: 201 Created | Location: /rest/agents/4/policy_snapshots/3
```

## [/agents/{id}/policy\\_snapshots/{snapshot\\_id}](#)

代理程序策略字段的快照。快照是不可变的，在创建快照时其效果将与主策略完全相同（请参阅 POST）。快照对象具有新的标识，并且任何设置的变量也可能已直接应用于新对象，但尽管如此在语义上应相同。

## GET

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的代理程序策略快照
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 的代理程序，或者找不到该代理程序的具有给定 {snapshot_id} 的快照，且该代理程序或快照之前未存在过。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。



模式

请参阅代理程序集合模式， *policy-snapshot* 对象。

示例

```
Request: GET /rest/agents/4/policy_snapshots/3
Response: 200 OK
{
  "id": 1,
  "notes": "This is a file agent for the CentOS7.x file system",
  "state": "INACTIVE",
  "policy": [
    {
      "id": 11,
      "agent_id": 4,
      "path": "/mrslate/misc",
      "key": "public",
      "data_type_id": 6,
      "group_set_id": 0,
      "path_set": null,
      "variables": {},
      "type": "file-policy"
    },
    {
      "id": 12,
      "agent_id": 4,
      "path": "/mrslate/company",
      "key": "company",
      "data_type_id": 7,
      "group_set_id": 0,
      "path_set": null,
      "variables": {},
      "type": "file-policy"
    }
  ],
  "type": "policy-snapshot"
}
```

PUT

针对代理程序启动快照激活作业。 在该作业运行之后，在执行 PUT 时处于活动状态的快照将变为不活动状态，而 PUT 的目标快照将变为活动状态。 如果该作业失败，那么旧快照将保持处于活动状态。

查询参数

参数	描述
<无>	

状态码

代码	描述
204	请求已成功，但未包含任何更改。 不需要刷新客户端表示。
205	请求已成功，且资源已更新。
400	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
401	当前请求未经过认证。
403	未授权当前用户激活策略快照。
404	找不到具有指定 {id} 的代理程序，或者找不到快照，且该代理程序或快照之前未存在过

<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

模式

请参阅代理程序集合模式，*install-bundle* 对象的 *policy\_snapshots* 字段。 **更改：** 只有 *state* 和 *notes* 可更改。

示例

```
Request: PUT /rest/agents/4/policy_snapshots/1

{
  "id": 1,
  "notes": "This is a file agent for the CentOS7.x file system",
  "state": "ACTIVE",
  "policy": [
    {
      "id": 11,
      "agent_id": 4,
      "path": "/mrslate/misc",
      "key": "public",
      "data_type_id": 6,
      "group_set_id": 0,
      "path_set": null,
      "variables": {},
      "type": "file-policy"
    },
    {
      "id": 12,
      "agent_id": 4,
      "path": "/mrslate/company",
      "key": "company",
      "data_type_id": 7,
      "group_set_id": 0,
      "path_set": null,
      "variables": {},
      "type": "file-policy"
    }
  ],
  "type": "policy-snapshot"
}

Response: 205 Reset Content
```

DELETE

删除代理程序策略快照

.

查询参数

参数	描述
<无>	

.

状态码

代码	描述
<b>204</b>	删除已成功
<b>401</b>	当前请求未经过认证。

<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 的代理程序或具有给定 {snapshot_id} 的快照，且该代理程序或快照之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

.

**模式**

无。

.

**示例**

Request: GET /rest/agents/4/policy\_snapshots/1  
Response: 204 No Content

**[/agents/{id}/policy](#)**

代理程序保护主策略。这是策略绑定（用于将一个路径或路径集、一组变量字段覆盖或各种不同的策略对象绑定在一起的对象）的列表。

.

**GET**

**查询参数**

参数	描述
<无>	

**状态码**

代码	描述
<b>200</b>	响应主体将包含所请求的代理程序主策略
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 的代理程序。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

模式

请参阅代理程序集合模式， *policy* 字段。

示例

```
Request: GET /rest/agents/4/policy
Response: 200 OK

[
  {
    "id": 13,
    "agent_id": 4,
    "key": "webkey",
    "device_label": "webroot",
    "mofn": "1:1",
    "type": "volume-policy"
  },
  {
    "id": 14,
    "agent_id": 4,
    "key": "nfskey",
    "device_label": "nfsshare",
    "mofn": "1:1",
    "type": "volume-policy"
  }
]

Request: GET /rest/agents/3/policy
Response: 200 OK

[
  {
    "id": 2,
    "agent_id": 3,
    "path": "/mrslate/misc",
    "key": "public",
    "data_type_id": 1,
    "group_set_id": 0,
    "path_set": null,
    "variables": {},
    "type": "file-policy"
  },
  {
    "id": 3,
    "agent_id": 3,
    "path": "/mrslate/company",
    "key": "company",
    "data_type_id": 7,
    "group_set_id": 3,
    "path_set": null,
    "variables": {
      "d7r2c1": "3",
      "d7r2c2": "permit, log",
      "d7r2c3": "read"
    },
    "type": "file-policy"
  }
]
```

POST

生成新的策略绑定

.

查询参数

参数	描述
<无>	

## 状态码

代码	描述
<b>201</b>	已创建绑定。 <b>Location</b> 标头将提供新绑定的 URL。
<b>400</b>	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户创建绑定。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅代理程序集合模式， *file-policy* 和 *volume-policy* 对象。

## 示例

```
Request: POST /rest/agents/3/policy/
```

```
{
  "agent_id": 3,
  "path": "/mrslate/company",
  "key": "company",
  "data_type_id": 7,
  "group_set_id": 3,
  "path_set": null,
  "variables": {
    "d7r2c1": "3",
    "d7r2c2": "permit, log",
    "d7r2c3": "read"
  },
  "type": "file-policy"
}
```

```
Response: 201 Created | Location: /rest/agents/3/policy/3
```

## [/agents/{id}/policy/{binding\\_id}](#)

## GET

### 查询参数

参数	描述
<无>	

## 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的代理程序策略绑定
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。

<b>404</b>	找不到具有指定 {id} 的代理程序，或者该代理程序的主策略中没有具有给定 {binding_id} 的绑定。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅代理程序集合模式，*file-policy* 和 *volume-policy* 对象。

## 示例

Request: GET /rest/agents/3/policy/3

Response: 200 OK

```
{
  "id": 3,
  "agent_id": 3,
  "path": "/mrslate/company",
  "key": "company",
  "data_type_id": 7,
  "group_set_id": 3,
  "path_set": null,
  "variables": {
    "d7r2c1": "3",
    "d7r2c2": "permit, log",
    "d7r2c3": "read"
  },
  "type": "file-policy"
}
```

## PUT

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>204</b>	请求已成功，但未包含任何更改。不需要刷新客户端表示。
<b>205</b>	请求已成功，且资源已更新。
<b>400</b>	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改策略绑定。
<b>404</b>	找不到具有指定 {id} 的代理程序或具有给定 {binding_id} 的绑定，且该代理程序或绑定之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。

<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅代理程序集合模式，*file-policy* 和 *volume-policy* 对象。

## 示例

Request: PUT /rest/agents/3/policy/3

```
{
  "id": 3,
  "agent_id": 3,
  "path": "/mrslate/company",
  "key": "company",
  "data_type_id": 7,
  "group_set_id": 4,
  "path_set": null,
  "variables": {
    "d7r2c1": "4",
    "d7r2c2": "deny, log",
    "d7r2c3": "read"
  },
  "storage": null,
  "type": "file-policy"
}
```

Response: 205 Reset Content

Request: PUT /rest/agents/3/policy/3 (for adding storage object)

```
{
  "id": 3,
  "agent_id": 3,
  "path": "/mrslate/company",
  "key": "company",
  "data_type_id": 7,
  "group_set_id": 4,
  "storage": {
    "protocol": "smvb1",
    "host": "nfs.sfc.local",
    "share_path": "/sfc",
    "share_pass": "Passw0rd",
    "options": "port=112, sec=ext,noexec",
    "type": "network-share"
  },
  "type": "file-policy"
}
```

Response: 205 Reset Content

## PATCH

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>204</b>	请求已成功，但未包含任何更改。不需要刷新客户端表示。
<b>205</b>	请求已成功，且资源已更新。

<b>400</b>	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改策略绑定。
<b>404</b>	找不到具有指定 {id} 的代理程序或具有给定 {binding_id} 的绑定， 且该代理程序或绑定之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅代理程序集合模式， *file-policy* 和 *volume-policy* 对象。

## 示例

Request: PATCH /rest/agents/3/policy/3

```
{
  "group_set_id": 4,
  "variables": {
    "d7r2c1": "4",
    "d7r2c2": "deny, log",
    "d7r2c3": "read"
  },
  "type": "file-policy"
}
```

Response: 205 Reset Content

Request: PATCH /rest/agents/3/policy/3 (to reset the storage )

```
{
  "storage": {
    "id": 1,
    "protocol": "smvb1",
    "host": "nfs.sfc.local",
    "share_path": "/sfc",
    "share_pass": "Passw0rd",
    "options": "port=112, sec=ext,noexec",
    "type": "network-share"
  },
  "type": "file-policy"
}
```

Response: 205 Reset Content

## DELETE

删除代理程序策略绑定

.

## 查询参数

参数	描述
----	----



<无>	
-----	--

#### 状态码

代码	描述
<b>204</b>	删除已成功
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 的代理程序或具有给定 {binding_id} 的绑定，且该代理程序或绑定之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

无。

#### 示例

Request: GET /rest/agents/3/policy/2

Response: 204 No Content

## [/agents/{id}/policy/{binding\\_id}/storage](#)

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的代理程序策略绑定。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 的代理程序，或者该代理程序的主策略中没有具有给定 {binding_id} 的绑定。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。

<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。
------------	---

## 模式

请参阅代理程序集合模式，file-policy 和 storage 对象。

## 示例

```
Request: GET /rest/agents/3/policy/3/storage
Response: 200 OK

{
  "id": 84,
  "protocol": "smvb1",
  "host": "nfs.sfc.local",
  "share_path": "/sfc",
  "share_pass": "Passw0rd",
  "options": "port=112, sec=ext,noexec",
  "type": "network-share"
}
```

## PUT

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>204</b>	请求已成功，但未包含任何更改。 不需要刷新客户端表示。
<b>205</b>	请求已成功，且资源已更新。
<b>400</b>	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改代理程序功能状态。
<b>404</b>	找不到具有指定 {id} 的代理程序、具有给定 {binding_id} 的绑定或具有给定 {storage_id} 的存储器，并且之前未存在过。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅代理程序集合模式，file-policy 和 storage 对象。

## 示例

```
Request: PUT /rest/agents/3/policy/3/storage
```

```
{
  "id": 3,
  "path": "/mrslate/company",
  "key": "company",
  "data_type_id": 7,
  "group_set_id": 4,
  "storage": {
    "protocol": "smvb1",
    "host": "nfs.sfc.local",
    "share_path": "/sfc",
    "share_pass": "Passw0rd",
    "options": "port=112, sec=ext,noexec",
    "type": "network-share"
  },
  "type": "file-policy"
}
```

Response: 205 Reset Content

## PATCH

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>204</b>	请求已成功，但未包含任何更改。不需要刷新客户端表示。
<b>205</b>	请求已成功，且资源已更新。
<b>400</b>	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改代理程序功能状态。
<b>404</b>	找不到具有指定 {id} 的代理程序、具有给定 {binding_id} 的绑定或具有给定 {storage_id} 的存储对象，并且之前未存在过。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅代理程序集合模式，file-policy 和 storage 对象。

### 示例

Request: PATCH /rest/agents/3/policy/3/storage

```
{
  "storage": {
    "id": 1,
    "protocol": "smvb1",
    "host": "nfs.sfc.local",
    "share_path": "/sfc",
    "share_pass": "Passw0rd",
```

```
    "options": "port=112, sec=ext,noexec",
    "type": "network-share"
  },
  "type": "file-policy"
}
Response: 205 Reset Content
```

## /agents/{id}/features

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的代理程序功能。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权
<b>404</b>	找不到具有指定 {id} 的代理程序。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

请参阅代理程序集合模式 *agent\_feature* 对象。

#### 示例

```
Request: GET /rest/agents/1/features
Response: 200 OK
{
  "policy": {
    "has_feature": true,
    "state": 1,
    "type": "agent_feature_status"
  },
  "volume": {
    "has_feature": false,
    "state": 0,
    "type": "agent_feature_status"
  },
  "su_block": {
    "has_feature": true,
    "state": 1,
    "type": "agent_feature_status"
  },
  "objstore": {
    "has_feature": false,
```

```
    "state": 0,
    "type": "agent_feature_status"
  },
  "type": "agent_feature"
}
```

## </agents/{id}/features/{feature}>

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的代理程序功能。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权
<b>404</b>	找不到具有指定 {id} 和 {feature} 的代理程序。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

请参阅代理程序集合模式 *agent\_feature\_status* 对象。

#### 示例

```
Request: GET /rest/agents/1/features/policy

Response: 200 OK
"policy": {
  "has_feature": true,
  "state": 1,
  "type": "agent_feature_status"
},
```

### PUT

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
----	----

<b>202</b>	将在异步作业中更新功能。 <b>Location</b> 标头将提供该作业的 URL，以用于状态检查。
<b>400</b>	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改代理程序功能状态。
<b>404</b>	找不到具有指定 {id} 和 {feature} 的代理程序。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅代理程序集合模式 *agent\_feature\_status* 对象。

## 示例

Request: PUT /rest/agents/3/feature/policy

```
{
  "has_feature": true,
  "state": 0,
  "type": "agent_feature_status"
}
```

Response: 202 Accepted | Location: /rest/jobs/schedulers/9

## PATCH

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>202</b>	将在异步作业中更新功能。 <b>Location</b> 标头将提供该作业的 URL，以用于状态检查。
<b>400</b>	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改策略绑定。
<b>404</b>	找不到具有指定 {id} 和 {feature} 的代理程序。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

模式

请参阅代理程序集合模式 *agent\_feature\_status* 对象。

示例

```
Request: PATCH /rest/agents/3/feature/su_block

{
  "state": 0,
}

Response: 202 Accepted | Location: /rest/jobs/schedulers/9
```

/agents/{id}/tools

GET

查询参数

参数	描述
<无>	

状态码

代码	描述
200	响应主体将包含所请求的代理程序工具
401	当前请求未经过认证。
403	未向当前用户进行授权
404	找不到具有指定 {id} 的代理程序。
406	<b>Accept</b> 标头必须设置为受支持的内容类型。
500	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
503	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

模式

请参阅代理程序集合模式 *agent\_tool* 对象。

示例

```
Request: GET /rest/agents/1/tools

Response: 200 OK
[
  {
    "tool_id": 1,
    "purpose": "BACKUP",
    "key": {
      "id": 3,
      "name": "Normal",
      "autogen": false,
      "scope": "User",
      "notes": "The key used for general encryption",
      "versions": [
        {
          "id": 3,
```

```

        "version": 1,
        "current": true,
        "timestamp": 1523371339398
      }
    ]
  },
  "active": true,
  "type": "agent_tool"
},
{
  "tool_id": 2,
  "purpose": "RESTORE",
  "key": {
    "id": 3,
    "name": "Normal",
    "autogen": false,
    "scope": "User",
    "notes": "The key used for general encryption",
    "versions": [
      {
        "id": 3,
        "version": 1,
        "current": true,
        "timestamp": 1523371339398
      }
    ]
  },
  "active": true,
  "type": "agent_tool"
},
{
  "tool_id": 3,
  "purpose": "OBJSTORE",
  "key": {
    "id": 3,
    "name": "Normal",
    "autogen": false,
    "scope": "User",
    "notes": "The key used for general encryption",
    "versions": [
      {
        "id": 3,
        "version": 1,
        "current": true,
        "timestamp": 1523371339398
      }
    ]
  },
  "active": true,
  "type": "agent_tool"
}
]

```

## POST

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>202</b>	将在异步作业中创建代理程序。 <b>Location</b> 标头将提供该作业的 URL，以用于状态检查。
<b>400</b>	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前代理程序创建新用户



<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>409</b>	新代理程序与现有代理程序相冲突。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅代理程序集合模式 *agent\_tool* 对象。

### 示例

```
Request: POST /rest/agents/1/tools
{
  "tool_id": 1,
  "purpose": "BACKUP",
  "key": "toolKeyB",
  "active": true,
  "type": "agent_tool"
}

Response: 202 Accepted | Location: /rest/jobs/schedulers/9
```

## /agents/{id}/tools/{tool\_id}

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的代理程序工具
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权
<b>404</b>	找不到具有指定 {id} 和 {tool_id} 的代理程序。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅代理程序集合模式 *agent\_tool* 对象。

示例

```
Request: GET /rest/agents/1/tools/1

Response: 200 OK

{
  "tool_id": 1,
  "purpose": "BACKUP",
  "key": {
    "id": 3,
    "name": "Normal",
    "autogen": false,
    "scope": "User",
    "notes": "The key used for general encryption",
    "versions": [
      {
        "id": 3,
        "version": 1,
        "current": true,
        "timestamp": 1523371339398
      }
    ]
  },
  "active": true,
  "type": "agent_tool"
}
```

PUT

查询参数

参数	描述
<无>	

状态码

代码	描述
202	将在异步作业中更新功能。 <b>Location</b> 标头将提供该作业的 URL，以用于状态检查。
400	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
401	当前请求未经过认证。
403	未授权当前用户更改代理程序工具。
404	找不到具有指定 {id} 和 {tool_id} 的代理程序。
406	<b>Accept</b> 标头必须设置为受支持的内容类型。
500	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
503	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

模式

请参阅代理程序集合模式 *agent\_tool* 对象。

示例

```
Request: PUT /rest/agents/86/object_store
{
  {
    "id": 11,
    "mofn": "2:3",
```

```

    "certificate": {
      "id": 45,
      "agent_id": 90,
      "subject": "CN=192.168.1.26",
      "fingerprint":
"432663eaf2ae18e44be41a9185f1b7cd7e902cdbc7006b884a2ec678fcedb9b",
      "expiry": 1606341305000,
      "has_privkey": true,
      "state": "ACTIVE",
      "notes": "",
      "type": "agent-certificate"
    },
    "backends": [
      {
        "id": 25,
        "url": "https://s3.us-south.objectstorage.softlayer.net",
        "key_id": "owerjl78937jkkkj8888",
        "api_key": null,
        "auth_type": "S3_IBM4",
        "share": 1,
        "type": "object-store-backend"
      },
      {
        "id": 26,
        "url": "https://s3.us-south.objectstorage.softlayer.net",
        "key_id": "78937jkkkj9owerjl999",
        "api_key": null,
        "auth_type": "S3_IBM4",
        "share": 2,
        "type": "object-store-backend"
      },
      {
        "id": 27,
        "url": "https://s3.us-south.objectstorage.softlayer.net",
        "key_id": "7jkkkj9ower7893jl999",
        "api_key": null,
        "auth_type": "S3_IBM4",
        "share": 3,
        "type": "object-store-backend"
      }
    ],
    "credentials": [
      {
        "id": 17,
        "key_id": "344Q75QUHMTRYUCN4QT7",
        "access_key": null,
        "auth_type": "S3_IBM4",
        "type": "object-store-credential"
      },
      {
        "id": 18,
        "key_id": "411QJK9Y5450MWALTKY5",
        "access_key": null,
        "auth_type": "S3_IBM4",
        "type": "object-store-credential"
      }
    ],
    "buckets": [
      {
        "id": 17,
        "name": "bucket1",
        "log_denial": false,
        "policy": [
          {
            "id": 33,
            "cred_id": "344Q75QUHMTRYUCN4QT7",
            "access": "RO",
            "log": false,
            "type": "object-store-bucket-policy"
          },
          {
            "id": 34,
            "cred_id": "411QJK9Y5450MWALTKY5",
            "access": "RW",
            "log": true,
            "type": "object-store-bucket-policy"
          }
        ],
        "type": "object-store-bucket"
      },
      {
        "id": 18,

```

```

    "name": "bucket2",
    "log_denial": false,
    "policy": [
      {
        "id": 35,
        "cred_id": "344Q75QUHMTRYUCN4QT7",
        "access": "RW",
        "log": true,
        "type": "object-store-bucket-policy"
      },
      {
        "id": 36,
        "cred_id": "411QJK9Y5450MWALTKY5",
        "access": "WO",
        "log": false,
        "type": "object-store-bucket-policy"
      }
    ],
    "type": "object-store-bucket"
  },
  {
    "source": 0,
    "snapshot": false,
    "snapshotIndex": 0,
    "type": "object-store"
  }
}

```

Response: 202 Accepted | Location: /rest/jobs/schedulers/9

## PATCH

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>202</b>	将在异步作业中更新功能。 <b>Location</b> 标头将提供该作业的 URL，以用于状态检查。
<b>400</b>	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改策略绑定。
<b>404</b>	找不到具有指定 {id} 和 {tool_id} 的代理程序。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅代理程序集合模式 *agent\_tool* 对象。

### 示例

Request: PATCH /rest/agents/1/tools/1

```
{
```

```
    "key": "toolKeyB"
  }
}
Response: 202 Accepted | Location: /rest/jobs/schedulers/9
```

## DELETE

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>202</b>	已创建作业来处理删除。请查看 <b>Location</b> 标头以获取作业 URL。
<b>204</b>	删除已成功
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 和 {tool_id} 的用户，且该用户之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

-

### 示例

```
Request: DELETE /rest/agents/1/tools/1
Response: 202 Accepted | Location: /rest/jobs/schedulers/9
```

## [/agents/{id}/object\\_store](#)

## GET

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的代理程序工具

<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权
<b>404</b>	找不到具有指定 {id} 的代理程序。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅代理程序集合模式，“对象存储”对象。

## 示例

Request: GET /rest/agents/1/object\_store

Response: 200 OK

```
{
  "id": 9,
  "mofn": "2:3",
  "certificate": {
    "id": 42,
    "agent_id": 86,
    "subject": "CN=192.168.1.27",
    "fingerprint": "24d2327171ce69c5080081579f2be4ddc3fc19ca988b8030c430460df17a931b",
    "expiry": 1606341293000,
    "has_privkey": false,
    "state": "ACTIVE",
    "notes": "",
    "type": "agent-certificate"
  },
  "backends": [
    {
      "id": 19,
      "url": "https://s3.us-south.objectstorage.softlayer.net",
      "key_id": "owerjl78937jkjkj8888",
      "api_key": null,
      "auth_type": "S3_IBM4",
      "share": 1,
      "type": "object-store-backend"
    },
    {
      "id": 20,
      "url": "https://s3.us-south.objectstorage.softlayer.net",
      "key_id": "78937jkjkj9owerjl999",
      "api_key": null,
      "auth_type": "S3_IBM4",
      "share": 2,
      "type": "object-store-backend"
    },
    {
      "id": 21,
      "url": "https://s3.us-south.objectstorage.softlayer.net",
      "key_id": "7jkjkj9ower7893jl999",
      "api_key": null,
      "auth_type": "S3_IBM4",
      "share": 3,
      "type": "object-store-backend"
    }
  ],
  "credentials": [
    {
      "id": 13,
      "key_id": "344Q75QUHMTRYUCN4QT7",
      "access_key": null,
      "auth_type": "S3_IBM4",
      "type": "object-store-credential"
    },
    {
      "id": 14,
```

```

        "key_id": "411QJK9Y5450MWALTKY5",
        "access_key": null,
        "auth_type": "S3_IBM4",
        "type": "object-store-credential"
    },
    "buckets": [
        {
            "id": 13,
            "name": "bucket1",
            "log_denial": true,
            "policy": [
                {
                    "id": 25,
                    "cred_id": "344Q75QUHMTRYUCN4QT7",
                    "access": "RO",
                    "log": false,
                    "type": "object-store-bucket-policy"
                },
                {
                    "id": 26,
                    "cred_id": "411QJK9Y5450MWALTKY5",
                    "access": "RW",
                    "log": true,
                    "type": "object-store-bucket-policy"
                }
            ],
            "type": "object-store-bucket"
        },
        {
            "id": 14,
            "name": "bucket2",
            "log_denial": false,
            "policy": [
                {
                    "id": 27,
                    "cred_id": "344Q75QUHMTRYUCN4QT7",
                    "access": "RW",
                    "log": true,
                    "type": "object-store-bucket-policy"
                },
                {
                    "id": 28,
                    "cred_id": "411QJK9Y5450MWALTKY5",
                    "access": "WO",
                    "log": false,
                    "type": "object-store-bucket-policy"
                }
            ],
            "type": "object-store-bucket"
        }
    ],
    "source": 0,
    "snapshot": false,
    "snapshotIndex": 0,
    "type": "object-store"
}

```

## PUT

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的代理程序工具
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权

<b>404</b>	找不到具有指定 {id} 的代理程序。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅代理程序集合模式，“对象存储”对象。

## 示例

```
Request: PUT /rest/agents/86/object_store
{
  {
    "id": 11,
    "mofn": "2:3",
    "certificate": {
      "id": 45,
      "agent_id": 90,
      "subject": "CN=192.168.1.26",
      "fingerprint":
"432663eaf2ae18e44be41a9185f1b7cd7e902cdbc7006b884a2ec678fcedb9b",
      "expiry": 1606341305000,
      "has_privkey": true,
      "state": "ACTIVE",
      "notes": "",
      "type": "agent-certificate"
    },
    "backends": [
      {
        "id": 25,
        "url": "https://s3.us-south.objectstorage.softlayer.net",
        "key_id": "owerjl78937jkjkj8888",
        "api_key": null,
        "auth_type": "S3_IBM4",
        "share": 1,
        "type": "object-store-backend"
      },
      {
        "id": 26,
        "url": "https://s3.us-south.objectstorage.softlayer.net",
        "key_id": "78937jkjkj9owerjl999",
        "api_key": null,
        "auth_type": "S3_IBM4",
        "share": 2,
        "type": "object-store-backend"
      },
      {
        "id": 27,
        "url": "https://s3.us-south.objectstorage.softlayer.net",
        "key_id": "7jkjkj9ower7893jl999",
        "api_key": null,
        "auth_type": "S3_IBM4",
        "share": 3,
        "type": "object-store-backend"
      }
    ],
    "credentials": [
      {
        "id": 17,
        "key_id": "344Q75QUHMYUCN4QT7",
        "access_key": null,
        "auth_type": "S3_IBM4",
        "type": "object-store-credential"
      },
      {
        "id": 18,
        "key_id": "411QJK9Y5450MWALTKY5",
        "access_key": null,
        "auth_type": "S3_IBM4",
        "type": "object-store-credential"
      }
    ]
  }
}
```



```

    ],
    "buckets": [
      {
        "id": 17,
        "name": "bucket1",
        "log_denial": false,
        "policy": [
          {
            "id": 33,
            "cred_id": "344Q75QUHMTRYUCN4QT7",
            "access": "R0",
            "log": false,
            "type": "object-store-bucket-policy"
          },
          {
            "id": 34,
            "cred_id": "411QJK9Y5450MWALTKY5",
            "access": "RW",
            "log": true,
            "type": "object-store-bucket-policy"
          }
        ],
        "type": "object-store-bucket"
      },
      {
        "id": 18,
        "name": "bucket2",
        "log_denial": false,
        "policy": [
          {
            "id": 35,
            "cred_id": "344Q75QUHMTRYUCN4QT7",
            "access": "RW",
            "log": true,
            "type": "object-store-bucket-policy"
          },
          {
            "id": 36,
            "cred_id": "411QJK9Y5450MWALTKY5",
            "access": "W0",
            "log": false,
            "type": "object-store-bucket-policy"
          }
        ],
        "type": "object-store-bucket"
      }
    ],
    "source": 0,
    "snapshot": false,
    "snapshotIndex": 0,
    "type": "object-store"
  }
}

```

Response: 202 Accepted | Location: /rest/jobs/schedulers/9

## PATCH

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>202</b>	将在异步作业中更新功能。 <b>Location</b> 标头将提供该作业的 URL，以用于状态检查。
<b>400</b>	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。

<b>403</b>	未授权当前用户更改策略绑定。
<b>404</b>	找不到具有指定 {id} 的代理程序。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅代理程序集合模式，“对象存储”对象。

### 示例

```
Request: PATCH /rest/agents/1/object_store
{
  "id" : 11,
  "credentials" : [{
    "id" : 17,
    "key_id" : "244Q75QUHMTRYUCN4QT7"
  }],
  "type" : "object-store"
}

Response: 202 Accepted | Location: /rest/jobs/schedulers/9
```

## /agents/{id}/object\_store/credentials

POST

### 查询参数

参数	描述
<none>	

### 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的代理程序工具
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 的代理程序。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅代理程序集合模式，*object\_store* 对象下的凭证对象

## 示例

```
Request: POST /rest/agents/1/object_store/credentials

{
  "key_id": "344Q75QUHMTRYUCN4QT7",
  "access_key": "hdc1Im0t2kQ7xxchmRkGxZyuGfG10notActualKey",
  "auth_type": "S3_IAM4"
}

Response: 202 Accepted | Location: /rest/jobs/schedulers/9
```

## DELETE 查询参数

参数	描述
<none>	

## 状态码

代码	描述
<b>202</b>	将在异步作业中更新功能。 <b>Location</b> 标头将提供该作业的 URL，以用于状态检查。
<b>400</b>	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改代理程序工具。
<b>404</b>	找不到具有指定 {id} 和 {cid} 的代理程序。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅代理程序集合模式， *object\_store* 下的凭证对象。

## 示例

```
Request: DELETE /rest/agents/86/object_store/credentials/<cid>

Response: 202 Accepted | Location: /rest/jobs/schedulers/9
```

## [/agents/{id}/object\\_store/buckets](#)

### GET

#### GET 查询参数

参数	描述
<none>	

## 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的代理程序工具
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 并且存在对象存储和存储区的代理程序。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅代理程序集合模式，“对象存储”代理程序对象和存储区对象。

## 示例

```
Request: GET /rest/agents/1/object_store/buckets

Response: 200 OK
[
  {
    "id": 17,
    "name": "bucket1",
    "log_denial": true,
    "policy": [
      {
        "id": 33,
        "cred_id": "344Q75QUHMTRYUCN4QT7",
        "access": "R0",
        "log": false,
        "type": "object-store-bucket-policy"
      },
      {
        "id": 34,
        "cred_id": "411QJK9Y5450MWALTKY5",
        "access": "RW",
        "log": true,
        "type": "object-store-bucket-policy"
      }
    ],
    "type": "object-store-bucket"
  },
  {
    "id": 18,
    "name": "bucket2",
    "log_denial": false,
    "policy": [
      {
        "id": 35,
        "cred_id": "344Q75QUHMTRYUCN4QT7",
        "access": "RW",
        "log": true,
        "type": "object-store-bucket-policy"
      },
      {
        "id": 36,
        "cred_id": "411QJK9Y5450MWALTKY5",
        "access": "W0",
        "log": false,
        "type": "object-store-bucket-policy"
      }
    ],
    "type": "object-store-bucket"
  }
]
```

## POST

### 查询参数

参数	描述
<none>	

### 状态码

代码	描述
200	响应主体将包含所请求的代理程序工具
401	当前请求未经过认证。
403	未向当前用户进行授权。
404	找不到具有指定 {id} } 和对象存储的代理程序。
406	<b>Accept</b> 标头必须设置为受支持的内容类型。
500	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
503	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅代理程序集合模式，*object\_store* 对象下的存储区对象

### 示例

Request: POST /rest/agents/1/object\_store/buckets

```
{
  "name": "bucket12",
  "log_denial": true,
  "policy": [
    {
      "cred_id": "344Q75QUHMTRYUCN4QT7",
      "access": "R0",
      "log": false,
      "type": "object-store-bucket-policy"
    },
    {
      "cred_id": "411QJK9Y5450MWALTKY5",
      "access": "RW",
      "log": true,
      "type": "object-store-bucket-policy"
    }
  ],
  "type": "object-store-bucket"
}
```

Response: 202 Accepted | Location: /rest/jobs/schedulers/9

## DELETE

### 查询参数

参数	描述
<none>	

### 状态码

代码	描述
202	将在异步作业中更新功能。 <b>Location</b> 标头将提供该作业的 URL，以用于状态检查。

<b>400</b>	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改代理程序工具。
<b>404</b>	找不到具有指定 {id} 和/或 {bucket id} 的代理程序。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅代理程序集合模式， *object\_store* 对象下的存储区对象

## 示例

```
Request: DELETE /rest/agents/86/object_store/buckets/<bid>
Response: 202 Accepted | Location: /rest/jobs/schedulers/9
```

## [/agents/{id}/object\\_store/buckets/<bid>/policy](#)

### POST

#### 查询参数

参数	描述
<b>&lt;none&gt;</b>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的代理程序工具
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 和/或 {bucket_id} 的代理程序。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅代理程序集合模式， *object\_store* 对象中存储区对象下的策略对象

## 示例

```
Request: POST /rest/agents/1/object_store/buckets/<bid>/policy
{
```

```
{
  "cred_id": "344Q75QUHMTRYUCN4QT7",
  "access": "R0",
  "log": false,
  "type": "object-store-bucket-policy"
}
```

## DELETE

### 查询参数

参数	描述
<none>	

### 状态码

代码	描述
<b>202</b>	将在异步作业中更新功能。 <b>Location</b> 标头将提供该作业的 URL，以用于状态检查。
<b>400</b>	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改代理程序工具。
<b>404</b>	找不到具有指定 {id} 和/或 {bucket_id} 和/或 {policy_id} 的代理程序。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅代理程序集合模式， *object\_store* 对象中存储区对象下的策略对象

### 示例

```
Request: DELETE /rest/agents/86/object_store/buckets/<bid>/policy/<pid>
Response: 202 Accepted | Location: /rest/jobs/schedulers/9
```

## /agents/{id}/policy/<pid>

特定受管代理程序的已知证书

## GET

### 查询参数

参数	描述
<none>	

### 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的代理程序证书

<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 的代理程序，且该代理程序之前未存在过。对于现存的代理程序， <b>certificates</b> 字段将始终存在，即使该字段为空。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅代理程序集合模式，*agent-certificate* 对象。

## 示例

```
Request: GET /rest/agents/4/certificates
Response: 200 OK
[
  {
    "subject": "CN=localhost,OU=Development,O=Test.,L=Test,ST=CA,C=US",
    "fingerprint": "a15b027baaa1182b32e160dff4955bb2064d2d0b",
    "expiry": 1478733066000,
    "id": 6,
    "type": "agent-certificate"
  }
]
```

## POST

### 查询参数

参数	描述
<b>&lt;none&gt;</b>	

## 状态码

代码	描述
<b>201</b>	已创建证书。 <b>Location</b> 标头将提供新证书的 URL。
<b>400</b>	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户创建证书。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>409</b>	新证书与现有证书相冲突。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式



请参阅代理程序集合模式，*agent-certificate* 对象。更改：*id* 不是必需字段，也不应设置这些字段。

### 示例

Request: POST /rest/agents/4/certificates

```
{
  "id": 4,
  "type": "client-file",
  "href": "/rest/files/upload/4"
  "notes": "After expiration, contact person@example.com for a new one"
}
```

Response: 201 Created | Location: /rest/agents/4/certificates/7



# 第 9 章 用户帐户

## /users

所有服务器级别的用户

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
200	响应主体将包含所有已知用户
401	当前请求未经过认证。
403	未向当前用户进行授权。
406	<b>Accept</b> 标头必须设置为受支持的内容类型。
500	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
503	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

```
{
  "type": "array",
  "minItems": 1,
  "uniqueItems": true,
  "items": [
    {
      "type": "object",
      "required": [ "type", "username",
"password", "password_confirm", "state", "roles", "properties"],
      "additionalProperties": false,
      "properties": {
        "type": { "enum": [ "user" ] },
        "username": "number",
        "password": "string",
        "password_confirm": "string",
        "state": { "enum": [ "enabled",
"disabled", "expired", "locked" ] },
        "roles": {
          "type": "array",
          "minItems": 0,
          "uniqueItems": true,
          "items": [
            { "type": "string" }
          ]
        },
        "properties": {
          "type": "object",
          "required": [ "notes", "passwordLastModified"],
          "additionalProperties": true,
          "properties": {
            "notes": "string",
```

```
        "passwordLastModified": "number",
      }
    }
  }
}
```

### 示例

```
Request: GET /rest/users
Response: 200 OK

[
  {
    "type": "user",
    "username": "admin",
    "password": null,
    "password_confirm": null,
    "state": "enabled",
    "roles": ["product-administrator", "admin"],
    "properties": {
      "notes": "Paul's user account",
      "passwordLastModified": 1460911763077
    }
  }
]
```

## POST

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>201</b>	已创建用户。 <b>Location</b> 标头将提供新用户的 URL。
<b>202</b>	将在异步作业中创建用户。 <b>Location</b> 标头将提供该作业的 URL，以用于状态检查。
<b>400</b>	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户创建新用户
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>409</b>	新用户与现有用户相冲突。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅用户集合模式， *user* 对象。 **更改：** 不应设置 *properties.passwordLastModified*

示例

```
Request: POST /rest/users

{
  "type": "user",
  "username": "admin-joe2",
  "password": "MyPassword12345",
  "password_confirm": "MyPassword12345",
  "state": "enabled",
  "roles": ["policy-server-administrator"],
  "properties": {
    "notes": "By management request, a second user account for Joe.",
  }
}

Response: 202 Accepted | Location: /jobs/67
```

[/users/{username}](#)

特定服务器用户

GET

查询参数

参数	描述
<无>	

状态码

代码	描述
200	响应主体将包含所请求的用户
401	当前请求未经过认证。
403	未向当前用户进行授权。
404	找不到具有指定 {username} 的用户，且该用户之前未存在过
406	<b>Accept</b> 标头必须设置为受支持的内容类型。
500	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
503	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

模式

请参阅用户集合模式，*user* 对象。

示例

```
Request: GET /rest/users/admin

Response: 200 OK

{
  "type": "user",
  "username": "admin",
  "password": null,
  "password_confirm": null,
  "state": "enabled",
  "roles": ["policy-server-administrator", "admin"],
}
```

```
"properties": {  
  "notes": "",  
  "passwordLastModified": 1460911763077  
}
```

## DELETE

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>202</b>	已创建作业来处理删除。请查看 <b>Location</b> 标头以获取作业 URL。
<b>204</b>	删除已成功
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {username} 的用户，且该用户之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

DELETE 中不发送主体

### 示例

Request: DELETE /rest/users/admin

Response: 202 Accepted | Location: /rest/jobs/5

## PATCH

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>202</b>	已创建作业来处理更改。请查看 <b>Location</b> 标头以获取作业 URL。
<b>204</b>	请求已成功，但未包含任何更改。不需要刷新客户端表示。

<b>205</b>	请求已成功，且资源已更新。
<b>400</b>	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改用户。
<b>404</b>	找不到具有指定 {username} 的用户，且该用户之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>409</b>	对用户所做的更改与另一个用户相冲突。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅用户集合模式，*user* 对象。 **更改**：只有 *type* 和 *username* 是必需字段。

## 示例

```
Request: PATCH /rest/users/admin

{
  "properties": {
    "notes": "updated notes for Paul",
  },
  "type": "user"
}

Response: 205 Reset Content
```

## PUT

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>202</b>	已创建作业来处理更改。请查看 <b>Location</b> 标头以获取作业 URL。
<b>204</b>	请求已成功，但未包含任何更改。不需要刷新客户端表示。
<b>205</b>	请求已成功，且资源已更新。
<b>400</b>	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改用户。

<b>404</b>	找不到具有指定 {username} 的用户，且该用户之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>409</b>	对用户所做的更改与另一个用户相冲突。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅用户集合模式，*user* 对象。

### 示例

```
Request: PUT /rest/users/admin

{
  "type": "user",
  "username": "admin",
  "password": null,
  "password_confirm": null,
  "state": "disabled",
  "roles": ["product-administrator", "admin"],
  "properties": {
    "notes": " Disabled 2016-04-15 due to temporary reassignment.",
    "passwordLastModified": 1460911763077
  }
}

Response: 205 Reset Content
```

## /roles

所有服务器用户角色

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含所有已知角色
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。



## 模式

```
{
  "type": "array",
  "minItems": 2,
  "uniqueItems": true,
  "items": [
    {
      "type": "object",
      "required": [ "type", "id", "permissions" ],
      "additionalProperties": false,
      "properties": {
        "type": { "enum": [ "rbac-role" ] },
        "id": "string",
        "permissions": {
          "type": "array",
          "minItems": 1,
          "uniqueItems": true,
          "items": [
            {
              "type": "object",
              "required": [ "type", "id", "actions" ],
              "additionalProperties": false,
              "properties": {
                "type": { "enum": [ "rbac-permission" ] },
                "id": "string",
                "actions": {
                  "type": "array",
                  "minItems": 1,
                  "uniqueItems": true,
                  "items": [
                    { "type": "string" }
                  ]
                }
              }
            }
          ]
        }
      }
    }
  ]
}
```

## 示例

Request: GET /rest/roles

Response: 200 OK

```
[
  {
    "type": "rbac-role",
    "id": "product-administrator",
    "permissions": [
      {
        "id": "see-keystore-settings",
        "actions": [
          "see-keystore-settings"
        ],
        "type": "rbac-permission"
      },
      {
        "id": "see-licenses",
        "actions": [
          "see-licenses"
        ],
        "type": "rbac-permission"
      }
    ]
  }
]
```

### [/roles/{id}](#)

特定服务器用户角色

# GET

## 查询参数

参数	描述
<无>	

## 状态码

代码	描述
<b>200</b>	响应主体将包含所请求的角色
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 的角色，且该角色之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅角色集合模式，*rbac-role* 对象。

## 示例

Request: GET /rest/roles/product-administrator

Response: 200 OK

```
{
  "type": "rbac-role",
  "id": "product-administrator",
  "permissions": [
    {
      "id": "see-keystore-settings",
      "actions": [
        "see-keystore-settings"
      ],
      "type": "rbac-permission"
    },
    {
      "id": "see-licenses",
      "actions": [
        "see-licenses"
      ],
      "type": "rbac-permission"
    }
  ]
}
```

## /sessions

所有服务器用户会话

# GET

## 查询参数

参数	描述
<无>	

## 状态码

代码	描述
<b>200</b>	响应主体将包含所有活动的用户会话
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

```
{
  "type": "array",
  "minItems": 1,
  "uniqueItems": true,
  "items": [
    {
      "type": "object",
      "required": [ "type", "id",
"user", "roles", "activated_permissions", "when_started"],
      "additionalProperties": false,
      "properties": {
        "type": { "enum": [ "web-session" ] },
        "id": "string",
        "user": "string",
        "when_started": "number",
        "roles": {
          "type": "array",
          "minItems": 0,
          "uniqueItems": true,
          "items": [
            { "type": "string" }
          ]
        },
        "activated_permissions": {
          "type": "array",
          "minItems": 0,
          "uniqueItems": true,
          "items": [
            { "type": "string" }
          ]
        }
      }
    }
  ]
}
```

## 示例

```
Request: GET /rest/sessions

Response: 200 OK

[
  {
    "type": "web-session",
```

```

    "id": "1d3eldnegio001ksj59gedjln5",
    "user": "admin",
    "when_started": 1460911832224,
    "roles": [
      "admin",
      "product-administrator"
    ],
    "activated_permissions": [
      "add-product-administrator-role",
      "approve-agents",
      "change-agents",
      "change-own-user"
    ]
  }
]

```

## 注意

目前将只公开当前已认证的用户会话。

## POST

针对此资源的此请求方法用于认证新的 HTTP 会话并将其启动。

### 支持的内容类型

内容类型	描述
<b>application/x-www-form-urlencoded</b>	在第一个发行版中用于表单认证和编程认证。

### 状态码

代码	描述
<b>201</b>	会话已启动。 在响应的 Location 标头中将提供当前会话资源 URL
<b>401</b>	凭证无效、主体的格式不正确，或其他问题 - 有意模糊处理，以便在成为恶意活动的目标时减少信息披露

### 模式

```

{
  "type": "string",
  "description":
    "URL encoded parameter list of j_username,
    j_password, and optionally j_directory with
    values set to the username, password, and
    directory respectively"
}

```

### 示例

```

Request: POST /rest/sessions
j_username=admin&j_password=MyPass12345
Response: 201 Created | Location: /rest/sessions/i1et8oc714vek1534fso3uw9

Request: POST /rest/sessions
j_username=admin&j_password=MyPass12345&j_directory=IT_LDAP
Response: 201 Created | Location: /rest/sessions/1a2ascg8qi7dr16pue3nvvv4dz

```

注意

要使 *POST* 被接受，*CSRF* 令牌必须有效。因此，必须先启动 *HTTP* 会话（通常是通过向另一个资源发出 *GET* 请求，*/rest/sessions* 是一个特别有用的选项），然后才能通过向会话集合发出 *POST* 请求来启动已认证的会话。

建议的流程是，对 */rest/sessions* 执行 *GET*，检查响应码，并且只有在响应码为 *401* 时才对 */rest/sessions* 执行 *POST*。

[/sessions/{id}](#)

特定服务器会话。

GET

查询参数

参数	描述
<无>	

状态码

代码	描述
200	响应主体将包含所请求的会话。
401	当前请求未经过认证。
403	未向当前用户进行授权。
404	找不到具有指定 {id} 的会话，且该会话之前未存在过
406	<b>Accept</b> 标头必须设置为受支持的内容类型。
500	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
503	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

模式

请参阅会话集合模式，*sessions* 对象。

示例

```
Request: GET /rest/sessions/1d3e1dnegio001ksj59gedjln5
Response: 200 OK
{
  "type": "web-session",
  "id": "1d3e1dnegio001ksj59gedjln5",
  "user": "admin",
  "when_started": 1460911832224,
  "roles": [
    "admin",
    "product-administrator"
  ],
  "activated_permissions": [
    "add-product-administrator-role",
    "approve-agents",
    "change-agents",
    "change-own-user"
  ]
}
```

# DELETE

## 查询参数

参数	描述
<无>	

## 状态码

代码	描述
<b>204</b>	删除已成功，这意味着目标会话不再有效
<b>401</b>	当前请求未经过认证。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

*DELETE* 中不发送主体

## 示例

Request: DELETE /rest/sessions/i1et8oc714vek1534fso3uw9

Response: 204 No content

# 第 10 章 服务器设置

## /settings

所有已知日志的集合

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
200	响应主体将包含对子设置资源的引用的集合。这些引用将不包含任何设置本身。
401	当前请求未经过认证。
403	未向当前用户进行授权。
406	<b>Accept</b> 标头必须设置为受支持的内容类型。
500	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
503	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

```
{
  "type": "array",
  "minItems": 0,
  "uniqueItems": true,
  "items": [
    {
      "type": "object",
      "required": ["type", "id", "href"],
      "additionalProperties": true,
      "properties": {
        "type": { "enum": [ "ref" ] },
        "id": { "enum": [ "directories", "keystores" ] },
        "href": "string"
      }
    }
  ]
}
```

#### 示例

```
Request: GET /settings

Response: 200 OK

[
  {
    "type": "ref",
    "id": "directories",
```

```

    "href": "/rest/settings/directories",
  },
  {
    "type": "ref",
    "id": "keystores",
    "href": "/rest/settings/keystores",
  }
]

```

## /settings/directories

服务器 LDAP 目录配置

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含产品密钥库设置
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

```

{
  "type": "array",
  "minItems": 0,
  "uniqueItems": true,
  "items": [
    {
      "type": "object",
      "required": ["type", "id", "host", "port", "binddn", "secure"],
      "additionalProperties": false,
      "properties": {
        "type": { "enum": [ "directory" ] },
        "id": "string",
        "host": "string",
        "port": "number",
        "binddn": "string",
        "directory_type": { "enum": [ "LDAP", "ACTIVE_DIRECTORY" ] },
        "secure": "boolean"
      }
    }
  ]
}

```

#### 注意

当 **directory\_type** 设置为 **ACTIVE\_DIRECTORY** 时，**binddn** 字段将被忽略。在这种情况下，建议调用者将 **binddn** 设置为 **null** 或空字符串以避免任何混淆。



示例

```
Request: GET /rest/settings/directories

Response: 200 OK

[
  {
    "id": "human-resources",
    "host": "192.168.4.114",
    "port": 389,
    "host": "LDAP",
    "binddn": "uid={username},cn=humanresources,ou=Groups,dc=example,dc=com",
    "secure": false,
    "type": "directory"
  }
]
```

POST

查询参数

参数	描述
<无>	

状态码

代码	描述
201	已创建用于创建新目录的作业。 <b>Location</b> 标头将提供新作业的 URL。
400	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
401	当前请求未经过认证。
403	未授权当前用户创建目录。
406	<b>Accept</b> 标头必须设置为受支持的内容类型。
409	新目录与现有目录标识相冲突。
500	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
503	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

模式

请参阅目录集合模式， *directory* 对象。 **更改**: *id* 不是必需字段，也不应为任一对象类型设置该字段。

示例

```
Request: POST /rest/settings/directories

{
  "id": "human-resources",
  "host": "192.168.4.114",
  "port": 389,
  "host": "LDAP",
  "binddn": "uid={username},cn=humanresources,ou=Groups,dc=example,dc=com",
  "secure": false,
  "type": "directory"
}

Response: 201 Created | Location: /rest/jobs/1
```

## /settings/directories/{id}

特定 LDAP 目录配置

### GET

#### 查询参数

#### 状态码

代码	描述
<b>200</b>	响应主体将包含特定接口的目录设置
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

请参阅目录设置集合模式，*directory* 对象。

#### 示例

Request: GET /rest/settings/directory/human-resources

Response: 200 OK

```
{
  "id": "human-resources",
  "host": "192.168.4.114",
  "port": 389,
  "host": "LDAP",
  "binddn": "uid={username},cn=humanresources,ou=Groups,dc=example,dc=com",
  "secure": false,
  "type": "directory"
}
```

### DELETE

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>204</b>	删除已成功
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 的目录，且该目录之前未存在过

<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

DELETE 中不发送主体

## 示例

```
Request: DELETE /rest/settings/directories/human-resources
Response: 204 No content
```

# PUT

## 查询参数

参数	描述
<无>	

## 状态码

代码	描述
<b>204</b>	请求已成功，但未包含任何更改。不需要刷新客户端表示。
<b>205</b>	请求已成功，且资源已更新。
<b>400</b>	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改进程集
<b>404</b>	找不到具有指定 {id} 的目录，且该目录之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>409</b>	对目录所做的更改与另一个目录相冲突。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅目录集合模式，*directory* 对象。

## 示例

```
Request: PUT /rest/settings/directories/tenant1
{
  "id": "human-resources",
  "host": "192.168.2.134",
  "port": 389,
  "host": "LDAP",
```

```
"binddn": "uid={${username}},cn=humanresources,ou=Groups,dc=example,dc=com",
"secure": false,
"type": "directory"
}
```

Response: 205 Reset content

## /settings/keystores

服务器密钥库配置

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含产品密钥库设置
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

```
{
  "type": "array",
  "minItems": 0,
  "uniqueItems": true,
  "items": [
    {
      "type": "object",
      "required": ["type", "id", "state", "name",
"product", "host", "port", "keystore", "keystore_alias",
"truststore", "truststore_alias", "truststore_password", "master_keystore_id" ],
      "additionalProperties": false,
      "properties": {
        "type": { "enum": [ "keystore-kmip", "keystore-hsm" ] },
        "id": "number",
        "name": "string",
        "state": { "enum": [ "ACTIVE", "INACTIVE" ] },
        "product": { "enum": [ "IBM_SKLM", "HSM" ] },
        "host": "string",
        "port": "number",
        "keystore": {
          "type": "object",
          "required": [ "type", "id", "name", "href"],
          "additionalProperties": false,
          "properties": {
            "type": { "enum": [ "ref" ] },
            "id": "number",
            "name": "string",
            "href": "string"
          },
          "keystore_alias": "string",
```

```

        "keystore_password": "string",
        "truststore": {
            "type": "object",
            "required": [ "type", "id", "name", "href"],
            "additionalProperties": false,
            "properties": {
                "type": { "enum": [ "ref" ] },
                "id": "number",
                "name": "string",
                "href": "string"
            },
            "truststore_alias": "string",
            "truststore_password": "string",
            "hsm_password": "string",
            "hsm_token": "string",
            "key_handle": "string",
            "master_keystore_id": "number"
        }
    }
}
]
}

```

### 示例

Request: GET /rest/settings/keystores

Response: 200 OK

```

[
  {
    "id": 1,
    "name": "builtin",
    "state": "INACTIVE",
    "product": "IBM_SKLM",
    "host": null,
    "port": 0,
    "keystore": null,
    "keystore_alias": null,
    "keystore_password": "*****",
    "truststore": null,
    "truststore_alias": null,
    "truststore_password": "*****",
    "master_keystore_id": 0,
    "type": "keystore-kmip"
  },
  {
    "id": 1,
    "name": "hsm",
    "state": "INACTIVE",
    "product": "HSM ",
    "hsm_password": null,
    "hsm_token": "0",
    "key_handle": "441",
    "master_keystore_id": 0,
    "type": "keystore-hsm"
  }
]

```

## POST

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
201	已创建用于创建新密钥库的作业。 <b>Location</b> 标头将提供新作业的 URL。

<b>400</b>	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户创建目录。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>409</b>	新目录与现有目录标识相冲突。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅密钥库集合模式，*keystore* 对象。 **更改：** *id* 不是必需字段，也不应设置该字段。

## 示例

```

KMIP Keystores:
Request: POST /rest/settings/keystores

{
  "name": "standby",
  "state": "INACTIVE",
  "product": "IBM_SKLM",
  "host": "192.168.4.133",
  "port": 5696,
  "keystore": {
    "name": "sklm-client-keystore.jks",
    "id": 42,
    "type": "ref",
    "href": "/rest/files/upload/42"
  },
  "keystore_alias": "sklmclient",
  "keystore_password": "keystorePa55word123",
  "truststore": {
    "name": "sklm-truststore.jks",
    "id": 43,
    "type": "ref",
    "href": "/rest/files/upload/43"
  },
  "keystore_alias": "sklmserver",
  "keystore_password": "trustp@ss123",
  "master_keystore_id": 1,
  "type": "keystore-kmip"
}

Response: 202 Accepted | Location: /rest/jobs/112

HSM keystores:
Request: POST /rest/settings/keystores

{
  "name": "hsm",
  "state": "INACTIVE",
  "product": "HSM",
  "hsm_password": null,
  "hsm_token": "0",
  "keystore_handle": "441",
  "master_keystore_id": 0,
  "type": "keystore-hsm"
}

Response: 202 Accepted | Location /rest/jobs/112

```

## /settings/keystores/{id}

特定密钥库配置

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含特定接口的密钥库设置
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

请参阅密钥库设置集合模式，*keystore* 对象。

#### 示例

Request: GET /rest/settings/keystores/1  
KMIP Keystores:  
Response: 200 OK

```
{
  "id": 1,
  "name": null,
  "state": "INACTIVE",
  "product": "IBM_SKLM",
  "host": null,
  "port": 0,
  "keystore": null,
  "keystore_alias": null,
  "keystore_password": "*****",
  "truststore": null,
  "truststore_alias": null,
  "truststore_password": "*****",
  "master_keystore_id": 0,
  "type": "keystore-kmip"
}
```

HSM Keystores:  
Request: GET /rest/settings/keystores/1

```
{
  "id": 1,
  "name": "hsm",
  "state": "INACTIVE",
  "product": "HSM",
  "hsm_password": null,
  "hsm_token": "0",
  "key_handle": "448",
  "master_keystore_id": 0,
}
```

```
    "type": "keystore-hsm"
  }
```

## PUT

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>204</b>	请求已成功，但未包含任何更改。不需要刷新客户端表示。
<b>205</b>	请求已成功，且资源已更新。
<b>400</b>	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改密钥库设置
<b>404</b>	找不到具有指定 {id} 的接口，且该接口之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>409</b>	对接口所做的更改与另一个接口相冲突。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅密钥库设置集合模式，*keystore* 对象。

### 示例

```
KMIP Keystores:
Request: PUT /rest/settings/keystores/1
```

```
{
  "id": 1,
  "name": null,
  "state": "INACTIVE",
  "product": "IBM_SKLM",
  "host": "192.168.2.87",
  "port": 5696,
  "keystore": null,
  "keystore_alias": null,
  "keystore_password": "*****",
  "truststore": null,
  "truststore_alias": null,
  "truststore_password": "*****",
  "master_keystore_id": 0,
  "type": "keystore-kmip"
}
```

```
Response: 205 Reset Content
```

```
HSM Keystores:
Request: PUT /rest/settings/keystores/1
{
  "id": 1,
```



```

    "name": "hsm",
    "state": "INACTIVE",
    "product": " HSM ",
    "hsmpassword": null,
    "hsmtoken": "0",
    "key_handle": "448",
    "master_keystore_id": 0,
    "type": "keystore-hsm"
  }

```

Response: 205 Reset Content

## PATCH

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>204</b>	请求已成功，但未包含任何更改。不需要刷新客户端表示。
<b>205</b>	请求已成功，且资源已更新。
<b>400</b>	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改密钥库设置
<b>404</b>	找不到具有指定 {id} 的接口，且该接口之前未存在过
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>409</b>	对接口所做的更改与另一个接口相冲突。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

请参阅密钥库设置集合模式，*keystore* 对象。

### 示例

```

KMIP Keystores:
Request: PATCH /rest/settings/keystores/1
{
  {
    "id": 1,
    "state": "INACTIVE",
    "type": "keystore-kmip"
  }
}

Response: 205 Reset Content

HSM Keystores:
Request: PATCH /rest/settings/keystores/1
{
  {

```

```
"id": 1,  
"state": "INACTIVE",  
"type": "keystore-hsm"  
}
```

Response: 205 Reset Content

## DELETE

### 模式

*DELETE* 中不发送主体

### 示例

KMIP or HSM Keystores:  
Request: DELETE /rest/settings/keystores/1  
Response: 204 No content

# 第 11 章 归档并上载文件

## /files

服务器所生成的归档（将来的发行版）和客户所上载的文件的集合

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含对子文件或文件集合的引用的集合。这些引用将不包含文件内容。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

```
{
  "type": "array",
  "minItems": 4,
  "uniqueItems": true,
  "items": [
    {
      "type": "object",
      "required": ["type", "id", "href"],
      "additionalProperties": true,
      "properties": {
        "type": { "enum": [ "ref" ] },
        "id": { "enum": [ "upload" ] },
        "href": "string"
      }
    }
  ]
}
```

#### 示例

```
Request: GET /rest/files

Response: 200 OK

[
  {
    "type": "ref",
    "id": "upload",
```

```
    "href": "/rest/files/upload",
  }
]
```

## /files/upload

这是一个位置，供客户机上载文件以便与其他资源配合使用。

此资源强制实施所有权可视性，因此，用户只能看到自己所上载的文件。文件一旦传递到请求中的其他资源，就可能被服务器除去。通常不需要手动删除文件。

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含对客户所上载文件的引用的集合。这些引用将不包含文件内容。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

```
{
  "type": "array",
  "minItems": 0,
  "uniqueItems": true,
  "items": [
    {
      "type": "object",
      "required": [ "type", "name", "id", "href" ],
      "additionalProperties": true,
      "properties": {
        "type": { "enum": [ "ref" ] },
        "name": { "type": "string", "id": "number", "href": "string" },
        "id": { "type": "string", "id": "number", "href": "string" },
        "href": { "type": "string", "id": "number", "href": "string" }
      }
    }
  ]
}
```

#### 示例

Request: GET /rest/files/upload

Response: 200 OK

```
[
  {

```

```
    "name": "cert.pem",
    "id": 1,
    "type": "ref",
    "href": "/rest/files/upload/1"
  }
]
```

## POST

将新文件上传到服务器

.

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>201</b>	已上传新文件。 <b>Location</b> 标头将提供该文件的 URL。
<b>400</b>	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户上传文件
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

### 模式

此 *POST* 中不应发送内容。

### 示例

```
Request: POST /rest/files/upload
Response: 201 Created | Location: /files/upload/my_saved_backup_2015-10-15.tar.gz
```

## [/files/upload/{id}](#)

特定客户机上载的文件

## GET

### 查询参数

参数	描述
<无>	

## 状态码

代码	描述
<b>200</b>	响应主体将包含文件的二进制下载
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 的上载内容
<b>406</b>	<b>Accept</b> 标头必须保持未设置状态，或者必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

客户上载内容被视为二进制数据，并且没有模式

## DELETE

### 查询参数

参数	描述
<无>	

## 状态码

代码	描述
<b>204</b>	删除已成功
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>404</b>	找不到具有指定 {id} 的上载内容
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

*DELETE* 中不发送主体

## 示例

```
Request: DELETE /rest/files/upload/my_saved_backup_2015-10-15.tar.gz
Response: 204 No Content
```

# 第 12 章 高级属性

## /properties

用于配置高级产品属性的资源

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
200	响应主体将包含属性摘要
401	当前请求未经过认证。
403	未向当前用户进行授权。
406	<b>Accept</b> 标头必须设置为受支持的内容类型。
500	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
503	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

```
{
  "type": "array",
  "minItems": 0,
  "uniqueItems": true,
  "items": [
    {
      "type": "object",
      "required": [ "type", "key",
"scope", "value", "defaultValue", "needsRestart" ],
      "additionalProperties": false,
      "properties": {
        "type": { "enum": [ "property" ] },
        "key": "string",
        "scope": { "enum": [ "LOCAL", "CLUSTER" ] },
        "value": "string",
        "defaultValue": "string",
        "needsRestart": "string"
      }
    }
  ]
}
```

#### 示例

```
Request: GET /rest/properties

Response: 200 OK

[
  {
```

```
    "key": "com.securityfirstcorp.atlantis.bundles.dataman.password",
    "alias": "node.db.password",
    "scope": "LOCAL",
    "value": "PASSWORD",
    "defaultValue": "PASSWORD",
    "needsRestart": false,
    "type": "property"
  },
  {
    "key": "com.securityfirstcorp.atlantis.bundles.dataman.initialConnections",
    "alias": "node.db.initConns",
    "scope": "LOCAL",
    "value": "1",
    "defaultValue": "1",
    "needsRestart": false,
    "type": "property"
  }
]
```

## /properties/{key}

特定属性

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含特定属性
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

请参阅属性模式，*property* 对象。

#### 示例

```
Request: GET /rest/property/com.securityfirstcorp.atlantis.bundles.dataman.password
Response: 200 OK

{
  "key": "com.securityfirstcorp.atlantis.bundles.dataman.password",
  "alias": "node.db.password",
  "scope": "LOCAL",
  "value": "PASSWORD",
  "defaultValue": "PASSWORD",
  "needsRestart": false,
  "type": "property"
}
```



# PATCH

## 查询参数

参数	描述
<无>	

## 状态码

代码	描述
<b>204</b>	请求已成功，但未包含任何更改。 不需要刷新客户端表示。
<b>205</b>	请求已成功，且资源已更新。
<b>400</b>	请求主体的格式不正确。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改时间设置
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅 *property* 对象模式。

## 示例

```
Request: PATCH /rest/property/com.securityfirstcorp.atlantis.bundles.dataman.password
{
  "value": "newPassword34"
}
Response: 205 Reset Content
```



# 第 13 章 设置和配置问题

## 设置和配置问题

### /issues

用于解决未决产品设置问题或持续存在的配置问题的资源

#### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
200	响应主体将包含问题摘要
401	当前请求未经过认证。
403	未向当前用户进行授权。
406	<b>Accept</b> 标头必须设置为受支持的内容类型。
500	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
503	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

```
{
  "type": "array",
  "minItems": 0,
  "uniqueItems": true,
  "items": [
    {
      "type": "object",
      "required": [ "type", "id",
"issue_id", "dismiss_status", "importance", "permission_to_dismiss" ],
      "additionalProperties": false,
      "properties": {
        "type": { "enum": [ "issue" ] },
        "id": "number",
        "issue_id": "string",
        "dismiss_status": { "enum":
[ "DISMISSED", "NOT_ALLOWED", "NOT_DISMISSED", "UNKNOWN" ] },
        "importance": { "enum":
[ "HIGH", "MEDIUM", "LOW", "UNKNOWN" ] },
        "permission_to_dismiss": "boolean"
      }
    }
  ]
}
```

示例

```
Request: GET /rest/properties

Response: 200 OK

[
  {
    "id": 1,
    "issue_id": "US_2_policyAndSecurityRoles",
    "dismiss_status": "NOT_DISMISSED",
    "importance": "LOW",
    "permission_to_dismiss": true,
    "type": "issue"
  },
  {
    "id": 2,
    "issue_id": "US_1_onlyOneUser",
    "dismiss_status": "NOT_ALLOWED",
    "importance": "HIGH",
    "permission_to_dismiss": true,
    "type": "issue"
  }
]
```

/issues/{id}

特定属性

GET

查询参数

参数	描述
<无>	

状态码

代码	描述
200	响应主体将包含特定问题
401	当前请求未经过认证。
403	未向当前用户进行授权。
406	<b>Accept</b> 标头必须设置为受支持的内容类型。
500	发生了内部错误。 请查看 <b>Warning</b> 标头以获取更多信息。
503	服务器服务不可用。 请查看 <b>Warning</b> 标头以获取更多信息。

模式

请参阅属性模式，*property* 对象。

示例

```
Request: GET /rest/issues/1

Response: 200 OK

{
  "id": 1,
  "issue_id": "US_2_policyAndSecurityRoles",
  "dismiss_status": "NOT_DISMISSED",
  "importance": "LOW",
  "permission_to_dismiss": true,
```

```
}    "type": "issue"
```

## PATCH

### 查询参数

参数	描述
<无>	

### 状态码

代码	描述
<b>204</b>	请求已成功，但未包含任何更改。不需要刷新客户端表示。
<b>205</b>	请求已成功，且资源已更新。
<b>400</b>	请求主体的格式不正确。请查看 <b>Warning</b> 标头以获取更多信息。
<b>401</b>	当前请求未经过认证。
<b>403</b>	未授权当前用户更改时间设置
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

## 模式

请参阅 *issues* 对象模式。

## 示例

Request: PATCH /rest/issues/1

```
{  "dismiss_status": "DISMISSED"}
```

Response: 205 Reset Content



# 第 14 章 全球化资源

## /locales

用于列示产品支持的语言环境的资源。

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
200	响应主体将包含受支持的语言环境
401	当前请求未经过认证。
403	未向当前用户进行授权。
406	<b>Accept</b> 标头必须设置为受支持的内容类型。
500	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
503	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

```
{
  "type": "array",
  "minItems": 0,
  "uniqueItems": true,
  "items": [
    {
      "type": "object",
      "required": [ "type", "id", "display_name",
"language", "country", "default" ],
      "additionalProperties": false,
      "properties": {
        "type": { "enum": [ "product-locale" ] },
        "id": "string",
        "display_name": "string",
        "language": "string",
        "country": "string",
        "default": "boolean"
      }
    }
  ]
}
```

#### 示例

```
Request: GET /rest/locales

Response: 200 OK

[
  {
```

```

    "id": "en_US",
    "display_name": "english (United States)",
    "language": "en",
    "country": "United States",
    "default": true,
    "type": "product-locale"
  },
  {
    "id": "es_MX",
    "display_name": "español (México)",
    "language": "es",
    "country": "Mexico",
    "default": false,
    "type": "product-locale"
  }
]

```

## /locales/{id}

特定语言环境

.

### GET

#### 查询参数

参数	描述
<无>	

#### 状态码

代码	描述
<b>200</b>	响应主体将包含特定语言环境
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

请参阅产品语言环境模式，*product-locale* 对象。

#### 示例

```

Request: GET /rest/locales/en_US

Response: 200 OK

{
  "id": "en_US",
  "display_name": "english (United States)",
  "language": "en",
  "country": "United States",
  "default": true,
  "type": "product-locale"
}

```



## /strings

用于列示产品字符串的资源。

### GET

#### 查询参数

参数	描述
<b>locale</b>	用于指示以何种语言返回字符串的语言环境值。如果未提供语言环境，将以缺省语言返回字符串。示例： <b>locale=en_US</b>

#### 状态码

代码	描述
<b>200</b>	响应主体将包含产品字符串
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

```
{
  "type": "array",
  "minItems": 0,
  "uniqueItems": false,
  "items": [
    {
      "type": "object",
      "required": [ "id" ],
      "minItems": 0,
      "uniqueItems": true,
      "items": [
        { "type": "string" }
      ]
    }
  ]
}
```

#### 示例

Request: GET /rest/strings

Response: 200 OK

```
[
  {
    "id": "web",
    "sLoadingRecords": "Loading...",
    "ButtonLabels_ExportCSV": "Export CSV",
    "Jobs_Status": "Status",
    "PageNames_Processes": "Processes",
    "US_1_ONLY_ONE_SECURITY_APPROVER_category": "User Setup"
  },
  {
    "id": "events",
    "PS00070003": "Backup data creation job {0} has started.",
  }
]
```

```

    "PS00070002": "User {0} has deleted backup file {1}.",
    "PS00070005": "Backup data creation job {0} has failed.",
    "PS00070004": "Backup data creation job {0} has completed successfully."
  }
]

```

## /strings/{id}

特定字符串集

### GET

#### 查询参数

参数	描述
<b>locale</b>	用于指示以何种语言返回字符串的语言环境值。如果未提供语言环境，将以缺省语言返回字符串。示例： <b>locale=en_US</b>

#### 状态码

代码	描述
<b>200</b>	响应主体将包含特定字符串集
<b>401</b>	当前请求未经过认证。
<b>403</b>	未向当前用户进行授权。
<b>406</b>	<b>Accept</b> 标头必须设置为受支持的内容类型。
<b>500</b>	发生了内部错误。请查看 <b>Warning</b> 标头以获取更多信息。
<b>503</b>	服务器服务不可用。请查看 <b>Warning</b> 标头以获取更多信息。

#### 模式

请参阅字符串模式。

#### 示例

Request: GET /rest/strings/web

Response: 200 OK

```

{
  "id": "web",
  "sLoadingRecords": "Loading...",
  "ButtonLabels_ExportCSV": "Export CSV",
  "Jobs_Status": "Status",
  "PageNames_Processes": "Processes",
  "US_1_ONLY_ONE_SECURITY_APPROVER_category": "User Setup"
}

```

# 声明

---

本信息是为在美国提供的产品和服务编写的。IBM 可能会以其他语言提供本资料。但是，您可能必须拥有该语言的产品或产品版本副本才能访问相应的资料。

IBM 可能在其他国家或地区不提供本文档中讨论的产品、服务或功能特性。有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务的操作，由用户自行负责。

IBM 公司可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可。您可以用书面形式将许可查询寄往：

*IBM Director of Licensing*  
*IBM Corporation*  
*North Castle Drive, MD-NC119*  
*Armonk, NY 10504-1785*  
*US*

有关双字节字符集 (DBCS) 信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

*Intellectual Property Licensing*  
*Legal and Intellectual Property Law*  
*IBM Japan Ltd.*  
*19-21, Nihonbashi-Hakozakicho, Chuo-ku*  
*Tokyo 103-8510, Japan*

## **本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区：**

International Business Machines Corporation “按现状”提供本出版物，不附有任何种类的（无论是明示的还是暗示的）保证，包括但不限于暗示的有关非侵权、适销和适用于某种特定用途的保证。

某些管辖区域不允许免除特定交易中明示的或暗示的保证，因此，该声明可能并不适用于您。

本信息可能包含技术方面不够准确的地方或印刷错误。本信息将定期更改；这些更改将编入本信息的新版本中。IBM 可以随时对本资料中描述的产品和/或程序进行改进和/或更改，而不另行通知。

本信息中对任何非 IBM Web 站点的引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。那些 Web 站点中的资料不是 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：(i) 使其能够在独立创建的程序和其它程序（包括本程序）之间进行信息交换，以及 (ii) 使其能够对已经交换的信息进行相互使用，请与下列地址联系：

*IBM Director of Licensing*  
*IBM Corporation*  
*North Castle Drive, MD-NC119*  
*Armonk, NY 10504-1785*  
*US*

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本文档中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际程序许可协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在可控环境下取得的。因此，其他操作环境中获得的结果可能存在很大差异。某些测量可能是在开发级系统上进行的，并且不能保证这些测量在通常可用的系统上是相同的。此外，某些

测量值可能是通过推断估算得到的。实际结果可能会有差异。本文档用户应该在其特定环境中验证适用的数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。对于非 IBM 产品的性能问题必须和这些产品的供应商一起解决。

所有关于 IBM 未来方向或意向的声明都可随时更改或撤消，而不另行通知，它们仅表示 IBM 的目标和意愿。

所有显示的 IBM 价格均是 IBM 的当前建议零售价，可随时更改而不另行通知。交易商价格可能会有所不同。

此信息仅用于规划目的。在所描述的产品可用之前，此处的信息会有更改。

本信息包含在日常业务操作中使用的数据和报告的示例。为了尽可能完整地说明这些示例，示例中可能会包括个人、公司、品牌和产品的名称。所有这些名字都是虚构的，若现实生活中实际业务企业使用的名字和地址与此相似，纯属巧合。

版权许可证：

本信息包含源语言形式的样本应用程序，用以阐明在不同操作平台上的编程技术。如果是为按照在编写样本程序的操作平台上的应用程序编程接口（API）进行应用程序的开发、使用、经销或分发为目的，您可以任何形式对这些样本程序进行复制、修改、分发，而无须向 IBM 付费。这些示例尚未在所有条件下经过全面测试。因此，IBM 不能担保或暗示这些程序的可靠性、可维护性或功能。这些实例程序“按现状”提供，不附有任何种类的保证。对于因使用样本程序所引起的任何损害，IBM 概不负责。

这些样本程序或任何衍生产品的每个副本或其任何部分，都必须包括版权声明，如下所示：

©（贵公司的名称）（年）。此部分代码源于 IBM Corp. 样本程序。© Copyright IBM Corp. \_输入年或年份区间\_.

如果您要以软拷贝形式查看此信息，则可能不会显示照片和彩色图例。

## 商标

---

SPx 和 Security First Corp 是 Security First Corp. 在全球许多管辖区域注册的商标或注册商标。其他产品和服务可能是 Security First Corp. 或其他公司的商标。

IBM、IBM 徽标和 ibm.com 是 International Business Machines Corp. 在全球许多管辖区域注册的商标或注册商标。其他产品和服务名称可能是 IBM 或其他公司的商标。IBM 商标的最新列表可在以下 Web 站点的“Copyright and trademark information”中获取：<http://www.ibm.com/legal/copytrade.shtml>。

Adobe、Adobe 徽标、PostScript 以及 PostScript 徽标是 Adobe Systems Incorporated 在美国和/或其他国家或地区的注册商标或商标。

Apache Software Foundation (ASF) 拥有所有与 Apache 相关的商标、服务标记以及代表 Apache 项目社区的图形徽标，并且所有 Apache 项目的名称都是 ASF 的商标。

Node.JS 是 Joyent, Inc. CORPORATION DELAWARE 345 California Street; Suite 2000 San Francisco CALIFORNIA 94104 的注册商标。

Unicode 和 Unicode 徽标是 Unicode, Inc. 在美国和其他国家或地区的注册商标。

CentOS Marks 是 Red Hat, Inc.（“Red Hat”）的商标。

“Red Hat”、Red Hat Linux、Red Hat “Shadowman”徽标以及列示的产品是 Red Hat, Inc. 在美国和其他国家或地区的商标或注册商标。

Linux 是 Linus Torvalds 在美国和/或其他国家或地区的注册商标。

Microsoft、Windows、Windows NT 以及 Windows 徽标是 Microsoft Corporation 在美国和/或其他国家或地区的商标。

Java 和所有基于 Java 的商标和徽标是 Oracle 和/或其子公司的商标或注册商标。

## 产品文档的条款和条件

---

如果符合以下条款和条件，那么将授予您使用这些出版物的许可权。

### 适用性

这些条款和条件是 IBM Web 站点使用条款的补充。

### 个人用途

在保留所有专有权声明的前提下，您可以为了个人用途而非商业性用途复制这些出版物。未经 IBM 明确同意，您不得分发、展示或制作这些出版物或其中任何部分的衍生作品。

### 商业用途

在保留所有专有权声明的前提下，您仅可在贵公司内部复制、分发和显示这些出版物。未经 IBM 明确同意，您不得制作这些出版物的衍生作品，或者在您的企业外部复制、分发或展示这些出版物或其中的任何部分。

### 权利

除非在此许可权中明确授予，否则未对出版物或其中包含的任何信息、数据、软件或其他知识产权（明示或暗示）授予任何其他许可权、许可证和权限。

IBM 保留撤销此处授予的许可权的权利，只要 IBM 自行认为出版物的使用损害了 IBM 的利益或 IBM 确定未正确遵循上述指示信息。

除非完全遵守所有适用的法律法规（其中包括美国的所有出口法律法规），否则不得下载、出口或者再出口本资料

IBM 对这些出版物的内容不作任何保证。“按现状”提供本出版物，不附有任何种类的（无论是明示的还是暗示的）保证，包括但不限于暗示的有关适销、非侵权和适用于某特定用途的保证。

## 隐私策略注意事项

---

IBM 软件产品，包括“软件即服务”解决方案（软件供应项目），可能会使用 Cookie 或其他技术来收集产品使用情况信息，以帮助改善最终用户体验、定制与最终用户的交互或者用于其他目的。在许多情况下，软件供应项目不收集个人可标识信息。我们的某些软件供应项目可以帮助您收集个人可标识信息。如果本软件供应项目使用 Cookie 来收集个人可标识信息，那么下文将描述有关本供应项目的 Cookie 使用情况的具体信息。本软件供应项目不使用 Cookie 或其他技术来收集个人可标识信息。

如果为本软件供应项目部署的配置允许客户通过 Cookie 和其他技术向最终用户收集个人可标识信息，那么您应该寻求自己的法律建议，了解所有适用于此类数据收集的法律，包括所有对声明和同意的要求。

有关将各种技术（包括 Cookie）用于这些用途的更多信息，请参阅 IBM 隐私策略 (<http://www.ibm.com/privacy>)、IBM 网上隐私声明 (<http://www.ibm.com/privacy/details>) 中标题为“Cookie、Web Beacon 和其他技术”的部分以及“IBM Software Products and Software-as-a-Service Privacy Statement” (<http://www.ibm.com/software/info/product-privacy>)。







部件号 CC0LUEN

GC43-5024-00



(1P) P/N: CC0LUEN

