

## #Secrets API参考

## GET /v1/secrets

获取机密信息列表

可以通过URL传递的参数过滤机密信息列表

此接口并不会列出实际的有效的机密信息，客户必须通过每个机密信息的索引来获取有效数据。

## 参数

名称	数据类型	说明
offset	int	待检索信息在总列表中的起始索引
limit	int	要返回的最大记录数（最大100个），默认限制为10
name	string	检索与此值相似的所有机密
alg	string	检索与此算法相似的所有机密
mode	string	检索与此模式相似的所有机密
bits	int	检索密钥长度为此值得所有机密
secret_type	string	检索此密钥类型的所有机密
acl_only	bool	检索包含ACL选择的所有机密，忽略项目范围
created	string	日期过滤器，选择所有符合指定创建日期的所有机密。详细信息参考下面的日期过滤器
updated	string	日期过滤器，选择所有符合指定更行日期的所有机密。详细信息参考下面的日期过滤器
expiration	string	日期过滤器，选择所有符合指定失效日期的所有机密。详细信息参考下面的日期过滤器
sort	string	确定返回列表的排序方式。详细信息参考下面的排序

## 日期过滤器

created、updated以及expiration参数是ISO 8601格式的时间戳，以逗号分隔的列表。时间戳可以使用任意的比较操作符做前缀：gt:大于，gte:大于等于，lt:小于，lte:小于等于。

示例：获取在2020年1月1日到2020年2月1日之间失效的所有机密

```
GET /v1/secrets?expiration=gte:2020-01-01T00:00:00,lt:2020-02-01T00:00:00
```

## 排序

排序参数的是一系列由逗号分隔的排序关键字的列表，可使用的排序关键字有：created, expiration, mode, name, secret\_type, status和updated。

每个排序关键字也都包括一个顺序：asc:升序, desc:降序。默认使用升序:asc。

示例：获取从最近到最初创建的机密列表

```
GET /v1/secrets?sort=created:desc
```

## 请求

```
GET /v1/secrets?offset=1&limit=2&sort=created
```

Headers:

Accept: application/json

X-Auth-Token: {keystone\_token}

(or X-Project-Id: {project id})

## 响应

```
{
  "next": "http://{barbican_host}:9311/v1/secrets?limit=2&offset=3",
  "previous": "http://{barbican_host}:9311/v1/secrets?limit=2&offset=0",
  "secrets": [
    {
      "algorithm": null,
      "bit_length": null,
      "content_types": {
        "default": "application/octet-stream"
      },
      "created": "2015-04-07T03:37:19.805835",
      "creator_id": "3a7e3d2421384f56a8fb6cf082a8efab",
      "expiration": null,
      "mode": null,
      "name": "opaque octet-stream base64",
      "secret_ref": "http://{barbican_host}:9311/v1/secrets/{uuid}",
      "secret_type": "opaque",
      "status": "ACTIVE",
      "updated": "2015-04-07T03:37:19.808337"
    },
    {
      "algorithm": null,
      "bit_length": null,
      "content_types": {
        "default": "application/octet-stream"
      },
      "created": "2015-04-07T03:41:02.184159",
      "creator_id": "3a7e3d2421384f56a8fb6cf082a8efab",
      "expiration": null,
      "mode": null,
```

```
        "name": "opaque random octet-stream base64",
        "secret_ref": "http://{barbican_host}:9311/v1/secrets/{uuid}",
        "secret_type": "opaque",
        "status": "ACTIVE",
        "updated": "2015-04-07T03:41:02.187823"
    },
],
"total": 5
}
```

## 响应属性

名称	类型	说明
secrets	list	包含一系列机密的集合，机密对象的属性与单个机密一致。
total	int	用户可用的机密信息总数
next	string	一个HATEOAS URL，用于根据offset和limit参数来获取下一组机密。该属性只有在机密总数大于offset和limit参数组合时可用
previous	string	一个HATEOAS URL，用于根据offset和limit参数来获取上一组机密。该属性只有在offset大于0时可用

## HTTP状态码

code	说明
200	成功的请求
401	无效的X-Auth-Token或者此token无权访问该资源

## POST /v1/secrets

创建一个机密实体。如果payload属性不在请求中，那么只创建机密的元数据，并且还需要后续的PUT请求才能成功创建机密实体。

## 属性

属性名称	类型	说明	默认值
name	string	(可选项) 用户设置的机密信息的名称	无
expiration	string	(可选项) ISO 8601格式的UTC时间戳。如果设置，那么在此日期之后将无法使用该机密	无
algorithm	string	(可选项) 用户或者系统提供的元数据信息	无
bit_length	int	(可选项) 用户或者系统提供的元数据信息，必须大于0	无
mode	string	(可选项) 用户或者系统提供的元数据信息	无

属性名称	类型	说明	默认值
payload	string	(可选项) 需要存储的数据。如果设置payload属性, 那么payload_content_type属性必须提供	无
payload_content_type	string	(可选项) payload内容的数据类型。更多信息请参阅Secret Types	无
payload_content_encoding	string	(可选项) (如果payload被编码过) 在JSON请求中payload的编码方式需要包含。目前只支持base64编码	无
secret_type	string	(可选项) 用于标识存储机密的类型。详细信息参见Secret Types	opaque

## 请求

```
POST /v1/secrets
Headers:
  Content-Type: application/json
  X-Auth-Token: <token>

Content:
{
  "name": "AES key",
  "expiration": "2015-12-28T19:14:44.180394",
  "algorithm": "aes",
  "bit_length": 256,
  "mode": "cbc",
  "payload": "YmVlcg==",
  "payload_content_type": "application/octet-stream",
  "payload_content_encoding": "base64"
}
```

## 响应

```
201 Created

{
  "secret_ref": "https://{barbican_host}/v1/secrets/{secret_uuid}"
}
```

## HTTP状态码

code	说明
201	成功创建机密
400	错误的请求

code	说明
401	无效的X-Auth-Token或者此token无权访问该资源
403	访问被拒绝。用户已经通过验证，但是无权创建。原因在于用户的角色或项目的配额
415	不支持的媒体类型

## GET /v1/secrets/{uuid}

检索机密的元数据

### 请求

```
GET /v1/secrets/{uuid}
Headers:
  Accept: application/json
  X-Auth-Token: {token}
  (or X-Project-Id: {project_id})
```

### 响应

```
200 OK

{
  "status": "ACTIVE",
  "created": "2015-03-23T20:46:51.650515",
  "updated": "2015-03-23T20:46:51.654116",
  "expiration": "2015-12-28T19:14:44.180394",
  "algorithm": "aes",
  "bit_length": 256,
  "mode": "cbc",
  "name": "AES key",
  "secret_ref": "https://{barbican_host}/v1/secrets/{secret_uuid}",
  "secret_type": "opaque",
  "content_types": {
    "default": "application/octet-stream"
  }
}
```

### HTTP状态码

code	说明
200	成功
401	无效的X-Auth-Token或者此token无权访问该资源

code	说明
404	未找到
406	访问不被接受

## PUT /v1/secrets/{uuid}

像只有元数据的机密数据（创建时POST请求中没有payload属性）中添加payload

### Headers

名称	说明	默认值
Content-Type	对应正常创建请求时的payload_content_type属性	text/plain
Content-Encoding	（可选项）对应正常创建请求时的payload_content_encoding属性	无

### 请求

```
PUT /v1/secrets/{uuid}
Headers:
  X-Auth-Token: <token>
  Content-Type: application/octet-stream
  Content-Encoding: base64

Content:
YmxhaA==
```

### 响应

```
204 No Content
```

### HTTP状态码

code	说明
204	成功
401	无效的X-Auth-Token或者此token无权访问该资源
404	未找到

## DELETE /v1/secrets/{uuid}

通过uuid删除机密数据

## 请求

```
DELETE /v1/secrets/{uuid}
Headers:
  X-Auth-Token: <token>
```

## 响应

```
DELETE /v1/secrets/{uuid}
Headers:
  X-Auth-Token: <token>
```

## HTTP状态码

code	说明
204	成功
401	无效的X-Auth-Token或者此token无权访问该资源
404	未找到

## GET /v1/secrets/{uuid}/payload

获取机密数据的内容

## 请求

```
GET /v1/secrets/{uuid}/payload
Headers:
  Accept: text/plain
  X-Auth-Token: <token>
```

## 响应

```
200 OK

beer
```

## HTTP状态码

code	说明
------	----

code	说明
204	成功
401	无效的X-Auth-Token或者此token无权访问该资源
404	未找到