

Consumers API 参考

GET {container_ref}/consumers

列出容器的消费者

可通过传递URL来过滤消费者列表。

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

请求

```
GET {container_ref}/consumers
Headers:
  X-Auth-Token: <token>
```

响应

```
200 OK

{
  "total": 3,
  "consumers": [
    {
      "status": "ACTIVE",
      "URL": "consumerurl",
      "updated": "2015-10-15T21:06:33.123878",
      "name": "consumername",
      "created": "2015-10-15T21:06:33.123872"
    },
    {
      "status": "ACTIVE",
      "URL": "consumerURL2",
      "updated": "2015-10-15T21:17:08.092416",
      "name": "consumername2",
      "created": "2015-10-15T21:17:08.092408"
    },
    {
      "status": "ACTIVE",
      "URL": "consumerURL3",
      "updated": "2015-10-15T21:21:29.970370",
```

```
        "name": "consumername3",
        "created": "2015-10-15T21:21:29.970365"
      }
    ]
  }
}
```

请求

```
GET {container_ref}/consumers?limit=1\&offset=1
Headers:
  X-Auth-Token: <token>
```

```
{
  "total": 3,
  "next":
"http://localhost:9311/v1/containers/{container_ref}/consumers?
limit=1&offset=2",
  "consumers": [
    {
      "status": "ACTIVE",
      "URL": "consumerURL2",
      "updated": "2015-10-15T21:17:08.092416",
      "name": "consumername2",
      "created": "2015-10-15T21:17:08.092408"
    }
  ],
  "previous":
"http://localhost:9311/v1/containers/{container_ref}/consumers?
limit=1&offset=0"
}
```

响应属性

名称	类型	说明
consumers	list	包含填充消费者元数据的词典列表
total	int	用户可用的消费者总数
next	string	一个HATEOAS URL，用于根据offset和limit参数来获取下一组消费者集合。该属性只有在消费者总数大于offset和limit参数组合时可用
previous	string	一个HATEOAS URL，用于根据offset和limit参数来获取上一组消费者集合。该属性只有在offset大于0时可用

HTTP状态码

code	说明
200	成功
401	无效的X-Auth-Token或无权访问此资源
403	拒绝访问。用户已通过认证，但是无权删除消费者。由用户角色决定

POST {container_ref}/consumers

创建消费者

属性

|属性名称|类型|说明|默认值| |name|string|用户设置的消费者名称|无| |url|string|用户或使用容器服务的URL|无|

请求

```
POST {container_ref}/consumers
Headers:
  X-Auth-Token: <token>
  Content-Type: application/json

Content:
{
  "name": "ConsumerName",
  "url": "ConsumerURL"
}
```

响应

```
200 OK

{
  "status": "ACTIVE",
  "updated": "2015-10-15T17:56:18.626724",
  "name": "container name",
  "consumers": [
    {
      "URL": "consumerURL",
      "name": "consumername"
    }
  ],
  "created": "2015-10-15T17:55:44.380002",
  "container_ref": "http://localhost:9311/v1/containers/74bbd3fd-9ba8-42ee-b87e-2eecf10e47b9",
  "creator_id": "b17c815d80f946ea8505c34347a2aeba",
  "secret_refs": [
```

```
{
  "secret_ref": "http://localhost:9311/v1/secrets/b61613fc-be53-4696-ac01-c3a789e87973",
  "name": "private_key"
},
"type": "generic"
}
```

HTTP状态请求

code	说明
200	成功
401	无效的X-Auth-Token或无权访问此资源
403	拒绝访问。用户已通过认证，但是无权创建消费者。由用户角色决定

DELETE {container_ref}/consumers

删除消费者

属性

|属性名称|类型|说明|默认值| |name|string|用户设置的消费者名称|无| |url|string|用户或使用容器服务的URL|无|

请求

```
DELETE {container_ref}/consumers
Headers:
  X-Auth-Token: <token>
  Content-Type: application/json

Content:
{
  "name": "ConsumerName",
  "URL": "ConsumerURL"
}
```

响应

```
200 OK

{
  "status": "ACTIVE",
  "updated": "2015-10-15T17:56:18.626724",
}
```

```
    "name": "container name",
    "consumers": [],
    "created": "2015-10-15T17:55:44.380002",
    "container_ref": "http://localhost:9311/v1/containers/74bbd3fd-9ba8-42ee-b87e-2eecf10e47b9",
    "creator_id": "b17c815d80f946ea8505c34347a2aeba",
    "secret_refs": [
      {
        "secret_ref": "http://localhost:9311/v1/secrets/b61613fc-be53-4696-ac01-c3a789e87973",
        "name": "private_key"
      }
    ],
    "type": "generic"
  }
```

HTTP状态请求

code	说明
200	成功
400	错误的请求
401	无效的X-Auth-Token或无权访问此资源
403	拒绝访问。用户已通过认证，但是无权创建消费者。由用户角色决定
404	未找到