

Merge Requests (

 Clone repository
(/gxkl/CRMS/wikis/git_access)

[Home \(/gxkl/CRMS/wikis/home\)](#)

Restful接口
(/qxkl/CRMS/wikis/restful%E6%8E%A5%E5%)

数据库

(/qxkl/CRMS/wikis/%E6%95%B0%E6%8D%A

脚本restful接口
(/qxkl/CRMS/wikis/%E8%84%9A%E6%9C%A

资源定义

Create

```
POST /root?ty=2 HTTP/1.1
Host: localhost:9080
x-crms-ri: 1111
```

```
{
  "ty" : 2,
  "val": {
    "onto": "dasdas"
  }
}
```

```
{
  "ty": 2,
  "val": {
    "ct": "2017-08-25 16:50:31.108",
    "lt": "2017-08-25 16:50:31.108",
    "onto": "dasdas",
    "pa": "/root",
    "ri": "/root/1",
    "ty": 2
  }
}
```

```
}  
}
```

Retrieve

使用Get方法可以对资源进行查询。对资源的查询目前分为三种：

- PartialRetrieve :对资源属性进行过滤
- Pagination :查询 DataTable 资源对应的表
- SubscriptionRetrieve :查询某个资源下的一个具体的 Subscription 下面分别进行举例。

PartialRetrieve

对一个资源进行查询，Url中不带任何参数，就表示要查询资源的所有参数。

```
GET /root/1 HTTP/1.1  
Host: localhost:9080  
x-crms-ri: 1111
```

得到的响应如下

```
{  
  "ty": 2,  
  "val": {  
    "ch": [],  
    "ct": "2017-08-25 16:50:31.108",  
    "lt": "2017-08-25 16:55:38.217",  
    "onto": "1231231231",  
    "pa": "/root",  
    "ri": "/root/1",  
    "subs": [  
      {  
        "id": "53b01a60-1efd-41be-b7b0-4b5a2155841a"  
      }  
    ],  
    "ty": 2  
  }  
}
```

如果想要查询指定的参数，需要指定查询类型 q=pr ，并如下的参数

- in :在参数内使用逗号隔开想要查询的属性的简称，代表要包含这些参数
 - ex :在参数内使用逗号隔开不想查询的属性的简称，带包不要包含这些参数
- 在上面两个参数同时存在的情况，in 的优先级高。比如：

```
GET /root/1?q=pr&in=ty,ri,pa HTTP/1.1  
Host: localhost:9080  
x-crms-ri: 1111
```

响应如下：

```
{
  "ty": 2,
  "val": {
    "pa": "/root",
    "ri": "/root/1",
    "ty": 2
  }
}
```

Pagination

对 DataTable 资源的对应的表进行查询。需要指定参数 q=pg ，可选的参数为

- offset : 偏移量
- len : 查询长度

比如:

```
GET /root/1/1?q=pg&len=3 HTTP/1.1
Host: localhost:9080
x-crms-ri: 1111
```

得到的响应如下:

```
{
  "ty": 5,
  "val": {
    "col": {
      "num": "int",
      "name": "text"
    },
    "con": [
      {
        "num": 3,
        "name": "dd",
        "id": "table_eababcbcb10b4e6d8e55a07600f14c3a",
        "time": "745865b7-8987-11e7-958c-95dbd7078590"
      },
      {
        "num": 3,
        "name": "dd",
        "id": "table_eababcbcb10b4e6d8e55a07600f14c3a",
        "time": "745865b6-8987-11e7-958c-95dbd7078590"
      },
      {
        "num": 3,
        "name": "dd",
        "id": "table_eababcbcb10b4e6d8e55a07600f14c3a",

```

```
        "time": "745865b5-8987-11e7-958c-95dbd7078590"
    },
    ],
    "ri": "/root/1/1",
    "ty": 5
}
}
```

SubscriptionRetrieve

对一个资源下的指定订阅进行查询。需要指定查询参数 `q=sr`，并且需要指定订阅的 `id`。比如：

```
GET /root/1?q=sr&id=53b01a60-1efd-41be-b7b0-4b5a2155841a HTTP/1.1
Host: localhost:9080
x-crms-ri: 1111
```

得到的响应如下：

```
{
  "ty": 20,
  "val": {
    "desc": "asdasdasdas",
    "ety": [
      1
    ],
    "id": "53b01a60-1efd-41be-b7b0-4b5a2155841a",
    "nct": 2,
    "nu": [
      "http://192.168.10.107:9999"
    ]
  }
}
```

Update

主要用来:

- 更新资源的属性
- 管理资源的订阅

下面分别进行举例：

更新资源属性

```
PUT /root/1 HTTP/1.1
Host: localhost:9080
x-crms-ri: 1111
```

```
{
  "ty" : 2,
  "val": {
    "onto": "1231231231"
  }
}
```

得到的响应如下：

```
{
  "ty": 2,
  "val": {
    "lt": "2017-08-26 16:19:55.599",
    "onto": "1231231231",
    "ty": 2
  }
}
```

响应的结果中包含了资源被更新后更新的属性，其中 ty 属性不允许更新，但会一直存在。
不是所有的资源属性都允许被更新，若该属性不允许被更新，即使用户更新此属性，也不会起作用。

管理订阅

通过在 put 方法的Url中添加不同的参数，可以针对某一个资源的订阅进行创建、更新、删除操作。举例如下：

创建订阅

创建订阅需要添加参数 so=1，表示这是一个添加订阅的更新。Http body 中需要填入JSON格式的 Subscription。

```
PUT /root/1?so=1 HTTP/1.1
Host: localhost:9080
x-crms-ri: 1111

{
  "ty" : 8,
  "val": {
    "desc": "resourceObjectSub1",
    "nu": "http://localhost:9497",
    "nct": 3,
    "ety": [1,2,3,4]
  }
}
```

得到的响应如下

```
{
  "ty": 1,
  "val": {
    "lt": "2017-09-12 13:58:36.828",
    "ri": "/root"
```

```

    ],
    "subs": [
        {
            "id": "338db269-9433-43a4-8e2a-b5d910a450db"
        }
    ],
    "ty": 1
}
}

```

添加订阅属于对资源的修改，返回资源的更新部分，其中 id 属性代表订阅的id，不允许用户来创建，而由系统自动生成，用户可以凭借这个 id 来对这个订阅进行管理。

更新订阅

对订阅的更新时，需要在Url中包含参数 so=2，表明这是一个订阅更新操作。并且需要在body中添加 id 属性，表示这个针对哪一个订阅的更新。

```

PUT /root/1?so=2 HTTP/1.1
Host: localhost:9080
x-crms-ri: 1111

{
  "ty" : 8,
  "val": {
    "id": "338db269-9433-43a4-8e2a-b5d910a450db",
    "desc": "asdadasdas"
  }
}

```

得到的响应如下：

```

{
  "ty": 1,
  "val": {
    "lt": "2017-09-12 14:00:12.827",
    "ri": "/root",
    "subs": [
      {
        "id": "338db269-9433-43a4-8e2a-b5d910a450db"
      }
    ],
    "ty": 1
  }
}

```

删除订阅

对订阅的删除时，需要在Url中包含参数 so=3，表明这是一个订阅删除操作。同样需要 id 属性来标明是针对哪一个订阅的删除。

```
PUT /root/1?so=3 HTTP/1.1
Host: localhost:9080
x-crms-ri: 1111
{
  "ty" : 8,
  "val": {
    "id": "338db269-9433-43a4-8e2a-b5d910a450db",
  }
}
```

得到的响应如下:

```
{
  "ty": 1,
  "val": {
    "lt": "2017-09-12 14:00:12.827",
    "ri": "/root",
    "subs": [
      {
        "id": "338db269-9433-43a4-8e2a-b5d910a450db"
      }
    ],
    "ty": 1
  }
}
```

Delete

对资源进行删除使用 Delete 方法，举例如下：

```
DELETE /root/1 HTTP/1.1
Host: localhost:9080
x-crms-ri: 1111
Content-Type: application/json
```

得到的响应为：

```
{
  "ty": 2,
  "val": "Success"
}
```

