LED2ResourcesAPI接口测试文档

分页查询资源

- 接口链接：URL地址：http://[域名]/app/led2/[version]/res_query
- 请求方式：POST
- 测试示例

```
@Test
    public void res_query() {
        JSONObject body = new JSONObject();
        body.put("skip", "0");
        body.put("limit", "3");
        body.put("id", "123");
        String sync = getClient().sync("led2/v1_0/res_query", body.toJSONString());
        System.out.println(sync);
    }
```

- 输出结果

```
{
    "code":200,
    "data":[
        {
            "uid":"5cb6d25df22affc6968ddd8d",
            "modifyUser":"",
            "modifyDate":"",
            "size":715566,
            "addUser":"5996644abb0b200d53c76d36",
            "name":"14_classicsportscars_chevroletcorvette.jpg",
            "type":2,
            "addDate":"2019-04-17 15:14:37"
        },
        {
            "uid":"5cc65460999a07073462e517",
            "modifyUser":"",
            "modifyDate":"",
            "size":191905440,
            "addUser":"5996644abb0b200d53c76d36",
            "name":"阿拉丁媒体采访-姜鸿飞.mp4",
            "type":3,
            "addDate":"2019-04-29 09:33:18"
        },
        {
            "uid":"5ccfcd54999a0708744d0fce",
            "modifyUser":"",
            "modifyDate":"",
            "size":10822335,
            "addUser":"5996644abb0b200d53c76d36",
            "name":"阿拉丁媒体采访-姜鸿飞(1).mp4",
            "type":3,
            "addDate":"2019-05-06 13:59:48"
        }
    ],
    "id":"123",
    "message":"OK",
    "status":"SUCCESS"
}
```

- 接口测试用例

参数	方法返回（无data）	结论
id/skip/limit为空时	{"code":20002,"id":"","message":"id/skip/limit","status":"MISSING_REQUEST_PARAM"}	正确
id为空字符串时	{"code":200,"id":"","message":"OK","status":"SUCCESS"}	错误
skip/limit为空字符串时	{"code":20003,"id":"123","message":"skip/limit","status":"BAD_PARAM"}	正确
skip/limit为不合法字符/不正确取值范围时	{"code":20003,"id":"123","message":"skip/limit","status":"BAD_PARAM"}	正确
id不为空字符串，skip和limit为正确取值范围时	{"code":200,"id":"123","message":"OK","status":"SUCCESS"}	正确

- 接口测试结果：分页查询LED2资源，id为空字符串时不应返回正确，输出数据结构缺少资源总数total，输出结果中type字段类型为Integer与接口说明文档类型不符合

根据UID查询资源

- 接口链接：URL地址：http://[域名]/app/led2/[version]/res_query_byId
- 请求方式：POST
- 测试示例

```
@Test
public void res_query_byId() {
    JSONObject body = new JSONObject();
    body.put("id", "123");
    body.put("uid", "5cb6d25df22affc6968ddd8d");
    String sync = getClient().sync("led2/v1_0/res_query_byId", body.toJSONString());
    System.out.println(sync);
}
```

- 输出结果

```
{
    "code":200,
    "data":[
        {
            "uid":"5cb6d25df22affc6968ddd8d",
            "modifyUser":"",
            "modifyDate":"",
            "size":715566,
            "addUser":"5996644abb0b200d53c76d36",
            "name":"14_classicsportscars_chevroletcorvette.jpg",
            "type":2,
            "addDate":"2019-04-17 15:14:37"
        }
    ],
    "id":"123",
    "message":"OK",
    "status":"SUCCESS"
}
```

- 接口测试用例

参数	方法返回（无**data**）	结论
id/uid为空时	{"code":20002,"id":"","message":"id/uid","status":"MISSING_REQUEST_PARAM"}	正确
id为空字符串时	{"code":200,"id":"","message":"OK","status":"SUCCESS"}	错误
id不为空字符串，uid不正确时	{"code":20004,"id":"123","message":"数据不存在","status":"NOT_FOUND_PARAM"}	错误
id不为空字符串，uid正确时	{"code":20004,"id":"123","message":"数据不存在","status":"NOT_FOUND_PARAM"}	错误

- 测试结果：代码逻辑有问题，接口文档中说明是根据uid查询资源，但是实际代码中却是根据name来查询的，而且当uid出现错误时，个人觉得查询结果应该为success，只不过没有查询数据而已，这个问题只在查询中有歧义，应该统一。

根据资源名称查询资源URL（暂未完成）

- 接口链接：URL地址：http://[域名]/app/led2/[version]/res_query_byName
- 请求方式：POST
- 测试示例

- 输出结果

- 测试结果：

添加资源

- 接口链接：URL地址：http://[域名]/app/led2/[version]/res_add
- 请求方式：POST
- 测试示例

```
@Test
    public void res_add() {
        JSONObject body = new JSONObject();
        body.put("id", "123");
        body.put("type", "2");
        body.put("name", "Marcos测试1");
        String sync = getClient().sync("led2/v1_0/res_add", body.toJSONString());
        System.out.println(sync);
    }
```

- 输出结果

- 测试结果：输出结果报错，具体入参请在接口文档中描述清楚

删除资源

- 接口链接：URL地址：http://[域名]/app/led2/[version]/
- 请求方式：POST
- 测试示例

```
    @Test
    public void res_del() {
        JSONObject body = new JSONObject();
        body.put("id", "123");
        body.put("uid", "5d42d82e74c9d31e48e68804");
        String sync = getClient().sync("led2/v1_0/res_del", body.toJSONString());
        System.out.println(sync);
    }
```

- 输出结果

```
delete ledresource failed
```

- 测试结果：数据库有数据但是无法删除

下载资源（暂未完成）

- 接口链接：URL地址：http://[域名]/app/led2/[version]/res_del
- 请求方式：POST
- 测试示例

- 输出结果

- 测试结果：

查询资源分组列表

- 接口链接：URL地址：http://[域名]/app/led2/[version]/res_group_query
- 请求方式：POST
- 测试示例

```
@Test
public void res_group_query() {
    JSONObject body = new JSONObject();
    body.put("skip", "0");
    body.put("limit", "1");
    body.put("id", "123");
    String sync = getClient().sync("led2/v1_0/res_group_query", body.toJSONString());
    System.out.println(sync);
}
```

- 输出结果

```
{
    "code":200,
    "data":[
        {
            "uid":"5c9c2b42999a0707e07c9953",
            "groupName":"123"
        }
    ],
    "id":"123",
    "message":"OK",
    "status":"SUCCESS"
}
```

- 测试结果：显示的字段太少，建议把资源数组与设备数组展示出来，分页查询没有显示查询总数

根据资源分组id查询资源

- 接口链接：URL地址：http://[域名]/app/led2/[version]/res_by_groupId

- 请求方式：POST
- 测试示例

```
    @Test
    public void res_by_groupId() {
        JSONObject body = new JSONObject();
        body.put("id", "123");
        body.put("uid", "5c9c2b42999a0707e07c9953");
        String sync = getClient().sync("led2/v1_0/res_by_groupId", body.toJSONString());
        System.out.println(sync);
    }
```

- 输出结果

```
{
    "code":200,
    "data":[
        {
            "uid":"5d2e8fe1f22aff4b8dfff15a",
            "modifyUser":"",
            "modifyDate":"",
            "size":67997,
            "addUser":"5996644abb0b200d53c76d36",
            "name":"company.png",
            "type":2,
            "addDate":"2019-07-17 11:02:57"
        }
    ],
    "id":"123",
    "message":"OK",
    "status":"SUCCESS"
}
```

- 测试结果：建议把资源数组与设备数组展示出来，单个查询时建议data类型为JsonObject

添加资源分组

- 接口链接：URL地址：http://[域名]/app/led2/[version]/res_group_add
- 请求方式：POST
- 测试示例

```
@Test
    public void res_group_add() {
        JSONObject body = new JSONObject();
        body.put("id", "123");
        body.put("name", "123");
        body.put("resources", "123,234,345");
        String sync = getClient().sync("led2/v1_0/res_group_add", body.toJSONString());
        System.out.println(sync);
    }
```

- 输出结果

```
com.shuncom.sdk.core.exception.ApiRequestException: Response body format error or unexcepted exception : {"Ret": 4004
```

- 测试结果：返回报错

删除资源分组

- 接口链接：URL地址：http://[域名]/app/led2/[version]/res_group_del
- 请求方式：POST
- 测试示例

```
@Test
public void res_group_del() {
    JSONObject body = new JSONObject();
    body.put("id", "123");
    body.put("uid", "5c9c2b42999a0707e07c9953");
    String sync = getClient().sync("led2/v1_0/res_group_del", body.toJSONString());
    System.out.println(sync);
}
```

- 输出结果

```
{"code":200,"id":"123","message":"OK","status":"SUCCESS"}
```

- 测试结果：返回正确

更新资源分组

- 接口链接：URL地址：http://[域名]/app/led2/[version]/res_group_update
- 请求方式：POST
- 测试示例

```
@Test
public void res_group_update() {
    JSONObject body = new JSONObject();
    body.put("id", "123");
    body.put("uid", "5cb02e89999a0713781782a8");
    body.put("name","123");
    body.put("resourcesList","123");
    String sync = getClient().sync("led2/v1_0/res_group_update", body.toJSONString());
    System.out.println(sync);
}
```

- 输出结果

```
{"code":200,"data":{"name":"123","resourcesList":[],"uid":"5cb02e89999a0713781782a8"},"id":"123","message":"OK","stat
```

- 测试结果：当resourcesList错误时，依然能够更新成功。device_list字段无法更新