

- **LED2接口测试文档**
 - **LED2Device接口测试文档**
 - 分页查询设备
 - 根据ID查询设备（未开发此接口）
 - 添加设备
 - 修改设备
 - 删除设备
 - 设备开屏
 - 设备关屏
 - 截屏
 - 硬件状态
 - 调节设备亮度
 - 查看设备网络类型
 - 更新设备APP版本
 - 更新设备参数设置
 - 设备重启
 - 设备加锁
 - 自动亮度
 - 设备同步
 - **LED2DeviceGroup接口测试文档**
 - 分页查询分组列表
 - 根据ID查询设备分组列表
 - 添加设备分组
 - 删除设备分组
 - 更新设备分组
 - **LED2Program接口文档**
 - 分页查询节目
 - 根据ID查询节目
 - 添加节目
 - 修改节目
 - 删除节目
 - 根据审核状态查询节目
 - 提交节目审核结果
 - **LED2Task**
 - 分页查询任务列表
 - 根据ID查询任务
 - 添加任务
 - 修改任务
 - 删除任务
 - 任务绑定设备
 - 任务发布
 - **LED2Task（定时亮度与定时开关屏）接口测试文档**
 - 根据ID查询定时亮度任务
 - 分页查询定时亮度任务
 - 添加定时亮度任务
 - 修改定时亮度任务
 - 删除定时亮度任务

- 开启定时亮度任务
- 关闭定时亮度任务
- 根据ID查询定时开关屏任务
- 分页查询定时开关屏任务
- 添加定时开关屏任务
- 修改定时开关屏任务
- 删除定时开关屏任务
- 开启定时开关屏任务
- 关闭定时开关屏任务
- 任务绑定设备
- 任务解除绑定设备
- LED2Resources接口测试文档
 - 分页查询资源
 - 根据UID查询资源
 - 根据资源名称查询资源URL（暂未完成）
 - 添加资源
 - 删除资源
 - 下载资源（暂未完成）
 - 查询资源分组列表
 - 根据资源分组id查询资源
 - 添加资源分组
 - 删除资源分组
 - 更新资源分组

LED2接口测试文档

LED2Device接口测试文档

分页查询设备

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

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

- 输出结果

```
{
  "code":200,
  "data":[
    {
      "screenState":1,
      "addUser":"yunboqun",
      "addDate":"2019-04-25 14:37:25",
      "uid":"5bffc9eb4cb28b3020053833",
      "lightPole":"59fa6bfa2773e41c80837892",
      "cardId":"y30-a18-00269",
      "screenLight":64,
      "name":"主模块",
      "width":"128",
      "online":0,
      "lock":0,
      "taskName":"",
      "brightnessTaskId":"",
      "taskId":"",
      "height":"160"
    }
  ],
  "id":"123",
  "message":"OK",
  "status":"SUCCESS"
}
```

- 接口测试用例

参数	方法返回（无data）	结论
skip/limit为负数	{"code":200,"id":"123","message":"OK","status":"SUCCESS"}	错误
skip和limit为正常取值	{"code":200,"id":"123","message":"OK","status":"SUCCESS"}	正确

- 接口测试结果：skip和limit出现负数的情况不应返回success，limit取值大小无限制，建议最大值为100，统一规范，分页查询无总数统计字段。

根据ID查询设备（未开发此接口）

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

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：

添加设备

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

```
@Test
public void dev_add() {
    JSONObject body = new JSONObject();
    body.put("cardId", "123");
    body.put("name", "1234");
    body.put("id", "123");
    body.put("height", "123");
    body.put("width", "123");
    body.put("lightPole", "123");
    String sync = getClient().sync("led2/v1_0/dev_add", body.toJSONString());
    System.out.println(sync);
}
```

- 输出结果

```
{
  "code":200,
  "data":{
    "uid":"5d552e40560d65403c530b3d",
    "lightPole":"123",
    "addUser":"root",
    "cardId":"123",
    "name":"1234",
    "width":"123",
    "addDate":1565863488685,
    "height":"123"
  },
  "id":"123",
  "message":"OK",
  "status":"SUCCESS"
}
```

- 接口测试用例

参数	方法返回（无data）	结论
cardId重复	success	错误
lightPole不存在	success	错误
width/height取值范围没有控制	success	错误
width/height没有校验只能为数字	success	错误

- 接口测试结果： cardId, lightPole, width, height没有规则校验

修改设备

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

```
@Test
public void dev_edit() {
    JSONObject body = new JSONObject();
    body.put("cardId", "123");
    body.put("name", "1234");
    body.put("id", "123");
    body.put("height", "123");
    body.put("width", "123");
    body.put("lightPole", "123");
    String sync = getClient().sync("led2/v1_0/dev_edit", body.toJSONString());
    System.out.println(sync);
}
```

- 输出结果

```
{
  "code":200,
  "data":{
    "uid":"5d552e40560d65403c530b3d",
    "lightPole":"123",
    "addUser":"root",
    "cardId":"123",
    "name":"1234",
    "width":"123",
    "addDate":1565863488685,
    "height":"123"
  },
  "id":"123",
  "message":"OK",
  "status":"SUCCESS"
}
```

- 接口测试用例

参数	方法返回（无data）	结论
cardId重复	success	错误
lightPole不存在	success	错误
width/height取值范围没有控制	success	错误
width/height没有校验只能为数字	success	错误

- 接口测试结果：cardId，lightPole，width，height没有规则校验

删除设备

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

```
@Test
public void dev_del() {
    JSONObject body = new JSONObject();
    body.put("uid", "5d552f3a560d65403c530b40");
    body.put("id", "123");
    String sync = getClient().sync("led2/v1_0/dev_del", body.toJSONString());
    System.out.println(sync);
}
```

- 输出结果

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

- 接口测试用例

参数	方法返回（无data）	结论
uid正确	success	正确
uid错误	{"code":20004,"id":"123","message":"Device does not exist","status":"NOT_FOUND_PARAM"}	正确

- 接口测试结果：接口测试通过

设备开屏

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

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：

设备关屏

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

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：

截屏

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

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：

硬件状态

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

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：

调节设备亮度

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

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：

查看设备网络类型

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

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：

更新设备APP版本

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

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：

更新设备参数设置

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

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：

设备重启

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

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：

设备加锁

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

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：

自动亮度

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

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：

设备同步

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

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：

LED2DeviceGroup接口测试文档

分页查询分组列表

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

```
@Test
public void dev_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/dev_group_query", body.toJSONString());
    System.out.println(sync);
}
```

- 输出结果

```
{
  "code":200,
  "data":{
    "deviceGroups":[
      {
        "uid":"5c7779ac999a071a205f96c4",
        "modifyUser":"",
        "modifyDate":"",
        "addUser":"",
        "name":"测试分组1",
        "deviceList":[
          "5c7cdbe4999a0722787d1b36",
          "5bffc9eb4cb28b3020053833"
        ],
        "addDate":""
      }
    ],
    "total":1
  },
  "id":"123",
  "message":"OK",
  "status":"SUCCESS"
}
```

- 接口测试用例

参数	方法返回（无data）	结论
skip/limit为负数	{"code":200,"id":"123","message":"OK","status":"SUCCESS"}	错误
skip和limit为正常取值	{"code":200,"id":"123","message":"OK","status":"SUCCESS"}	正确

- 接口测试结果：skip和limit出现负数的情况不应返回success，limit取值大小无限制，建议最大值为100，统一规范。

根据ID查询设备分组列表

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

```
@Test
public void dev_by_groupId() {
    JSONObject body = new JSONObject();
    body.put("uid", "5c7779ac999a071a205f96c4");
    body.put("id", "123");
    String sync = getClient().sync("led2/v1_0/dev_by_groupId", body.toJSONString());
    System.out.println(sync);
}
```

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

参数	方法返回（无data）	结论
uid不正确	{"code":200,"data":{},"id":"123","message":"OK","status":"SUCCESS"}	正确

- 接口测试结果：填写错误uid时，接口正确，但是uid正确时，接口内部业务逻辑错误。

添加设备分组

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

```
@Test
public void dev_group_add() {
    JSONObject body = new JSONObject();
    body.put("devices", "5bffc9eb4cb28b3020053833");
    body.put("name", "123");
    body.put("id", "123");
    String sync = getClient().sync("led2/v1_0/dev_group_add", body.toJSONString());
    System.out.println(sync);
}
```

- 输出结果

```
{
  "code":20004,
  "id":"123",
  "message":"NOT FOUND DEVICE [5bffc9eb4cb28b3020053833]",
  "status":"NOT_FOUND_PARAM"
}
```

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：Led2DeviceGroupServiceImpl类中第158行代码有问题

删除设备分组

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

```
@Test
public void dev_group_del() {
    JSONObject body = new JSONObject();
    body.put("uid", "5d554809560d65403c530b41");
    body.put("id", "123");
    String sync = getClient().sync("led2/v1_0/dev_group_del", body.toJSONString());
    System.out.println(sync);
}
```

- 输出结果

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

- 接口测试用例

参数	方法返回（无data）	结论
uid错误	{"code":20010,"id":"123","message":"失败","status":"FAILURE"}	错误
uid正确	{"code":200,"id":"123","message":"OK","status":"SUCCESS"}	正确

- 接口测试结果：当uid不存在时，返回code不正确，不应为20010.

更新设备分组

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

```
@Test
public void dev_group_update() {
    JSONObject body = new JSONObject();
    body.put("uid", "5d2d68526c1152184855b39f");
    body.put("id", "123");
    body.put("name", "123");
    body.put("devices", "5cb695cbf22affbf32eca9a4");
    String sync = getClient().sync("led2/v1_0/dev_group_update", body.toJSONString());
    System.out.println(sync);
}
```

- 输出结果

```
{"code":20004,"id":"123","message":"DEVICE_GROUP DOES NOT EXIST!","status":"NOT_FOUND_PARAM"}
```

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：无论devices为何值，返回都是查询不到设备，Led2DeviceGroupServiceImpl类的220行有错误。

LED2Program接口文档

分页查询节目

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

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：

根据ID查询节目

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

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：

添加节目

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

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：

修改节目

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

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：

删除节目

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

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：

根据审核状态查询节目

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

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：

提交节目审核结果

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

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：

LED2Task

分页查询任务列表

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

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：

根据ID查询任务

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

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：

添加任务

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

- 请求方式：POST
- 测试示例

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：

修改任务

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

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：

删除任务

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

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：

任务绑定设备

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

- 请求方式：POST
- 测试示例

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：

任务发布

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

- 输出结果

- 接口测试用例

参数	方法返回（无data）	结论
----	-------------	----

- 接口测试结果：

LED2Task（定时亮度与定时开关屏）接口测试文档

根据ID查询定时亮度任务

分页查询定时亮度任务

添加定时亮度任务

修改定时亮度任务

删除定时亮度任务

开启定时亮度任务

关闭定时亮度任务

根据ID查询定时开关屏任务

分页查询定时开关屏任务

添加定时开关屏任务

修改定时开关屏任务

删除定时开关屏任务

开启定时开关屏任务

关闭定时开关屏任务

任务绑定设备

任务解除绑定设备

LED2Resources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](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","status":"SUCCESS"}
```

- 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