

- **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
 - 提交节目审核结果
 - **LED2Task**
 - 分页查询任务列表
 - 根据ID查询任务（未开发此接口）
 - 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
- 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]/screen_open
- 请求方式：POST
- 测试示例

```
@Test
public void screen_open() {
    JSONObject body = new JSONObject();
    body.put("uid", "5d3818626c115218f829bf76");
    body.put("id", "123");
    String sync = getClient().sync("led2/v1_0/screen_open", 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不正确	{"code":20004,"id":"123","message":"Device does not exist","status":"NOT_FOUND_PARAM"}	正确

- 接口测试结果：测试通过

设备关屏

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

```
@Test
public void screen_close() {
    JSONObject body = new JSONObject();
    body.put("uid", "5d3818626c115218f829bf76");
    body.put("id", "123");
    String sync = getClient().sync("led2/v1_0/screen_close", 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不正确	{"code":20004,"id":"123","message":"Device does not exist","status":"NOT_FOUND_PARAM"}	正确

- 接口测试结果：测试通过

截屏

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

```
@Test
public void screen_capture() {
    JSONObject body = new JSONObject();
    body.put("uid", "5d3818626c115218f829bf76");
    body.put("id", "123");
    String sync = getClient().sync("led2/v1_0/screen_capture", body.toJSONString());
    System.out.println(sync);
}
```

- 输出结果

```
{
  "code":200,
  "data":{
    "src":"/9j/4AAQSkZJRgABAQAAQABAAD/2wBDAAEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQH/2
  },
  "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":20004,"id":"123","message":"DEVICE DOES NOT EXIST!","status":"NOT_FOUND_PARAM"}	正确

- 接口测试结果：测试通过

硬件状态

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

```
@Test
public void hardware_status() {
    JSONObject body = new JSONObject();
    body.put("uid", "5d3818626c115218f829bf76");
    body.put("id", "123");
    String sync = getClient().sync("led2/v1_0/hardware_status", body.toJSONString());
    System.out.println(sync);
}
```

- 输出结果

```
{
  "code":200,
  "data":{
    "externalVoltage1":"0.0V",
    "externalVoltage2":"0.0V",
    "cardVoltage":"4.86V",
    "width":3,
    "smoke":"Normal",
    "temperature":36.0℃,
    "humidity":"0.0%",
    "doorOpened":"Open",
    "version":"97bc",
    "height":4
  },
  "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不正确	{"code":20004,"id":"123","message":"Device does not exist","status":"NOT_FOUND_PARAM"}	正确

- 接口测试结果：设备上报数据出现乱码。

调节设备亮度

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

```
@Test
public void adjust_light() {
    JSONObject body = new JSONObject();
    body.put("uid", "5d3818626c115218f829bf76");
    body.put("id", "123");
    body.put("lightValue", "123456");
    String sync = getClient().sync("led2/v1_0/adjust_light", 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":20004,"id":"123","message":"DEVICE DOES NOT EXIST!","status":"NOT_FOUND_PARAM"}	正确
lightValue错误取值	{"code":200,"id":"123","message":"OK","status":"SUCCESS"}	错误
lightValue出现非数字	报错	错误
uid不在线	{"code":20006,"id":"123","message":"设备离线","status":"DEVICE_OFFONLINE"}	正确

- 接口测试结果：本接口开发文档中URL链接错误，望改正。限制接口调用方输入lightValue的范围。

查看设备网络类型

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

```
@Test
public void network_query() {
    JSONObject body = new JSONObject();
    body.put("uid", "5d3818626c115218f829bf76");
    body.put("id", "123");
    String sync = getClient().sync("led2/v1_0/network_query", body.toJSONString());
    System.out.println(sync);
}
```

- 输出结果

```
{
  "code":200,
  "data":{
    "network":"ETH"
  },
  "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":20004,"id":"123","message":"Device does not exist","status":"NOT_FOUND_PARAM"}	正确
uid正确	success	正确

- 接口测试结果：接口测试通过

更新设备APP版本

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

```
@Test
public void version_update() {
    JSONObject body = new JSONObject();
    body.put("uid", "5d3818626c115218f829bf76");
    body.put("id", "123");
    body.put("appUrl", "123");
    String sync = getClient().sync("led2/v1_0/version_update", 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
appUrl不合法	success	成功

- 接口测试结果：appUrl需要与B/S端一样通过格式校验

更新设备参数设置

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

```
@Test
public void config_adv() {
    JSONObject body = new JSONObject();
    body.put("uid", "5d3818626c115218f829bf76");
    body.put("id", "123");
    body.put("serverURL", "123");
    body.put("companyId", "123");
    body.put("realtimeURL", "123");
    body.put("usbProgramPwd", "123");
    String sync = getClient().sync("led2/v1_0/config_adv", 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
usbProgramPwd参数缺失	{"code":200,"id":"123","message":"OK","status":"SUCCESS"}	错误
serverURL和companyId和realtimeURL和usbProgramPwd无校验逻辑	success	错误

- 接口测试结果：返回体中应包括更新的内容。

设备重启

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

```
@Test
public void dev_reboot() {
    JSONObject body = new JSONObject();
    body.put("uid", "5d3818626c115218f829bf76");
    body.put("id", "123");
    String sync = getClient().sync("led2/v1_0/dev_reboot", 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":20004,"id":"123","message":"Device does not exist","status":"NOT_FOUND_PARAM"}	正确
uid正确	success	正确

- 接口测试结果：接口测试通过

设备加锁

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

```
@Test
public void dev_lock() {
    JSONObject body = new JSONObject();
    body.put("uid", "5d3818626c115218f829bf76");
    body.put("id", "123");
    body.put("lock", "1");
    String sync = getClient().sync("led2/v1_0/dev_lock", body.toJSONString());
    System.out.println(sync);
}
```

- 输出结果

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

- 接口测试用例

参数	方法返回（无data）	结论
lock为取值范围之外	{"code":200,"data": {"code":20003,"message":"参数校验失败","status":"BAD_PARAM"},"id":"123","message":"OK","status":"SUCCESS"}	错误
当设备处于无锁状态， lock=0调用接口	{"code":200,"data":{},"id":"123","message":"OK","status":"SUCCESS"}	错误
当设备处于无所状态时， lock=1调用接口	数据库字段没有改变	错误
当设备不在线	{"code":200,"data":{},"id":"123","message":"OK","status":"SUCCESS"}	错误

- 接口测试结果：没有判断设备当前是否处于有锁或无锁状态，数据库字段没有改变。当设备不在线时，不应返回success。

自动亮度

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

```
@Test
public void dev_autoBrightness() {
    JSONObject body = new JSONObject();
    body.put("uid", "5d3818626c115218f829bf76");
    body.put("id", "123");
    body.put("sensitivity", "111111");
    body.put("minBrightness", "111111");
    String sync = getClient().sync("led2/v1_0/dev_autoBrightness", 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
sensitivity和minBrightness无大小限制	success	错误
sensitivity和minBrightness不是数字	报错	错误

- 接口测试结果：设备不在线不应返回成功,返回体中应包括下发指令的参数。

设备同步

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

```
@Test
public void dev_sync() {
    JSONObject body = new JSONObject();
    body.put("uid", "5d3818626c115218f829bf76");
    body.put("id", "123");
    body.put("time", "111qa111");
    body.put("brightness", "111qa111");
    body.put("volume", "111qa111");
    body.put("screenSwitch", "111qa111");
    body.put("delaySync", "111qa111");
    body.put("checkNtpTime", "111qa111");
    String sync = getClient().sync("led2/v1_0/dev_sync", 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
----	-------------	----

- 接口测试结果：所有的参数都无校验逻辑。

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
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]/prog_query
- 请求方式：POST
- 测试示例

```
@Test
public void prog_query() {
    JSONObject body = new JSONObject();
    body.put("skip", "0");
    body.put("limit", "1");
    body.put("id", "123");
    body.put("validate", 0);
    String sync = getClient().sync("led2/v1_0/prog_query", 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
----	-------------	----

- 接口测试结果：DeviceDaoImpl类141行代码逻辑有问题

根据ID查询节目

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

```
@Test
public void prog_query_byId() {
    JSONObject body = new JSONObject();
    body.put("uid", "5c88c4f5999a07202436e51b");
    body.put("id", "123");
    String sync = getClient().sync("led2/v1_0/prog_query_byId", body.toJSONString());
    System.out.println(sync);
}
```

- 输出结果


```
{
  "code":200,
  "data":{
    "modifyDate":"","
    "addUser":"yunboqun",
    "checkDate":"2019-07-29 14:38:32",
    "addDate":"2019-05-22 17:25:01",
    "uid":"5c88c4f5999a07202436e51b",
    "checkUser":"yunboqun",
    "modifyUser":"","
    "name":"测试文字节目7",
    "width":"128",
    "failReason":"1",
    "config":{
      "totalSize":32999439,
      "name":"","
      "width":0,
      "layers":[
        {
          "sources":[
            {
              "entryEffectTimeSpan":0,
              "_type":"SingleLineText",
              "exitEffect":"MOVING_LEFT",
              "entryEffect":"MOVING_LEFT",
              "speed":20,
              "top":0,
              "left":0,
              "name":"SingleText",
              "width":98,
              "playTime":1,
              "timeSpan":233,
              "html":"<span style='font-family:FangSong_GB2312;font-size:40px;color: #ffa500;background: #143643'><b><i>测
              "id":"","
              "exitEffectTimeSpan":1,
              "height":77
            }
          ],
          "repeat":false
        }
      ],
      "_id":"","
      "height":0
    },
    "height":"160",
    "validate":2
  },
  "id":"123",
  "message":"OK",
  "status":"SUCCESS"
}
```

- 接口测试用例

参数	方法返回（无data）	结论
id和uid正确	success	正确
uid不正确	success	正确

- 接口测试结果：返回结果结构太复杂，接口返回不需要这么复杂，建议按照同级结构返回字段。

添加节目

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

```
@Test
public void prog_del() {
    JSONObject body = new JSONObject();
    body.put("uid", "5d1c226dcc5c862ce055c6a0");
    body.put("id", "123");
    String sync = getClient().sync("led2/v1_0/prog_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不正确	{"code":20004,"id":"123","message":"PROGRAM DOES NOT EXIST!","status":"NOT_FOUND_PARAM"}	正确
uid正确	{"code":200,"id":"123","message":"OK","status":"SUCCESS"}	正确

- 接口测试结果：测试通过

提交节目审核结果

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

@Test

```
public void prog_check() {
    JSONObject body = new JSONObject();
    body.put("uid", "5c88c4f5999a07202436e51b");
    body.put("id", "123");
    body.put("validate", "1");
    body.put("failReason", "123");
    String sync = getClient().sync("led2/v1_0/prog_check", 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
uid不存在	{"code":200,"id":"123","message":"OK","status":"SUCCESS"}	错误
validate取值不合法	{"code":200,"id":"123","message":"OK","status":"SUCCESS"}	错误
validate不是数字	报错	错误

- 接口测试结果：接口校验不通过，而且当节目已经被审核后还能继续修改审核，应先判断节目是否被审核过

LED2Task

分页查询任务列表

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

@Test

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

- 输出结果

```
{
  "code":200,
  "data":{
    "tasks":[
      {
        "uid":"5d3fe7c9f22affc6027ad9bc",
        "modifyUser":"yunboqun",
        "task":{
          "items":[
            {
              "programId":"5d3fe7b2f22affc6027ad9bb",
              "repeatTimes":"100",
              "schedules":[
                {
                  "dateType":"All",
                  "endDate":"","",
                  "timeType":"All",
                  "startTime":"","",
                  "endTime":"","",
                  "filterType":"None",
                  "startDate":"","",
                  "weekFilter":[]
                }
              ]
            }
          ]
        }
      ],
      "modifyDate":"2019-08-14 18:30:56",
      "addUser":"yunboqun",
      "name":"文字啊啊啊啊啊啊",
      "deviceList":[
        "5d3818626c115218f829bf76"
      ],
      "programList":[
        "5d3fe7b2f22affc6027ad9bb"
      ],
      "sceneList":[]
    ],
    "addDate":"2019-07-30 14:46:33"
  },
  "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	全量数据返回	错误

- 接口测试结果：skip和limit取值范围校验。返回体结构太复杂。

根据ID查询任务（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：

添加任务

- 接口链接：URL地址：http://[域名]/app/led2/[version]/task_add
- 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：建议添加时不要让接口调用方拼凑数据结构体，数据格式在业务中自己拼装。

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：建议不要让接口调用方拼凑数据结构体，数据格式在业务中自己拼装。

删除任务

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

```
@Test
public void task_del() {
    JSONObject body = new JSONObject();
    body.put("id", "123");
    body.put("uid", "5d3fe7c9f22affc6027ad9bc");
    String sync = getClient().sync("led2/v1_0/task_del", body.toJSONString());
    System.out.println(sync);
}
```

- 输出结果

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

- 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

- 接口测试结果：Led2TaskServiceImpl类457行逻辑错误。

任务绑定设备

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

```
@Test
public void task_bind_dev() {
    JSONObject body = new JSONObject();
    body.put("id", "123");
    body.put("uid", "5d3fe7c9f22affc6027ad9bc");
    body.put("devices", "5c7cdbe4999a0722787d1b36");
    String sync = getClient().sync("led2/v1_0/task_bind_dev", body.toJSONString());
    System.out.println(sync);
}
```

- 输出结果

```
{
  "code":200,
  "data":{"
    "deviceList":[
      "5c7cdbe4999a0722787d1b36"
    ],
    "uid":"5d3fe7c9f22affc6027ad9bc"
  },
  "id":"123",
  "message":"OK",
  "status":"SUCCESS"
}
```

- 接口测试用例

参数	方法返回（无data）	结论
devices不存在	{"code":20003,"id":"123","message":"DEVICE_ID NOT FOUND! DEVICE_ID: 5c7cdbe4999a0722787111d1b36","status":"BAD_PARAM"}	正确
devices多个时没有去重	success	错误
devices每一次调用会覆盖数据库之前的数据	success	错误

- 接口测试结果：建议绑定设备为原子操作，一次只能绑定一个设备，业务逻辑有问题，不应该覆盖devices的数据。

任务发布

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

```
@Test
public void task_publish() {
    JSONObject body = new JSONObject();
    body.put("id", "123");
    body.put("uid", "5d3fe7c9f22affc6027ad9bc");
    String sync = getClient().sync("led2/v1_0/task_publish", body.toJSONString());
    System.out.println(sync);
}
```

- 输出结果

```
{
  "code":200,
  "data":{
    "failureDevList":[
      "y30-718-00580"
    ],
    "successDevList":[

    ]
  },
  "id":"123",
  "message":"OK",
  "status":"SUCCESS"
}
```

- 接口测试用例

参数	方法返回（无data）	结论
当任务绑定的设备都不在线时	success	错误

- 接口测试结果：当任务绑定的设备都不在线时不应返回正确，应给接口调用方提示准确信息。

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`
- 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, "Msg": "Invalid s
```

- 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":"5cb02e8999a0713781782a8"},"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