

信发平台 开放接口 说明文档

v2.1

2020 年 07 月 30 日

修订文档历史记录

日期	版本	说明
2020-04-22	1.0	开发接口初版
2020-04-26	1.4	后台增加 debugmode 模式
2020-04-28	1.5	修复一些 bug
2020-05-07	1.6	制作广告接口增加背景音乐（详见 3.10） 服务器版本 ≥ 2.46
2020-05-16	1.7	新增素材管理接口（详见 3.16-3.19）；另广告编辑的素材 mh 以及 id 有新处理（此前代码不修改也兼容）服务器版本 ≥ 2.47
2020-06-01	1.8	3.3 获取设备接口返回值增加 online、reg 字段（在线数量和已录入） 新增获取广告场景数据接口（3.20） 查询设备、素材、广告接口增加条件查询功能 增加 logo 台标设置接口（3.22）
2020-06-17	1.9	广告编辑接口增加文本元素（详见 3.10 文本元素相关）
2020-07-08	2.0	新增设备分组相关、撤销广告、跑马灯相关接口
2020-07-30	2.1	新增休眠唤醒接口

目录

1、获取 access_token 、 access_secret.....	5
2、设备登录、管理.....	6
3、开放接口.....	6
3.1 开发约定.....	6
3.2 第三方获取设备 cMark	6
3.3 查询当前用户设备总数.....	7
3.4 批量获取设备标识.....	8
3.5 批量获取设备状态.....	9
3.6 查询单设备状态.....	11
3.7 录入设备/删除设备	12
3.8 获取广告数量.....	13
3.9 获取广告列表.....	13
3.10 新增/编辑广告	15
3.11 审核广告.....	21
3.12 删除广告.....	22
3.13 发布广告计划.....	22
3.14 获取单设备当前广告列表.....	25
3.15 撤销广告.....	27
3.16 获取素材数量.....	28
3.17 获取素材列表.....	29
3.18 删除素材.....	31
3.19 上传素材.....	32
3.20 重启设备.....	33
3.21 设备改名.....	34
3.22 设备设置 LOGO	35
3.23 新建/编辑设备分组	36
3.24 查询设备分组数量.....	37
3.25 查询设备分组列表信息.....	38
3.26 查询单个设备分组信息.....	39
3.27 删除设备分组信息.....	40
3.28 查询设备分组的设备数量.....	41
3.29 查询设备分组的设备 cMarks	42
3.30 添加设备到分组/从分组移除设备	43
3.31 查询跑马灯数量.....	44
3.32 查询跑马灯列表信息.....	45
3.33 查询单个跑马灯信息.....	46
3.34 新增/修改跑马灯	47
3.35 删除跑马灯.....	48
3.36 发布跑马灯.....	49
3.37 撤销跑马灯.....	50
3.38 休眠、唤醒.....	51
3.39 休眠、唤醒计划.....	52

4、错误码&原因说明	54
5、结束	55

2、设备登录、管理

请确保前端设备已成功连接服务器

3、开放接口

3.1 开发约定

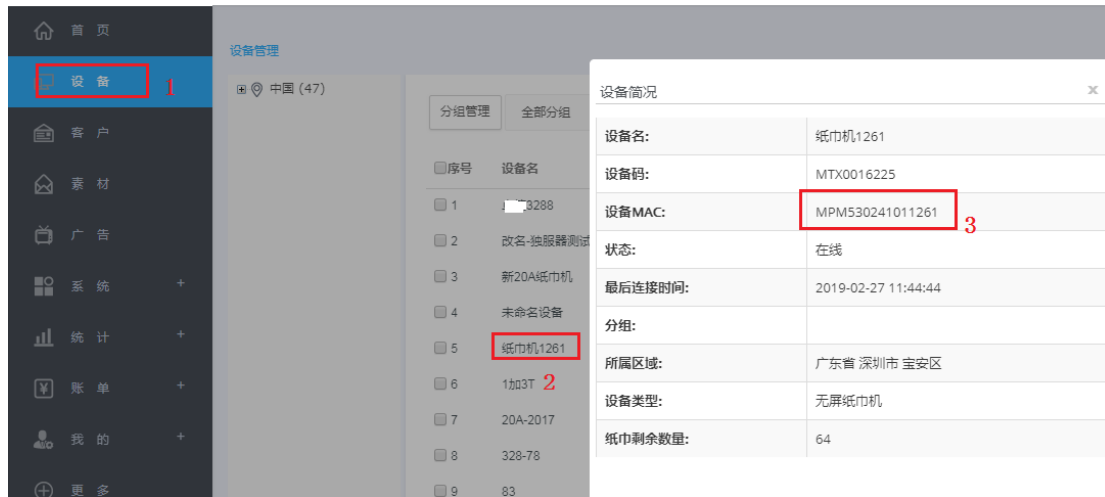
- 1、使用 HTTP 协议进行信息交互，字符编码统一采用 UTF-8 格式
- 2、HTTP 请求方式：post
- 3、服务器返回结果：json 格式，返回的 json 数据键值对可能有变化增减（retCode 必然存在），第三方应用为程序健壮计，取值时最好判断对应字段值
- 4、参数表中，类型一栏声明的定义 int 代表整数类型，string 代表字符串类型
- 5、参数大小写敏感
- 6、接口 URL http://server:port/adc/sdk/中，其中 server 是服务器地址，port 是 web 端口

3.2 第三方获取设备 cMark

每一台前端设备都有一个唯一标识符(不是 MTX 打头的编码)，本系统中统一以 cMark 表示。

有两种方式可以获取设备的 cMark 值，用户可根据实际需求选择使用

- 1、你可以通过 web 平台查询设备 cMark 值，如下图示：



2、通过开放接口批量获取（详见 3.3 & 3.4 章节，用法自行决定）

3.3 查询当前用户设备总数

接口：

`http://server:port/adc/sdk/dev_get_devcount`

应用参数说明：

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳，秒 since 1970
sign	String	Y	接口调用签名
state	int	N	可选，无此参数则录入未录入都查询； state=1 录入、非 1 则未录入
select_name	string	N	检索条件，可为设备名和设备码（模糊搜索）

sign 的生成方式参考 js

```
//用户token
var access_token = "9657d408-cb99-11e8-89642-00163e046215";

//用户secret，这里仅仅为了演示调用方法直接暴露了，很危险，开发人员请慎重，勿明文显示暴露
var access_secret = "a733613";

// hex_md5是取字符串md5值
var acces_secret_md5 = hex_md5(access_secret);
```

//unix时间戳 :秒 since 1970, 请校准信发服务器时间, 差值过大接口会返回失败

```
var timestamp = getTimeStamp();
```

```
var sign = hex_md5(access_token+acces_secret_md5+timestamp);
```

返回说明

正确调用后的返回示例:

```
{
  "result": "success",
  "count": 2,
  "retInfo": "OK",
  "retCode": 0
}
```

参数说明:

名称	类型	描述
retCode	int	0 表示成功, 其他错误码
retInfo	String	错误描述
count	int	当前用户所有的设备数量 (含录入与未录入)
online	int	符合条件的在线的设备数量
reg	int	符合条件的已录入数量

3.4 批量获取设备标识

接口:

http://server:port/adcsdk/dev_get_devs_cmarks

应用参数说明:

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳, 秒 since 1970
sign	String	Y	接口调用签名
state	int	N	可选, 无此参数则录入未录入都查询; state=1 录入、非 1 则未录入
startIndex	int	Y	起始下标 (范围 0 ~count-1)
queryCount	int	Y	单次查询数量 (max 500)

select_name	string	N	检索条件，可为设备名或设备码（模糊搜索）
-------------	--------	---	----------------------

返回说明

正确调用后的返回示例：



参数说明：

名称	类型	描述
retCode	int	0 表示成功，其他错误码
retInfo	String	错误描述
count	int	返回查询到的数量
cMarkList	List	查询的的设备标识列表

3.5 批量获取设备状态

接口：

http://server:port/adk/sdk/dev_get_devs_info

应用参数说明：

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳，秒 since 1970
sign	String	Y	接口调用签名
cMarks	String	Y	设备组的字符串，cMark 之间用英文字符逗号分割，最大限制单次请求 500 个 cMark

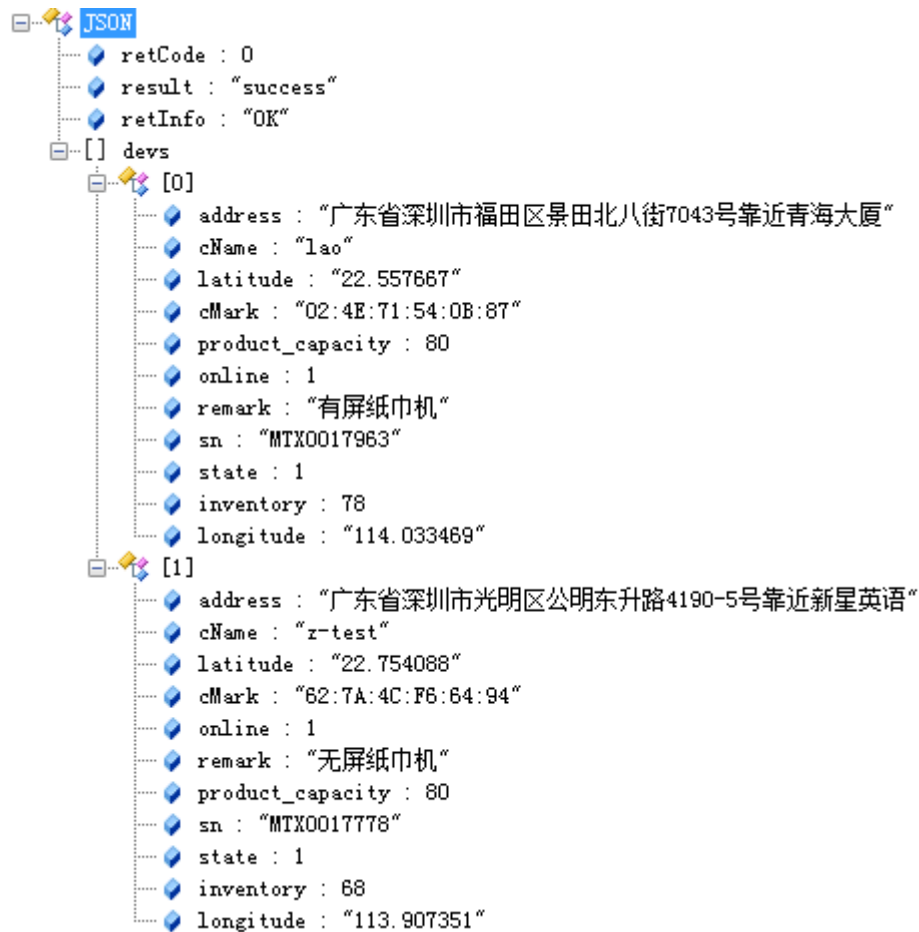
请求 post 参数举例

```
{
```

```
"access_token": "3bvc5f282cba611e8aaa2-00163e046949",
"timestamp": 158323292939,
"sign": "adsfk sdfkssweirweiioewrwoi",
"cMarks": "02:7A:4C:F6:64:94, 02:4E:71:54:0B:87"
}
```

返回说明

正确调用后的返回示例:



```
{
  "retCode": 0,
  "result": "success",
  "retInfo": "OK",
  "devs": [
    {
      "address": "广东省深圳市福田区景田北八街7043号靠近青海大厦",
      "cName": "lao",
      "latitude": "22.557667",
      "cMark": "02:4E:71:54:0B:87",
      "product_capacity": 80,
      "online": 1,
      "remark": "有屏纸巾机",
      "sn": "MTX0017963",
      "state": 1,
      "inventory": 78,
      "longitude": "114.033469"
    },
    {
      "address": "广东省深圳市光明区公明东升路4190-5号靠近新星英语",
      "cName": "z-test",
      "latitude": "22.754088",
      "cMark": "62:7A:4C:F6:64:94",
      "online": 1,
      "remark": "无屏纸巾机",
      "product_capacity": 80,
      "sn": "MTX0017778",
      "state": 1,
      "inventory": 68,
      "longitude": "113.907351"
    }
  ]
}
```

参数说明:

名称	类型	描述
retCode	int	0 表示成功，其他错误码
retInfo	String	错误描述
devs	List	设备列表
longitude	String	设备经度， retCode==0 时有效
latitude	String	设备纬度， retCode==0 时有效
online	Int	0:不在线; 1: 在线 retCode==0 时有效
state	Int	0:未录入; 1: 已录入 retCode==0 时有效

cMark	String	设备标识
sn	String	设备码
address	String	设备地址

3.6 查询单设备状态

接口:

`http://server:port/adc/sdk/dev_get_devstatus`

应用参数说明:

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳, 秒 since 1970
sign	String	Y	接口调用签名
cMark	String	Y	设备的标识

返回说明

正确调用后的返回示例:

```
{
  "result": "success",
  "address": "华夏二路富森大厦",
  "latitude": "25.156638",
  "online": 0,
  "retInfo": "OK",
  "state": 1,
  "sn": "MTX8007029",
  "retCode": 0,
  "longitude": "128.459289"
}
```

参数说明:

名称	类型	描述
retCode	int	0 表示成功, 其他错误码
retInfo	String	错误描述
longitude	String	设备经度, retCode==0 时有效
latitude	String	设备纬度, retCode==0 时有效

online	Int	0:不在线; 1: 在线 retCode==0 时有效
state	Int	0:未录入; 1: 已录入 retCode==0 时有效, 未录入的设备不会显示在设备列表, 需先录入才能管理控制
sn	string	设备码

备注：单台设备查询时建议用此接口，效率高

3.7 录入设备/删除设备

接口：

`http://server:port/adc/sdk/dev_register_dev`

应用参数说明：

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳, 秒 since 1970
sign	String	Y	接口调用签名
cMarks	String	Y	设备的标识, 多个以英文逗号分割
state	Int	Y	录入 or 删除 1: 录入、0: 删除

返回说明

正确调用后的返回示例：

```
{"result": "success", "retInfo": "OK", "retCode": 0}
```

参数说明：

名称	类型	描述
retCode	int	0 表示成功, 其他错误码
retInfo	String	错误描述

3.8 获取广告数量

接口：

`http://server:port/adc/sdk/advert_get_itemcount`

应用参数说明：

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳，秒 since 1970
sign	String	Y	接口调用签名
select_name	string	N	可选，可输入广告名搜索（模糊搜索）

返回说明

正确调用后的返回示例：

```
{  "result": "success",
  "retInfo": "success",
  "retCode": 0,
  "itemCount": 1057
}
```

参数说明：

名称	类型	描述
retCode	int	0 表示成功，其他错误码
retInfo	String	错误描述
itemCount	int	当前用户的所有广告数量（按创建日期降序排列）

3.9 获取广告列表

`http://server:port/adc/sdk/advert_get_itemlist`

应用参数说明：

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳，秒 since 1970
sign	String	Y	接口调用签名

select_name	string	N	可选，可输入广告名搜索（模糊搜索）
showCount	int	Y	接口获取数据按分页模式处理，每次获取数据按广告的创建时间降序排列，后台按每页 showCount 数量组织数据返回结果集，每页最多获取条数（500Max）
currentPage	int	Y	第几页数据（每页按 showCount 计，从 1 计数开始）

返回说明

正确调用后的返回示例：

```
{
  "result": "success",
  "showCount": 10,
  "retInfo": "success",
  "retCode": 0,
  "currentPage": 1,
  "itemCount": 2,
  "itemList": [
    {
      "upTime": "2020-04-26 10:11:38",
      "rowNo": 1,
      "itemName": "测试图片视频混播广告 1",
      "needAudit": 1,
      "auditStatus": 0,
      "allFileSize": "9973250",
      "id": "1789861689@88@77@000001",
      "sTime": 40,
      "thumbnails": "http://abc.pvtest.cn:803/Public/Uploads/20190101/01zxxa.jpg",
      "userName": "满天星",
      "resolution": "1920*1080"
    },
    {
      "upTime": "2020-04-26 10:11:37",
      "rowNo": 2,
      "itemName": "测试图片视频混播广告 2",
      "needAudit": 1,
      "auditStatus": 0,
      "allFileSize": "9973250",
      "id": "1789728522@88@77@000001",
      "sTime": 40,
      "thumbnails": "http://www.mtstar.online:6080/adc/i/20190812/_1565578896420682493584@482@77@000001.png",
    }
  ]
}
```

```

        "userName": "满天星",
        "resolution": "1920*1080"
    },
]
}

```

参数说明:

名称	类型	描述
retCode	int	0 表示成功, 其他错误码
retInfo	String	错误描述
showCount	int	用户提交的 showCount 参数, 原路返回以作确认
currentPage	int	用户提交的 currentPage 参数
itemCount	int	返回的广告数量 (按制作日期降序排列)
itemList	Array	返回的结果集
itemList.id	string	广告 id, 发布广告会用到此 id
itemList.sTime	int	广告时长, 单位秒
itemList.thumbnails	string	广告封面缩略图
itemList.needAudit	Int	广告是否需要审核 1: 需审核; 0: 不需要审核
itemList.auditStatus	int	广告是否已审核 1: 已审; 0: 未审核 (特别提示: 审核和后面的广告发布有关联, 如果用户开启了广告审核功能[needAudit==1], 并通过接口来发布广告时, 系统并不检测该广告是否已审核, 而是将具体审核业务流程由接口调用方自行处理)
itemList.allFileSize	long	广告总的文件大小

3.10 新增/编辑广告

http://server:port/adk/sdk/advert_edit_item

应用参数说明:

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳, 秒 since 1970
sign	String	Y	接口调用签名

itemId	String	Y	此参数很重要！当 itemId 为空值时，表示“新增广告”，当 itemId 为有意义的广告 id 值时，表示“编辑”该广告 id 所代表的广告（用新的 scene 值替换之）
scene	String	Y	场景参数，json 格式转换后的字符串，具体参数项目多且复杂，下文以 js 代码示例说明以方便调用者参考

调用示例 js 代码（仅仅是举例，请开发者理解后灵活运用）：

```
function do_test_sdk_edit_advert(){
    //itemId为空，则为新建，为有意义的广告id，则为编辑，其他无意义的值，则会返回失败
    var itemId = "";

    //以两个分屏元素(左右分屏)一个广告举例
    var scene = {};

    /*元素数目，用户如果为素材全屏的节目，一般设置为1，
    **这里为演示接口能力，故意设置左右2分屏（另外再加一个不可见的音乐元素）
    **更多分屏，用户可自行理解处理
    **/
    scene.elementcount = 3;

    //广告名称
    scene.scenename = "测试图片视频混播广告";

    //时长，秒；这个参数很重要,限定场景时间，这个时间是这个广告播放的时间
    //假如场景内存在100秒的视频，若此值为40，则后60秒视频不会播
    //一般情况用户根据广告内元素完整展示一轮的最小时长来给
    scene.scenetime = 40; //很重要，整个场景播放时间，

    /**
     * 以下为广告节目屏幕分辨率参数，此参数具有设备无关性---和前端设备实际屏幕分辨率大小无关，
     * 包括后面的元素位置仅仅为广告的“设计空间”坐标系参数，
     * 实际播放时会等比例映射到是设备的实际屏幕坐标系上去
     */
    //固定，自定义设计坐标系分辨率
    scene.screen_orientation = 2;
    scene.width = 1920;
    scene.height = 1080;

    //广告的封面图片，类似缩略图概念，无此值，则默认用元素1的素材1的第一张图缩略图作为广告缩略图
```



```
scene.coverpage = "http://abc.pytest.cn:803/Public/Uploads/20190101/01zxaxa.jpg";

//以下元素参数
scene.datalist = new Array();

//元素1
var element1 = {};

// x、y、width、height 元素位置坐标，用户自行设置正确空间位置，请勿溢出
element1.x = 0;
element1.y = 0;
element1.width = 960;
element1.height = 1080;

element1.delay = 5;
element1.curimglist = new Array();
//素材1，假定为 图片
var mat1 = {};
mat1.type = "img";// img 表示图片类型
//源文件 url
mat1.source = "http://www.mtstar.online:6080/adc/i/20190812/1565578896420682493584@482@77@000001.jpg";
//缩略图 url，不能省，没有就以source值填充
mat1.src = "http://www.mtstar.online:6080/adc/i/20190812/_1565578896420682493584@482@77@000001.png";

mat1.name = "图片1";
mat1._long = 5 ;
mat1.mh = 0; //不能少！素材存储区编号、存于满天星的素材，则使用素材的mh，否则一律为0
mat1.id = ""; // 源文件id,互联网素材（mh==0）此值为空，满天星（mh>0）素材此值为素材id
mat1.fileSize = 16467; // 源文件大小，long型
element1.curimglist.push(mat1);

//素材2，假定为视频
var mat2 = {};
mat2.type = "video";// video 表示视频类型

//源文件 url
mat2.source = "http://www.mtstar.online:6080/adc/v/20200325/48713207575@78@77@000001.mp4";

//缩略图 url，不能省
mat2.src = "http://www.mtstar.online:6080/adc/v/20200325/48713207575@78@77@000001.jpg";

mat2.name = "视频1";
mat2._long = 35 ; //视频的时长
```

```
mat2.mh = 0; //不能少! 素材存储区编号、存于满天星的素材, 则使用素材的mh, 否则一律为0
mat2.id = ""; // 源文件id, 互联网素材 (mh==0) 此值为空, 满天星 (mh>0) 素材此值为素材id
mat2.fileSize = 9860593; // 源文件大小, long型
element1.curimglist.push(mat2);

scene.datalist.push(element1);

//元素2 屏幕右部分
var element2 = {};
element2.x = 960;
element2.y = 0;
element2.width = 960;
element2.height = 1080;
element2.delay = 5;
element2.curimglist = new Array();
//素材3 图片
var mat3 = {};
mat3.type = "img"; // img 表示图片类型
//源文件 url
mat3.source = "http://www.mtstar.online:6080/adc/i/20190812/1565578896420682493584@482@77@000001.jpg";

//缩略图 url, 不能省, 没有就以source值填充
mat3.src = "http://www.mtstar.online:6080/adc/i/20190812/_1565578896420682493584@482@77@000001.png";

mat3.name = "图片2";
mat3._long = 5 ;
mat3.mh = 0; //不能少! 素材存储区编号、存于满天星的素材, 则使用素材的mh, 否则一律为0
mat3.id = ""; // 源文件id, 互联网素材 (mh==0) 此值为空, 满天星 (mh>0) 素材此值为素材id
mat3.fileSize = 16467; // 源文件大小, long型
element2.curimglist.push(mat3);

//素材4图片
var mat4 = {};
mat4.type = "img"; // video 表示图片类型
//源文件 url
mat4.source = "http://www.mtstar.online:6080/adc/i/20191211/1576025442590486732584@482@77@000001.jpg";

//缩略图 url, 不能省, 没有就以source值填充
mat4.src = "http://www.mtstar.online:6080/adc/i/20191211/_1576025442590486732584@482@77@000001.png";

mat4.name = "图片3";
mat4._long = 5 ;
mat4.mh = 0; //不能少! 素材存储区编号、存于满天星的素材, 则使用素材的mh, 否则一律为0
```

```
mat4.id = ""; // 源文件id,互联网素材 (mh==0) 此值为空, 满天星 (mh>0) 素材此值为素材id
mat4.fileSize = 79723; // 源文件大小, long型
element2.curimglist.push(mat4);
scene.datalist.push(element2);

//元素3背景音乐, 背景音乐的xy宽高固定 一个广告内最多一个music元素, 用户自己拿捏
var element3 = {};
element3.x = 0;
element3.y = 0;
element3.width = 50;
element3.height = 50;
element3.type = "music";// music 表示音频类型 有此值表无窗体背景音乐元素, 无此值默认为img-video
element3.curimglist = new Array();
//素材3 图片
var mat5 = {};
mat5.type = "music";// music 表示音频类型
//源文件 url
mat5.source = "http://www.mtstar.online:6080/adc/a/20190523/1558609770991795579829@78@77@000001.mp3";
mat5.name = "音乐";
mat5._long = 155 ; //时长, 秒
mat5.mh = 0; //不能少! 素材存储区编号、存于满天星的素材, 则使用素材的mh, 否则一律为0
mat5.id = ""; // 源文件id,互联网素材 (mh==0) 此值为空, 满天星 (mh>0) 素材此值为素材id
mat5.fileSize = 6184788; // 源文件大小, long型
element3.curimglist.push(mat5);

scene.datalist.push(element3);

//元素4文本,
var element4 = {};
element4.x = 0;
element4.y = 0;
element4.width = 1920;
element4.height = 120; //这个高度是显示区域高度, 请注意后面的fontsize比较大的话, 此值应相应调整
element4.type = "text";// text 表示文本类型
element4.content = "测试文本元素"; //文本内容
element4.fontsize = 100; //字体大小, 注意, 这个字体大小是设备端的字体大小
element4.bgcolor = 'ffffff'; //背景颜色
element4.tarnsparent = 1; //文本是否透明 1: 透明; 0 不透明
element4.txtcolor = '#000000'; //文本颜色
element4.txtmovetype = 0; //滚动模式 0: 静态; 1: 向左; 2: 向右; 3: 向上
element4.txtmovespeed = 1; //移动速度 1:超快; 2: 快; 3: 正常; 4、慢; 5: 超慢
element4.txtreadtext = 0; //固定
element4.fontJustify = 2; // 1:居左; 2: 居中; 3: 居右
scene.datalist.push(element4);
```

```
var str_scene = JSON.stringify(scene); //转为字符串

//用户token
var access_token = "9657d408-cb99-11e8-89642-00163e046215";

//用户secret, 这里仅仅为了演示调用方法直接暴露了, 很危险, 开发人员请慎重, 勿明文显示暴露
var access_secret = "a733613";

// hex_md5是取字符串md5值
var acces_secret_md5 = hex_md5(access_secret);

//unix时间戳 :秒 since 1970, 请校准信发服务器时间, 差值过大接口会返回失败
var timestamp = getTimeStamp();

var sign = hex_md5(access_token+acces_secret_md5+timestamp);

$.ajax({
  type: "POST",
  url: 'sdk/advert_edit_item.do',
  data: {access_token:access_token,timestamp:timestamp,sign:sign,
        itemId:itemId,scene:str_scene},
  dataType: 'json',
  cache: false,
  success: function(data){
    if("success" == data.result){
      console.log("success! advert id:"+data.id);
    }else {
      console.log("failed! retInfo:"+data.retInfo);
    }
  }
}); //end ajax
}
```

返回说明

正确调用后的返回示例:

```
{ "result": "success",
  "retInfo": "success",
  "retCode": 0,
  "id": "1789861689@88@77@0000001"
}
```

参数说明:

名称	类型	描述
retCode	int	0 表示成功, 其他错误码
retInfo	String	错误描述
id	string	该广告 id 值, retCode==0 时有值、有效, 新增的模式下此值为新增的广告 id, 编辑模式为原广告 id

3.11 审核广告

http://server:port/adc/sdk/advert_audit_items

应用参数说明:

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳, 秒 since 1970
sign	String	Y	接口调用签名
itemIds	string	Y	需要删除的广告 id 字符串, 批量审核时多个 id 间以英文逗号分隔
auditStatus	int	Y	审核结果: 0 未审核 1 审核通过 2 审核拒绝

返回说明

正确调用后的返回示例:

```
{ "result": "success",  
  "retInfo": "success",  
  "retCode": 0,  
  "row": 2  
}
```

参数说明:

名称	类型	描述
----	----	----

retCode	int	0 表示成功，其他错误码
retInfo	String	错误描述
row	int	成功的行数

3.12 删除广告

`http://server:port/adk/sdk/advert_del_items`

应用参数说明：

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳，秒 since 1970
sign	String	Y	接口调用签名
itemIds	string	Y	需要删除的广告 id 字符串，批量删除时多个 id 间以英文逗号分隔

返回说明

正确调用后的返回示例：

```
{  "result": "success",
  "retInfo": "success",
  "retCode": 0,
}
```

参数说明：

名称	类型	描述
retCode	int	0 表示成功，其他错误码
retInfo	String	错误描述

3.13 发布广告计划

接口：

`http://server:port/adk/sdk/advert_dist_items`

应用参数说明：

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳，秒 since 1970
sign	String	Y	接口调用签名
itemName	String	Y	本次发布的名称（请赋有意义的值方便后续查询）
itemIds	string	Y	需要发布的广告 id 字符串，多个 id 间以英文逗号分隔
cMarks	string	Y	发布的目标设备标识，多个 cMark 以英文逗号隔开
groupId	string	N	设备分组 id，当 cMarks 为空字符时有效（设备分组相关请参考后续 3.23—3.26 节）
sDate	String	Y	播放起始日 yyyy-MM-dd 格式
eDate	String	Y	播放截止日 yyyy-MM-dd 格式
beginTime	String	Y	开始时间 hh:mm:ss ;24hour 时分秒
endTime	String	Y	结束时间 hh:mm:ss ;24hour 时分秒
playtimetype	Int	Y	播放模式，仅提供两种值（1、10），其他值勿尝试。1：普通按序循环播放模式；10：循环间隔限次播放模式
playCount	int	Y	播放限定次数 playtimetype==10 有效
interval_second			播放间隔，秒 playtimetype==10 有效
pType	Int	Y	发布类型，0：追加、1：替换、2：清除、3：下载完立即清除（另：1 替换模式可以某种程度上当做撤销来使用）

调用 js 示例代码：

```
function do_test_sdk_send_advert_items(){
    //用户token
    var access_token = "6547d408-cb99-11e8-bbc2-00163e046235";
    //用户secret，这里仅仅为了演示调用方法直接暴露了，很危险，开发人员请慎重，勿明文显示暴露
    var access_secret = "a733613";
    var acces_secret_md5 = hex_md5(access_secret);

    //unix时间戳：秒 since 1970，请校准信发服务器时间
    var timestamp = getTimeStamp();
    var sign = hex_md5(access_token+acces_secret_md5+timestamp);
```

//自定义名字, 请用有意义的名称, 方便查询发布详情, 单个广告的话建议直接借用广告名

```
var itemName = "发布计划-test";
```

//目标播放设备标识码; 多个以英文, 分割

```
var cMarks = "44:EF:BF:0E:86:8B,64:EF:BF:0E:86:8B ";
```

//待发布的广告, 多个以英文, 分割

```
var itemIds = "2053836713@88@77@000001,2053836713@88@77@000001";
```

//以下四个参数 表示起~止日期内 的开始~结束时间段内播放

```
var sDate = "2020-04-19"; //开始日期
```

```
var eDate = "2020-05-19" //结束日期
```

```
var beginTime = "00:00:00"; //开始时间
```

```
var endTime = "23:59:59"; //结束时间
```

// 每天播放限定次数playtimetype ==10时候有效

```
var playCount = 180;
```

// 最小播放间隔, 单位秒, playtimetype ==10时候有效

```
var interval_second = 60*5;
```

/* playtimetype ==10:广告每隔设定时间间隔按天最多播放N次

* playtimetype==1 普通播放, 在有效时间内有空闲时间就播放,

* 此值为1时, playCount和interval_second无效

*/

```
var playtimetype = 10;
```

/*

pType ==0 追加模式, pType == 1替换模式(下载完等待在播节目结束再替换);

pType==2 清除模式, 特殊应用模式 pType=3 下载完立即替换

*/

```
var pType = 0;
```

```
$.ajax({
```

```
  type: "POST",
```

```
  url: 'sdk/advert_dist_items.do',
```

```
  data: {access_token:access_token,timestamp:timestamp,sign:sign,
```

```
        cMarks:cMarks,itemIds:itemIds,itemName:itemName,
```

```
        sDate:sDate,eDate:eDate,beginTime:beginTime,endTime:endTime,
```

```
        playtimetype:playtimetype,playCount:playCount,
```

```
        interval_second:interval_second,pType:pType
```

```
  },
```

```
  dataType: 'json',
```

```
  cache: false,
```

```
  success: function(data){
```



```
        if("0" == data.retCode){
            console.log("success");
        }else {
            console.log("failed:"+data.retInfo);
        }
    }
}); //end ajax
}
```

返回说明

正确调用后的返回示例：

```
{ "result": "success",
  "retInfo": "success",
  "retCode": 0,
}
```

参数说明：

名称	类型	描述
retCode	int	0 表示成功，其他错误码
retInfo	String	错误描述

3.14 获取单设备当前广告列表

http://server:port/adc/sdk/dev_get_dev_advertlist

应用参数说明：

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳，秒 since 1970
sign	String	Y	接口调用签名
cMark	string	Y	设备 cMark
currentPage	int	Y	第几页数据（每页按 showCount 计）

返回说明

正确调用后的返回示例：

```
{
```

```
"retCode": 0
"result": "success",
"count": 2,
"retInfo": "OK",
"list": [
  {
    "playtimetype": 10,
    "interval_second": 300,
    "playCount": 180,
    "userName": "满天星",
    "sDate": "2020-04-19",
    "eDate": "2020-05-19",
    "beginTime": "00:00:00",
    "endTime": "23:59:59",
    "createTime": "2020-04-19 11:22:33",
    "name": "发布计划-test",
    "pType": "add",
    "progress": "0%",
    "status": 0
  },
  {
    "playtimetype": 1,
    "interval_second": 300,
    "playCount": 180,
    "userName": "满天星",
    "itemList": "2053836713@88@77@000001,2053836713@88@77@000001",
    "sDate": "2020-04-19",
    "beginTime": "00:00:00",
    "endTime": "23:59:59",
    "eDate": "2020-05-19",
    "createTime": "2020-04-19 11:22:33",
    "name": "发布计划-test",
    "pType": "add",
    "progress": "100%",
    "status": 4
  }
]
}
```

参数说明:

名称	类型	描述
retCode	int	0 表示成功, 其他错误码

retInfo	String	错误描述
count	int	返回的广告数量
list	Array	返回的结果集
list.itemList	string	广告 id 的 list，多个以英文逗号分割
list.playtimetype	string	播放模式，1：普通、10：间隔限次播放
list.status	int	接收状态：0：未发布、1：已发布未接收、2：已接收未下载、3：正下载、4：已下载完成
list.progress	string	下载进度，status==3 时有效
list.pType	string	发布模式（和上不同）add、replace、clear
list.name	String	发布时的名称

3.15 撤销广告

本接口需前端设备 APK 版本不低于 498

http://server:port/adc/sdk/advert_revoke_item

应用参数说明：

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳，秒 since 1970
sign	String	Y	接口调用签名
itemId	string	Y	广告 id
itemName	string	Y	撤销的时间标注名，后台日志检索用
cMarks	string	Y	设备 cMark 拼接而成的字符串，多个以英文逗号分割
groupId	string	N	设备组 id，当 cMarks 为空字符串时有效

返回说明

正确调用后的返回示例：

```
{ "result": "success",
  "retCode": 0
}
```

参数说明：

名称	类型	描述
retCode	int	0 表示成功，其他错误码
retInfo	String	错误描述

3.16 获取素材数量

接口：

`http://server:port/adc/sdk/material_get_itemcount`

应用参数说明：

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳，秒 since 1970
sign	String	Y	接口调用签名
itemType	int	Y	素材类型 0: img;1:audio;4:video
select_name	string	N	可选；检索条件，可为素材名（模糊搜索），也可以是确定的素材 id（精确匹配），须与 select_id 配合使用
select_id	int	N	select_name 有值时有效；0：搜素材名、1：搜确定 id 的素材

返回说明

正确调用后的返回示例：

```
{ "result": "success",  
  "retInfo": "success",  
  "retCode": 0,  
  "itemCount": 4097,  
  "itemType": 4  
}
```

参数说明：

名称	类型	描述
retCode	int	0 表示成功，其他错误码
retInfo	String	错误描述
itemCount	int	当前类型的所有素材数量（按创建日期降序排列）

itemType	int	用户传入的类型
----------	-----	---------

3.17 获取素材列表

http://server:port/adc/sdk/material_get_itemlist

应用参数说明：

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳，秒 since 1970
sign	String	Y	接口调用签名
itemType	int	Y	素材类型 0: img;1:audio;4:video
showCount	int	Y	接口获取数据按分页模式处理，每次获取数据按广告的创建时间降序排列，后台按每页 showCount 数量组织数据返回结果集，每页最多获取条数（500Max）
currentPage	int	Y	第几页数据（每页按 showCount 计，从 1 计数开始）
select_name	string	N	检索条件，可为素材名（模糊搜索），也可以是确定的素材 id（精确匹配），须与 select_id 配合使用
select_id	int	N	select_name 有值时有效；0：搜素材名、1：搜确定 id 的素材

返回说明

正确调用后的返回示例：

```
{
  "result": "success",
  "showCount": 2,
  "itemList": [
    {
      "rowNo": 1,
      "id": "4216566543@000001",
      "itemName": "QQ 短视频 20190620154650",
      "filePath": "http://192.168.3.215:6080/adc/uf/v/20200429/4216563790@88@77@000001.mp4",
      "mh": 322,
    }
  ]
}
```

```

        "tsc": 4,
        "fileSize": "2703161",
        "sTime": 6,
        "duration": 5043,
        "thumbnails": "http://192.168.3.215:6080/adc/uf/v/20200429/4216563790@88@77@000001.jpg",
        "upTime": "2020-04-29 16:56:08",
        "itemType": 4,
        "itemReady": 1,
        "resolution": "720x1280"
    },
    {
        "rowNo": 2,
        "id": "5509884880@000001",
        "itemName": "490A06308",
        "filePath": "http://192.168.3.215:6080/adc/uf/v/20200327/5509871543@88@77@000001.mp4",
        "mh": 322,
        "tsc": 4,
        "fileSize": "6659876",
        "sTime": 21,
        "duration": 20011,
        "thumbnails": "http://192.168.3.215:6080/adc/uf/v/20200327/5509871543@88@77@000001.jpg",
        "upTime": "2020-03-27 20:31:38",
        "itemType": 4,
        "itemReady": 1,
        "resolution": "1280x720"
    }
],
"retInfo": "success",
"retCode": 0,
"currentPage": 1,
"itemCount": 2
}

```

参数说明:

名称	类型	描述
retCode	int	0 表示成功，其他错误码
retInfo	String	错误描述
showCount	int	用户提交的 showCount 参数，原路返回以作确认
currentPage	int	用户提交的 currentPage 参数
itemCount	int	返回的广告数量（按制作日期降序排列）
itemList	Array	返回的结果集
itemList.id	string	素材 id， 制作广告会用到此 id

<code>itemList.itemName</code>	string	素材名
<code>itemList.filePath</code>	string	素材 url
<code>itemList.mh</code>	int	素材对象存储区编号，大于 0 的值是满天星服务器对象存储区编号，等于 0 则代表未知互联网对象存储区编号，制作广告的素材参数需要此值，理论上制作广告时都填 0 是可以的（满天星存储区本身也属于存储区的一类），但建议区别对待为佳（特别是用户没有做 CDN 相关配置情况下）
<code>itemList.fileSize</code>	long	素材大小
<code>itemList.sTime</code>	int	时长，单位秒，向上取整（视频、音频此参数有效）
<code>itemList.duration</code>	int	时长、毫秒（视频、音频此参数有效）
<code>itemList.thumbnails</code>	string	缩略图（视频、图片等有效）
<code>itemList.upTime</code>	String	上传时间
<code>itemList.itemType</code>	int	素材类型 0: img;1:audio;4:video
<code>itemList.itemReady</code>	int	素材是否可用 1: ready, 可用; 非 1: 不可用; 在服务器采用大文件加速配置情况下刚刚上传成功情况下可能出现非 1（加速有延迟若干秒不等），用户可以等待一段时间 <code>itemReady==1</code> 时再使用
<code>itemlist.resolution</code>	string	分辨率（图片、视频等此值有效）

3.18 删除素材

http://server:port/adc/sdk/material_del_items

应用参数说明：

名称	类型	必选	描述
<code>access_token</code>	String	Y	用户调用 token 获取方式见第 1 章
<code>timestamp</code>	Long	Y	Unix 时间戳，秒 since 1970
<code>sign</code>	String	Y	接口调用签名
<code>itemIds</code>	string	Y	需要删除的素材 id 字符串，批量删除时多个 id 间以英文逗号分隔

返回说明

正确调用后的返回示例：

```
{ "result": "success",  
  "retInfo": "success",  
  "retCode": 0,  
}
```

参数说明:

名称	类型	描述
retCode	int	0 表示成功, 其他错误码
retInfo	String	错误描述

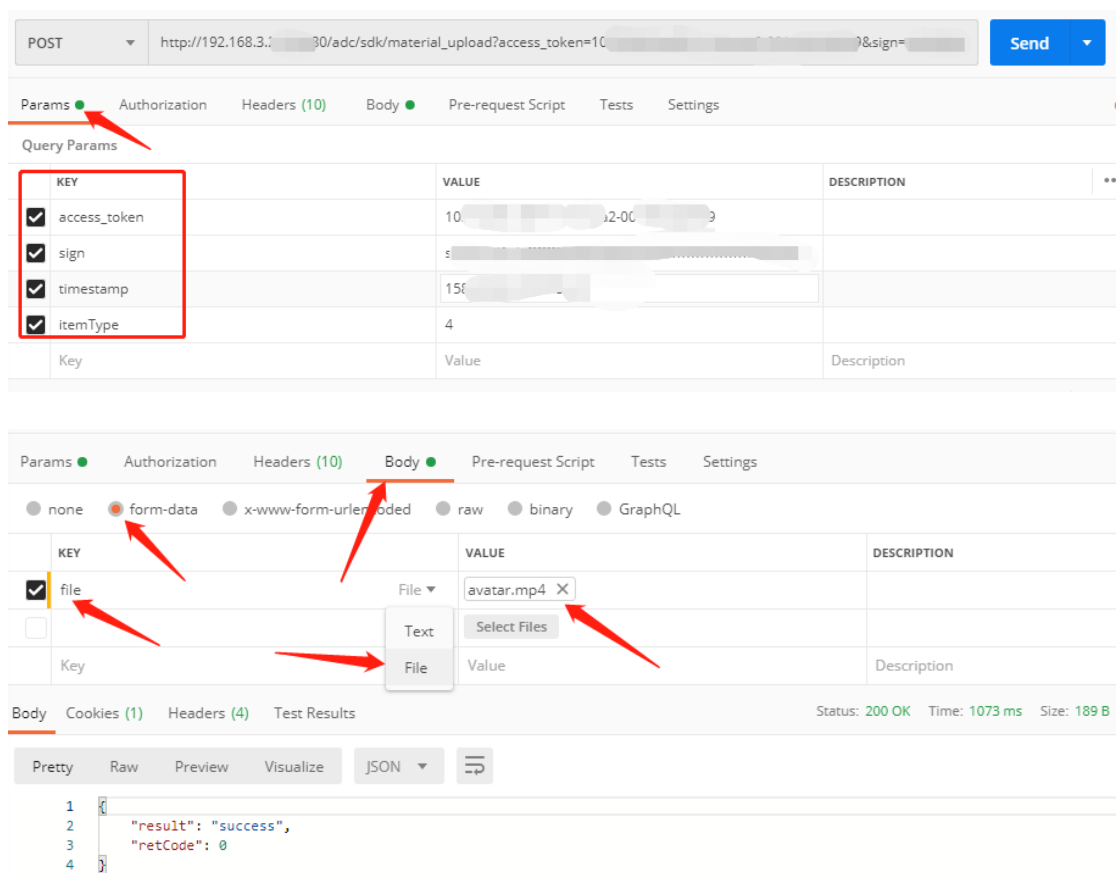
3.19 上传素材

http://server:port/adc/sdk/material_upload

应用参数说明:

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳, 秒 since 1970
sign	String	Y	接口调用签名
itemType	string	Y	素材类型 0: img;1:audio;4:video 此参数很重要, 不能用错也不能漏, 需一一对应! 因为素材的多样性, 不同的类型需要不同的处理方式, 后台单纯依靠文件后缀名既不可靠、也不足以判断其真实类型, 而且未来新的格式类型的文件也可能出现, 为兼容及扩展性计, 故将类型赋值交给客户端程序自行对应
file	MultipartFile	Y	媒体文件类型, 注意, 本接口只支持单个文件, 用户多个文件的情况, 请在客户端循环调用

调用参考:



返回说明

正确调用后的返回示例:

```
{
  "result": "success",
  "retCode": 0,
}
```

3.20 重启设备

`http://server:port/adc/sdk/dev_reboot_dev`

应用参数说明:

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳, 秒 since 1970
sign	String	Y	接口调用签名
cMarks	string	Y	英文逗号分割的 cMark, 一次调用限制 100Max

重启设备属于敏感操作，因此服务器不重发指令，在线的设备才能控制成功，

返回说明

正确调用后的返回示例：

```
{ "result": "success",  
  "retCode": 0,  
  "count" : 1  
}
```

参数说明：

名称	类型	描述
retCode	int	0 表示成功，其他错误码
retInfo	String	错误描述
count	int	成功发送重启指令的个数（其实就是发送指令中在线的数量，在线的才能控制）

3.21 设备改名

http://server:port/adc/sdk/dev_change_dev_name

应用参数说明：

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳，秒 since 1970
sign	String	Y	接口调用签名
cMark	string	Y	设备 cMark
cName	string	Y	设备新名称

返回说明

正确调用后的返回示例：

```
{ "result": "success",  
  "retCode": 0,  
  "count" : 1  
}
```

参数说明：

名称	类型	描述
retCode	int	0 表示成功，其他错误码
retInfo	String	错误描述

3.22 设备设置 LOGO

`http://server:port/adc/sdk/dev_set_devparam`

本接口需前端设备 APK 版本不低于 470

应用参数说明：

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳，秒 since 1970
sign	String	Y	接口调用签名
cMarks	string	Y	设备 cMark 以英语逗号拼接而成的设备 cMark 字符串
paramType	int	Y	固定值 5
params	string	Y	JSON 格式的字符串，结构详见调用示例

调用示例

```
var cMarks = "aa:bb:cc:dd:ee:ff,a1:b1:c1:d1:e1:f1" ;
var paramType = 5;//设备 logo
var params = {};
params.display = 1; //1: 启用 logo; 0: 取消 logo
params.positon = 3; // 1:左上角、2: 右上角、3: 左下角、4: 右下角
params.logourl = ""; // display==1 时候不能为空，图片的 url 地址
```

//关于 logo 宽高，由于前端设备分辨率可能不同，因此采用比例来界定

```
params.width = "1.0"; // logo 的宽度（范围 0~1）
params.height = "0.2"; // logo 的高度（范围 0~1）
var str_params = JSON.stringify(params);
```

```
$.ajax({
    type: "POST",
    url: 'sdk/dev_set_devparam',
    data: {access_token:access_token,timestamp:timestamp,sign:sign,
          cMarks:cMarks,
          paramType:paramType,params:str_params},
    dataType:'json',
    cache: false,
    success: function(data){
        if("success" == data.result){

        }else {

        }
    }
}); //end ajax
```

返回说明

正确调用后的返回示例:

```
{ "result": "success",
  "retCode": 0
}
```

参数说明:

名称	类型	描述
retCode	int	0 表示成功, 其他错误码
retInfo	String	错误描述

3.23 新建/编辑设备分组

http://server:port/adc/sdk/edit_group

应用参数说明:

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳, 秒 since 1970
sign	String	Y	接口调用签名
gType	Int	Y	固定 1
gName	string	Y	分组名称; 平台基于效率等因素计, 会

			对分组名称做唯一值校验
groupId	String	Y	为空字符串，则为新增分组，不为空的有意义的组 id，则为编辑分组（更名）

返回说明

正确调用后的返回示例：

```
{
  "result": "success",
  "retInfo": "OK",
  "retCode": 0
}
```

组名称已存在调用后的返回示例：

```
{
  "result": "failed",
  "retInfo": "组名已存在",
  "retCode": 40007
}
```

参数说明：

名称	类型	描述
retCode	int	0 表示成功，其他错误码
retInfo	String	错误描述

3.24 查询设备分组数量

http://server:port/adcsdk/get_group_count

应用参数说明：

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳，秒 since 1970
sign	String	Y	接口调用签名
gType	Int	Y	固定 1

返回说明

正确调用后的返回示例：

```
{
  "result": "success",
  "count": 105,
  "retInfo": "OK",
  "retCode": 0
}
```

参数说明：

名称	类型	描述
retCode	int	0 表示成功，其他错误码
retInfo	String	错误描述
count	int	分组数量

3.25 查询设备分组列表信息

http://server:port/adc/sdk/get_group_list

应用参数说明：

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳，秒 since 1970
sign	String	Y	接口调用签名
gType	Int	Y	固定 1
select_name	string	N	可选，可输入分组名搜索（模糊搜索）
showCount	int	Y	接口获取数据按分页模式处理，每次获取数据按分组的创建时间升序排列，后台按每页 showCount 数量组织数据返回结果集，每页最多获取条数（500Max）
currentPage	int	Y	第几页数据（每页按 showCount 计，从 1 计数开始）

返回说明

正确调用后的返回示例:

```
{
  "result": "success",
  "count": 105,
  "retInfo": "OK",
  "list": [
    {
      "gName": "住宅測試機",
      "gType": 1,
      "count": 6,
      "id": "1571986336140811890523@000001"
    }
  ]
}
```

参数说明:

名称	类型	描述
retCode	int	0 表示成功, 其他错误码
retInfo	String	错误描述
count	int	分组数量
list	array	分组结果集
list.gName	string	分组名称
list.id	string	分组 id
list.gType	int	分组类型
list.count	int	分组所属设备数量

3.26 查询单个设备分组信息

http://server:port/adcsdk/get_group_item_info

应用参数说明:

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳, 秒 since 1970
sign	String	Y	接口调用签名
gType	Int	Y	固定 1
groupId	string	Y	分组 id

返回说明

正确调用后的返回示例：

```
{
  "result": "success",
  "retInfo": "OK",
  "retCode": 0,
  "info": {
    "gType": 1,
    "gName": "住宅",
    "count": 6,
    "id": "1571986336140811890523@000001"
  }
}
```

参数说明：

名称	类型	描述
retCode	int	0 表示成功，其他错误码
retInfo	String	错误描述
count	int	分组数量
info	object	数据
info.gName	string	分组名称
info.id	string	分组 id
info.gType	int	分组类型
info.count	int	分组所属设备数量

3.27 删除设备分组信息

http://server:port/adcsdk/group_item_del

应用参数说明：

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳，秒 since 1970
sign	String	Y	接口调用签名

gType	Int	Y	固定 1
groupId	string	Y	分组 id

返回说明

正确调用后的返回示例：

```
{
  "result": "success",
  "retInfo": "OK",
  "retCode": 0
}
```

3.28 查询设备分组的设备数量

http://server:port/adc/sdk/get_dev_group_devs_count

应用参数说明：

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳，秒 since 1970
sign	String	Y	接口调用签名
gType	Int	Y	固定 1
groupId	string	Y	分组 id

返回说明

正确调用后的返回示例：

```
{
  "result": "success",
  "count": 2,
  "retInfo": "OK",
  "retCode": 0
}
```

参数说明：

名称	类型	描述
----	----	----

retCode	int	0 表示成功，其他错误码
retInfo	String	错误描述
count	int	返回的 cMark 数量

3.29 查询设备分组的设备 cMarks

http://server:port/adc/sdk/get_dev_group_devs

应用参数说明：

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳，秒 since 1970
sign	String	Y	接口调用签名
gType	Int	Y	固定 1
groupId	string	Y	分组 id
showCount	int	N	接口获取数据按分页模式处理，每次获取数据按设备的接入时间升序排列，后台按每页 showCount 数量组织数据返回结果集，每页最多获取条数（500Max）
currentPage	int	N	第几页数据（每页按 showCount 计，从 1 计数开始）
pageget	int	N	1:按页获取数据，0：一次性全部获取，默认是分页获取

返回说明

正确调用后的返回示例：

```
{
  "result": "success",
  "count": 2,
  "cMarks": "94:E0:D6:EF:EB:21,44:EF:BF:0E:86:8B",
  "retInfo": "OK",
  "retCode": 0
}
```

参数说明：

名称	类型	描述
retCode	int	0 表示成功，其他错误码

retInfo	String	错误描述
count	int	返回的 cMark 数量
cMarks	string	返回的设备 cMark 拼接而成的字符串

3.30 添加设备到分组/从分组移除设备

http://server:port/adcsdk/set_devs_group

应用参数说明:

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳, 秒 since 1970
sign	String	Y	接口调用签名
gType	Int	Y	固定 1
groupId	string	Y	分组 id
action	int	Y	1: 添加到分组、0: 从分组移除
cMarks	string	Y	设备 cMark 拼接而成的字符串, 多个以英文逗号分隔

返回说明

正确调用后的返回示例:

```
{
  "result": "success",
  "total": 2,
  "retInfo": "OK",
  "retCode": 0
}
```

参数说明:

名称	类型	描述
retCode	int	0 表示成功, 其他错误码
retInfo	String	错误描述

total	int	当前分组的设备数量
-------	-----	-----------

3.31 查询跑马灯数量

http://server:port/adc/sdk/get_marque_count

应用参数说明：

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳，秒 since 1970
sign	String	Y	接口调用签名
select_name	string	N	可选，可输入标题搜索（模糊搜索）

返回说明

正确调用后的返回示例：

```
{
  "result": "success",
  "count": 3,
  "retInfo": "OK",
  "retCode": 0
}
```

参数说明：

名称	类型	描述
retCode	int	0 表示成功，其他错误码
retInfo	String	错误描述
count	int	分组数量

3.32 查询跑马灯列表信息

http://server:port/adc/sdk/get_marque_list

应用参数说明:

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳, 秒 since 1970
sign	String	Y	接口调用签名
gType	Int	Y	固定 1
select_name	string	N	可选, 可输入标题搜索 (模糊搜索)
showCount	int	Y	接口获取数据按分页模式处理, 每次获取数据按创建时间降序排列, 后台按每页 showCount 数量组织数据返回结果集, 每页最多获取条数 (500Max)
currentPage	int	Y	第几页数据 (每页按 showCount 计, 从 1 计数开始)

返回说明

正确调用后的返回示例:

```
{
  "result": "success",
  "count": 1,
  "retInfo": "OK",
  "list": [
    {
      "sContent": " testg1",
      "rowNo": 1,
      "fontType": 0,
      "bgColor": 3,
      "startTime": "2020-04-08",
      "id": "3179641678@000001",
      "endTime": "2030-04-06",
      "title": "test",
      "fontColor": 4
    }
  ]
}
```

参数说明:

名称	类型	描述
----	----	----

retCode	int	0 表示成功，其他错误码
retInfo	String	错误描述
count	int	分组数量
list	array	结果集
list.title	string	标题
list.id	string	id
list.sContent	String	跑马灯内容文本
list.fontColor	int	文本颜色 0~7 0: 透明、1: 红色、2: 蓝、3: 黄、4: 绿、 5: 白色、6: 灰色、7: 黑色
list.bgColor	int	背景颜色 0~7 0: 透明、1: 红色、2: 蓝、3: 黄、4: 绿、 5: 白色、6: 灰色、7: 黑色
list.startTime	string	开始日期，yyyy-MM-dd 格式
list.endTime	string	结束日期，yyyy-MM-dd 格式

3.33 查询单个跑马灯信息

http://server:port/adc/sdk/get_marque_info

应用参数说明：

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳，秒 since 1970
sign	String	Y	接口调用签名
marqueId	string	Y	跑马灯 id

返回说明

正确调用后的返回示例：

```
{
  "result": "success",
  "retInfo": "OK",
  "itemInfo": {
    "sContent": " testg1",
    "fontType": 0,
    "bgColor": 3,
    "startTime": "2020-04-08",
```

```
    "id": "3179641678@000001",
    "endTime": "2030-04-06",
    "title": "test",
    "fontColor": 4
  },
  "retCode": 0
}
```

参数说明:

名称	类型	描述
retCode	int	0 表示成功, 其他错误码
retInfo	String	错误描述
count	int	分组数量
itemInfo	object	分组数据
itemInfo.title	string	标题
itemInfo.id	string	id
itemInfo.sContent	String	跑马灯内容文本
itemInfo.fontColor	int	文本颜色 0~7 0: 透明、1: 红色、2: 蓝、3: 黄、4: 绿、 5: 白色、6: 灰色、7: 黑色
itemInfo.bgColor	int	背景颜色 0~7 0: 透明、1: 红色、2: 蓝、3: 黄、4: 绿、 5: 白色、6: 灰色、7: 黑色
itemInfo.startTime	string	开始日期, yyyy-MM-dd 格式
itemInfo.endTime	string	结束日期, yyyy-MM-dd 格式

3.34 新增/修改跑马灯

http://server:port/adc/sdk/edit_marque

应用参数说明:

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳, 秒 since 1970

sign	String	Y	接口调用签名
itemId	string	Y	跑马灯 id, 当为空字符串, 则为新增, 为有意义的跑马灯 id, 则编辑该跑马灯
title	string	Y	跑马灯标题, 用于标记查询用
sContent	string	Y	跑马灯内容文本
fontColor	int	Y	文本颜色 0~7 0: 透明、1: 红色、2: 蓝、3: 黄、4: 绿、 5: 白色、6: 灰色、7: 黑色
bgColor	int	Y	背景颜色 0~7 0: 透明、1: 红色、2: 蓝、3: 黄、4: 绿、 5: 白色、6: 灰色、7: 黑色
fontSpeed	int	Y	跑马灯滚动速度 1~5 1: 超慢、2: 慢、3: 正常、4: 快、5: 超快
startTime	string	Y	开始日期, yyyy-MM-dd 格式
endTime	string	Y	结束日期, yyyy-MM-dd 格式

返回说明

正确调用后的返回示例:

```
{  "result": "success",
  "retCode": 0,
  "id": "kdskfksfsdkfs"
}
```

参数说明:

名称	类型	描述
retCode	int	0 表示成功, 其他错误码
retInfo	String	错误描述
id	string	返回的跑马灯 id

3.35 删除跑马灯

http://server:port/adc/sdk/marque_del_items

应用参数说明:

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳, 秒 since 1970
sign	String	Y	接口调用签名
itemIds	string	Y	跑马灯 id 拼接而成的字符串, 英文逗号分割

返回说明

正确调用后的返回示例:

```
{ "result": "success",  
  "retCode": 0  
}
```

参数说明:

名称	类型	描述
retCode	int	0 表示成功, 其他错误码
retInfo	String	错误描述

3.36 发布跑马灯

http://server:port/adc/sdk/marque_dist_item

应用参数说明:

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳, 秒 since 1970
sign	String	Y	接口调用签名
marqueId	string	Y	跑马灯 id
cMarks	string	Y	设备 cMark 拼接而成的字符串, 多个以英文逗号分割
groupId	string	N	设备组 id, 当 cMarks 为空时有效

返回说明

正确调用后的返回示例:

```
{ "result": "success",  
  "retCode": 0  
}
```

参数说明:

名称	类型	描述
retCode	int	0 表示成功, 其他错误码
retInfo	String	错误描述

3.37 撤销跑马灯

http://server:port/adc/sdk/marque_revoke_item

应用参数说明:

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳, 秒 since 1970
sign	String	Y	接口调用签名
marqueId	string	Y	跑马灯 id
cMarks	string	Y	设备 cMark 拼接而成的字符串, 多个以英文逗号分割
groupId	string	N	设备组 id, 当 cMarks 为空字符串时有效

返回说明

正确调用后的返回示例:

```
{ "result": "success",  
  "retCode": 0  
}
```

参数说明:

名称	类型	描述
retCode	int	0 表示成功, 其他错误码
retInfo	String	错误描述

3.38 休眠、唤醒

`http://server:port/adc/sdk/sendClientParams`

此接口属于临时起意的调用，设备不在线或者其他原因未接收到，就忽略；前端设备重启或其他原因重置，也不保留前一次的休眠唤醒状态；如下设置休眠唤醒日程，请看后面 3.39

注：休眠唤醒与前端设备屏幕接口有关系，本接口只在设备在线时生效，如要设置休眠唤醒计划，请参照 3.39

应用参数说明：

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳，秒 since 1970
sign	String	Y	接口调用签名
paramType	string	Y	固定 17
cMarks	string	Y	设备 cMark 拼接而成的字符串，多个以英文逗号分割
params	string	Y	参数字符串，详见调用说明
retry	int	Y	固定 0，勿传其他值

调用参数

```
var sCmd = {  
    paramType:17,  
    onoff:1, //1:立即唤醒；0：立即休眠  
    subcmd:'ctrl_lcd_onoff_ontime'  
}  
  
var params = JSON.stringify(sCmd);
```

返回说明

正确调用后的返回示例：

```
{ "result": "success",  
  "retCode": 0  
}
```

参数说明：

名称	类型	描述
retCode	int	0 表示成功，其他错误码
retInfo	String	错误描述

3.39 休眠、唤醒计划

`http://server:port/adk/sdk/sendOnOffSet`

应用参数说明：

名称	类型	必选	描述
access_token	String	Y	用户调用 token 获取方式见第 1 章
timestamp	Long	Y	Unix 时间戳，秒 since 1970
sign	String	Y	接口调用签名
paramType	string	Y	固定 16
cMarks	string	Y	设备 cMark 拼接而成的字符串，多个以英文逗号分割
params	string	Y	参数字符串，详见调用说明

调用参数

// 休眠唤醒日程计划

```
var params = 16842753_0_12:00^16842753_1_14:00
```

其含义如下：

```
var params = 16842753_0_12:00^16842753_1_14:00
```

时间点：hh:mm格式

1：唤醒、0：休眠

以^分割多个休眠或唤醒的配置，休眠唤醒应成对出现，否则前端设备可能不会应用生效

int型，是有一个周期7天的16进制值转换而来的数值，比如此处的16842753对应的16进制数为0x1010001，其表示为周一、周五、周日12:00时分开始休眠

整个表示周一、周五、周日的12:00休眠，周一、周五、周日的14:00唤醒，其他时间（周二、周三、周四、周六）无动作

//取消休眠唤醒很简单，直接如下

```
var params = "";
```

返回说明

正确调用后的返回示例：

```
{ "result": "success",  
  "retCode": 0  
}
```

参数说明：

名称	类型	描述
retCode	int	0 表示成功，其他错误码
retInfo	String	错误描述

4、错误码&原因说明

错误码 0	成功（无错误）
错误码 40000	未知错误
错误码 40001	access_token 或 cMark 或 userKey 参数不正确
错误码 40002	设备不存在（用户设备列表不存在此 cMark 设备）
错误码 40003	cMark 错误（用户设备列表不存在此设备 cMark）
错误码 40004	设备不在线
错误码 40005	设备未录入（无错误）
错误码 40006	发送控制命令失败
错误码 40008	设备忙（其他用户正在操作设备）
错误码 40010	设备未设置物品库存数量
错误码 40011	控制指令发送失败
错误码 40012	控制指令发送异常
错误码 40013	库存不足
错误码 40014	免费份额已用完
错误码 40015	免费机会已达上限
错误码 40016	吸粉已达上限
错误码 40017	用户免费机会已用完
错误码 40018	设备设置的公众号参数不正常
错误码 40019	设备未设置物品数量（购买模式下）
错误码 40020	系统异常
错误码 40021	参数不全
错误码 40025	出货失败
错误码 40026	等待结果
错误码 40027	缺货
错误码 40028	超时

错误码 40029 未知

错误码 40030 权限不足

5、结束

感谢对满天星的关注！