**酷控匹配SDK接口文档v1.5**

Author:shuai@kookong.com

2017月10月20日

|  |  |  |
| --- | --- | --- |
| 时间 | 内容 | 作者 |
| 20171020 | 初次创建 | 高帅 |
| 20171024 | 增加红外部分, 升级版本1.4->1.5 | 高帅 |

[概述 2](#_Toc256000000)

[红外部分 2](#_Toc256000001)

[数据结构 2](#_Toc256000002)

[接口介绍 4](#_Toc256000004)

[EPG部分 8](#_Toc256000006)

[数据结构 8](#_Toc256000007)

[EPG接口介绍 9](#_Toc256000008)

# 概述

本文档为酷控SDK接口文档, 具体的代码使用可参考随SDK一起提供的Demo

# 红外部分

## 数据结构

**设备品牌-Brand**

|  |  |
| --- | --- |
| brandId | 品牌Id |
| initial | 品牌名称首字母 |
| cname | 品牌名称 |
| ename | 品牌英文名称 |
| pinyin | 品牌拼音 |

**机顶盒运营商-Sp**

|  |  |
| --- | --- |
| spId | 运营商Id |
| type | 运营商类型 0-有线运营商，1-IPTV运营商 |
| spName | 运营商名称 |

**IPTV品牌-Stb**

|  |  |
| --- | --- |
| bid | 品牌Id |
| bname | 品牌名称 |
| remotes | 品牌下的遥控器Id |

**单键测试按键-RcTestRemoteKeyV3**

|  |  |
| --- | --- |
| remoteIds | 测试结果 |
| frequency | 频率 |
| functionId | 按键Id |
| functionName | 按键英文名 |
| displayName | 按键名称 |
| pluseData | 波形 |

**红外码-IrData**

|  |  |
| --- | --- |
| rid | 红外码Id |
| fre | 频率 |
| type | 红外码类型 0-组合码(空调)，1-波形码 |
| keys | 此套红外码的遥控器按键集合 |
| exts | 遥控器扩展数据 |

**遥控器按键-IrKey**

|  |  |
| --- | --- |
| fid | 按键Id |
| fkey | 按键key |
| fname | 按键名称 |
| format | 格式码 |
| scode | 系统码 |
| dcode | 键码 |
| pulse | 波形 |
| exts | 键码扩展数据 |

**遥控器id集合 RemoteList**

|  |  |
| --- | --- |
| rids | List<Integer>, 遥控器id集合 |

**设备类型定义-Device**

|  |  |
| --- | --- |
| STB | 1（机顶盒） |
| TV | 2（电视） |
| BOX | 3（网络盒子） |
| DVD | 4（DVD） |
| AC | 5（空调） |
| PROJECTOR | 6（投影仪） |
| AV | 7（功放） |
| FAN | 8（风扇） |
| SLR\_CAMERA | 9（单反） |
| LIGHT | 10（灯泡） |

## 接口介绍

酷控SDK红外部分的功能接口都定义在KookongSDK类中

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 描述 | 参数 | 返回值 |
| init | 此接口会初始化SDK的so、基本设置等，初始化需要酷控提供的APP\_KEY（secret），初始化成功后才能进行其他操作。APP\_KEY获取方式: 厂商提供签名的MD5指纹和应用的包名, 酷控根据此信息生成 | Context context: App上下文, 建议传入Application的Context  secret：酷控授权的key | boolean  true 初始化成功  false 初始化失败 |
| getBrandListFromNet | 设备品牌列表 | int deviceType：设备类型, 类型定义参考上文Device类  IRequestResult<BrandList>  requestResult：接口回调, 参考**注释1-1** | 异步返回BrandList  ` |
| getAreaId | 传入通过定位或手动选择的省市区获取该地区的areaId | String province：省份  String city：城市  String district：地区  IRequestResult<Integer>  requestResult：接口回调 | 异步返回Integer |
| getOperaters | 传入地区id获取该地区下的机顶盒运营商列表 | areaid：地区id  IRequestResult<SpList> requestResult | 异步返回SpList |
| getIPTV | 传入iptv运营商的id获取该运营商下的品牌列表，如果运营商的 sp.type=1，就是IPTV运营商 | spid：iptv运营商Id  IRequestResult<StbList> requestResult | 异步返回StbList: |
| getAllRemoteIds | 此接口用于获取设备全部的RemoteId  如果是有线运营商，只需要传spId、areaId、deviceTypeId，brandId传0；  如果是其他设备，只需要传deviceTypeId、brandid。 | deviceTypeId：设备类型（Device.TV、Device.STB.....）  brandid：品牌Id  spId：运营商id  areaId：地区id  IRequestResult<RemoteList> requestResult | 异步返回RemoteList |
| getMatchKey | 此接口是单键测试的入口函数，通过该函数，获取待测的一组测试按键，按键包含了波形和频率用于发送。 | int deviceType：设备类型（Device.TV、Device.STB.....）  String allRemoteId：设备全部的remoteId  testSwitch ：是否测试电源键（如果电器是打开的，可跳过测试电源键）  ISingleMatchResult response 参考**注释1-2** |  |
| keyIsWorking | 如果一组中的某个按键响应了电器，调用此接口，sdk会根据按键数据返回处理结果，结果由response回调 | RcTestRemoteKeyV3  remoteKey：响应的按键数据  ISingleMatchResult response |  |
| groupKeyNotWork | 接口返回的一组待测按键如果都不响应，调用此接口反馈给sdk，sdk会把处理接口通过response返回 | List<RcTestRemoteKeyV3>  groupKeyList：不响应的一组按键  ISingleMatchResult response |  |
| getIRDataById | 匹配成功后，下载遥控器的红外码 | String remoteids：下载的红外码Id  IRequestResult<IrDataList> requestResult |  |

**注释1-1**

**IRequestResult接口返回结果回调：**

public interface IRequestResult<T>

{

//接口调用成功回调，数据封装在t中

public void onSuccess(String msg, T result);

//接口调用失败回调，status：错误代码，msg：错误信息

public void onFail(Integer errorCode,String msg);

}

**注释1-2**

**ISingleMatchResult单键对码结果回调：**

public interface ISingleMatchResult

{

//匹配流程结束，匹配到对应的遥控器，返回遥控器的Id

public void onMatchedIR(String remoteId);

//匹配流程继续，返回一组待测的按键

public void onNextGroupKey(List<RcTestRemoteKeyV3> groupKeyList);

//匹配流程结束，没有匹配到对应的遥控器

public void onNotMatchIR();

//匹配过程出错，流程直接结束

public void onError();

}

# EPG部分

## 数据结构

Channel 频道数据

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 介绍 |
| cid | int | 频道id |
| name | String | 频道名 |
| logo | String | 频道图标url地址 |
| num | int | 频道号(台号) |
| hd | int | 是否高清频道, 0:标清 1:高清 |
| category | int | 频道的分类，如卫视，央视，数字等, 其值参考注释1 |
| fee | int | 是否付费频道 0 非付费,1付费 |
| ctryId | String | 频道国家id, 注: 中国为空串 |
| type | int | 0 直播 ， 1轮播 |
| ext | String | 扩展，比如轮播的url |

ManualMatchLineupData 匹配频道表使用到的数据结构

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 介绍 |
| resultcode | int | 0: 成功  1: 继续  2失败 |
| result | String | 附加返回信息, (如 "参数错误"等) |
| channels | List<Channel> | 如果resultCode = 1匹配过程中待测试的频道  如果resultCode = 0 表示匹配到的频道表 |

**注释2-1:**

频道分类，定义在ChTypeConst中

public static final short CCTV\_TYPE = 1;//央视

public static final short SATELLITE\_TYPE = 2;//卫视

public static final short LOCAL\_TYPE = 3;//本地

public static final short DIGITAL\_TYPE = 4; //数字

public static final short HONGKONG\_TYPE = 5;//香港

public static final short AUMEN\_TYPE = 6; //澳门

public static final short TAIWAN\_TYPE = 7; //台湾

public static final short FOREIGN\_TYPE = 8; //外国

ManualMatchLineupData

## EPG接口介绍

酷控EPG SDK接口统一通过KookongSDK类提供

|  |  |  |  |
| --- | --- | --- | --- |
| 函数名 | 介绍 | 参数 | 返回 |
| manualMatchLineup | 匹配频道表接口, 具体匹配流程参考**注释2** | int areaid： 区域id, 获取方式参考红外SDK  int spid：运营商id ,获取方式参考红外SDK  String mr：匹配结果首次传null, 其余传用户选择的频道信息, 多次匹配的频道以逗号隔开, 格式:  频道num\_频道cid,道num\_频道cid...  IRequestResult<ManualMatchLineupData> 参考**注释1-1** | ManualMatchData  {  String result: 附加返回信息  int resultCode:  0: 成功  1: 继续  2失败  List<Channel> channels匹配过程中待测试的频道  } |

**注释2-2**

在同一个城市, 同一个区域下, 可能存在多分lineup频道表, 酷控云端会进行对频道表进行差异化对比, 将差异的频道提取出来让用户测试, 最终确定用户所使用的频道表.

例子:

**Step 1:**

参数

{

areaid = 37330

spid = 50

mr = null;

}

返回

ManualMatchData

{

resultCode = 1

channels = {

name=北京卫视, cid= 25,num= 25

name =天津卫视, cid = 27,num= 25

name = 湖南卫视, cid = 29,num= 25

}

}

此时App需要发送25对应的红外码,

UI需要展示北京卫视, 天津卫视,湖南卫视三个频道

**Step 2:**

假设在第一步用户选择了 "name =天津卫视(cid = 27), 则

参数

{

areaid = 37330

spid = 50

mr = 25\_27;

}

返回

ManualMatchData

{

resultCode = 1

channels = {

name=山东卫视, cid= 35, num= 32

name =西藏卫视, cid = 37, num= 32

}

}

此时App需要发送32频道, UI展示山东卫视, 西藏卫视让用户选择.

**Step 3**

如果用户选择了山东卫视(cid = 35):

请求参数

{

areaid = 37330

spid = 50

mr = 25\_27,32\_35;

}

返回数据

ManualMatchData

{

resulCode = 0

channels = {

name=北京卫视, cid= 25 num = 1,

name = 湖南卫视, cid = 29 num = 2

name =西藏卫视, cid = 37, num = 3

........

name =天津卫视, cid = 27, num = 25

.........

name=山东卫视, cid= 35, num = 32

........

}

}

resultCode = 0 表示获取到了频道表, channels即为用户的频道表

任何一步, 如果resultCode = 2,channels必定是null表示没有从酷控云端数据库找到对应的频道表信息, 匹配失败.