Siri快捷控制设计文档

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 修订内容 | 时间 | 修订人 |
| 1.0 | 创建文档 | 2019/5/29 | 邰光源 |
| 1.1 | Siri快捷控制整体流程设计 | 2019/5/31 | 邰光源 |
|  |  |  |  |

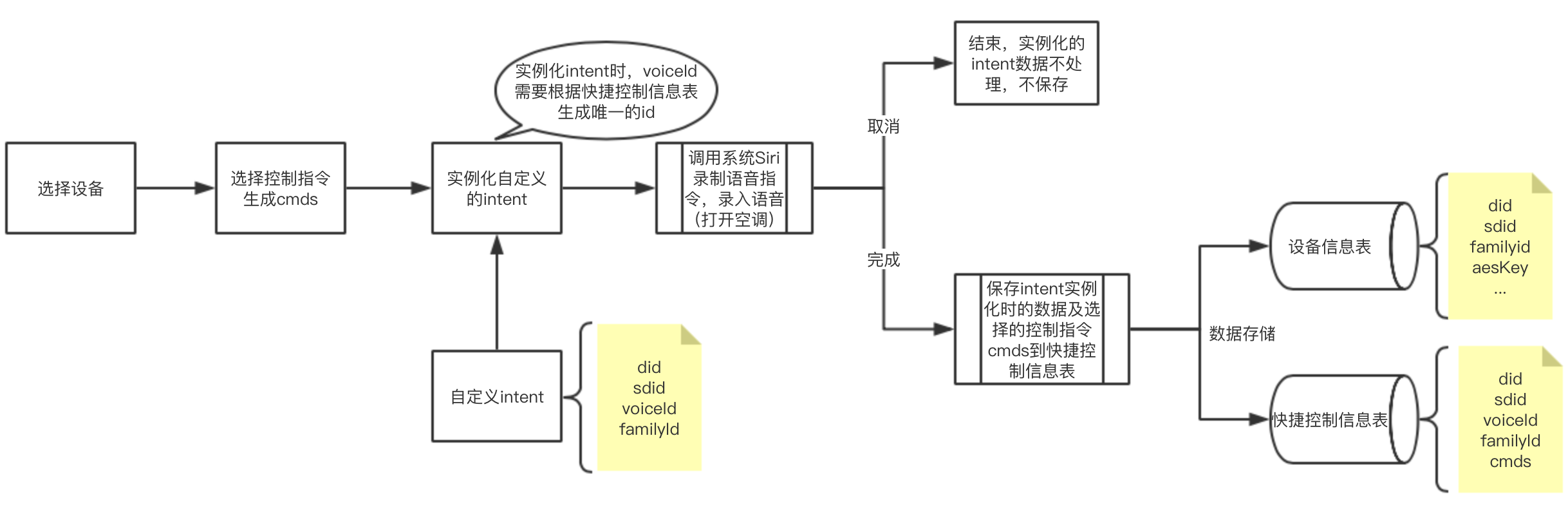
# 需求背景

Siri Shortcuts（快捷控制）是iOS 12中引入的新功能，允许开发人员使用Siri添加自定义的快捷方式，以便在未启动APP的情况下使用语音短语完成某些特定的快捷操作，提升用户体验及用户粘性。

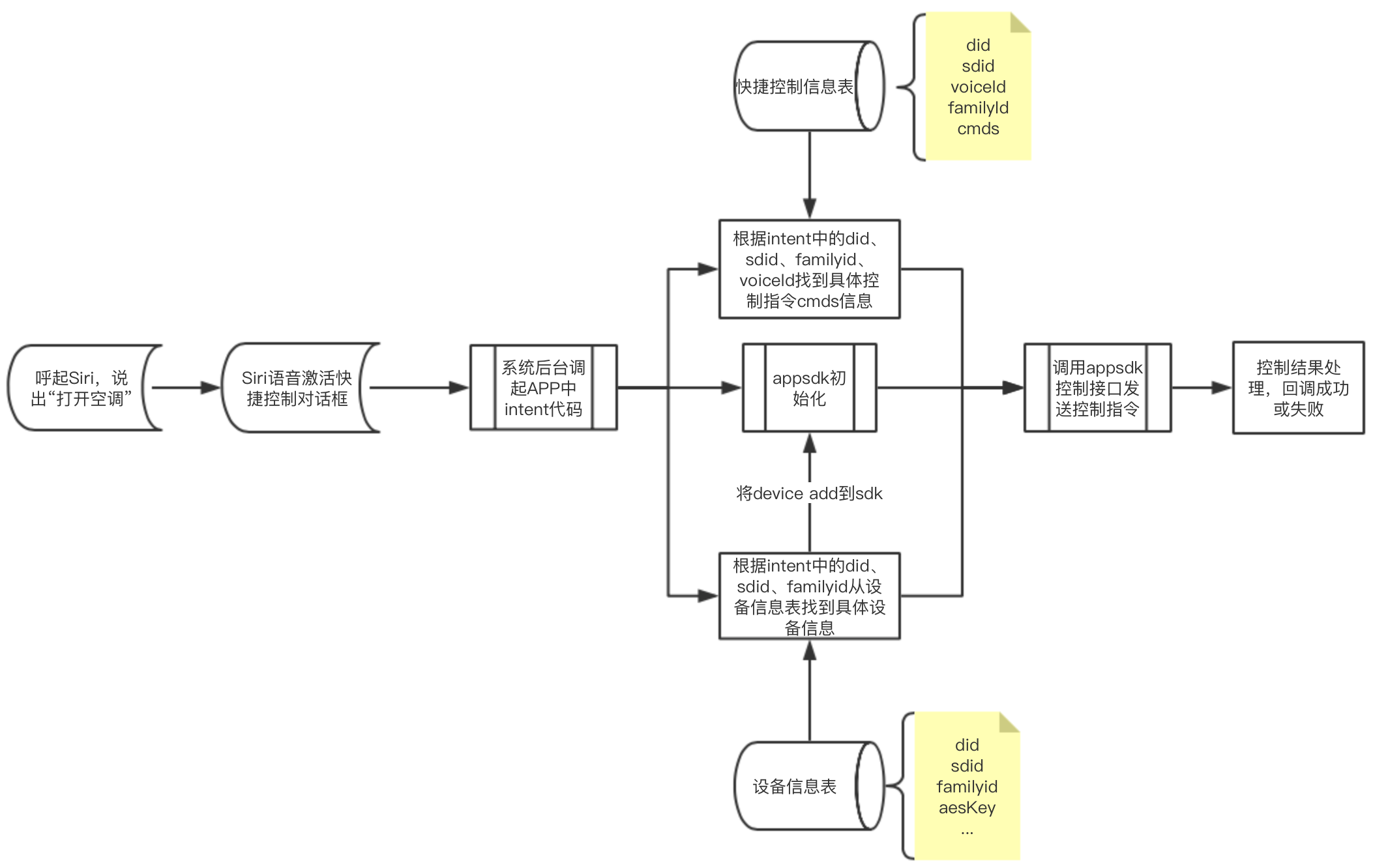
# 方案实现

使用Xcode自带的Intents及Intents UI框架，通过添加自定义语音交互页面，完成Siri快捷控制的功能，其中设备控制走appsdk现有接口。

# Siri快捷控制添加



# Siri快捷控制流程图



# 数据库设计

使用Siri shortcuts，需要用到两张表，一张是完整的设备信息表（包含did、sdid、familyId、aeskey、pid等），一张是shortcut与设备信息的关联表。设备信息表不做过多阐述。

快捷控制数据库表结构

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| voiceId | long | 对应快捷控制的voiceId，native自动生成。 |
| did | varchar(32) | 设备did |
| sdid | varchar(32) | 设备的sdid |
| familyId | varchar(32) | 设备所在家庭id，防止单个设备属于多个家庭的情况发生 |
| cmds | varchar(512) | 快捷控制指令，通过appsdk发送；也可通过http接口发送，如云定时中的延时功能。 |

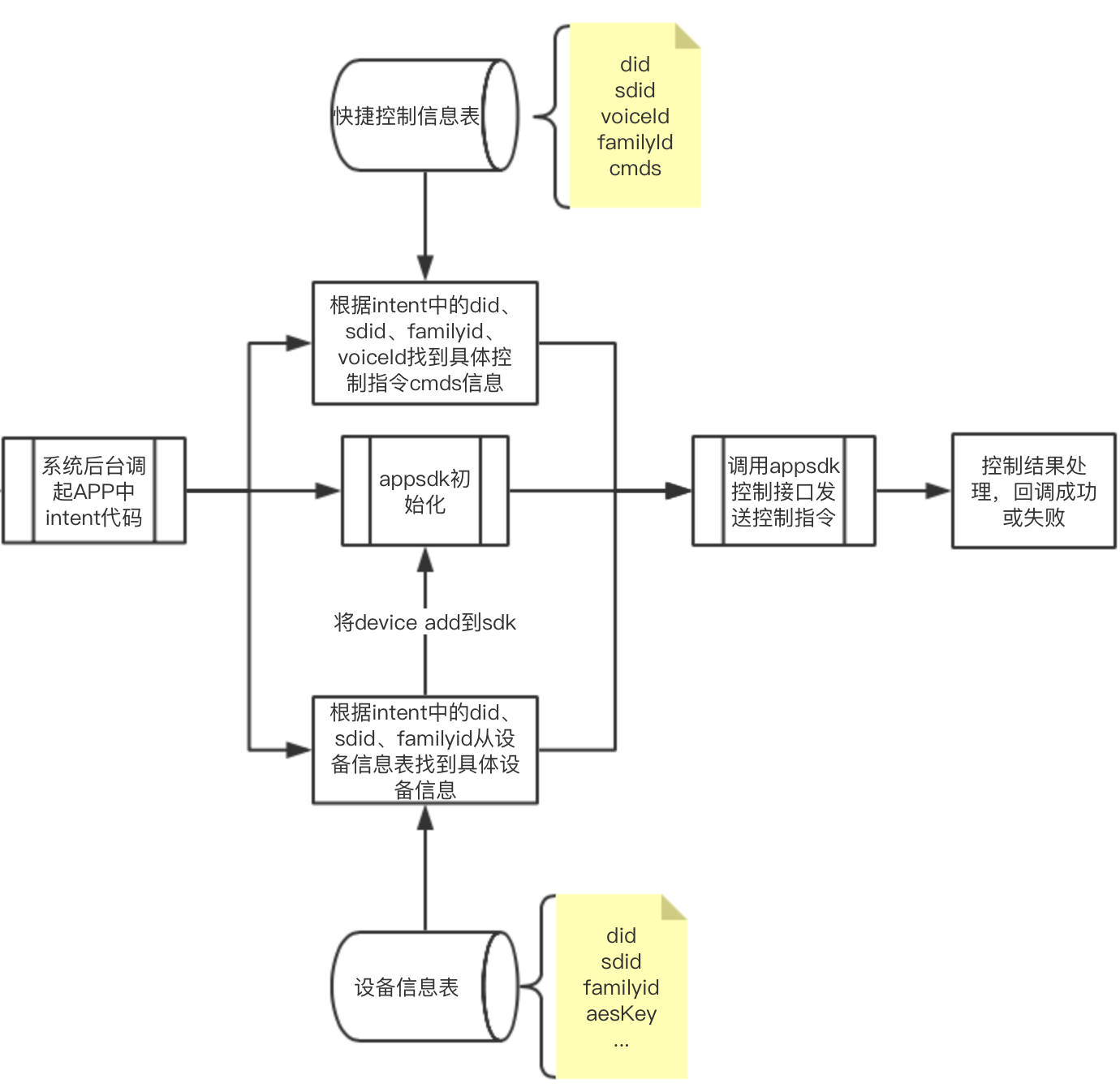
# APP通用内容

## 数据通用处理

1. 设备信息表及快捷控制数据表为通用表，这里会对两张表进行封装打包成sdk（暂定SiriShortcutsDataSDK），暴露相应增、删、改、查接口。
2. Intent参数结构统一化处理，如DeviceControlIntent中必须有did、sdid、familyId、voiceId这四个字段。

|  |  |
| --- | --- |
| 字段 | 描述 |
| did | 设备的did信息 |
| sdid | 设备的sdid信息 |
| familyId | 设备所在家庭的familyId信息 |
| voiceId | 设备快捷控制语音id，有APP自己生成，且必须保证唯一 |

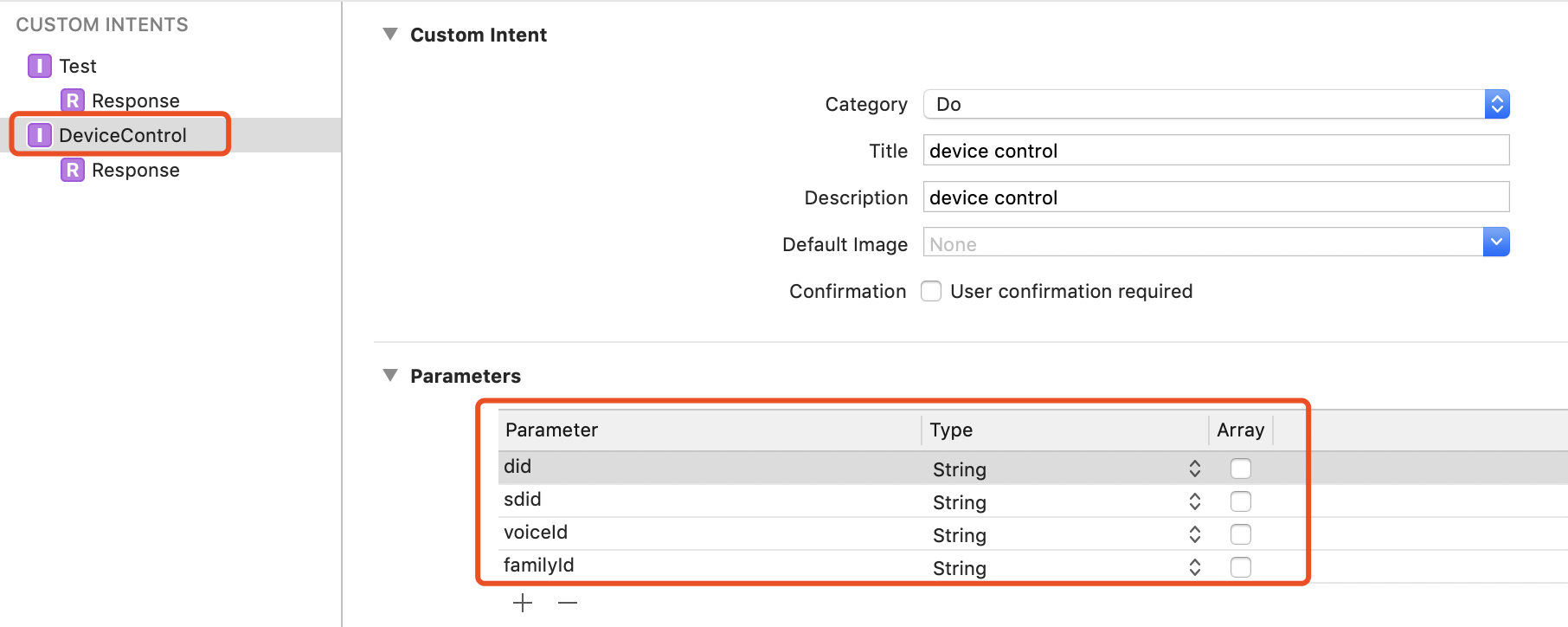
## 通用控制逻辑处理



1. appsdk初始化逻辑一致，在IntentHandler中handlerForIntent方法里进行初始化
2. 获取device信息，通过intent中的did、sdid、familyId从SiriShortcutsDataSDK提供的接口中获取相应完整的device信息。
3. 将device添加到appsdk中
4. 获取控制指令cmds，通过intent中的did、sdid、family、voiceId获从SiriShortcutsDataSDK提供的接口中获取对应快捷控制的cmds。
5. 正常设备控制调用appsdk的dnaControl接口发送控制指令
6. 处理返回结果

# APP非通用内容（自定义）

1. Intent需要每个工程单独创建，创建Intent时的名称（DeviceControl）可自定义，参数不可自定义，可新增不可删除，必须有did、sdid、familyId、voiceId字段

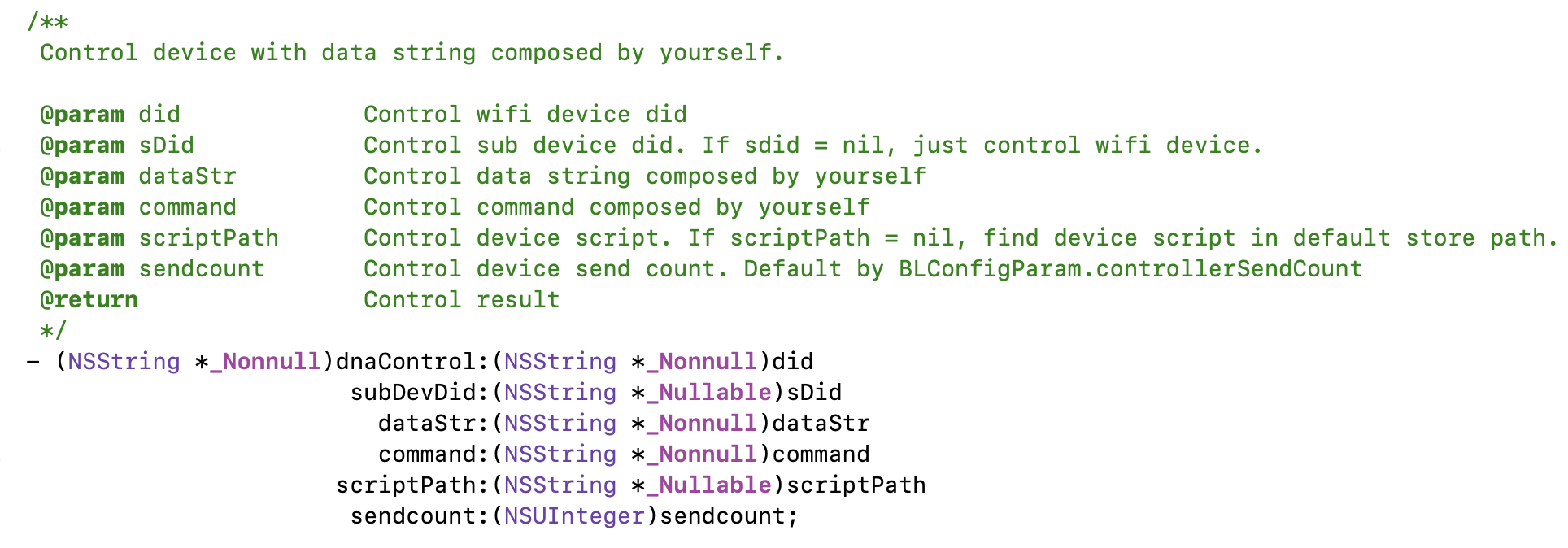


1. Intent UI可自定义
2. 其它除控制实现均为Xcode自动生成内容的统一内容

# 接口定义

控制相关使用appsdk现有控制接口，由于sdk需要在初始化成功后并且将device加入sdk后才可控制设备，所以在控制时初始化sdk及add device这两个步骤必不可少。延时控制采用云定时接口中的延时任务来实现。添加捷径则使用系统自有接口。

## 控制接口



## 延时控制接口

url:https://域名/appfront/v1/timertask/manage

方法:POST

header:

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| userId | string | 用户userid |
| loginsession | string | 登录获取的loginsession |
| familyId | string | 用户账号下家庭id |
| messageId | string | 随机生成 |

body:

{

"license": "",

"version":"v1", //v1表示老格式，v2表示新格式

"command":"upsert",//v1时有效

"type": 0, //0:devcontrol(设备控制); 1:scene(场景下发)

"endPoint":{

"did":"",

"pid":"",

"subdevdid":"",

"subdevpid":"",

"devicetypeflag":2, //0代表普通设备,1代表虚拟设备,2代表分组设备

"aeskey":"",//为空即可，从家庭中获取

"aesid":0,//为０即可，从家庭中获取

"extern":""//暂时不会使用

},

"payload":{

"type":0,//0定时任务，2周期任务

"index":3,　//如果有此参数，表示修改，不是新建，标示修改的序号

"enable":true,//任务开关

"timezone":"Asia/Chongqing",

"repeat":[1,2,3,4,5,6,7],//type为２关心，标示周期定时的执行时间，为空则在最近匹配时执行

"time":"yyyy-MM-dd hh:mm:ss",//格式：yyyy-MM-dd hh:mm:ss　或hh:mm:ss

"data":{"params":["pwr"],"vals":[[{"idx":1,"val":1}]]}

}

}

返回：

{

"status": 0,

"msg": "",

"payload":{

"timerlist":[

{

"index":0,

"timezone":"Asia/Chongqing",

"enable":true,//任务开关

"time":"yyyy-MM-dd hh:mm:ss"//格式：yyyy-MM-dd hh:mm:ss　或hh:mm:ss

}

],

"periodlist":[

{

"index":0,

"enable":true,//任务开关

"timezone":"Asia/Chongqing",

"repeat":[1,2],

"time":"yyyy-MM-dd hh:mm:ss"//格式：yyyy-MM-dd hh:mm:ss　或hh:mm:ss

}

]

}

}