SaiPush API Reference for Android - v1.0.4

概述

SaiPush是实现智能音箱设备和移动设备之间消息推送、记录同步等功能的设备通信解决方案。

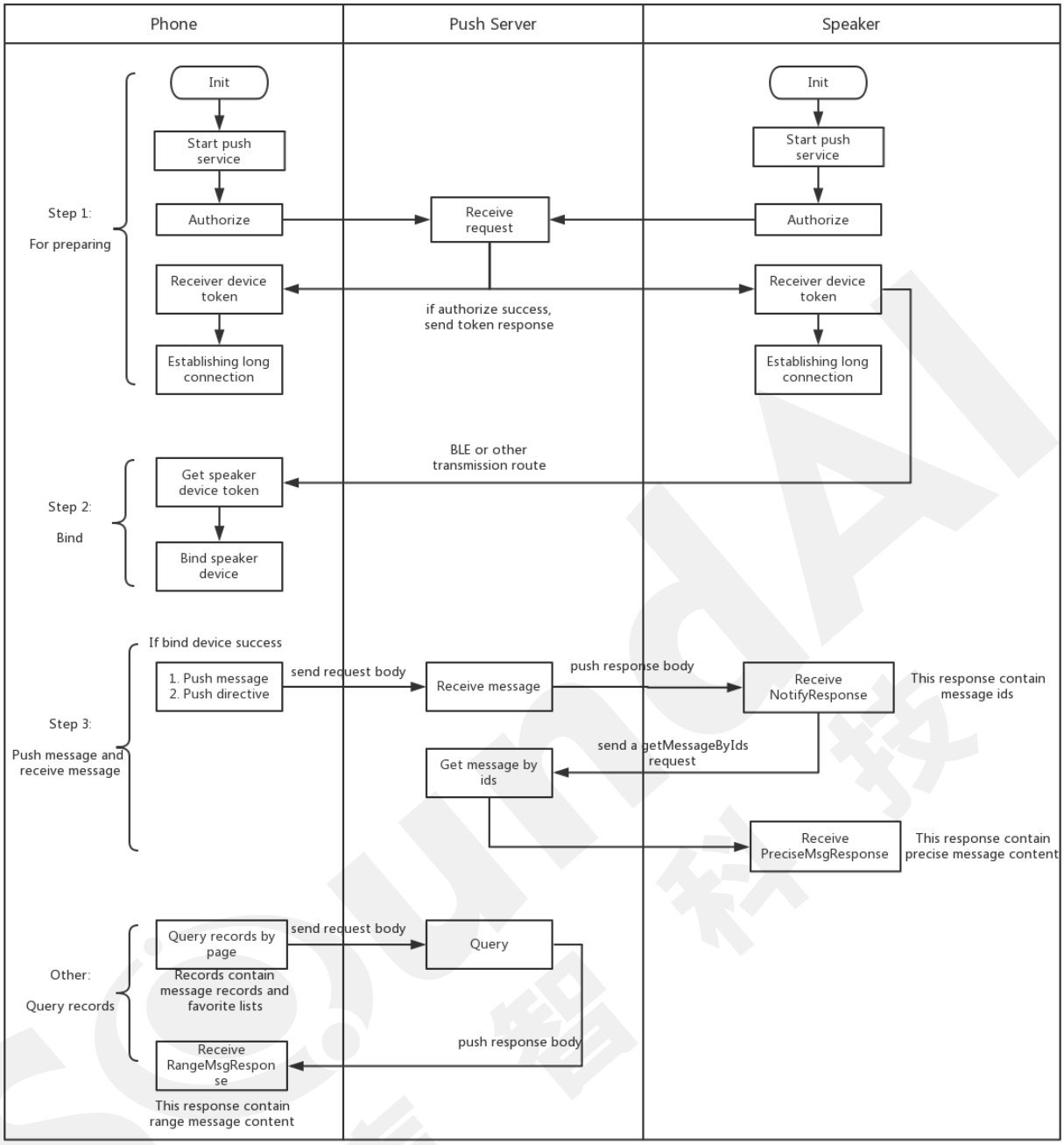
目前支持功能包括：

1. 消息推送、同步
2. 消息记录查询
3. 喜好、收藏内容存储
4. 提醒内容存储
5. 指令控制音箱设备
6. 智能家居

名词解释

名词	描述
SAI_PUSH_KEY	Push Server对SDK的授权密钥。如果Key不正确或失效，则无法使用相关功能。
Device Id	开发者提供的当前设备或当前用户(移动端app)的唯一标识。
Device Token	Push Server提供给设备的设备密钥，使用Push服务的必要参数
Ids	Payload和RequestBody中的字段，是消息的标识，用于查找、更新或删除该条消息(记录)。
Notify Id	此次推送的标识。
Dialog Id	主要用于Trace使用。

交互流程



集成步骤

- 解压缩SDK集成压缩包。
- 复制 libs/saipush-release-x.x.x.aar 到工程 libs/ 目录下。
- 配置AndroidManifest.xml

说明 1： 使用aar时需要配置build.gradle文件。

```
repositories {
    flatDir {
        dirs 'libs'
    }
```

```
}

dependencies {
    implementation fileTree(include: ['*.jar'], dir: 'libs')

    ...

    implementation (name:'saipush-release-x.x.x', ext:'aar')
}
```

> aar支持armeabi，armeabi-v7a，arm64-v8a平台，其他平台暂不支持。

如果需要指定当前支持的平台版本，可以在build.gradle中配置：

```
defaultConfig {
        ndk {
            // 配置相应需要的平台即可
            abiFilters "armeabi-v7a", "arm64-v8a"
        }
    }
```

说明 2： 根据该说明或demo工程中AndroidManifest.xml样例文件，来配置应用程序项目的AndroidManifest.xml。

> 注释标注为**Required**的部分为必须配置项。

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="Your Package Name">

    <!--Required:  接收PushReceiver必须权限-->
    <permission
          android:name="Your Package Name.permission.SAIPUSH"
          android:protectionLevel="signature" />

    <uses-permission android:name="Your Package Name.permission.SAIPUSH" />

    <application
        ...>

        <!--Required:  授权App Key-->
        <meta-data
            android:name="SAI_PUSH_KEY"
            android:value="soundai_push_app_key" />

        ...

        <!--接收推送的消息或响应-->
        <receiver
            android:name=".push.PushReceiver"
            android:enabled="true"
            android:exported="false">
```

```
            <intent-filter>
                <action android:name="saipush.intent.receiver.AUTH" />
                <action android:name="saipush.intent.receiver.BIND" />
                <action android:name="saipush.intent.receiver.PUSH" />
                <action android:name="saipush.intent.receiver.NOTIFY" />
                <action android:name="saipush.intent.receiver.RECEIVE_MESSAGE" />
                <action android:name="saipush.intent.receiver.RECEIVE_RECORDS" />
                <action android:name="saipush.intent.receiver.SMART_DEVICES" />
                <action android:name="saipush.intent.receiver.EXCEPTION" />
            </intent-filter>
        </receiver>

    </application>

</manifest>
```

权限说明

权限	用途
Your Package Name.permission.SAIPUSH	注册接收消息广播必须的权限，否则无法获取到推送消息。
android.permission.INTERNET	允许应用可以访问网络。
android.permission.ACCESS_NETWORK_STATE	允许应用获取网络状态。
android.permission.ACCESS_WIFI_STATE	允许应用获取WIFI状态。

混淆说明

请在工程的混淆文件中添加以下配置：

```
-keep class com.soundai.** {*;}
-keep class org.sai.pushservice.** {*;}
```

API说明

SaiPush

SaiPush类中包含应用程序使用Push服务用到的主要函数。

Class

- Options

Function

- init
- stop

- bindDevice
- unbindDevice
- getBoundDevices
- pushMessage
- pushDirective
- getMessagesByIds
- queryRecordsByPage
- refreshSmartDevices
- querySmartDevices
- isInit
- getSpeakerDeviceToken

Static Field

DeviceType

SaiPush初始化时必要参数，用于区分Phone和Speaker。

- DeviceType.SPEAKER
  - 设备为音箱时，设置该属性，只可被绑定
- DeviceType.PHONE
  - 设备为移动设备时，设置该属性，可绑定多台Speaker

RequestBody

SaiPush中发送请求需要的参数实体。

Class

- AddMsgRequestBody
- PreciseMsgRequestBody
- RangeMsgRequestBody
- UpdateRecordRequestBody
- RemoveRecordRequestBody

Static Field

RequestType

SaiPush中推送的消息数据类型。

- RequestType.MESSAGE
  - 普通消息类型
- RequestType.ALARM
  - 提醒消息类型
- RequestType.FAVORITE
  - 偏好消息类型
- RequestType.TRACE
  - 用于追踪测试的消息类型

ResponseBody

SaiPush可以接收到的响应实体。例如：获取到的认证请求的响应。

Class

- AuthResponseBody
- BindResponseBody
- PushResponseBody
- NotifyResponseBody
  - Notify
    - Payload
- PreciseMsgResponseBody
  - ContentsBean
- RangeMsgResponseBody
  - ContentsBean
- SmartDevicesResponse
  - Endpoint

Static Field

ResponseType

ResponseBody中区分响应类型。

- ResponseType.AUTH
  - 认证登录Push服务的Response类型
- ResponseType.BIND
  - 绑定设备的Response类型
- ResponseType.PUSH
  - 推送后的响应Response类型
- ResponseType.NOTIFY
  - 接收到推送消息的Response类型
- ResponseType.PRECISE
  - Message Id获取到的单条消息或消息列表
- ResponseType.RANGE
  - 根据Page获取到的消息列表

DirectiveParamProvider

SaiPush中推送指令需要的Parameters。

Function

- clientId
- dialogId
- query

- ip

Exception

Class

- AppKeyNotFoundException
  - 未设置SAI_PUSH_KEY
- DirectiveErrorExceptionn
  - 推送指令请求失败
- SaiPushException
  - 推送异常，errorCode详见ErrorCode

---

init

```
@MainThread
public static void init(Context context, Options options)
```

说明 初始化SaiPush，可以在Application的 `onCreate()` 或者Activity的 `onCreate()` 中执行，执行后会启动SaiPushService。

参数 *context*

- Application或Activity的上下文环境。

options

- 详见Options。

---

stop

```
@MainThread
public static void stop()
```

说明 停止SaiPush，执行后会停止服务。

---

bindDevice

```
public static void bindDevice(String speakerDeviceToken)
```

说明 绑定Speaker设备，只有Phone设备调用有效。一个Phone设备可以绑定多个Speaker设备。

参数 *speakerDeviceToken*

- Speaker设备的DeviceToken。

---

unbindDevice

```
public static void unbindDevice()
public static void unbindDevice(String speakerDeviceToken)
```

说明 解除绑定。 如果不传递参数，则解除当前绑定的Speaker设备;否则解除指定的SpeakerDeviceToken设备。

参数 *speakerDeviceToken*

- Speaker设备的DeviceToken。

---

getBoundDevices

```
public static void getBoundDevices()
```

说明 Speaker端调用获取被绑定的Phone端DeviceToken; Phone端调用获取已绑定的Speaker端DeviceToken列表。

---

pushMessage

```
public static void pushMessage(RequestBody body)
```

说明 推送消息的主要函数。

参数 *body*

- 请求body对象。接收RequestBody类型：AddMsgRequestBody、UpdateRecordRequestBody、RemoveRecordRequestBody。

---

pushDirective

```
public static void pushDirective(String localDeviceToken, final String
speakerDeviceToken, @Def.RequestType final String reqType, DirectiveParamProvider
provider)
```

说明 Phone推送操作指令给Speaker。例如：provider中query()函数参数传递“播放下一首”，Speaker则会收到“播放下一首”的NLP解析内容。

参数

localDeviceToken

- 当前设备的DeviceToken，**不区分Phone或Speaker**。

speakerDeviceToken

- Speaker端DeviceToken。

reqType

- 详见RequestType。

provider

- 详见DirectiveParamProvider。

---

getMessagesByIds

```
public static void getMessagesByIds(RequestBody body)
```

说明 根据Ids获取消息。 Ids从ContentsBean中获取。

参数 *body*

- 请求body对象。接收的RequestBody类型：PreciseMsgRequestBody。

---

queryRecordsByPage

```
public static void queryRecordsByPage(RequestBody body)
```

说明 查询历史消息记录。

参数 *body*

- 请求body对象。接收的RequestBody类型：RangeMsgRequestBody。

---

refreshSmartDevices

```
public static void refreshSmartDevices()
```

说明

刷新智能家居设备列表。

---

querySmartDevices

```
public static void querySmartDevices()
```

说明

查询当前帐号绑定的智能家居设备列表。

---

isInit

```
public static boolean isInit()
```

说明 SaiPush是否初始化成功。

返回值 *boolean*

- true 成功
- false 失败

---

getSpeakerDeviceToken

```
public static String getSpeakerDeviceToken()
```

说明 获取当前绑定或使用的SpeakerDeviceToken。

返回值 *String*

- SpeakerDeviceToken。

Options

初始化使用的配置

函数

```
public Options setDeviceType(@Def.DeviceType int deviceType)
```

- **说明**

  设置设备类型，详见DeviceType。

- **参数**

  详见DeviceType。

函数

```
public Options setDeviceId(@NonNull String deviceId)
```

- **说明**

  设置设备唯一标识，详见名词解释。

函数

```
public Options setRetryTime(int time)
```

- **说明**

  设置请求重试次数。

函数

```
public Options setRetryInterval(long interval)
```

- **说明**

设置请求重试间隔(单位:ms)。

AddMsgRequestBody

推送新消息的请求载体。

函数

```
public AddMsgRequestBody setContents(List<String> contents)
```

- **说明** 设置推送的具体字符串内容。
- **参数** *contents*
  - 字符串内容，可以是列表。
- **返回值** *AddMsgRequestBody*
  - 当前AddMsgRequestBody对象。

函数

```
public AddMsgRequestBody setNotify(boolean notify)
```

- **说明** 是否下发推送内容到远端。
- **参数** *notify*
  - true 推送内容到远端。
  - false 不推送内容，仅在云端保存，可以被查询。
- **返回值** *AddMsgRequestBody*
  - 当前AddMsgRequestBody对象。

函数

```
public AddMsgRequestBody setSpeakerDeviceToken(String deviceToken)
```

- **说明** 设置SpeakerDeviceToken。
- **参数** *deviceToken*
  - SpeakerDeviceToken。
- **返回值** *AddMsgRequestBody*
  - 当前AddMsgRequestBody对象。

函数

```
public AddMsgRequestBody setDialogId(String dialogId)
```

- **说明** 设置本次推送的会话Id，用于调试使用。
- **参数** *dialogId*
  - 会话Id，可以使用UUID，**长度不超过32位**。
- **返回值** *AddMsgRequestBody*

- 当前AddMsgRequestBody对象。

函数

```
public AddMsgRequestBody setReqType(String reqType)
```

- **说明** 设置请求类型。
- **参数** *reqType*
  - 详见RequestType。
- **返回值** *AddMsgRequestBody*
  - 当前AddMsgRequestBody对象。

PreciseMsgRequestBody

根据Payload中的Ids获取具体推送内容的请求载体。

函数

```
public PreciseMsgRequestBody setIds(List<Long> ids)
```

- **说明** 设置需要查询推送内容的Ids。
- **参数** *ids*
  - ids可以从Payload中获取。
- **返回值** *PreciseMsgRequestBody*
  - 当前PreciseMsgRequestBody对象。

函数

```
public PreciseMsgRequestBody setSpeakerDeviceToken(String deviceToken)
```

- **说明** 设置SpeakerDeviceToken。
- **参数** *deviceToken*
  - SpeakerDeviceToken。
- **返回值** *PreciseMsgRequestBody*
  - 当前PreciseMsgRequestBody对象。

函数

```
public PreciseMsgRequestBody setDialogId(String dialogId)
```

- **说明** 设置本次推送的会话Id，用于调试使用。
- **参数** *dialogId*
  - 会话Id，可以使用UUID，**长度不超过32位**。
- **返回值** *PreciseMsgRequestBody*
  - 当前PreciseMsgRequestBody对象。

函数

```
public PreciseMsgRequestBody setReqType(String reqType)
```

- **说明** 设置请求类型。
- **参数** *reqType*
  - 详见RequestType。
- **返回值** *PreciseMsgRequestBody*
  - 当前PreciseMsgRequestBody对象。

RangeMsgRequestBody

获取消息记录、定时记录或收藏列表记录等记录的请求载体。

函数

```
public RangeMsgRequestBody setPageNum(int pageNum)
```

- **说明** 设置需要查询的页码。**默认从0开始，第0页为最新记录**。
- **参数** *pageNum*
  - 页码。
- **返回值** *RangeMsgRequestBody*
  - 当前RangeMsgRequestBody对象。

函数

```
public RangeMsgRequestBody setPageSize(int pageSize)
```

- **说明** 设置一页查询的消息数量。
- **参数** *pageSize*
  - 一页的消息数量。
- **返回值** *RangeMsgRequestBody*
  - 当前RangeMsgRequestBody对象。

函数

```
public RangeMsgRequestBody setSpeakerDeviceToken(String deviceToken)
```

- **说明** 设置SpeakerDeviceToken。
- **参数** *deviceToken*
  - SpeakerDeviceToken。
- **返回值** *RangeMsgRequestBody*
  - 当前RangeMsgRequestBody对象。

函数

```
public RangeMsgRequestBody setDialogId(String dialogId)
```

- **说明** 设置本次推送的会话Id，用于调试使用。
- **参数** *dialogId*
  - 会话Id，可以使用UUID，**长度不超过32位**。
- **返回值** *RangeMsgRequestBody*
  - 当前RangeMsgRequestBody对象。

函数

```
public RangeMsgRequestBody setReqType(String reqType)
```

- **说明** 设置请求类型。
- **参数** *reqType*
  - 详见RequestType。
- **返回值** *RangeMsgRequestBody*
  - 当前RangeMsgRequestBody对象。

UpdateRecordRequestBody

更新记录的请求实体。**一次只能更新一条**。

函数

```
public UpdateRecordRequestBody setId(long id)
```

- **说明** 设置需要更新的记录Id。
- **参数** *id*
  - 记录Id。
- **返回值** *UpdateRecordRequestBody*
  - 当前UpdateRecordRequestBody对象。

函数

```
public UpdateRecordRequestBody setContents(List<String> contents)
```

- **说明** 设置需要更新的记录的内容。
- **参数** *contents*
  - 具体的更新内容。
- **返回值** *UpdateRecordRequestBody*
  - 当前UpdateRecordRequestBody对象。

函数

```
public UpdateRecordRequestBody setNotify(boolean notify)
```

- **说明** 是否下发推送到远端。
- **参数** *notify*
  - true 推送内容到远端。
  - false 不推送内容，仅在云端保存，可以被查询。
- **返回值** *UpdateRecordRequestBody*
  - 当前UpdateRecordRequestBody对象。

函数

```
public UpdateRecordRequestBody setSpeakerDeviceToken(String deviceToken)
```

- **说明** 设置SpeakerDeviceToken。
- **参数** *deviceToken*
  - SpeakerDeviceToken。
- **返回值** *UpdateRecordRequestBody*
  - 当前UpdateRecordRequestBody对象。

函数

```
public UpdateRecordRequestBody setDialogId(String dialogId)
```

- **说明** 设置本次推送的会话Id，用于调试使用。
- **参数** *dialogId*
  - 会话Id，可以使用UUID，**长度不超过32位**。
- **返回值** *UpdateRecordRequestBody*
  - 当前UpdateRecordRequestBody对象。

函数

```
public UpdateRecordRequestBody setReqType(String reqType)
```

- **说明** 设置请求类型。
- **参数** *reqType*
  - 详见RequestType。
- **返回值** *UpdateRecordRequestBody*
  - 当前UpdateRecordRequestBody对象。

RemoveRecordRequestBody

删除记录的请求实体。可以一次删除多条记录。

函数

```
public RemoveRecordRequestBody setIds(List<Long> ids)
```

- **说明** 设置需要删除的记录对应的Id列表。
- **参数** *ids*
  - 记录的Id列表。
- **返回值** *RemoveRecordRequestBody*
  - 当前RemoveRecordRequestBody对象。

函数

```
public RemoveRecordRequestBody setNotify(boolean notify)
```

- **说明** 是否下发推送到远端。
- **参数** *notify*
  - true 推送内容到远端。
  - false 不推送内容，仅在云端保存，可以被查询。
- **返回值** *RemoveRecordRequestBody*
  - 当前RemoveRecordRequestBody对象。

函数

```
public RemoveRecordRequestBody setSpeakerDeviceToken(String deviceToken)
```

- **说明** 设置SpeakerDeviceToken。
- **参数** *deviceToken*
  - SpeakerDeviceToken。
- **返回值** *RemoveRecordRequestBody*
  - 当前RemoveRecordRequestBody对象。

函数

```
public RemoveRecordRequestBody setDialogId(String dialogId)
```

- **说明** 设置本次推送的会话Id，用于调试使用。
- **参数** *dialogId*
  - 会话Id，可以使用UUID，**长度不超过32位**。
- **返回值** *RemoveRecordRequestBody*
  - 当前RemoveRecordRequestBody对象。

函数

```
public RemoveRecordRequestBody setReqType(String reqType)
```

- **说明** 设置请求类型。

- **参数** *reqType*
  - 详见RequestType。
- **返回值** *RemoveRecordRequestBody*
  - 当前RemoveRecordRequestBody对象。

AuthResponseBody

认证结果的响应实体。

函数

```
public String getDeviceToken()
```

- **说明** 获取认证后本端的DeviceToken。
- **返回值** *String*
  - LocalDeviceToken。

函数

```
@Def.ResponseType
public int getRespType()
```

- **说明** 获取Response类型。
- **返回值** *int*
  - 详见ResponseType。

BindResponseBody

绑定设备、解绑设备、获取绑定设备的响应实体。

函数

```
public List<String> getBindDeviceTokens()
```

- **说明** 获取当前设备的绑定列表。
- **返回值** *List*
  - Phone，获取已绑定Speaker的Token列表。
  - Speaker，获取当前被绑定的Phone的Token。

函数

```
@Def.ResponseType
public int getRespType()
```

- **说明** 获取Response类型。
- **返回值** *int*

- 详见ResponseType。

PushResponseBody

`PushMessage()`或`PushDirective()`请求的响应实体。

函数

```
public int getCode()
```

- **说明** `PushMessage()`或`PushDirective()`请求的响应。用于判断该推送是否成功。
- **返回值** *int*
  - 详见ErrorCode。

函数

```
public String getMsg()
```

- **说明** 该推送的状态码的描述信息。
- **返回值** *String*
  - 描述信息

函数

```
public List<Long> getIds()
```

- **说明** 当前消息推送的Id列表（如果推送多条消息）。
- **返回值** *List*
  - 推送消息列表的对应Id列表。

函数

```
@Def.ResponseType
public int getRespType()
```

- **说明** 获取Response类型。
- **返回值** *int*
  - 详见ResponseType。

NotifyResponseBody

当远端设置下发推送后，本端会接收到该通知响应实体，包含推送内容Id等相关信息，需要再根据Id请求`getMessageByIds()`获取具体的推送内容。

函数

```
public List<Notify> getNotifies()
```

- **说明** 推送载体，包含本次推送的相关信息。
- **返回值** *List*
  - 详见Notify。

函数

```
@Def.ResponseType
public int getRespType()
```

- **说明** 获取Response类型。
- **返回值** *int*
  - 详见ResponseType。

Notify

包含了推送Id、推送时间戳和推送内容载体信息。

函数

```
public String getNotifyId()
```

- **说明** 获取本次推送的推送Id，用于调试使用。
- **返回值** *String*
  - 推送Id。

函数

```
public String getTimestamp()
```

- **说明** 获取本次推送的时间戳。
- **返回值** *String*
  - 时间戳。

函数

```
public Payload getPayload()
```

- **说明** 获取推送内容载体。
- **返回值** *Payload*
  - 详见Payload。

Payload

推送内容载体，包含了推送内容的消息Ids，可以根据Ids请求`getMessageByIds()`获取具体的推送内容。

函数

```
public List<Long> getIds()
```

- **说明** 获取推送内容的对应Id列表。 **需要使用该Id列表，请求`getMessageByIds()`获取具体的推送消息内容。**
- **返回值** *List*
  - 具体推送消息的Id列表。

函数

```
public String getType()
```

- **说明** 推送的消息类型。
- **返回值** *String*
  - 与RequestType中包含的类型相同。

函数

```
public String getOp()
```

- **说明** 本次推送的操作类型。
- **返回值** *String*
  - "ADD"、"UPDATE"、"DELETE"三种类型中的一种，对应增改删三种操作。

函数

```
public String getDialogId()
```

- **说明** 获取本次推送的会话Id。
- **返回值** *String*
  - 会话Id。

PreciseMsgResponseBody

`getMessageByIds()`函数请求的响应实体，包含具体的推送内容消息。

函数

```
public int getCode()
```

- **说明** `getMessageByIds()`请求的响应。用于判断该请求是否成功。
- **返回值** *int*
  - 详见ErrorCode。

函数

```
public String getMsg()
```

- **说明** 该请求的状态码的描述信息。
- **返回值** *String*
  - 描述信息

函数

```
public List<ContentsBean> getContents()
```

- **说明** 获取具体的推送消息内容。
- **返回值** *List*
  - 详见ContentsBean。

函数

```
@Def.ResponseType
public int getRespType()
```

- **说明** 获取Response类型。
- **返回值** *int*
  - 详见ResponseType。

RangeMsgResponseBody

`queryRecordsByPage()` 函数请求的响应实体，包含了查询的记录内容。

函数

```
public int getCode()
```

- **说明** `queryRecordsByPage()` 请求的响应。用于判断该请求是否成功。
- **返回值** *int*
  - 详见ErrorCode。

函数

```
public String getMsg()
```

- **说明** 该请求的状态码的描述信息。
- **返回值** *String*
  - 描述信息

函数

```
public List<ContentsBean> getContents()
```

- **说明** 获取查询的消息记录的具体内容。
- **返回值** *List*
  - 详见ContentsBean。

函数

```
public int getTotalCount()
```

- **说明** 获取消息记录的总数。
- **返回值** *int*
  - 消息记录总数量。

函数

```
@Def.ResponseType
public int getRespType()
```

- **说明** 获取Response类型。
- **返回值** *int*
  - 详见ResponseType。

ContentsBean

PreciseMsgResponseBody、RangeMsgResponseBody中包含的具体消息内容。

函数

```
public int getId()
```

- **说明** 获取该条消息的Id。
- **返回值** *int*
  - 消息Id。

函数

```
public String getContent()
```

- **说明** 获取该条消息的具体字符串内容。
- **返回值** *String*
  - 字符串消息内容。

SmartDevicesResponse

函数

```
public List<Endpoint> getEndpoints()
```

- **说明** 获取终端设备信息。
- **返回值** *Endpoint*
    - 终端设备实体。

---

merchantId

```
public DirectiveParamProvider merchantId(String merchantId)
```

说明 设置客户Id。

参数 *merchantId*

- 客户Id

返回值 *DirectiveParamProvider*

- 当前DirectiveParamProvider对象。

---

clientId

```
public DirectiveParamProvider clientId(String clientId)
```

说明 设置客户端ID。

参数 *clientId*

- 客户端ID

返回值 *DirectiveParamProvider*

- 当前DirectiveParamProvider对象。

---

dialogId

```
public DirectiveParamProvider dialogId(String dialogId)
```

说明 设置会话ID，长整型。

参数 *dialogId*

- 会话ID，**长整型的String**

返回值 *DirectiveParamProvider*

- 当前DirectiveParamProvider对象。

---

query

```
public DirectiveParamProvider query(String query)
```

说明 具体的指令内容。

参数 *query*

- 例如：播放下一首、今天天气怎么样

返回值 *DirectiveParamProvider*

- 当前DirectiveParamProvider对象。

---

ip

```
public DirectiveParamProvider ip(String ip)
```

说明 设置客户端IP地址。如果不填IP，对于天气类问题将默认为北京。

参数 *ip*

- ip

返回值 *DirectiveParamProvider*

- 当前DirectiveParamProvider对象。

ErrorCode

```
0  ：请求成功
400：请求格式错误，比如没有填写device_token
401：Token不正确
403：Token已过期或被禁用
404：URL路径不对
500：服务器内部错误，可稍后重试
```