阿里巴巴AiLabs Genie SDK

使用手册（Android版）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 版本 | 修订章节 | 修订原因 | 日期 | 修订人 |
| 1 | 1.0 | 初始版本 |  | 2018.4.22 |  |
| 2 | 2.0 |  | 简化SDK接入流程 | 2018/7/27 | 紫岸 |
| 3 | 2.1 |  | 更新快速开始章节；  增加多轮对话章节；  增加demo code repo地址；  增加常用对话ProtocolData数据示例 | 2018/8/7 | 紫岸 |
| 4 | 2.2 |  | 更新媒体播控 | 2018/8/15 | 梓同 |
| 5 | 2.3 |  | 更新初始化接口；  增加音频输出控制接口  更新设备控制指令集 | 2018/8/24 | 紫岸 |
| 6 | 2.4 |  | 增加账号绑定部分 | 2018/9/6 | 梓同 |

目录

目录

[前言 1](#_Toc526950408)

[1 整体架构 2](#_Toc526950409)

[2 快速开始 3](#_Toc526950410)

[2.1 配置gradle 3](#_Toc526950411)

[2.1.1 添加maven仓库 3](#_Toc526950412)

[2.1.2 添加库依赖 3](#_Toc526950413)

[2.1.3 混淆配置 3](#_Toc526950414)

[2.1.4 集成prodconf.json 3](#_Toc526950415)

[2.2 初始化SDK 4](#_Toc526950416)

[2.3 获取Recorder实例 5](#_Toc526950417)

[2.4 实现接口IUiManager（可选） 5](#_Toc526950418)

[2.5 实现接口IPreOnNLPResult（可选） 6](#_Toc526950419)

[2.6 打通数据链路 6](#_Toc526950420)

[2.6.1 原始音频数据 6](#_Toc526950421)

[2.6.2 语音信号处理后的单路音频数据 6](#_Toc526950422)

[2.6.3 纯文本流数据 7](#_Toc526950423)

[2.6.4 近场模式 7](#_Toc526950424)

[2.7 控制音频输出 7](#_Toc526950425)

[3 输入模块 9](#_Toc526950426)

[3.1 语音输入 9](#_Toc526950427)

[3.2 多轮对话 10](#_Toc526950428)

[3.2.1 接口类：IMicController 10](#_Toc526950429)

[4 UI自定义 11](#_Toc526950430)

[4.1 UI 管理接口 11](#_Toc526950431)

[4.1.1 类：IBaseView 11](#_Toc526950432)

[4.2 业务数据 18](#_Toc526950433)

[4.2.1 类：ProtocolData 18](#_Toc526950434)

[4.2.2 类：ProtocolData.DataType 20](#_Toc526950435)

[4.2.3 类：ProtocolData.DataCode 22](#_Toc526950436)

[5 媒体播放控制接口 23](#_Toc526950437)

[5.1 安卓Mediaplayer封装 23](#_Toc526950438)

[5.1.1 类：MediaOutputBridge 23](#_Toc526950439)

[5.1.2 类：IMediaOutputListener 24](#_Toc526950440)

[5.2 影视端端方案 24](#_Toc526950441)

[6 第三方应用指令接入 25](#_Toc526950442)

[7 上下文信息 26](#_Toc526950443)

[7.1 上下文监听器 26](#_Toc526950444)

[7.1.1 类：TvContextListener 26](#_Toc526950445)

[7.2 业务数据 27](#_Toc526950446)

[7.2.1 类：PageType 27](#_Toc526950447)

[7.2.2 类：SoundType 28](#_Toc526950448)

[8 账号绑定（针对带屏设备） 30](#_Toc526950449)

[8.1 实现IUiManager中的onNotify方法 30](#_Toc526950450)

[8.2 将demo中的BindDeviceGuideActivity集成到自己的APK中 31](#_Toc526950451)

[8.3 将demo中的AndroidManifest集成到自己的APK中 31](#_Toc526950452)

[9 附录 32](#_Toc526950453)

[9.1 demo仓库地址 32](#_Toc526950454)

[9.2 常用技能ProtocolData数据示例 32](#_Toc526950455)

[9.3 设备控制预处理命令onPretreatedResult回调数据 40](#_Toc526950456)

[9.4 应用控制预处理命令onPretreatedResult回调数据 43](#_Toc526950457)

# 前言

AliGenie SDK是是阿里巴巴人工智能实验室(AI-Labs)面向软硬件厂商和开发者推出的，将人工智能中ASR语音识别、NLP自然语言处理、TTS语音合成等自然语言处理技术和能力对外共享的软件开发工具。

# 整体架构

整体架构如下图，可以分为以下几部分：

* Platform & OS： 系统相关服务和接口；
* Platform Implementation Layer：平台适配层，实现AliGenie的抽象接口；
* AliGenie Abstract Layer：AliGenie平台抽象层，包括语音输入/输出、交互、操控等对系统的依赖接口；
* AliGenie：提供语音交互、设备管理、音频播放等功能；
* APP Customization UI：应用上层实现UI页面；
* AliGenie APP SDK：提供第三方接入语音的SDK；
* Vender APP：第三方应用

其中，绿色模块有合作厂商适配，蓝色模块有AliGenie SDK提供；



# 快速开始

只需进行简单的几部操作就可以让您的设备具备语音交互的能力，赶快开始吧。

## 配置gradle

在进行开发前需要对gradle进行简单的配置工作。

### 添加maven仓库

在您工程目录根目录的gradle.build文件中加入如下maven仓库：

maven {  
 url 'http://maven.aliyun.com/nexus/content/repositories/releases/'  
 }

### 添加库依赖

将我方提供的sdk aar包和jar包放至app module的libs目录，并在app module的gradle.build文件中添加如下库依赖：

compile(name: 'allinone\_vxxx', ext: 'aar')

compile 'com.alibaba:fastjson:1.1.45'  
 compile 'com.aliyun.openservices:aliyun-log-android-sdk:0.4.0'

注：xxx表示sdk的版本号

### 混淆配置

如果您的APP需要进行混淆，请将如下配置添加到proguard-rules.pro文件中

-keep class com.alibaba.ailabs.\*\*{ \*; }

-keep class com.yunos.tv.\*\*{ \*; }

-keep class com.aliyun.sls.android.sdk.\*\*{ \*; }

### 集成prodconf.json

在开放平台下载配置文件：



然后将prodconf.json文件放入app module的assets目录。其中的biz\_type和biz\_group为各厂商申请到的，且需要和key匹配

**"biz\_type"**: **"XXX"**,  
**"biz\_group"**: **"XXX"**

env是用来切换配置线上或者预发的配置的

**"env"**: **"online"**,

## 初始化SDK

在使用SDK之前您需要对SDK进行初始化，初始化的方法为调用com.alibaba.ailabs.geniesdk\_adapter.core.AliGenieSDKAdapter的init方法，init方法的参数列表如下：

|  |  |  |
| --- | --- | --- |
| **参数名** | **参数类型** | **功能描述** |
| workpath | String | 设置工作目录，可以为null |
| uiManager | IUiManager | UI显示处理器，参见2.4 |
| recorder | BaseRecorder | 虚拟拾音器，不能为空 |
| nlpPreTreat | IPreOnNLPResult | NLP结果的预处理器，参见2.5 |

可参考demo code，见8.1节。

## 获取Recorder实例

工厂类com.alibaba.ailabs.geniesdk\_adapter.audioin.RecorderFactory提供了四个静态方法分别返回四种不同的Recorder实例，用于接收原始或消噪算法处理后的音频数据并完成整个数据链路的组装，**不能为空**，SDK中已提供四种默认拾音器的实现，开发者可通过工厂类com.alibaba.ailabs.geniesdk\_adapter.audioin.RecorderFactory获取拾音器实例，无需自己实现

方法描述如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **方法名** | **参数列表** | | **返回值** |
| getRawDataFarFieldRecorder | int frequence //采样率  int channel //通道数  int source //输入源  int format //采样格式 | | RawDataFarFieldRecorder //原始音频数据拾音器（远场） |
| getOutAudioDataFarFieldRecorder | int frequence //采样率  int channel //通道数  int source //输入源  int format //采样格式 | | OutAudioDataFarFieldRecorder //单路音频数据拾音器（远场） |
| getTextStreamFarFieldRecorder | 无 | TextStreamFarFieldRecorder //文本流数据拾音器 | |
| getNearFieldRecorder | int frequence //采样率  int channel //通道数  int source //输入源  int format //采样格式 | | NearFieldRecorder //近场模式拾音器 |

## 实现接口IUiManager（可选）

在自定义类中实现IUiManager接口，SDK初始化方法中需传入IUiManager的实例。接口实现可参考demo code。

注意在IUiManager中有几个重要接口需要和demo中的返回值保持一样：

* getTvContext(boolean isScreen, String thridContext)
* isUiShowing()
* onNotify接口有关于账号绑定部分可参看8.1节

## 实现接口IPreOnCommand（可选）

在自定义类中实现IPreOnCommand接口,所有天猫精灵NLP服务下发的指令都会先经过IPreOnCommand具体实现类做预处理，如果厂商不处理该命令，则继续透传给SDK做内部的默认处理，通过返回值ReturnCode来通知SDK处理的结果：

|  |  |  |
| --- | --- | --- |
| **方法名** | **参数列表** | **返回值** |
| pretreatedNLPResult | *Int* ***sessionId String command*** | ReturnCode:   * CONTINUE = 0; 表示数据继续向SDK分发 * STOP = 1; 表示数据停止向SDK分发，并调用会触发UIManager调用hideUI来关闭界面 * STOP\_NO\_HIDE = 2;表示数据停止向SDK分发，保留界面 |

SDK初始化方法中可传入IPreOnCommand的实例，也可直接传null。

## 打通数据链路

这是最后一个环节，打通数据链路之后SDK就能正常工作了。数据链路的打通按照输入数据的不同分为以下四种情况：

### 原始音频数据

对于原始音频数据，首先通过RecordManager的getRecord方法获取RawDataFarFieldRecorder类型的拾音器实例，在唤醒引擎的唤醒回调中调用RawDataFarFieldRecorder的wakeup和doa方法，将唤醒的置信度、唤醒词和唤醒角度作为参数传入；之后调用RawDataFarFieldRecorder的receiveRawData方法将数据灌入链路；最后在VAD引擎判断语音结束后调用RawDataFarFieldRecorder的vadEnd方法结束本次对话，详见Demo工程示例代码。

### 语音信号处理后的单路音频数据

对于经消噪、回声消除等语音信号算法处理过的单路音频数据，首先通过RecordManager的getRecord方法获取OutAudioDataFarFieldRecorder类型的拾音器实例，在唤醒引擎的唤醒回调中调用OutAudioDataFarFieldRecorder的wakeup和doa方法，将唤醒的置信度、唤醒词和唤醒角度作为参数传入；之后调用OutAudioDataFarFieldRecorder的receiveVadData方法将数据灌入链路；最后在VAD引擎判断语音结束后调用OutAudioDataFarFieldRecorder的vadEnd方法结束本次对话，详见Demo工程示例代码。

### 纯文本流数据

对于纯文本流的数据，首先通过RecordManager的getRecord方法获取TextStreamFarFieldRecorder类型的拾音器实例，在唤醒引擎的唤醒回调中调用TextStreamFarFieldRecorder的wakeup和doa方法，将唤醒的置信度、唤醒词和唤醒角度作为参数传入；之后调用TextStreamFarFieldRecorder的receiveTextSteam方法将文本数据灌入链路；最后在VAD引擎判断语音结束后调用TextStreamFarFieldRecorder的vadEnd方法结束本次对话，详见Demo工程示例代码。

根据不同的数据类型选择不同拾音器，然后在合适的时机调用拾音器的方法完成数据链路的打通，之后就可以体验到语音交互服务了。

### 近场模式

近场模式无需开发者关心数据链路的打通，开发者只需关心何时调用NearFieldRecorder的wakeup方法开始拾音，以及何时调用vadEnd方法结束拾音，SDK会完成拾音、音频数据处理、数据链路打通这一系列工作。

## 控制音频输出

您可以使用com.alibaba.ailabs.geniesdk\_adapter.core.AliGenieSDKAdapter的下列方法控制精灵音频的输出：

|  |  |  |  |
| --- | --- | --- | --- |
| **方法名** | **参数列表** | **返回值** | **功能描述** |
| suspend | 无 | 无 | 暂停精灵一切音频播放并退出UI显示 |
| resume | 无 | 无 | 恢复已经存在的播放列表 |

# 输入模块

AliGenie模块提供对输入设备的抽象，支持语音遥控器、远场以及外挂设备，可以是纯录音数据、文本数据以及协议数据等多种输入类型。

## 语音输入

语音输入部分包括三个模块：语音信号处理、语音唤醒和语音识别。根据具体平台的项目需求可以没有语音信号处理，直接传输麦克风录音给SDK；也可以没有语音唤醒模块，直接通过按键或者其它方式唤醒。

为了同时支持以上几种情况，SDK内置四种虚拟拾音器：

* RawDataFarFieldRecorder：接收原始音频数据，经语音信号处理算法处理后合成一路音频上传至服务端进行语音识别和语义识别；
* OutAudioDataFarFieldRecorder：接收语音信号处理后的单路音频数据，上传至服务端进行语音识别和语义识别；
* TextStreamFarFieldRecorder：接收三方ASR服务进行语音识别后的文本流，上传至服务端进行语义识别；
* NearFarFieldRecorder：近场模式，不支持语音唤醒，需要开发者指定开麦和关麦的时机，开麦后SDK自行录制音频并上传至服务端进行语义识别；

前三个类具有共同的父类FarFieldRecorder，下表为三个类的public接口及接口描述：

|  |  |  |  |
| --- | --- | --- | --- |
| **接口名** | **入参列表** | **返回值** | **接口描述** |
| **enableVad** | **boolean enable** | **无** | **控制是否开启vad的开关，默认为开启** |
| **setChannelCount** | **int count** | **无** | **设置音频通道数，默认值为初始化时传入的通道数** |
| **wakeup** | **double confidence //唤醒置信度**  **String wakeupWord //唤醒词** | **无** | **通知SDK设备已唤醒，准备接收音频/文本数据** |
| **doa** | **int angle //唤醒角度** | **无** | **通知SDK唤醒角度** |
| **vadBegin** | **无** | **无** | **通知SDK语音对话开始** |
| **vadEnd** | **无** | **无** | **通知SDK语音对话结束** |
| **receiveRawData** | **byte[] //原始音频数据** | **int //音频处理结果 -1失败 0 成功** | **将原始音频数据灌入SDK** |
| **receiveVadData** | **byte[] //语音信号处理后的数据** | **int //音频处理结果 -1失败 0 成功** | **将语音信号处理后的数据灌入SDK** |
| **receiveTextSteam** | **String question //实时文本流**  **boolean isFinished //对话是否结束** | **无** | **将文本流数据灌入SDK** |
| **setMultiTalkCallback** | **IMultiTalkCallback callback //注册多轮对话回调** | **无** | **开启多轮对话时会回调IMultiTalkCallback的onMultiTalkStart的方法，开发者需要在该方法中执行开麦动作** |

## 多轮对话

远程模式的某些场景下会出现多轮对话的场景，比如闹钟技能，用户说“我要定闹铃”后nlu服务端回复“请问您要定什么时间的闹铃？”此时用户无需说唤醒词唤醒设备，而是直接回复“明天早上七点”便可完成闹钟设定，实现多轮对话需要开发者调用FarFieldRecorder的接口setMicController注册开麦回调。

### 接口类：IMicController

接口类说明：开麦回调接口，支持多轮对话必须实现的接口。

函数说明：

|  |  |
| --- | --- |
| **函数名称** | **功能描述** |
| openMic | 开麦回调接口，开发者需要在这个接口中执行开麦动作。 |

可参考demo code，见8.1节。

# UI自定义

AILabs Genie SDK提供了数据与显示的分享，提供了UI显示层的抽象，支持第三方客户做自定义UI开发。

## UI 管理接口

### 类：IBaseView

**类说明：**UI定制化的核心接口，接口会遵循整个会话流程时序，将AILabs的UI标准化数据透出。

* **成员变量**

**成员变量说明：**

|  |  |  |
| --- | --- | --- |
| **变量名称** | **类型** | **变量描述** |
| NOTIFY\_TYPE\_ONLINE\_STATUS | int | 取值0；主动上报类型：是否在线，指设备是否连上AiLabs服务器。 |
| NOTIFY\_TYPE\_ACTIVE\_STATUS | int | 取值1；主动上报类型：激活状态。 |
| NOTIFY\_TYPE\_LANCONNECTOR\_STATUS | int | 取值2；主动上报类型：局域网通信数据状态。 |
| NOTIFY \_THIRDAPP\_CONTEXT | int | 取值3；主动上报类型：第三方应用上下文信息。 |
| NOTIFY \_PUSH\_MESSAGE | int | 取值4；主动上报类型：推送消息。 |
| NOTIFY \_AUDIO\_STATUS | int | 取值5；主动上报类型：播放状态消息。 |
| NOTIFY \_BINDER\_USER | int | 取值6；主动上报类型：用户绑定结果通知。 |
| NOTIFY \_QRCODE\_MESSAGE | int | 取值7；主动上报类型：用户绑定二维码消息。 |

**函数说明：**

|  |  |
| --- | --- |
| **函数名称** | **功能描述** |
| onRecordStart | 录音开始（会话开始）的通知。 |
| onStreaming | 会话开始后的实时ASR结果文本 |
| onVolume | 会话开始后的实时语音音量 |
| onRecordStop | 录音结束的通知。 |
| onPretreatedResult | 语义解析结果的预处理 |
| onRecognizeResult | 语义解析结果 |
| onNotify | 主动上报（非会话流程中） |
| hideUi | 隐藏界面 |
| isUiShowing | 界面是否显示中 |
| onLowMemory | 释放内存 |

**其他：**所有接口的回调均发生为主线程。

* **函数：onRecordStart**

**函数说明：**录音开始，即会话开始。

**入参说明：**

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **参数描述** |
| seesionId | int | 会话标识 |

**返回值说明：**

|  |  |  |
| --- | --- | --- |
| **返回值** | **类型** | **参数描述** |
| 无 | 无 | 无 |

**其他：**无

* **函数：onStreaming**

**函数说明：**会话开始后的实时ASR结果文本。

**入参说明：**

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **参数描述** |
| seesionId | int | 会话标识 |
| streamText | String | 实时ASR结果文本 |
| isFinish | boolean | 是否结束。true：已结束；false：未结束。 |

**返回值说明：**

|  |  |  |
| --- | --- | --- |
| **返回值** | **类型** | **参数描述** |
| 无 | 无 | 无 |

**其他：**无

* **函数：onVolume**

**函数说明：**会话开始后的实时语音音量。

**入参说明：**

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **参数描述** |
| seesionId | int | 会话标识 |
| volume | int | 会话开始后的实时语音音量。取值：0-100 |

**返回值说明：**

|  |  |  |
| --- | --- | --- |
| **返回值** | **类型** | **参数描述** |
| 无 | 无 | 无 |

**其他：**无

* **函数：onRecordStop**

**函数说明：**录音结束的通知。

**入参说明：**

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **参数描述** |
| seesionId | int | 会话标识 |

**返回值说明：**

|  |  |  |
| --- | --- | --- |
| **返回值** | **类型** | **参数描述** |
| 无 | 无 | 无 |

**其他：**无

* **函数：onPretreatedResult**

**函数说明：**语义解析结果的预处理。

**入参说明：**

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **参数描述** |
| seesionId | int | 会话标识 |
| data | Protocol | 语义解析结果的标准化数据。 |
| commandDomain | String | 指令域。 |
| command | String | 指令。 |
| commandParams | JSONObject | 指令参数。 |
| question | String | ASR结果文本。 |

**返回值说明：**

|  |  |  |
| --- | --- | --- |
| **返回值** | **类型** | **参数描述** |
| 无 | 无 | 无 |

**其他：**无

* **函数：onRecognizeResult**

**函数说明：**

**语义解析结果。**

**入参说明：**

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **参数描述** |
| seesionId | int | 会话标识 |
| data | ProtocolData | 语义解析结果的标准化数据。 |

**返回值说明：**

|  |  |  |
| --- | --- | --- |
| **返回值** | **类型** | **参数描述** |
| 无 | 无 | 无 |

**其他：**无

* **函数：onNotify**

**函数说明：**主动上报消息，非原会话流程中。

**入参说明：**

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **参数描述** |
| type | int | 主动上报类型：   1. NOTIFY\_TYPE\_ONLINE\_STATUS； 2. NOTIFY\_TYPE\_ACTIVE\_STATUS； 3. NOTIFY\_TYPE\_LANCONNECTOR\_STATUS； 4. NOTIFY\_THIRDAPP\_CONTEXT； 5. NOTIFY\_PUSH\_MESSAGE； 6. NOTIFY\_AUDIO\_STATUS； 7. NOTIFY\_TTS\_STATUS 8. NOTIFY\_BINDER\_USER； 9. NOTIFY\_QRCODE\_MESSAGE； |
| data | Object | 主动上报的数据：   1. 当NOTIFY\_TYPE\_ACTIVE\_STATUS时的激活信息的JSON字符串；取值：String； 2. 当NOTIFY\_THIRDAPP\_CONTEXT时的上下文的JSON字符串，取值：String； 3. 当NOTIFY\_PUSH\_MESSAGE时的语义解析结果的标准化数据，取值：ProtocolData； 4. 当NOTIFY\_BINDER\_USER时的用户数据，取值：UserData； 5. 当NOTIFY\_QRCODE\_MESSAGE时的二维码内容，取值，String： |
| arg1 | int | 预留参数1：   1. 当NOTIFY\_TYPE\_ONLINE\_STATUS时的在线状态，取值：1：在线；0：离线； 2. 当NOTIFY\_TYPE\_LANCONNECTOR\_STATUS时的详细类型。取值：1：请求绑定；2、连接；3、清屏；4、绑定成功；5、心跳；6、tts状态； 3. 当NOTIFY\_AUDIO\_STATUS时的播放状态。取值；0、播放准备；1、播放开始；2、播放暂停；3、继续播放；4、播放中断(会自动恢复)；5、中断后继续播放；6、播放完成；7、播放中断(不会自动恢复)；8、播放停止；9、播放失败；10、未知 4. 当NOTIFY\_TTS\_STATUS时的tts播放状态。取值：1、播放开始；6、播放完成 5. 当NOTIFY\_BINDER\_USER时的用户绑定状态。取值：1、成功；0、失败； |
| arg2 | int | 预留参数2：   1. 当NOTIFY \_AUDIO\_STATUS时，取值：播放的ID； |

**返回值说明：**

|  |  |  |
| --- | --- | --- |
| **返回值** | **类型** | **参数描述** |
| 无 | 无 | 无 |

**其他：**无

* **函数：hideUi**

**函数说明：**隐藏UI层界面。

**入参说明：**

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **参数描述** |
| 无 | 无 | 无 |

**返回值说明：**

|  |  |  |
| --- | --- | --- |
| **返回值** | **类型** | **参数描述** |
| 无 | 无 | 无 |

**其他：**无

* **函数：isUiShowing**

**函数说明：**UI层界面是否正在显示中。

**入参说明：**

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **参数描述** |
| 无 | 无 | 无 |

**返回值说明：**

|  |  |  |
| --- | --- | --- |
| **返回值** | **类型** | **参数描述** |
| boolean | boolean | UI界面是否正在显示中。 |

**其他：**无

* **函数：onLowMemory**

**函数说明：**

**释放UI层的内存。**

**入参说明：**

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **参数描述** |
| 无 | 无 | 无 |

**返回值说明：**

|  |  |  |
| --- | --- | --- |
| **返回值** | **类型** | **参数描述** |
| 无 | 无 | 无 |

**其他：**无

## 业务数据

### 类：ProtocolData

**类说明：**AILabs语义解析结果的标准化数据结构。

**成员变量说明：**

|  |  |  |
| --- | --- | --- |
| **变量名称** | **类型** | **变量描述** |
| code | int/DataCode | 结果码，默认为DataCode.CODE\_SUCCESS。 |
| type | int/DataType | 界面类型，默认为DataType.UNKNOWN.。 |
| question | String | 语义解析的ASR文本内容 |
| domain | String | 领域 |
| intent | String | 意图 |
| spokenText | String | 文本内容：tts播报的文本。 |
| writtenText | String | 文本内容：界面显示的文本。 |
| data | Object | 标准化业务数据： |
| extraData | Object | 标准化业务扩展数据 |
| recordTime | long | 录音时长（单位：ms） |
| hasCommand | boolean | 是否有指令 |
| openFarMic | boolean | 是否在会话完后自动再次开麦 |
| needTts | boolean | 是否需要播放tts |
| voiceId | String | 声纹ID |

**函数说明：**

|  |  |
| --- | --- |
| **函数名称** | **功能描述** |
| onRecordStart | 录音开始（会话开始）的通知。 |
| onStreaming | 会话开始后的实时ASR结果文本 |
| onVolume | 会话开始后的实时语音音量 |
| onRecordStop | 录音结束的通知。 |
| onPretreatedResult | 语义解析结果的预处理 |
| onRecognizeResult | 语义解析结果 |
| onNotify | 主动上报（非会话流程中） |
| ~~initMainDialog~~ | ~~初始化界面（废弃）~~ |
| hideMainDialog | 隐藏界面 |
| isMainDialogShowing | 界面是否显示中 |
| onLowMemory | 释放内存 |

**其他：**所有接口的回调均发生在主线程。

### 类：ProtocolData.DataType

**类说明：**AILabs语义解析结果的标准化数据的界面数据类型。

**成员变量说明：**

|  |  |  |
| --- | --- | --- |
| **变量名称** | **类型** | **变量描述** |
| UNKNOWN | int(0) | 取值0；标准化数据类型：未知 |
| ~~FULL\_SEARCH~~ | ~~int(1)~~ | 取值1；~~标准化数据类型：搜索结果（废弃）~~ |
| ~~FULL\_SEARCH\_RECOMMAND~~ | ~~int(2)~~ | 取值2；~~标准化数据类型：搜索推荐结果（废弃）~~ |
| ~~WEATHER~~ | ~~int(3)~~ | 取值3；~~标准化数据类型：天气类型（废弃）~~ |
| FEEDBACK | int(4) | 取值4；标准化数据类型：反馈类型 |
| ~~CHAT~~ | ~~int(5)~~ | 取值5；~~标准化数据类型：聊天领域（废弃）~~ |
| ~~TEXT~~ | ~~int(6)~~ | 取值6；~~标准化数据类型：文本类型（废弃）~~ |
| ~~MATH~~ | ~~int(7)~~ | 取值7；~~标准化数据类型：算术领域（废弃）~~ |
| ~~CALENDAR~~ | ~~int(8)~~ | 取值8；~~标准化数据类型：日期领域（废弃）~~ |
| Video\_History | int(9) | 取值9；标准化数据类型：最近观看历史 |
| Video\_Favor | int(10) | 取值10；标准化数据类型：追剧和收藏 |
| Video\_Selection | int(11) | 取值11；标准化数据类型：选集 |
| Video\_Definition | int(12) | 取值12；标准化数据类型：清晰度 |
| Video\_Ratio | int(13) | 取值13；标准化数据类型：画面比例 |
| AI\_TEXT | int(14) | 取值14；标准化数据类型：AILabs文本类型 |
| ~~AI\_FILM\_NEWS~~ | ~~int(15)~~ | 取值15；~~标准化数据类型：影讯类型（废弃）~~ |
| ~~AI\_MEMO~~ | ~~int(16)~~ | 取值16；~~标准化数据类型：提醒类型（废弃）~~ |
| ~~AI\_TAKEOUT\_DISH\_INFO~~ | ~~int(17)~~ | 取值17；~~标准化数据类型：外卖商品（废弃）~~ |
| ~~AI\_TAKEOUT\_ORDER\_INFO~~ | ~~int(18)~~ | 取值18；~~标准化数据类型：订单详情（废弃）~~ |
| ~~AI\_TAKEOUT\_PAY~~ | ~~int(19)~~ | 取值19；~~标准化数据类型：声纹支付（废弃）~~ |
| ~~AI\_TAKEOUT\_PAY\_STATUS~~ | ~~int(20)~~ | 取值20；~~标准化数据类型：声纹支付结果（废弃）~~ |
| ~~X!\_RECOMMAND~~ | ~~int(21)~~ | 取值21；~~标准化数据类型：X1日常推荐（废弃）~~ |
| APP\_History | int(22) | 取值22；标准化数据类型：我的应用列表 |
| MUSIC | int(23) | 取值23；标准化数据类型：音乐类型 |
| AI\_VIDEO\_FULLSEARCH | int(24) | 取值24；标准化数据类型：AILabs影视搜索结果 |
| AI\_WEATHER | int(26) | 取值26；标准化数据类型：AILabs天气 |
| AI\_APP | int(27) | 取值27；标准化数据类型：AILabs应用搜索结果 |
| AI\_ALARM | int(28) | 取值28；标准化数据类型：AILabs闹钟 |
| AI\_ENCYCLOPEDIA | int(29) | 取值29；标准化数据类型：AILabs百科 |

**函数说明：无**

**其他：**无

### 类：ProtocolData.DataCode

**类说明：**AILabs语义解析结果的标准化数据的结果码。

**成员变量说明：**

|  |  |  |
| --- | --- | --- |
| **变量名称** | **类型** | **变量描述** |
| CODE\_SUCCESS | int(0) | 取值0；结果码：成功 |
| CODE\_UNKNOWN | int(1) | 取值1；结果码：位置错误 |
| CODE\_RECORD\_ERROR | int(2) | 取值2；结果码：录音失败 |
| CODE\_NLP\_ERROR | int(3) | 取值3；结果码：语义解析请求失败 |
| CODE\_NO\_QUESTION | int(4) | 取值4；结果码：录音时间过短，未识别出ASR文本。 |
| CODE\_CUSTOM\_MIN | int(1000) | 取值1000；结果码：最小的自定义结果码。 |

**函数说明：无**

**其他：**无

# 媒体播放控制接口

## 安卓Mediaplayer封装

### 类：MediaOutputBridge

**类说明：该类主要用于处理各类音频资源播放**，如tts，流媒体音乐，闹铃，系统提示音等。以及提供播放状态的查询，并向底层同步的功能。

**函数说明：**

|  |  |  |
| --- | --- | --- |
| **函数名称** | **参数** | **功能描述** |
| startOneshotAudioPlaying | String path  音频资源的路径 | **开始闹钟等一次性音频资源的播放。** |
| stopOneshotAudioPlaying | 无 | **停止闹钟等一次性音频资源的播放** |
| startAudioPlaying | String path  音频资源的路径 | **开始音乐等流媒体音频资源的播放。** |
| seekAudioPlaying | int msec,  boolean relative | **对流媒体进行seek操作。** |
| pauseAudioPlaying | 无 | **暂停流媒体的播放** |
| resumeAudioPlaying | 无 | **恢复流媒体的播放** |
| stopAudioPlaying | 无 | **停止流媒体的播放** |
| startTtsPlaying | String path  音频资源的路径 | **开始tts音频资源的播放。** |
| stopTtsPlaying | 无 | **停止tts音频资源的播放** |
| clean | 无 | **停止所有音频资源的播放，并清空队列** |
| composeGenieTts | String text | **利用天猫精灵云端TTS合成引擎，将文字转成TTS音频流** |

### 类：IMediaOutputListener

**类说明：该类主要用于**提供播放各种媒体资源状态的回调。

|  |  |  |
| --- | --- | --- |
| **函数名称** | **参数** | **功能描述** |
| onAudioPlayStart | 无 | **流媒体开始播放的回调** |
| onAudioPlayStop | 无 | **流媒体停止播放的回调** |
| onAudioPlayComplete | 无 | **流媒体播放完成的回调** |
| onAudioPlayPause | 无 | **流媒体暂停播放的回调** |
| onAudioPlayResume | 无 | **流媒体恢复播放的回调** |
| onAudioPlaySuspend | 无 | **流媒体挂起的回调** |
| onAudioPlaySuspend2Resume | 无 | **流流媒体挂起后恢复的回调** |
| onAudioSeek | **int** progress | **流媒体seek的回调** |
| onOneshotAudioPlayStart | 无 | **一次性音频开始播放的回调** |
| onOneshotAudioPlayStop | 无 | **一次性音频停止播放的回调** |
| onTtsStart | 无 | **tts音频开始播放的回调** |
| onTtsStop | 无 | **tts音频停止播放的回调** |

## 影视端端方案

服务端根据biztype和bizgroup来判断是否要返回优酷媒资内容

# 第三方应用指令接入

AliGenie支持第三方应用中注册到语音应用中，语音应用将AI能力透到第三方应用中，并制定一些处理策略。详见文档《阿里巴巴AILABS第三方应用语音SDK使用手册》。

手册如下：



# 上下文信息

## 上下文监听器

### 类：TvContextListener

**类说明：**该类获取应用上下文信息(包含系统信息、当前场景信息、包名信息等)，传递给SDK底层，用来请求后台数据。

**成员变量说明：**

|  |  |  |
| --- | --- | --- |
| **变量名称** | **类型** | **变量描述** |
| 无 | 无 | 无 |

**函数说明：**

|  |  |
| --- | --- |
| **函数名称** | **功能描述** |
| getTvContext | 用来获取应用上下文信息。 |

**其他：**所有接口的回调均发生在主线程。

* **函数：getTvContext**

**函数说明：**获取应用上下文信息。

**入参说明：**

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **参数描述** |
| 无 | 无 | 无 |

**返回值说明：**

|  |  |  |
| --- | --- | --- |
| **返回值** | **类型** | **参数描述** |
| String | String | 应用上下文信息（内部为JSON的String结构），包含：   1. sceneExtInfo：场景上下文信息；  * pageType：唤醒时后台页面类型，类型PageType； * pageData：唤醒时后台页面的上下文数据，类型String； * package：唤醒时后台页面的应用包名，类型String； * soundType：前台语音页面类型，类型SoundType； * soundData：前台语音页面的上下文数据； * context：上下文信息集合。  1. protocolVersion：支持的协议版本号信息。类型int，目前取值2。 |

## 业务数据

### 类：PageType

**类说明：**该类代表唤醒时后台页面类型。

**成员变量说明：**

|  |  |  |
| --- | --- | --- |
| **变量名称** | **类型** | **变量描述** |
| PAGE\_TYPE\_DEFAULT | int | 取值0；全局默认； |
| PAGE\_TYPE\_YINGSHI\_DEFAULT | int | 取值1；影视默认； |
| PAGE\_TYPE\_MOVIE\_DETAIL\_PLAY | int | 取值2；电影详情页； |
| PAGE\_TYPE\_MOVE\_FULL\_PLAY | int | 取值3；电影全屏页； |
| PAGE\_TYPE\_SERIES\_DETAIL\_PLAY | int | 取值4；电视剧详情页； |
| PAGE\_TYPE\_ SERIES\_FULL\_PLAY | int | 取值5；电视剧全屏页； |
| PAGE\_TYPE\_VERIETY\_SHOW\_DETAIL\_PLAY | int | 取值6；综艺详情页； |
| PAGE\_TYPE\_VERIETY\_SHOW\_FULL\_PLAY | int | 取值7；综艺全屏页； |

**函数说明： 无。**

**其他：**无。

### 类：SoundType

**类说明：**该类代表唤醒时前台语音页面类型。

**成员变量说明：**

|  |  |  |
| --- | --- | --- |
| **变量名称** | **类型** | **变量描述** |
| SOUND\_TYPE\_DEFAULT | int | 取值0；默认，无显示； |
| SOUND\_TYPE\_ONLY\_TITLE | int | 取值1；仅语音条显示； |
| SOUND\_TYPE\_RESULT\_COMMAND | int | 取值2；全屏推荐页； |
| SOUND\_TYPE\_RESULT\_YINGSHI | int | 取值3；影视搜索结果页； |
| SOUND\_TYPE\_ RESULT\_HISTORY | int | 取值4；观看历史结果页； |
| SOUND\_TYPE\_ RESULT\_FAVOR | int | 取值5；我的收藏结果页； |
| SOUND\_TYPE\_ RESULT\_SELECTION | int | 取值6；选集结果页； |
| SOUND\_TYPE\_ RESULT\_DEFINITION | int | 取值7；清晰度结果页； |
| SOUND\_TYPE\_ RESULT\_RATIO | int | 取值8；画面比例结果页； |
| SOUND\_TYPE\_ RESULT\_WEATHER | int | 取值9；天气结果页； |
| SOUND\_TYPE\_ RESULT\_FILM\_NEWS | int | 取值10；影讯页； |
| SOUND\_TYPE\_ RESULT\_MEMO | int | 取值11；日程提醒页； |
| SOUND\_TYPE\_ RESULT\_TAKEOUT\_DISH | int | 取值12；外卖菜品页； |
| SOUND\_TYPE\_ RESULT\_TAKEOUT\_ORDER | int | 取值13；外卖订单页； |
| SOUND\_TYPE\_ RESULT\_TAKEOUT\_PAY | int | 取值14；外卖支付页； |
| SOUND\_TYPE\_ RESULT\_TAKEOUT\_PAY\_RESULT | int | 取值15；外卖支付结果页； |

**函数说明： 无。**

**其他：**无。

# 账号绑定（针对带屏设备）

**天猫精灵开放平台上提供的部分技能是依赖淘宝账号的，如：家居，购物等。因此需要在使用这些技能时登录天猫精灵APP，然后扫码绑定设备，SDK已经提供了完整的二维码页面，厂商只需要按如下步骤进行配置即可**

实现IUiManager

## 实现IUiManager中的onNotify方法

onNotify方法是SDK通知厂商的通道，代码如下：

@Override  
 **public void** onNotify(**int** type, Object data, **int** arg1, **int** arg2) {  
 **switch** (type) {  
**case** IBaseView.***NOTIFY\_QRCODE\_MESSAGE***: {  
 *//技能需要显示二维码时会收到该通知* **if** (data != **null**) {  
 **try** {  
 Intent intent = **new** Intent();  
 intent.setAction(***SHOW\_QRCODE***);  
 intent.setPackage(getPackageName());  
 intent.addFlags(Intent.***FLAG\_ACTIVITY\_NEW\_TASK***);  
 intent.putExtra(***QRCODE\_KEY***, data.toString());  
 startActivity(intent);  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 hideUi();  
 }  
 }  
 **break**;  
 **case** IBaseView.***NOTIFY\_BINDER\_USER***: {  
 *//绑定设备后会收到该通知* **if** (*bindDeviceListener* != **null** && *bindDeviceListener*.get() != **null**) {  
 BindDeviceListener listener = *bindDeviceListener*.get();  
 listener.bindResponse(0, arg1 == 0 ? **false** : **true**, (UserData) data);  
 }  
 }  
 **break**;  
 }  
 }

实现BindDeviceGuideActivity

## 将demo中的BindDeviceGuideActivity集成到自己的APK中

BindDeviceGuideActivity二维码显示的页面，已包含在demo中，源码开发，厂商可以自行修改风格，布局，配色等。注意不要修改二维码生成相关的回调函数

## 将demo中的AndroidManifest集成到自己的APK中

<**activity  
 android:name="com.yunos.tv.alitvasr.account.BindDeviceGuideActivity"  
 android:launchMode="singleTop"**>  
 <**intent-filter**>  
 <**action android:name="com.yunos.tv.altvasr.showQrCode"** />  
 <**action android:name="com.yunos.tv.altvasr.bindResponse"** />  
 <**action android:name="com.yunos.tv.altvasr.bindUser"** />  
  
 <**category android:name="android.intent.category.DEFAULT"** />  
 </**intent-filter**>  
  
 <**intent-filter**>  
 <**action android:name="android.intent.action.VIEW"** />  
 <**category android:name="android.intent.category.DEFAULT"** />  
 <**data  
 android:host="user\_bind"  
 android:scheme="alitv\_asr"** />  
 </**intent-filter**>  
</**activity**>

# 附录

## demo仓库地址

git clone git@gitlab.alibaba-inc.com:miyang.my/geniesdk\_demo.git

## 常用技能ProtocolData数据示例

* 技能：天气
* 示例：今天的天气

|  |  |  |
| --- | --- | --- |
| **变量名称** | **类型** | **变量取值** |
| code | int/DataCode | DataCode.CODE\_SUCCESS。 |
| type | int/DataType | DataType. AI\_WEATHER |
| question | String | null |
| domain | String | null |
| intent | String | null |
| spokenText | String | 今天上海市多云，气温27度到33度 ，天气炎热，适宜穿短袖衣裤等夏季服装，空气质量优，湿度 60，不适宜户外运动。 |
| writtenText | String | 今天上海市多云，气温27度到33度 ，天气炎热，适宜穿短袖衣裤等夏季服装，空气质量优，湿度 60，不适宜户外运动。 |
| data | WeatherData | WeatherData {   weatherDetailList = [WeatherDetail {    datetimestamp = '1533188700000', area = '上海市', pm25 = '27', level\_desc = '优', curTemp = '32', maxTemp = '33', minTemp = '27', weather\_code = '100', weather\_name = '多云', humidity = '60', wind\_desc = '西南风', wind\_power\_desc = '一级'   }, WeatherDetail {    datetimestamp = '1533225600000', area = '上海市', pm25 = 'null', level\_desc = 'null', curTemp = '', maxTemp = '31', minTemp = '28', weather\_code = '203', weather\_name = '中雨', humidity = 'null', wind\_desc = '东北风', wind\_power\_desc = '五级到六级'   }, WeatherDetail {    datetimestamp = '1533312000000', area = '上海市', pm25 = 'null', level\_desc = 'null', curTemp = '', maxTemp = '35', minTemp = '28', weather\_code = '100', weather\_name = '多云', humidity = 'null', wind\_desc = '东南风', wind\_power\_desc = '三级到四级'   }, WeatherDetail {    datetimestamp = '1533398400000', area = '上海市', pm25 = 'null', level\_desc = 'null', curTemp = '', maxTemp = '34', minTemp = '28', weather\_code = '100', weather\_name = '多云', humidity = 'null', wind\_desc = '东南风', wind\_power\_desc = '三级'   }, WeatherDetail {    datetimestamp = '1533484800000', area = '上海市', pm25 = 'null', level\_desc = 'null', curTemp = '', maxTemp = '35', minTemp = '28', weather\_code = '201', weather\_name = '小雨', humidity = 'null', wind\_desc = '东南风', wind\_power\_desc = '三级'   }, WeatherDetail {    datetimestamp = '1533571200000', area = '上海市', pm25 = 'null', level\_desc = 'null', curTemp = '', maxTemp = '36', minTemp = '28', weather\_code = '201', weather\_name = '小雨', humidity = 'null', wind\_desc = '东南风', wind\_power\_desc = '三级'   }, WeatherDetail {    datetimestamp = '1533657600000', area = '上海市', pm25 = 'null', level\_desc = 'null', curTemp = '', maxTemp = '34', minTemp = '26', weather\_code = '201', weather\_name = '小雨', humidity = 'null', wind\_desc = '东南风', wind\_power\_desc = '二级'   }, WeatherDetail {    datetimestamp = '1533744000000', area = '上海市', pm25 = 'null', level\_desc = 'null', curTemp = '', maxTemp = '35', minTemp = '26', weather\_code = '201', weather\_name = '小雨', humidity = 'null', wind\_desc = '东风', wind\_power\_desc = '二级'   }, WeatherDetail {    datetimestamp = '1533830400000', area = '上海市', pm25 = 'null', level\_desc = 'null', curTemp = '', maxTemp = '35', minTemp = '27', weather\_code = '201', weather\_name = '小雨', humidity = 'null', wind\_desc = '东北风', wind\_power\_desc = '三级'   }, WeatherDetail {    datetimestamp = '1533916800000', area = '上海市', pm25 = 'null', level\_desc = 'null', curTemp = '', maxTemp = '34', minTemp = '26', weather\_code = '201', weather\_name = '小雨', humidity = 'null', wind\_desc = '东北风', wind\_power\_desc = '二级'   }, WeatherDetail {    datetimestamp = '1534003200000', area = '上海市', pm25 = 'null', level\_desc = 'null', curTemp = '', maxTemp = '36', minTemp = '28', weather\_code = '000', weather\_name = '晴', humidity = 'null', wind\_desc = '东南风', wind\_power\_desc = '三级'   }, WeatherDetail {    datetimestamp = '1534089600000', area = '上海市', pm25 = 'null', level\_desc = 'null', curTemp = '', maxTemp = '37', minTemp = '30', weather\_code = '201', weather\_name = '小雨', humidity = 'null', wind\_desc = '东南风', wind\_power\_desc = '三级'   }, WeatherDetail {    datetimestamp = '1534176000000', area = '上海市', pm25 = 'null', level\_desc = 'null', curTemp = '', maxTemp = '36', minTemp = '30', weather\_code = '201', weather\_name = '小雨', humidity = 'null', wind\_desc = '东南风', wind\_power\_desc = '三级'   }, WeatherDetail {    datetimestamp = '1534262400000', area = '上海市', pm25 = 'null', level\_desc = 'null', curTemp = '', maxTemp = '36', minTemp = '28', weather\_code = '201', weather\_name = '小雨', humidity = 'null', wind\_desc = '东南风', wind\_power\_desc = '四级'   }, WeatherDetail {    datetimestamp = '1534348800000', area = '上海市', pm25 = 'null', level\_desc = 'null', curTemp = '', maxTemp = '30', minTemp = '30', weather\_code = '000', weather\_name = '晴', humidity = 'null', wind\_desc = '东南风', wind\_power\_desc = '四级'   }]  } |
| extraData | ProtocolExtra | {, nlgResultJSON = , nluResultJSON = , dmResultJson = , toUrls = []  } |
| recordTime | long | 0 |
| hasCommand | boolean | false |
| openFarMic | boolean | false |
| needTts | boolean | false |
| voiceId | String | null |

* 技能：音乐
* 示例：我要听歌

|  |  |  |
| --- | --- | --- |
| **变量名称** | **类型** | **变量取值** |
| code | int/DataCode | DataCode.CODE\_SUCCESS。 |
| type | int/DataType | DataType.MUSIC |
| question | String | null |
| domain | String | null |
| intent | String | null |
| spokenText | String | 一起欣赏：刘惜君的遥远的歌。 |
| writtenText | String | 一起欣赏：刘惜君的遥远的歌。 |
| data | MusicList | MusicList {  playIndex = 1, currentItem = MusicItem {  audioId = '1797262138', audioName = '遥远的歌', audioAnchor = '刘惜君', audioAlbum = '你好,旧时光 电视原声带', audioUrl = 'http://m128.xiami.net/162/29162/2102904991/1797262138\_1509680320541.mp3?auth\_key=1533870000-0-0-3defa14dcb3305c832309c3c5ac385d7', audioImageUrl = 'http://pic.xiami.net/images/album/img62/34/5a12941783b89\_1713362\_1511166999.jpg', audioSource = 'xiami', audioFormat = 'mp3', audioExt = '{"sound\_delta":"0","playlist\_request\_id":"0bb9f0fd15332800252906931d30f5","audioType":"music","audioLength":"174","rec\_from":"AILAB\_REC","request\_id":"0bb9f0fd15332800252906931d30f5"}', audioLength = 0, progress = 0, position = 1  }, displayItemList = [MusicItem {  audioId = '1797262138', audioName = '遥远的歌', audioAnchor = '刘惜君', audioAlbum = '你好,旧时光 电视原声带', audioUrl = 'http://m128.xiami.net/162/29162/2102904991/1797262138\_1509680320541.mp3?auth\_key=1533870000-0-0-3defa14dcb3305c832309c3c5ac385d7', audioImageUrl = 'http://pic.xiami.net/images/album/img62/34/5a12941783b89\_1713362\_1511166999.jpg', audioSource = 'xiami', audioFormat = 'mp3', audioExt = '{"sound\_delta":"0","playlist\_request\_id":"0bb9f0fd15332800252906931d30f5","audioType":"music","audioLength":"174","rec\_from":"AILAB\_REC","request\_id":"0bb9f0fd15332800252906931d30f5"}', audioLength = 0, progress = 0, position = 1  }], pageNum = 1, pageSize = 5  } |
| extraData | ProtocolExtra | {, nlgResultJSON = , nluResultJSON = , dmResultJson = , toUrls = []  } |
| recordTime | long | 0 |
| hasCommand | boolean | false |
| openFarMic | boolean | false |
| needTts | boolean | false |
| voiceId | String | null |

* 技能：百科
* 示例：马云是谁

|  |  |  |
| --- | --- | --- |
| **变量名称** | **类型** | **变量取值** |
| code | int/DataCode | DataCode.CODE\_SUCCESS。 |
| type | int/DataType | DataType. AI\_ENCYCLOPEDIA |
| question | String | null |
| domain | String | null |
| intent | String | null |
| spokenText | String | 马云，1964年9月10日生于浙江省杭州市， 1999年创办阿里巴巴，并与2014年于纽约正式挂牌上市，现担任阿里巴巴集团董事局主席。 |
| writtenText | String | 我已为您找到如下答案 |
| data | AIEncyclopediaData | AIEncyclopediaData {  type = 1, response = '马云，1964年9月10日生于浙江省杭州市， 1999年创办阿里巴巴，并与2014年于纽约正式挂牌上市，现担任阿里巴巴集团董事局主席。'  } |
| extraData | ProtocolExtra | {, nlgResultJSON = , nluResultJSON = , dmResultJson = , toUrls = []  } |
| recordTime | long | 0 |
| hasCommand | boolean | false |
| openFarMic | boolean | false |
| needTts | boolean | false |
| voiceId | String | null |

* 技能：提醒
* 示例：提醒我一分钟之后喝水

|  |  |  |
| --- | --- | --- |
| **变量名称** | **类型** | **变量取值** |
| code | int/DataCode | DataCode.CODE\_SUCCESS。 |
| type | int/DataType | DataType. AI\_TEXT |
| question | String | null |
| domain | String | null |
| intent | String | null |
| spokenText | String | 好，你去忙吧，我会在一分钟后准时提醒你喝水 |
| writtenText | String | 好，你去忙吧，我会在一分钟后准时提醒你喝水 |
| data | String | 好，你去忙吧，我会在一分钟后准时提醒你喝水 |
| extraData | ProtocolExtra | {, nlgResultJSON = , nluResultJSON = , dmResultJson = , toUrls = []  } |
| recordTime | long | 0 |
| hasCommand | boolean | false |
| openFarMic | boolean | false |
| needTts | boolean | false |
| voiceId | String | null |

* 技能：闹钟
* 示例：我要订闹铃

|  |  |  |
| --- | --- | --- |
| **变量名称** | **类型** | **变量取值** |
| code | int/DataCode | DataCode.CODE\_SUCCESS。 |
| type | int/DataType | DataType. AI\_ALARM |
| question | String | null |
| domain | String | null |
| intent | String | null |
| spokenText | String | 好的，主人要设定几点的闹钟呢？ |
| writtenText | String | 好的，主人要设定几点的闹钟呢？ |
| data | Object | null |
| extraData | ProtocolExtra | {, nlgResultJSON = , nluResultJSON = , dmResultJson = , toUrls = []  } |
| recordTime | long | 0 |
| hasCommand | boolean | false |
| openFarMic | boolean | false |
| needTts | boolean | false |
| voiceId | String | null |

## 设备控制预处理命令onPretreatedResult回调数据

* 技能：打开设置

|  |  |  |
| --- | --- | --- |
| **变量名称** | **类型** | **变量取值** |
| commandDomain | String | AliGenie.System.Control。 |
| command | String | OpenSetting |
| commandParams | Json | null |

* 技能：打开网络设置

|  |  |  |
| --- | --- | --- |
| **变量名称** | **类型** | **变量取值** |
| commandDomain | String | AliGenie.System.Control。 |
| command | String | SetNetwork |
| commandParams | Json | null |

* 技能：返回主页/返回桌面

|  |  |  |
| --- | --- | --- |
| **变量名称** | **类型** | **变量取值** |
| commandDomain | String | AliGenie.System.Control。 |
| command | String | GoToDesktop |
| commandParams | Json | null |

* 技能：返回

|  |  |  |
| --- | --- | --- |
| **变量名称** | **类型** | **变量取值** |
| commandDomain | String | AliGenie.System.Control。 |
| command | String | Back |
| commandParams | Json | null |

* 技能：退出

|  |  |  |
| --- | --- | --- |
| **变量名称** | **类型** | **变量取值** |
| commandDomain | String | AliGenie.System.Control。 |
| command | String | Exit |
| commandParams | Json | null |

* 技能：音量调节

|  |  |  |
| --- | --- | --- |
| **变量名称** | **类型** | **变量取值** |
| commandDomain | String | AliGenie.System.Control。 |
| command | String | 音量大一点：AdjustUpVolume  音量小一点：AdjustDownVolume  音量调节到：Volume |
| commandParams | Json | {"volumeStep":"10"}或者  {"volumeValue":"50"} |

* 技能：暂停

|  |  |  |
| --- | --- | --- |
| **变量名称** | **类型** | **变量取值** |
| commandDomain | String | AliGenie.System.Control。 |
| command | String | Pause |
| commandParams | Json | null |

* 技能：继续/继续播放

|  |  |  |
| --- | --- | --- |
| **变量名称** | **类型** | **变量取值** |
| commandDomain | String | AliGenie.System.Control。 |
| command | String | Play |
| commandParams | Json | null |

* 技能：快进/快退

|  |  |  |
| --- | --- | --- |
| **变量名称** | **类型** | **变量取值** |
| commandDomain | String | AliGenie.System.Control。 |
| command | String | FastForward/ FastBackward |
| commandParams | Json | null |

* 技能：快进x分钟/快退x分钟

|  |  |  |
| --- | --- | --- |
| **变量名称** | **类型** | **变量取值** |
| commandDomain | String | AliGenie.System.Control。 |
| command | String | FastForward/ FastBackward |
| commandParams | Json | {"time":"180"} 单位为s |

* 技能：上一页/下一页

|  |  |  |
| --- | --- | --- |
| **变量名称** | **类型** | **变量取值** |
| commandDomain | String | AliGenie.System.Control。 |
| command | String | NextPage/ PreviousPage |
| commandParams | Json |  |

* 技能：休眠/取消休眠

|  |  |  |
| --- | --- | --- |
| **变量名称** | **类型** | **变量取值** |
| commandDomain | String | AliGenie.System.Control。 |
| command | String | SetSleep/ CancelSleep |
| commandParams | Json |  |

* 技能：全屏/退出全屏

|  |  |  |
| --- | --- | --- |
| **变量名称** | **类型** | **变量取值** |
| commandDomain | String | AliGenie.System.Control。 |
| command | String | FullScreen/ ExitFullScreen |
| commandParams | Json |  |

## 应用控制预处理命令onPretreatedResult回调数据

* 技能：打开应用中心

|  |  |  |
| --- | --- | --- |
| **变量名称** | **类型** | **变量取值** |
| commandDomain | String | AliGenie.System.Control。 |
| command | String | OpenAppCenter |
| commandParams | Json |  |

* 技能：打开应用中心

|  |  |  |
| --- | --- | --- |
| **变量名称** | **类型** | **变量取值** |
| commandDomain | String | AliGenie.System.Control。 |
| command | String | OpenAppCenter |
| commandParams | Json |  |