

DUEROS 接入快速指南

(App 接入)



百度有屏音箱事业部

2019.06



目录

1、概述.....	3
1.1 什么是 DUER OS?.....	3
1.2 目标读者.....	3
2、关键背景知识.....	4
2.1 基本概念.....	4
2.2 技能交互模型.....	4
2.3 DUER OS 通讯协议.....	6
3、DuerOS App 接入技能概念架构.....	8
3.1 概念架构图及基本交互流程.....	8
4、DuerOS App 接入的基本步骤.....	10
4.1 登录 DBP 平台注册成为开发者.....	10
4.2 创建技能.....	10
4.3 切换技能类型到 Android App.....	11
4.4 App 端集成 BotSDK.....	11
4.5 接收用户意图.....	12
4.5 注册自定义用户交互.....	13
4.6 打开意图调用时序图.....	15
参考文档.....	16



1、概述

1.1 什么是 DUER OS?

DuerOS 是百度度秘事业部研发的对话式人工智能系统。搭载 DuerOS 的设备可让用户以自然语言对话的交互方式，通过最自然的语音方式与设备进行交互。

1.2 目标读者

本文档服务于以 App 方式接入 DuerOS 的第三方技能开发者。介绍了把一个 Android App 接入到 DuerOS 中的基本过程，实现对接入 App 的智能语音操控。



2、关键背景知识

在开始集成之前，有一些重要的关键背景知识需要开发者进行了解。这些背景知识有助于更顺利地实现 App 的接入。本文档只列出对于进行 App 接入必须的核心知识，更深入的细节可以进入 [DuerOS 开放平台](#)的[文档中心](#)进行了解。

2.1 基本概念

2.1.1 技能 (Bot)

指 DuerOS 为用户提供的对话式交互服务，如天气预报、百科、互动游戏...

2.1.2 技能开放平台 (DBP)

DuerOS Bots Platform，是 DuerOS 为第三方技能开发者提供技能开发、测试、部署工具的开放平台。开发者可以在 DBP 平台上开发自己的技能，提供自己想要支持的能力。

- DBP 平台地址: <https://developer.dueros.baidu.com/dbp>
- DBP 平台开发者[文档中心](#)
- DBP 平台技术课堂-[视频教程](#)

2.1.3 用户请求 (Query)

用户发起的语音请求，例如：“今天多少度”、“播放音乐”

2.2 技能交互模型

交互模型定义了用户如何与技能进行交互。用户的请求在交互模型中通过一个或多个意图来表示。

2.2.1 [意图 \(Intent\)](#)

意图指技能要满足的用户的请求或目的。意图是技能实现用户请求的基本单元，包括自定义意图和系统意图。一个技能可以包含一个或多个意图，可以创建自定义意图，也可以引用系统意图。

- **自定义意图**



第三方技能创建的意图是自定义意图。如查询天气信息的技能，当用户说“今天天气怎么样”、“今天多少度”、“今天有雨吗”等请求时，技能解析用户的意图是“天气查询”。

- **系统意图**

DBP 将一些通用的意图，如欢迎意图、取消意图、确认意图等做成了系统意图，提供给开发者，开发者可以直接引用。系统意图支持用户扩展常用表达。

2.2.2 常用表达/话术

常用表达指用户表达意图时具体的样例，用户表达样例越多，意图识别能力越强。如在天气查询的意图中，用户可能会通过如下方式询问天气，这些就是常用表达。

- 今天天气怎么样
- 今天北京的天气怎么样
- 告诉我今天的天气情况
- 帮我查一下今天北京的天气
- 北京今天的天气好吗
- 今天下雨吗

2.2.3 槽位 (Slots)

槽位是意图的可变参数信息。比如当用户询问“今天北京天气怎么样”时，其中“今天”和“北京”就是参数信息，这些信息被抽象成为槽位。在这个查询天气的意图中，可以根据“今天”抽象出日期 date 槽位，根据“北京”抽象出位置 location 槽位。槽位信息更准确的描述用户的意图。每个槽位都需要关联一个词典。

2.2.4 词典

词典指某领域词汇的集合。查询天气的意图中北京是城市信息，将北京、天津、上海等所有城市信息集合起来就组成了中国城市的词典。常见的词典有音乐歌曲词典、数字词典、动植物词典、名人词典等。DuerOS 根据词典去解析相应的槽位信息，发送给技能处理。

技能可以自己创建词典也可以引用[系统词典](#)。

2.2.5 意图数据结构

用户的语音请求经过 DuerOS 服务端的识别，如果是第三方技能在 DBP 平台上定义的意图，会转换为类似下面这样的数据结构：



```
"intent" : {  
  "name" : "{{STRING}}",  
  "slots" : {  
    "{{STRING}}" : {  
      "name" : "{{STRING}}",  
      "value" : "{{STRING}}",  
    }  
  }  
},
```

图 2.1 Intent 数据结构示例

- intent.name
意图名称。通过技能开放平台定义的意图，或来自系统意图。
- intent.slots
意图中的槽位。slot 结构是 key-value 结构，key 为 slot 名字，value 为 slot 槽位信息。
- intent.slots.name
槽位名称。
- intent.slots.value
槽位值。表示解析出的槽位值。

2.3 DUER OS 通讯协议

2.3.1 词汇

- DCS

DuerOS Conversational Service，面向智能设备的语音交互服务。

- DCS 协议

DCS 协议是 DuerOS 服务端与设备端之间的通讯协议，是一套把 DuerOS 的智能语音交互能力向所有设备开放的 API。

DCS 协议简介：[协议简介](#)

2.3.2 DCS 协议概述

DCS 协议由指令、事件、端状态三个部分组成。

- 指令 (directive)

服务端下发给设备端，设备端需要执行的操作。如播放语音 (Speak 指令)，设置闹钟



(SetAlert 指令) 等等。

- **事件 (event)**

设备端上报给服务端，通知服务端在设备端发生了需要服务端关注的事件。比如设备被唤醒并开始接收用户语音请求 (ListenStarted 事件)、音乐播放开始 (PlaybackStarted 事件) 等等。

- **端状态 (ClientContext)**

客户端上报 Event 时，需要带上客户端的状态。比如当前是否有音乐正在播放，摄像头是启用还是禁用等等。服务端在决定如何处理事件时会参考这些端状态。

2.3.3 客户端自定义交互 (UI Control)

描述基于当前页面、场景的可交互元素，以便进行语音交互的技术方案 (包含在 DCS 协议之中)。对于进行 App 接入并希望可以页面内容进行交互的开发者，需要特别关注。

- **UiContext**

UI Control 的端状态，描述当前页面的可交互元素。

- **ClickLink 指令**

UI Control 的自定义交互指令，描述用户通过语音对当前页面的可交互元素的操作。服务端会解析用户 query，结合 UiContext 得到意图，然后将对应的 URI 及参数下发到 App，App 根据 url 就知道用户发起了什么 query。

例子：当前页面有一个按钮【登录】，此时我们可以新增自定义的交互元素：

- 话术列表：“登录”，“点击登录”，...
- 话术的 URL：“botsdkdemo://login”

此描述信息会上报到服务端。之后用户说：“登录” 或 “点击登录”，App 就会收到一个 ClickLink 指令，其中包含 url: botsdkdemo://login。这时 App 就可以知道用户希望进行登录操作。



3、DuerOS App 接入技能概念架构

3.1 概念架构图及基本交互流程

下图描述了以 App 方式提供第三方技能的概念架构图。基本交互过程如下：

- ① 用户唤醒设备发起 query，DuerOS 上报事件给到服务端
- ② 服务端理解用户 query，生成对应意图，并通过指令下发到 DuerOS 设备端
- ③ 设备端再通过 BotSDK 分发指令到 App
- ④ DuerOS 通过 BotSDK 向 App 开放 DuerOS 的能力，App 端可以通过获取 BotSDK 实例调用 DuerOS 能力。

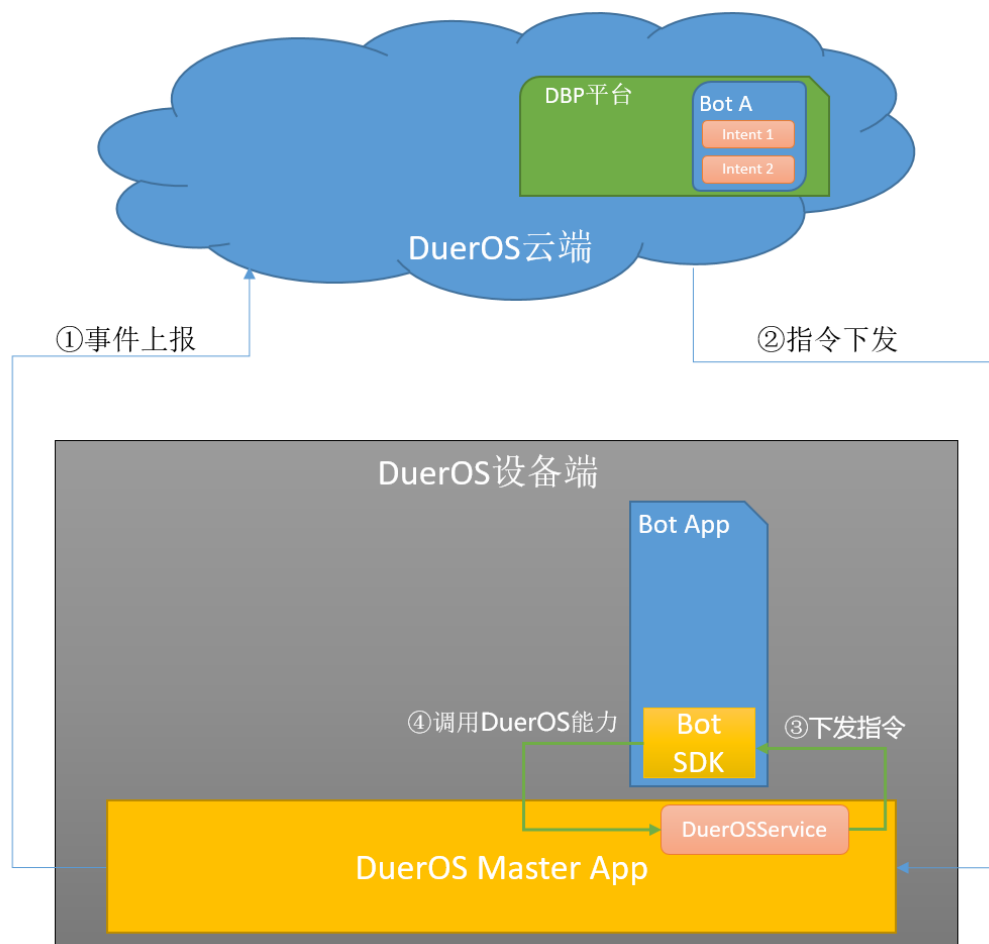


图 3.1 App 技能概念架构



对于开发者来说，上图中需要关心的有两部分：

- DBP 平台：DuerOS 服务端部分，在 DBP 上面开发 Bot
- BotSDK:App 客户端需要集成 BotSDK 以提供启动 App 以及进行语音交互的能力。



4、DuerOS App 接入的基本步骤

4.1 登录 DBP 平台注册成为开发者

首先要登录百度 [DuerOS 开放平台](#)，注册账号。登录成功后，请在屏幕右侧上方进行开发者认证（认证申请之后一般 48 小时内会开通）。



图 4.1 DuerOS 开发平台登录及开发者认证

4.2 创建技能

创建技能：[技能创建入门](#)

视频教程：百度开放平台-[技术课堂](#)

技能创建成功后，点击查看基础信息，要特别留意下图红色框中的属性，后续 App 端集成 BotSDK 的时候需要用到。

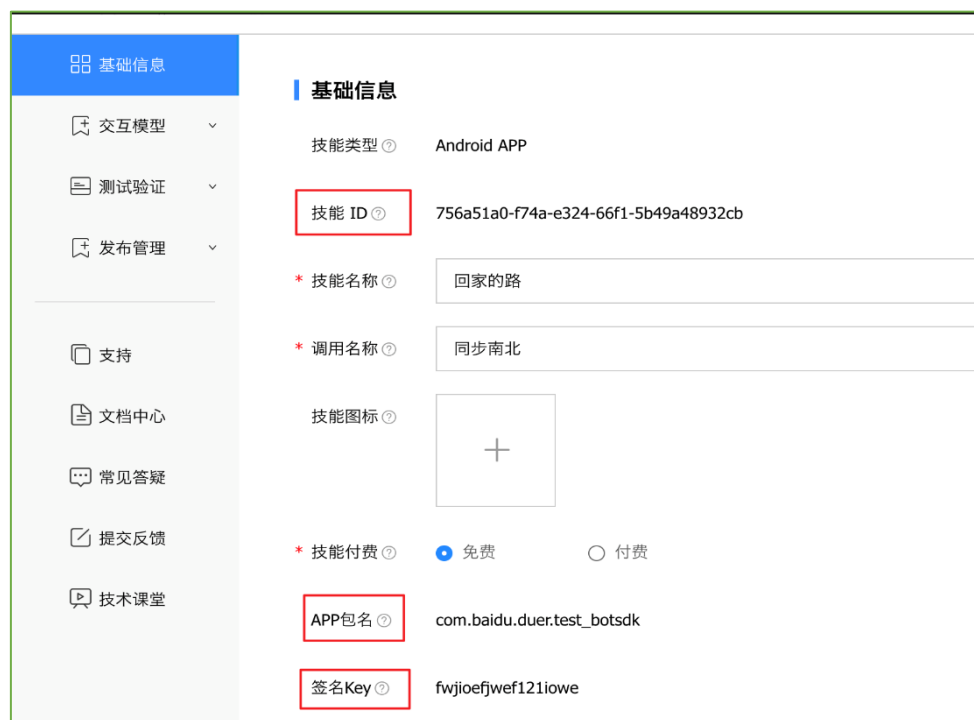


图 4.2 Bot 基本信息



- 技能 ID：技能唯一标识符
- App 包名：填写自己 App 的包名，包名错误会导致 App 调起失败
- 签名 key：用于 App 端向 BotSDK 注册自身信息以及访问 DuerOS 服务的时候进行校验，对长度和内容无限制。

4.3 切换技能类型到 Android App

DBP 平台上面 Android App 技能尚未对外开放，所以需要提交技能信息到百度方面接口人，以便接口人可以把自定义类型技能切换到 Android App 类型技能。

需要提交的就是上面步骤中的三个信息：技能 ID、App 包名、签名密钥（Sign key）。

技能类型修改成功之后，刷新页面就可以看到技能类型变成了 Android App。

4.4 App 端集成 BotSDK

如第三节“[DuerOS App 接入技能概念架构](#)”中所述，App 与 DuerOS 通信是依赖 BotSDK 实现的。App 端需要集成 BotSDK，以建立与 DuerOS 的连接。

4.4.1 集成 BotSDK

- 1) 下载 SDK 并放入到 app/libs 目录：[下载链接](#)
- 2) 在 app/build.gradle 中加入依赖

```
implementation(name:'botsdk-1.0.0.12',ext:'aar')
```

4.4.2 初始化 BotSDK

建议在 application 启动的时候进行：

```
BotSdk.getInstance().init(application);
```

4.4.3 注册 App

App 启动之后需要通过 BotSDK 向 Master App 进行注册。如果不能正确注册，将无法收到解析出来的指令。只有收到注册成功（图 5.1 中的 ResigerSucc）的回调才可以开始语音交互。

注册流程示例代码（注册时序图可以参考图 5.1）：



```
public void register(){
    if (!BotSdk.getInstance().isRegister()) {
        String rand1="hongyang"+Math.random();
        String rand2= "yanghong"+Math.random();
        String botId="3fcc17e3-7e97-b9ec-84cd-5211f6271394"; //来自DBP平台
        BotSdk.getInstance().register(
            messageListener,
            botId,
            rand1,
            sign(rand1),
            rand2,
            sign(rand2)
        );
    }
}

public String sign(String rand){
    String key = "abcdefghijk12345"; //来自DBP平台
    return getMD5(rand+key);
}
```

图 4.3 注册 App

参数介绍

- *messageListener*: App 端需要实现的回调接口，接收 BotSDK 消息。其中最重要的消息就是注册成功的消息，只有注册成功才可以进行语音交互
- *botId*: DBP 平台的技能 ID，如图 4.2 中所示
- *key*: DBP 平台的技能签名 key，如图 4.2 中所示。
- *rand1* 和 *rand2* 为[加密盐值](#)

4.5 接收用户意图

App 端通过上面注册的 Listener 回调方法收到 BotIntent 对象，通过 intent.name 获取意图名称，并通过 slots 获取对应槽位取值。

如下示例代码，App 端收到了叫 “light_on” 的意图，然后通过解析槽位值确定是要点亮灯的颜色。



下图为用户 query 时候的时序图

云端预定义的intent

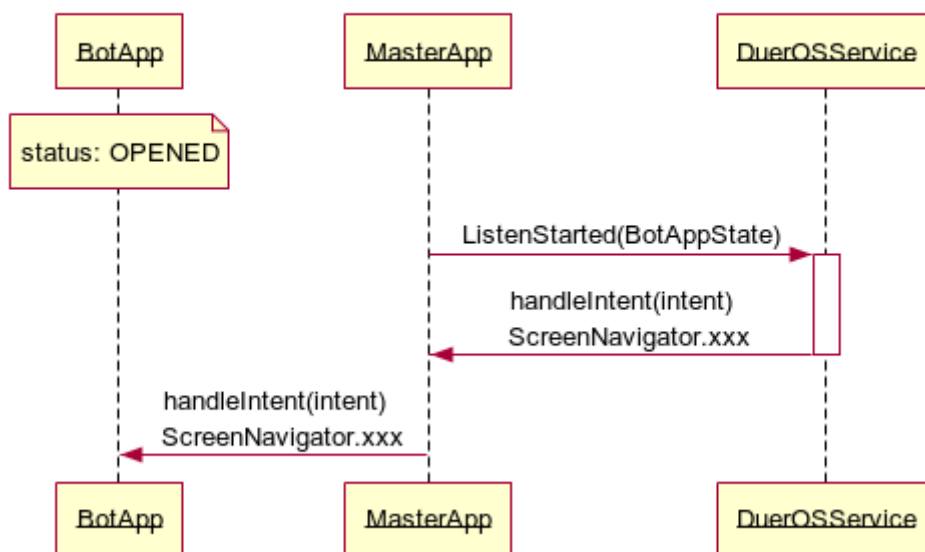


图 4.4 Query 时序图

- 1) 用户发起 query, Master App 上报 ListenStarted 事件
- 2) DuerOS 服务端解析用户 query 生成 Intent 下发给到 BotSDK
- 3) BotSDK 通过回调通知 App 端处理 Intent

4.5 注册自定义用户交互

如 2.3.3 节“客户端自定义交互”所述, App 增加当前页面或场景的交互元素描述, 不需要去服务端 Bot 开发意图, 就可以实现自定义用户交互。

4.5.1 简单自定义交互实例

如下图中,通过定义 query: “试一试” 和 url: “dskmemo://clicktest”的映射关系。App 增加了用户语音说“试一试”的支持。



```
private void updateUiContext() {
    UiContextPayload payload = new UiContextPayload();
    //String[] words = {"试一试", "点击试一试"};
    HashMap<String, String> params;
    params = new HashMap<>();
    params.put("name", "试一试");
    payload.addHyperUtterance( s: "sdkdemo://clicktest",
        list: null, s1: "link",
        params);

    BotSdk.getInstance().updateUiContext(payload);
}
```

如果此时用户发起语音 query：“试一试”，App 会收到 onClickLink 的回调，URL 就是我们前面定义的 sdkdemo://clicktest。我们可以把这个 URL 映射到某些 UI 组件的点击动作。

```
@Override
public void onClickLink(String url, HashMap<String, String> paramMap) {
    Log.d(TAG, msg: "onClickLink: " + url + " , " + paramMap);
    if("sdkdemo://clicktest".equals(url)){
        findViewById(R.id.test).performClick();
    }

    Toast.makeText( context: MainActivity.this, text: "url = " + url, Toast.LENGTH_SHORT).show();
}
```

4.5.2 使用 DuerOS 内建交互

除了上述简单的自定义交互，DuerOS 提供了一些内建交互。使用这些内建交互，不需要自己定义用户 query 话术，只引用对应类型即可。DuerOS 会处理这些类型和用户 query 话术的映射。

例如 input 类型，支持“输入 XX”，select 类型支持“选择 XX”，“打开 XX”。通过使用内建类型，极大简化自定义交互工作。DuerOS 支持的内建类型可以参考文档：[自定义交互支持的 type 列表](#)

如下图,引用系统 input 交互类型,参数类型是 city,实现了一个输入城市名称的 demo。对应的 url 是自定的 sdkdemo://input（输入城市名称会接在后面参数里）。

```
private void updateUiContext() {
    UiContextPayload payload = new UiContextPayload();
    //String[] words = {"试一试", "点击试一试"};
    HashMap<String, String> params;
    params = new HashMap<>();
    params.put("name", "地址");
    params.put("type", "city");
    payload.addHyperUtterance( s: "sdkdemo://input",
        list: null,
        s1: "input",
        params);

    BotSdk.getInstance().updateUiContext(payload);
}
```



此时用户发起语音请求：“输入北京”，App 会收到 onClickLink 的回调，url 就是我们前面定义的 sdkdemo://input，从 URL 中后续参数可以取出输入值：北京。

```
@Override
public void onClickLink(String url, HashMap<String, String> paramMap) {
    Log.d(TAG, "onClickLink: " + url + ", " + paramMap);
    if("sdkdemo://input".equals(url)){
        ((TextView)findViewById(R.id.input)).setText(paramMap.get("content"));
    }

    Toast.makeText(context: MainActivity.this, text: "url = " + url, Toast.LENGTH_SHORT).show();
}
```

4.6 打开意图调用时序图

下图是打开一个第三方 App 时，事件传递和调用的时序图，展示了处理一个打开意图时 App 客户端与 DuerOS 的数据交互和 App 状态变化。

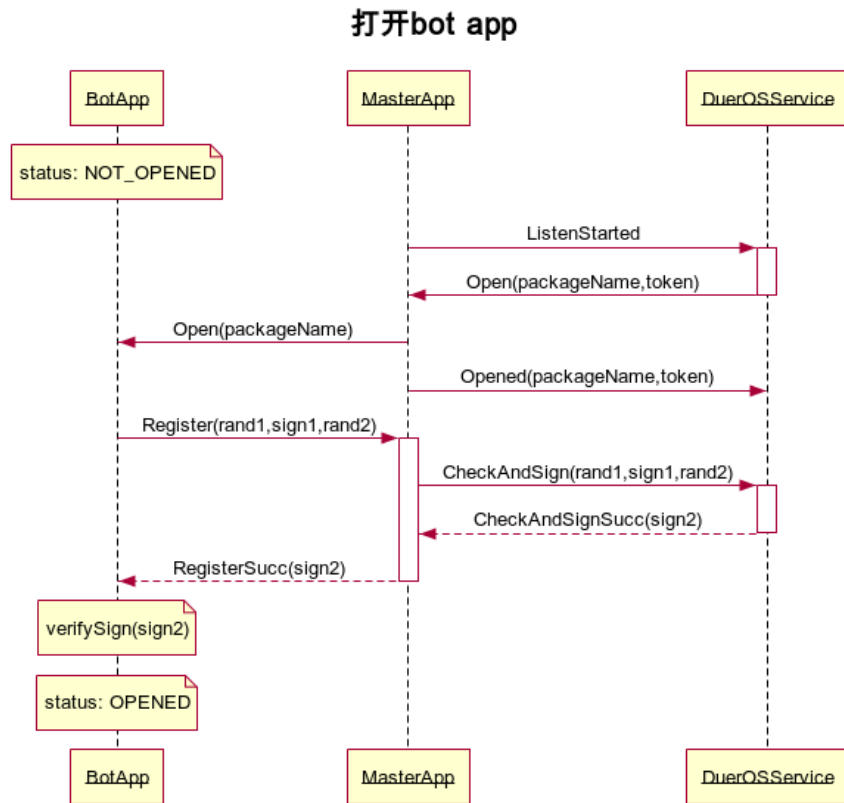


图 4.5 打开意图时序图

- 1) 用户唤醒设备，发起“打开 XXX”的请求，Maste App 上报 ListenStarted 事件。
- 2) DuerOS 服务端解析用户意图，从 DBP 平台获取对应 Bot 信息（App 包名）并下发 Open 指令



- 3) 设备收到打开指令，解析其中 App 包名，调起目标 App
- 4) App 启动后向 BotSKD 发起注册请求
- 5) 注册成功后，DuerOS 透过 BotSDK 向 App 下发注册成功的事件，开始语音交互。

参考文档

- 接入文档 (非常重要): <https://github.com/dueros/AndroidBotSdkDemo>
- 技能课堂中心(重要):
<https://dueros.baidu.com/didp/news/technicalclass?id=dbp>
- 开放平台文档中心: https://dueros.baidu.com/didp/doc/dueros-bot-platform/Introduction_markdown
- 百度开放平台: <https://dueros.baidu.com/dbp/main/console> (需要登录)