PDX BaaP 平台

开发人员手册

（安卓智能设备）

2017 年 11 月

北京全息智信（PDX）

目 录

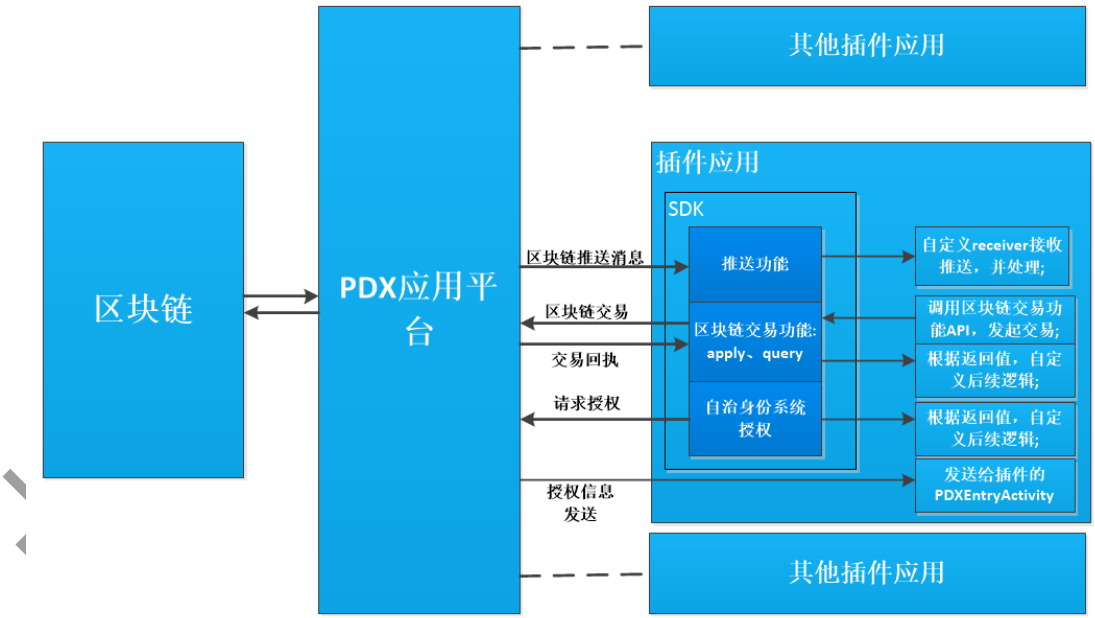
北京金思智信（PDX）

1 修订记录

编号	修改日期	修改内容
1	2016 年 6 月 30 日	1.0.0 版
2	2017 年 6 月 30 日	2.0.0 版
3	2017 年 11 月 17 日	2.1.0 版

2 概述

区块链、PDX 平台、SDK、插件应用之间关系



2.1 阅读提示

本文是 pdx-plugin Android SDK 标准的概述、集成、API 使用文档。目标读者为对区块链移动应用有兴趣的读者、开发者、合作伙伴等。

©北京全息智信科技有限公司。保留所有权利。

2.2 SDK 特点

开发者需要了解，所开发的移动应用(插件应用)是运行在 PDX 移动平台上的。运行在平台上，可以实现三大功能：与区块链交互、接收区块链消息推送、自治身份系统。SDK 提供了简洁的 API，轻松几行代码即可完成集成。集成 SDK 后，插件即可运行在 PDX 移动平台上。

2.3 SDK 支持版本

目前 SDK 支持 API Level >= 11（Android 3.0 及以上）的各个版本的手机系统。

IDE 要求使用 Android Studio 3.0 及以上（SDK 采用 jdk1.8 编译）。

3 SDK 集成及 API 使用

3.1 SDK 集成

3.1.1 使用提示

该部分是 pdx-plugin Android SDK 的集成指南。用以指导 SDK 的集成，默认读者已经熟悉 IDE（Android Studio）的基本使用方法，已经具有一定的 Android 编程知识基础。

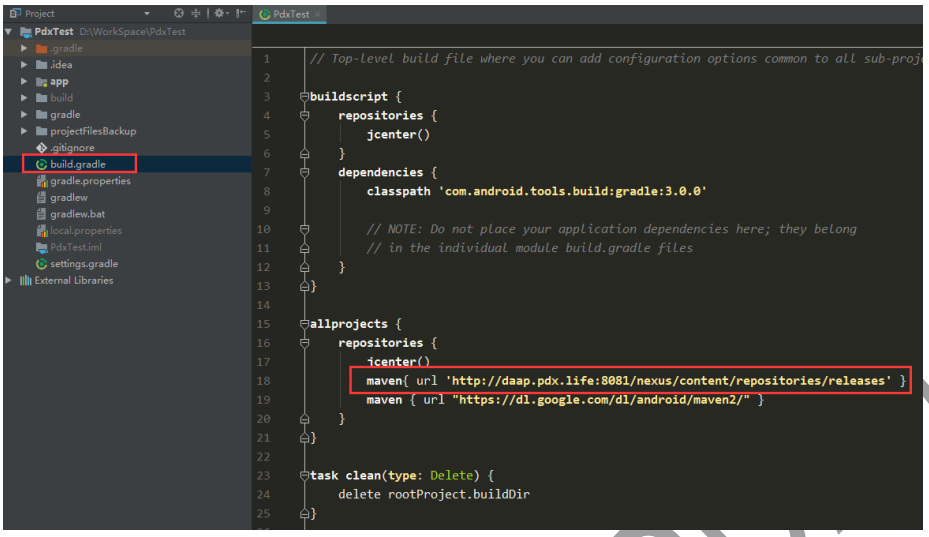
3.1.2 Gradle 集成步骤

1、在项目 build.gradle 中，增加一行 maven 仓库地址配置：

maven{url 'http://daap.pdx.life:8081/nexus/content/repositories/releases' }

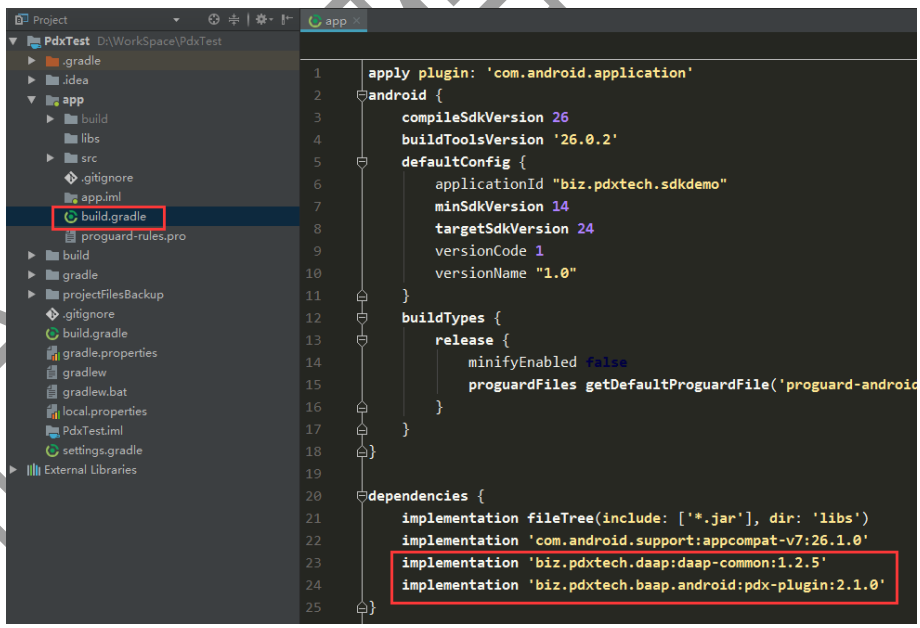
北京全息智信（PDX）

©北京全息智信科技有限公司。保留所有权利。



2、 配置完 maven 仓库后，在主 Module 的 build.gradle 中，增加如下配置：

- implementation 'biz.pdxtech.daap:daap-common:1.2.5'
- implementation 'biz.pdxtech.baap.android:pdx-plugin:2.1.0'



注意，完成以上步骤后：

- 如需集成区块链交易功能，请参考下文与区块链交互配置部分；
- 如需集成消息推送功能，请参考下文消息推送部分;

©北京全息智信科技有限公司。保留所有权利。

- 如需集成自治身份系统授权功能，请参考下文自治身份系统配置部分；
- 如需混淆，请参考下文混淆配置部分；

3.1.3 附加配置部分

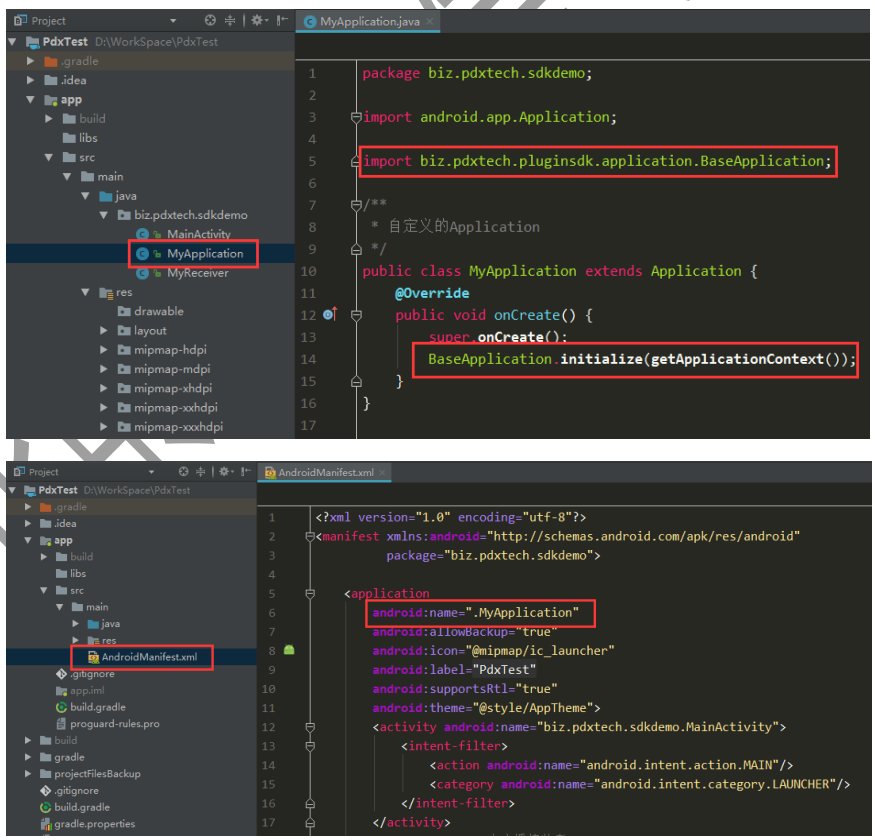
1、 与区块链交互配置

在自定义的 Application 的 onCreate 方法中，增加一行代码：

BaseApplication.initialize(getApplicationContext());

注意：

- 需要 import biz.pdxtech.pluginsdk.application.BaseApplication;
- 别忘了在 Manifest 中注册 MyApplication！



至此，与区块链交互功能的配置已完成。

北京全息智信科技有限公司（PDX）

©北京全息智信科技有限公司。保留所有权利。

2、 消息推送配置

消息推送采用广播，开发者需自定义 BroadcastReceiver 来接收广播消息，别忘了配置 Manifest。

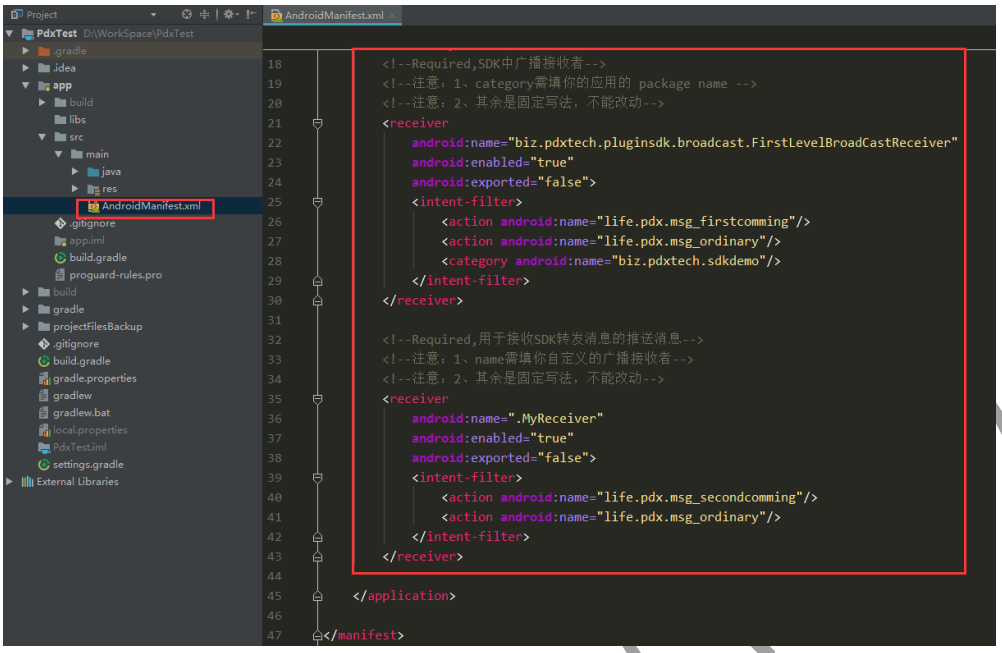
- 自定义 BroadcastReceiver：

```
public class MyReceiver extends BroadcastReceiver {
    private String receiverMsg = "";

    @Override
    public void onReceive(Context context, Intent intent) {
        Bundle bundle = intent.getExtras();
        if (intent.getAction().equals(MsgType.MSG_COMMING)) {
            receiverMsg = bundle.getString(MsgType.MSG_CONTENT);
            Log.d("区块链推送消息", receiverMsg);
            //TODO 获取到 receiverMsg 后，自定义后续操作
        }
    }
}
```

- 配置 AndroidManifest.xml：

北京全息



步骤为：

- 复制下面代码到项目的 AndroidManifest.xml 中；
- 将备注为“** your package name **”的部分，替换为当前应用程序的包名；*自定义一个广播接收者，将备注为“** your receiver name **”的部分，替换为该广播接收者。（自定义广播接收者及其代码，参照下文“自定义推送消息接收者”部分）。

```
<!--Required,SDK 中广播接收者-->
<!--注意：1、category 需填你的应用的 package name -->
<!--注意：2、其余是固定写法，不能改动-->
<receiver
    android:name="biz.pdxtech.pluginsdk.broadcast.FirstLevelBroadCastReceiver"
    android:enabled="true"
    android:exported="false">
    <intent-filter>
        <action android:name="life.pdx.msg_firstcomming"/>
        <action android:name="life.pdx.msg_ordinary"/>
```

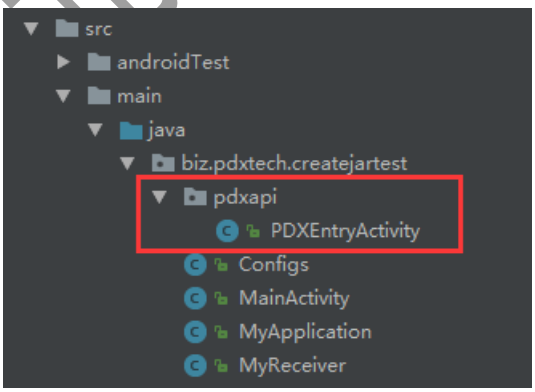
©北京全息智信科技有限公司。保留所有权利。

```
        <category android:name="** your package name **"/>
    </intent-filter>
</receiver>

<!--Required,用于接收 SDK 转发消息的推送消息-->
<!--注意：1、name 需填你自定义的广播接收者-->
<!--注意：2、其余是固定写法，不能改动-->
<receiver
    android:name="** your receiver name **"
    android:enabled="true"
    android:exported="false">
    <intent-filter>
        <action android:name="life.pdx.msg_secondcomming"/>
        <action android:name="life.pdx.msg_ordinary"/>
    </intent-filter>
</receiver>
```

3、 自治身份系统授权配置

A、自治身份系统授权采用 Intent 传数据，开发者需在你的包名相应目录下新建一个 pdxapi 目录，并在该 pdxapi 目录下新增一个 PDXEntryActivity 类，该类继承自 Activity(例如应用程序包名为 biz.pdxtech.createjartest，则新添加的类如下图所示)



北京全息智信

©北京全息智信科技有限公司。保留所有权利。

并在 manifest 文件里加上 exported 属性，设置为 true,例如：

```
<activity
    android:name=".pdxapi.PDXEntryActivity"
    android:exported="true"/>
```

B、实现 IPDXAPIEventHandler 接口，PDX 发送的请求将回调到 onReq 方法，发送到 PDX 请求的相应结果将回调到 onResp 方法

C、在 PDXEntryActivity 中将接收到的 intent 及实现了 IPDXAPIEventHandler 接口的对象传递给 IPDXAPI 接口的 handleIntent 方法，实例如下图：

```
api.handleIntent(intent, var2: this);
```

当 PDX 发送请求到你的应用，将通过 IPDXAPIEnventHandler 接口的 onReq 方法进行回调，类似的，应用请求 PDX 的相应结果将通过 onResp 回调。

4、 混淆配置

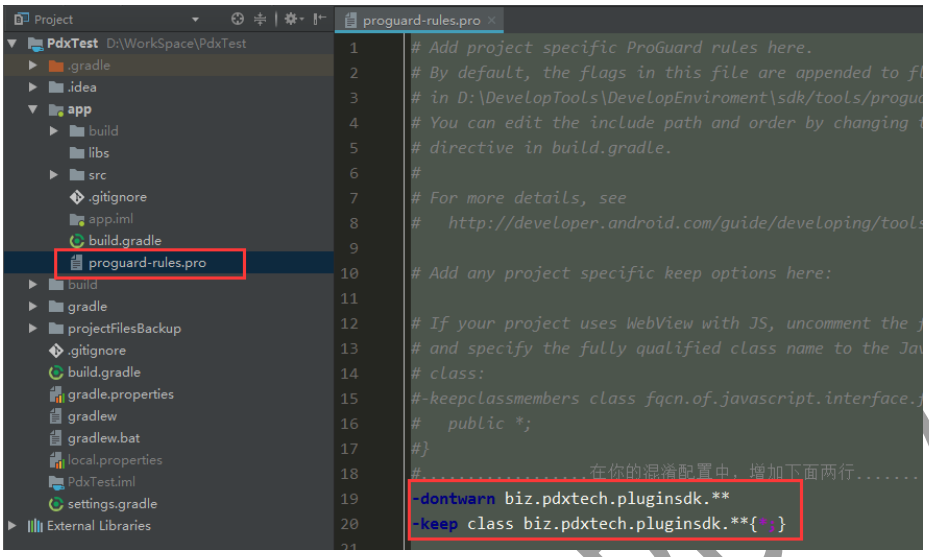
需要做代码混淆的开发者，请在主 Module 的混淆文件中添加以下配置：

-dontwarn biz.pdxtech.pluginsdk.**

-keep class biz.pdxtech.pluginsdk.**{*;}

北京全息智信（PDX）

©北京全息智信科技有限公司。保留所有权利。



检查确认

- 确认所需的权限都已添加。如果必须的权限未添加，日志中会提示错误。
- 确认 Required 部分已经正确写入 AndroidManifest。

3.2 SDK API 使用

3.2.1 使用提示

该部分是 pdx-plugin Android SDK 中 API 的使用文档。旨在使开发者了解在集成 SDK 后，如何调用 SDK 中的 API，来实现与区块链交互、接收区块链消息推送、自治身份系统授权。默认读者已经先阅读了上文 SDK 集成部分，并且完成了所需功能的集成。

3.2.2 API 的使用

1、 与区块链交互功能

北京全息智信（PDX）

©北京全息智信科技有限公司。保留所有权利。

使用 API 前，开发者需要知道，SDK 中区块链的操作有两种：apply（写）、query（查），都是以 Transaction 类的实例（如下文的 transaction）为操作单元。 在 API 中，这两种操作的调用方法为：

String result1 = BlockchainCtx.getInstance().apply(transaction)；

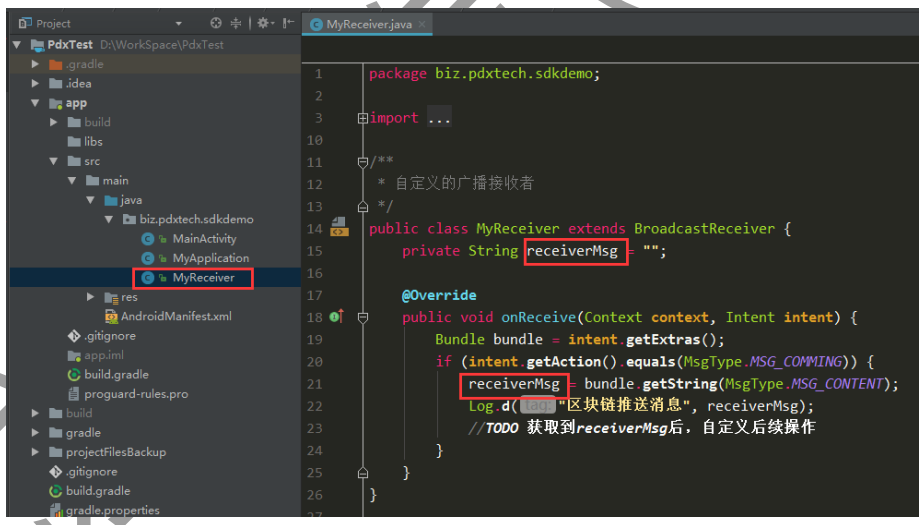
List<Transaction> result2= BlockchainCtx.getInstance().query(transaction)；

注意:

上述 Transaction 和 transaction,都是用的 biz.pdxtech.baap.api.contract.Transaction 下的。

2、 接收区块链消息推送功能

参考下图（上述的自定义的广播接收者 MyReceiver）：



图中 receiveMsg 即是你接收到的消息推送的内容，根据内容自定义后续处理逻辑即可。

3、 自治身份系统授权功能

北京全息智信科技有限公司（PDX）

©北京全息智信科技有限公司。保留所有权利。

自治身份系统是将用户的信息保存在 PDX 移动平台，最大限度保护用户信息。插件可以通过访问自治身份系统，请求用户证书信息，系统会记录请求，同时提示用户授权，授权记录不可更改。自治身份系统主要提供两大功能：

- 插件请求自治身份系统的证书信息；
- 插件用户之间互相请求对方证书信息；

3.2.3 主要 API 插件请求自治身份系统的证书信息：

1、 插件查询 PDX 支持的证书类型（会有多种类型，按需选一个）

AuthUtil. getAuthType ();

2、 插件查询 1 中某种证书类型对应的证书公钥（参数为：步骤 1 的类型）

AuthUtil. getAuthPubKey (String method)；

3、 插件请求某种类型证书信息（参数为： 1 中查询出的类型，插件公钥）

AuthUtil. applyCert (String method, String pluginKey)；

4、 插件校验步骤 3 中请求的证书 Token（若过期，需重新请求）

AuthUtil. isTokenExpire (String token);

插件用户之间互相请求对方证书信息：

5、 插件用户之间请求对方的某一类型的证书（参数为：证书类型，插件应用公钥）

AuthUtil. applyUserCert (String method, String pluginKey)；

6、 插件校验证书的合法性（参数为：证书信息，步骤 2 中证书公钥）

AuthUtil.isEndorsement(String endorse, String endorsementPubKey)；

北京全息智信（PDX）

©北京全息智信科技有限公司。保留所有权利。