支付平台接口说明书

Payment Platform API

V 1.0

1. 介绍

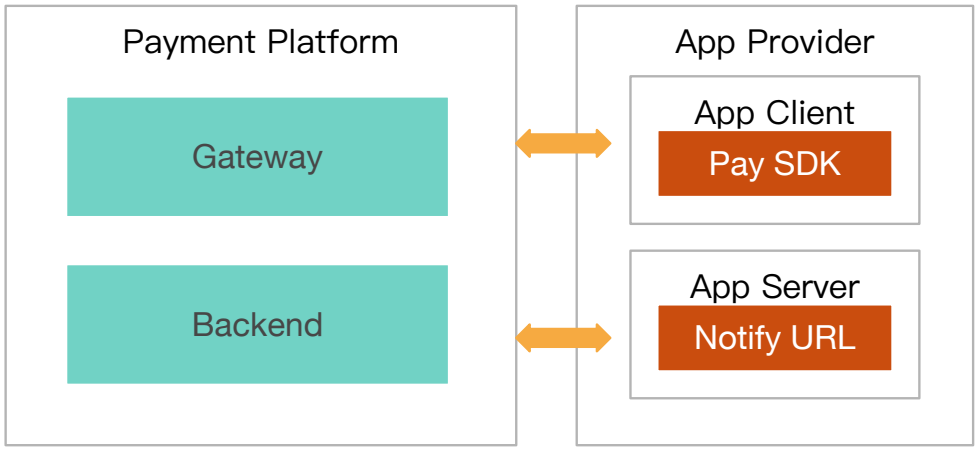
1.1. 关于支付平台接口说明书

本文档用于描述 支付平台（Payment Platform）与App之间的API接口规范，App接入时，需参考该文档进行相关程序开发和测试。

1.2. API接口网络架构

应用提供商将支付平台提供的支付SDK嵌入App，并提供用于接收支付结果通知的回调接口URL地址。

支付平台API接口架构图



2. 接口说明

API接口包括以下两部分：

1. **支付SDK接口** 由支付平台提供，App下单、发起支付请求时使用。

2. **支付结果通知接口** 由App提供商提供，用于接收支付平台回传的订单支付结果通知。

2.1. 支付SDK接口

2.1.1. 接入方式

将 paylibs-all-debug.aar 加入到libs文件夹，然后在项目 build.gradle 中添加依赖:

```
repositories {
    flatDir {
        dirs 'libs'
    }
}
dependencies {
    ……
    compile(name:'paylibs-all-debug', ext:'aar')
}
```

在清单文件中添加:

```
<activity
    android:name="com.msymobile.payment.sdk.paylibs.H5PayCenterActivity"
    android:configChanges="orientation|keyboardHidden|navigation|screenSize"
    android:exported="false"
    android:screenOrientation="behind"
    android:windowSoftInputMode="adjustResize|stateHidden">
</activity>
```

2.1.2. 接口调用

第一步、初始化，初始化支付中心对象，配置payId，paySecret，根据需求配置弹窗：

```
PayCenter.newBuilder(this)
    .paymentPopup(mPaymentPopup)
    .payId(payId)
    .paySecret(secret)
    .build();
//支付结果弹窗
mPaymentResultPopup = new PaymentResultPopup(this);
```

第二步、根据需要开启日志

```
Logger.setDebug(true);
Logger.setTag(“hannibal");
```

第三步、支付方式。本sdk提供了简易的UI界面。

调用原生控件：

```
mPaymentPopup.show("测试商品2",1);
```

自定义处理：

```
PayCenter.requestPayments(new PaymentCallBack() {
    @Override
    public void success(int what, ArrayList<PaymentEntity> payments, Response response) {
        //请求支付方式，开发者可以选择从payments获取封装好的javabean，
        //也可以用response.get().toString()获取返回的json
    }

    @Override
    public void failed(int what, Response response) {
        ...
    }
});
```

根据用户选择的PaymentEntity，调用相应的微信支付或支付宝支付。

第四步、进行支付

如果开发者使用原生控件，则不需要对支付过程进行处理。

自定义处理:

```
 /**
    * 微信支付支付
    *
    * @param appOrderNum 自定义订单号 (必填)
    * @param attach      拓展信息 (选择)
    * @param payment     支付方式(必填)
    * @param amount      支付金额(必填)
    * @param productName 商品名称(必填)
    * @param callBack    回调
    */
PayCenter.weChatWebPay(appOrderNum, attach, 1, "测试商品1", new PayCallBack() {
    @Override
    public void success(int what, Response response) {
        //此处已经进行过跳转处理，可以在response对象中拿到跳转的url。
    }

    @Override
    public void failed(int what, Response response) {
        ...
    }
});

//支付宝支付
PayCenter.aliWebPay(appOrderNum, attach, 1, "测试商品1", new PayCallBack() {
    @Override
    public void success(int what, Response response) {
        ...
    }

    @Override
    public void failed(int what, Response response) {
        ...
    }
});
```

第五步、支付结果处理

由于是用webview调起支付，客户端无法自动执行查单操作，所以让用户去点击按钮触发查单操作。

原生控件:

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    ...
    mPaymentResultPopup.onActivityResult(requestCode, resultCode, data);
}
```

自定义处理:

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    ...
    if(requestCode == 10086 && resultCode == 10087) {
        this.show();
        this.mOid = data.getStringExtra("oid");//获取到订单号
    }
}

//查单操作
PayCenter.requestPayResult(this.mOid, new PayResultCallBack() {
    public void success(int what, PayStatusEnum flag, Response response) {
        //flag表示该订单号状态
        //UMPAY(0),//未支付
        //PAY(1),//支付成功
        //FAIL(2),//支付失败
        //NET_ERROR(-2),//网络错误
        //UNKNOWN(-1);//未知
    }

    public void failed(int what, Response response) {
        ....
    }
});
```

第六步、支付完毕，资源释放。

```
@Override
protected void onDestroy() {
    super.onDestroy();
    //关闭弹窗
    mPaymentResultPopup.dismiss();
    mPaymentPopup.dismiss();
}
```

2.2. 支付结果通知接口

接口协议：HTTP/HTTPS

请求方式：POST

参数格式：JSON

2.2.1. 接口描述

请求接口时，POST提交的JSON内容格式：

```
{
    "oid":"234234234234234",
    "app_order_num":"2342342342",
    "total_amt":"1000",
    "pay_status":"1",
    "pay_time":"123124234",
    "signature":"sdfasdfl3r"
}
```

接收成功后，接口回应：

```
success
```

返回其他内容或无响应表示失败，支付平台可能回根据情况进行重试。

2.2.2. 签名加密算法

App方服务器在接收到回调通知后，可通过验证传递的签名是否有效，来确认通知内容的合法性。

签名生成规则：将Json中的参数（signature除外）按参数名进行字符排序后，拼接成字符串，参数名与值之间以“=”分隔，参数与参数之间以“\n”分隔（最后一行

后加“\n”），然后再将字符串进行HMAC SHA256加密，加密的Key使用paySecret。

Java样例代码：

```
TreeMap<String, String> sortedDataMap = new TreeMap<>()
sortedDataMap.put("oid", oid);
sortedDataMap.put("app_order_num", app_order_num);
sortedDataMap.put("total_amt", total_amt);
sortedDataMap.put("pay_status", pay_status);
sortedDataMap.put("pay_time", pay_time);
StringBuilder stringBuilder = new StringBuilder();
for (Map.Entry<String, String> entry : sortedDataMap.entrySet()) {
    stringBuilder.append(entry.getKey()).append("=").append(entry.getValue());
    stringBuilder.append("\n");
}

byte[] b = HmacUtils.hmacSha256(paySecret, stringBuilder.toString());
String signature = Base64Utils.encodeToString(b);
```