

计费中间件 SDK 嵌入使用说明

文件集成：

1. 在 AndroidManifest.xml 中添加用户权限：

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.READ_CONTACTS" />
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />
<uses-permission android:name="android.permission.GET_TASKS" />
```

2. 在 AndroidManifest.xml 注册 service：

```
<service android:name="com.ckmobilling.CkService"
android:exported="true"
android:process=":remote" >
<intent-filter>
<action android:name="com.ckmobilling.ICkService" />
</intent-filter>
</service>
```

3. 在 AndroidManifest.xml 添加应用信息：

```
<meta-data android:name="app_id" android:value="/xxx" />
<meta-data android:name="app_channel" android:value="/xxx" />
```

app_id: 应用 id

app_channel: 渠道号

注：“xxxxx”为字符串。

4. 分别拷贝 assets、libs、res 资源至应用对应目录下；

5. 拷贝 runtime 目录至应用根目录下；

6. 支付界面的 Activity 需继承自 CkSdkActivity；

7. 应用的 Application 需继承自 CkSdkApp。

代码集成：

1. 短信支付处理

调用 CkSdkApi 类下的 doPayment 函数进行支付操作。函数内部会根据服务器设置，自动选择使用支付 SDK 进行支付处理。

函数声明如下：

```
public void doPayment(String ckPayCode, PaymentCallback callback);
```

参数说明：

ckPayCode 触控计费代码，服务器会根据具体支付 sdk 生成对应的支付代码或订单。

callback 支付结果反馈，提供开发者支付成功、失败和取消反馈信息。详见后续说明。

PaymentResult：支付结果类。详见后续说明。

示例代码：

```
CkSdkApi.getInstance().doPayment("0008", new PaymentCallback() {  
    public void paySuccess(PaymentResult result) {  
        Log.d(TAG, result.toString());  
        Map<String, String> map = result.getItemMapping();  
        String name = map.get(PaymentResult.FEED_BACK_ITEM_NAME);  
        String code = map.get(PaymentResult.FEED_BACK_ITEM_CODE);  
        String num = map.get(PaymentResult.FEED_BACK_ITEM_NUMBER);  
        Log.d(TAG, "name: "+ name+ ", code: "+ code+ ", num: "+ num);  
    }  
    public void payFailed(PaymentResult result) {  
    }  
    public void payCancelled(PaymentResult result) {  
    }  
});
```

2. PaymentCallback 支付结果回调接口类定义

应用需要 new 一个继承此接口类的对象来接收购买结果回调通知。类定义说明

如下：

```
public interface PaymentCallback {  
  
    public void paySuccess(PaymentResult result);  
  
    public void payFailed(PaymentResult result);  
  
    public void payCancel(PaymentResult result);  
  
}
```

paySuccess：支付成功回调函数。

payFailed：支付失败回调函数。

PayCancel：用户取消支付操作回调函数。

回调参数中的 **PaymentResult** 为结果描述类，包含支付使用的渠道及相关信息。

参见后面 **PaymentResult** 类的说明。

3. **PaymentResult** 支付结果描述类

方法描述：

getPayResult：获取支付结果，返回类型为 **int**（0 成功，1 失败，2 取消）。

getSdkType：获取支付类型，返回类型为 **int**（0 游戏基地，1 移动 MM，2 联动优势）。

getItemMapping：获取道具信息，返回类型为 **Map** 其 **key** 值分别对应为

PaymentResult.FEED_BACK_ITEM_NAME 、

PaymentResult.FEED_BACK_ITEM_CODE 、

PaymentResult.FEED_BACK_ITEM_NUMBER。

4. 道具购买参数映射关系结构表

“道具购买参数映射关系结构定义_V1.0_20140527.xml”文件描述了商品了具体信息，需由运营相关人员填写。具体参数说明详见该文件。