

SOMASDK

接入文档(外部)

版本：V1.0.1
请注意保密

目录

Android 接入	1
1.1. 工程设置.....	1
1.2. 初始化.....	2
1.3. 使用 SOMA 登录	2
1.4. 调用支付页面	3

Android 接入

1.1. 工程设置

1. 将 somasdk.aar 放到工程目录下的 libs 文件夹下
2. 在项目中的 app/build.gradle 文件中添加

```
repositories {  
    flatDir {  
        dirs 'libs'  
    }  
}  
  
dependencies {  
    implementation(name: 'somasdk', ext: 'aar')  
}
```

混淆重要说明： 在 proguard.cfg 文件中添加（不加会导致异常）

```
-dontwarn com.instanza.somasdk.**  
-keep class com.instanza.somasdk.**{  
    *;  
}
```

3. 修改 manifests.xml 文件中的 application 里面添加属性

```
<meta-data  
    android:name="com.instanza.sdk.ApplicationId"  
    android:values="@string/soma_app_id" />
```

说明： soma_app_id 需要自己在 strings.xml 里面配置。

soma_app_id 需要向 SOMA 这边申请和分配

1.2. 初始化

APPLication 里初始化 sdk

```
// 初始化 SDK
SomaSdk.sdkInitialize(getApplicationContext());
```

1.3. 使用 SOMA 登录

1, 创建 CallbackManager

```
callbackManager = CallbackManager.Factory.create();
```

2, 登录事件监听

```
// 注册登录的监听
LoginManager.getInstance().registerCallback(callbackManager, new
SomaCallback<LoginResult>() {
    @Override
    public void onSuccess(LoginResult loginResult) {
        // 登录成功, 可以从 loginResult 里得到 token 和用户信息
    }

    @Override
    public void onCancel() {
        // 取消了登录;
    }

    @Override
    public void onError(SomaException error) {
        // 登录失败;
    }
});
```

3, 移除登录事件监听

```
LoginManager.getInstance().unregisterCallback(callbackManager);
```

4, 调用 SOMA 登录

```
LoginManager.getInstance().loginWithReadPermissions (Context context)
```

5, 需要在 Activity 的 onActivityResult 里面做一下处理

```
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable
Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (null != callbackManager) {
        callbackManager.onActivityResult(requestCode, resultCode, data);
    }
}
```

```
}  
}
```

1.4. 调用支付页面

注意：如果当前没有使用 SOMA 成功登录或者登录过期，即没有有效的 `AccessToken`，调用支付功能会立即收到失败的回调。

1, 创建 `CallbackManager`

```
callbackManager = CallbackManager.Factory.create();
```

2, 注册支付事件监听

```
// 注册支付监听  
PurchaseManager.getInstance().registerCallback(callbackManager, new  
SomaCallback<PurchaseResult>() {  
    @Override  
    public void onSuccess(PurchaseResult purchaseResult) {  
        // 支付成功
```

支付成功后，调用方需要根据订单号，到 SOMA 服务器查询订单的状态。

```
    }  
  
    @Override  
    public void onCancel() {  
        // 支付取消  
    }  
  
    @Override  
    public void onError(SomaException error) {  
        // 支付失败  
    }  
});
```

3, 移除支付事件监听

```
PurchaseManager.getInstance().unregisterCallback(callbackManager);
```

4, 调用支付页面

```
// 调用支付  
findViewById(R.id.action_pay).setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        PurchaseClient.Request request = new PurchaseClient.Request("1000",  
"2028", "product",  
        "description",  
"https://d3jmlm80veil0l.cloudfront.net/ic_your_opinion_matters@3x.png",  
        "28", "Merchant");
```

```
PurchaseManager.getInstance().startPurchase(MainActivity.this,
request);
}
});
```

注意：构造 PurchaseClient.Request 时候的参数，都是调用方传进来的。
支付成功后，调用方需要根据订单号，到 SOMA 服务器查询订单的状态。

```
public Request(String app_order_id, String product_id, String product_name,
String description, String unit_img, String unit_price, String quantity, String
money_amount, String merchant_name, String sign)
```

app_order_id: 订单号

product_id: 商品的 ID，即商品 SKU

product_name: 商品的名字，用于页面展示

description: 商品的描述，用于页面展示

unit_img: 商品的图标，用于页面展示

unit_price: 商品的价格

quantity: 商品的数量（int 类型）

money_amount: 商品的总价

merchant_name: 商店的名字

sign: 签名，按照约定的方式加密，注意的是需要向 SOMA 申请 app_secret

5, 需要在 Activity 的 onActivityResult 里面做一下处理

```
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable
Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (null != callbackManager) {
        callbackManager.onActivityResult(requestCode, resultCode, data);
    }
}
```

版权说明：

此文档的版权归杭州英坦网络科技有限公司所有，作为本系统的最终用户，可以拥有该份文档的使用权，但未征得杭州英坦网络科技有限公司的书面批准，不得修改、公布本文档，不得向第三方借阅、出让、出版本文档。