MobKooSDK 对接文档

一、说明

版本 2.0

二、接入SDK

1、导包

Android Studio

> 将sdk android_studio 目录下的 libs 中文件复制到你项目libs中
>
> 项目中引用 aar 包
>
> 在 android 同级标签中增加以下代码
>
> repositories { flatDir { dirs 'libs' } }
>
> 方法、使用gradle 引用

```
implementation(name: 'mobkoolib_xx', ext: 'aar')
implementation 'com.facebook.android:facebook-android-sdk:4.29.0'
implementation 'com.facebook.android:facebook-login:4.29.0'
implementation 'com.google.android.gms:play-services-auth:16.0.1'
```

完整配置参考demo中的配置

2、配置

1、权限

```
<uses-permission android:name="android.permission.INTERNET" />
<!-- 谷歌支付权限 -->
<uses-permission android:name="com.android.vending.BILLING" />
```

2、在AndroidManife.xml 中配置

```
<application>
      <activity
            android:name="com.mobkoo.pay.newUI.MobKooPayActivity"
            android:configChanges="keyboardHidden|orientation|screenSize" />

        <meta-data
            android:name="com.facebook.sdk.ApplicationId"
            android:value="@string/facebook_app_id" />
    <provider
    android:name="com.facebook.FacebookContentProvider"
android:authorities="com.facebook.app.FacebookContentProviderXXX"
    android:exported="true" />

<activity
android:name="com.facebook.FacebookActivity"
android:configChanges="keyboard|keyboardHidden|screenLayout|screenSize|orientation"
android:label="@string/app_name" />

        <activity
            android:name="com.facebook.CustomTabActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.VIEW" />

                <category android:name="android.intent.category.DEFAULT" />
                <category android:name="android.intent.category.BROWSABLE" />

                <data android:scheme="@string/fb_login_protocol_scheme" />
            </intent-filter>
        </activity>
```

```
        <meta-data
            android:name="mobkoo_appid"
            android:value="YOUAPPID" />
        <meta-data
            android:name="mobkoo_appkey"
            android:value="YOUAPPKEY" />

        <meta-data
            android:name="mobkoo_sandbox"
            android:value="false" />

        <meta-data
            android:name="mobkoo_google_clientid"
            android:value="YOUGOOGLECLIENTID" />

        <meta-data
            android:name="mobkoo_google_licensekey"
            android:value="YOUGOOGLELICENSEKEY" />

</application>
```

3、初始化

1、初始化

```
 /**
     * @param context :上下文对象
     * @param appid : 应用标识
     * @param key :   网络秘钥
     * @param sandbox :沙盒模式 true 打开 false 关闭（打开时为开发测试）
     */
MobKooHelp.init(Application application,
                String APPID ,
                String APPKEY ,
                boolean sandbox);

// 如果在manifest设置过appID APPKEY 参数 可以直接调用无参init方法
MobKooHelp.init(Application application);
```

2 、 设置日志开关

参数 true 为打开log日志 false 为关闭日志

```
   MobKooHelp.setEnvDebug(true);
```

3、设置GoogleClientID（如果在manifest设置过 无需重复调用）

```
// googleClientID: google登录使用  值为google后台的 web客户端参数
  MobKooHelp.singleton().setGoogleClientID(String googleClientID);
```

4、设置Google许可秘钥（如果在manifest设置过 无需重复调用）

```
// GoogleLicenseKey: google支付使用  值为google后台的许可秘钥
  MobKooHelp.singleton().setGoogleLicenseKey(String GoogleLicenseKey);
```

5、 绑定声明周期

```
 // 每一个调用支付的Activity 都需要绑定 onActivityResult 否则客户端接收不到支付结果;
 @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent
data) {
        super.onActivityResult(requestCode, resultCode, data);
        MobKooHelp.singleton().onActivityResult(requestCode, resultCode, data);
    }

  /*
   横竖屏绑定
   */
    @Override
    public void onConfigurationChanged(@NonNull Configuration newConfig) {
        super.onConfigurationChanged(newConfig);
        MobKooHelp.singleton().onConfigurationChanged(newConfig);
    }

     @Override
    protected void onDestroy() {
        super.onDestroy();
        MobKooHelp.onDestory(this);

    }
```

1. 其他参见接口文档和demo

三、支付

1、 设置支付模式

```
 /**
    * 设置支付模式 默认模式 PAY_TYPE_ALL
    *  MobKooHelp.PAY_TYPE_ALL     :             google和MOBKOO支付
    *  MobKooHelp.PAY_TYPE_MOBKOO  :             只支持MOBKOO支付
    *  MobKooHelp.Pay_TYPE_GOOGLE  :             只支持Google支付
    */

   public void setPayType(int payType);

//示例代码:
    MobKooHelp.singleton().setPayType(MobKooHelp.Pay_TYPE_GOOGLE);
```

###

```
   /**
   Activity activity      上下文
   String productID,      google商品ID
   double price,         商品价格 单位是USD
   String outOrderId,     商品订单ID
   */
      public void StartPay(Activity activity,
                                   String productID,
                                   double price,
                                   String outOrderId,
                                   final MobKooPayCallBack callBack);

//示例代码:
    MobKooHelp.singleton().StartPay(activity,
                                   "productid",
                                   3.0,
                                   "extrastring",
                                   new MobKooPayCallBack() {
                  @Override
                  public void paySuccess(String payJson) {
                      Log.e("支付结果页 成功: ", "" + payJson);
                  }

                  @Override
                  public void payFail(String message) {
                      Log.e("支付结果页 失败: ", "" + message);
                  }
              });
```

1、 Android支付返回结果

```
/*
{"state":"succ",
"order_id":"2020052514522017616044",
"out_order_id":"userid_1_serverId_6",
"price":"3.00",
"currency":"USD",
"sandbox":1,
"platform_name":"Wing",
"product_id":"sku1"}
*/

String state                    //支付状态
String order_id                 // 订单ID
String out_order_id             // 商户订单ID
String price                    //价格
String currency                 //货币单位
String sandbox                  // 是否是沙盒测试
String platform_name            // 支付平台
String product_id               // 商品ID
```

四、登录

1、登录

```
 /**
   * @param activity 上下文
   * @param callBack 登录结果回调
   */
public void login(Activity activity, final MobKooLoginCallBack callBack) ;

//调用示例
  MobKooHelp.singleton().login(activity,
                               new MobKooLoginCallBack() {
          @Override
          public void loginSuccess(String userJson) {
              Log.e("登录结果 success: ", userJson + "");
           }

          @Override
          public void loginFail(String message) {
              Log.e("登录结果: error", message + "");
          }
      });
```

2、 查询用户信息

```
    /**
     * 获取正在登录中的用户信息
     * @param callBack 用户信息回调
     */
    public void getUserInfo(MobKooLoginCallBack callBack);

// 查询上次登录是否有效
    MobKooHelp.singleton().getUserInfo(new MobKooLoginCallBack() {

            @Override
            public void loginSuccess(String userJson) {
                Log.e("登录结果 success: ", userJson + "");
             }

            @Override
            public void loginFail(String message) {
                Log.e("登录结果: error", message + "");
            }

        });
```

6、 退出登录

```
MobKooHelp.singleton().logout();// 调用后getUserInfo获得的返回值为error
```

用户信息（Json字符串）

```
/*
示例数据
{"user_id":"11",
"nickname":"xiaoming",
"email":"xiaoming123@qq.com",
"token":"82f305ef4d7c9d0289998e4a3c53",
"log_type":"mobkoo",
"userIcon":""}
*/
String user_id                // 用户ID
String nickname               // 用户名
String email                  // 用户邮箱
String token                  // 用户秘钥
String log_type               // 登录方式
String userIcon               // 用户头像
```