

MobKooSDK 对接文档

一、说明

二、接入SDK

- 1、导包
- 2、配置
- 3、初始化
- 5、绑定声明周期

三、支付

- 1、支付
- 1、 Android支付返回结果

四、登录

- 1、登录
- 2、 查询用户信息
- 3、退出登录
- 4、用户信息 (Json字符串)

一、说明

版本 2.1.0

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

```
<!-- 联网权限 -->
<uses-permission android:name="android.permission.INTERNET" />
<!-- 谷歌支付权限 -->
<uses-permission android:name="com.android.vending.BILLING" />
<!-- 获取网络状态 -->
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
```

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" />
        </intent-filter>
    </activity>
</application>
```

```

        <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 Application :application对象
 * @param appid : 应用标识
 * @param key : 网络密钥
 * @param debug : 日志开关 true 打开 false 关闭
 * @param sandbox :沙盒模式 true 打开 false 关闭 (打开时为开发测试)
 * @param type :支付模式 默认模式 PAY_TYPE_ALL
 *                      MobKooManager.PAY_TYPE_ALL : google和MOBKOO支付
 *                      MobKooManager.PAY_TYPE_MOBKOO : 只支持MOBKOO支付
 *                      MobKooManager.PAY_TYPE_GOOGLE : 只支持Google支付
 * @param GoogleClientID : google登录使用 值为google后台的 web客户端参数
 * @param GoogleLicenseKey :google支付使用 值为google后台的许可密钥
 */

MobKooManager.getInstance()
    .setContext(Application application)
    .setAppInfo(String appid,String appkey)
    .setDebug(boolean debug)
    .setSandBox(boolean sandbox)
    .setPayType(int type)
    // 设置GoogleClientID (如果在manifest设置过 无需重复调用)

```

```

        .setGoogleClientID(String GoogleClientID )
        // 设置GoogleClientID (如果在manifest设置过 无需重复调用)
        .setGoogleLicenseKey(String GoogleLicenseKey )
        .init();

```

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);
    MobKooManager.getInstance().onActivityResult(requestCode, resultCode, data);
}

@Override
protected void onDestroy() {
    super.onDestroy();
    MobKooManager.getInstance().onDestory(this);
}

```

1. 其他参见接口文档和demo

三、支付

1、支付

```

/**
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:
MobKooManager.getInstance().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
MobKooManager.getInstance().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 Activity 上下文
 * @param callBack 用户信息回调
 */
public void getUserInfo(Activity activity, MobKooLoginCallBack callBack);

// 查询上次登录是否有效
MobKooManager.getInstance().getUserInfo(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 + "");
    }
}
});

```

3、退出登录

```

MobKooManager.getInstance().logout(Activity); // 调用后getUserInfo获得的返回值为error

```

4、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            // 用户密钥
String log_type         // 登录方式
String userIcon         // 用户头像
```