

海游登录SDK

海游登录SDK

- 一、基本信息
- 二、快速对接
 - 1. 准备工作
 - 2. 配置 AndroidManifest.xml
 - 3. build.gradle 配置
- 三、使用
 - 1、初始化
 - 2、启用登录
 - 3、查询上次登录信息
 - 4、退出登录
 - 5、打开公共模块悬浮框
 - 6、使用class路径引用
- 三、用户参数说明
- 四、更新记录

一、基本信息

- SDK版本：3.0

二、快速对接

1. 准备工作

1. 复制 libs 中的文件到自己的项目 libs 中

2. 配置 AndroidManifest.xml

- 权限声明

```
<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.READ_PHONE_STATE" />
```

- Application声明

```
<meta-data
    android:name="wallet_app_id"
    android:value="APP ID" /> 代码中的优先使用

<!-- 海游登录sdk 配置 -->
<activity
```

```

        android:name="com.walletfun.login.ui.RegisterOrForgotActivity"
        android:configChanges="screenSize|keyboardHidden|orientation" />

<activity
    android:name="com.walletfun.login.ui.LoginActivity"
    android:configChanges="screenSize|keyboardHidden|orientation" />
<!-- 还有活动页面 如果已经注册 无需重复-->
<activity
    android:name="com.walletfun.common.app.HaiYouShowActivity"
    android:theme="@style/walletPayTheme"
    android:configChanges="screenSize|keyboardHidden|orientation" />

<!-- facebook 登录 -->
<meta-data
    android:name="com.facebook.sdk.ApplicationId"
    android:value="@string/facebook_app_id" />

<!-- 注意: 951132541753416 这个值请更换成 facebook_app_id 的值-->
<provider
    android:name="com.facebook.FacebookContentProvider"
    android:authorities="com.facebook.app.FacebookContentProvider951132541753416"
    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>

<!-- facebook end -->

```

3. build.gradle 配置

```

minSdkVersion 16 (大于等于16)

// aar 必须配置的
repositories {
    flatDir {
        dirs 'libs'
    }
}

```

```

    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])

    // 海游 SDK xx用版本号代替
    implementation(name: 'haiyou_common_xx', ext: 'aar')
    implementation(name: 'haiyou_login_xx', ext: 'aar')

    // 其他第三方 SDK 有两种方式 根据自己开发环境选择一种
    // 1 、如果支持网络库的依赖
    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'

    // 2 、本地库依赖
    implementation(name: 'play-services-auth-16.0.1', ext: 'aar')
    implementation(name: 'play-services-auth-base-16.0.0', ext: 'aar')
    implementation(name: 'play-services-base-16.0.1', ext: 'aar')
    implementation(name: 'play-services-basement-16.0.1', ext: 'aar')
    implementation(name: 'play-services-tasks-16.0.1', ext: 'aar')

    implementation(name: 'facebook-core', ext: 'aar')
    implementation(name: 'facebook-common', ext: 'aar')
    implementation(name: 'facebook-login', ext: 'aar')
    implementation(name: 'cardview-v7-28.0.0', ext: 'aar')
}

```

三. 使用

1、初始化

```

//将登录SDK加入到公共模块
walletHelp.addSDK(HaiYouLoginSdk.instance());
// 设置google登录参数
HaiYouLoginSdk.setGoogleServerClientId(googlekey);
/* 初始化登录SDK ,
初始化公共参数 appid (测试id) 加密密钥key 沙盒模式 (沙盒模式暂时只有充值使用) ; */
walletHelp.init(this, appid, key, walletHelp.SANDBOX_CLOSE);
// 设置DEBUG模式 和日志相关 上线时建议关闭
walletHelp.setEnvDebug(false);
// 初始化 登录辅助类
helper = HaiYouLoginHelper.getInstance(this);

// 设置登录回调方法
helper.setLoginCallback(new HaiYouLoginCallback() {
    @Override
    public void onResult(HaiYouLoginResult result) {

```

```

        // 登录结果
        switch (result.getStatus()) {
            case HaiYouLoginStatus.LOGIN_SUCCESS:
                // 成功
                resultEdit.append("\n 登录成功");
                loginSuccess(result.getUser());
                break;
            case HaiYouLoginStatus.LOGIN_CANCEL:
                // 用户取消了登录
                resultEdit.append("\n用户取消了登录" +
                    helper.platformName(result.getPlatform()));
                break;
            default:
                // 登录失败
                resultEdit.append("\n登录失败: " +
                    helper.platformName(result.getPlatform()));
                loginFailure(result.getPlatform(), result.getFailed());
                break;
        }
    }
});
// 在onActivityResult 中调用 help的handlerActivityResult方法 如未调用 收不到回调数据
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    // 很重要的一步, 自己使用 startActivityForResult 时, requestCode 请不要设置成 10001 到 10005
    if (helper.handlerActivityResult(requestCode, resultCode, data)) return;
    super.onActivityResult(requestCode, resultCode, data);
}
// 在 onConfigurationChanged中 调用help中的 onConfigurationChanged方法
@Override
public void onConfigurationChanged(Configuration newConfig) {
    super.onConfigurationChanged(newConfig);
    if (helper != null) helper.onConfigurationChanged(newConfig);
}
// 销毁
@Override
protected void onDestroy() {
    super.onDestroy();
    if (helper != null) helper.destroy();
}

```

2、启用登录

```

private void login(){
    // 使用默认登录方式
    helper.showLogin(HaiYouLoginHelper.LOGIN_DEFAULT);

    // 以下为单独调用代码, 自定义登录界面
    // facebook
    // helper.showLogin(HaiYouLoginHelper.LOGIN_PLATFORM_FACEBOOK);
}

```

```

        // google
        // helper.showLogin(HaiYouLoginHelper.LOGIN_PLATFORM_GOOGLE);

        // 海游
        // helper.showLogin(HaiYouLoginHelper.LOGIN_PLATFORM_HAIYOU);

        // 游客
        // helper.showLogin(HaiYouLoginHelper.LOGIN_PLATFORM_GUEST);
    }

    private void loginSuccess(HaiYouUser user) {
        // 登录成功的平台
        int platform = helper.platformName(user.getPlatform());

        // 用户的id
        String UserId =user.getId();

        // 用户的 token
        String token = user.getToken();

        // 用户的昵称, 可能为空为null
        String nickName = user.getNickname();

        // 用户头像 可能为 null 为空
        String userIcon = user.getUserIcon();
    }

    private void loginFailure(int platform, Object err){

        switch (platform) {
            case HaiYouLoginHelper.LOGIN_PLATFORM_FACEBOOK:
                // 使用 facebook 登录失败
                if (exception instanceof FacebookException) {
                    String msg = err.toString();
                }

                break;
            case HaiYouLoginHelper.LOGIN_PLATFORM_GOOGLE:
                if (exception instanceof ApiException) {
                    // 谷歌登录失败
                    int errorCode = (ApiException) exception).getStatusCode();
                }
                break;
            case HaiYouLoginHelper.REQUEST_LOGIN_HAIYOU:
            case HaiYouLoginHelper.LOGIN_PLATFORM_GUEST:

                if (exception instanceof Exception) {
                    // 错误信息
                    String msg = exception.toString();
                }
            }
        }
    }

```

```

        }
        break;
    default:
        // 未知原因
        break;
    }
}

```

3、查询上次登录信息

// 查询上次登录是否有效，目前谷歌登录需要每次进来都登录

```

// 查询上次登录是否有效，目前谷歌登录需要每次进来都登录
helper.hrtUser(new HaiYouUserCallback() {
    @Override
    public void onUserResult(HaiYouUser user) {
        // 如果为 null 则说明登录过期，或者是谷歌登录
        if (user == null) {
            resultEdit.append("请重新登录\n");
        } else {
            loginSuccess(user);
        }
    }
});

```

4、退出登录

```
helper.logout();
```

5、打开公共模块悬浮框

```
walletHelp.showFloat(this); // 在walletHelp.init()方法后调用
```

6、使用class路径引用

```

import com.facebook.FacebookException;
import com.google.android.gms.common.api.ApiException;
import com.walletfun.common.app.walletHelp;
import com.walletfun.login.HaiYouLoginCallback;
import com.walletfun.login.HaiYouLoginHelper;
import com.walletfun.login.HaiYouLoginResult;
import com.walletfun.login.HaiYouLoginSdk;
import com.walletfun.login.HaiYouLoginStatus;
import com.walletfun.login.HaiYouUser;
import com.walletfun.login.HaiYouUserCallback;

```

三、用户参数说明

```

public class HaiYouUser {

    // 用户ID
    private String id;
    // 用户所属国家CODE
    private String country_code;
    // 昵称
    private String nickname;
    // token
    private String token;
    // 头像
    private String userIcon;
    // 登录的平台
    private int platform;

    // (facebook 和google登录成功时 返回的数据)
    private ExtraDataBean extra_data;

    public static class ExtraDataBean implements Serializable{

        private String openID;           // 用户ID

        private String nickname;         // 用户昵称

        private int sex;                  // 性别

        private String email;            // 邮箱

        private String address;           // 地址
        private String userIcon;          // 头像
        private String platform;          // 平台 facebook or google
        private String accessToken;       // 平台token
        private String mobilePhone;       // 联系方式
    }
}

```

四、更新记录

版本	日期	说明	其他
3.0	2019.1.30	功能优化	支持内部数据统计
2.2	2018.12.24	bug修复	更改用户信息
2.1	2018.12.17	优化逻辑	
2.0.0	2018.11.16	完善功能	
1.0.0	2018.10.22	基本功能	