

集成指南

一、前言

本文档仅适用于个验3.0.0.0及以上版本，要求最小安卓版本为android-19，目前安卓sdk只提供aar包供集成。
使用限制说明：一键登录和号码校验的使用前提是用户打开蜂窝网络，目前支持的网络制式有移动 2G、3G、4G、5G。联通 3G、4G、5G，电信 4G，5G。对于双卡手机，以开启数据流量的 SIM 卡进行认证；对于Wifi环境，在数据网络开通的前提下，会强制使用数据网络触发网关请求，请求仅为几K，可正常校验。

3.0.0.0以前的版本 或者 仍旧使用的老版本一键登录接口，请查看以下文档：

[已废弃：老版SDK接入与初始化](#)

[已废弃：老版一键登录接口](#)

[已废弃：老版本机号码校验](#)

[已删除：动画验证码](#)

[已删除：反欺诈](#)

[已废弃：老版状态码汇总](#)

二、升级指南

新接入用户 请跳过本章节，**老版本用户** 请仔细核对以下升级指南。

1. 升级指南：1.x、2.x -> 3.0:

为了简化集成和性能优化，我们对整体个验SDK进行了较大的重构调整，升级时候请注意：

1.1 修改app/build.gradle的dependencies

去掉ellogin的依赖，现在ellogin已经合并进gysdk，再也不用当心gysdk和ellogin不匹配了

升级gtc请依赖

1.2 删除了动画验证、短信验证、和反欺诈相关的接口

删除了这些功能相关的接口：

```
picVerify、nonSenseVerify、smsCodeVerify、verify、checkLogin、checkRegister
```

1.3 授权页必要组件

如果**授权页的必要组件**不符合授权页规范的，**会无法一键登录**，打印日志：“UI不合规不能登录”。不合规行为比如：不可见、遮挡、修改内容等。

1.4 修改了部分errorCode值的定义

现在所有GYResponse.msg都会包含errorCode和errorDesc这两个字段，**3.0版本对一些errorCode定义进行了调整**，如果您app中有判断**errorCode**并进行对应特殊处理的话，请对照[《errorCode码含义》](#)检查修改。

1.5 修改了部分资源的名字和id

名字非gy开头的资源全部改成gy开头

请检查 ELoginThemeConfig.Builder().setXxx 是否使用了这些gysdk内置的drawable

请检查 是否使用了 fr_nav、animate_surface_view 这两个view id，前面加了gy_e_login_

现在andResGuard只需要配置 "R.drawable.gy_*" 以及ELoginThemeConfig.Builder().setXxx用到的资源

1.6 EloginActivity的setTheme可以用了

请检查AuthPageListener.onAuthActivityCreate中是否调用了setTheme动态修改EloginActivity的主题而导致。

1.7 废弃了不带参的sdk初始化接口和结果广播

废弃了 不带参的init接口和GyMessageReceiver：

```
class GyMessageReceiver;
void init(Context context);
```

建议使用带回调的sdk初始化接口，删除manifest或者代码中的结果广播注册

```
void init(Context context, GyCallBack gyCallBack);
```

1.7 废弃了授权页面相关接口

废弃了 使用ELoginThemeConfig配置UI属性，由gysdk启动授权页面activity的相关接口：

```
class ELoginThemeConfig;
interface AuthPageListener;
void eAccountLogin(ELoginThemeConfig themeConfig, GyCallBack gyCallBack);
void login(int timeout, ELoginThemeConfig config, GyCallBack gyCallBack);
void addRegisterViewConfig(String id, AuthRegisterViewConfig viewConfig);
void setAuthPageListener(AuthPageListener authPageListener);
boolean isPrivacyChecked();
void stopLoading();
void cancelELogin();
void finishAuthActivity();
```

建议自行定义授权页activity，启动后设置好隐私协议，然后调用新增加的登录接口

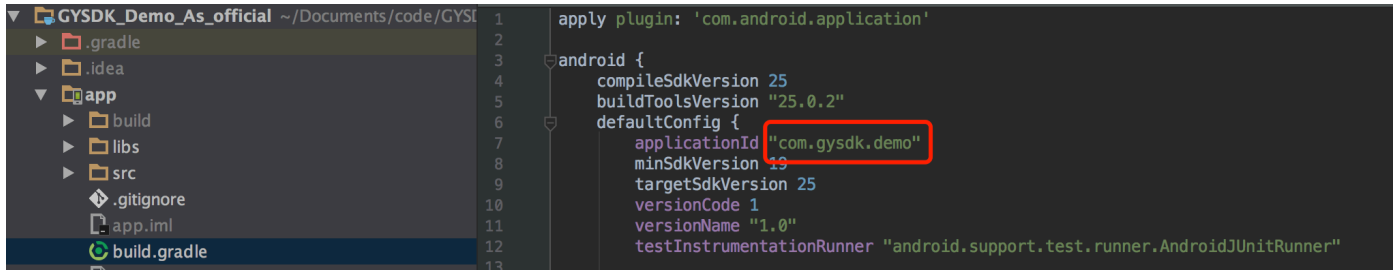
```
eAccountLogin(EloginActivityParam activityParam, int timeout, GyCallBack gyCallBack);
```

三、创建应用

GYSDK接入方需要提供 App的包名，签名证书的MD5、SHA1、SHA256等信息，用于鉴权。

1. 提取应用包名

Android 应用获取包名 `packageName`，根据开发工具不同，获取位置有所不同，以 Android Studio 为例说明如下：



2. 提取签名信息

步骤

1、在终端执行 `keytool -list -v -keystore` 您的签名文件路径
比如：`keytool -list -v -keystore debug.keystore`

2、输入证书密码，android自带的debug.keystore密码为android，用户自己生成的证书用签名时的密码，回车；

3、结果展示如下，选择对应的值

证书指纹：

MD5：

1E:F8:05:BA:86:C9:12:94:89:DC:4F:38:8F:6E:16:F7

SHA1：

3C:8A:B0:A4:F1:5C:B5:E6:BE:41:22:1E:67:80:5F:98:4C:05:AF:0E

SHA256：

E7:E8:47:2A:E1:BF:63:F7:A3:F8:D1:A5:E1:A3:4A:47:88:0F:B5:F3:EA:68:3F:5C:D8:BC:0B:BA:3E:C2:D2:61

注意： debug 和 release 版本 app 的 md5, sha256 值如果不一致则需要创建两个应用

3. 开通个验服务

在开发者中心提交 APP 包名等信息后即可获取 APPID 等参数，然后等十分钟左右运营商侧报备成功后就可以正常使用了。[点击前往开发者中心创建应用](#)

已选服务： 个像·用户画像

添加服务至

新建应用

demo

已开通服务：

* 应用名称: demo

* 应用类型: 系统工具

* 应用平台: ☒ Android ☒ iOS

* Android包名: ② ent.com.getui.demo

* Android签名(SHA256): ① 最大长度95个字符

* iOS bundleID: ③ ent.com.getui.demo

上一步

下一步

四、集成SDK

1. maven 接入

1.1 配置maven

在工程根目录build.gradle中的 allprojects 块中添加 maven 仓库地址：

```
buildscript {  
    .....  
}  
  
allprojects {  
    repositories {  
        jcenter()  
  
        maven { url 'https://mvn.getui.com/nexus/content/repositories/releases' }  
    }  
}
```

1.2 配置依赖项

在`app/build.gradle` 中的 `dependencies` 块中添加 `gysdk` 和 `gtc` 的依赖，如下：

```
apply plugin: 'com.android.application'
android {
    .....
}

dependencies {
    //个验sdk
    implementation 'com.getui:gysdk:3.0.1.0'
    //个推公共库，如已接其他个推sdk则保留一个最高版本即可
    implementation 'com.getui:gtc:3.1.7.0'
}
```

2. 配置 APPID和渠道

在 `app/build.gradle` 文件中的 `android.defaultConfig` 下添加 `manifestPlaceholders`

```
defaultConfig {
    applicationId "你的应用包名"
    ....

    manifestPlaceholders = [
        GETUI_APPID      : "你的 GETUI_APPID",
        //渠道若为纯数字则不能超过 int 表示的范围。
        GT_INSTALL_CHANNEL : "你的 GT_INSTALL_CHANNEL"
    ]
}
```

注意: 对于同时集成个推多个产品SDK，且SDK之间的APPID值不一致的用户，可以任选一个SDK的APPID配置到 `GETUI_APPID` 占位符中，其余SDK在AndroidManifest文件中务必添加对应的标签来补充APPID，参考以下配置：

```
<application>
    ....
    <!-- 补充个推SDK的appid -->
    <meta-data
        android:name="PUSH_APPID"
        android:value="个推SDK的appid" />

    <!-- 补充个验SDK的appid -->
    <meta-data
        android:name="GY_APP_ID"
        android:value="个验SDK的appid" />

    <!-- 补充个数SDK的appid -->
```

```

<meta-data
    android:name="GS_APPID"
    android:value="个数SDK的appid" />

<!-- 补充个像SDK的appid -->
<meta-data
    android:name="GI_APPID"
    android:value="个像SDK的appid" />

....
</application>

```

渠道也可通过接口动态设置：

`GYManager.getInstance().setChannel(String channel)`，其优先级高于`GT_INSTALL_CHANNEL`。

3. 配置 so 库

目前个验SDK支持 armeabi、armeabi-v7a、arm64-v8a、x86、x86_64 这几种 CPU 架构，请根据项目情况指定所需的架构。

注意：如果项目中其他 so 库只支持其中某几种 CPU 架构，那么应该根据其他 so 库所支持的 CPU 架构的最小集来配置。否则如果在特定架构上未能支持所有 so 库，则很可能导致程序运行异常。切记！

在 `app/build.gradle` 文件中的 `android.defaultConfig` 下指定所需的 CPU 架构，如下所示：

```

defaultConfig {
    ndk {
        // 注意：这里需要添加项目所需 CPU 类型的最小集
        abiFilters "armeabi", "armeabi-v7a", "x86_64", "x86"
    }
}

```

4. andResGuard配置

如果app启用了资源精简，并且是用老接口 `ELoginThemeConfig` 来定制UI，由gysdk来启动activity的话，请配置资源白名单

```

andResGuard {
    //...
    whiteList = [
        //如果使用ELoginThemeConfig进行UI定制，则设置如下
        "R.drawable.gy_*",

        //记得也保留 ELoginThemeConfig.Builder().setXxx 用到的资源
        //R.drawable.your_elogin_drawables
    ]
}

```

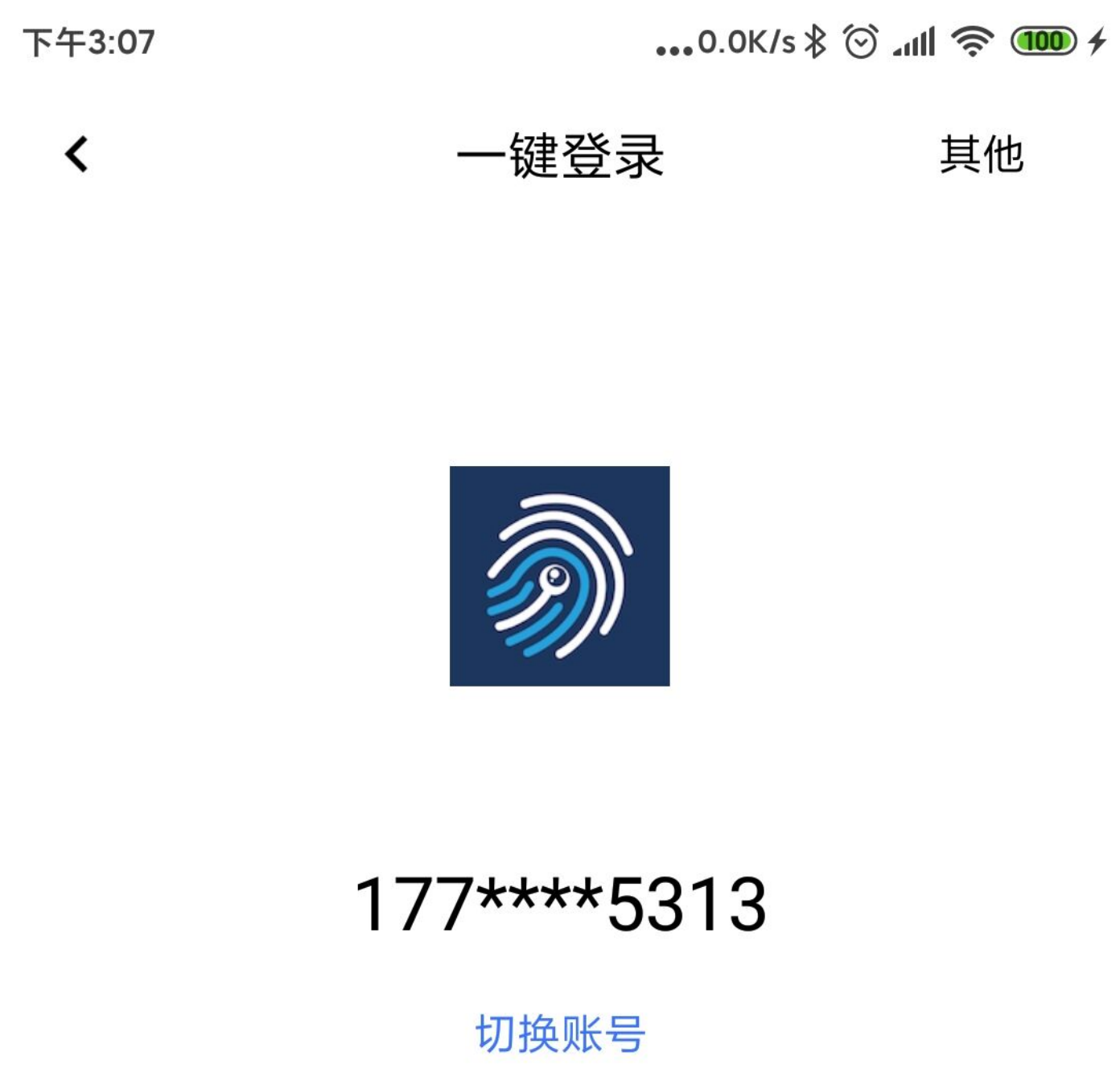
4. 其他配置

- 代码混淆配置
gysdk.aar已经自带proguard.txt， **app**开发者无需手动添加
- 启用http请求
联通sdk需要使用http请求，所以需要在manifest中增加启用http
gysdk.aar的自带manifest中已经增加， **app**开发者无需手动添加

五、授权页规范

为了确保用户在登录过程中将手机号码信息授权给开发者使用的知情权，一键登录需要开发者提供授权页登录页面供用户授权确认。开发者必须弹出授权页，明确告知用户当前操作会将用户的本机号码信息传递给应用。

授权页示例如下：



一键登录

天翼账号提供认证服务

其他方式登录

☒ 登录即认可天翼账号服务与隐私协议和自定义协议1、自定义协议2并使用本机号码登录

注意：

- 1、页面必要元素有：号码栏(*NumberTextview*)，品牌露出(*SloganTextview*)，登录按钮(*LoginButton*)，隐私确认(*PrivacyCheckbox*)，隐私标题(*PrivacyTextview*)；
- 2、开发者不得通过任何技术手段，破解授权页，或将上述授权页面的必要元素内容隐藏、覆盖、或者动态变更；
- 2、登录按钮文字描述必须包含“登录”或“注册”等文字，不得诱导用户授权；
- 3、对于接入GYSDK并上线的应用，我方和运营商会正式上线的应用授权页面做审查，如果有出现未按要求弹出或设计授权页面的，将关闭应用的认证取号服务。

六、简单使用

[下载demo](#)，更换 `app/build.gradle` 中的GETUI_APPID、signingConfigs即可使用；

注意：资料包中的“gy-demo”为新版本推荐的使用示例，“gy-demo-old”为使用老版本接口的示例。

1. 初始化

1.1) 在Application.onCreate()中判断用户已经同意隐私政策后、或者弹出确认框用户同意后初始化gysdk:

```
GYManager.getInstance().init(this.getApplicationContext(), new GyCallBack(){});
```

1.2) 接着提前进行预登录，以便节省后续用户的等待时间，建议超时设置为8s以上，至少5s以上：

```
GYManager.getInstance().ePreLogin(8000, new GyCallBack(){});
```


如果是3.0以前的老版本，初始化接口不带GyCallBack入参，首次预登录必须在广播回调GYReceiver.onGyUidReceived 中进行，否则成功率会非常低，请在本文档开头查找老版本文档

2. 一键登录

2.1) 然后在需要一键登录的时候，先判断预登录：

```
if (GYManager.getInstance().isPreLoginResultValid()) {
    eLogin();//预登录有效，启动登录授权页
} else {
    //考虑到是用户在等待，建议超时5s以上，至少设置3s以上
    GYManager.getInstance().ePreLogin(5000, new GyCallBack() {
        @Override
        public void onSuccess(GYResponse response) {
            eLogin();//预登录成功，启动登录授权页
        }
        @Override
        public void onFailed(GYResponse response) {
        }
    });
};
```

2.2) 预登录有效后或者/进行预登录成功后，启动自行创建的授权登录页面activity

```
private void eLogin() {
    //创建授权页面activity代码，并在manifest中进行注册
    //activity至少包含号码栏(NumberTextview)、品牌露出(SloganTextview)、登录按钮(LoginButton)、隐私确认(PrivacyCheckbox)、隐私标题(PrivacyTextview)。
    Intent intent = new Intent(this, MyELoginActivityFullscreen.class);
    intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    startActivity(intent);
}
```

2.3) 在授权页activity的onCreate中，自行设置隐私协议，然后调用一键登录接口（sdk会设置号码栏、品牌的内容）

```
GyPreloginResult preLoginResult = GYManager.getInstance().getPreLoginResult();
textView.setText("");
textView.append("登录即认可");
textView.append(generateSpan(preLoginResult.getPrivacyName(),
preLoginResult.getPrivacyUrl()));
textView.append("并使用本机号码登录");

EloginActivityParam eloginActivityParam = new EloginActivityParam()
    .setActivity(this)
    .setNumberTextview(numberTv)
    .setNumberTextview(numberTv)
    .setSloganTextview(sloganTv)
    .setLoginButton(loginBtn)
```

```

        .setPrivacyCheckbox(checkBox)
        .setPrivacyTextview(privacyTv)
        .setUiErrorListener(new EloginActivityParam.UIErrorListener() {
            @Override
            public void onError(String msg) {
                //隐私协议未打勾、界面不合规、setLoginOnClickListener抛出异常等情况下的回调
                Log.e(TAG, "UIErrorListener.onError:" + msg);
                hideLoadingDialog();
            }
        })
        .setLoginOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Log.d(TAG, "一键登录按钮 onLoginClick:");
                if (!mCheckBox.isChecked()) {
                    showToast("请先仔细阅读协议并勾选, 然后再点击登录");
                    // 抛出异常, 避免sdk进行后续登录动作 (否则eAccountLogin会回调onFailed错误)
                    throw new IllegalStateException("请先仔细阅读协议并勾选, 然后再点击登录");
                }
                //启动登录时候的转圈圈
                showLoadingDialog();
            }
        });
        GYManager.getInstance().eAccountLogin(eloginActivityParam, 5000, new GyCallBack()
        {}));

```

2.4) eAccountLogin成功回调中解析出token, 调用服务端接口gy_get_pn置换手机号

//具体解析看Api接口中的说明, gy_get_pn请看服务端文档

2.5) 上面启动授权页和预登录的顺序也可根据需要进行对换

//也可根据需要, 先启动授权页activity (此时如果未与登录, 则号码栏、品牌、隐私协议等还无法确定内容只能展示为空),
 //然后在activity中判断预登录有效后/进行预登录成功后, 再设置隐私协议、调用一键登录接口