

技术接入

iOS问题

1. 移动手机号报获取 token 失败: desc = 错误的请求签名, resultCode = 103101

SDK1.7.3 以下版本会出现，可更新SDK或按照下面方法进行解决:

工程中的 Info.plist 中添加 bundleId 对应的 key-value。搜索 Xcode 项目工程中的 .strings 国际化文件，如果 app 存在本地国际化文件，请在每个本地国际化文件中添加 CFBundleIdentifier，各个语言版本文件中都需要添加

2. 预取号的时机

预取号比较耗时，若直接预取号之后再进入授权页面，会影响用户体验。

建议用户提前预取号，可以放到 didFinishLaunchingWithOptions 方法中，因获取运营商信息的时候需要用到 window，建议要放到设置了 window 的 rootViewController 之后

3. 预取号成功后，调用取号方法，取号页面无法弹起

有两种情况可能会导致该问题，请参照下述方案解决： a. 在授权页面，取号成功后，必须调用 SDK 的 dismissAuthViewController 方法关掉授权页面;

b. 由于移动进入授权页面时要检测网络状态，若网络极差则无法进入授权页面

4. SDK 报 8001 错误/服务条款页面加载不出来

ATS未设置白名单，参考下图设置 ATS 白名单，也可以参考官网部署文档：

Key		Type
▼ Information Property List		Dictionary
Localization native development region	↕	String
Bundle display name	↕	String
Executable file	↕	String
▶ Icon files (iOS 5)	↕	Dictionary
▶ CFBundleIcons~ipad	↕	Dictionary
Bundle identifier	↕	String
InfoDictionary version	↕	String
Bundle name	↕	String
Bundle OS Type code	↕	String
Bundle versions string, short	↕	String
Bundle version	↕	String
Application requires iPhone environment	↕	Boolean
▼ App Transport Security Settings	↕	Dictionary
▼ Exception Domains	↕	Dictionary
▼ wap.cmpassport.com		Dictionary
NSExceptionAllowsInsecureHTTPLoads		Boolean
NSAllowsArbitraryLoadsInWebContent		Boolean
▼ id6.me		Dictionary
NSExceptionAllowsInsecureHTTPLoads		Boolean
NSAllowsArbitraryLoadsInWebContent		Boolean
▼ mdn.open.wo.cn		Dictionary
NSExceptionAllowsInsecureHTTPLoads		Boolean
NSAllowsArbitraryLoadsInWebContent		Boolean
▶ onepass.geetest.com		Dictionary
Launch screen interface file base name	↕	String
Main storyboard file base name	↕	String
▶ Required device capabilities	↕	Array
UIRequiresFullScreen	↕	Boolean
▶ Supported interface orientations	↕	Array
▶ Supported interface orientations (iPad)	↕	Array
View controller-based status bar appearance	↕	Boolean

linkerflags

5. 仅支持横屏的 app 在双卡手机上无法获取运营商信息

▼ Deployment Info



Main

Device C

Status

linkerflags

获取运营商信息时需要用到状态栏信息，而只支持横屏的 app 中无法获取状态栏信息。可以通过工程支持竖屏，在对应的 ViewController 中设置为只支持横屏，解决该问题。

6. CocoaPods 无法获取 OneLogin

若在集成极验 SDK 之前就安装了 CocoaPods，安装完成之后，会在本地有一份缓存，缓存中并没有 OneLogin，可通过下述步骤解决：
a. 删掉缓存的索引，使用 `pod setup` 重新建立索引即可；
b. 使用 `pod install --repo-update` 安装。

7. 自定义协议点击无响应

可能是下面两种情况导致，可参照排查：
a. 初始化协议时，对应的 block 没有传 nil，但是在 block 中又没做任何操作；
b. 多个协议使用了相同的 URL

8. 在授权页面，当前流量卡为电信时，点击一键登录报 40399

1.8.0以下版本会出现该问题，可更新SDK或按照下面方法进行解决：
app 内部存在多个 window 的情况时，授权页面要放到 keyWindow

9. 联通预取号报 -40201 错误 用户修改了手机时间，手机调为正常时间即可。若当前时间恢复正常后依然报错，检查下当前网络环境

10. iOS 13 双卡预取号失败

1.7.3 之前的版本不支持 iOS 13 双卡，请升级到 1.8.0或以上 版本

11. - (BOOL)application:(UIApplication*)application didFinishLaunchingWithOptions:(NSDictionary*)launchOptions 方法中(程序启动时)预取号失败

预取号方法会先调用系统 API 检测当前手机流量对应的运营商，而在程序刚刚启动时，该检测方法往往会检测不到正确的运营商。

可以把预取号的方法放到 - (BOOL)application:(UIApplication*)application didFinishLaunchingWithOptions:(NSDictionary*)launchOptions 方法的最后面执行,或者稍微延迟一下去执行预取号方法

12. 联通 SDK 预取号始终超时

可能是联通对区域做了限速，或者设置的超时时间过短，建议超时时间设置5s左右进行调试。

13. push 方式进入授权页面或者服务条款页面

OneLogin 授权页面和服务条款页面不是 UINavigationController，不支持 push 方式进入，默认是通过 presentViewController 方法进入的，可以通过添加动画来模拟 push 的方式进入，代码如下：

```
    CATransition *animation = [CATransition animation];
    animation.duration = 0.5;
    animation.timingFunction = [CAMediaTimingFunction functionWithName:kCAMediaTimingFunctionEaseInEaseOut];
    animation.type = kCATransitionPush;
    animation.subtype = kCATransitionFromRight;

    CustomProtocolViewController *customProtocolController = [CustomProtocolViewController new];
    [[OneLogin currentAuthViewController].view.window.layer addAnimation:animation forKey:nil];
    customProtocolController.modalPresentationStyle = UIModalPresentationFullScreen;
    [[OneLogin currentAuthViewController] presentViewController:customProtocolController animated:NO completion:nil];
```

14. 授权页面卡死

通过 presentViewController 方法进入授权页面时，默认的 UIModalPresentationStyle 是 UIModalPresentationFullScreen，在某些场景下，进入到授权页面后，调用 dismissViewControllerAnimated 方法关闭授权页面时，会出现页面不消失的情况，然后实际上，授权页面已经被释放掉了，故此时点击授权页面上的任何控件均无响应，造成页面卡死现象，判断为 iOS 系统 bug，解决方案是，将 UIModalPresentationFullScreen 改为 UIModalPresentationOverFullScreen 即可

15. preGetTokenWithCompletion 方法回调中使用 if ([status isEqualToNumber:@(200)]) 方法判断预取号是否成功导致 crash

不能用该方法判断预取号是否成功，正确的判断方法如下：

```
// 预取号成功
if (result.count > 0 && result[@"status"] && 200 == [result[@"status"] integerValue]) {
}
}
```

16. 隐私条款是否支持设置距离屏幕底部偏移

不支持设置距离屏幕底部偏移，只支持设置距离屏幕顶部偏移，请根据屏幕尺寸，以 375*667 为基准，给偏移量乘一个比例即可

17. OneLogin 如何判断预取号是否完成？

进阶模式下：

调用 registerWithAppID 方法后 SDK 内部会自动进行预取号和失败重试、超时重新获取等动作，开发者无需关心预取号是否完成。开发者可在调用 + (void)requestTokenWithViewController:(nullable UIViewController *)viewController viewModel:(nullable OLAAuthViewModel *)viewModel completion:(void (^)(NSDictionary * _Nullable result))completion 之前调用 isPreGetTokenResultValidate 接口判断预取号是否有效，无效时可先加载自定义 loading 等待对话框。

常规模式下：

调用预取号 preGetTokenWithCompletion 方法后，收到 completion 回调表示预取号结束，预取号是否成功可根据返回结果区分；未收到回调表示预取号还未完成。

18. 自定义服务条款时出现服务条款无法点击的情况

使用 - (instancetype)initWithTitle:(NSString *)title linkURL:(NSURL *)url index:(NSInteger)index block:(OLViewPrivacyTermItemBlock _Nullable)block; 方法初始化 termItem 时，实现了 block，但是 block 中的方法为空。解决方案：
a、block 传 nil；
b、在 block 中实现自定义跳转

19. 使用弹窗模式时，弹起弹窗或者关闭弹窗时出现闪屏或者阴影

使用弹窗模式时，pullAuthVCStyle 一定要为 OLPullAuthVCStyleModal，+(void)requestTokenWithViewController:(nullable UIViewController *)viewController viewModel:(nullable OLAAuthViewModel *)viewModel completion:(void (^)(NSDictionary * _Nullable result))completion 方法中的 viewController 需要传 navigationController：

```
[OneLoginPro requestTokenWithViewController:self.navigationController viewModel:self.olAuthViewModel
completion:^(NSDictionary * _Nullable result) {
}
}];
```

20. 如何判断当前手机是否为双卡手机？

```

- (BOOL)hasDoubleSIMCard {
    CTTelephonyNetworkInfo *telephonyInfo = [[CTTelephonyNetworkInfo alloc] init];
    if (@available(iOS 12.0, *)) {
        NSDictionary *cellularProviders = nil;
        if ([telephonyInfo isKindOfClass:[CTTelephonyNetworkInfo class]] && [telephonyInfo
respondsToSelector:@selector(serviceSubscriberCellularProviders)]) {
            cellularProviders = telephonyInfo.serviceSubscriberCellularProviders;
        }
        if (cellularProviders.count > 1) {
            return YES;
        }
    }
    return NO;
}

```

21. 如何自定义服务条款及运营商服务条款的位置？

自定义服务条款和运营商服务条款，都有一个 index 属性，值默认为 0，在添加自定义服务条款时，可通过设置 index 值来确定服务条款的位置，若不设置，则默认运营商服务条款在最前面，自定义服务条款按添加顺序排列。

```

/**
 条款索引，默认为0，当有多条条款时，会根据此属性升序排列条款
 */
@property (nonatomic, assign) NSInteger index;

- (instancetype)initWithTitle:(NSString *)title linkURL:(NSURL *)url;
- (instancetype)initWithTitle:(NSString *)title linkURL:(NSURL *)url index:(NSInteger)index;
- (instancetype)initWithTitle:(NSString *)title linkURL:(NSURL *)url index:(NSInteger)index block:
(OLViewPrivacyTermItemBlock _Nullable)block;
- (instancetype)initWithTitle:(NSString *)title urlRequest:(NSURLRequest *)urlRequest index:(NSInteger)index block:
(OLViewPrivacyTermItemBlock _Nullable)block;

```

假如有三个自定义服务条款，需要的服务条款顺序为：自定义服务条款1 -> 运营商服务条款 -> 自定义服务条款2 -> 自定义服务条款3，则可通过如下代码进行设置：

```

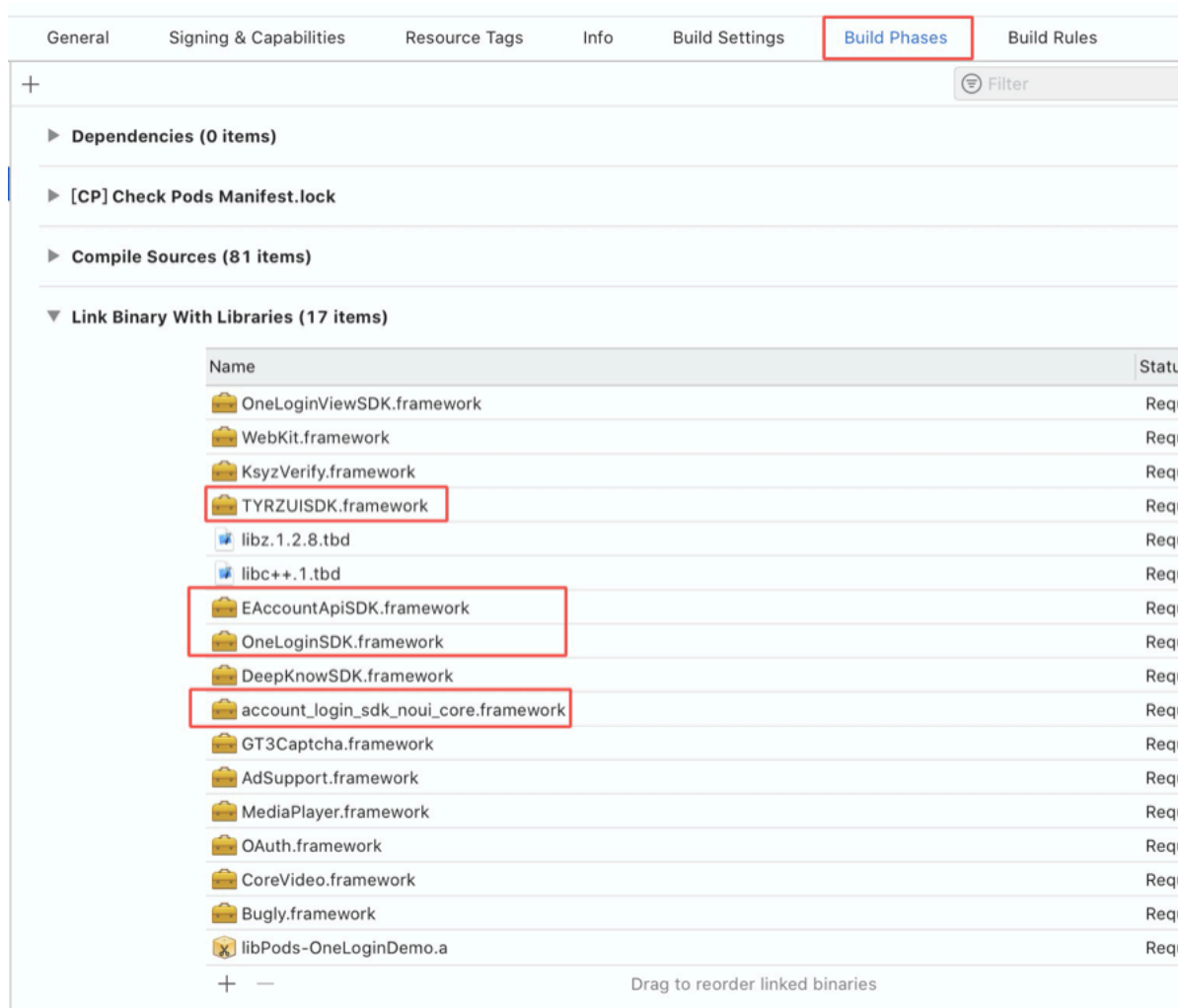
// 额外自定义服务条款，注意index属性，默认的index为0，SDK会根据index对多条服务条款升序排列，假如想设置服务条款顺序为 自定义服务条款1 默认服务条款 自定义服务条款2，则，只需将自定义服务条款1的index设为-1，自定义服务条款2的index设为1即可
OLPrivacyTermItem *item1 = [[OLPrivacyTermItem alloc] initWithTitle:@"自定义服务条款1"
                                linkURL:[NSURL URLWithString:@"https://www.baidu.com"]
                                index:-1
                                block:^(OLPrivacyTermItem * _Nonnull termItem,
                                        UIViewController *controller) {
                                    NSLog(@"termItem.termLink: %@", termItem.termLink);
                                    // 自定义操作，可进入自定义服务条款页面
                                    dispatch_async(dispatch_get_main_queue(), ^{
                                        GTDebugViewController *debugController =
[GTDebugViewController new];
                                        debugController.modalPresentationStyle =
UIModalPresentationFullScreen;
                                        [controller presentViewController:debugController
animated:YES completion:nil];
                                    });
                                });
OLPrivacyTermItem *item2 = [[OLPrivacyTermItem alloc] initWithTitle:@"自定义服务条款2"
                                linkURL:[NSURL URLWithString:@"https://docs.geetest.com/"]
                                index:0];
NSURL *URL = [[NSBundle mainBundle] URLForResource:@"index.html" withExtension:nil];
NSURLRequest *URLRequest = [NSURLRequest requestWithURL:URL];
OLPrivacyTermItem *item3 = [[OLPrivacyTermItem alloc] initWithTitle:@"自定义服务条款3"
                                urlRequest:URLRequest
                                index:0
                                block:nil];
viewModel.additionalPrivacyTerms = @[item1, item2, item3];

```

22. 收到 -20407、-20408、-20409 错误时如何排查？

-20407 - 移动 SDK TYRZUISDK.framework 有误
-20408 - 联通 SDK account_login_sdk_noui_core.framework 有误
-20409 - 移动 SDK EAccountApiSDK.framework 有误

这几个错误表示 OneLoginSDK 找不到依赖的运营商 SDK 或者使用的运营商 SDK 版本与 OneLoginSDK 的版本不匹配，请到极验官网下载最新的 SDK，并确保 Xcode -> Build Phases -> Link Binary With Libraries 中有添加 TYRZUISDK.framework、account_login_sdk_noui_core.framework 和 EAccountApiSDK.framework，如下图所示：



linkerflags

23. OneLogin iOS 授权页面元素缺失，图片显示失效

授权页面能弹出，但是未显示和图片有关的界面元素，可能是 OneLoginResource.bundle 未正确导入到工程导致。手动导入时，确认勾选 Copy items if need。

24. OneLoginSDK iOS 集成时，Xcode 编译报 `_nw_` 相关的符号无法找到 (symbol not found)

用户未在使用 2.5.2 的 OneLogin iOS SDK 未引入系统的 `Network.framework`，这在新版本的移动运营商中对需要支持 iOS 12 以下的设备时被强制要求。

Xcode 编译错误提示 `symbol not found`，说明相关的符号无法被链接器访问到。排查方式：先确认缺失库的名称，再确认相关的库是是否正确导入（手动导入时勾选 Copy items if need 等）。否则可以尝试重启 Xcode 或者系统。

25. OneLoginSDK iOS 集成时，Xcode 编译报符号重复 (Duplicated symbol)

如果 Xcode 提示 `Duplicated symbol`，说明有符号冲突（代码冲突）。排查方式：确实冲突的符号所属的库，在确认不同的库怎么被同时引入。OneLogin 中一般是运营商的 SDK 重复导致，确认重复的库的版本一致后，剔除其中一个即可。

26. OneLoginSDK iOS 集成时，Xcode 编译报 `No such module 'OneLoginSDK'`

请检查 OneLoginSDK.framework 是否正确被导入工程，且 Xcode 的 Build Setting 中的 Search Path 进行了正确的配置。

27. 使用 Xcode 14.3 运行项目时报链接不到 ARC 库的错

原因是 Xcode 14.3 移除了 ARC 相关的库，iOS 9.0 及以上系统内置了 ARC 库，所以有两种解决方案：

- 可以选择将项目目标最低版本设置为 9.0 或以上
- 将在 14.3 以下的 Xcode 版本中的 ARC 库拷贝到 Xcode 14.3 版本对应路径上

Android问题

1. 全机型，全系统兼容吗？

本产品依赖第三方运营商 SDK，移动、电信最低可兼容 Android 4.0，联通 SDK 对 4.x 系统不保证取号成功率，因此最低可支持 Android 4.0，失败走降级方式。Android 6.0 因部分版本系统 bug 原因，WIFI+4G 情况下可能存在切换数据网络失败的情况，建议走降级方式。

2. 只有一个回调，该怎样处理页面切换等逻辑呢？

页面的所有状态都有错误码输出，可以根据错误码来实现。

3. 怎样控制关闭授权页的时机？

在 requestToken() 方法执行完成之后，根据错误码来自主实现关闭授权页的时机，当请求成功之后需要页面跳转的时候是关闭授权页的时机。开发者自定义控件的事件发生时也是关闭授权页的时机。

4. 授权页面的沉浸式状态栏或者设置状态栏的颜色怎么实现呢？

在 0.6.1 版本之后，可以通过 setStatusBar 方法设置。

5. 怎样实现横屏呢？

一键登录授权页通过 screenOrientation="behind" 主题配置跟随前一个 Activity 方向，在需要登录的前一个页面通过主题配置横屏 Activity，或者在页面初始化时调用 setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE) 配置横屏，拉起的授权页即为横屏样式。

6. 怎样设置主题？

0.9.0 版本之后，SDK 内部 AndroidManifest 默认提供了 授权页 与 隐私页 Activity 的声明及默认主题配置。自定义主题时需要在用户自己的清单文件中添加 activity 的相关配置，如下代码片段所示。针对多个需要覆盖的配置，tools:replace 后面可填多个配置项，通过西文逗号，分隔即可。

```
<activity
    android:name="com.geetest.onelogin.activity.OneLoginActivity"
    tools:replace="android:theme"
    android:theme="@style/Theme.ActivityDialogStyle" />
<activity
    android:name="com.geetest.onelogin.activity.OneLoginWebActivity"
    tools:replace="android:theme"
    android:theme="@style/Theme.ActivityDialogStyle" />
```

Dialog实现的弹窗样式授权页的主题可以直接在app中配置，主题名称相同即可覆盖SDK配置的主题，如下是默认Dialog样式。

```
<style name="GtOneLoginDialogTheme" parent="android:Theme.Dialog">
    <!-- 除去title-->
    <item name="android:windowNoTitle">true</item>
    <!-- 显示区域以外是否使用黑色半透明背景-->
    <item name="android:backgroundDimEnabled">true</item>
    <!-- 设置dialog的背景 默认背景是个点九图，会导致底部浮窗不能铺满-->
    <item name="android:windowBackground">@android:color/transparent</item>
</style>
```

7. 怎样设置弹窗形式？

在 0.6.1 版本之后，可以通过 setDialogTheme 方法设置。0.9.0 版本之后，如果不改theme，则只需要这一步即可。

8. 授权页面弹框背景为黑色，如何设置为透明？

授权页面以对话框形式弹出时，支持通过主题设置背景样式。未设置主题的情况下，默认背景为黑色，可参考以下方式设置为透明背景：

在主题配置文件 styles.xml 中增加对话框主题样式：

```
<style name="Theme.ActivityDialogStyle" parent="Theme.AppCompat.Light.NoActionBar"> <!-- 背景透明-->
    <item name="android:windowBackground">@android:color/transparent</item>
    <item name="android:windowIsTranslucent">true</item> <!-- dialog的整个屏幕的背景是否有遮罩层-->
    <item name="android:backgroundDimEnabled">true</item>
</style>
```

在 AndroidManifest.xml 中给授权页面应用主题样式（参考问题6）：

```
<activity
    android:name="com.geetest.onelogin.activity.OneLoginActivity"
    tools:replace="android:theme"
    android:theme="@style/Theme.ActivityDialogStyle" />
```

9. 从授权页返回，为什么会出现闪屏或者黑屏？

授权页默认主题配置了 windowIsTranslucent=true 配置，会对 Activity 的切换动画产生一些负面影响，比如部分手机上出现的闪屏或者黑屏。

闪屏现象：当在进入授权页后延时关闭前一个页面时，部分手机上可能会出现前一个页面短暂弹出到前台然后又消失的现象，也就是闪屏现象；针对这问题开发者可以考虑通过自定义授权页主题覆盖掉默认配置的方式解决，覆盖方式参考问题6。

黑屏现象：当进入授权页之前弹出一个过渡Activity用于加载进度时，从授权页返回时同时关闭过渡Activity与授权页，相当于同时关闭两个Activity，部分机型也可能出现短暂黑屏的现象。针对于这问题开发者可以考虑弹出自定义对话框的形式加载进度，授权页弹出时关闭进度框，即可解决黑屏问题。

10. 一键登录授权页如何配置进入退出动画？

授权页转场动画修改，可以通过主题配置动画（进入退出动画）样式，主题修改可参考 **FAQ:6**, 动画配置示例如下：

```
<style name="Theme.ActivityDialogStyle" parent="Theme.AppCompat.Light.NoActionBar">
    ...
    <item name="android:windowAnimationStyle">@style/AnimationActivity</item>
    ...
</style>
<style name="AnimationActivity" parent="Theme.AppCompat.Light.NoActionBar">
    <item name="android:activityOpenEnterAnimation">@anim/push_left_in</item>
    <item name="android:activityOpenExitAnimation">@anim/push_left_out</item>
    <item name="android:activityCloseEnterAnimation">@anim/push_right_in</item>
    <item name="android:activityCloseExitAnimation">@anim/push_right_out</item>
</style>
```

SDK会自动在授权页 finish() 时调用出场动画，入场动画需要在 actionAfterStartActivityCalled 回调中找到当前Activity调用 overridePendingTransition 。

```
@Override
public void actionAfterStartActivityCalled() {
    Activity mainActivity = activityWeakReference.get(); // 弱引用，防止内存泄漏
    if (mainActivity != null) {
        mainActivity.overridePendingTransition(R.anim.push_left_in, R.anim.push_left_out);
    }
}
```

11. OneLoginThemeConfig 使用注意事项？

1. 配置项的图片，地址都应为 drawable 目录下。
2. OffsetY 与 OffsetY_B 两者必须有一个值为 0，偏移量以不为 0 的方向作为基准。

12. 如何设置授权页面的 Logo？

在 0.6.1 版本之后，可以通过 setLogoImageView 方法设置。

13. 怎样隐藏返回图标？

在 0.6.1 版本之后，可以通过 setAuthNavReturnImageView 方法设置。

14. 怎样实现第三方登录的自定义？

在Demo中有详细的方法展示，在这里展示部分代码。


```
private void initLogin() {
    LayoutInflater inflater1 = LayoutInflater.from(context);
    RelativeLayout relativeLayout = (RelativeLayout) inflater1.inflate(R.layout.relative_item_view, null);
    RelativeLayout.LayoutParams layoutParamsOther = new RelativeLayout.LayoutParams(RelativeLayout.LayoutParams.WRAP_CONTENT,
        RelativeLayout.LayoutParams.WRAP_CONTENT);
    layoutParamsOther.setMargins(0, dip2px(context, 430), 0, 0);
    layoutParamsOther.addRule(RelativeLayout.CENTER_HORIZONTAL);
    relativeLayout.setLayoutParams(layoutParamsOther);
    ImageView weixin = relativeLayout.findViewById(R.id.weixin);
    ImageView qq = relativeLayout.findViewById(R.id.qq);
    ImageView weibo = relativeLayout.findViewById(R.id.weibo);
    weixin.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Toast.makeText(context, "微信登录", Toast.LENGTH_SHORT).show();
        }
    });
    qq.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Toast.makeText(context, "qq登录", Toast.LENGTH_SHORT).show();
        }
    });
    weibo.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Toast.makeText(context, "微博登录", Toast.LENGTH_SHORT).show();
        }
    });
    OneLoginHelper.with().addOneLoginRegisterViewConfig("title_button", new AuthRegisterViewConfig.Builder()
        .setView(relativeLayout)
        .setRootViewId(AuthRegisterViewConfig.RootViewId.ROOT_VIEW_ID_BODY)
        .setCustomInterface(new CustomInterface() {
            @Override
            public void onClick(Context context) {
                Toast.makeText(context, "动态注册的其他按钮", Toast.LENGTH_SHORT).show();
            }
        })
        .build()
    );
}
```

15. 授权页标题栏返回按钮、登录按钮、切换账号如何设置圆角背景图片？

标题栏返回按钮、登录按钮和切换账号均支持背景配置，支持静态图片与 xml 资源，静态图片需要预先切成圆角的图片，也支持 .9 图片；xml 资源通过自定义 shape，指定 corners 属性配置的圆角背景实现。

16. 如何实现授权页登录按钮点击动态效果？

标题栏登录按钮支持配置 xml 背景资源，动态背景图片可通过配置 selector 格式的 xml 资源来实现，参考以下示例：

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <!-- 按压时 -->
    <item
        android:state_pressed="true"
        android:drawable="@drawable/button_bg_pressed"/>
    <!-- 可用 -->
    <item
        android:state_enabled="true"
        android:drawable="@drawable/button_bg_normal"/>
    <!-- 默认 -->
    <item android:drawable="@drawable/button_bg_normal"/>
</selector>
```

17. 如何实现授权页登录按钮在未勾选选择框时处于禁用(不可用、不可点击)状态？

同上配置 selector 格式的 xml 资源来实现随状态自动切换的背景资源，然后通过 setLogBtnDisableIfUnChecked(true) 配置登录按钮在未勾选选择框时处于禁用(不可用、不可点击)状态。拉起授权页后，如果选择框为勾选状态时，按钮处于可用状态；反之不可用。

18. 如何自定义未勾选选择框时 Toast 样式？

目前 SDK 使用的 Toast 应用的系统 Toast 默认样式，不支持定制修改位置，背景等。如有修改需求，可以参照以下步骤自定义 Toast：

1. setPrivacyUnCheckedToastText 接口第一个参数（重载方法 2.3.1.3 版本才支持）传 false 禁用默认 Toast
2. 在 onLoginButtonClick 回调里面调用 OneLoginHelper.with().isPrivacyChecked() 判断是否勾选
3. 如果未勾选时应用内弹出自定义的 Toast

19. 如何隐藏隐私栏选择框？

1. 使用透明背景图片使选择框隐藏
2. 为了减少间隙，也可配置1像素大小透明背景图片。

以上两种方法都可实现隐藏效果，因涉及到用户隐私授权，需在授权页面添加“登录即同意xxx条款并使用一键登录”等类似文字说明。

20. OneLogin 授权页一键登录能否使用自定义对话框式 loading?

默认 loading 是显示在一键登录按钮上的，如果需要自定义弹出对话框式 loading，可按如下步骤实现：

1. 在调 `setLogBtnLoadingView` 进行授权页配置 loading 图片的时，第一个参数传入空值
2. 授权页拉起后会有一个 `onAuthActivityCreate` 回调，在这里您可以初始化自己的自定义 loading 对话框，注意携带授权页的 `activity` 实例作为对话框的 `context`，或者临时持有这个 `activity` 引用
3. 授权页用户点击一键登录后会有一个 `onLoginLoading` 回调，在这里就可以进行自定义 loading 对话框的 `show` 操作
4. 收到 `onResult` 回调后关闭 loading 对话框

21. 授权页切换账号文案字数是否有限制？

无限制，当前切换账号支持背景配置，默认背景宽度下仅可容纳默认字体下的四个中文汉字，如需配置更多字体或者调大字号，建议通过 `setSwitchViewLayout` 修改背景大小适配。

22. OneLogin 如何判断预取号是否完成？

进阶模式下：

调用 `register` 方法后 SDK 内部会自动进行预取号和失败重试、超时重新获取等动作，开发者无需关心预取号是否完成。开发者可在调用 `requestToken` 之前调用 `isPreGetTokenResultValidate` 接口判断预取号是否有效，无效时可先加载自定义 loading 等待对话框。

常规模式下：

调用预取号 `preGetToken` 方法后，收到 `onResult` 回调表示预取号结束，预取号是否成功可根据返回结果区分；未收到回调表示预取号还未完成。

23. 为什么会提示“模式错误，请不要新老逻辑混用”？

进阶模式和常规模式是一键登录的两种调用逻辑，内部的预取号重试机制不同，为了避免重试机制异常，不可以混用两种逻辑。Android 端 `register` 为进阶逻辑的预取号接口，`preGetToken` 为常规逻辑的预取号接口，请勿同时使用。

24. 为什么电信手机卡取号会报80102错误码，提示预登录异常？

电信 SDK 内部方法通过 native 实现，历史版本迭代过程中 native 方法有变更，如果 so 跟 SDK 版本不匹配或者缺失就可能导致预登录异常。我们提供的 SDK 为 aar 文件，内部集成了运营商 SDK 与必要的资源文件，建议集成时不要解压集成。SDK 默认提供了 `'armeabi'`，`'armeabi-v7a'`，`'arm64-v8a'`，`'x86'`，`'x86_64'` 五个架构下的 so，可以覆盖绝大部分移动设备的架构支持，建议项目的 `abiFilters` 至少保留 arm 三个架构。

25. 为什么有些场景点一键登录只有 onLoginButtonClick 回调，但不能取号？

一键登录 SDK 只支持同一时间拉起一个授权页，为了避免重复拉起，导致内部统计混乱，当前 SDK 限制了同时拉起多个授权页，具体表现为第二次拉起的授权页一键登录按钮点击不会进行取号。如果切换登录方式，建议在第二次拉起授权页之前关闭前一个授权页。

26. READ_PHONE_STATE 权限是否必须？

非必须。Android Q 以下版本系统用于在双卡手机上，SDK 内更准确的识别开启数据流量卡的运营商类型，未开启该权限可能导致部分手机取号失败的风险。

27. 一键登录 SDK 能否在子线程中初始化？

可以，SDK 已兼容子线程初始化 SDK 能力。但考虑到 SDK 本身对外接口比较轻量，所有耗时操作与网络请求皆在子线程中进行，建议在主线程中初始化 SDK 即可。

28. 一键登录通过自定义控件跳转第三方登录后，如何接收第三方登录返回的 onActivityResult 回调？

- 方案一：三方登录结束时回到一键登录授权页

1. 在自定义控件的点击回调里用一键登录授权页 `OneLoginActivity` 的引用（记为 `oneLoginActivity`，下同）的 `Context` 来调用 `startActivityForResult` 实现跳转功能：

```
// 自定义授权页三方登录控件，在控件点击回调里跳转三方登录
relativeLayout.findViewById(R.id.weixin).setOnClickListener(v -> {
    // context 为一键登录授权页 的 Context(即 OneLoginActivity 的 Context)
    oneLoginActivity.startActivityForResult(new Intent(oneLoginActivity, ThirdLoginActivity.class),
        Constants.THIRD_LOGIN_CODE);
});
```

2. 跳转第三方登录，登录完成后返回结果，以下为三方登录示例（实际以接入第三方登录为准）：

```
// 模拟登录成功
private void loginSuccess() {
    Intent intent = new Intent();
    intent.putExtra(Constants.THIRD_LOGIN_EXTRA_DATA, "100");
    setResult(RESULT_OK, intent);
    finish();
}

// 模拟登录失败
private void loginFailure() {
    Intent intent = new Intent();
    intent.putExtra(Constants.THIRD_LOGIN_EXTRA_DATA, "500");
    setResult(RESULT_CANCELED, intent);
    finish();
}
```

- 在调用 requestToken 拉起一键登录授权页时传参 AbstractOneLoginListener 的实现中覆写 onAuthActivityResult 方法，来接收三方登录的 onActivityResult 回调并解析返回参数：

```
@Override
public void onAuthActivityResult(int requestCode, int resultCode, Intent data) {
    super.onAuthActivityResult(requestCode, resultCode, data);
    if (requestCode == Constants.THIRD_LOGIN_CODE && data != null && data.hasExtra(Constants.THIRD_LOGIN_EXTRA_DATA)) {
        String loginResult = data.getStringExtra(Constants.THIRD_LOGIN_EXTRA_DATA);
        Toast.makeText(context, "LoginResult: " + loginResult, Toast.LENGTH_LONG).show();
        // 解析返回参数，处理登录结果
        // 如成功则关闭一键登录授权页，更新登录状态，或者跳转登录成功后的页面
        // 失败时，可继续停留在一键登录授权页面，可继续登录
    }
}
```

- 方案二：跳转三方登录时关闭一键登录授权页

- 在自定义控件的点击回调里用需要接收回调的 Activity（记为 ActivityA，下同）的 Context 来调用 startActivityForResult 实现跳转功能：

```
// 自定义授权页三方登录控件，在控件点击回调里跳转第三方登录
RelativeLayout.findViewById(R.id.weixin).setOnClickListener(v -> {
    // 跳转时关闭一键登录授权页，这样三方登录返回结果后直接返回 ActivityA，
    // 否则需要等关闭一键登录授权页后才能收到 onActivityResult 回调
    OneLoginHelper.with().dismissAuthActivity();
    // context 为需要接收回调的 Activity 的 Context(即 ActivityA 的 Context)
    context.startActivityForResult(new Intent(context, ThirdLoginActivity.class), Constants.THIRD_LOGIN_CODE);
});
```

- 跳转第三方登录，登录完成后返回结果，以下为三方登录示例（实际以接入第三方登录为准）：

```
// 模拟登录成功
private void loginSuccess() {
    Intent intent = new Intent();
    intent.putExtra(Constants.THIRD_LOGIN_EXTRA_DATA, "100");
    setResult(RESULT_OK, intent);
    finish();
}

// 模拟登录失败
private void loginFailure() {
    Intent intent = new Intent();
    intent.putExtra(Constants.THIRD_LOGIN_EXTRA_DATA, "500");
    setResult(RESULT_CANCELED, intent);
    finish();
}
```

- 在 ActivityA 中实现 onActivityResult 回调并解析返回参数：

```
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == Constants.THIRD_LOGIN_CODE && data != null && data.hasExtra(Constants.THIRD_LOGIN_EXTRA_DATA)) {
        String loginResult = data.getStringExtra(Constants.THIRD_LOGIN_EXTRA_DATA);
        Toast.makeText(context, "LoginResult: " + loginResult, Toast.LENGTH_LONG).show();
    }
}
```

29. 一键登录隐私页能否使用自己的页面(Activity)?

可以，请参考以下步骤进行自定义隐私页：

- 在进行一键登录主题配置时，setPrivacyLayout 接口最后一个参数 isUseNormalWebActivity 传 false
- 在拉起授权页方法 requestToken 的回调实现中覆写(Override) onPrivacyClick(String name, String url) 方法，实现跳转到自定义的页面加载指定的隐私页内容

30. OneLogin Android 首次进入授权页无法正确获取脱敏手机号

如果客户希望在号码栏展示类似“本机号码：156****1234”（默认为仅显示脱敏手机号“156****1234”）这样的文本，那么需要在构建 OneLoginThemeConfig 时调用 setNumberText(CharSequence numberText) 设置号码栏的文本，这个文本中必须包含正确的脱敏手机号，获得脱敏手机号的方式如下：

```
OneLoginHelper.with().requestSecurityPhone(new SecurityPhoneListener{  
    @Override  
    public void onSuccess(String phone, String operator) {  
        //phone即为脱敏手机号  
    }  
  
    @Override  
    public void onFailed(JSONObject jsonObject) {  
        //预取号失败，按错误码弹出提示或者走降级流程  
    }  
})
```

也可以通过如下方式：

```
if (OneLoginHelper.with().isPreGetTokenResultValidate()) {  
    String phone = OneLoginHelper.with().getSecurityPhone();  
    //phone即为脱敏手机号  
} else {  
    OneLoginHelper.with().requestSecurityPhone(new SecurityPhoneListener() {  
        @Override  
        public void onSuccess(String phone, String operator) {  
            //phone即为脱敏手机号  
        }  
  
        @Override  
        public void onFailed(JSONObject jsonObject) {  
            //预取号失败，按错误码弹出提示或者走降级流程  
        }  
    });  
}
```

31. OneLogin Android 资源文件找不到

可能是资源被二次混淆，需要在配置中对 OneLogin 的资源设置混淆白名单。

32. Android 集成使用 .9 图设置图片时，表现异常

通过 getIdentifier 获取 ID 时，名称需要全称

33.Dialog实现的弹窗样式授权页与Activity模拟的弹窗样式授权页有何区别？

Activity模拟的授权页是Activity实现，设置底部弹窗样式可以控制背景是否侵入导航栏，并支持设置导航栏颜色，但该场景下导航栏按钮的颜色风格不能很好地受控。Dialog实现的授权页由系统自动控制授权页的布局不延伸到导航栏，且不会影响弹出授权页的Activity生命周期。二者都能实现相同的布局效果，按需选择。

34.弹出Dialog实现的授权页时如何隐藏导航栏？

SDK提供三个Dialog授权页弹出的相关回调，可在其中修改window的flag。以下代码供参考：

```

/**
 * Dialog 实现的授权页在 void onWindowFocusChanged(boolean hasFocus) 时间调
 */
@Override
public void onAuthDialogFocusChanged(Dialog dialog, boolean hasFocus) {

}

/**
 * Dialog 实现的授权页在 show() 方法执行前回调
 */
@Override
public void actionBeforeAuthDialogShow(Dialog dialog) {
    dialog.getWindow().setFlags(WindowManager.LayoutParams.FLAG_NOT_FOCUSABLE,
        WindowManager.LayoutParams.FLAG_NOT_FOCUSABLE);
}

/**
 * Dialog 实现的授权页在 show() 方法执行后回调
 */
@Override
public void actionAfterAuthDialogShow(Dialog dialog) {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT) {
        int uiOptions = View.SYSTEM_UI_FLAG_LAYOUT_STABLE
            | View.SYSTEM_UI_FLAG_LAYOUT_HIDE_NAVIGATION
            | View.SYSTEM_UI_FLAG_HIDE_NAVIGATION
            | View.SYSTEM_UI_FLAG_IMMERSIVE_STICKY
            | View.SYSTEM_UI_FLAG_LAYOUT_FULLSCREEN
            | View.SYSTEM_UI_FLAG_FULLSCREEN;
        dialog.getWindow().getDecorView().setSystemUiVisibility(uiOptions);
    }
    dialog.getWindow().clearFlags(WindowManager.LayoutParams.FLAG_NOT_FOCUSABLE);
}
}

```

35.预取号成功后，点击登录按钮却没有相关onResult回调？

SDK提供了添加自定义控件的方法 `OneLoginHelper.with().addOneLoginRegisterViewConfig(id, authRegisterViewConfig)`。如果调用了该方法，请注意自定义的控件不允许覆盖 SDK 默认的 UI。授权页的登录逻辑依赖于SDK原有的UI组件，被覆盖将会导致 `requestToken` 没有 `onResult` 回调。

H5 问题

1. OnePass API H5 中checkNetInfo接口不支持iOS，iOS如何处理？

`checkNetInfo` 接口在iOS上获取到的结果不准确，这个接口只是一个辅助判断，对验证不影响，失败后可走降级验证。

2. H5移动手机号突然无法进行验证？

移动运营商限制，同一个号码每天只能校验10次。

3. H5产品是否支持嵌入iframe中使用？

h5产品在V1.2.7之前版本支持在iframe中嵌套使用，V1.2.7之后的版本因为电信运营商原因 目前不支持在iframe中嵌套使用。

4. H5本机手机号的运营商协议如何添加？

OnePass API产品没有授权页面，不需要添加运营商条款的。

5. OnePass API 小程序产品，联通手机无法登录？

目前只支持电信和移动的运营商，联通的目前不支持的。

6. 移动运营商报错“无效的请求-referer校验失败”？

生产上有https访问的，会导致上报的referer为空，需在head中添加代码 `<meta name="referrer" content="unsafe-url">`。

7. h5产品是否支持在app中使用？

支持在原生app中使用。

8. 部署时报错GOP is not defined？

js未正常引入，建议进行检查是否正确引入js。

9. 移动接口报错“无效的请求-origin效验失败”？

后台查看当前页面使用的APP ID配置中的origin 是否和接入页面的请求头中的origin完全一致。

10. 101报错

查看初始化参数是否符合要求，appid是否有效。

11. 104错误

1: 是否开启数据网络； 2: 后台查看对应APP ID备案的referer是否与当前页面referer一致。

12. 500错误

移动运营商的referer配置问题导致，需检查对应的配置。

13. 联通接口3011错误，源ip鉴权失败

客户端IP地址不在对应运营商IP列表内； 查看是否开启VPN代理或者使用免流浏览器进行访问。

14. 小程序中使用需要配置的合法域名？

<https://id6.me>； <https://onepass.geetest.com>； <https://www.cmpassport.com>； <https://opencloud.wostore.cn>；
<https://nishub1.10010.com:38750>；