

# 概述与资源

本文是 OneLogin Android SDK 的部署文档，用于指导 OneLogin Android SDK 的集成，读者需具有一定 Android 编程知识基础。

## 1、环境需求

条目	资源
开发目标	Android 5.0 及以上
开发环境	Android Studio 2.1.3
Gradle 版本	3.3.0 以上
SDK 三方依赖	<code>com.cmic.gen.sdk</code> 、 <code>com.unigeetest</code> 、 <code>cn.com.chinatelecom.account</code>
包增量	1M
网络制式	移动 2G/3G/4G/5G，联通 3G/4G/5G，电信 4G/5G（2G/3G 网络下时延相对较高，成功率相对较低）
网络环境	打开蜂窝数据流量并且给予应用蜂窝数据权限

## 2、相关开发资料

条目	资源
SDK 资源包	<a href="#">点击下载</a>
SDK 当前版本	<b>2.9.2</b> <b>(Tips: 0.8.0（不含）以下版本升级到该版本，服务端接口必须一同调整，否则会导致电信取号失败)</b>

# 准备工作

## 1、获取应用

向商务或客服获取 APPID 和 Key。

## 2、快速体验 Demo

Android 压缩包附带的 demo 文件夹中的示例工程，使用 Android Studio 打开示例工程，直接运行起来即可进行快速体验测试。

如果开发者需要将 SDK 集成到自己的项目进行体验，可完成以下配置步骤进行本地测试：

1. 将包名修改成对应的测试包名；
2. 将签名配置改成您的签名配置；
3. 将 APPID 换成您获取的APPID的信息(需注意 APPID 和包名/包签名的一致性)；
4. 参照服务端接入文档完成服务端接口的对接，**注意客户端 APPID 与服务端保持一致。如未完成该步骤则仅能体验APP端功能，不能获取真实手机号。**

## 3、开发环境搭建

### 3.1、gradle 集成

```
implementation 'com.geetest.android:onelogin:2.9.2'
```

注：library 中集成 SDK 应改用 `api` 依赖方式。`app` Module 依赖 `登录管理` Module，由该 `登录管理` Module 来依赖一键登录 SDK，这样的依赖关系就是 library 中集成 SDK 的情况。

### 3.2、手动集成

导入 SDK 到项目工程并配置开发环境

1. 在 sdk 目录下，将获取的 `geetest_onelogin_android_vx.x.x.xxxxxxxx.aar` 文件拖拽到工程中的 `libs` 文件夹下。
2. 在拖入 `.aar` 到 `libs` 文件夹后，还要检查 `.aar` 是否被添加到 **Library**，要在项目的 `build.gradle` 下添加如下代码：

```
repositories {
    flatDir {
        dirs 'libs'
    }
}
```

3. 手动将aar包添加依赖：

```
implementation(name: 'geetest_onelogin_android_vx.x.x.xxxxxxxx', ext: 'aar')
```

4. AndroidManifest.xml 权限配置：

SDK 已默认配置必要的权限以及内部 Activity 声明，正常情况下直接集成即可。**如有解压 aar 包单独集成 jar 与 资源文件的需求，请务必添加以下必要的权限。**  
必要的权限包含：

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

可选配的权限：

```
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
```

Android 6.0 以后 READ\_PHONE\_STATE 权限需要动态申请才有效，**SDK 不声明该权限，如需该权限，请添加权限声明并在应用内动态申请。**SDK 使用 READ\_PHONE\_STATE 权限主要用于获取网络制式，提升取号的成功率，建议开发者申请本权限。**未申请该权限不影响 SDK 正常取号功能。**

权限说明：

权限名称	权限说明	使用说明
INTERNET	允许应用程序联网	用于访问运营商网关和认证服务器
ACCESS_WIFI_STATE	允许访问 WiFi 网络状态信息	允许 SDK 获取 WiFi 网络状态与连接信息
ACCESS_NETWORK_STATE	允许访问网络状态	允许 SDK 判断当前网络处于移动网络或 WiFi 网络
CHANGE_NETWORK_STATE	允许改变网络连接状态	设备在 WiFi 跟数据双开时，允许 SDK 内部请求（仅用于网关取号的请求）强行走数据网络通道
CHANGE_WIFI_STATE	允许改变 WiFi 网络连接状态	设备在 WiFi 跟数据双开时，切允许 SDK 内部请求（仅用于网关取号的请求）强行走数据网络通道
READ_PHONE_STATE	允许读取手机状态	获取网络制式 (非必须权限)

## 5. 混淆配置：

SDK 的 aar 包已做混淆处理，两次混淆会导致不可预期的错误，默认情况下 aar 包内包含当前 SDK 的混淆配置，远端依赖或者本地 aar 依赖都能确保 SDK 不被二次混淆。**如有解压 aar 包单独集成 jar 与 资源文件的需求，请务必将解压目录中的 proguard.txt 文件内容拷贝到应用的混淆配置中，跳过对 SDK 的二次混淆操作。**

如过使用了 andRes 进行资源混淆，请将 SDK 内部包含资源排除混淆，避免因找不到资源而报错。排除配置如下：

```

whitelist = [
    //onelogin
    "R.drawable.umcsdk_load_dot_white",
    "R.drawable.gt_one_login_",
    "R.layout.gt_",
    "R.id.gt_one_login_",
    "R.style.GtOneLoginTheme",
    "R.xml.network_security_config",
    "R.color.gt_",
    "R.anim.gt_",
    "R.string.gt_"
]

```

6. Android 9.0 以上设备配置对 http 协议的支持:

- 方式一:

```

<application android:usesCleartextTraffic="true">
    //...
</application>

```

注: `android:usesCleartextTraffic="true"` 请务必配到 `application` 节点上, 配置到其他节点不生效

- 方式二:

在 `res` 文件夹下创建一个 `xml` 文件夹, 然后创建一个 `networksecurityconfig.xml` 文件, 文件内容如下 (具体可参考[官方文档](#)):

```

<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
    <base-config cleartextTrafficPermitted="true">
        <trust-anchors>
            <certificates src="system" />
        </trust-anchors>
    </base-config>
</network-security-config>

```

示例代码:

```

<application android:networkSecurityConfig="@xml/network_security_config">
    //...
</application>

```

- 方式三:

目前运营商个别接口为 `http` 请求, 对于全局禁用 `Http` 的项目, 需要设置 `Http` 白名单。以下为运营商 `http` 接口域名和 `ip` 名单: `cmpassport.com`、`10010.com`、`id6.me`

```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
  <!-- Android 9.0 上 http 请求域名白名单配置 -->
  <domain-config cleartextTrafficPermitted="true">
    <domain includeSubdomains="true">cmpassport.com</domain>
    <domain includeSubdomains="true">10010.com</domain>
    <domain includeSubdomains="true">id6.me</domain>
  </domain-config>
</network-security-config>
```

# OneLogin（一键登录）

进阶模式和常规模式是一键登录的两种调用逻辑，如无特别需求建议直接使用进阶模式，两模式不可混合调用。

## 1、进阶模式

### 1.1、调用逻辑

预取号逻辑封装在 SDK 内部，开发者只需控制授权页拉起时机，无需关心预取号是否完成。SDK 内部处理预取号的逻辑，包括预取号超期后的重新预取号，以及弱网状态下的重试等。

1. `init()` 初始化 SDK 并配置 APPID
2. `register()` 注册（应用启动或进入登录页的前一个页面、用户登出时是调用该方法的时机）
3. `requestToken()` 拉起授权页面（调用该方法前可以调用 `isPreGetTokenResultValidate` 判断预取号是否成功，失败时可先加载自定义 loading 等待对话框）
4. `dismissAuthActivity()` 关闭授权页面

`register` 方法内会自动进行预取号，如果应用启动就调用，建议先判断或者请求网络权限之后再调用，否则可能因为网络不可达导致预取号失败。另外建议 `register` 与 `requestToken` 方法不要同时调用，预留预取号一定的时间以及内部初始化。

注：**preGetToken** 为常规逻辑预取号方法，请勿与进阶逻辑 **register** 混用

### 1.2、初始化

- 初始化 `init`

SDK 初始化接口 `init`，可放在 `Application` 的 `onCreate` 方法中进行初始化。

#### 方法描述

```
public OneLoginHelper init(Context context);
public OneLoginHelper init(Context context, String appId);
```

#### 参数说明

参数	类型	说明
context	Context	上下文
appId	String	唯一产品 APPID，注意与服务端保持一致

- 注册 `register`

## 方法描述

注册接口，可同 `init` 方法一起放在 `Application` 的 `onCreate` 方法中进行注册，或者在登录页前一个页面初始化时调用；登录成功后 SDK 内部不再维护预取号的有效性，如果用户退出登录后，为了方便下次重新登录能快速拉起授权页，也可以重新调用 `register` 进行预取号。

```
public void register(String appId)
public void register(String appId, int timeout)
```

## 参数说明

参数	类型	说明
app_id	String	唯一产品 APPID，如 <code>init</code> 接口传了 <code>appId</code> 此处可传空值
timeout	int	超时时间，单位: <code>ms</code> ，取值范围: 1000~15000，默认 8000。传递该参数会统一设置预取号超时时间和取号超时时间为设定值，分别设置请使用不带 <code>timeout</code> 的方法并参考 <code>setRequestTimeout</code> 。

## 代码示例

```
OneLoginHelper
    .with()
    //开启 SDK 日志打印功能
    .setLogEnable(true)
    .setRequestTimeout(8000, 8000)
    //第一个参数为当前 Application 或 Activity 的 Context
    //第二个参数为所需要配置的 APPID，注意与服务端保持一致
    .init(this, APPID)
    .register("");
```

## 1.3、拉起授权页

- 拉起授权页 `requestToken`

## 方法描述

在需要登录的地方调用 `requestToken` 接口拉起一键登录授权页，待用户点一键登录授权后获取运营商 `token`，获取成功后即可请求服务端换取本机手机号码。

```
public void requestToken(OneLoginThemeConfig oneLoginThemeConfig,
    AbstractOneLoginListener listener)
public void requestToken(Activity activity, OneLoginThemeConfig
    oneLoginThemeConfig, AbstractOneLoginListener oneLoginListener)
```

## 参数说明

参数	类型	说明
activity	Activity	Activity实例，需要使用Dialog实现弹窗样式授权页的必须传递此参数
oneLoginThemeConfig	OneLoginThemeConfig	自定义全局配置接口，用来配置授权页面 UI 样式
listener	AbstractOneLoginListener	回调监听器，需要开发者自己实现

注：Dialog实现的授权页暂不支持分屏或小窗口中滚动展示

### 代码示例

取号并获取免密登录的 token，通过接口进行校验，并获取登录信息。

```
OneLoginHelper
    .with()
    .requestToken(new OneLoginThemeConfig.Builder().build(), new
AbstractOneLoginListener() {
    @Override
    public void onResult(JSONObject jsonObject) {
        try {
            int statusResult = jsonObject.getInt("status");
            // status=200 为取号成功，其他返回码请参考返回码章节
            if (statusResult != 200) {
                Toast.makeText(getApplicationContext(), jsonObject.toString(),
Toast.LENGTH_SHORT).show();
                return;
            }
            new Thread(() -> {
                JSONObject json = new JSONObject();
                try {
                    json.put("process_id", jsonObject.getString("process_id"));
                    json.put("token", jsonObject.getString("token"));
                    json.put("authcode", jsonObject.optString("authcode")); //必
选参数

                } catch (JSONException e) {
                    e.printStackTrace();
                }
                String result = HttpUtils.requestNetwork(<---获取用户登录信息的接口-
-->, json);

                ///
                /// TODO 判断 result 是否有效
                ///
            }).start();
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
});
```

## 1.4、关闭授权页

- 关闭授权页 dismissAuthActivity

## 方法描述

主动关闭授权页，包括关闭Dialog实现的授权页。SDK 除了返回按钮触发关闭以外，默认是不 finish 授权页的，需要开发者在回调结束后自行实现关闭授权页。

```
public void dismissAuthActivity()
```

## 代码示例

```
OneLoginHelper.with().dismissAuthActivity();
```

## 2、常规模式

如无特别需求建议直接使用进阶模式

## 授权页面 UI 修改

通过实例化 `OneLoginThemeConfig` 并进行自定义配置，完成授权页面 UI 的个性化设计，每次调用拉起授权页方法前必须先传入该实例，否则授权界面会展示异常。授权页 UI 设计规范如下：



## 1、创建实例

- 创建实例

```
OneLoginThemeConfig oneLoginThemeConfig = new
OneLoginThemeConfig.Builder().build();
```

修改主题配置的方法请在 `build` 方法调用之前调用，所有配置方法参考下文接口说明。

## 2、弹框模式设置

- 设置弹窗主题相关 `setDialogTheme`

### 方法描述

设置弹窗主题相关。

```
setDialogTheme(boolean isDialogTheme, int dialogwidth, int dialogHeight, int dialogX, int dialogY, boolean isDialogBottom, boolean iswebViewDialogTheme)
```

### 参数说明

参数	参数类型	说明	默认值
isDialogTheme	boolean	是否使用弹窗模式，true 为使用，false 为不使用	false
dialogWidth	int	弹窗模式宽度，长宽的单位都为 <code>dp</code> ， <b>以下单位与之保持一致</b>	300
dialogHeight	int	弹窗模式高度	500
dialogX	int	授权页弹窗X偏移量（以屏幕心为原点）	0
dialogY	int	授权页弹窗Y偏移量（以屏幕心为原点）	0
isDialogBottom	boolean	授权页弹窗是否贴于屏幕底部 true：显示到屏幕底部，dialogY 参数设置将无效 false：不显示到屏幕底部，以 dialogY 参数为准	false
isWebViewDialogTheme	boolean	隐私页面是否使用弹窗模式	false

## 3、添加自定义控件

- 添加自定义控件 `addOneLoginRegisterViewConfig`

### 方法描述

在 `requestToken()` 方法之前实现。允许开发者在授权页面 `titlebar` 和 `body` 添加自定义的控件  
注意：自定义的控件不允许覆盖 SDK 默认的 UI。

```
public OneLoginHelper addOneLoginRegisterViewConfig(String id, com.geetest.onelogin.config.AuthRegisterViewConfig authRegisterViewConfig)
```

### 代码示例

```
OneLoginHelper.with().addOneLoginRegisterViewConfig(id, authRegisterViewConfig);
```

### 参数说明

参数	类型	说明
id	String	开发者自定义控件名称
authRegisterViewConfig	AuthRegisterViewConfig	配置开发者自定义控件的控件来源，位置和处理逻辑等

- 实例化 AuthRegisterViewConfig 自定义控件配置

### 接口描述

创建自定义控件配置

```
AuthRegisterViewConfig authRegisterViewConfig = new
AuthRegisterViewConfig.Builder().build();
```

### 方法说明

方法名	传参类型	说明
setView	View	开发者传入自定义的控件，需要设置好控件的布局属性，SDK <b>只支持 RelativeLayout 布局</b> ，控件位置请通过 RelativeLayout.LayoutParams 配置指定，具体配置可参考 demo 示例代码
setRootViewId	int	设置控件的位置。 RootViewId.ROOT_VIEW_ID_TITLE_BAR 指标题栏 RootViewId.ROOT_VIEW_ID_BODY 指授权页空白处
setCustomInterface	CustomInterface	设置控件事件。CustomInterface 是一个接口，自定义实现点击方法

## 4、授权页面其他配置

- 设置授权页面背景 setAuthBGImgPath

### 方法描述

设置背景图片。

```
setAuthBGImgPath(String authBGImgPath)
```

### 参数说明

参数	参数类型	说明	默认值
authBGImgPath	String	设置背景图片。放在 <code>drawable</code> 目录下，以下背景图片路径与之保持一致。不需要加后缀，比如图片在 <code>drawable</code> 中的存放目录是 <code>res/drawable-xxhdpi/demo_bg.png</code> ，则传入参数为 <code>demo_bg</code>	gt_one_login_bg

- 设置授权页面视频背景 `setAuthBgVideoUri`

### 方法描述

设置授权页面视频背景。

```
setAuthBgVideoUri(String authBgVideoUri)
```

### 参数说明

参数	参数类型	说明	默认值
authBgVideoUri	String	视频背景 Uri 字符串。放在 <code>raw</code> 文件下，如： <code>"android.resource://" + this.getPackageName() + "/" + R.raw.geetest_demo_video</code>	null

注：当背景图片和背景视频都为空（null或""）时，SDK将生成一个附带生命周期的背景容器，见回调 `onAuthActivityViewInit` 的参数 `bgContainer`，接入方可在该回调中向容器添加背景。Dialog实现的授权页中的背景容器生命周期只有 `onDestroy`，在 `Dialog` 的 `onDetachedFromWindow` 时回调。

- 状态栏设置 `setStatusBar`

### 方法描述

设置状态栏颜色、背景是否延伸到状态栏、状态栏字体颜色风格。

```
setStatusBar(int statusBarColor, UserInterfaceStyle statusBarStyle, boolean bgLayoutInStatusBar)
```

### 参数说明

参数	参数类型	说明	默认值
statusBarColor	int	自定义状态栏背景颜色，优先按资源id查找，找不到则按固定颜色值设置	0
statusBarStyle	UserInterfaceStyle	状态栏内容的颜色风格， <code>UserInterfaceStyle.LIGHT</code> 内容为黑色， <code>UserInterfaceStyle.DARK</code> 内容为白色， <code>UserInterfaceStyle.UNSPECIFIED</code> 根据系统深色模式自动适配内容颜色	UserInterfaceStyle.UNSPECIFIED
bgLayoutInStatusBar	boolean	背景是否延伸到状态栏	true

注：状态栏内容的颜色风格在 Android 6.0及以上系统生效，其他配置在Android 5.0及以上系统设置才能生效。在背景不延伸到状态栏时应搭配非透明的颜色。Dialog实现的授权页中设置无效。

- 导航栏设置 `setNavigationBar`

### 方法描述

设置导航栏颜色、背景是否延伸到导航栏、导航栏按钮颜色风格。

```
setNavigationBar(int navigationBarColor, UserInterfaceStyle navigationBarStyle,
boolean bgLayoutInNavigationBar)
```

### 参数说明

参数	参数类型	说明	默认值
navigationBarColor	int	自定义底部导航栏背景颜色，优先按资源id查找，找不到则按固定颜色值设置	0
navigationBarStyle	UserInterfaceStyle	导航栏按钮的颜色风格，UserInterfaceStyle.LIGHT 按钮为黑色，UserInterfaceStyle.DARK 按钮为白色，UserInterfaceStyle.UNSPECIFIED 根据系统深色模式自动适配按钮颜色	UserInterfaceStyle.UNSPECIFIED
bgLayoutInNavigationBar	boolean	背景是否延伸到导航栏	true

注：导航栏按钮的颜色风格在 Android 8.0及以上系统生效，其他配置在Android 5.0及以上系统设置才能生效。在背景不延伸到导航栏时应搭配非透明的颜色。另外，底部浮窗模式下导航栏按钮的颜色不能很好受控，建议设置 `navigationBarColor` 为不透明的颜色，如纯黑色，防止导航栏按钮不可见。Dialog实现的授权页中设置无效。

- 标题栏设置 `setAuthNavLayout`

### 方法描述

设置标题栏布局

```
setAuthNavLayout(int navColor, int authNavHeight, boolean authNavTransparent,
boolean authNavGone)
```

### 参数说明

参数	参数类型	说明	默认值
navColor	int	标题栏颜色，优先按资源id查找，找不到则按固定颜色值设置	默认0xFF3973FF，黑暗模式0x2E2E30
authNavHeight	int	标题栏高度	49
authNavTransparent	boolean	标题栏是否透明	true
authNavGone	boolean	标题栏是否隐藏，此处的隐藏为 <code>View.GONE</code>	false

- 服务条款页面标题栏设置 `setWebNavLayout`

### 方法描述

设置服务条款页面标题栏布局

```
setWebNavLayout(int webNavColor, int webNavHeight, boolean webNavTransparent)
```

参数说明

参数	参数类型	说明	默认值
webNavColor	int	标题栏颜色，优先按资源id查找，找不到则按固定颜色值设置	默认0xFF3973FF，黑暗模式0x2E2E30
webNavHeight	int	标题栏高度	49
webNavTransparent	boolean	标题栏是否透明	true

- 标题栏文字设置 `setAuthNavTextView`

方法描述

设置标题栏中间文字相关

```
setAuthNavTextView(String navText, int navTextColor, int navTextSize, boolean navWebTextNormal, String navWebViewText, int navWebViewTextColor, int navWebViewTextSize)
setAuthNavTextView(String navText, int navTextColor, int navTextSize, boolean navWebTextNormal, String navWebViewText, int navWebViewTextColor, int navWebViewTextSize, int navTextMargin)
```

参数说明

参数	参数类型	说明	默认值
navText	String	文字设置，默认为空	""
navTextColor	int	字体颜色，优先按资源id查找，找不到则按固定颜色值设置	默认 0xFF000000， 黑暗模式 0xFFFFFFFF
navTextSize	int	字体大小，单位为 <code>sp</code> ，以下设置字体大小的单位与之保持一致	17
navWebTextNormal	boolean	设置是否隐私条款页面的标题栏中间文字使用默认值，true 为使用 <code>navWebViewText</code> ，false 为使用默认隐私条款的名字	false
navWebViewText	String	隐私条款页面的标题栏中间文字	服务条款
navWebViewTextColor	int	隐私条款页面的标题栏中间文字颜色，优先按资源id查找，找不到则按固定颜色值设置	默认 0xFF000000， 黑暗模式 0xFFFFFFFF
navWebViewTextSize	int	隐私条款页面的标题栏中间文字大小	17
navTextMargin	int	标题栏文本的左右间距，可选配	36

- 标题栏字体设置 `setAuthNavTextViewTypeface`

### 方法描述

设置标题栏中间文字的字体相关

```
setAuthNavTextViewTypeface(Typeface navTextTypeface, Typeface navWebViewTextTypeface)
```

### 参数说明

参数	参数类型	说明	默认值
navTextTypeface	Typeface	标题栏的文字的字体	Typeface.DEFAULT
navWebViewTextTypeface	Typeface	条款页面标题栏的文字的字体	Typeface.DEFAULT

- 标题栏返回按钮设置 `setAuthNavReturnImgview`

### 方法描述

设置标题栏返回按钮相关

```
setAuthNavReturnImgView(String returnImgPath, int returnImgWidth, int returnImgHeight, boolean returnImgHidden, int returnImgOffsetX)
setAuthNavReturnImgView(String returnImgPath, int returnImgWidth, int returnImgHeight, boolean returnImgHidden, int returnImgOffsetX, int returnImgOffsetY)
```

参数说明

参数	参数类型	说明	默认值
returnImgPath	String	返回按钮图片	gt_one_login_ic_chevron_left_black
returnImgWidth	int	返回按钮图片宽度	24
returnImgHeight	int	返回按钮图片高度	24
returnImgHidden	boolean	返回按钮是否隐藏	false
returnImgOffsetX	int	返回按钮图片距离屏幕左边X轴偏移量	12
returnImgOffsetY	int	返回按钮图片距离标题栏顶部偏移量，可选配，不传该参数默认标题栏垂直居中	0

- 服务条款页面标题栏返回按钮设置 `setWebNavReturnImgView`

方法描述

设置服务条款页面标题栏返回按钮相关

```
setWebNavReturnImgView(String returnImgPath, int returnImgWidth, int returnImgHeight, int returnImgOffsetX, boolean returnImgCenterInVertical, int returnImgOffsetY)
```

参数说明

参数	参数类型	说明	默认值
returnImgPath	String	返回按钮图片	gt_one_login_ic_chevron_left_black
returnImgWidth	int	返回按钮图片宽度	24
returnImgHeight	int	返回按钮图片高度	24
returnImgHidden	boolean	返回按钮是否隐藏	false
returnImgOffsetX	int	返回按钮图片距离屏幕左边X轴偏移量	12
returnImgCenterInVertical	boolean	返回按钮图片是否在标题栏中垂直居中，若值为true将忽略returnImgOffsetY	true
returnImgOffsetY	int	返回按钮图片距离标题栏顶部偏移量，可选配，不传该参数默认标题栏垂直居中	0

- 返回事件屏蔽设置 `setBlockReturnEvent`

### 方法描述

设置是否屏蔽返回键与返回按钮返回事件，屏蔽后仍然会发生对应的错误码回调，只是 SDK 默认不关闭授权页。

```
setBlockReturnEvent(boolean blockReturnKey, boolean blockReturnBtn)
```

### 参数说明

参数	参数类型	说明	默认值
blockReturnKey	boolean	是否屏蔽返回键返回事件	false
blockReturnBtn	boolean	是否屏蔽返回按钮返回事件	false

注：2.3.1 以前版本升级到最新版本且配置了屏蔽返回事件后，注意修改 `onResult` 回调内针对 -20301 与 -20302 错误码的处理，避免还是调用了 `dismissAuthActivity`。

- logo设置 `setLogoImgView`

### 方法描述

设置logo相关

```
setLogoImgView(String logoImagePath, int logowidth, int logoHeight, boolean logoHidden, int logoOffsetY, int logoOffsetY_B, int logoOffsetX)
```

### 参数说明

参数	参数类型	说明	默认值
logoImgPath	String	logo 图片	gt_one_login_logo
logoWidth	int	logo 图片宽度	71
logoHeight	int	logo 图片高度	71
logoHidden	boolean	logo 是否隐藏	false
logoOffsetY	int	logo 相对于状态栏下边缘 y 偏移	125
logoOffsetY_B	int	logo 相对于底部 y 偏移	0
logoOffsetX	int	logo 相对于屏幕左边 x 轴偏移量，当为0时表示居中显示	0

- 号码显示设置 `setNumberView`

### 方法描述

设置号码相关

```
setNumberView(int numberColor, int numbersSize, int numberOffsetY, int numberOffsetY_B, int numberOffsetX)
```

### 参数说明

参数	参数类型	说明	默认值
numberColor	int	号码栏字体颜色，优先按资源id查找，找不到则按固定颜色值设置	默认0xFF3D424C，黑暗模式0xFFFFFFFF
numberSize	int	号码栏字体大小	24
numberOffsetY	int	号码栏相对于标题栏下边缘 y 偏移	200
numberOffsetY_B	int	号码栏相对于底部 y偏移	0
numberOffsetX	int	号码栏相对于屏幕左边 x 轴偏移量，当为 0 时表示居中显示	0

- 号码字体设置 `setNumberText`

### 方法描述

设置号码栏富文本样式，配置后将使用该富文本填充脱敏手机号内容

```
setNumberText(CharSequence numberText)
```

参数说明

参数	参数类型	说明	默认值
numberText	CharSequence	号码栏的富文本内容, 参考 TextView 的 <code>setText(CharSequence text)</code> 方法	null

注：富文本内容应该包含正确的脱敏手机号，且配置之前必须确保能拿到正确的脱敏手机号

- 号码字体设置 `setNumberViewTypeface`

方法描述

设置号码栏字体相关

```
setNumberViewTypeface(Typeface numberTypeface)
```

参数说明

参数	参数类型	说明	默认值
numberTypeface	Typeface	号码栏的文字的字体	Typeface.DEFAULT

- 号码栏布局设置 `setNumberLayout`

方法描述

设置号码栏布局

```
setNumberLayout(int numberwidth, int numberHeight)
```

参数说明

参数	参数类型	说明	默认值
numberWidth	int	号码栏的宽度，仅支持设置宽度自适应或者是固定值	ViewGroup.LayoutParams.WRAP_CONTENT
numberHeight	int	号码栏的高度，仅支持设置高度自适应或者是固定值	ViewGroup.LayoutParams.WRAP_CONTENT

- 切换账号视图设置 `setSwitchView`

方法描述

设置切换账号相关

```
setSwitchView(String switchText, int switchColor, int switchSize, boolean switchHidden, int switchOffsetY, int switchOffsetY_B, int switchOffsetX)
```

参数说明

参数	参数类型	说明	默认值
switchText	String	切换账号文字	切换账号，默认情况下受背景大小限制，只能显示4个字，如遇显示不全，建议通过 setSwitchViewLayout 调整背景大小试试
switchColor	String	切换账号字体颜色，优先按资源id查找，找不到则按固定颜色值设置	默认0xFF3973FF，黑暗模式 0xFF6794FF
switchSize	int	切换账号字体大小	14
switchHidden	boolean	切换账号是否隐藏	false
switchOffsetY	int	切换账号相对于标题栏下边缘 y 偏移	249
switchOffsetY_B	int	切换账号相对于底部 y 偏移	0
switchOffsetX	int	切换账号相对于屏幕左边 x 轴偏移量，当为0时表示居中显示	0

- 切换账号视图背景设置 setSwitchViewLayout

方法描述

设置切换账号背景图片与宽高相关

```
setSwitchViewLayout(String switchImgPath, int switchwidth, int switchHeight)
```

参数说明

参数	参数类型	说明	默认值
switchImgPath	String	切换账号背景图片	默认无背景
switchWidth	int	切换账号背景宽度	80
switchHeight	int	切换账号背景高度	25

- 切换账号字体设置 setSwitchViewTypeface

方法描述

设置切换账号字体相关

```
setSwitchViewTypeface(Typeface switchTypeface)
```

参数说明

参数	参数类型	说明	默认值
switchTypeface	Typeface	切换账号的文字的字体	Typeface.DEFAULT

- 登录按钮布局设置 setLogBtnLayout

方法描述

设置登录按钮布局

```
setLogBtnLayout(String logBtnImgPath, int logBtnwidth, int logBtnHeight, int logBtnOffsetY, int logBtnOffsetY_B, int logBtnOffsetX)
setLogBtnLayout(String logBtnImgPath, String logBtnUncheckedImgPath, int logBtnwidth, int logBtnHeight, int logBtnOffsetY, int logBtnOffsetY_B, int logBtnOffsetX)
```

参数说明

参数	参数类型	说明	默认值
logBtnImgPath	String	登录按钮背景图片	gt_one_login_btn
logBtnUncheckedImgPath	String	未选中隐私栏勾选框时登录按钮的背景图片	gt_one_login_btn_unchecked
logBtnWidth	int	登录按钮宽度	268
logBtnHeight	int	登录按钮高度	36
logBtnOffsetY	int	登录按钮相对于标题栏下边缘 y 偏移	324
logBtnOffsetY_B	int	登录按钮相对于底部 y 偏移	0
logBtnOffsetX	int	登录按钮相对于屏幕左边 x 轴偏移量,当为0时表示居中显示	0

- 登录按钮文字设置 setLogBtnTextView

方法描述

设置登录按钮中间文字相关

```
setLogBtnTextView(String logBtnText, int logBtnColor, int logBtnTextSize)
```

参数说明

参数	参数类型	说明	默认值
logBtnText	String	文字设置	一键登录
logBtnColor	int	文字颜色，优先按资源id查找，找不到则按固定颜色值设置	0xFFFFFFFF
logBtnTextSize	int	文字大小	15

- 登录按钮文字字体设置 `setLogBtnTextViewTypeface`

方法描述

设置登录按钮中间文字的字体相关

```
setLogBtnTextViewTypeface(Typeface logBtnTextTypeface)
```

参数说明

参数	参数类型	说明	默认值
logBtnTextTypeface	Typeface	登录按钮中间的文字的字体	Typeface.DEFAULT

- 登录按钮可用状态随选择框选择状态变化设置 `setLogBtnDisableIfUnchecked`

方法描述

设置登录按钮在选择框未选择时禁用状态

```
setLogBtnDisableIfUnchecked(boolean disableBtnIfUnchecked)
```

参数说明

参数	参数类型	说明	默认值
disableBtnIfUnChecked	boolean	登录按钮是否在选择框未选择时自动禁用	false

- 登录按钮可多次点击设置 `setAuthBtnMultipleClick`

方法描述

设置取号成功后点击登录按钮能否收到点击回调(指 `void onLoginButtonClick()`)

```
setAuthBtnMultipleClick(boolean enable)
```

参数说明

参数	参数类型	说明	默认值
enable	boolean	在取号成功后点击登录按钮能否收到点击回调	false

- 登录加载视图设置 `setLogBtnLoadingView`

方法描述

设置loading图片相关

```
setLogBtnLoadingview(String loadingview, int loadingviewwidth, int loadingviewHeight, int loadingViewOffsetRight)
setLogBtnLoadingview(String loadingview, int loadingviewwidth, int loadingviewHeight, int loadingViewOffsetRight, int loadingViewOffsetY)
```

参数说明

参数	参数类型	说明	默认值
loadingView	String	loading 图片地址	umcsdk_load_dot_white
loadingViewWidth	int	loading 图片宽度	20
loadingViewHeight	int	loading 图片高度	20
loadingViewOffsetRight	int	loading 图片距离登录框右边 x 轴偏移量	12
loadingViewOffsetY	int	loading 图片距离登录框 y 轴偏移量，可选配，不传该参数默认登录框垂直居中	0

- 标语是否显示 `setSlogan`

方法描述

设置 Slogan 是否显示

```
setSlogan(boolean visible)
```

参数说明

参数	参数类型	说明	默认值
visible	boolean	slogan 是否显示	true

- 标语文本设置 `setSloganText`

方法描述

设置 Slogan 文本

```
setSloganText(String sloganText)
```

参数说明

参数	参数类型	说明	默认值
sloganText	String	slogan 文本	运营商标语，如 中国移动提供认证服务

- 标语设置 `setSloganView`

方法描述

设置 Slogan 相关

```
setSloganView(int sloganColor, int sloganSize, int sloganOffsetY, int sloganOffsetY_B, int sloganOffsetX)
```

参数说明

参数	参数类型	说明	默认值
sloganColor	int	设置 Slogan 字体颜色，优先按资源id查找，找不到则按固定颜色值设置	默认0xFFA8A8A8，黑暗模式0xFFD0D1E2
sloganSize	int	设置 Slogan 字体大小	10
sloganOffsetY	int	设置 Slogan 相对于标题栏下边缘 y 偏移	382
sloganOffsetY_B	int	设置 Slogan 相对于底部 y 偏移	0
sloganOffsetX	int	设置 Slogan 相对于屏幕左边 x 轴偏移量，当为0时表示居中显示	0

- 标语字体设置 `setSloganViewTypeface`

方法描述

设置 Slogan 文字的字体相关

```
setSloganViewTypeface(Typeface sloganTypeface)
```

参数说明

参数	参数类型	说明	默认值
sloganTypeface	Typeface	Slogan 文字的字体	Typeface.DEFAULT

- Slogan 布局设置 `setSloganLayout`

方法描述

设置 Slogan 布局

```
setSloganLayout(int sloganwidth, int sloganHeight)
```

参数说明

参数	参数类型	说明	默认值
sloganWidth	int	Slogan 的宽度，仅支持设置宽度自适应或者是固定值	ViewGroup.LayoutParams.WRAP_CONTENT
sloganHeight	int	Slogan 的高度，仅支持设置高度自适应或者是固定值	ViewGroup.LayoutParams.WRAP_CONTENT

- 隐私栏布局设置 `setPrivacyLayout`

方法描述

设置隐私条款布局

```
setPrivacyLayout(int privacyLayoutWidth, int privacyOffsetY, int privacyOffsetY_B, int privacyOffsetX, boolean isUseNormalWebActivity)
setPrivacyLayout(int privacyLayoutWidth, int privacyOffsetY, int privacyOffsetY_B, int privacyOffsetX, boolean isUseNormalWebActivity, int gravity)
```

参数说明

参数	参数类型	说明	默认值
privacyLayoutWidth	int	设置隐私条款宽度，隐私条款大部分为字体，只规定宽度，高度自适应。设置宽度为 <code>ViewGroup.LayoutParams.MATCH_PARENT</code> ，隐私条款的宽度将填充父布局；设置宽度为 <code>ViewGroup.LayoutParams.WRAP_CONTENT</code> ，隐私条款的宽度将根据文本内容自动扩展；其他情况将按照固定值设置宽度。	256
privacyOffsetY	int	设置隐私条款相对于标题栏下边缘 y 偏移	0
privacyOffsetY_B	int	设置隐私条款相对于底部 y 偏移	18
privacyOffsetX	int	设置隐私条款对于屏幕左边 x 轴偏移量，当为 0 时表示居中显示	0
isUseNormalWebActivity	boolean	设置是否跳转到默认的隐私条款页面	true
gravity	int	设置隐私条款选择框和文本的对齐方式，参考 <code>LinearLayout</code> 的 <code>setGravity(int gravity)</code> 方法，可选配	<code>Gravity.CENTER_HORIZONTAL</code>

注：privacyOffsetY与privacyOffsetY\_B两个偏移量必须有一个为0，否则优先采用privacyOffsetY\_B。

- 隐私条款布局的抖动动画样式设置 setProtocolShakeStyle

方法描述

设置隐私条款布局的抖动动画样式

```
setProtocolShakeStyle(ProtocolShakeStyle protocolShakeStyle)
```

参数说明

参数	参数类型	说明	默认值
protocolShakeStyle	ProtocolShakeStyle	抖动动画的样式，支持水平抖动和竖直抖动。	NONE（无抖动）

- 隐私栏勾选框设置 setPrivacyCheckBox

方法描述

设置隐私条款选择框相关

```
setPrivacyCheckBox(String uncheckedImgPath, String checkedImgPath, boolean privacyState, int privacyCheckBoxWidth, int privacyCheckBoxHeight)
setPrivacyCheckBox(String uncheckedImgPath, String checkedImgPath, boolean privacyState, int privacyCheckBoxWidth, int privacyCheckBoxHeight, int privacyCheckBoxOffsetY)
setPrivacyCheckBox(String uncheckedImgPath, String checkedImgPath, boolean privacyState, int privacyCheckBoxWidth, int privacyCheckBoxHeight, int privacyCheckBoxOffsetY, int privacyCheckBoxMarginRight)
```

参数说明

参数	参数类型	说明	默认值
uncheckedImgPath	String	设置未选中下按钮的图片地址	gt_one_login_unchecked
checkedImgPath	String	设置选中下按钮的图片地址	gt_one_login_checked
privacyState	boolean	设置选择框是否默认勾选	false
privacyCheckBoxWidth	int	选择框图片宽度	9
privacyCheckBoxHeight	int	选择框图片高度	9
privacyCheckBoxOffsetY	int	选择框图片Y轴偏移，可选配	0
privacyCheckBoxMarginRight	int	选择框图片右边距，可选配	5

- 隐私条款文字显示设置 `setPrivacyClauseView`

方法描述

设置隐私条款字体相关

```
setPrivacyClauseView(int baseClauseColor, int clauseColor, int privacyClauseTextSize)
```

参数说明

参数	参数类型	说明	默认值
baseClauseColor	int	设置隐私条款基础文字颜色，优先按资源id查找，找不到则按固定颜色值设置	默认0xFFA8A8A8， 黑暗模式 0xFFD0D1E2
clauseColor	int	设置隐私条款协议文字颜色，优先按资源id查找，找不到则按固定颜色值设置	0xFF3973FF
privacyClauseTextSize	int	设置隐私条款字体大小	10

- 隐私条款文字设置 `setPrivacyTextView`

方法描述

设置除了隐私条款其他的字体相关

```
setPrivacyTextView(String privacyTextViewTv1, String privacyTextViewTv2, String privacyTextViewTv3, String privacyTextViewTv4)
```

参数说明

参数	参数类型	说明	默认值
privacyTextViewTv1	String	设置隐私条款文字1	登录即同意
privacyTextViewTv2	String	设置隐私条款文字2	和
privacyTextViewTv3	String	设置隐私条款文字3	、
privacyTextViewTv4	String	设置隐私条款文字4	并使用本机号码登录

- 设置隐私栏文字的字体 `setPrivacyClauseViewTypeface`

方法描述

设置隐私栏文字的字体相关

```
setPrivacyClauseViewTypeface(Typeface privacyClauseBaseTypeface, Typeface privacyClauseTypeface)
```

### 参数说明

参数	参数类型	说明	默认值
privacyClauseBaseTypeface	Typeface	隐私栏基础的文字的字体	Typeface.DEFAULT
privacyClauseTypeface	Typeface	隐私条款的文字的字体	Typeface.DEFAULT

- 开发者隐私条款设置 `setPrivacyClauseText`

### 方法描述

设置开发者隐私条款相关。

自定义隐私条款顺序。按顺序设置，当有条款名称与条款 URL 一起为空时，则使用默认的运营商隐私条款。否则按照顺序排列，自定义隐私条款最多只能两个。

```
setPrivacyClauseText(String clauseNameOne, String clauseUrlOne, String clauseNameTwo, String clauseUrlTwo, String clauseNameThree, String clauseUrlThree)
```

### 参数说明

参数	参数类型	说明	默认值
clauseNameOne	String	设置隐私条款1名称	null
clauseUrlOne	String	设置隐私条款1 URL	null
clauseNameTwo	String	设置隐私条款2名称	null
clauseUrlTwo	String	设置隐私条款2 URL	null
clauseNameThree	String	设置隐私条款3名称	null
clauseUrlThree	String	设置隐私条款3 URL	null

- 开发者隐私条款设置 `setPrivacyClauseTextStrings`

### 方法描述

设置多个开发者隐私条款相关。

自定义多个隐私条款。按顺序设置，长度为 4 的倍数，配置后优先使用该接口的配置，否则使用 `setPrivacyClauseText` 的配置。可通过预留一组空字符串配置让 SDK 自动添加运营商协议。

```
setPrivacyClauseTextStrings(String... privacyClauseTextStrings)
```

### 参数说明

参数	参数类型	说明	默认值
privacyClauseTextStrings	String[]	设置隐私条款数组	null

代码示例

```
/**
 *连续四个参数为一组隐私协议条款 第一个为条款说明前缀或者分隔字符，第二个为条款名称，第三个为条款 URL，第四个为说明后缀或者分隔字符
 *任意一组留空，SDK 会自动添加运营商隐私条款，留空的位置可前后调整。
 */
.setPrivacyClauseTextStrings("登录即同意", "应用自定义服务条款一", "http://a.b.c", "",
                              "和", "应用自定义服务条款二", "http://x.y.z", "",
                              "和", "应用自定义服务条款三", "http://x.y.z", "",
                              "和", "", "", "并使用本机号码登录")
```

注:

运营商隐私协议必须显示，开发者需保留一组条款参数留空

- 隐私条款名称显示书名号相关 `setPrivacyAddFrenchQuotes`

方法描述

设置授权页面隐私条款名称显示书名号。

```
setPrivacyAddFrenchQuotes(boolean privacyAddFrenchQuotes)
```

注: 个别第三方系统可能存在对包含中文字符的富文本渲染异常，出现文字超出控件右边界显示被截断的情况，建议不要添加书名号。

参数说明

参数	参数类型	说明	默认值
privacyAddFrenchQuotes	boolean	是否设置隐私条款名称显示书名号	false

- 隐私条款文字对齐方式 `setPrivacyTextGravity`

方法描述

设置隐私条款文字对齐方式。

```
setPrivacyTextGravity(int privacyTextGravity)
```

参数说明

参数	参数类型	说明	默认值
privacyTextGravity	int	设置隐私条款文字对齐方式，参考 <code>TextView.setGravity</code> 方法	<code>Gravity.TOP Gravity.START</code>

- 隐私条款文字外部上边距 `setPrivacyTextTopMargin`

### 方法描述

设置隐私条款文字外部上边距。

```
setPrivacyTextTopMargin(int privacyTextTopMargin)
```

### 参数说明

参数	参数类型	说明	默认值
privacyTextTopMargin	int	设置隐私条款文字外部上边距	1

- 隐私条款文字行间距 `setPrivacyLineSpacing`

### 方法描述

设置隐私条款文字行间距，参考 `TextView.setLineSpacing` 方法。

```
setPrivacyLineSpacing(float lineSpacingExtra, float lineSpacingMultiplier)
```

### 参数说明

参数	参数类型	说明	默认值
lineSpacingExtra	float	设置隐私条款文字行间距，单位: dp	8.0f
lineSpacingMultiplier	float	设置隐私条款文字行间距的倍数	1.0f

- 未同意隐私条款时的文字提示 `setPrivacyUncheckedToastText`

### 方法描述

设置未同意隐私条款时的文字提示相关。

```
setPrivacyUncheckedToastText(String privacyUncheckedToastText)
setPrivacyUncheckedToastText(boolean enableToast, String
privacyUncheckedToastText)
```

### 参数说明

参数	参数类型	说明	默认值
enableToast	boolean	设置是否弹出 Toast 提示文字，可选配	true
privacyUnCheckedToastText	String	设置未同意隐私条款的文字提示相关	请同意服务条款

- 授权页语言 `setLanguageType`

方法描述

设置授权页（含服务条款页面，但不包含运营商服务条款的具体内容）语言，仅支持设置中文简体、中文繁体、英文，默认为中文简体（语言类型仅代表UI展示，一键登录功能仅支持中国移动、中国联通、中国电信的SIM卡）。

```
setLanguageType(OLLanguageType languageType)
```

参数说明

参数	参数类型	说明	默认值
languageType	OLLanguageType	语言类型	OLLanguageType.SIMPLIFIEDD_CHINESE

- 授权页在服务条款未选中时的弹窗主题 `setAuthDialogTheme`

方法描述

设置授权页在服务条款未选中时的弹窗主题相关。

```
setAuthDialogTheme(String authDialogBg, int authDialogwidth, int authDialogHeight,int authDialogx, int authDialogY,boolean isAuthDialogBottom,boolean authDialogCanceledOnTouchOutside, boolean isAuthPrivacyDialog)
```

参数说明

参数	参数类型	说明	默认值
authDialogBg	String	设置弹窗背景(资源id)	null
authDialogWidth	int	设置弹窗宽度	null
authDialogHeight	int	设置弹窗高度	null
authDialogX	int	设置弹窗X偏移量（以屏幕心为原点）	null
authDialogY	int	设置弹窗Y偏移量（以屏幕心为原点）	null
isAuthDialogBottom	boolean	设置弹窗是否居底	null
authDialogCanceledOnTouchOutside	boolean	设置点击弹窗外部区域是否可以关闭弹窗	null
isAuthPrivacyDialog	boolean	设置是否启用授权页服务条款弹窗	null

- 授权页在服务条款未选中时的弹窗标题 `setAuthDialogTitle`

方法描述

设置授权页在服务条款未选中时的弹窗标题相关属性。

```
setAuthDialogTitle(String authDialogTitleText, int authDialogTitleColor,int authDialogTitleSize, Typeface authDialogTitleTypeface)
```

参数说明

参数	参数类型	说明	默认值
authDialogTitleText	String	设置弹窗标题文字	null
authDialogTitleColor	int	设置弹窗标题文字颜色(资源id或色值)	null
authDialogTitleSize	int	设置弹窗标题文字大小	17
authDialogTitleTypeface	Typeface	设置弹窗标题字体	null

- 授权页在服务条款未选中时的弹窗富文本 `setAuthDialogContent`

方法描述

设置授权页在服务条款未选中时的弹窗富文本（协议条款内容）相关属性。

```
setAuthDialogContent(int authDialogContentSize, Typeface authDialogContentTypeface)
```

参数说明

参数	参数类型	说明	默认值
authDialogContentSize	String	设置弹窗富文本文字大小	12
authDialogContentTypeface	Typeface	设置弹窗富文本字体	null

- 授权页在服务条款未选中时的同意按钮 `setAuthDialogAgreeBtn`

### 方法描述

设置授权页在服务条款未选中时的弹窗同意按钮相关属性。

```
setAuthDialogAgreeBtn(String authDialogAgreeBtnBg, String authDialogAgreeBtnText,int authDialogAgreeTextColor, int authDialogAgreeTextSize,Typeface authDialogAgreeTextTypeface)
```

### 参数说明

参数	参数类型	说明	默认值
authDialogAgreeBtnBg	String	设置弹窗同意按钮背景	null
authDialogAgreeBtnText	String	设置弹窗同意按钮文字	null
authDialogAgreeTextColor	int	设置弹窗同意按钮文字颜色(资源id或色值)	null
authDialogAgreeTextSize	int	设置弹窗同意按钮文字大小	14
authDialogAgreeTextTypeface	Typeface	设置弹窗同意按钮文字字体	null

- 授权页在服务条款未选中时的同意按钮 `setAuthDialogDisagreeBtn`

### 方法描述

设置授权页在服务条款未选中时的弹窗不同意按钮相关属性。

```
setAuthDialogDisagreeBtn(String authDialogDisagreeBtnBg, String authDialogDisagreeBtnText,int authDialogDisagreeTextColor, int authDialogDisagreeTextSize,Typeface authDialogDisagreeTextTypeface)
```

### 参数说明

参数	参数类型	说明	默认值
authDialogDisagreeBtnBg	String	设置弹窗不同意按钮背景	null
authDialogDisagreeBtnText	String	设置弹窗不同意按钮文字	null
authDialogDisagreeTextColor	int	设置弹窗不同意按钮文字颜色(资源id或色值)	null
authDialogDisagreeTextSize	int	设置弹窗不同意按钮文字大小	14
authDialogDisagreeTextTypeface	Typeface	设置弹窗不同意按钮文字字体	null

方法描述

设置授权按钮监听事件。

```
setOneLoginAuthListener(OneLoginAuthListener oneLoginAuthListener)
```

参数说明

参数	参数类型	说明	默认值
oneLoginAuthListener	OneLoginAuthListener	授权按钮监听	null

- OneLoginAuthListener 接口方法 void onOLoginAuthListener(Context context, OneLoginAuthCallback oneLoginAuthCallback)，通过 OneLoginAuthCallback 的回调 onOLoginAuthCallback(boolean o) 决定是否继续登录流程。可以实现二次弹窗确认的功能;

# 其他接口说明

## 1、横竖屏设置

- 设置横竖屏 setRequestedOrientation

方法描述

设置授权页前个页面的方向，授权页默认通过 android:screenOrientation="behind" 跟随前个页面的方向。**该方法 2.3.0 以后版本废弃**，建议直接在 AndroidManifest 中配置登录页前一个 Activity 的方向，或者调用系统api setRequestedOrientation 设置屏幕方向来替换此方法。

```
public void setRequestedOrientation(Activity activity, boolean isPortrait)
```

代码示例

```
OneLoginHelper.with().setRequestedOrientation(MainActivity.this, true);
```

参数说明

参数	类型	说明
activity	Activity	当前调用 OneLogin 的页面
isPortrait	boolean	是否是竖屏

## 2、回调接口实现

通过实现 AbstractOneLoginListener 接口，通过它的多个回调可获取预取号、拉起授权页取号时的各种返回状态与结果，每次调用拉起授权页方法前必须先传入该接口的实现，否则不能正常拉起授权页。

## 2.1、结果返回

### 方法描述

当前流程的结果返回。取号成功、取号失败、切换账号、点返回或者按返回键从授权页返回取消登录等都从该回调返回结果，具体请参考 OneLogin 返回码。

```
public onResult(JSONObject jsonObject)
```

### 参数说明

参数	类型	说明
jsonObject	JSONObject	回调信息

### jsonObject 的参数说明

- 取号成功

参数名	必须	类型	说明
msg	是	String	运营商返回的状态信息
process_id	是	String	流水号(有效期10分钟)
app_id	是	String	配置唯一 id
operator	是	String	客户端获取的运营商
clienttype	是	String	客户端, 1 表示 Android
sdk	是	String	SDK 的版本号
status	是	int	状态码, 状态码为 200
token	是	String	运营商返回的 accessToken
authcode	否	String	电信运营商返回的 authcode

- 取号失败

参数名	必须	类型	说明
errorCode	是	String	错误码
msg	是	String	运营商返回的状态信息
process_id	是	String	流水号(有效期10分钟)
app_id	是	String	后台配置唯一 id 具体的
metadata	是	JSONObject	错误原因
operator	是	String	客户端获取的运营商
clienttype	是	String	客户端, 1 表示 Android
sdk	是	String	SDK 的版本号
status	是	int	状态码, 状态码为 500

## 2.2、获取脱敏手机号

### 方法描述

获取用于界面展示的脱敏手机号

```
public void onRequestTokenSecurityPhone(String phone)
```

### 参数说明

参数	类型	说明
phone	String	脱敏手机号

## 2.3、隐私条款点击

### 方法描述

隐私条款点击回调

```
public void onPrivacyClick(String name, String url)
```

### 参数说明

参数	类型	说明
name	String	隐私条款名称
url	String	隐私条款地址

## 2.4、登录按钮点击

### 方法描述

登录按钮点击回调

```
public void onLoginButtonClick()
```

## 2.5、授权页面拉起

### 方法描述

授权页面拉起回调

```
public void onAuthActivityCreate(Activity activity)
```

### 参数说明

参数	类型	说明
activity	Activity	授权页面

## 2.6、隐私条款页面拉起

### 方法描述

隐私条款页面拉起回调

```
public void onAuthWebActivityCreate(Activity activity)
```

### 参数说明

参数	类型	说明
activity	Activity	隐私条款页面

## 2.7、隐私条款 CheckBox 点击

### 方法描述

CheckBox点击回调

```
public void onPrivacyCheckBoxClick(boolean isChecked)
```

### 参数说明

参数	类型	说明
isChecked	boolean	CheckBox 是否选择

## 2.8、一键登录加载 loading

### 方法描述

点一键登录时弹出自定义对话框式 loading 时机

```
public void onLoginLoading()
```

## 2.9、返回按钮点击

### 方法描述

点授权页面标题栏返回按钮或者手机返回键时的回调。

注: 该回调与带 -20301/-20302 返回码的 `onResult` 回调共存, 两者都会返回

```
public void onBackButtonClick()
```

## 2.10、切换账号点击

### 方法描述

点授权页面切换账号栏的回调。

注: 该回调与带 -20303 返回码的 `onResult` 回调共存, 两者都会返回

```
public void onSwitchButtonClick()
```

## 2.11、视图初始化完成

### 方法描述

授权页加载布局, 视图初始化完成后回调 (包括Dialog实现的授权页)。注意, Activity实现的授权页在进入或者退出分屏模式 SDK 会重新加载布局, 并触发回调。

```

public void onAuthActivityViewInit(TextView tvTitle, TextView tvNumber, TextView
tvSwitchAccount,
                                TextView tvLogin, TextView tvSlogan, TextView
tvPrivacy,
                                CheckBox checkBox, RelativeLayout rvCheckBox,
RelativeLayout rvTitle,
                                RelativeLayout rvSecondRoot, RelativeLayout
rvMainLayout, LinearLayout lvPrivacy,
                                RelativeLayout rvLogin, RelativeLayout
rvSwitchAccount, ImageView ivBack,
                                ImageView ivLogo, LoadingImageView ivLoading,
GTGifView gifLoading,
                                GTContainerWithLifecycle bgContainer,
GTVideoView vdoViBackground,
                                ImageView imageBackground, FrameLayout
flRoot)

```

## 参数说明

参数	类型	说明
tvTitle	TextView	顶部标题栏的文本视图
tvNumber	TextView	脱敏手机号的文本视图
tvSwitchAccount	TextView	切换账号文本视图
tvLogin	TextView	登录按钮的文本视图
tvSlogan	TextView	运营商标语的文本视图（例如“中国移动提供认证服务”）
tvPrivacy	TextView	隐私条款的文本视图
checkBox	CheckBox	隐私栏勾选框
rvCheckBox	RelativeLayout	隐私栏勾选框的父视图
rvTitle	RelativeLayout	顶部标题栏布局，包含返回按钮和标题栏文本
rvSecondRoot	RelativeLayout	授权页的二级布局，除背景外其他视图都加载在此布局上
rvMainLayout	RelativeLayout	授权页除标题栏和运营商标语外其他部分的布局
lvPrivacy	LinearLayout	隐私栏布局，包含勾选框和文本
rvLogin	RelativeLayout	登录布局
rvSwitchAccount	RelativeLayout	切换账号布局
ivBack	ImageView	左上角返回按钮
ivLogo	ImageView	logo

参数	类型	说明
ivLoading	com.geetest.onelogin.view.LoadingImageView	登录按钮内部右侧的加载中图片类型自定义View
gifLoading	com.geetest.onelogin.view.GTGifView	登录按钮内部右侧的加载中gif类型自定义View
bgContainer	com.geetest.onelogin.view.GTContainerWithLifecycle	授权页的背景容器，仅当背景图片和背景视频都为空时SDK才会生成该容器
vdoViBackground	com.geetest.onelogin.view.GTVideoView	授权页背景播放视频的自定义View
imageBackground	ImageView	授权页加载背景图片资源或gif资源的视图，当背景为视频资源时此视图加载一张纯白背景
flRoot	FrameLayout	授权页的根布局

## 2.12、申请权限结果回调

### 方法描述

动态申请权限时收到申请的结果回调

```
public void onRequestPermissionsResult(int requestCode, String[] permissions,
int[] grantResults)
```

### 参数说明

参数	类型	说明
requestCode	int	请求码
permissions	String[]	权限数组
grantResults	int[]	权限申请结果

## 2.13、其他验证方式

### 方法描述

用户点击授权按钮后先执行其他验证方式

```
public boolean onRequestOtherVerify()
```

### 返回值说明

类型	说明	默认值
boolean	返回 true 表示点击授权按钮后先执行其他验证方式，可在此回调中执行接入方的逻辑，验证通过后再调用 requestTokenDelay 取号获取token。返回 false 表示不使用其他验证方式，点击授权按钮后直接进行取号	false

## 2.14、onAuthActivityResult 回调

### 方法描述

完全透传授权页收到的 onAuthActivityResult 回调的参数。

```
public void onAuthActivityResult(int requestCode, int resultCode, Intent data)
```

### 参数说明

参数	类型	说明
requestCode	int	请求码
resultCode	int	结果码
data	Intent	回调数据

## 2.15、onAuthDialogCreate 回调

### 方法描述

Dialog实现的授权页 onCreate 时回调

```
public void onAuthDialogCreate(Dialog dialog)
```

### 参数说明

参数	类型	说明
dialog	Dialog	授权页对话框实例

## 2.16、actionBeforeAuthDialogShow 回调

### 方法描述

Dialog实现的授权页在 show() 方法执行前回调

```
public void actionBeforeAuthDialogShow(Dialog dialog)
```

### 参数说明

参数	类型	说明
dialog	Dialog	授权页对话框实例

## 2.17、actionAfterAuthDialogShow 回调

### 方法描述

Dialog实现的授权页在 show() 方法执行后回调

```
public void actionAfterAuthDialogShow(Dialog dialog)
```

### 参数说明

参数	类型	说明
dialog	Dialog	授权页对话框实例

## 2.18、onAuthDialogFocusChanged 回调

### 方法描述

Dialog实现的授权页在 onWindowFocusChanged(boolean hasFocus) 调用时回调

```
public void onAuthDialogFocusChanged(Dialog dialog, boolean hasFocus)
```

### 参数说明

参数	类型	说明
dialog	Dialog	授权页对话框实例
hasFocus	boolean	是否有焦点

## 2.19、actionAfterStartActivityCalled 回调

### 方法描述

启动授权页时，调用 startActivity 后立即回调。接入方可以在此回调中找到当前的Activity调用 overridePendingTransition执行授权页的入场动画。

```
public void actionAfterStartActivityCalled()
```

## 2.20、onAuthDialogDisagreeClick 回调

### 方法描述

授权页未选中时隐私协议的弹框 disagree 点击事件回调

```
public void onAuthDialogDisagreeClick(Dialog dialog)
```

### 参数说明

参数	类型	说明
dialog	Dialog	授权页服务条款弹框实例

## 2.21、onAuthDialogOutsideClick 回调

### 方法描述

授权页未选中时隐私协议的弹框外部区域点击事件回调

```
public void onAuthDialogOutsideClick()
```

## 3、日志打印

- 日志打印控制 `setLogEnable`

### 方法描述

设置是否开启 SDK 日志打印功能，默认是开启的，开启后 SDK 运行过程中会打印以 Geetest\_OneLogin 为 TAG 的 logcat 日志。正式上线应用可以调用该接口关闭日志。

```
public OneLoginHelper setLogEnable(boolean logEnable)
```

### 参数说明

参数	类型	说明
logEnable	boolean	是否开启日志

### 代码示例

```
OneLoginHelper.with().setLogEnable(false);
```

注：开启日志后，Release 模式也会打印 SDK 内日志

## 4、超时时间

- 设置超时时间 `setRequestTimeout`

### 方法描述

分别设置预取号和取号请求的超时时间，需要在调用预取号之前设置。`register` 方法和 `preGetToken` 已标记过期的方法中传递超时时间会统一设置预取号和取号超时时间，建议迁移到另一个重载的方法。

```
public void setRequestTimeout(@IntRange(from = 1000, to = 15000) int
preGetTokenTimeout,
                                @IntRange(from = 1000, to = 15000) int
requestTokenTimeout)
```

#### 参数说明

参数	类型	说明	默认值
preGetTokenTimeout	int	预取号请求的超时时间，单位:ms，取值范围:1000~15000	8000
requestTokenTimeout	int	取号请求的超时时间，单位:ms，取值范围:1000~15000	8000

#### 代码示例

```
OneLoginHelper.with().setRequestTimeout(8000, 8000);
```

## 5、预取号状态判断

- 取号是否有效 isPreGetTokenResultValidate

#### 方法描述

判断是否预取号结果是否有效，预取号失败、预取号结果超期、预取号结果已使用情况均返回预取号结果失效。

```
public boolean isPreGetTokenResultValidate()
```

#### 代码示例

```
OneLoginHelper.with().isPreGetTokenResultValidate();
```

## 6、SDK 版本号

- 获取 SDK 版本号 sdkVersion

#### 方法描述

获取 SDK 版本号

```
public void sdkVersion()
```

#### 代码示例

```
OneLoginHelper.with().sdkVersion()
```

## 7、初始化状态判断

- 初始化是否成功 `isInitSuccess`

### 方法描述

判断是否初始化成功，cancel 之后必须重新初始化。

```
public boolean isInitSuccess()
```

### 代码示例

```
OneLoginHelper.with().isInitSuccess();
```

## 8、隐私条款是否勾选

- 隐私条款是否勾选 `isPrivacyChecked`

### 方法描述

判断是否勾选同意隐私条款声明。

```
public boolean isPrivacyChecked()
```

### 代码示例

```
OneLoginHelper.with().isPrivacyChecked();
```

## 9、改变隐私条款勾选框的选中状态

- 改变隐私条款勾选框的选中状态 `setProtocolCheckState`

### 方法描述

动态改变隐私条款勾选框的选中状态。

```
public void setProtocolCheckState(boolean checked)
```

### 参数说明

参数	类型	说明
checked	boolean	是否选中

### 代码示例

```
OneLoginHelper.with().setProtocolCheckState(true);
```

## 10、关闭授权页加载动画

- 关闭授权页面的 loading `stopLoading`

#### 方法描述

关闭授权页面的 loading，在校验方法中实现

```
public void stopLoading()
```

#### 代码示例

```
OneLoginHelper.with().stopLoading();
```

## 11、隐私页 WebView 配置

- 配置隐私页 WebView 控件的 WebViewClient 属性 `setWebViewClient`

#### 方法描述

配置隐私页 WebView 控件的 WebViewClient 属性，用以自定义处理 WebView 加载隐私后的事件回调

```
public void setWebViewClient(WebViewClient webViewClient)
```

#### 代码示例

```
OneLoginHelper.with().setWebViewClient(new WebViewClient() {  
    @Override  
    public boolean shouldOverrideUrlLoading(WebView view, WebResourceRequest  
request) {  
        return false;  
    }  
  
    @Override  
    public void onReceivedSslError(WebView view, SslErrorHandler handler,  
SslError error) {  
        handler.proceed();  
    }  
});
```

## 12、SDK 内存清理

- 释放 SDK 所有内部引用 `cancel`

#### 方法描述

一键登录结束后，如果不需要继续使用一键登录相关功能，可以调用 cancel 方法释放 SDK 内部的引用，减少内存泄漏。**调用该方法后，如需再次登录，需要重新预取号。**

```
public void cancel()
```

注：该方法会清理掉 SDK 内部核心单例并置空，导致部分缓存失效，部分接口耗时增加，建议酌情使用。

#### 代码示例

```
OneLoginHelper.with().cancel();
```

- 释放 SDK 持有的 listener 引用 `removeOneLoginListener`

#### 方法描述

一键登录结束后，可以调用 `removeOneLoginListener` 释放 `preGetToken` 与 `requestToken` 传入的 listener 参数，**防止 listener 持有外部类的引用导致内存泄漏**。

```
public void removeOneLoginListener()
```

#### 代码示例

```
OneLoginHelper.with().removeOneLoginListener();
```

## 13、延迟取号

#### 方法描述

其他验证方式验证通过后调用此方法继续取号流程

```
public void requestTokenDelay()
```

#### 代码示例

```
OneLoginHelper.with().requestTokenDelay();
```

## 14、删除预取号的缓存

#### 方法描述

在预取号处于成功态时改为失败态，并删除内存中预取号获得的相关数据

```
public void deletePreResultCache()
```

#### 代码示例

```
OneLoginHelper.with().deletePreResultCache();
```

## 15、获取当前的网络类型和运营商类型

#### 方法描述

获取当前的网络类型和运营商类型

```
public String getCurrentNetworkInfo(Context context)
```

#### 代码示例

```
JSONObject currentNetworkInfo =  
OneLoginHelper.with().getCurrentNetworkInfo(mContext);
```

#### 参数说明

参数	类型	说明
context	Context	上下文

#### 返回值说明

类型	说明
JSONObject	json对象

#### jsonObject 的参数说明

参数名	必须	类型	说明
operatorType	是	String	运营商类型。0：未知 1：中国移动 2：中国联通 3：中国电信
networkType	是	String	网络类型。0：未知 1：流量 2：wifi 3：数据流量+wifi

## 16、更新授权按钮的文本

#### 方法描述

更新授权按钮的文本，包含内容、样式等

```
public void updateAuthButtonTitle(CharSequence authButtonTitle)
```

#### 代码示例

```
OneLoginHelper.with().updateAuthButtonTitle(checked?"一键登录":"先查看并同意服务条款");
```

## OnePass（本机号码认证）

### 1、调用逻辑

- `init()` 初始化 SDK
- `getToken()` 获取本机号码对应的Token

### 2、接口配置

#### 2.1、初始化

##### 方法描述

SDK 初始化接口，在项目的具体页面的 `onCreate` 方法里面进行初始化。

```
@Deprecated
public OnePassHelper init(Context context);

public OnePassHelper init(Context context, String appId, int timeout);
```

参数说明

参数	类型	说明
context	Context	上下文
appld	String	配置唯一产品 APPID，OneLogin 与 OnePass属于不同的产品，注意产品 APPID 不可混用
timeout	int	超时时间，单位:ms，取值范围: 1000~15000，默认 8000

代码示例

```
OnePassHelper.with().init(MainActivity.this, appId, 8000);
```

2.2、本机号校验接口

方法描述

在点击登录时调用此接口获取认证传入的手机号码是否为本机号码、是否与本机 SIM 卡一致（对于双卡手机，则认证是否为当前流量对应的本机号码）需要的 token，具体认证需调用服务端接口，传入该接口获取到的 token 进行验证。

```
@Deprecated
public void getToken(String phone, String appId, OnePassListener onePassListener);

public void getToken(String phone, OnePassListener onePassListener);
```

参数说明

参数	类型	说明
phone	String	需进行认证的手机号码
appld	String	配置唯一产品 APPID，OneLogin 与 OnePass 属于不同的产品，注意产品APPID 不可混用
onePassListener	OnePassListener	回调监听器，需要开发者自己实现

示例代码

```
OnePassListener onePassListener = new OnePassListener() {
    @Override
    public void onTokenFail(JSONObject jsonObject) {
        Log.i(Constants.TAG, "onTokenFail:" + jsonObject.toString());
        Utils.showToastMessage(OnePassActivity.this, "本机认证失败");
        progressDialog.dismiss();
    }

    @Override
    public void onTokenSuccess(final JSONObject jsonObject) {
        Log.i(Constants.TAG, "onTokenSuccess:" + jsonObject.toString());
        try {
```

```

        jsonObject.put("id_2_sign", Constants.APPID);
        jsonObject.put("phone", phoneNumber);
    } catch (JSONException e) {
        e.printStackTrace();
    }
    Log.i(Constants.TAG, "开始请求 checkgateway:" + jsonObject.toString());
    //请求服务端接口校验手机号码
    CheckGatewayTask checkGatewayTask = new
    CheckGatewayTask(OnePassActivity.this, jsonObject, progressDialog);
    checkGatewayTask.execute(Constants.CHECK_GATE_WAY);
}
};

// 参数一为本机手机号
// 参数二为本机号码认证的回调接口实现
OnePassHelper.with().getToken(phone, onePassListener);

```

## 2.3、SDK 内存清理

### 方法描述

在页面关闭的时候执行此方法。执行此方法后，会释放 SDK 所有内部引用。**调用该方法后，如需再次校验，需要重新初始化。**

```
public void cancel();
```

### 代码示例

```

@Override
protected void onDestroy() {
    super.onDestroy();
    OnePassHelper.with().cancel();
}

```

## 3、OnePassListener 回调接口实现

### 3.1 校验成功

#### 方法描述

```
public void onTokenSuccess(JSONObject jsonObject)
```

#### 参数说明

参数	类型	说明
jsonObject	JSONObject	回调信息

jsonObject 的参数说明

参数名	必须	类型	说明
process_id	是	String	流水号(有效期10分钟)
accesscode	是	String	运营商返回的 accesscode
phone	是	String	待认证的手机号码

返回数据示例：

```
{
  "process_id" : "9dca9ca39a4ec4b404f2f042b3c25d1b", // 流水号
  "accesscode" :
  "CM__1__3996159873d7ccc36f46803b88dda97a__2.5.3.1__STsid0000001631534638077NnrwI
  DzIvuCswl0wviUEVjnFGU4skzvX", // accesscode
  "phone" : "15012345678" // 待认证的手机号码
}
```

### 3.2 校验失败

方法描述

```
public void onTokenFail(JSONObject jsonObject)
```

参数说明

参数	类型	说明
jsonObject	JSONObject	回调信息

jsonObject 的参数说明

参数名	必须	类型	说明
process_id	是	String	流水号(有效期10分钟)
code	是	String	错误码
custom_id	是	String	appld
metadata	是	String	错误描述信息的 json 字符串
real_op	是	String	sdk检测到的运营商类型
op	是	String	本次校验使用的运营商类型
clienttype	是	String	客户端类型，1表示Android
sdk	是	String	sdk版本号

返回数据示例：

```
{
  "process_id" : "d705fb57473e4d7b0e9911243a06fae4", //流水号
  "code" : "-20203", //错误码
  "custom_id" : "3996159873d7cdc36f25803b88dea95a", //appid
  "metadata" : {
    "error_data" : "Currently getting operators error:unknown" //错误描述
  },
  "real_op": "unknown", //sdk检测到的运营商类型
  "op": "unknown", //本次校验使用的运营商类型
  "clienttype": "1", //客户端类型, 1表示Android
  "sdk": "2.5.3.1" //sdk版本号
}
```

## 4、其他接口说明

### 4.1、配置是否缓存手机号

- 配置 SDK 是否缓存手机号到本地 `setCacheNumberEnable`

#### 方法描述

获取 token 之前调用该接口配置是否开启缓存手机号功能。配置开启后，每次调用 `getToken` 方法时，会加密保存当前传入的手机号到本地配置，用于减少用户登录页重复输入手机号流程，提升登录效率；配置关闭时，则会清除本地缓存的手机号，并且调用 `getToken` 方法时，不会记录手机号。

```
public OnePassHelper setCacheNumberEnable(boolean enable)
```

注：配置开启后，**手机号仅加密缓存到本地，每次 `getToken` 后都会存储或者覆盖更新，最多保存 100 个手机号码，不会搜集与上传。**

#### 参数说明

参数	类型	说明	默认值
enable	boolean	是否开启手机号缓存功能	true

#### 代码示例

```
OnePassHelper.with().setCacheNumberEnable(true);
```

### 4.2、获取最近一条缓存的手机号

- 获取本地缓存的手机号 `getCachedNumber`

#### 方法描述

开启缓存手机号功能后，每次进入登录页面，可调用该方法获取上一次调用 `getToken` 时缓存的手机号。未开启手机号缓存、首次调用、内部解密失败或其他异常时，返回结果为空字符串。

```
public String getCachedNumber()
```

#### 代码示例

```
String number = OnePassHelper.with().getCachedNumber();
```

## 4.3、获取匹配的手机号列表

- 获取匹配的手机号列表 `getCachedNumbers`

### 方法描述

开启缓存手机号功能后，在登录页面如需根据输入动态匹配缓存的多条手机号记录，可调用该方法并传入输入内容进行实施匹配，匹配多条缓存的手机号记录。未开启手机号缓存、首次调用、内部解密失败或其他异常时，返回结果为空 `ArrayList`。

```
public List<String> getCachedNumbers(String input)
```

### 代码示例

```
List<String> numbers = OnePassHelper.with().getCachedNumbers("132");
```

注：匹配规则为，缓存的手机号“包含”输入字符串即为匹配。

如，已缓存了：`{`13288888888`,`13288886666`,`13266666666`}`

输入>132

输出>`{`13288888888`,`13288886666`,`13266666666`}`

输入>8888

输出>`{`13288888888`,`13288886666`}`

输入>188 //未匹配到任何手机号，返回空列表

输出>`{}`;