

# Android v2.2.0.6 免密&本机

---

## 一.准备工作

概述

创建应用

快速体验demo

开发环境搭建

## 二.SDK使用说明

1.初始化

2.预取号

3.拉起授权页

4.销毁授权页

5.置换手机号

## 三.授权页界面修改

1.设计规范

2.授权页配置

3.添加自定义控件

## 四.本机认证

1.初始化

2.本机号校验

3.校验手机号

## 五.返回码

# 一.准备工作

## 概述

本文是闪验SDK\_Android本机号认证整合版本的接入文档，用于指导SDK的使用方法，默认读者已经熟悉 IDE（Eclipse 或者 Android Studio）的基本使用方法，以及具有一定的 Android 编程知识基础。

在对接之前您需要花5–10分钟阅读以下条目，可减少对接过程中的问题

- 常见问题请移步至「[对接问题](#)」
- 崩溃问题请移步至「[崩溃问题](#)」
- 升级版本请移步至「[升级指南](#)」
- 下载SDK资源请移步至「[资源下载](#)」

前置条件

- 闪验SDK支持minSdkVersion 16及以上版本。
- 闪验SDK支持中国移动3/4G、联通3/4G、电信4G的取号能力，在3G网络下时延会更高
- 闪验SDK支持单数据网络/数据网络与WiFi网络双开，不支持单WiFi网络
- 对于双卡手机，闪验SDK只对当前流量卡取号，双卡均未开数据流量SDK将会返回错误码。

## 创建应用

应用的创建流程及APPID/APPKEY的获取，请查看「[账号创建](#)」文档

**注意：如果应用有多个包名或签名不同的马甲包，须创建多个对应包名和签名的应用，否则马甲包将报包名或签名校验不通过。**

## 快速体验demo

- Android压缩包附带的apk文件夹中是闪验demo的安装包，可以直接安装到Android手机上。并快速体验闪验在您的手机上的表现。
- Android压缩包附带的demo文件夹中是闪验的示例工程，使用Android studio打开示例工程，完成以下步骤配置，然后直接运行起来测试。
  - a.将build里面的applicationId换成对应的测试包名
  - b.将签名配置改成您的签名配置
  - c.将AppId和AppKey换成您在闪验平台创建应用后生成的信息

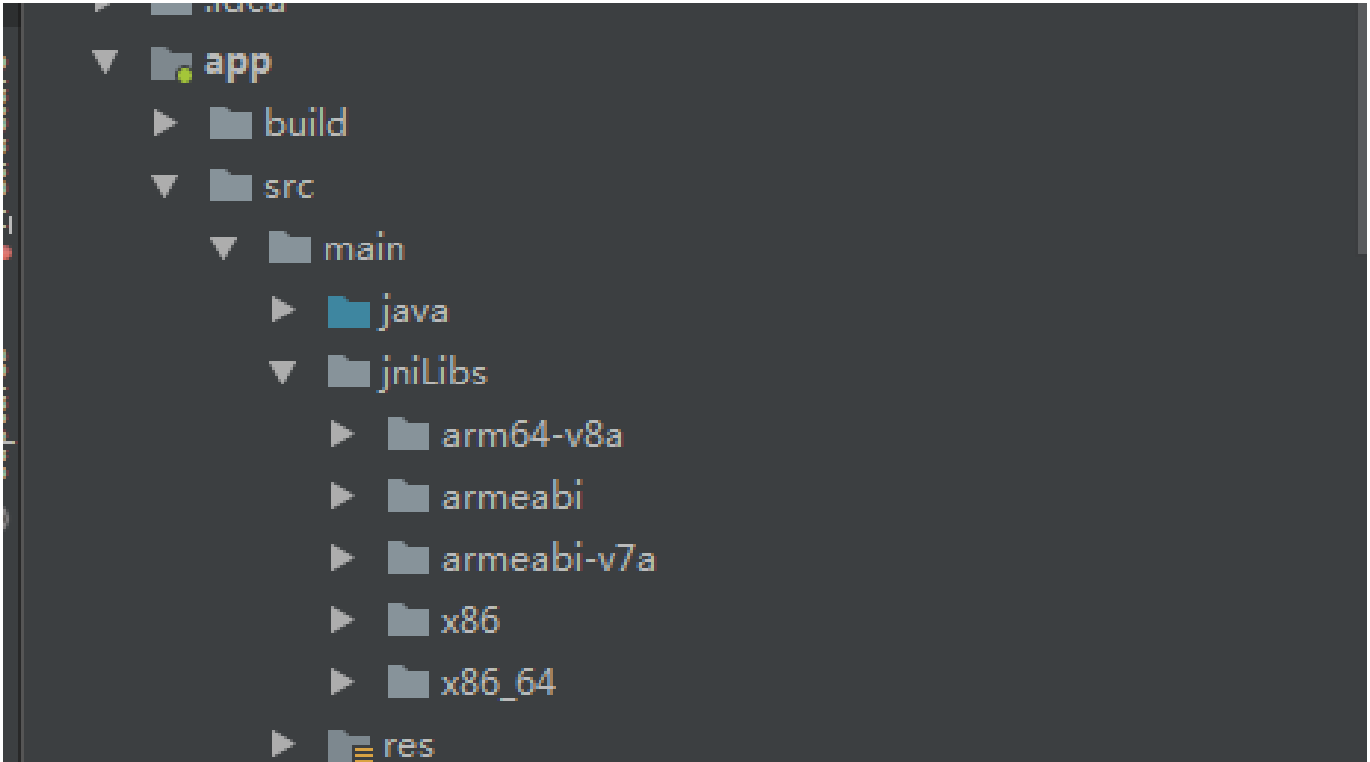
## 开发环境搭建

(1) 将开发包拷贝到工程

- a.将SDK中libs目录下的aar包拷贝到自己工程的libs目录下，如没有该目录需新建。
- b.SDK中jniLibs目录下包含多个so库，分别支持armeabi，armeabi-v7a，arm64-v8a，x86，x86\_64等cpu架构，请根据项目情况，选择相应的so库。如果您的项目包含某个abi目录，则复制对应的so文件到您

的项目，例如，您的项目中只有armeabi-v7a目录，则只复制jniLibs中的armeabi-v7a文件到您的项目；如果您的项目没有abi目录，请自行创建并复制。

jniLibs所在目录结构如下图：



在app文件夹下的build.gradle的dependencies中配置对应版本的aar依赖并添加repositories，详细代码如下：

```
repositories {  
    flatDir {  
        dirs '../app/libs'  
    }  
}  
  
dependencies {api(name: 'aar包名', ext: 'aar')}
```

## (2) 配置AndroidManifest.xml文件

在manifest标签内添加必要的权限支持

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

建议的权限：如果选用该权限，需要在预取号步骤前提前动态申请。

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

建议开发者申请本权限，本权限只用于移动运营商在双卡情况下，更精准的获取数据流量卡的运营商类型，  
缺少该权限，存在取号失败概率上升的风险。

#### 配置权限说明

权限名称	权限说明	使用说明
INTERNET	允许应用程序联网	用于访问网关和认证服务器
ACCESS_WIFI_STATE	允许访问WiFi网络状态信息	允许程序访问WiFi网络状态信息
ACCESS_NETWORK_STATE	允许访问网络状态	区分移动网络或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，强行切换使用数据网络
READ_PHONE_STATE	允许读取手机状态	获取IMSI用于判断双卡和换卡（可选）
WRITE_SETTINGS	允许读写系统设置项	6.0以下添加，可增加电信成功率，6.0以上可去除
GET_TASKS	允许应用程序访问TASK	

在application标签内配置授权登录activity

```

<!-- *****联通授权页activity*****-->
<activity
    android:name="com.sdk.mobile.manager.login.cucc.OauthActivity"
    android:launchMode="singleTop"
    android:screenOrientation="portrait" />
<!-- *****移动授权页activity*****-->
<activity
    android:name="com.cmik.sso.sdk.activity.LoginAuthActivity"
    android:launchMode="singleTop"
    android:screenOrientation="portrait" />
<!-- *****电信授权页activity*****-->
<activity
    android:name="com.chuanglan.shanyan_sdk.view.ShanYanOneKeyActivity"
    android:launchMode="singleTop"
    android:screenOrientation="portrait" />
<!-- *****协议页activity*****-->
<activity
    android:name="com.chuanglan.shanyan_sdk.view.CTCCPrivacyProtocolActivity"
    android:launchMode="singleTop"
    android:screenOrientation="portrait" />

```

配置Android9.0对http协议的支持两种方式：

方式一：

```

android:usesCleartextTraffic="true"

```

示例代码：

```

<application
    android:name=".view.MyApplication"
    android:allowBackup="true"

```

```
android:icon="@mipmap/shanyan_sdk_icon"
android:label="@string/app_name"
android:roundIcon="@mipmap/shanyan_sdk_icon"
android:supportsRtl="true"
android:theme="@style/AppTheme"
android:usesCleartextTraffic="true"
</application>
```

方式二：

目前只有移动运营商个别接口为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：cmpassport.com

(3) 混淆规则：

```
-dontwarn com.cmic.sso.sdk.**
-dontwarn com.sdk.**
-keep class com.cmic.sso.sdk.**{*;}
-keep class com.sdk.** { *;}
-keep class cn.com.chinatelecom.account.api.**{*;}
```

通过上面的几个步骤，工程就配置完成了，接下来就可以在工程中使用闪验SDK进行开发了。

## 二.SDK使用说明

### 1.初始化

使用一键登录功能前，必须先进行初始化操作，**放在Application的onCreate()方法中（必须放在主线程）**

方法原型

```
public void init(Context context, String appId, String appKey, InitListener initListener){}
```

参数描述

参数	类型	说明
context	Context	<b>必须传ApplicationContext对象</b>
appId	String	闪验平台获取到的appId
appKey	String	闪验平台获取到的appKey
initListener	InitListener	初始化回调监听，getInitStatus是该监听唯一的抽象方法，即 void getInitStatus(int code, String result)

示例代码

```
OneKeyLoginManager.getInstance().init(getApplicationContext(), appId, appKey, new InitListener() {  
    @Override  
    public void getInitStatus(int code, String result) {  
    }  
});
```

getInitStatus(int code, String result)方法返回参数分为外层code和result，含义如下：

字段	类型	含义
code	Int	code为1022:成功；其他：失败
result	String	返回信息

## 2.预取号

在初始化成功回调里调用，可缩短拉起授权页时间，该方法会进行网络请求。

- **建议在判断当前用户属于未登录状态时使用，已登录状态用户请不要调用该方法**
- 建议在执行拉起授权登录页的方法前，提前一段时间调用预取号方法，中间最好有2-3秒的缓冲（因为预取号方法需要1~3s的时间取得临时凭证），比如放在启动页的onCreate（）方法中，或者app启动的application中的onCreate（）方法中去调用，不建议放在用户登录时和拉起授权登录页方法一起调用，会影响用户体验和成功率。
- 预取号缓存有效期：10min(电信)/30min(联通)/60min(移动)，该缓存在有效期内使用一次后失效。
- **请勿频繁的多次调用、请勿与拉起授权登录页同时和之后调用。**

方法原型：

```
public void getPhoneInfo(GetPhoneInfoListener getPhoneInfoListener){}
```

参数描述

参数	类型	说明
getPhoneInfoListene r	GetPhoneInf oListener	预取号回调监听，getPhoneInfoStatus是该监听中唯一的抽象方法，即 void getPhoneInfoStatus(int code, String result)

示例代码：

```
OneKeyLoginManager.getInstance().getPhoneInfo(new GetPhoneInfoListener(  
) {  
    @Override  
    public void getPhoneInfoStatus(int code, String result) {  
    }  
});
```

`getPhoneInfoStatus(int code, String result)` 方法返回参数分为外层code和result，含义如下：

字段	类型	含义
code	Int	code为1022：成功；其他：失败
result	String	返回信息

### 3.拉起授权页

在预取号回调成功之后调用，可以在多个需要登录的页面中调用。该方法调用成功会拉起登录界面，**已登录状态请勿调用**。注意：每次调用拉起授权页方法前必须先调用授权页配置方法，详情请看第三节中的授权页配置。



方法原型：

```
public void openLoginAuth(boolean isFinish, int time, OpenLoginAuthListener openLoginAuthListener, OneKeyLoginListener oneKeyLoginListener){}
```

参数描述

字段	类型	含义
isFinish	boolean	点击授权页一键登录按钮有回调时是否自动销毁授权页： true：自动销毁 false:不自动销毁，开发者需主动调用销毁授权页方法进行授权页销毁操作
time	int	拉起授权页超时时间，区间范围3-10（单位秒） 当初始化、预取号未成功时，超时时间稍微延后
openLoginAuthListener	OpenLoginAuthListener	拉起授权页监听，getOpenLoginAuthStatus是该接口中唯一的抽象方法，即 void getOpenLoginAuthStatus(int code,String result)
oneKeyLoginListener	OneKeyLoginListener	一键登录监听，getPhoneCode是该接口中唯一的抽象方法，即 void getOneKeyLoginStatus(int code,String result)

示例代码

```
OneKeyLoginManager.getInstance().openLoginAuth(false, 10, new OpenLoginAuthListener() {  
    @Override  
    public void getOpenLoginAuthStatus(int code, String result) {  
    }  
}, new OneKeyLoginListener() {  
    @Override  
    public void getOneKeyLoginStatus(int code, String result) {  
    }  
});
```

getOpenLoginAuthStatus(int code,String result)方法返回参数分为外层code和result，含义如下：

--	--	--

字段	类型	含义
code	int	code为1000：授权页成功拉起 其他：失败
result	String	返回信息

getOneKeyLoginStatus(int code,String result)方法返回参数分为外层code和result，含义如下：

字段	类型	含义
code	int	code为1000：成功；其他：失败
result	String	返回信息

当外层code为1000时，result的返回为

```
{
  "appId": "",
  "accessToken": "",
  "telecom": "",
  "timestamp": "",
  "randoms": "",
  "version": "",
  "sign": "",
  "device": ""
}
```

含义如下：

字段	类型	含义
appId	String	当前项目的appid
accessToken	String	用来和后台置换手机号。一次有效，有效期3min
telecom	String	当前数据流量卡的运营商类型： CMCC 移动 CTCC 电信 CUCC 联通
timestamp	String	网络时间戳 有效期3min

randoms	String	随机数
version	String	后台接口版本号
sign	String	签名
device	String	设备型号

注意：返回字段中的accessToken、sign、device字段值为base64字符串，其中包含特殊字符，在客户APP与客户后台交互时应避免使用URL拼接字符串的传值方式，会导致特殊字符丢失，最终导致请求闪验后台的置换手机号接口报“请求非法，签名校验不通过”。

## 4. 销毁授权页

### A. 授权页面自动销毁

1. 在授权登录页面，当用户主动点击左上角返回按钮时，返回码为1011，SDK将自动销毁授权页；
2. 安卓 SDK，当用户点击手机的硬件返回键（相当于取消登录），返回码为1011，SDK将自动销毁授权页
3. 当用户设置一键登录或者其他自定义控件为自动销毁时，得到回调后，授权页面会自动销毁

### B. 授权页手动销毁

1. 当设置一键登录回调为手动销毁时，请在成功获取token后继续等待，在得到服务端传回的用户信息后执行手动销毁，如果失败则选择跳入其他登录方式，并执行销毁操作
2. 当设置自定义控件为手动销毁时，在监听到回调后执行销毁操作。
3. 如在得到回调后未销毁授权页而C端用户再次进入该页面，此页面将无法响应任何按键。

**注意：在销毁过程中，不要在主线程中做耗时操作，否则可能出现主线程堵塞，授权页不能及时销毁，再次拉起授权页时会出现延迟拉起。**

方法原型

```
public void finishAuthActivity() {}
```

示例代码


```
OneKeyLoginManager.getInstance().finishAuthActivity();
```

## 5. 置换手机号

当一键登录外层code为1000时，您将获取到返回的参数，请将这些参数传递给后端开发人员，并参考「[服务端](#)」文档来实现获取手机号码的步骤。

## 三.授权页界面修改

### 1.设计规范

闪验样式	Andriod规范	iOS规范
 <p>欢迎来到闪验</p> <p>188****7669</p> <p>中国联通提供认证服务</p> <p>本机号码一键登录</p> <p>其他方式登录</p> <p>同意 <a href="#">《天翼服务及隐私协议》</a> 和 <a href="#">《用户自定义协议条款》</a>、<a href="#">《用户服务协议》</a> 并授权闪验获取本机号码</p> <p>准则：</p> <ul style="list-style-type: none"><li>1.不允许隐藏手机号</li><li>2.不允许隐藏登录按钮</li><li>3.不允许隐藏运营商协议</li><li>4.不允许隐藏运营商slogan</li></ul>	<p><b>1.状态栏</b></p> <ul style="list-style-type: none"><li>可修改文字颜色、可隐藏，不可添加自定义控件</li></ul> <p><b>2.导航栏</b></p> <p>存在返回键，可换图片，可隐藏</p> <p>可添加自定义控件</p> <p>可修改免密登录文字内容以及文字颜色，可隐藏</p> <p>可隐藏，可透明，背景颜色可修改</p> <p><b>3.LOGO/手机号</b></p> <ul style="list-style-type: none"><li>可更换LOGO图片，尺寸可自调，默认70dp</li><li>LOGO可隐藏，可上下左右调整</li><li>可修改本机号码文字颜色、大小，可上下左右调整</li><li>可修改号码栏高度，<b>不允许隐藏手机号</b></li></ul> <p><b>4.运营商slogan</b></p> <ul style="list-style-type: none"><li>可修改文字颜色、可上下左右调整</li><li><b>不可隐藏</b></li></ul> <p><b>5.按钮</b></p> <p><b>5.1 一键登录按钮</b></p> <p>本机号码一键登录</p> <ul style="list-style-type: none"><li>可修改按钮文字内容、颜色、大小</li><li>可修改按钮图片、大小，可上下左右调整</li><li>可通过图片方式修改按钮形状、文字的字样</li><li><b>需带有“登录”或“注册”字样，不可隐藏</b></li></ul> <p><b>5.2 其他方式登录</b></p> <ul style="list-style-type: none"><li>默认隐藏，可通过添加自定义控件方式添加</li></ul> <p><b>6.背景</b></p> <ul style="list-style-type: none"><li>默认白底，可通过图片方式修改；全屏显示</li></ul> <p><b>7.底部协议/授权信息</b></p> <p>同意 <a href="#">《天翼服务及隐私协议》</a> 和 <a href="#">《用户自定义协议条款》</a>、<a href="#">《用户服务协议》</a> 并授权闪验获取本机号码</p> <ul style="list-style-type: none"><li>可添加两个协议，<b>协议内容需带书名号，无法隐藏</b></li><li><b>默认居中</b></li><li><b>无法修改字体、大小</b>；可修改颜色，可上下左右调整</li><li><b>默认有复选框，可隐藏</b></li><li>可设置是否默认勾选，可设置选中或未选中时的图片</li><li>底部授权信息「并授权闪验获取本机号码」默认为获取应用application配置的label字段，可修改</li></ul>	<p><b>1.状态栏</b></p> <ul style="list-style-type: none"><li>可隐藏、修改文字颜色（黑或白），不可添加自定义控件</li></ul> <p><b>2.导航栏</b></p> <p>存在返回键，可换图片，可隐藏</p> <p>可添加自定义控件在左右两侧</p> <p>可修改免密登录文字内容以及文字颜色，可隐藏</p> <p>可隐藏，可透明，背景颜色可修改也可通过图片方式</p> <p><b>3.LOGO/手机号</b></p> <ul style="list-style-type: none"><li>可更换LOGO图片，尺寸可自调</li><li>LOGO可隐藏，可上下左右调整，可圆角</li><li>可修改本机号码字体、大小、颜色，可上下左右调整</li><li>可修改号码栏高度，<b>不允许隐藏手机号</b></li></ul> <p><b>4.运营商slogan</b></p> <ul style="list-style-type: none"><li>可修改字体、颜色、对齐方式，可上下左右调整</li><li>可修改slogan栏高度，<b>不可隐藏</b></li></ul> <p><b>5.按钮</b></p> <p><b>5.1 一键登录按钮</b></p> <p>本机号码一键登录</p> <ul style="list-style-type: none"><li>可修改按钮文字内容、字体、颜色、大小，上下左右调整</li><li>可换图片，可修改按钮背景色、圆角、边框颜色</li><li><b>需带有“登录”或“注册”字样，不可隐藏</b></li></ul> <p><b>5.2 其他方式登录</b></p> <ul style="list-style-type: none"><li>默认隐藏，可通过添加自定义控件方式添加</li></ul> <p><b>6.背景</b></p> <ul style="list-style-type: none"><li>默认白底，可直接设置背景图片；全屏显示</li></ul> <p><b>7.底部协议/授权信息</b></p> <p>同意 <a href="#">中国移动认证服务条款和测试连接A、测试连接B</a> 并授权闪验获得本机号码</p> <ul style="list-style-type: none"><li>可添加两个协议，<b>无法隐藏</b>，默认居中，第二行可设置对齐方式</li><li>可修改字体、颜色、大小，可上下调整</li><li>默认有勾选的复选框，尺寸15dp，可隐藏</li><li>可设置是否默认勾选，可设置选中或未选中时的图片</li><li>底部授权信息「并授权闪验获取本机号码」默认为获取应用application配置的label字段</li></ul>

注意：开发者不得通过任何技术手段，将授权页面的隐私栏、品牌露出内容隐藏、覆盖，对于接入闪验SDK并上线的应用，我方和运营商会对接权页面做审查，如果有出现未按要求设计授权页面，我方有权将应用的登录功能下线。

## 2.授权页配置

写在拉起授权页前，可以实现对三网运营商授权页面个性化设计，**每次调用拉起授权页的前都必须设置一次**，该方法中设置的图片**必须放在drawable文件夹下**，引号内填图片名称，图片名称不能为空，偏移量、宽高等以dp为单位。

方法原型

```
public void setAuthThemeConfig(ShanYanUIConfig shanYanUIConfig) {}
```

参数说明

参数	类型	说明
shanYanUIConfig（必填）	ShanYanUIConfig	主题配置对象，开发者在 ShanYanUIConfig.java类中调用对应的方法配置授权页中对应的元素

示例代码

```
//自定义运营商授权页界面
OneKeyLoginManager.getInstance().setAuthThemeConfig(ConfigUtils.getUiConfig(getApplicationContext()));
```

ShanYanUIConfig.java配置元素说明

方法	说明
setAuthBGImgPath	设置授权页背景图片

授权页导航栏

方法	说明
setNavColor	设置导航栏颜色

setNavText	设置导航栏标题文字
setNavTextColor	设置导航栏标题文字颜色
setNavReturnImgPath	设置导航栏返回按钮图标
setNavReturnImgHidden	设置导航栏返回按钮是否隐藏（true：隐藏；false：不隐藏）
setAuthNavHidden	设置导航栏是否隐藏（true：隐藏；false：不隐藏）
setAuthNavTransparent	设置导航栏是否透明（true：透明；false：不透明）

## 授权页logo

方法	说明
setLogoImgPath	设置logo图片
setLogoWidth	设置logo宽度
setLogoHeight	设置logo高度
setLogoOffsetY	设置logo相对于标题栏下边缘y偏移
setLogoHidden	设置logo是否隐藏（true：隐藏；false：不隐藏）
setLogoOffsetX	设置logo相对屏幕左侧X偏移

## 授权页号码栏

方法	说明
setNumberColor	设置号码栏字体颜色
setNumFieldOffsetY	设置号码栏相对于标题栏下边缘y偏移
setNumFieldWidth	设置号码栏宽度
setNumberSize	设置号码栏字体大小
setNumFieldOffsetX	设置号码栏相对屏幕左侧X偏移

## 授权页登录按钮

方法	说明
setLogBtnText	设置登录按钮文字
setLogBtnTextColor	设置登录按钮文字颜色

setLogBtnImgPath	设置授权登录按钮图片
setLogBtnOffsetY	设置登录按钮相对于标题栏下边缘Y偏移
setLogBtnTextSize	设置登录按钮字体大小
setLogBtnHeight	设置登录按钮高度
setLogBtnWidth	设置登录按钮宽度
setLogBtnOffsetX	设置登录按钮相对屏幕左侧X偏移

## 授权页隐私栏

方法	说明
setAppPrivacyOne	设置开发者隐私条款1名称和URL(名称, url)
setAppPrivacyTwo	设置开发者隐私条款2名称和URL(名称, url)
setAppPrivacyColor	设置隐私条款名称颜色(基础文字颜色, 协议文字颜色)
setPrivacyOffsetBottomY	设置隐私条款相对于授权页面底部下边缘y偏移
setPrivacyOffsetX	设置隐私条款相对屏幕左侧X偏移
setPrivacyState	设置隐私条款的CheckBox是否选中 (true: 选中; false: 未选中)
setUncheckedImgPath	设置隐私条款的CheckBox未选中时图片
setCheckedImgPath	设置隐私条款的CheckBox选中时图片
setCheckBoxHidden	设置隐私条款的CheckBox是否隐藏 (true: 隐藏; false: 不隐藏)

## 授权页slogan

方法	说明
setSloganTextColor	设置slogan文字颜色
setSloganOffsetY	设置slogan相对于标题栏下边缘y偏移
setSloganHidden	设置slogan是否隐藏 (true: 隐藏; false: 不隐藏)
setSloganOffsetBottomY	设置slogan相对屏幕底部Y偏移
setSloganOffsetX	设置slogan相对屏幕左侧X偏移
setSloganTextSize	设置slogan字体大小

- 注意：
- a.设置的**图片必须放在drawable文件夹下，引号内填图片名称，图片支持xml。**
  - b.控件X偏移如果不设置默认居中。
  - c.带虚拟键的机型在虚拟键隐藏和显示切换时设置的相对屏幕底部Y偏移会发生变化，**协议和slogan放到底部时，用相对屏幕底部Y偏移方法，放到上部用相对标题栏下边缘Y偏移方法。**
  - d.协议栏中“并授权XX获取本机号码”中的应用名称获取为null，解决方案是：**将清单文件中application标签里的lable设置配置到string.xml中，分渠道打包也是如此。**

### 3.添加自定义控件

调用该方法可实现在授权页添加自定义控件。

方法原型

```
public ShanYanUIConfig.Builder addCustomView(View view, boolean isFinish, boolean type, ShanYanCustomInterface shanYanCustomInterface) {}
```

参数说明

参数	类型	说明
view（必填）	View	自定义控件对象
isFinish（必填）	boolean	是否需要销毁授权页：true销毁 false不销毁
type（必填）	boolean	设置自定义控件的位置： true为授权页导航栏 false为授权页空白处
shanYanCustomInterface	ShanYanCustomInterface	自定义控件监听

注意：如果添加布局为自定义控件，监听实现请参考demo示例。

示例代码：

```
/**
 * 闪验三网运营商授权页配置类
```



```

*
* @param context
* @return
*/
public static ShanYanUIConfig getUiConfig(Context context) {
/*****自定义控件*****/
//其他方式登录
TextView otherTV = new TextView(context);
otherTV.setText("其他方式登录");
otherTV.setTextColor(0xff3a404c);
otherTV.setTextSize(TypedValue.COMPLEX_UNIT_SP, 13);
RelativeLayout.LayoutParams mLayoutParams1 = new RelativeLayout.LayoutParams(RelativeLayout.LayoutParams.WRAP_CONTENT, RelativeLayout.LayoutParams.WRAP_CONTENT);
mLayoutParams1.setMargins(0, AbScreenUtils.dp2px(context, 270), 0, 0);
mLayoutParams1.addRule(RelativeLayout.CENTER_HORIZONTAL);
otherTV.setLayoutParams(mLayoutParams1);
//标题栏下划线
/* View view = new View(context);
view.setBackgroundColor(0xffe8e8e8);
RelativeLayout.LayoutParams mLayoutParams3 = new RelativeLayout.LayoutParams(RelativeLayout.LayoutParams.MATCH_PARENT, AbScreenUtils.dp2px(context, 1));
mLayoutParams3.setMargins(0, AbScreenUtils.dp2px(context, 0), 0, 0);
view.setLayoutParams(mLayoutParams3);*/

/*****设置授权页*****/
ShanYanUIConfig uiConfig = new ShanYanUIConfig.Builder()
//授权页导航栏：
.setNavColor(Color.parseColor("#ffffff")) //设置导航栏颜色
.setNavText("") //设置导航栏标题文字
.setNavTextColor(0xff080808) //设置标题栏文字颜色
.setNavReturnImgPath("") //设置导航栏返回按钮图标
.setAuthNavHidden(true)
.setAuthBGImgPath("sy_login_bg")

//授权页logo（logo的层级在次底层，仅次于自定义控件）
.setLogoImgPath("shanyan_logo") //设置logo图片
.setLogoWidth(140) //设置logo宽度
.setLogoHeight(70) //设置logo高度

```

```

.setLogoOffsetY(230) //设置logo相对于标题栏下边缘y偏移
.setLogoHidden(false) //是否隐藏logo
.setLogoOffsetX(20)

//授权页号码栏：
.setNumberColor(0xffffffff) //设置手机号码字体颜色
.setNumFieldOffsetY(200) //设置号码栏相对于标题栏下边缘y偏移
.setNumFieldHeight(50)
.setNumFieldWidth(110)
.setNumberSize(18)
.setNumFieldOffsetX(20)

//授权页登录按钮：
.setLogBtnText("本机号码一键登录") //设置登录按钮文字
.setLogBtnTextColor(0xff492C5A) //设置登录按钮文字颜色
.setLogBtnImgPath("sysdk_login_bg") //设置登录按钮图片
.setLogBtnOffsetY(350) //设置登录按钮相对于标题栏下边缘y偏移
.setLogBtnTextSize(15)
.setLogBtnHeight(45)
.setLogBtnWidth(ABScreenUtils.getScreenWidth(context,true)-40)

//授权页隐私栏：
//.setAppPrivacyOne("用户自定义协议条款", "https://www.253.com") //设置开发者
隐私条款1名称和URL(名称, url)
//.setAppPrivacyTwo("用户服务条款", "https://www.253.com") //设置开发者隐私条
款2名称和URL(名称, url)
.setAppPrivacyColor(0xffffffff, 0xff0085d0) // 设置隐私条款名称颜色(基础文字
颜色, 协议文字颜色)
.setPrivacyOffsetBottomY(30) //设置隐私条款相对于屏幕下边缘y偏
.setUncheckedImgPath("sy_uncheck")
.setCheckedImgPath("sy_check")
.setCheckBoxHidden(true)

//授权页slogan：
.setSloganTextColor(0xff999999) //设置slogan文字颜色
.setSloganOffsetY(245) //设置slogan相对于标题栏下边缘y偏移
.setSloganHidden(true)
// 添加自定义控件：
//其他方式登录及监听
.addCustomView(otherTV, true, false, new ShanYanCustomInterface() {

```

```

@Override
public void onClick(Context context, View view) {
    Toast.makeText(context, "其他方式登录", Toast.LENGTH_SHORT).show();
}
})
//标题栏下划线，可以不写
//.addCustomView(view, true, false, null)
.build();
return uiConfig;
}

```

注意：修改授权页需要以上方法全部配置（除隐私栏外），具体实现可以参考demo

## 四.本机认证

注：本机认证同免密登录，需要初始化，本机认证、免密登录可共用初始化，两个功能同时使用时，只需调用一次初始化即可。

### 1.初始化

同SDK使用说明-->初始化

### 2.本机号校验

在本地初始化和网络初始化执行之后调用，本机号校验界面需自行实现，可以在多个需要校验的页面中调用。

注意：该步骤为获取的token，需要将返回的参数原封不动的传到后台接口进行校验，通过接口返回确定是否为本机号码，具体请看第三节校验手机号。

方法原型：

```

public void startAuthentication(String phone, AuthenticationExecuteList
ener authenticationExecuteListener) {}

```

参数描述：

参数	类型	说明
phone（必填）	String	需要校验的手机号
authenticationExecuteListener（必填）	AuthenticationExecuteListener	本机号校验回调监听器，需要调用者自己实现； AuthenticationExecuteListener 是接口中的本机号校验参数回调接口，authenticationRespond是该接口中唯一的抽象方法，即  void authenticationRespond(int code,String result)

示例代码：

```
OneKeyLoginManager.getInstance().startAuthentication(phone, new AuthenticationExecuteListener() {  
    @Override  
    public void authenticationRespond(int code, String result) {  
    }  
});
```

authenticationRespond(int code,String result)方法返回参数分为外层code和result，含义如下：

字段	类型	含义
code	int	code为2000：成功 其他：失败
result	String	返回信息

当外层code为2000时，result的返回为

```
{  
    "appId": "",  
    "accessCode": "",  
    "mobile": "",  
    "telecom": "",
```

```
"timestamp":"","  
"randoms":"","  
"sign":"","  
"version":"","  
"device":"","  
}
```

含义如下：

字段	类型	含义
appId	String	当前项目的appid
accessToken	String	用来和后台校验手机号。一次有效，有效期3min
mobile	String	输入的手机号
telecom	String	当前数据流量卡的运营商类型： CMCC 移动 CTCC 电信 CUCC 联通
timestamp	String	网络时间戳 有效期3min
randoms	String	随机数
sign	String	签名
version	String	后台接口版本号
device	String	设备型号

注意：返回字段中的accessToken、sign字段值为base64字符串，其中包含特殊字符，在客户APP与客户后台交互时应避免使用URL拼接字符串的传值方式，会导致特殊字符丢失，最终导致请求闪验后台的校验手机号接口报“请求非法，签名校验不通过”。

### 3.校验手机号

当本机号校验外层code为2000时，您将获取到返回的参数，请将这些参数传递给后端开发人员，并参考「[服务端](#)」文档来实现校验本机号的步骤

## 五.返回码

该返回码为闪验SDK自身的返回码，请注意1003及1023错误内均含有运营商返回码，具体错误在碰到之后查阅「[返回码](#)」

返回码	返回码描述
1000	一键登录成功，解析result，可得到网络请求参数
1001	运营商通道关闭
1002	运营商信息获取失败
1003	一键登录获取token失败
1007	网络请求失败
1008	未开启网络
1009	未检测到SIM卡
1011	点击返回，用户取消免密登录
1014	SDK内部异常
1016	APPID为空
1017	APPKEY为空
1019	其他错误
1020	非三大运营商，无法使用一键登录功能
1021	运营商信息获取失败（accessToken失效）
1022	网络初始化、预取号成功
1023	预初始化失败
1025	非联通号段（目前联通号段 46001 46006 46009）
1031	请求过于频繁
2000	本机号校验返回成功，解析result，可得到网络请求参数
2001	传入的手机号phone为空
2003	本机号校验返回失败
2020	非三大运营商