

OneLogin

OneLogin 的主要外部调用接口

Macro

OneLoginUseCUSDK

是否使用联通的SDK，注释该宏定义表示不使用联通的SDK

Declaration

```
#define OneLoginUseCUSDK
```

OneLoginUseCTSDK

是否使用电信的SDK，注释该宏定义表示不使用电信的SDK

Declaration

```
#define OneLoginUseCTSDK
```

Discussion

SDK默认提供移动、联通、电信三网预取号及取号能力，若只需要其中的一个或两个运营商预取号及取号能力，可注释掉宏定义并且删除对应运营商的SDK，目前可支持的组合有：

1. 移动、联通、电信
2. 移动、联通，注释 `// #define OneLoginUseCTSDK`，剔除电信的**SDK** --
`EAccountApiSDK.framework`
3. 移动、电信，注释 `// #define OneLoginUseCUSDK`，剔除联通的**SDK** --
`account_login_sdk_noui_core.framework`
4. 移动，注释 `// #define OneLoginUseCTSDK`、`// #define OneLoginUseCUSDK`，剔除电信和联通的**SDK** -- `EAccountApiSDK.framework`、`account_login_sdk_noui_core.framework`

Protocol

userDidDismissAuthViewController

用户点击了授权页面的返回按钮

Declaration

```
- (void)userDidDismissAuthViewController;
```

userDidSwitchAccount

用户点击了授权页面的"切换账户"按钮

Declaration

```
- (void)userDidSwitchAccount;
```

Method

currentNetworkInfo

获取当前 OneLogin 可用的网络信息

Declaration

```
+ (nullable OLNetworkInfo *)currentNetworkInfo;
```

Discussion

5. 当使用的是非移动、联通、电信三大运营商, 则返回nil。
6. OneLogin 仅在大陆支持移动、联通、电信三大运营商。

Return Value

描述当前可用的网络信息

Seealso

OLNetworkInfo

registerWithAppID:

向SDK注册AppID

Declaration

```
+ (void)registerWithAppID:(NSString *)appID;
```

Parameters

| Param | Description |
|-------|----------------|
| appID | 产品id, 请在官网注册获取 |

setDelegate:

设置代理对象

Declaration

```
+ (void)setDelegate:(nullable id<OneLoginDelegate>) delegate;
```

Parameters

| Param | Description |
|--------------|-------------|
| delegat e | 代理对象 |

setRequestTimeout:

设置请求超时时长。默认8秒。

Declaration

```
+ (void)setRequestTimeout:(NSTimeInterval) timeout;
```

Parameters

| Para m | Description |
|-------------|-------------|
| timeo ut | 请求超时的时 长 |

preGetTokenWithCompletion:

预取号接口

Declaration

```
+ (void)preGetTokenWithCompletion:(void (^)(NSDictionary *sender)) completion;
```

Parameters

| Param | Description |
|----------------|--------------|
| completi on | 预取号的处理回 调 |

Discussion

调用限制说明: 在调用该方法后, 未回调之前, 再次调用该方法时, 方法会直接跳出, 不执行预取号逻辑, 并返回相关错误。

预取号成功后的有效期说明: 有效期内需要调用
requestTokenWithViewController:viewModel:completion: , 否则需要重新访问
preGetTokenWithCompletion: 。其中中国移动有效期为 1 小时, 中国联通和中国电信为 10 分钟。
回调返回数据示例:

```
// 成功的返回格式
{
  status = 200; // NSNumber, 200为成功的状态码
  processID = 47dab9b7c26629cd9bc117f88e2f9233; // NSString, 流水号
  appID = 2*****d; // NSString, 产品 ID
  operatorType = CU; // NSString, 运营商类型 (CM/CU/CT)
  msg = "\U83b7\U53d6accessCode\U6210\U529f"; // NSString, 相关的描述消息
}

// 失败的返回格式
{
  status = 500; // NSNumber, 500为失败的状态码
  processID = 47dab9b7c26629*****f88e2f9233; // NSString, 流水号
  appID = 2*****d; // NSString, 产品 ID
  operatorType = CT; // NSString, 运营商类型 (CM/CU/CT)
  errorCode = -30003, // NSNumber, 当运营商不成功时的错误码
  msg = "\U83b7\U53d6accessCode\U6210\U529f"; // NSString, 相关的描述消息
}
```

requestTokenWithViewController:viewModel:completion:

取号接口

Declaration

```
+ (void)requestTokenWithViewController:(UIViewController *)viewController viewModel:(nullable OLAuthViewModel *)viewModel completion:(void (^)(NSDictionary * _Nullable result))completion;
```

Parameters

| Param | Description |
|----------------|-------------|
| viewController | 视图控制器 |
| viewModel | 自定义的试图模型 |
| completion | 取号接口的处理回调 |

Discussion

调用限制说明: 为避免授权页面多次弹出, 在调用该方法后, 授权页面弹出后, 再次调用该方法时, 该方法会直接跳出, 不执行授权逻辑。

需要用户在弹出的页面上同意服务条款后, 才会进行免密认证。

授权页面弹出后, 需要手动调用 `[OneLogin dismissAuthViewController];` 关闭 `OLAuthViewContorller`。

token有效期: 中国移动的有效期为 2 分钟, 中国联通的为 30 分钟, 中国电信的为 30 天。

回调返回数据示例:

```
// 成功返回的格式:
{
    status = 200; // NSNumber, 200为成功的状态码
    processID = 47dab9b7c26629cd9bc117*****33; // NSString, 流水号
    appID = 2*****d; // NSString, 产品ID
    token = 62718774ad1247188bc678*****d3; // NSString, 运营商返回的accessToken, 用于查询真实
    operatorType = CU; // NSString, 运营商类型 (CM/CU/CT)
    msg = "\U83b7\U53d6accessCode\U6210\U529f"; // NSString, 运营商返回的Msg
}

// 失败返回的格式
{
    status = 500; // NSNumber, 500为失败的状态码
    processID = 47dab9b7c26629cd9bc117*****33; // NSString, 流水号
    appID = 2*****d; // NSString, 产品ID
    operatorType = CT; // NSString, 运营商类型 (CM/CU/CT)
    errorCode = -30003, // NSNumber, 运营商返回的错误码
    msg = "\U83b7\U53d6accessCode\U6210\U529f"; // NSString, 相关的描述消息
}
```

dismissAuthViewController

关闭当前的授权页面

Declaration

```
+ (void)dismissAuthViewController:(BOOL)animated completion:(void (^__nullable)(void))completion;
+ (void)dismissAuthViewController:(void (^__nullable)(void))completion;
```

Discussion

请不要使用其他方式关闭授权页面, 否则可能导致 OneLogin 无法再次调起

stopLoading

停止点击授权页面登录按钮之后的加载进度条

Declaration

```
+ (void)stopLoading;
```

enableAuthButton

enable授权页面登录按钮

Declaration

```
+ (void)enableAuthButton;
```

disableAuthButton

disable授权页面登录按钮

Declaration

```
+ (void)disableAuthButton;
```

isProtocolCheckboxChecked

服务条款左边复选框是否勾选

Declaration

```
+ (BOOL)isProtocolCheckboxChecked;
```

Return Value

服务条款左边复选框勾选状态

isPreGettedTokenValidate

预取号拿到的token是否还在有效期

Declaration

```
+ (BOOL)isPreGettedTokenValidate;
```

Return Value

YES - 还在有效期，可直接调用requestTokenWithViewController方法进行取号

NO - 已失效，需重新调用preGetTokenWithCompletion进行预取号之后再调用requestTokenWithViewController方法进行取号

sdkVersion

获取SDK版本号

Declaration

```
+ (NSString *)sdkVersion;
```

Return Value

当前的SDK版本号

setLogEnabled:

关闭日志



Declaration

```
+ (void)setLogEnabled:(BOOL)enabled;
```

Parameters

| Para m | Description |
|-------------|----------------------|
| enabl ed | YES，允许打印日志 NO，禁止打印日志 |

Discussion

OneLogin SDK内部的日志可以通过  [OneLoginSDK]  来筛选

Return Value

当前的SDK版本号

isLogEnabled

获取日志开关状态

Declaration

```
+ (BOOL)isLogEnabled;
```

Return Value

YES，允许打印日志 NO，禁止打印日志

OLAuthViewModel

授权页面自定义UI模型

Enum

OLRect

布局结构体

Declaration

```
typedef struct OLRect {  
    /**  
     * 竖屏时  
     * 导航栏隐藏时，为控件顶部到状态栏的距离；导航栏显示时，为控件顶部到导航栏底部的距离  
     * 弹窗时  
     * 为控件顶部到弹窗顶部的距离  
     */  
    CGFloat portraitTopYOffset;  
  
    /**  
     * 竖屏时  
     * 控件的x轴中点到屏幕x轴中点的距离，默认为0  
     * 弹窗时  
     * 控件的x轴中点到弹窗x轴中点的距离，默认为0  
     */  
    CGFloat portraitCenterXOffset;  
  
    /**  
     * 竖屏时  
     * 控件的左边缘到屏幕左边缘的距离，默认为0  
     * 弹窗时  
     * 控件的左边缘到屏幕左边缘的距离，默认为0  
  
     * portraitLeftXOffset与portraitCenterXOffset设置一个即可，portraitLeftXOffset优先级大于port  
     * 设置此属性时，portraitCenterXOffset属性失效  
     */  
    CGFloat portraitLeftXOffset;  
  
    /**  
     * 横屏时  
     * 导航栏隐藏时，为控件顶部到屏幕顶部的距离；导航栏显示时，为控件顶部到导航栏底部的距离  
     * 弹窗时  
     * 为控件顶部到弹窗顶部的距离  
     */  
    CGFloat landscapeTopYOffset;  
  
    /**  
     * 横屏时  
     * 控件的x轴中点到屏幕x轴中点的距离，默认为0  
     * 弹窗时  
     * 控件的x轴中点到弹窗x轴中点的距离，默认为0  
     */  
};
```



```

    */
    CGFloat landscapeCenterXOffset;

    /**
    横屏时
    控件的左边缘到屏幕左边缘的距离，默认为0
    弹窗时
    控件的左边缘到屏幕左边缘的距离，默认为0

    landscapeLeftXOffset与landscapeCenterXOffset设置一个即可，landscapeLeftXOffset优先级大于1
    设置此属性时，landscapeCenterXOffset属性失效
    */
    CGFloat landscapeLeftXOffset;

    /**
    控件大小，只有宽度、高度同时大于0，设置的size才会生效，否则为控件默认的size
    */
    CGSize size;
} OLRect;

```

Discussion

1. 若授权页面只支持竖屏，只设置竖屏方向偏移；
2. 若授权页面只支持横屏，只设置横屏方向偏移；
3. 若授权页面支持旋转自动切换横竖屏，则同时设置竖屏方向和横屏方向偏移；
4. 弹窗模式，同以上1、2、3；
5. size默认都可以不用设置，会根据字体大小自适应；
6. x轴方向偏移量有两个值可以设置，portraitCenterXOffset为控件的x轴中点到弹窗x轴中点的距离，portraitLeftXOffset为控件的左边缘到屏幕左边缘的距离，两者选其一即可。

OLAuthPopupAnimationStyle

弹窗模式时支持的动画类型

Declaration

```

/**
 * @abstract 弹窗模式时支持的动画类型
 */
typedef NS_ENUM(NSInteger, OLAuthPopupAnimationStyle)
{
    OLAuthPopupAnimationStyleCoverVertical = 0,
    OLAuthPopupAnimationStyleFlipHorizontal,
    OLAuthPopupAnimationStyleCrossDissolve,
    OLAuthPopupAnimationStyleCustom
};

```

OLLoadingViewBlock

授权页自定义Loading的Block

Declaration

```
typedef void(^OLLoadingViewBlock) (UIView *containerView);
```

Discussion

授权页自定义Loading，会在点击登录按钮之后触发 `containerView` 为loading的全屏蒙版view。请自行在 `containerView` 添加自定义loading。设置block后，默认loading将无效。

OLStopLoadingViewBlock

停止授权页自定义Loading的Block

Declaration

```
typedef void(^OLStopLoadingViewBlock) (UIView *containerView);
```

Discussion

停止授权页自定义Loading，会在调用[OneLogin stopLoading]时触发

OLCustomUIHandler

授权登录页面自定义视图

Declaration

```
typedef void(^OLCustomUIHandler) (UIView *customAreaView);
```

Discussion

`customAreaView` 为授权页面的 view，如，可将三方登录添加到授权登录页面

OLAuthViewLifeCycleBlock

授权页面视图生命周期回调

Declaration

```
typedef void(^OLAuthViewLifeCycleBlock) (NSString *viewLifeCycle, BOOL animated);
```

Discussion

`viewLifeCycle` 值为 `viewDidLoad`、`viewWillAppear`、`viewWillDisappear`、`viewDidAppear`、

`viewDidDisappear`

`animated` 是否有动画

OLClickAuthButtonBlock

点击授权页面授权按钮的回调

Declaration

```
typedef void(^OLClickAuthButtonBlock) (void);
```

OLClickCheckboxBlock

点击授权页面隐私协议前勾选框的回调

Declaration

```
typedef void(^OLClickCheckboxBlock) (BOOL isChecked);
```

OLTapAuthBackgroundBlock

点击授权页面弹窗背景的回调

Declaration

```
typedef void(^OLTapAuthBackgroundBlock) (void);
```

Property

statusBarStyle

状态栏样式。默认 `UIStatusBarStyleDefault` 。

Declaration

```
@property (nonatomic, assign) UIStatusBarStyle statusBarStyle;
```

naviTitle

授权页导航的标题。默认为空字符串。

Declaration

```
@property (nullable, nonatomic, strong) NSAttributedString *naviTitle;
```

naviBgColor

授权页导航的背景颜色。默认白色。

Declaration

```
@property (nullable, nonatomic, strong) UIColor *naviBgColor;
```

naviBackImage

授权页导航左边的返回按钮的图片。默认白色系统样式返回图片。尺寸约束为22x22。

Declaration

```
@property (nullable, nonatomic, strong) UIImage *naviBackImage;
```

naviRightControl

授权页导航右边的自定义控件。

Declaration

```
@property (nullable, nonatomic, strong) UIView *naviRightControl;
```

naviHidden

导航栏隐藏。默认不隐藏。

Declaration

```
@property (nonatomic, assign) BOOL naviHidden;
```

backButtonRect

返回按钮位置及大小，返回按钮最大size为CGSizeMake(40, 40)。

Declaration

```
@property (nonatomic, assign) CGRect backButtonRect;
```

backButtonHidden

返回按钮隐藏。默认不隐藏。

Declaration

```
@property (nonatomic, assign) OLRect backButtonHidden;
```

appLogo

授权页面上展示的图标。默认为 "OneLogin" 图标。

Declaration

```
@property (nullable, nonatomic, strong) UIImage *appLogo;
```

logoRect

Logo 位置及大小。

Declaration

```
@property (nonatomic, assign) OLRect logoRect;
```

logoHidden

Logo 图片隐藏。默认不隐藏。

Declaration

```
@property (nonatomic, assign) BOOL logoHidden;
```

logoCornerRadius

logo圆角，默认为0。

Declaration

```
@property (nonatomic, assign) CGFloat logoCornerRadius;
```

phoneNumColor

号码预览文字的颜色。默认黑色。

Declaration

```
@property (nullable, nonatomic, strong) UIColor *phoneNumColor;
```

phoneNumFont

号码预览文字的字体。默认粗体，24pt。

Declaration

```
@property (nullable, nonatomic, strong) UIFont *phoneNumFont;
```

phoneNumRect

号码预览 位置及大小，电话号码不支持设置大小，大小根据电话号码文字自适应。

Declaration

```
@property (nonatomic, assign) CGRect phoneNumRect;
```

switchButtonText

授权页切换账号按钮的文案。默认为“切换账号”。

Declaration

```
@property (nullable, nonatomic, copy) NSString *switchButtonText;
```

switchButtonColor

授权页切换账号按钮的颜色。默认蓝色。

Declaration

```
@property (nullable, nonatomic, strong) UIColor *switchButtonColor;
```

switchButtonFont

授权页切换账号的字体。默认字体，15pt。

Declaration

```
@property (nullable, nonatomic, strong) UIFont *switchButtonFont;
```

switchButtonRect

授权页切换账号按钮 位置及大小。

Declaration

```
@property (nonatomic, assign) CGRect switchButtonRect;
```

switchButtonHidden

隐藏切换账号按钮。默认不隐藏。

Declaration

```
@property (nonatomic, assign) BOOL switchButtonHidden;
```

authButtonImages

授权页认证按钮的背景图片, @[正常状态的背景图片, 不可用状态的背景图片, 高亮状态的背景图片]。默认正常状态为蓝色纯色, 不可用状态的背景图片时为灰色, 高亮状态为灰蓝色。

Declaration

```
@property (nullable, nonatomic, strong) NSArray<UIImage *> *authButtonImages;
```

authButtonTitle

授权按钮文案。默认白色的"一键登录"。

Declaration

```
@property (nullable, nonatomic, strong) NSAttributedString *authButtonTitle;
```

authButtonRect

授权按钮 位置及大小。

Declaration

```
@property (nonatomic, assign) CGRect authButtonRect;
```

authButtonCornerRadius

授权按钮圆角, 默认为0。

Declaration

```
@property (nonatomic, assign) CGFloat authButtonCornerRadius;
```

clickAuthButtonBlock

点击授权页面授权按钮的回调。

Declaration

```
@property (nullable, nonatomic, copy) OLClickAuthButtonBlock clickAuthButtonBlock;
```

sloganRect

Slogan 位置及大小。

Declaration

```
@property (nonatomic, assign) OLRect sloganRect;
```

sloganTextColor

Slogan 文字颜色。默认灰色, 12pt。

Declaration

```
@property (nonatomic, strong) UIColor *sloganTextColor;
```

sloganTextFont

Slogan字体。默认字体, 12pt。

Declaration

```
@property (nonatomic, strong) UIFont *sloganTextFont;
```

defaultCheckBoxState

授权页面上条款勾选框初始状态。默认 YES。

Declaration

```
@property (nonatomic, assign) BOOL defaultCheckBoxState;
```

checkedImage

授权页面上勾选框勾选的图标。默认为蓝色图标。推荐尺寸为12x12。

Declaration

```
@property (nullable, nonatomic, strong) UIImage *checkedImage;
```

uncheckedImage

授权页面上勾选框未勾选的图标。默认为白色图标。推荐尺寸为12x12。

Declaration

```
@property (nullable, nonatomic, strong) UIImage *uncheckedImage;
```

checkBoxSize

授权页面上条款勾选框大小。

Declaration

```
@property (nonatomic, assign) CGSize checkBoxSize;
```

privacyTermsAttributes

隐私条款文字属性。默认基础文字灰色, 条款蓝色高亮, 12pt。

Declaration

```
@property (nullable, nonatomic, strong) NSDictionary<NSAttributedStringKey, id> *privacyTermsAttributes;
```

additionalPrivacyTerms

额外的条款。默认为空。

Declaration

```
@property (nullable, nonatomic, strong) NSArray<OLPrivacyTermItem *> *additionalPrivacyTerms;
```

Seealso

[OLPrivacyTermItem](#)

termTextColor

服务条款普通文字的颜色。默认灰色。

Declaration

```
@property (nullable, nonatomic, strong) UIColor *termTextColor;
```

termsRect

隐私条款 位置及大小, 隐私条款, 宽需大于50, 高需大于20, 才会生效。

Declaration

```
@property (nonatomic, assign) OLRect termsRect;
```

auxiliaryPrivacyWords

除隐私条款外的其他文案，数组大小必须为4，元素依次为：条款前的文案、条款一和条款二连接符、条款二和条款三连接符，条款后的文案。

默认为@[@"登录即同意", @"和", @"、", @"并使用本机号码登录"]。

Declaration

```
@property (nullable, nonatomic, copy) NSArray<NSString *> *auxiliaryPrivacyWords;
```

clickCheckboxBlock

点击授权页面隐私协议前勾选框的回调。

Declaration

```
@property (nullable, nonatomic, copy) OLClickCheckboxBlock clickCheckboxBlock;
```

customUIHandler

自定义区域视图的处理block

Declaration

```
@property (nullable, nonatomic, copy) OLCustomUIHandler customUIHandler;
```

Discussion

提供的视图容器使用NSLayoutConstraint与相关的视图进行布局约束。如果导航栏没有隐藏，顶部与导航栏底部对齐，左边与屏幕左边对齐，右边与屏幕右边对齐，底部与屏幕底部对齐。如果导航栏隐藏，顶部与状态栏底部对齐，左边与屏幕左边对齐，右边与屏幕右边对齐，底部与屏幕底部对齐。

Seealso

```
typedef void(^OLCustomUIHandler) (UIView *customAreaView);
```

backgroundColor

授权页背景颜色。默认白色。

Declaration

```
@property (nullable, nonatomic, strong) UIColor *backgroundColor;
```

backgroundImage

授权页面背景图片

Declaration

```
@property (nullable, nonatomic, strong) UIImage *backgroundImage;
```

landscapeBackgroundImage

横屏模式授权页面背景图片

Declaration

```
@property (nullable, nonatomic, strong) UIImage *landscapeBackgroundImage;
```

supportedInterfaceOrientations

授权页面支持的横竖屏方向

Declaration

```
@property (nonatomic, assign) UIInterfaceOrientationMask supportedInterfaceOrientations;
```

isPopup

是否为弹窗模式

Declaration

```
@property (nonatomic, assign) BOOL isPopup;
```

popupRect

弹窗 位置及大小。弹窗模式时，x轴偏移只支持portraitLeftXOffset和landscapeLeftXOffset

Declaration

```
@property (nonatomic, assign) CGRect popupRect;
```

popupCornerRadius

弹窗圆角，默认为6

Declaration

```
@property (nonatomic, assign) CGFloat popupCornerRadius;
```

popupAnimationStyle

弹窗动画类型，当popupAnimationStyle为OLAuthPopupAnimationStyleStyleCustom时，动画为用户自定义，用户需要传一个CATransition对象来设置动画

Declaration

```
@property (nonatomic, assign) OLAuthPopupAnimationStyle popupAnimationStyle;
```

popupTransitionAnimation

弹窗自定义动画

Declaration

```
@property (nonatomic, strong) CATransition *popupTransitionAnimation;
```

closePopupImage

弹窗关闭按钮图片，弹窗关闭按钮的尺寸跟图片尺寸保持一致。弹窗关闭按钮位于弹窗右上角，目前只支持设置其距顶部偏移和距右边偏移

Declaration

```
@property (nullable, nonatomic, strong) UIImage *closePopupImage;
```

closePopupTopOffset

弹窗关闭按钮距弹窗顶部偏移

Declaration

```
@property (nonatomic, strong) NSNumber *closePopupTopOffset;
```

closePopupRightOffset

弹窗关闭按钮距弹窗右边偏移

Declaration

```
@property (nonatomic, strong) NSNumber *closePopupRightOffset;
```

canClosePopupFromTapGesture

是否需要通过点击弹窗的背景区域以关闭授权页面

Declaration

```
@property (nonatomic, assign) BOOL canClosePopupFromTapGesture;
```

tapAuthBackgroundBlock

点击授权页面弹窗背景的回调

Declaration

```
@property (nonatomic, copy) OLTapAuthBackgroundBlock tapAuthBackgroundBlock;
```

loadingViewBlock

授权页面，点击登录按钮之后的回调

Declaration

```
@property (nonatomic, copy, nullable) OLLoadingViewBlock loadingViewBlock;
```

webNaviHidden

服务条款页面导航栏隐藏。默认不隐藏

Declaration

```
@property (nonatomic, assign) BOOL webNaviHidden;
```

webNaviTitle

服务条款页面导航的标题。默认为"服务条款"，粗体、17pt

Declaration

```
@property (nullable, nonatomic, strong) NSAttributedString *webNaviTitle;
```

webNaviBgColor

服务条款页面导航的背景颜色。默认白色

Declaration

```
@property (nullable, nonatomic, strong) UIColor *webNaviBgColor;
```

notCheckProtocolHint

未勾选服务条款复选框时，点击登录按钮的提示。默认为"请同意服务条款"

Declaration

```
@property (nullable, nonatomic, copy) NSString *notCheckProtocolHint;
```

viewLifeCycleBlock

授权页面视图生命周期回调。

Declaration

```
@property (nullable, nonatomic, copy) OLAuthViewLifeCycleBlock viewLifeCycleBlock;
```

modalPresentationStyle

present授权页面时的样式，默认为UIModalPresentationFullScreen。

Declaration

```
@property (nonatomic, assign) UIModalPresentationStyle modalPresentationStyle;
```

OLNetworkInfo

Enum

OLNetworkType

设备网络的类型

Declaration

```
typedef NS_ENUM(NSInteger, OLNetworkType) {  
    /** 网络类型未知 */  
    OLNetworkTypeNone = 0,  
    /** 仅移动蜂窝数据网络 */  
    OLNetworkTypeCellular,  
    /** 仅 WIFI 网络 */  
    OLNetworkTypeWiFi,  
    /** 移动蜂窝数据网络及 WIFI 网络 */  
    OLNetworkTypeCellularAndWiFi,  
};
```

Property

carrierName

运营商名称

Declaration

```
@property (nullable, nonatomic, strong) NSString *carrierName;
```

networkType

网络类型

Declaration

```
@property (nonatomic, assign) OLNetworkType networkType;
```

Discussion

即使返回非 `OLNetworkTypeNone`，也可能因为终端用户未授权数据网络网络权限而无法访问设备的移动蜂窝数据网络

Seealso

[OLNetworkType](#)

detailNetworkType

具体的网络类型，如2G、3G、4G、WIFI

Declaration

```
@property (nonatomic, copy) NSString *detailNetworkType;
```

Discussion

即使返回非 `OLNetworkTypeNone`，也可能因为终端用户未授权数据网络网络权限而无法访问设备的移动蜂窝数据网络

OLPrivacyTermItem

隐私和条款条目

Property

termTitle

隐私和条款的标题

Declaration

```
@property (nonatomic, strong) NSString *termTitle;
```

termLink

隐私和条款的链接

Declaration

```
@property (nonatomic, strong) NSURL *termLink;
```

urlRequest

隐私和条款的链接

Declaration

```
@property (nonatomic, strong) NSURLRequest *urlRequest;
```

block

查看条款链接回调，当设置该回调时，查看服务条款时不进入SDK内部服务条款页面，由客户自定义处理

Declaration

```
@property (nonatomic, copy, nullable) OLViewPrivacyTermItemBlock block;
```

index

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

Declaration

```
@property (nonatomic, assign) NSInteger index;
```

Method

initWithTitle:linkURL:

使用标题和链接创建实例

Declaration

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

Return Value

返回新的 `OLPrivacyTermItem` 实例