## 准备工作：

各种文案以及申请账号

## SDK接入

除了导入闭包，还有lib项目

## 登录

sina的登录建议不要把对象做成全局的，好像有问题。

### 示例代码：

|  |
| --- |
| private AuthInfo mAuthInfo;  /\*\* 注意：SsoHandler 仅当 SDK 支持 SSO 时有效 \*/  private SsoHandler sinaSsoHandler;  mAuthInfo = new AuthInfo(this, SinaConstants.APP\_KEY,  SinaConstants.REDIRECT\_URL, SinaConstants.SCOPE);  sinaSsoHandler = new SsoHandler(this, mAuthInfo);  SinaOAuth.*login*(LoginActivity.**this**, sinaSsoHandler); |

### 你的凭据：

|  |
| --- |
| public interface SinaConstants {  /\*\* 当前 DEMO 应用的 APP\_KEY，第三方应用应该使用自己的 APP\_KEY 替换该 APP\_KEY \*/  public static final String APP\_KEY = "535673299";  /\*\*  \* 当前 DEMO 应用的回调页，第三方应用可以使用自己的回调页。  \*  \* <p>  \* 注：关于授权回调页对移动客户端应用来说对用户是不可见的，所以定义为何种形式都将不影响， 但是没有定义将无法使用 SDK 认证登录。  \* 建议使用默认回调页：https://api.weibo.com/oauth2/default.html  \* </p>  \*/  public static final String REDIRECT\_URL = "https://api.weibo.com/oauth2/default.html";  /\*\*  \* Scope 是 OAuth2.0 授权机制中 authorize 接口的一个参数。通过 Scope，平台将开放更多的微博  \* 核心功能给开发者，同时也加强用户隐私保护，提升了用户体验，用户在新 OAuth2.0 授权页中有权利 选择赋予应用的功能。  \*  \* 我们通过新浪微博开放平台-->管理中心-->我的应用-->接口管理处，能看到我们目前已有哪些接口的 使用权限，高级权限需要进行申请。  \*  \* 目前 Scope 支持传入多个 Scope 权限，用逗号分隔。  \*  \* 有关哪些 OpenAPI 需要权限申请，请查看：http://open.weibo.com/wiki/%E5%BE%AE%E5%8D%9AAPI  \* 关于 Scope 概念及注意事项，请查看：http://open.weibo.com/wiki/Scope  \*/  public static final String SCOPE = "email,direct\_messages\_read,direct\_messages\_write,"  + "friendships\_groups\_read,friendships\_groups\_write,statuses\_to\_me\_read,"  + "follow\_app\_official\_microblog," + "invitation\_write";  } |

### 回调接口实现：

|  |
| --- |
| \*/  public class DummySinaAuthListener implements WeiboAuthListener {    private final static String tag= "sinaauth";  private final Activity mActivity;  private Oauth2AccessToken mAccessToken;  public DummySinaAuthListener(final Activity activity) {  mActivity = activity;  mAccessToken = AccessTokenKeeper.readAccessToken(mActivity);  }  @Override  public void onComplete(Bundle values) {  Log.d(tag, "complete");  // 从 Bundle 中解析 Token    mAccessToken = Oauth2AccessToken.parseAccessToken(values);  // 从这里获取用户输入的 电话号码信息  String phoneNum = mAccessToken.getPhoneNum();  if (mAccessToken.isSessionValid()) {  // 显示 Token TODO  Log.d(tag, "check result:\r\n"+mAccessToken.toString());  // 保存 Token 到 SharedPreferences  AccessTokenKeeper.writeAccessToken(mActivity, mAccessToken);  Toast.makeText(mActivity, "授权成功", Toast.LENGTH\_SHORT).show();  Log.d(tag, "sina 授权成功");  } else {  // 以下几种情况，您会收到 Code：  // 1. 当您未在平台上注册的应用程序的包名与签名时；  // 2. 当您注册的应用程序包名与签名不正确时；  // 3. 当您在平台上注册的包名和签名与您当前测试的应用的包名和签名不匹配时。  String code = values.getString("code");  String message = "授权失败";  if (!TextUtils.isEmpty(code)) {  message = message + "\nObtained the code: " + code;  }  Toast.makeText(mActivity, message, Toast.LENGTH\_LONG).show();    Log.d(tag,message);  }  }  @Override  public void onCancel() {  Log.d(tag, "cancel");  // Toast.makeText(mActivity, "授权取消", Toast.LENGTH\_LONG).show();  }  @Override  public void onWeiboException(WeiboException e) {  Log.wtf(tag, "Auth exception : " + e.getMessage());  // Toast.makeText(mActivity, "Auth exception : " + e.getMessage(),  // Toast.LENGTH\_LONG).show();  }  } |

### 登录关键代码：

|  |
| --- |
| public static void login(final Activity activity,SsoHandler sinaSsoHandler) {  sinaSsoHandler.authorize(new DummySinaAuthListener(activity));  } |

### 返回数据示例

|  |
| --- |
| uid: 3849667183, access\_token: 2.00LSoWME0vFdPaa3a135ec63iacytC, refresh\_token: 2.00LSoWME0vFdPafbc81d6ac7vl4XYD, phone\_num: , expires\_in: 1457549998059 |

access\_token 是返回的token

refresh\_token 是返回的refresh token 下次刷新需要传入此参数

uid 用户id

remind\_in 从现在起到授权token失效的  **秒数**。

## 出错解决策略

## 分享

分享统一走OpenAPI

本篇从简，分享支持类型很多，会结合实际选取一部分，写成业务，具体等写完一起整合一份文档