

WB Android SDK 4.0 doc

All new weibo SDK update, if you are a SDK old users, please read the following before access to change logic, if you are a weibo SDK new users, please skip this step directly into to access.

Diff with old version:

[p-1]

```
android:name=".sdk.share.WbShareTransActivity">
<intent-filter>
    <action android:name="com.sina.weibo.sdk.action.ACTION_SDK_REQ_ACTIVITY" />
    <category android:name="android.intent.category.DEFAULT" />
</intent-filter>
</activity>
```

1. Delete old Activity Intent-filter in Manifest.xml --[p-1]
2. Remove SDK Net Engine. If you use the old SDK Net Engine to request a http connect for OpenApi, you need to trans it to your Engine. (Relevate Class file: `HttpManager,AsyncWeiboRunner`)
3. Added a unified portal for third-party client information collection. Please add the Application in your Application.

`WbSdk.install(Context context,AuthInfo authInfo)`

4. Delete the **IWeiboShareApi** and related operations, and the new share entry is changed to **WbShareHandler**.
5. Delete the **IWeiboShareApi** and related operations, and the new share entry is changed to **WbShareHandler**.

If you have completed the above operation, come down and officially enter the SDK's access instructions.

To access:

1.1 Certificate authority

Sina weibo the Android SDK provides developers with OAuth2.0 authorized certification, and integrated SSO login function, make the third party applications without understanding complex validation mechanism for authorization login operation, and provides the weibo sharing function, third-party applications can share directly through the microblogging families.

This document introduces three methods of authorization for sina weibo Android SDK, sharing and giving simple sample analysis to help third-party developers to integrate applications quickly.

1.2 interpretation of terms

	annotations
AppKey	Assigned to each third party applications only identity authentication, used to distinguish the source functions (access: http://open.weibo.com/development/mobile)
Scope	Scope is a function of OAuth2.0 new authorization page, through the Scope

	platform will open more weibo core functionality to developers, but also strengthen the privacy protection, enhance the user experience, users in the new page OAuth2.0 authorized has the right to choose to give the application of the function.
AccessToken	Token that indicates user identity can be acquired through SSO authorization
Oauth 2.0 Web	Authorization by WebView, return Token information, no need to install the micro-blog
SSO	The Token information is returned by invoking the micro-blog account for authorization

2: Preparation before integration

2.1 Apply for APP_KEY

The third party is required to register the application on the microblog platform and get the APP_KEY to add the application's authorization callback page (Redirect URI). Please refer to : [Weibo mobile access platform.](#)

Before integrating the SDK, you need to open the platform on weibo Register the application's package name and signature.

Note: the package name and the signature are not registered, or the inconsistency between the compile run time signature and the registered signature can lead to unauthorised (debug runs apk and the apk when published).

Application package name method: the value of applicationId in build.gradle in the engineering master modules directory, or the data of the package node in the androidmanifest.xml.

Application signature: this signature is MD5 value generated through the official signature tool (as shown in the figure below)



Download the signature software: [Download address](#)

Method of use: first of all you have to install the application of the signature, then install the tool, after installation, enter application package name, click the generate button, can obtain the MD5 signature, as shown in the figure below.

Note: the application to be signed must be installed on the device to generate

the corresponding MD5 signature. If you only fill in the signature MD5 of the release version on the development platform, make sure that the signature of the debug version is consistent with the release version, otherwise the authorization cannot be passed.

The SDK authorization and sharing must ensure that the four parameters of appKey, scope, redirectUrl, packageName, and the development of the website are consistent, or they cannot be authorized to share successfully.

2 Access

The new SDK has been uploaded to the central warehouse

Here, as an example of Android studio, set up the central repository in the build.

Gradle of the project root

```
maven { url "https://dl.bintray.com/thelasterstar/maven/" }
```

The SDK -core dependency is introduced in build. Gradle in the module directory that needs to introduce the SDK

```
55 dependencies {
56     compile 'org.jbundle.util.osgi.wrapped:org.jbundle.util.osgi.wrapped.org.apache.http.client:4.1.2'
57     compile 'com.sina.weibo.sdk:openDefault:1.0.0:openDefaultRelease@aar'
58 }
```

```
dependencies {
    compile 'com.sina.weibo.sdk:core:2.0.0:openDefaultRelease@aar'
}
```

Click the sync button and wait for the SDK library to download.

You can also download the micro blog SDK demo directly on github, and refer to the demo access plan.

Sample code analysis

1: Certificate authority

- 1) First initialize the WbSdk object (before the Application of your Application or call the Sdk functional code)

```
WbSdk.install(this,new AuthInfo(this, Constants.APP_KEY, Constants.REDIRECT_URL, Constants.SCOPE));
```

AuthInfo maintains the basic information required for authorization, APP_KEY (the unique key generated by the development platform), the Redirect URI (authorization callback), SCOPE (the permission function that needs to be requested, and the data in the default reference demo).

2) Initialize the SsoHandler object

```
mSsoHandler = new SsoHandler(WBAuthActivity.this);
```

3) Call the authorization

There are three modes of authorization in the SDK:

1: **AuthorizeClientSSo**: authorization only via the micro-blog

2: **AuthorizeWeb**: open the H5 page for authorization via the WebView

that comes with the SDK

3: **Authorize**: if you install the micro-blog user authorization via the client, if it is not authorized through the Web

```
*
* @see {@link Activity#onActivityResult}
*/
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    // SSO 授权回调
    // 重要：发起 SSO 登陆的 Activity 必须重写 onActivityResult
    if (mSsoHandler != null) {
        mSsoHandler.authorizeCallback(requestCode, resultCode, data);
    }
}
```

```

mSsoHandler = new SsoHandler(WBAuthActivity.this);
// SSO client
findViewById(R.id.obtain_token_via_sso).setOnClickListener((v) -> {
    mSsoHandler.authorizeClientSso(new SelfWbAuthListener());
});

// SSO web
findViewById(R.id.obtain_token_via_web).setOnClickListener((v) -> {
    mSsoHandler.authorizeWeb(new SelfWbAuthListener());
});

// SSO all
findViewById(R.id.obtain_token_via_signature).setOnClickListener((v) -> {
    mSsoHandler.authorize(new SelfWbAuthListener());
});

```

```

private class SelfWbAuthListener implements com.sina.weibo.sdk.auth.WbAuthListener{
    @Override
    public void onSuccess(final OAuth2AccessToken token) {
        WBAuthActivity.this.runOnUiThread(() -> {
            mAccessToken = token;
            if (mAccessToken.isSessionValid()) {
                // update Token
                updateTokenView(false);
                // save Token to SharedPreferences
                AccessTokenKeeper.writeAccessToken(WBAuthActivity.this, mAccessToken);
                Toast.makeText(WBAuthActivity.this,
                    R.string.weibosdk_demo_toast_auth_success, Toast.LENGTH_SHORT).show();
            }
        });
    }

    @Override
    public void cancel() {
        Toast.makeText(WBAuthActivity.this,
            R.string.weibosdk_demo_toast_auth_canceled, Toast.LENGTH_LONG).show();
    }

    @Override
    public void onFailure(WbConnectErrorMessage errorMessage) {
        Toast.makeText(WBAuthActivity.this, errorMessage.getErrorMessage(), Toast.LENGTH_LONG).show();
    }
}

```

3 Weibo Share

3.1 Share the app from third-party apps

3.1.1 Preparation:

1. Implement the WbShareCallback interface with the Activity of the current page.

The effect is as follows:

```

/*
public class WbShareActivity extends Activity implements OnClickListener, WbShareCallback {
    public static final String KEY_SHARE_TYPE = "key_share_type";
    public static final int SHARE_CLIENT = 1;
    public static final int SHARE_ALL_IN_ONE = 2;
}

```

2 : Initialize the WbShareHandler and register the application

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_share);
    initView();
    mShareType = getIntent().getIntExtra(KEY_SHARE_TYPE, SHARE_CLIENT);
    shareHandler = new WbShareHandler(this);
    shareHandler.registerApp();
    shareHandler.setProgressColor(0xff33b5e5);
}

@Override

```

3: share weibo

```
private void sendMultiMessage(boolean hasText, boolean hasImage) {  
  
    WeiboMultiMessage weiboMessage = new WeiboMultiMessage();  
    if (hasText) {  
        weiboMessage.textObject = getTextObj();  
    }  
    if (hasImage) {  
        weiboMessage.imageObject = getImageObj();  
    }  
    if (multiImageCheckbox.isChecked()) {  
        weiboMessage.multiImageObject = getMultiImageObject();  
    }  
    if (videoCheckbox.isChecked()) {  
        weiboMessage.videoSourceObject = getVideoObject();  
    }  
    shareHandler.sendMessage(weiboMessage, false);  
}
```

4: Handle sharing callback: override the onNewIntent method

```
@Override  
protected void onNewIntent(Intent intent) {  
    super.onNewIntent(intent);  
    shareHandler.doResultIntent(intent, this);  
}
```

Each of these four steps is likely to lead to a Shared failure

**** the following access may result in not sharing some of the description *****

Because the old version of the microblogging door side share has a certain logical flaw, under the conditions of no user login to share for the first time may lead to the third party programs to identify failure cause can't normal to share. In this particular note, for the first time sharing, you can login with SSO authorization to ensure that the user is logged in and then share it.

For details, please refer to the demo in the official github address of the weibo SDK: https://github.com/sinaweibosdk/weibo_android_sdk