

# 微博Android SDK 11.6.0使用指南

[申请你的App Key](#)

[注册应用程序的包名和签名](#)

[集成SDK](#)

[参数配置](#)

[使用微博授权](#)

[分享内容到微博](#)

## 申请你的App Key

1. 请到[微博开放平台](#)上 [我的应用](#)页面，完善身份信息，通过开发者身份认证审核。



2. 请到[微博开放平台](#)上 [移动应用](#)页面，点击立即接入，完善应信息，将该应用提交审核，只有审核通过的应用才能进行开发。

- 创建新应用



↓↓↓



↓↓↓

[首页](#) > [应用开发](#) > [创建新应用](#)

创建新应用

[查看帮助?](#)

应用名称:

该名称也用于来源显示, 不超过10个汉字或20个字母

应用分类:

客户端

手机

[查看移动客户端接入指南](#)

应用平台:

☐ iPhone

☐ Android

☐ BlackBerry

☐ Windows Phone

☐ Symbian

☐ WebOS

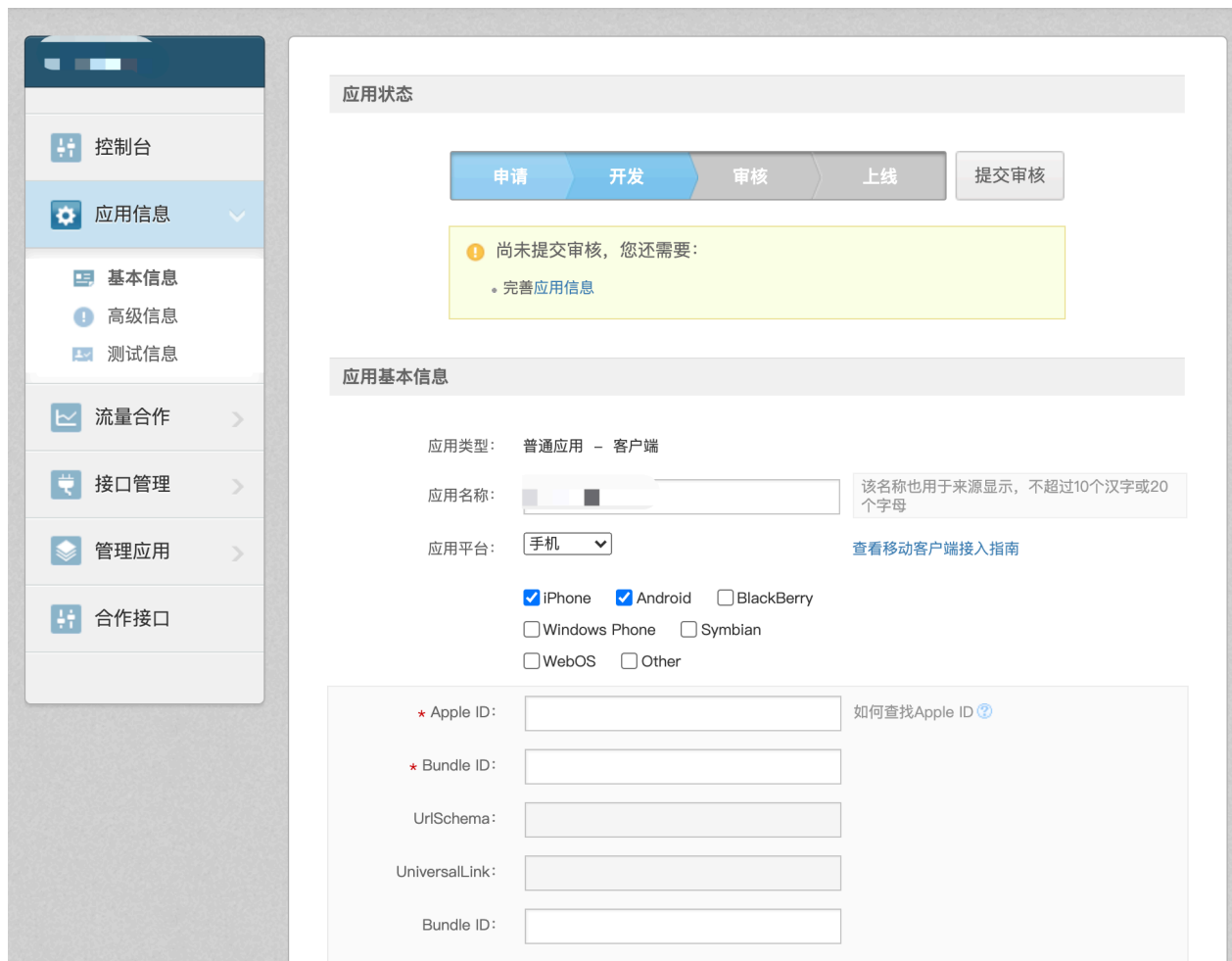
☐ Other

☒ 我已阅读并接受《[微博开发者协议](#)》

[申请SAE应用托管服务](#)

创建

- 完善信息应用信息，提交审核



## 注册应用程序的包名和签名

完善信息的时候，需要填写包名和签名，在[微博开放平台](#)上注册应用程序的包名和签名后才能正确授权

- 应用程序包名：

工程主modules目录下build.gradle中的applicationId的值，或者AndroidManifest.xml中package节点的数据。

- 应用程序签名：

该签名是通过官方提供的[签名工具](#)生成的MD5值。

### 签名工具使用方式：

首先要安装你已经签名的应用。然后再安装该工具APP，打开工具APP，输入应用的包名，点击生成按钮，既可以获取MD5签名，如下图所示。

# MD5签名生成器

包名

生成

MD5签名

复制

注：

1. 要签名的应用程序必须安装到该设备上才能生成对应的MD5签名。如果你在开发平台上只填写的是发布版本的签名MD5,请确保你平时debug版本的签名和发布版本签名一致，否则授权不能通过。
2. 包名和签名未注册，或者签名注册不正确，都会导致无法授权。

## 集成SDK

1. 在project根目录的build.gradle文件中添加依赖配置

```
allprojects {  
    repositories {  
        mavenCentral()  
        .....  
    }  
}
```

2. 在module的build.gradle文件中添加依赖和属性配置

```

android {
    .....
    defaultConfig {
        ndk {

            abiFilters 'armeabi' //, 'armeabi-v7a', 'arm64-v8a'

        }
    }
}

dependencies {
    compile 'com.sina.weibo.sdk:core:9.12.0:openDefaultRelease@aar'
}

```

## 参数配置

- 在AndroidManifest.xml中添加权限

```

<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />

```

- 混淆配置

```
-keep public class com.sina.weibo.sdk.**{*;}

```

## 初始化SDK

- 初始化代码如下

```

private static final String APP_KEY = "在微博开发平台为应用申请的App Key";
private static final String REDIRECT_URL = "在微博开放平台设置的授权回调页";
private static final String SCOPE = "在微博开放平台为应用申请的高级权限";

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    initSdk();
}

//init sdk

```

```
private void initSdk() {
    AuthInfo authInfo = new AuthInfo(this, APP_KY, REDIRECT_URL, SCOPE);
    mWBAPI = WBAPIFactory.createWBAPI(this);
    mWBAPI.registerApp(this, authInfo);
}
```

## 使用微博授权

### 1. 微博授权示例代码如下

```
/**
 * 授权操作
 */
private void startAuth() {
    //auth
    mWBAPI.authorize(new WbAuthListener() {
        @Override
        public void onComplete(Oauth2AccessToken token) {
            Toast.makeText(MainActivity.this, "微博授权成功",
Toast.LENGTH_SHORT).show();
        }

        @Override
        public void onError(UiError error) {
            Toast.makeText(MainActivity.this, "微博授权出错",
Toast.LENGTH_SHORT).show();
        }

        @Override
        public void onCancel() {
            Toast.makeText(MainActivity.this, "微博授权取消",
Toast.LENGTH_SHORT).show();
        }
    });
}

/**
 * 指定通过客户端授权操作
 */
private void startClientAuth() {
    mWBAPI.authorizeClient(new WbAuthListener() {
        @Override
        public void onComplete(Oauth2AccessToken token) {
            Toast.makeText(MainActivity.this, "微博授权成功",
Toast.LENGTH_SHORT).show();
        }
    });
}
```

```

        @Override
        public void onError(UiError error) {
            Toast.makeText(MainActivity.this, "微博授权出错",
Toast.LENGTH_SHORT).show();
        }

        @Override
        public void onCancel() {
            Toast.makeText(MainActivity.this, "微博授权取消",
Toast.LENGTH_SHORT).show();
        }
    });
}

/**
 * 指定通过网页（H5）授权操作
 */
private void startWebAuth() {
    mWBAPI.authorizeWeb(new WbAuthListener() {
        @Override
        public void onComplete(Oauth2AccessToken token) {
            Toast.makeText(MainActivity.this, "微博授权成功",
Toast.LENGTH_SHORT).show();
        }

        @Override
        public void onError(UiError error) {
            Toast.makeText(MainActivity.this, "微博授权出错:" +
error.errorDetail, Toast.LENGTH_SHORT).show();
        }

        @Override
        public void onCancel() {
            Toast.makeText(MainActivity.this, "微博授权取消",
Toast.LENGTH_SHORT).show();
        }
    });
}

```

2. 重写Activity的onActivityResult()方法，示例如下

```

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable
Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (mWBAPI != null) {
        mWBAPI.authorizeCallback(requestCode, resultCode, data);
    }
}

```

注：这里是设置授权的回调，如果不重写，将收不到授权结果。

## 分享内容到微博

1. 实现回调接口，代码示例如下

```

private static class ShareCallback implements WbShareCallback {
    @Override
    public void onComplete() {
        Toast.makeText(ShareActivity.this, "分享成功",
Toast.LENGTH_SHORT).show();
    }

    @Override
    public void onError(UiError error) {
        Toast.makeText(ShareActivity.this, "分享失败:" + error.errorMessage,
Toast.LENGTH_SHORT).show();
    }

    @Override
    public void onCancel() {
        Toast.makeText(ShareActivity.this, "分享取消",
Toast.LENGTH_SHORT).show();
    }
}

```

2. 重写Activity的onActivityResult()方法，示例如下



```

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable
Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (mWBAPI != null) {
        mWBAPI.doResultIntent(data, this);
    }
}

```

注：和授权登录同理，这里是设置分享的回调，如果不重写，将收不到分享结果。

### 3. 微博分享示例代码如下

```

private void doWeiboShare() {
    WeiboMultiMessage message = new WeiboMultiMessage();

    TextObject textObject = new TextObject();
    String text = "我正在使用微博客户端发博器分享文字。";

    // 分享文字
    if (mShareText.isChecked()) {
        text = "这里设置您要分享的内容! ";
        textObject.text = text;
        message.textObject = textObject;
    }

    // 分享图片
    if (mShareImage.isChecked()) {
        ImageObject imageObject = new ImageObject();
        Bitmap bitmap = BitmapFactory.decodeResource(getResources(),
R.drawable.share_image);
        imageObject.setImageData(bitmap);
        message.imageObject = imageObject;
    }

    // 分享网页
    if (mShareUrl.isChecked()) {
        WebpageObject webObject = new WebpageObject();
        webObject.identify = UUID.randomUUID().toString();
        webObject.title = "标题";
        webObject.description = "描述";
        Bitmap bitmap = BitmapFactory.decodeResource(getResources(),
R.drawable.ic_logo);
        ByteArrayOutputStream os = null;
        try {
            os = new ByteArrayOutputStream();
            bitmap.compress(Bitmap.CompressFormat.JPEG, 85, os);
            webObject.thumbData = os.toByteArray();
        } catch (Exception e) {

```

```

        e.printStackTrace();
    } finally {
        try {
            if (os != null) {
                os.close();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    webObject.actionUrl = "https://weibo.com";
    webObject.defaultText = "分享网页";
    message.mediaObject = webObject;
}

if (mShareMultiImage.isChecked()) {
    // 分享多图
    MultiImageObject multiImageObject = new MultiImageObject();
    ArrayList<Uri> list = new ArrayList<>();
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.N) {
        String authority = this.getPackageName() + ".fileprovider";
        list.add(FileProvider.getUriForFile(this, authority, new
File(getExternalFilesDir(null) + "/aaa.png")));
        list.add(FileProvider.getUriForFile(this, authority, new
File(getExternalFilesDir(null) + "/ccc.JPG")));
        list.add(FileProvider.getUriForFile(this, authority, new
File(getExternalFilesDir(null) + "/ddd.jpg")));
        list.add(FileProvider.getUriForFile(this, authority, new
File(getExternalFilesDir(null) + "/fff.jpg")));
        list.add(FileProvider.getUriForFile(this, authority, new
File(getExternalFilesDir(null) + "/ggg.JPG")));
        list.add(FileProvider.getUriForFile(this, authority, new
File(getExternalFilesDir(null) + "/eee.jpg")));
        list.add(FileProvider.getUriForFile(this, authority, new
File(getExternalFilesDir(null) + "/hhhh.jpg")));
        list.add(FileProvider.getUriForFile(this, authority, new
File(getExternalFilesDir(null) + "/kkk.JPG")));
    } else {
        list.add(Uri.fromFile(new File(getExternalFilesDir(null) +
"/aaa.png")));
        list.add(Uri.fromFile(new File(getExternalFilesDir(null) +
"/ccc.JPG")));
        list.add(Uri.fromFile(new File(getExternalFilesDir(null) +
"/ddd.jpg")));
        list.add(Uri.fromFile(new File(getExternalFilesDir(null) +
"/fff.jpg")));
        list.add(Uri.fromFile(new File(getExternalFilesDir(null) +
"/ggg.JPG")));
    }
}

```

```

        list.add(Uri.fromFile(new File(getExternalFilesDir(null) +
"/eee.jpg")));
        list.add(Uri.fromFile(new File(getExternalFilesDir(null) +
"/hhhh.jpg")));
        list.add(Uri.fromFile(new File(getExternalFilesDir(null) +
"/kkk.JPG")));
    }
    multiImageObject.imageList = list;
    message.multiImageObject = multiImageObject;
}

if (mShareVideo.isChecked()) {
    // 分享视频
    VideoSourceObject videoObject = new VideoSourceObject();
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.N) {
        String filePath = getExternalFilesDir(null) + "/eeee.mp4";
        File videoFile = new File(filePath);
        if (!videoFile.getParentFile().exists()) {
            videoFile.getParentFile().mkdir();
        }
        videoObject.videoPath = FileProvider.getUriForFile(this,
this.getPackageName() + ".fileprovider", videoFile);
    } else {
        videoObject.videoPath = Uri.fromFile(new
File(getExternalFilesDir(null) + "/eeee.mp4"));
    }

    message.videoSourceObject = videoObject;
}

boolean isClientOnly = mShareClientOnly.isChecked(); // 是否指定用客户端分享。
mWBAPI.sendMessage(message, isClientOnly);
}

```

## 注：

1. 当前SDK已经适配Android 11 ,支持通过FileProvider分享。

## 2. 多图分享

- 设置的是本地图片文件的路径，并且是当前应用可以访问的路径，目前不支持网络路径。
- 接入程序必须有文件读写权限，否则会造成失败。
- 使用前通过WbSdk.isSupportMultipleImage(context)判断当前微博版本是否支持多图分享

### 3. 视频分享

- 设置的是本地视频文件的路径，并且是当前应用可以访问的路径，目前不支持网络路径。
- 目前支持视频格式为：mp4。

### 4. H5分享

- H5分享只支持分享文字和单图。
- 图片要求是合规图片。

5. 当前版本已支持分享超话内容，若有需要，请找微博商务洽谈，取得白名单权限后再进行该功能开发。