微博SDK-Android接入文档

SDK版本:13.10.5-ciopt-4

<u>更新日志</u>

工程配置

使用微博授权

分享内容到微博

接入须知

工程要求

属性	限制内容
targetSdkVersion	<= 34
minSdk	>= 22(低于22时,调用API前检 查,Build.VERSION.SDK_INT>=22)
ABI	无so依赖
FileProvider	默认authority:"\${你的包名}.wbsdk.fileprovider";同时支持开发 者自定义FileProvider及authority

业务限制

属性	分享对象	限制内容
Web 分享	WebpageObject	Web分享只支持分享文字
单图 分享	ImageObject	目前支持图片格式:jpg、png、webp 图片大小限制:图片转码后需小于200k,可通过 ImageObject.checkArgs检查 说明:单图分享存在转码操作,有单图分享需求建议同 样使用多图分享API
多图 分享	MultiImageObject	目前支持图片格式:jpg、png、gif 图片大小限制:多图分享时,每张图片建议小于10MB 图片个数限制:最多18张
视频 分享	VideoSourceObject	目前支持视频格式:mp4
超话 分享	SuperGroupObject	非完全开放类型,请联系微博商务洽谈,取得白名单权限后再进行该功能开发; <u>超话分享开发文档</u>

工程配置

集成SDK(请选择其中一种方式集成)

方式一(使用远程依赖)

• 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文件中添加远程依赖

```
dependencies {
  implementation 'io.github.sinaweibosdk:core:13.10.5-ciopt-4@aar'
}
```

方式二(使用aar本地依赖)

- 1. <u>打开最新文档</u>下载需要依赖的SDK版本的aar, <u>将AAR文件作为工程依赖</u>
- 2. 在module的build.gradle文件中添加本地依赖

```
dependencies {
  implementation(name:'core-13.10.5-ciopt-4',ext:'aar')
}
```

Proguard (混淆配置)

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

接入代码

初始化SDK

```
public class DemoApplication extends Application {
   private static final String APP_KY = "在微博开发平台为应用申请的App I
   private static final String REDIRECT_URL = "在微博开放平台设置的授权
   private static final String SCOPE = "在微博开放平台为应用申请的高级权
   protected void onCreate() {
       super.onCreate();
       initSdk();
    * 初始化sdk。
    * 理论上使用前只要初始化一次即可,具体分享及授权登录时将不需要再次初始化。
   private void initSdk() {
       AuthInfo authInfo = new AuthInfo(this, APP_KY, REDIRECT_URL,
       mWBAPI = WBAPIFactory.createWBAPI(this); // 传Context即可, 不真
       {\tt mWBAPI.registerApp}\,({\tt this},\ {\tt authInfo},\ {\tt new}\ {\tt SdkListener}\,()\ \{
           public void onInitSuccess() {
              // SDK初始化成功回调,成功一次后再次初始化将不再有任何回调
           @Override
          public void onInitFailure(Exception e) {
               // SDK初始化失败回调
       });
                                                              F
```

使用微博授权

1. 在Activity中调用授权API(请选择其中一种方式)

```
//方式一,优先客户端授权,未安装时使用Web授权
    private void startAuth() {
       mWBAPI.authorize(this, new WbAuthListener() {
           @Override
           public void onComplete(Oauth2AccessToken token) {
               Toast.makeText(MainActivity.this, "微博授权成功", Toast.LE
           @Override
           public void onError(UiError error) {
               Toast.makeText(MainActivity.this, "微博授权出错", Toast.LE
          @Override
           public void onCancel() {
               Toast.makeText(MainActivity.this, "微博授权取消", Toast.LE
       });
        //方式二,指定通过客户端授权操作
   private void startClientAuth() {
       mWBAPI.authorizeClient(this, new WbAuthListener() {
           @Override
           public void onComplete(Oauth2AccessToken token) {
               Toast.makeText(MainActivity.this, "微博授权成功", Toast.LE
           @Override
           public void onError(UiError error) {
               Toast.makeText(MainActivity.this, "微博授权出错", Toast.LE
           @Override
           public void onCancel() {
               Toast.makeText(MainActivity.this, "微博授权取消", Toast.LE
       });
       //方式三,指定通过网页 (Web) 授权操作
   private void startWebAuth() {
       mWBAPI.authorizeWeb(this, new WbAuthListener() {
           @Override
           public void onComplete(Oauth2AccessToken token) {
               Toast.makeText(MainActivity.this, "微博授权成功", Toast.LE
           @Override
           public void onError(UiError error) {
               Toast.makeText(MainActivity.this, "微博授权出错:" + error.
           @Override
           public void onCancel() {
               Toast.makeText(MainActivity.this, "微博授权取消", Toast.LE
       });
4
                                                                   F
2. 重写Activity的onActivityResult()方法,示例如下
   @Override
   protected void onActivityResult(int requestCode, int resultCode, @Nu
       super.onActivityResult(requestCode, resultCode, data);
       if (mWBAPI != null) {
           mWBAPI.authorizeCallback(requestCode, resultCode, data);
4
```

注:这里是设置授权的回调,如果不重写,将收不到授权结果。

3. 错误码定位: OAuth 2.0 错误码

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_SF
    }

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

    @Override
    public void onCancel() {
        Toast.makeText(ShareActivity.this, "分享取消", Toast.LENGTH_SF
    }
}
```

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

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() {
  //相册-照片
  if (mShareMediaImg.isChecked()) {
   openGallery(REQ_PICK_IMAGE_COMPOSER, false);
   return:
  //相册-视频
 if (mShareMediaVideo.isChecked()) {
   openGallery(REQ_PICK_VIDEO_COMPOSER, true);
   return:
 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.d
   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.d
   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<>();
       File externalFilesDir = getExternalFilesDir(null);
       File externalCacheDir = getExternalCacheDir();
       if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.N) {
              String authority = this.getPackageName() + ".wbsdk.fileprovide
              {\tt list.add}\,({\tt FileProvider.getUriForFile}\,({\tt this},\ {\tt authority},\ {\tt new}\ {\tt File}\,({\tt outhority},\ {\tt new}\ {\tt File}\,({\tt outhority},\ {\tt new}\ {\tt outhority},\ {\tt o
              list.add(FileProvider.getUriForFile(this, authority, new File(
              list.add(FileProvider.getUriForFile(this, authority, new File(
              {\tt list.add}\,({\tt FileProvider.getUriForFile}\,({\tt this},\ {\tt authority},\ {\tt new}\ {\tt File}\,({\tt outhority},\ {\tt new}\ {\tt File}\,({\tt outhority},\ {\tt new}\ {\tt outhority},\ {\tt o
              list.add(FileProvider.getUriForFile(this, authority, new File(
              {\tt list.add}\,({\tt FileProvider.getUriForFile}\,({\tt this},\ {\tt authority},\ {\tt new}\ {\tt File}\,({\tt outhority},\ {\tt new}\ {\tt File}\,({\tt outhority},\ {\tt new}\ {\tt outhority},\ {\tt o
              list.add(FileProvider.getUriForFile(this, authority, new File(
              list.add(FileProvider.getUriForFile(this, authority, new File(
       } else
              list.add(Uri.fromFile(new File(externalFilesDir, "aaa.png")));
              list.add(Uri.fromFile(new File(externalCacheDir, "bbb.jpg")));
              list.add(Uri.fromFile(new File(externalFilesDir + "/file_image
              list.add(Uri.fromFile(new File(externalFilesDir + "/cache_imag
              list.add(Uri.fromFile(new File(externalCacheDir + "/file_image
              list.add(Uri.fromFile(new File(externalFilesDir + "/file_image
              list.add(Uri.fromFile(new File(externalFilesDir+"/file_images/
              list.add(Uri.fromFile(new File(externalFilesDir + "/file_image
      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) + "/file_video/eee
              File videoFile = new File(filePath);
              if (!videoFile.getParentFile().exists()) {
                    videoFile.getParentFile().mkdir();
              videoObject.videoPath = FileProvider.getUriForFile(this, this.
       else
             videoObject.videoPath = Uri.fromFile(new File(getExternalFiles
      message.videoSourceObject = videoObject;
if (mShareSuperGroup.isChecked()) {
         // 如果想要拉起超话类型发布器需要传superGroupobject对象
       SuperGroupObject superGroupObject = new SuperGroupObject();
         * 超话名。可以传可以不传
                   1、如果不传该参数则拉起选择超话弹层
                   2、如果传该参数,不出选择超话弹层,直接选中超话
       String sgName = ((EditText) findViewById(R.id.edit_sg_name)).get'
       if (TextUtils.isEmpty(sgName)) {
             sgName = this.getString(R.string.demo_sg_name);
       superGroupObject.sqName = sqName;
        // 超话板块,可以传可以不传
       String sgSection = ((EditText)findViewById(R.id.edit_sg_section)
       superGroupObject.secName = sgSection;
        // 扩展参数,需要透传的信息加在这个参数里
       String sgExt = ((EditText) findViewById(R.id.edit_sg_ext)).getTe:
       superGroupObject.sgExtParam = sgExt;
      message.superGroupObject = superGroupObject;
mWBAPI.shareMessage(this, message, mShareClientOnly.isChecked());
```

4