Android开放平台SDK使用说明（外部app）

一、设计目标

开放平台sdk的设计目标是：第三方应用使用该sdk能够方便、快速的调起快手主app的功能页面。如个人主页、私信页、视频编辑页等，从而将第三方内容发布到快手平台上或者使用快手主app提供的开放功能。

二、项目描述

项目demo的git库：https://github.com/KwaiSocial/KwaiSDK-Demo-Android

三、第三方接入说明

1、接入aar

版本要求 minsdkversion:19

引用aar方式：

```
dependencies {
   implementation "com.github.kwaisocial:kwai-opensdk-withauth:2.1.0"  // 版本号请设置最新的版本
}
```

混淆配置
```
-keep class com.kwai.opensdk.sdk.** {*;}

-keep class com.kwai.opensdk.auth.** {*;}
```

2、api使用说明

(1) 登录认证并获取openId

第三方app需要通过渠道获取分配给第三应用的appId，当使用需要认证的接口时，请先获取必须的参数openId

```
KwaiOpenSdkAuth.init(this); // Application

private IKwaiOpenSdkAuth mKwaiOpenSdkAuth; //
mKwaiOpenSdkAuth = new KwaiOpenSdkAuth(); //

// openIdopenid
```

```
mKwaiOpenSdkAuth.addKwaiAuthListener(new IKwaiAuthListener() {
  @Override
  public void onSuccess(final String code) {
    new Thread(new Runnable() {
      public void run() {
        String result = null;
        int retry = 0;
        while (null == result && retry < NETWORK_MAX_RETRY_TIMES) {
          result = getOpenIdByNetwork(code);
          retry++;
          LogUtil.i(TAG, "retry=" + retry);
        }
        final String openId = result;
        Handler mainHandler = new Handler(Looper.getMainLooper());
        mainHandler.post(new Runnable() {
          @Override
          public void run() {
            mOpenId = openId;
            if (TextUtils.isEmpty(mOpenId)) {
              mOpenIdTv.setText("openId:" + "get openId error");
            } else {
              mOpenIdTv.setText("openId:" + mOpenId);
            }
          }
        });
      }
    }).start();
  }

  @Override
  public void onFailed(String errMsg) {
    mOpenIdTv.setText("get code error and msg is " + errMsg);
  }

  @Override
  public void onCancel() {
    mOpenIdTv.setText("get code is canceled");
  }
});


private String getOpenIdByNetwork(final String code) {
  String url =
      getRequestOpenIdUrl("code", SOCIAL_SHARE_FT.getAppId(), SOCIAL_SHARE_FT.getAppKey(), code);

  String result = NetworkUtil.get(url, null, null);
  String openId = null;
  try {
    LogUtil.i(TAG, "result=" + result);
    JSONObject obj = new JSONObject(result);
    openId = obj.getString("open_id");
    LogUtil.i(TAG, "openId=" + openId);
  } catch (Throwable t) {
    LogUtil.e(TAG, "getOpenId exception");
  }
  return openId;
}

private String getRequestOpenIdUrl(String grantType, String appId, String appKey, String code) {
  StringBuilder builder = new StringBuilder();
  builder.append(URL_HOST);
  builder.append("/oauth2/access_token?");
  builder.append("grant_type=" + grantType);
  builder.append("&app_id=" + appId);
  builder.append("&app_secret=" + appKey);
  builder.append("&code=" + code);
  return builder.toString();
}

// APP

mKwaiOpenSdkAuth.sendAuthReqToKwai(this, SOCIAL_SHARE_FT.getAppId(), Config.SCOPE);


//

mKwaiOpenSdkAuth.removeKwaiAuthListener();
```

（2）开放平台api使用（当第三方应用已经获取了openId时，使用kwai-opensdk依赖）

```
private IKwaiOpenAPI mKwaiOpenAPI; //
```

```
mKwaiOpenAPI = KwaiOpenAPIFactory.createKwaiOpenAPI(this, SOCIAL_SHARE_FT.getAppId()); //


// sdkloadingfalseloading

mKwaiOpenAPI.setShowDefaultLoading(true);


//

mKwaiOpenAPI.addKwaiAPIEventListerer(new IKwaiAPIEventListener() {

  @Override
  public void onRespResult(@NonNull BaseResp resp) {
    Log.i(TAG, "resp=" + resp);
    if (resp != null) {
      Log.i(TAG, "errorCode=" + resp.errorCode + ", errorMsg="
          + resp.errorMsg + ", cmd=" + resp.getCommand()
          + ", transaction=" + resp.transaction);
      mCallbackTv.setText("CallBackResult: errorCode=" + resp.errorCode + ", errorMsg="
          + resp.errorMsg + ", cmd=" + resp.getCommand()
          + ", transaction=" + resp.transaction);
    } else {
      mCallbackTv.setText("CallBackResult: resp is null");
    }
  }
});

//

mKwaiOpenAPI.removeKwaiAPIEventListerer();
```

(3) 社交方向相关的业务请求示例代码：

```
//
public void shareMessage(View view) {
  // base params
  ShareMessageToKwai.Req req = new ShareMessageToKwai.Req();
  req.openId = mOpenId;
  req.sessionId = mKwaiOpenAPI.getOpenAPISessionId();
  req.transaction = "shareMessage";

  // business params
  req.message = new KwaiMediaMessage();
  req.message.mediaObject = new KwaiWebpageObject();
  ((KwaiWebpageObject) req.message.mediaObject).webpageUrl =
      "https://blog.csdn.net/a249900679/article/details/51386660";
  req.message.title = "test";
  req.message.description = "webpage test share, hahahah";
  Bitmap b = BitmapFactory.decodeResource(getResources(), R.mipmap.ic_launcher);
  ByteArrayOutputStream baos = new ByteArrayOutputStream();
  b.compress(Bitmap.CompressFormat.PNG, 100, baos);
  req.message.thumbData = baos.toByteArray();

  // send request
  mKwaiOpenAPI.sendReq(req, this);
}
```

```
// TargetOpenId
public void shareMessageToBuddy(View view) {
  if (TextUtils.isEmpty(HistoryOpenIdActivity.sTargetOpenId)) {
    Toast.makeText(this, "sTargetOpenId is null, ", Toast.LENGTH_SHORT).show();
    return;
  }

  ShareMessageToKwai.Req req = new ShareMessageToKwai.Req();
  req.openId = mOpenId;
  req.sessionId = mKwaiOpenAPI.getOpenAPISessionId();
  req.transaction = "sharemessageToBuddy";

  req.targetOpenId = HistoryOpenIdActivity.sTargetOpenId;
  req.message = new KwaiMediaMessage();
  req.message.mediaObject = new KwaiWebpageObject();
  ((KwaiWebpageObject) req.message.mediaObject).webpageUrl =
      "https://blog.csdn.net/a249900679/article/details/51386660";
  req.message.title = "test";
  req.message.description = "webpage test share, hahahah";
  Bitmap b = BitmapFactory.decodeResource(getResources(), R.mipmap.ic_launcher);
  ByteArrayOutputStream baos = new ByteArrayOutputStream();
  b.compress(Bitmap.CompressFormat.PNG, 100, baos);
  req.message.thumbData = baos.toByteArray();
  mKwaiOpenAPI.sendReq(req, this);
}


// 打开TargetOpenId指向的个人主页
public void showProfile(View view) {
  if (TextUtils.isEmpty(HistoryOpenIdActivity.sTargetOpenId)) {
  Toast.makeText(this, "sTargetOpenId is null, 请先设置", Toast.LENGTH_SHORT).show();
  return;
  }

 ShowProfile.Req req = new ShowProfile.Req();
 req.openId = mOpenId;
 req.sessionId = mKwaiOpenAPI.getOpenAPISessionId();
 req.transaction = "showProfile";
 req.targetOpenId = HistoryOpenIdActivity.sTargetOpenId;
 mKwaiOpenAPI.sendReq(req, this);
}
```

(4) 生产方向相关的业务请求示例代码：

```
//
public void publishPicture(File file) {
  // base params
  SinglePicturePublish.Req req = new SinglePicturePublish.Req();
  req.sessionId = mKwaiOpenAPI.getOpenAPISessionId();
  req.transaction = "singlePicturePublish_0";

  // business params
  req.mediaInfo = new PostShareMediaInfo();
  ArrayList<String> imageFile = new ArrayList<>();
  imageFile.add(file.getAbsolutePath());
  req.mediaInfo.mMultiMediaAssets = imageFile;
  if (!TextUtils.isEmpty(mTagList.getText().toString())) {
    req.mediaInfo.mTag = mTagList.getText().toString();
  }
  req.mediaInfo.mDisableFallback = mDisableFallBack.isChecked();
  if (!TextUtils.isEmpty(mExtraEdit.getText().toString())) {
    req.mediaInfo.mExtraInfo = mExtraEdit.getText().toString();
  }

  // send request
  mKwaiOpenAPI.sendReq(req, getActivity());
}
```

```
//
public void editPicture(File file) {
  SinglePictureEdit.Req req = new SinglePictureEdit.Req();
  req.sessionId = mKwaiOpenAPI.getOpenAPISessionId();
  req.transaction = "singlePictureEdit_0";

  req.mediaInfo = new PostShareMediaInfo();
  ArrayList<String> imageFile = new ArrayList<>();
  imageFile.add(file.getAbsolutePath());
  req.mediaInfo.mMultiMediaAssets = imageFile;
  if (!TextUtils.isEmpty(mTagList.getText().toString())) {
    req.mediaInfo.mTag = mTagList.getText().toString();
  }
  req.mediaInfo.mDisableFallback = mDisableFallBack.isChecked();
  if (!TextUtils.isEmpty(mExtraEdit.getText().toString())) {
    req.mediaInfo.mExtraInfo = mExtraEdit.getText().toString();
  }
  mKwaiOpenAPI.sendReq(req, getActivity());
}


//
public void publishSingleVideo(File file) {
  SingleVideoPublish.Req req = new SingleVideoPublish.Req();
  req.sessionId = mKwaiOpenAPI.getOpenAPISessionId();
  req.transaction = "singleVideoPublish_0";

  req.mediaInfo = new PostShareMediaInfo();
  ArrayList<String> imageFile = new ArrayList<>();
  imageFile.add(file.getAbsolutePath());
  req.mediaInfo.mMultiMediaAssets = imageFile;
  if (!TextUtils.isEmpty(mSingleVideoCoverPath.getText())) {
    req.mCover = mSingleVideoCoverPath.getText().toString();
  }
  if (!TextUtils.isEmpty(mTagList.getText().toString())) {
    req.mediaInfo.mTag = mTagList.getText().toString();
  }
  req.mediaInfo.mDisableFallback = mDisableFallBack.isChecked();
  if (!TextUtils.isEmpty(mExtraEdit.getText().toString())) {
    req.mediaInfo.mExtraInfo = mExtraEdit.getText().toString();
  }
  mKwaiOpenAPI.sendReq(req, getActivity());
}



//
public void editSingleVideo(File file) {
  SingleVideoEdit.Req req = new SingleVideoEdit.Req();
  req.sessionId = mKwaiOpenAPI.getOpenAPISessionId();
  req.transaction = "singleVideoEdit_0";

  req.mediaInfo = new PostShareMediaInfo();
  ArrayList<String> imageFile = new ArrayList<>();
  imageFile.add(file.getAbsolutePath());
  req.mediaInfo.mMultiMediaAssets = imageFile;

  if (!TextUtils.isEmpty(mTagList.getText().toString())) {
    req.mediaInfo.mTag = mTagList.getText().toString();
  }
  req.mediaInfo.mDisableFallback = mDisableFallBack.isChecked();
  if (!TextUtils.isEmpty(mExtraEdit.getText().toString())) {
    req.mediaInfo.mExtraInfo = mExtraEdit.getText().toString();
  }
  mKwaiOpenAPI.sendReq(req, getActivity());
}
```

```
//
public void clipSingleVideo(File file) {
  SingleVideoClip.Req req = new SingleVideoClip.Req();
  req.sessionId = mKwaiOpenAPI.getOpenAPISessionId();
  req.transaction = "singleVideoClip_0";

  req.mediaInfo = new PostShareMediaInfo();
  ArrayList<String> imageFile = new ArrayList<>();
  imageFile.add(file.getAbsolutePath());
  req.mediaInfo.mMultiMediaAssets = imageFile;

  if (!TextUtils.isEmpty(mTagList.getText().toString())) {
    req.mediaInfo.mTag = mTagList.getText().toString();
  }
  req.mediaInfo.mDisableFallback = mDisableFallBack.isChecked();
  if (!TextUtils.isEmpty(mExtraEdit.getText().toString())) {
    req.mediaInfo.mExtraInfo = mExtraEdit.getText().toString();
  }
  mKwaiOpenAPI.sendReq(req, getActivity());
}


//MultiMedia
public void clipMultiMedia(ArrayList<String> multiMedia) {
  MultiMediaClip.Req req = new MultiMediaClip.Req();
  req.sessionId = mKwaiOpenAPI.getOpenAPISessionId();
  req.transaction = "multiMediaClip_0";

  req.mediaInfo = new PostShareMediaInfo();
  req.mediaInfo.mMultiMediaAssets = multiMedia;

  if (!TextUtils.isEmpty(mTagList.getText().toString())) {
    req.mediaInfo.mTag = mTagList.getText().toString();
  }
  req.mediaInfo.mDisableFallback = mDisableFallBack.isChecked();
  if (!TextUtils.isEmpty(mExtraEdit.getText().toString())) {
    req.mediaInfo.mExtraInfo = mExtraEdit.getText().toString();
  }
  mKwaiOpenAPI.sendReq(req, getActivity());
}
```

3、错误码定义

```
public interface KwaiOpenSdkErrorCode {

  //ios
  int ERR_KWAI_APP_NOT_LOGIN = -1011;
  int ERR_INVALID_PARAMETERS = -1010;
  int ERR_KWAI_APP_UNSUPPORT = -1006;
  int ERR_NO_KWAI_APP = -1005;
  int ERR_CANCEL = -1;
  int ERR_OK = 1;
  // add new
  int ERR_NO_AUTH_OPENID = -1012;
  int ERR_FALL_BACK_REJECT = -1013;
  int ERR_TRRNAGER_MODE = -1014;
  int ERR_NETWORK = -1015;
  // post share
  int ERR_NO_PERMISSON = -1016;
  int ERR_COMPRESS_PICTURE = -1017;
```

```
//server api errorcode
 int ERR_UNAUTHORIZED_CMD = 20088; //cmd
 int ERR_UNACCESSIBLE_USERID = 20089; //openIduserId
 int ERR_INVALID_CMD = 20090; //cmd
 int ERR_INCONSISTEN_OPENID_LOGIN_USERID = 20091; //userId  openIduserId
 int ERR_TARGET_NOT_BUDDY = 20092; //targetOpenId
 int ERR_INVALID_TARGET_OPEN_ID = 20094; // targetOpenId

 /**
 * 100200100, //
 * 100200101, // client app  developer
 * 100200102, // token
 * 100200103, // responseType
 * 100200104, // grantType
 * 100200105, // code
 * 100200106, // scope
 * 100200107, // openid
 * 100200108, // access_token  refresh_token
 * 100200109, //  app
 * 100200110, //
 * 100200111, //
 */
 int ERR_SERVER_CHECK_INVALID_PARAMETER = 100200100; //SERVER
}
```