# qq第三方支持文档

author:leobert.lan

## 接入准备：

申请开发者账号，创建应用，获取id和key……

下载sdk

## 导入sdk：

此处精华，qq的文档时看不到的

eclipse：

* sdk包拷贝到libs中（adt>17）
* 配置lib的path，基本没有adt小于17的了

studio：

常规导入

AndroidManifest文件：

* 各种权限
* 必须添加以下activity，注意黄色内容

|  |
| --- |
| **<activity**  **android:name="com.tencent.tauth.AuthActivity"**  **android:launchMode="singleTask"**  **android:noHistory="true" >**  **<intent-filter>**  **<action android:name="android.intent.action.VIEW" />**  **<category android:name="android.intent.category.DEFAULT" />**  **<category android:name="android.intent.category.BROWSABLE" />**  **<!-- tencent+appid -->**  **<data android:scheme="tencent1105135204" />**  **</intent-filter>**  **</activity>**  **<activity**  **android:name="com.tencent.connect.common.AssistActivity"**  **android:configChanges="orientation|keyboardHidden"**  **android:screenOrientation="behind"**  **android:theme="@android:style/Theme.Translucent.NoTitleBar" />** |

## 实例化Tencent

在程序入口或者application中完成 this is the best practice.

|  |
| --- |
| **public static Tencent mTencent;**  **if (mTencent == null) {**  **mTencent = Tencent.createInstance("1105135204", this); //你的id**  **}** |

然后，尽情的使用吧

## qq授权登录

### 直接给一坨代码：

|  |
| --- |
| public class QQOAuth {  private static final String TAG = "QQOAuth";  /\*\*  \* @Title: login  \* @Description: 登录 最好写一个 IUiListener的实现类，因为使用场景可能有几个  \* @author: leobert.lan  \* @param activity  \* @param callback  \*/  public static void login(final Activity activity, final IUiListener callback) {  // if (!mTencent.isSessionValid())  UMengActivity.mTencent.logout(activity);  UMengActivity.mTencent.login(activity, "all", callback);  // getQAuthInfo(activity, callback);  }  public static void login(final android.support.v4.app.Fragment fragment,  final IUiListener callback) {  UMengActivity.mTencent.login(fragment, "all", callback);  Log.d("SDKQQAgentPref",  "FirstLaunch\_SDK:" + SystemClock.elapsedRealtime());  }  public static void logout(final Context ctx) {  UMengActivity.mTencent.logout(ctx);  }  /\*\*  \* @Title: getQAuthInfo  \* @Description: 用于已经登录情况下再次获取缓存的信息  \* @author: leobert.lan  \*/  public static void getQAuthInfo(final Activity activity,  IUiListener callback) {  if (UMengActivity.mTencent != null && UMengActivity.mTencent.isSessionValid()) {  UserInfo info = new UserInfo(activity, UMengActivity.mTencent.getQQToken());  info.getUserInfo(callback);  } else {  Log.wtf(TAG, "mTencent == null 或者 验证非法，好好检验");  }  }  } |

### 调用的activity或者fragment必须实现onActivityResult

|  |
| --- |
| protected void onActivityResult(int requestCode, int resultCode, Intent data) {  if (requestCode == Constants.REQUEST\_LOGIN  || requestCode == Constants.REQUEST\_APPBAR) {  Tencent.onActivityResultData(requestCode, resultCode, data,  logincallback);  }  super.onActivityResult(requestCode, resultCode, data);  }; |

### IUiListener的实现

建议写接口实现类，业务化

### 成功返回数据：都是坑啊！

|  |
| --- |
| {  "ret": 0,  "openid": "0FB36DC4FA8FA7F31EC9943F5F59C526",  "access\_token": "CA512D2AB64C5954568EDD9D8EDBB941",  "pay\_token": "65C5B1D9DB56F4595DABB4E4EF9BAE76",  "expires\_in": 7776000,  "pf": "desktop\_m\_qq-10000144-android-2002-",  "pfkey": "48cc958597014956a316a2b0e7750a1f",  "msg": "",  "login\_cost": 282,  "query\_authority\_cost": 181,  "authority\_cost": -92466938  } |

其中openid是qq用户的唯一标识

另外如果需要获取用户信息，本次暂不处理

UserInfo info = new UserInfo(this, MainActivity.mQQAuth.getQQToken());

info.getUserInfo(new BaseUIListener(this,"get\_simple\_userinfo"));

## qq授权登出

### 就这样：

public static void logout(final Context ctx) {

UMengActivity.mTencent.logout(ctx);

}

## qq分享

### 分享概述：

* qq分享不需要登录授权
* 本篇包含qq分享和空间分享
* 分享类型，4种
  + 图文消息分享
  + 纯图片分享
  + 音乐分享
  + 应用分享

可能音乐分享使用可能不大，暂不整理，应用分享也暂不整理

### 图文消息分享

参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **是否必传** | **类型** | **参数说明** |
| QQShare.SHARE\_TO\_QQ\_KEY\_TYPE | 必填 | Int | 分享的类型。图文分享(普通分享)填*Tencent.SHARE\_TO\_QQ\_TYPE\_DEFAULT* |
| QQShare.PARAM\_TARGET\_URL | 必填 | String | 这条分享消息被好友点击后的跳转URL。 |
| QQShare.PARAM\_TITLE | 必填 | String | 分享的标题, 最长30个字符。 |
| QQShare.PARAM\_SUMMARY | 可选 | String | 分享的消息摘要，最长40个字。 |
| QQShare.SHARE\_TO\_QQ\_IMAGE\_URL | 可选 | String | 分享图片的URL或者本地路径 |
| QQShare.SHARE\_TO\_QQ\_APP\_NAME | 可选 | String | 手Q客户端顶部，替换“返回”按钮文字，如果为空，用返回代替 |
| QQShare.SHARE\_TO\_QQ\_EXT\_INT | 可选 | int | 分享额外选项，两种类型可选（默认是不隐藏分享到QZone按钮且不自动打开分享到QZone的对话框）：  Tencent.SHARE\_TO\_QQ\_FLAG\_QZONE\_AUTO\_OPEN，分享时自动打开分享到QZone的对话框。  Tencent.SHARE\_TO\_QQ\_FLAG\_QZONE\_ITEM\_HIDE，分享时隐藏分享到QZone按钮 |

参数整理temp

{

“type”:,

“targetUrl”:目标url,

“title”:30字符内,

“image”:建议，使用云盘图标、或者图片本身，但是除了发起者，谁都看不到

“summary”:概要，建议，40字以内

“from”:建议，appname

“to”:区分qq还是空间

}

调用分享接口的示例代码如下：

|  |
| --- |
| private void qqShare() {  Log.d(tag, "call share");  final Bundle params = new Bundle();  int mExtarFlag = 0x00;  params.putInt(QQShare.SHARE\_TO\_QQ\_KEY\_TYPE,  QQShare.SHARE\_TO\_QQ\_TYPE\_DEFAULT);  params.putString(QQShare.SHARE\_TO\_QQ\_TITLE, "test qqShare");  params.putString(QQShare.SHARE\_TO\_QQ\_SUMMARY,  "this is a test in vso\_pan,url is baidu,");  params.putString(QQShare.SHARE\_TO\_QQ\_TARGET\_URL, "http://www.baidu.com");  params.putString(QQShare.SHARE\_TO\_QQ\_IMAGE\_URL,  "/storage/emulated/0/icon.png");  params.putString(QQShare.SHARE\_TO\_QQ\_APP\_NAME, "创意云盘");  params.putInt(QQShare.SHARE\_TO\_QQ\_EXT\_INT, mExtarFlag);  doShareToQQ(params);  }  private void doShareToQQ(final Bundle params) {  // QQ分享要在主线程做  ThreadManager.getMainHandler().post(new Runnable() {  @Override  public void run() {  if (null != UMengActivity.mTencent) {  UMengActivity.mTencent.shareToQQ(SettingActivity.this,  params, qqShareListener);  }  }  });  }  IUiListener qqShareListener = new IUiListener() {  @Override  public void onCancel() {  Log.d(tag, "onCancel: ");  }  @Override  public void onComplete(Object response) {  // TODO Auto-generated method stub  Log.d(tag, "onComplete: " + response.toString());  }  @Override  public void onError(UiError e) {  // TODO Auto-generated method stub  Log.d(tag, "onError: " + e.errorMessage);  }  }; |

### 分享纯图片

参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **是否必传** | **类型** | **参数说明** |
| QQShare.SHARE\_TO\_QQ\_KEY\_TYPE | 必选 | int | 分享类型，分享纯图片时填写*Tencent.*  *SHARE\_TO\_QQ\_TYPE\_IMAGE* |
| QQShare.SHARE\_TO\_QQ\_IMAGE\_LOCAL\_URL | 必选 | String | 需要分享的本地图片路径 |
| QQShare.SHARE\_TO\_QQ\_APP\_NAME | 可选 | String | 手Q客户端顶部，替换“返回”按钮文字，如果为空，用返回代替 |
| QQShare.SHARE\_TO\_QQ\_EXT\_INT | 可选 | int | 分享额外选项，两种类型可选（默认是不隐藏分享到QZone按钮且不自动打开分享到QZone的对话框）：  Tencent.SHARE\_TO\_QQ\_FLAG\_QZONE\_AUTO\_OPEN，分享时自动打开分享到QZone的对话框。  Tencent.SHARE\_TO\_QQ\_FLAG\_QZONE\_ITEM\_HIDE，分享时隐藏分享到QZone按钮  分享到qq就是1  分享到空间就是2 |

参数整理 temp

{

“type”:类型

“image”:

“from”:

“to”:

}

调用分享接口的示例代码如下：

|  |
| --- |
| params.putString(QQShare.SHARE\_TO\_QQ\_IMAGE\_LOCAL\_URL,  "/storage/emulated/0/icon.png");  params.putString(QQShare.SHARE\_TO\_QQ\_APP\_NAME, "创意云盘");  params.putInt(QQShare.SHARE\_TO\_QQ\_KEY\_TYPE,  QQShare.SHARE\_TO\_QQ\_TYPE\_IMAGE);  // 如果是分享到空间  params.putInt(QQShare.SHARE\_TO\_QQ\_EXT\_INT,  QQShare.SHARE\_TO\_QQ\_FLAG\_QZONE\_AUTO\_OPEN);    //如果是分享到qq  params.putInt(QQShare.SHARE\_TO\_QQ\_EXT\_INT,  QQShare.SHARE\_TO\_QQ\_FLAG\_QZONE\_ITEM\_HIDE); |

### 分享音乐

略

### 分享应用

略

temp：

|  |  |
| --- | --- |
| {  “type”:,  “targetUrl”:目标url,  “title”:30字符内,  “image”:建议，使用云盘图标、或者图片本身，但是除了发起者，谁都看不到  “summary”:概要，建议，40字以内  “from”:建议，appname  “to”:区分qq还是空间  } | {  “type”:类型  “image”:  “from”:  “to”:  } |