手游社交组件

API 文档

产品文档

腾讯云

文档目录

API 文档
Android API 使用说明
QQ登录与注销

最近更新时间：2017-11-02 15:19:05

登录/授权

功能描述

通过调用 Tencent 类的 login 函数发起登录/校验登录态。

该API具有两个作用：

- 如果开发者没有调用 mTencent 实例的 setOpenId 、setAccessToken API，则该 API 执行正常的登录操作。
- 如果开发者先调用 mTencent 实例的 setOpenId 、setAccessToken API，则该 API 执行校验登录态的操作。
  - 如果登录态有效，则返回成功给应用。
  - 如果登录态失效，则会自动进入登录流程，将最新的登录态数据返回给应用。

注意：

- 建议开发者在每次应用启动时调用一次该 API (先调用 setOpenId 、setAccessToken)，以确保每次打开应用时用户都是有登录态的。
- 在某些低端机上调用登录后，由于内存紧张导致 APP 被系统回收，登录成功后无法成功回传数据。解决办法如下：
  在调用 login 的 Activity 或者 Fragment 重写 onActivityResult 方法，示例代码如下：

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
if(requestCode == Constants.REQUEST_API) {
if(resultCode == Constants.RESULT_LOGIN) {
mTencent.handleLoginData(data, loginListener);
}
super.onActivityResult(requestCode, resultCode, data);
}
```

方法原型

```
public void login(Activity activity, String scope, IUiListener listener)
```

参数说明

参数名	参数说明
activity	调用者 activity。应用使用 SDK 时，会从应用自己的 Activity 跳转到 SDK 的 Activity，应用调用 SDK 的 Activity 即为这里的调用者 activity。

参数名	参数说明
scope	应用需要获得的 API 的权限，由“，”分隔。例如：SCOPE = “get_user_info,add_t”；所有权限用 “all”。
listener	回调 API，IUiListener 实例。

返回码

返回码	返回码描述
10	解码失败。
110201	票据无效。
110405	登录请求被限制，请稍后在登录。
110404	请求参数缺少 APPID。
110401	请求的应用不存在。
110407	应用已经下架。
110406	应用没有通过审核。
100044	错误的 sign。
110500	获取用户授权信息失败。
110501	获取应用的授权信息失败。
110502	设置用户授权失败。
110503	获取 token 失败。
110504	系统内部错误。
110505	参数错误。
110506	获取 APP info 信息失败。
110507	校验 APP info 签名信息失败。
110508	获取 code 失败。
110509	SKEY 校验失败。
110510	应用被禁止登录。

实例演示

```
private void doLogin() {
IUiListener listener = new BaseUiListener() {
@Override
protected void doComplete(JSONObject values) {
updateLoginButton();
}
```

```
};
mTencent.login(this, SCOPE, listener);
}
```

注销

功能描述

通过调用 Tencent 类的 logout 函数注销。

方法原型

```
public void logout(Context context)
```

参数说明

参数名	参数说明
context	调用者的 context 。Context 是上下文的意思，每一个 Activity 都有对应的 Context 。示例中的 this 为调用者 Activity 对应的 Context。

实例演示

```
mTencent.logout(this);
```

分享消息到 QQ

最近更新时间：2017-11-02 15:25:40

功能描述

分享消息到 QQ 的接口。可将新闻、图片、文字、应用等消息分享给 QQ 好友、讨论组和群。Tencent 类的 shareToQQ 函数可直接调用，不用用户授权（使用手机 QQ 当前的登录态）。调用将打开分享的界面，用户选择好友、讨论组或群之后，单击【确定】即可完成分享，并进入相应的对话窗口。

本接口支持多种模式，每种模式的参数设置不同，下面分别进行介绍：

方法原型

public void shareToQQ(Activity activity, Bundle **params**, IUiListener listener)

参数说明

分享图文消息：

参数名	必选/可选	类型	参数说明
QQShare.SHARE_TO_QQ_KEY_TYPE	必选	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	必选	String	分享给好友的 URL。
QQShare.PARAM_TITLE	必选	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）： QQShare.SHARE_TO_QQ_FLAG_QZONE_AUTO_OPEN：分享时自动打开分享到 QZone 的对话框。 QQShare.SHARE_TO_QQ_FLAG_QZONE_ITEM_HIDE：分享时隐藏分享到 QZone按钮。

分享纯图片：

参数名	必选/可选	类型	参数说明
QQShare.SHARE_TO_QQ_KEY_TYPE	必选	Int	分享的类型。分享纯图片时填写QQShare.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）：QQShare.SHARE_TO_QQ_FLAG_QZONE_AUTO_OPEN：分享时自动打开分享到 QZone 的对话框。QQShare.SHARE_TO_QQ_FLAG_QZONE_ITEM_HIDE：分享时隐藏分享到 QZone按钮。

实际示例

分享图文消息：

```
private void onClickShare() {
final Bundle params = new Bundle();
params.putInt(QQShare.SHARE_TO_QQ_KEY_TYPE, QQShare.SHARE_TO_QQ_TYPE_DEFAULT);
params.putString(QQShare.SHARE_TO_QQ_TITLE, "要分享的标题");
params.putString(QQShare.SHARE_TO_QQ_SUMMARY, "要分享的摘要");
params.putString(QQShare.SHARE_TO_QQ_TARGET_URL, "http://www.qq.com/news/1.html");
params.putString(QQShare.SHARE_TO_QQ_IMAGE_URL,"http://imgcache.qq.com/qzone/space_item/pre/0/66768.gif");
params.putString(QQShare.SHARE_TO_QQ_APP_NAME, "测试应用222222");
params.putInt(QQShare.SHARE_TO_QQ_EXT_INT, "其他附加功能");
mTencent.shareToQQ(MainActivity.this, params, new BaseUiListener());
}
```

分享纯图片：

```
private void onClickShare() {
Bundle params = new Bundle();
params.putString(QQShare.SHARE_TO_QQ_IMAGE_LOCAL_URL,imageUrl.getText().toString());
params.putString(QQShare.SHARE_TO_QQ_APP_NAME, appName.getText().toString());
params.putInt(QQShare.SHARE_TO_QQ_KEY_TYPE, QQShare.SHARE_TO_QQ_TYPE_IMAGE);
params.putInt(QQShare.SHARE_TO_QQ_EXT_INT, QQShare.SHARE_TO_QQ_FLAG_QZONE_AUTO_OPEN);
mTencent.shareToQQ(MainActivity.this, params, new BaseUiListener());
}
```

分享到 QQ 空间

最近更新时间：2017-11-02 15:27:32

功能描述

分享类型参数 Tencent.SHARE_TO_QQ_KEY_TYPE，目前只支持图文分享，而 Tencent.shareToQzone() 函数既完善了分享消息到 QZone 的功能，又可直接调用，不用用户授权（使用手机 QQ 当前的登录态）。开发者可以调用此 API 打开手机 QQ 或浏览器 QQ 的空间界面进行分享操作。

方法原型

public void shareToQzone(Activity activity, Bundle **params**, IUiListener listener)

参数说明

参数名	必选/可选	类型	参数说明
QzoneShare.SHARE_TO_QQ_KEY_TYPE	选填	Int	SHARE_TO_QZONE_TYPE_IMAGE_TEXT（图文）。
QzoneShare.SHARE_TO_QQ_TITLE	必选	Int	分享的标题，最多 200 个字符。
QzoneShare.SHARE_TO_QQ_SUMMARY	选填	String	分享的摘要，最多 600 字符。
QzoneShare.SHARE_TO_QQ_TARGET_URL	必选	String	需要跳转的链接，URL 字符串。
QzoneShare.SHARE_TO_QQ_IMAGE_URL	选填	String	分享的图片, 以 ArrayList 的类型传入，以便支持多张图片（注：图片最多支持 9 张图片，多余的图片会被丢弃）。

注意：
QZone 接口暂不支持发送多张图片的能力，若传入多张图片，则会自动选入第一张图片作为预览图。分享多图的能力将会在以后实现。

实际示例

```
private void shareToQzone () {//分享类型
    params.putString(QzoneShare.SHARE_TO_QQ_KEY_TYPE,SHARE_TO_QZONE_TYPE_IMAGE_TEXT );
params.putString(QzoneShare.SHARE_TO_QQ_TITLE, "标题");//必填
params.putString(QzoneShare.SHARE_TO_QQ_SUMMARY, "摘要");//选填
params.putString(QzoneShare.SHARE_TO_QQ_TARGET_URL, "跳转URL");//必填
params.putStringArrayList(QzoneShare.SHARE_TO_QQ_IMAGE_URL, "图片链接ArrayList");
```

```
mTencent.shareToQzone(activity, params, new BaseUiListener());
}
```

添加游戏好友

最近更新时间：2017-11-02 15:37:45

功能描述

游戏 SDK 通过调用 Tencent 类的 gameMakeFriend API 添加游戏好友。

方法原型

public void gameMakeFriend(Activity activity, Bundle **params**)

参数说明

参数名	必选/可选	类型	参数说明
MGameAppOperation.GAME_FRIEND_OPENID	必选	String	要添加好友的 openid
MGameAppOperation.GAME_FRIEND_LABEL	必选	String	要添加好友的备注
MGameAppOperation.GAME_FRIEND_ADD_MESSAGE	必选	String	验证信息
MGameAppOperation.GAME_SDK_VERSION	必选	String	是否来自游戏 SDK

注意：
和原来的加好友接口不同：需要传递参数 MGameAppOperation.GAME_SDK_VERSION。

实际示例

```
Bundle params = new Bundle();
params.putString(MGameAppOperation.GAME_FRIEND_OPENID, fopenid.getText() + "");
params.putString(MGameAppOperation.GAME_FRIEND_LABEL, label.getText() + "");
params.putString(MGameAppOperation.GAME_FRIEND_ADD_MESSAGE, message.getText() + "");
params.putString(MGameAppOperation.GAME_SDK_VERSION, "true");
mTencent.makeFriend(GameLogicActivity.this, params);
```

一键加群

最近更新时间：2017-11-02 15:29:10

功能描述

通过调用 Tencent 类的 gameJoinQQGroup 调用绑定群接口。

> **注意：**
> CGI 调用需要登录态！若是没有登录态则不会跳到手 Q。

方法原型

public boolean **gameJoinGroup**(Activity activity, Bundle **params**)

参数说明

参数名	必选/可选	类型	参数说明
MGameAppOperation.GAME_GUILD_ID	必选	String	公会 ID
MGameAppOperation.GAME_GUILD_ZONE_ID	必选	String	游戏区域 ID
MGameAppOperation.GAME_ROLE_ID	必选	String	游戏角色 ID

实际示例

```
mTencent = Tencent.createInstance(APPID, this);
Bundle params = new Bundle();
params.putString(MGameAppOperation.GAME_GUILD_ID, guild_id.getText());
params.putString(MGameAppOperation.GAME_GUILD_ZONE_ID , zone_id.getText());
params.putString(MGameAppOperation.GAME_ROLE_ID, role_id.getText());
mTencent.gameJoinQQGroup(this, params);
```

一键绑群

最近更新时间：2017-11-02 15:30:37

功能描述

通过调用 Tencent 类的 gameBindGroup 调用绑定群接口。

> **注意：**
> CGI 调用需要登录态！若是没有登录态则不会跳到手 Q。

方法原型

public boolean **gameBindGroup**(Activity activity, Bundle **params**)

参数说明

参数名	必选/可选	类型	参数说明
MGameAppOperation.GAME_UNION_ID	必选	String	公会 ID。
MGameAppOperation.GAME_ZONE_ID	必选	String	游戏区域 ID。
MGameAppOperation.GAME_UNION_NAME	必选	String	公会名称。
MGameAppOperation.GAME_ROLE_ID	必选	String	游戏角色 ID。
MGameAppOperation.GAME_APP_KEY	必选	String	与 APPID 对应，第三方申请。

实际示例

```
mTencent = Tencent.createInstance(APPID, this);
Bundle params = new Bundle();
params.putString(MGameAppOperation.GAME_UNION_ID, unionid.getText().toString());
params.putString(MGameAppOperation.GAME_ZONE_ID , zoneid.getText());
params.putString(MGameAppOperation.GAME_UNION_NAME, union_name.getText());
params.putString(MGameAppOperation.GAME_ROLE_ID, role_id.getText());
params.putString(MGameAppOperation.GAME_APP_KEY, appkey.getText());
mTencent.gameBindGroup(this, params);
```

解除绑群

最近更新时间：2017-11-02 15:33:17

功能描述

通过调用 Tencent 类的 unBindGroup 调用解绑群接口。

> **注意：**
> CGI 调用需要登录态！若是没有登录态则不会跳到手 Q。

方法原型

public void unBindGroup(Context context, Bundle **params**, IUiListener listener)

参数说明

参数名	必选/可选	类型	参数说明
context	必选	Context	上下文。
params	必选	Bundle	外部传递参数，参数请查询下表。
listener	必选	IUiListener	回调实例。成功：返回码为 0 并且群号不为空。

参数名	必选/可选	类型	参数说明
MGameAppOperation.GAME_GUILD_ID	必选	String	公会 ID
MGameAppOperation.GAME_GUILD_ZONE_ID	必选	String	游戏区域 ID
MGameAppOperation.GAME_ROLE_ID	必选	String	游戏角色 ID

返回值

返回码	返回码描述
0	成功
221002	未绑定群
221005	当前用户不是群主

实际示例

```
mTencent = Tencent.createInstance(APPID, this);
Bundle params = new Bundle();
params.putString(MGameAppOperation.GAME_GUILD_ID, guild_id.getText());
params.putString(MGameAppOperation.GAME_GUILD_ZONE_ID , zone_id.getText());
params.putString(MGameAppOperation.GAME_ROLE_ID, role_id.getText());
IUiListener listener = new IUiListener() {
@Override
public void onComplete(Object response) {
Util.toastMessage(this, “onComplete:” + response.toString());
JSONObject jsonObject = (JSONObject )response;
try{
String retMsg = jsonObject.getString( “retMsg” );
int retCode = jsonObject.getInt( “retCode” );
//todo 根据返回码处理
}catch (JSONException e) {
e.printStackTrace();
}
}
@Override
public void onCancel() {
Util.toastMessage(this, “onCancel ” );
}
@Override
public void onError(UiError e) {
Util.toastMessage(this, “onError:” + e.errorMessage, “e” );
}
}
mTencent.unBindGroup(this, params , listener);
```

查询公会是否绑定群

最近更新时间：2017-11-02 15:34:38

功能描述

通过调用 Tencent 类的 checkBindGroup 调用查询公会是否绑定群接口。

> **注意：**
> CGI 调用需要登录态！若是没有登录态则不会跳到手 Q。

方法原型

```
public void checkBindGroup(Context context, Bundle params, IUiListener listener)
```

参数说明

参数名	必选/可选	类型	参数说明
context	必选	Context	上下文。
params	必选	Bundle	外部传递参数，参数请查询下表。
listener	必选	IUiListener	回调实例。成功：返回码为 0 并且群号不为空。

参数名	必选/可选	类型	参数说明
MGameAppOperation.GAME_GUILD_ID	必选	String	公会 ID
MGameAppOperation.GAME_GUILD_ZONE_ID	必选	String	游戏区域 ID
MGameAppOperation.GAME_ROLE_ID	必选	String	游戏角色 ID

实际示例

```
mTencent = Tencent.createInstance(APPID, this);
Bundle params = new Bundle();
params.putString(MGameAppOperation.GAME_GUILD_ID, guild_id.getText());
params.putString(MGameAppOperation.GAME_GUILD_ZONE_ID , zone_id.getText());
params.putString(MGameAppOperation.GAME_ROLE_ID, role_id.getText());
IUiListener listener = new IUiListener() {
```

```
@Override
public void onComplete(Object response) {
Util.toastMessage(this, "onComplete:" + response.toString());
JSONObject jsonObject = (JSONObject )response;
try{
String retMsg = jsonObject.getString( "retMsg" );
int retCode = jsonObject.getInt( "retCode" );
//todo 根据返回码处理
}catch (JSONException e) {
e.printStackTrace();
}
}
@Override
public void onCancel() {
Util.toastMessage(this, "onCancel " );
}
@Override
public void onError(UiError e) {
Util.toastMessage(this, "onError:" + e.errorMessage, "e" );
}
}
mTencent.unBindGroup(this, params , listener);
```

查询公会 id 和成员 openid 关系

最近更新时间：2017-11-02 15:35:42

功能描述

通过调用 Tencent 类的 checkJoinGroup 调用查询工会和成员关系接口。

> **注意：**
> CGI 调用需要登录态！若是没有登录态则不会跳到手 Q。

方法原型

```
public void checkJoinGroup(Context context, Bundle params, IUiListener listener)
```

参数说明

参数名	必选/可选	类型	参数说明
context	必选	Context	上下文。
params	必选	Bundle	外部传递参数，参数请查询下表。
listener	必选	IUiListener	回调实例。成功：返回码为 0 并且群号不为空。

参数名	必选/可选	类型	参数说明
MGameAppOperation.GAME_GUILD_ID	必选	String	公会 ID
MGameAppOperation.GAME_GUILD_ZONE_ID	必选	String	游戏区域 ID

实际示例

```
mTencent = Tencent.createInstance(APPID, this);
Bundle params = new Bundle();
params.putString(MGameAppOperation.GAME_GUILD_ID, guild_id.getText());
params.putString(MGameAppOperation.GAME_GUILD_ZONE_ID , zone_id.getText());

IUiListener listener = new IUiListener() {
@Override
public void onComplete(Object response) {
```

```
Util.toastMessage(this, "onComplete:" + response.toString());
JSONObject jsonObject = (JSONObject )response;
try{
String retMsg = (String)jsonObject.getString("retMsg");
int retCode = (Integer)jsonObject.getInt("retCode");
int relation = (Integer)jsonObject.getInt("relation");
//todo 根据返回码 和 群关系进行处理
}catch (JSONException e) {
e.printStackTrace();
}
}
@Override
public void onCancel() {
Util.toastMessage(this, "onCancel ");
}
@Override
public void onError(UiError e) {
Util.toastMessage(this, "onError:" + e.errorMessage, "e");
}
}
mTencent.checkJoinGroup(this, params , listener);
```

Android 游戏 SDK 错误码解析

最近更新时间：2017-11-02 15:42:54

Android 游戏 SDK 错误码解析如下：

错误码	错误码含义
221001	已有绑定群
221002	未绑定群
221003	绑群回包错误
221004	登录态错误
221005	当前用户不是群主
221006	无效的 openid
221007	参数无效
221008	该 APP 无接口调用权限
221009	不是公会成员
221010	达到创建群上限
221011	创建群频率过高
221012	群被删除了

注意：
其他错误码为系统错误。

内部浏览器接入（兴趣部落）

最近更新时间：2017-11-02 15:36:48

AndroidManifest 配置

样例如下：

```
<activity
android:name="com.tencent.connect.webview.BrowserActivity"
android:configChanges="orientation|screenSize"
android:process=":web" >
<meta-data
android:name="titlebar_hideable"
android:value="true"/>

<meta-data
android:name="toolbar_portrait_hideable"
android:value="false"/>

<meta-data
android:name="toolbar_landscape_hideable"
android:value="false"/>
</activity>

<activity
android:name="com.tencent.connect.webview.JumpShareActivity" />
```

注意：
如果浏览器需要运行在不同的进程，可参照样例将 BrowserActivity 设置在不同的进程中，但是 JumpShareActivity 必须在主进程中（持有 Tencent 对象的进程）。另外如果运行屏幕旋转，建议设置：

```
android:configChanges="orientation|screenSize"，
```

滑动隐藏特性

Game SDK 浏览器组件支持向上滑动时隐藏导航栏以给玩家更大的浏览空间，此项功能是可配置的。参考配置样例，各个参数意义如下：

参数	参数含义
titlebar_hideable	是否隐藏标题栏。
toolbar_portrait_hideable	浏览器竖屏时，导航栏是否可滑动隐藏(true 为可滑动隐藏)。
toolbar_landscape_hideable	浏览器横屏时，导航栏是否可滑动隐藏。

打开浏览器

内部浏览器支持提供 bid 直接打开兴趣部落，调用样例：

```
WebViewHelper.getInstance().setTencent(mTencent).openBuluo(v.getContext(), 227061, ActivityInfo.SCREEN_ORIENTATION_SENSOR);
```

打开 HTTPS 的链接，调用样例：

```
WebViewHelper.getInstance().setTencent(mTencent).openUrl(v.getContext(), "https://www.qq.com", ActivityInfo.SCREEN_ORIENTATION_SENSOR);
```

> **注意：**
> 内部浏览器仅支持 HTTPS 链接。

setTencent(Tencent tencent) 方法传入通过 Game SDK 创建的 Tencent 对象，要求非空。

openBuluo(Context context,long bid,int screenOrientation) 方法通过传入 bid 和屏幕方向，打开兴趣部落页面，要求 context 非空。

openUrl(Context context, String url, int screenOrientation) 接口通过传入 url 和屏幕方向，打开相应页面。

screenOrientation 参数：

参数名	参数说明
ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE	固定横屏
ActivityInfo.SCREEN_ORIENTATION_PORTRAIT	固定竖屏
ActivityInfo.SCREEN_ORIENTATION_SENSOR	支持用户旋转屏幕

QQ 钱包支付方式

最近更新时间：2018-09-19 15:48:06

请参考 QQ 钱包 文档。

会员接口（后台接口）

最近更新时间：2019-03-06 20:26:16

会员包月活动

适用场景

用户在游戏 App 内开通会员包月服务，可获得专属礼包，同时刺激双方用户相互活跃。

接入方式

步骤 1: 业务方将游戏互联 APPID、游戏名称、管理员 QQ 和回调接口统一提交给 QQ 会员侧，由会员侧提供回调密钥和 aid。
步骤 2: 业务方在游戏 App 内根据URL参数规则拼凑 URL，通过 openUrl 接口打开内部浏览器跳转到支付页面。

其中 URL 为 `https://mq.vip.qq.com/m/connect/open` ，参数以 GET 参数形式传入，例如：`https://mq.vip.qq.com/m/connect/open?code=svip&month=1&aid=mvip.p.hz.demo`

URL 参数说明

参数	是否必须	说明	备注
code	是	小写的业务代码	svip：超级会员 vip：QQ 会员
month	是	开通月份	整数
aid	是	业务渠道	由 QQ 会员侧分配，必须保证准确无误。

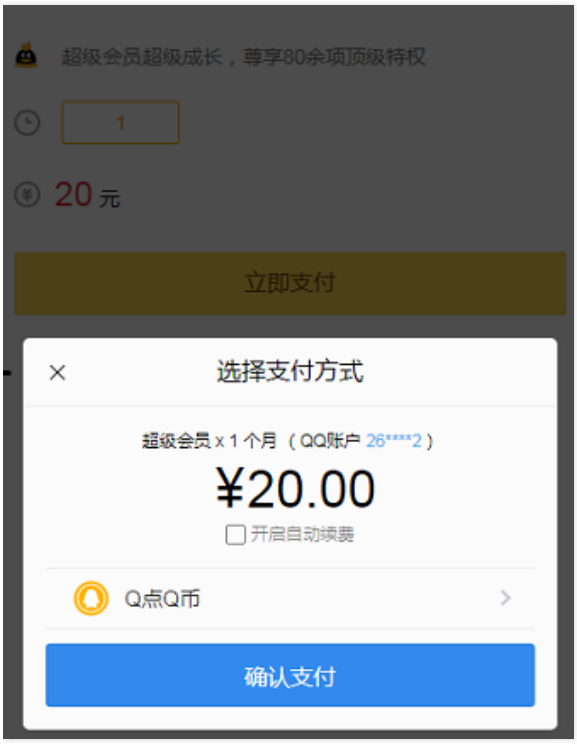
Android-游戏开放 SDK 调用样例：
`openUrl(Context context, String url, int screenOrientation)` 接口通过传入 url 和屏幕方向，打开相应页面

```
WebViewHelper.getInstance().setTencent(mTencent).openUrl(v.getContext(), "https://mq.vip.qq.com/m/connect/open?code=svip&month=1&aid=mvip.p.hz.demo", ActivityInfo.SCREEN_ORIENTATION_SENSOR);
```

效果展示

展示界面由业务方设计开发。

单击【开通超级会员】按钮弹出内部浏览器。



开通流水接入指引

管理端使用

对外开发的管理端可以提供给外部录入用户配置信息和开通 aid（开通来源标记，用户录入信息时由系统自动生成）标记信息。
其中用户配置信息标识游戏情况，开通 aid 配置信息标识该游戏在不同场景下的开通情况。

配置信息录入

用户配置信息如下：

名称	备注

名称	备注
appid	每款游戏的标识(用户申请，必填)
url	开通流水接收地址(比如一个 cgi 地址，必填)
userinfo	申请人的联系方式(例如 QQ 号码、手机号码等，必填)
appkey	转发流水过程中校验 key(与 appid 对应，接收方根据该 key 校验数据，系统生成)
desc	描述信息(例如游戏类别)

⚠ 注意：

- 用户配置信息中，appid 为主键，每款游戏有一个对应的 appid；
- 针对新添加的 appid，系统负责生成 appkey，在以后信息修改的过程中不会修改 appkey；开通流水转发过程中使用 appkey+data 进行 md5 加密，流水接收方根据 data 和 appkey 校验数据。

用户配置信息标识了一款游戏的情况，根据用户开通情况以及游戏运营活动种类的不同，可以为该款游戏配置多种情况下的开通 aid 信息，aid 配置信息表如下：

名称	备注
aid	款游戏的标识(用户申请，系统生成)
appid	开通流水接收地址(比如一个 cgi 地址，必填)
openmonth	该 aid 对应的开通月份(必填)
desc	描述信息(例如开通页面入口等)

⚠ 注意：

aid 配置信息表中 aid 为主键，每款游戏可以有多个 aid。

数据的录入默认都会提交到测试环境，如果配置信息需要发布现网，可以提交审核，审核通过10分钟内流水转发生效。

配置信息查询

- 用户配置信息的查询支持 appid、userinfo 字段的精确查询 。
- 开通 aid 信息的查询支持 aid 的模糊查询、appid 的精确查询。

开通流水转发

开通方式的接入可以参考 会员包月活动 接入文档。

开通流水的转发流程：

1. 根据用户开通流水中的 aid 判断是否属于合作方的开通。
2. 根据该 aid 在配置表中的转发地址转发开通数据，通过 HTTP/HTTPS 请求，报体数据格式为 json。

流水转发请求

流水转发使用 HTTP/HTTPS 请求转发的方式，其中 url 为管理端的用户配置信息内容。其他字段和含义下面举例说明：
某款游戏申请到的 appid：1111
流水接收地址为： http://10.100.70.5/cgi-bin/srfentry.fcgi?
数据校验 appkey：ffff
该款游戏下合法的 aid 为：aaaa
那么会员侧在接收到用户开通之后，根据用户的配置信息，流水转发的 url 为（data 为 urlencode）：
http://proxyvac.qq.com/cgi-bin/srfentry.fcgi?ts=1508210911065&data=
{"aid":"aaaaa","msg_time":"1507799551","open_months":"1","open_type":"vip","openid":"xjfdaldfjrefjfljljgfjg","order_id":"20171012-1712128958"}&sign=fefaf9ddd9d4dce23da3b664ec457413

请求参数

ulr 中各个字段的说明如下：

http://proxyvac.qq.com/cgi-bin/srfentry.fcgi //用户在配置表中的转发url
Ts=1508210911065 //流水开始转发的时间戳(ms)

data 域为 json 格式数据(每个字段值都为 string)

```
{
"aid": "aaaaa",
"msg_time": "1507799551",
"open_months": "1",
"open_type": "vip",
"openid": "xjfdaldfjrefjfljljgfjg",
"order_id": "20171012-1712128958"
}
sign=fefaf9ddd9d4dce23da3b664ec457413 //data域数据(在这里是上面json数据)appkey组成字符串后md5值，md5(data+appkey)
```

请求 data 域字段说明

字段名	类型	说明
aid	String	开通 aid
msg_time	String	开通时间戳
open_months	String	开通 aid
msg_time	String	开通时长(单位为月)
open_type	String	开通类型(vip：会员，svip：超级会员)
openid	String	用户 openid
order_id	String	开通订单 ID

⚠ **注意：**

openid 和 order_id 可以唯一确定用户的开通记录，接收方需要处理流水去重。

流水转发回应

对应开通流水的转发结果，需要返回接收结果情况。

返回参数

返回包体约定为 json 数据格式，包含的字段及其含义如下所示：

```
{
"ret": 0,
"msg": "succ"
}
```

返回字段说明

字段名	类型	说明
ret	int	返回值，0 为正常
msg	String	返回信息

说明：

流水转发统一使用 get 请求，根据用户表中转发地址的配置，支持 HTTP 和 HTTPS 两种方式。

异账号判断

最近更新时间：2018-01-15 15:09:43

使用腾讯云手游社交组件接口时，如果需要判断当前登录游戏的账号是否是之前已绑定的账号，可以通过调用 `Tencent.getOpenId()` 接口来获取当前已登录的账号的 openid，用此 openid 和已绑定的账号 openid 进行对比。

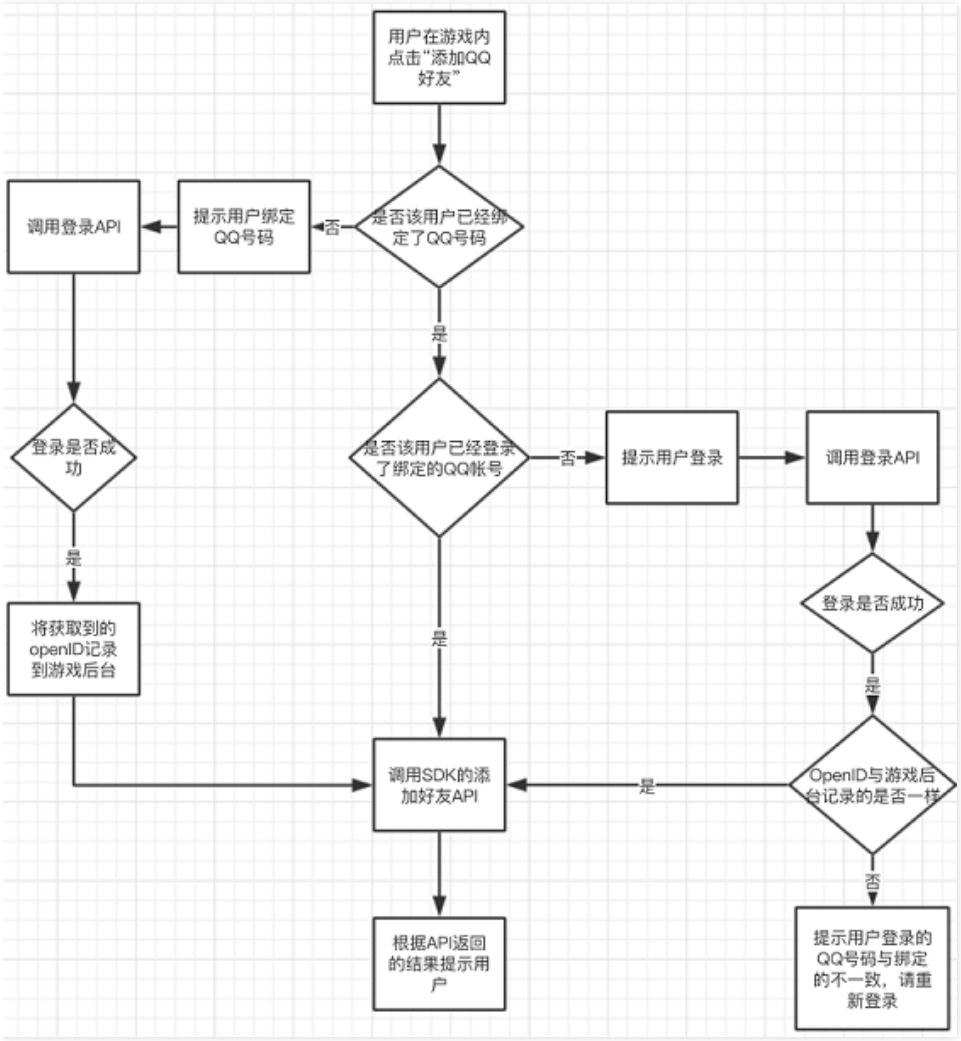
无登录状态使用说明

最近更新时间：2018-01-15 14:23:18

游戏 SDK 支持无登录态调用 API，游戏接入方无须使用手机 QQ 登录游戏，仅需要在调用某些需要登录态的 API 时候进行一次 QQ 帐号和游戏账号的绑定即可。

目前强制需要登录之后才能调用的 API 包括：游戏加群、绑群、解散群功能、创建 QZone相册、获取相册列表、上传图片、发表说说、验证空间粉丝、游戏 SDK 新增 API。

游戏接入方需要在后台数据库记录一个游戏账号绑定的 QQ OpenID 字段，当用户首次使用登录 API 的功能时，提示用户将此 QQ 帐号与当前登录的游戏账号绑定，用户确认之后再调用 SDK 内的登录接口，获取到 OpenID 之后记录到游戏后台。后续如果用户再次使用登录 API 功能，游戏接入方需要首先判断用户是否已经绑定了 QQ 帐号，然后进一步判断绑定的 QQ 帐号是否已经登录（通过是否 TencentOAuth 的 accessToken 值是否为空来判断）。具体的流程如下（简化版）：



社交分享消息

最近更新时间：2018-01-15 14:30:26

功能描述

使用 GET 调用方式，实现点对点定向分享(分享消息给手机 QQ 好友，在公众账号中显示)。

CGI 名字

send_share_for_inner

输入参数

参数	是否必须	含义
fopenids	是	接收方信息
imageUrl	是	消息缩略图
targetUrl	是	跳转链接
game_tag	是	消息类型： MSG_INVITE(邀请) MSG_FRIEND_EXCEED(超越炫耀) MSG_HEART_SEND(送心) MSG_SHARE_FRIEND_PVP(PVP对战)
dst	是	msf-手q(包括 iphone, android qq等)，目前只能填 1001
flag	是	漫游 (0：是；1：否，目前只能填 1)

实际示例

输入示例

```
http://proxy.vip.qq.com/cgi-bin/QQConnect.fcgi?cmd=send_share_for_inner&access_token=8d340568 2be0dd70510325272b495c5e&fopenids=[{"openid":"1983e77ac7517e997967168816c41867","type":0}]&imageUrl=http://imgcache.qq.com/club/gamecenter/cms/2016-04-01/1459504923_868_750x1182.jpg&targetUrl=http://gamecenter.qq.com/gcjump?appid=1104466820&oauth_consumer_key=100330589&openid=1983e77ac7517e997967168816c41867&dst=1001&flag=1&game_tag=MSG_SHARE_FRIEND_DOUBLE&sign=86ad69be5551c514b39693fc436c89e4
```

输出示例

```
{
"ret":0, //中转cgi返回码
```

```
"msg":"ok", //中转cgi返回信息
"data":
    {
"ret":0, //0-成功，其他-失败
"msg":"xx", //错误描述
    }
}
```

`

iOS API 使用说明
授权登录

最近更新时间：2017-11-02 15:46:24

功能描述

整个授权过程可以分为 **发送请求获取用户授权** 和 **获取用户授权结果** 两个部分。

- 发送请求获取用户授权：
  在取得用户授权之前，开发者必须清楚自己需要用户的哪些信息。 iOS SDK 提供多种选择，开发者可以根据自己的需要请求用户进行授权。具体可以获取的授权信息参见下表的参数说明。设置完需要请求的授权信息之后即可发送请求。

> **注意：**
> inSafari 参数从 iOS SDK1.3 版本后弃用。

- 获取用户授权结果:
  获取用户授权结果，采用的是代理回调的方式，所以开发者需要实现 TencentLoginDelegate 协议。登录结果分为三种：登录成功、登录失败和登录取消。

```
- (void)tencentDidLogin; // 登录成功后的回调
- (void)tencentDidNotLogin:(BOOL)cancelled; // 登录失败后的回调
- (void)tencentDidNotNetWork; // 登录时网络有问题的回调
```

方法原型

```
/**
* 登录授权
*
* \param permissions 授权信息列
*/
- (BOOL)authorize:(NSArray *)permissions;
```

参数说明

参数名	参数说明
kOPEN_PERMISSION_GET_USER_INFO	获取用户信息。
kOPEN_PERMISSION_GET_SIMPLE_USER_INFO	移动端获取用户信息。

参数名	参数说明
kOPEN_PERMISSION_GET_INFO	获取登录用户自己的详细信息。
kOPEN_PERMISSION_GET_VIP_RICH_INFO	获取会员用户详细信息。
kOPEN_PERMISSION_GET_VIP_INFO	获取会员用户基本信息。
kOPEN_PERMISSION_GET_OTHER_INFO	获取其他用户的详细信息。
kOPEN_PERMISSION_ADD_TOPIC	发表一条说说到QQ空间 (**需要申请权限**)。
kOPEN_PERMISSION_ADD_ONE_BLOG	发表一篇日志到QQ空间 (**需要申请权限**)。
kOPEN_PERMISSION_ADD_ALBUM	创建一个QQ空间相册 (**需要申请权限**)。
kOPEN_PERMISSION_UPLOAD_PIC	上传一张照片到QQ空间相册 (**需要申请权限**)。
kOPEN_PERMISSION_LIST_ALBUM	获取用户QQ空间相册列表 (**需要申请权限**)。
kOPEN_PERMISSION_ADD_SHARE	同步分享到QQ空间、腾讯微博。
kOPEN_PERMISSION_CHECK_PAGE_FANS	验证是否认证空间粉丝。

实际示例

发送请求获取用户授权：

```
// 其中_tencentOAuth为TencentOAuth类的实例
NSArray* permissions = [NSArray arrayWithObjects:kOPEN_PERMISSION_GET_USER_INFO,
kOPEN_PERMISSION_GET_SIMPLE_USER_INFO,kOPEN_PERMISSION_ADD_SHARE,nil];
TencentOAuth *_tencentOAuth = [TencentOAuth initWithAppid:appid andDelegate:delegate];
[_tencentOAuth authorize:permissions inSafari:NO];
```

获取用户授权结果：

```
// 登录时权限信息的获得
- (NSArray *)getAuthorizedPermissions:(NSArray *)permissions
withExtraParams:(NSDictionary *)extraParams;
```

分享到 QQ 好友

最近更新时间：2017-11-02 15:47:34

功能描述

在用户安装了手机 QQ 时通过手机 QQ 进行分享，否则调用浏览器页面进行分享。其中文本消息，图文消息和音频消息的 title 是必须填写的，summary 可以不填，具体调用请参考实际示例。使用分享到 QQ 好友功能需要设置 QQ 业务回调，请参考处理 QQ 业务的回调。

方法原型

```
/**
向手Q发起分享请求
\param req 分享内容的请求
\return 请求发送结果码
*/
+ (QQApiSendResultCode)sendReq:(QQBaseReq *)req;
```

参数说明

参数名	必选	类型	参数说明
req	必选	QQBaseReq*	分享内容的请求

实际示例

纯文本分享

```
//开发者分享的文本内容
QQApiTextObject *txtObj = [QQApiTextObject objectWithText:@"text"];
SendMessageToQQReq *req = [SendMessageToQQReq reqWithContent:txtObj];
//将内容分享到qq
QQApiSendResultCode sent = [QQApiInterface sendReq:req];
```

纯图片分享

```
//开发者分享图片数据
NSData *imgData = [NSData dataWithContentsOfFile:path];
QQApiImageObject *imgObj = [QQApiImageObject objectWithData:imgData
previewImageData:imgData
title:@"title"
description :@"description"];
SendMessageToQQReq *req = [SendMessageToQQReq reqWithContent:imgObj];
```

```
//将内容分享到qq
QQApiSendResultCode sent = [QQApiInterface sendReq:req];
```

分享文件（仅数据线）（2.8.1）

```
NSString *filePath = [[[NSBundle mainBundle] resourcePath] stringByAppendingPathCo mponent:@"test.txt"];
NSData *fileData = [NSData dataWithContentsOfFile:filePath];
QQApiFileObject *fileObj = [QQApiFileObject
objectWithData:fileData
previewImageData:nil
title:self.binding_title ? : @""
description:self.binding_description ? : @""];
if (self.binding_description != nil && ![self.binding_description isEqualToString:@""])
fileObj.fileName = self.binding_description;
else
fileObj.fileName = @"test.txt";
[fileObj setCflag:kQQAPICtrlFlagQQShareDataline];
SendMessageToQQReq *req = [SendMessageToQQReq reqWithContent:fileObj];
//将内容分享到qq
//QQApiSendResultCode sent = [QQApiInterface sendReq:req];
```

分享到 QQ 空间

最近更新时间：2017-11-02 16:09:34

功能描述

分享到 QQ 空间的接口用于取代旧版本的分享接口 addShareWithParams（该接口已被废弃）。

在用户安装了手机 QQ（4.6 版本以上）时通过手机 QQ 中的 QZone 结合版进行分享，否则调用浏览器页面进行分享。其中文本消息，图文消息和音频消息的 title 是必须填写的，summary 可以不填。使用分享到 QQ 空间功能需要设置 QQ 业务回调，请参考处理 处理 QQ 业务的回调。

在分享到 QQ 好友和 QQ 空间的时候，根据是本地分享还是浏览器分享的不同，支持分享的消息类型也不同。因为 webQQ 好友分享和 webQQ 空间的分享都不支持非 URL 类型的分享，所以这里建议在分享到 QQ 好友或者 QQ 空间的时候尽量避免这两种类型的调用，避免发生不支持的错误。下表列举了在不同应用中，不同消息类型是否被支持分享。

分享消息类型	QQ好友	QQ空间	web QQ好友	web QQ空间
QQApiTextObject	支持	不支持	不支持	不支持
QQApiImageObject	支持	不支持	不支持	不支持
QQApiNewsObject	支持	支持	支持	支持
QQApiAudioObject	支持	支持	支持	支持
QQApiVideoObject	支持	支持	支持	支持
QQApiGroupTribeImageObject	仅群部落	不支持	不支持	不支持
QQApiAddFriendObject	游戏好友	不支持	不支持	不支持
QQApiFileObject	仅数据线	不支持	不支持	不支持
QQApiGameConsortiumBindingGroupObject	仅群部落	不支持	不支持	不支持

方法原型

```
/**
向手Q QZone结合版发起分享请求
\note H5分享只支持单张网络图片的传递
\param req 分享内容的请求
\return 请求发送结果码
*/
+ (QQApiSendResultCode)SendReqToQZone:(QQBaseReq *)req;
```

参数说明

参数名	必选	类型	参数说明
req	必选	QQBaseReq *	分享内容的请求

获取用户信息

最近更新时间：2017-11-02 15:49:20

功能描述

开发者可以通过调用 API 获取 用户的个人信息。

获取用户信息分为两个步骤：

1. 发送获取用户信息的请求

```
// _tencentOAuth为TencentOAuth类的实例
[_tencentOAuth getUserInfo])
```

2. 实现协议 TencentSessionDelegate 的获取到用户信息后的回调方法

```
//可以通过response获取数据的返回结果
- (void)getUserInfoResponse:(APIResponse*) response;
```

方法原型

```
/**
* 获取用户个人信息
* \return 处理结果，YES表示API调用成功，NO表示API调用失败，登录态失败，重新登录
*/
- (BOOL)getUserInfo;
```

实际示例

```
- (void)getUserInfoResponse:(APIResponse*) response {
if (URLREQUEST_SUCCEED == response.retCode
&& kOpenSDKErrorSuccess == response.detailRetCode) {
NSMutableString *str = [NSMutableString stringWithFormat:@""];
for (id key in response.jsonResponse) {
[str appendString: [NSString stringWithFormat:
@"%@:%@\n", key, [response.jsonResponse objectForKey:key]]];
}
UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"操作成功"
message: [NSString stringWithFormat:@"%@",str];
delegate:self cancelButtonTitle:@"我知道啦" otherButtonTitles: nil];
[alert show];
} else {
NSString *errMsg = [NSString stringWithFormat:@"errorMsg:%@\n%@",
response.errorMsg, [response.jsonResponse objectForKey:@"msg"]];
```

```
UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"操作失败"
message:errMsg
delegate:self
cancelButtonTitle:@"我知道啦"
otherButtonTitles: nil];
[alert show];
}
}
```

调用 OpenAPI

最近更新时间：2017-11-02 15:50:15

功能描述

iOS SDK 中具体支持的 API 种类和每条 API 的参数说明，请参照 API 列表。这里用设置用户头像举例说明。

OpenAPI参数字典封装

在封装各接口的参数字典时，推荐使用为每个接口新增的参数封装辅助类，如：接口 (BOOL)addShareWithParams: (NSMutableDictionary *)params 对应辅助类 TCAddShareDic。

TCAddShareDic 辅助类中属性 @property (nonatomic, retain) TCRequiredStr paramTitle 对应于 CGI 请求中参数 title。

参数名	类型
TCRequiredStr（必选）	String
TCOptionalStr（可选）	String

使用增量授权

当第三方应用调用某个 API 接口时，如果服务器返回操作未被授权，则会触发增量授权逻辑。第三方应用需自行实现 tencentNeedPerformIncrAuth:withPermissions 协议接口才能够进入增量授权逻辑，否则默认第三方应用放弃增量授权。示例如下：

```
- (BOOL)tencentNeedPerformIncrAuth:(TencentOAuth *)tencentOAuth withPermissions: (NSArray *) permissions {
// incrAuthWithPermissions是增量授权时需要调用的登录接口
// permissions是需要增量授权的权限列表
[tencentOAuth incrAuthWithPermissions:permissions];
// 返回NO表明不需要再回传未授权API接口的原始请求结果，否则返回YES
return NO;
}
```

注意：
在用户通过增量授权页重新授权登录后，第三方应用需更新自己维护的 token 及有效期限等信息。

用户在增量授权时是可以更换帐号进行登录的，强烈要求第三方应用核对增量授权后的用户 openid 是否一致，以添加必要的处理逻辑（用户帐号变更需重新拉取用户的资料等信息）。

- 增量授权成功时，会通过 tencentDidUpdate 协议接口通知第三方应用：

```
- (void)tencentDidUpdate:(TencentOAuth *)tencentOAuth {
_labelTitle.text = @"增量授权完成";
if (tencentOAuth.accessToken
&& 0 != [tencentOAuth.accessToken length]) {
// 在这里第三方应用需要更新自己维护的token及有效期限等信息
// **务必在这里检查用户的openid是否有变更，变更需重新拉取用户的资料等信息**
```

```
_labelAccessToken.text = tencentOAuth.accessToken;
} else {
_labelAccessToken.text = @"增量授权不成功，没有获取accesstoken";
}
}
```

- 增量授权失败时，会通过 tencentFailedUpdate 协议接口通知第三方应用：

```
- (void)tencentFailedUpdate:(UpdateFailType)reason {
switch (reason) {
case kUpdateFailNetwork:
_labelTitle.text=@"增量授权失败，无网络连接，请设置网络";
break;
case kUpdateFailUserCancel:
_labelTitle.text=@"增量授权失败，用户取消授权";
break;
case kUpdateFailUnknown:
default:
_labelTitle.text=@"增量授权失败，未知错误";
break;
}
}
```

返回数据说明

APIResponse属性：

参数名	参数说明
retCode	网络请求返回码，主要表示服务器是否成功返回数据。
seq	请求的序列号，依次递增，方便内部管理。
errorMsg	错误消息。
jsonResponse	由服务器返回的 json 格式字符串转换而来的 json 字典数据（具体参数字段请参见对应 API 说明文档）。
message	服务器返回的原始字符串数据。
detailRetCode	新增的详细错误码，以区分不同的错误原因（v1.2 以及之前的 SDK 接口无此参数）。

返回码

retCode 网络请求返回码说明：

返回码	含义
0	表示成功，请求成功发送到服务器，并且服务器返回的数据格式正确。

返回码	含义
1	表示失败，可能原因有网络异常，或服务器返回的数据格式错误，无法解析。

detailRetCode 详细错误码说明：

错误码类别	返回码	含义
公共错误码：	kOpenSDKInvalid	无效的错误码
	kOpenSDKErrorSuccess	成功
	kOpenSDKErrorUnknown	未知错误
	kOpenSDKErrorUserCancel	用户取消
	kOpenSDKErrorReLogin	token 无效或用户未授权相应权限需要重新登录
	kOpenSDKErrorOperationDeny	第三方应用没有该 API 操作的权限
网络相关错误码：	kOpenSDKErrorNetwork	网络错误，网络不通或连接不到服务器
	kOpenSDKErrorURL	URL 格式或协议错误
	kOpenSDKErrorDataParse	数据解析错误，服务器返回的数据解析出错
	kOpenSDKErrorParam	传入参数错误
	kOpenSDKErrorConnTimeout	http 连接超时
	kOpenSDKErrorSecurity	安全问题
	kOpenSDKErrorIO	下载和文件 IO 错误
	kOpenSDKErrorServer	服务器端错误
webview中特有错误：	kOpenSDKErrorWebPage	页面错误
设置头像自定义错误码段：	kOpenSDKErrorUserHeadPicLarge	图片过大 设置头像自定义错误码

供第三方游戏使用接口

最近更新时间：2017-11-02 15:51:02

添加游戏好友

```
QQApiAddFriendObject *object = [[QQApiAddFriendObject alloc] initWithOpenID:@"A57597D3BA6F31A0F5D0049A354756FC"];
object.description = @"一起玩游戏吧";
object.subID = @"1234";
object.remark = @"备注这里填写";
SendMessageToQQReq* req = [SendMessageToQQReq reqWithContent:object];
QQApiSendResultCode sent = [QQApiInterface sendReq:req];
```

注意：

APPID 要有添加好友的权限。

游戏公会

错误码定义：

GuildErrorCode	说明
GuildErrNone	
GuildErrNet	网络错误， 请求失败。
GuildErrLocalApi	调用本地接口错误， 具体错误码见回调 QQApiSendResultCode。
GuildErrLocalParamsInvalid	本地参数错误。
GuildErrInvalidLogin	无效的登录状态。
GuildErrExistBind	已有绑定群，逻辑错误。
GuildErrNonExistBind	未绑定群，逻辑错误。
GuildErrInvalidGrpOwner	当前用户不是群主。
GuildErrParamsInvalid	参数无效。
GuildErrInvalidRight	该 APP 无接口调用权限。
GuildErrInvalidMember	不是公会成员。
GuildErrOverGrpCout	达到创建群上限。
GuildErrOverCreateGrp	创建群频率过高。

GuildErrorCode	说明
GuildErrGrpDeleted	群被删除了。
GuildErrOther	其他错误。

注意：
下面的 CGI 调用都需要有登录态！

绑定公会和 QQ 群

```
TCGuildBindGroupDic *info = [[TCGuildBindGroupDic alloc] init];
info.paramRoleId = @"角色id";
info.paramZoneId = @"分区id";
info.paramGuildId =@"公会id";
info.paramAppKey = @"appkey"; // appkey为申请appid时平台生成的
[_tencentOAuth sendGuildBindGroup:info callback:^(TCGuildRspBindGroupInfo *info) {

}];
```

其中，`_tencentOAuth` 为 `TencentOAuth` 类的实例。

查询公会和 QQ 群是否绑定

```
TCGuildQueryBindStateDic *info = [TCGuildQueryBindStateDic dictionary];
info.paramZoneId = @"分区id";
info.paramGuildId = @"公会id";
info.paramRoleId = @"角色id";
[_tencentOAuth sendGuildQueryBindState:info callback:^(TCGuildRspQueryBindStateInfo *info) {
//处理回调结果
}];
```

其中，`_tencentOAuth` 为 `TencentOAuth` 类的实例。

解绑公会(群主校验)

```
TCGuildUnBindDic *info = [TCGuildUnBindDic dictionary];
info.paramZoneId = @"分区id";
info.paramGuildId = @"公会id";
info.paramRoleId = @"角色id";
[_tencentOAuth sendGuildUnbind:info callback:^(TCGuildRspUnBindInfo *info) {
//处理回调结果
}];
```

其中， _tencentOAuth 为 TencentOAuth 类的实例。

一键加群

```
TCGuildJoinGroupDic *info = [TCGuildJoinGroupDic dictionary];
info.paramZoneId = @"分区id";
info.paramGuildId = @"公会id";
info.paramRoleId = @"角色id";
[_tencentOAuth sendGuildJoinGroup:info callback:^(TCGuildRspJoinGroupInfo *info) {
//处理回调结果
if (info.retCode == GuildErrNone)
// success
else {
if (info.retCode == GuildErrLocalApi) {
// 处理info.apiSendResultCode
}
}
}];
```

其中， _tencentOAuth 为 TencentOAuth 类的实例。

查询用户是否加入公会

```
TCGuildQueryMemberRelationDic *info = [TCGuildQueryMemberRelationDic dictionary];
info.paramZoneId = @"分区id";
info.paramGuildId = @"公会id";
info.paramRoleId = @"角色id";
[_tencentOAuth sendGuildQueryMemberRelation:info callback:^(TCGuildRspQueryMemberRelationInfo *info){

}];
```

其中， _tencentOAuth 为 TencentOAuth 类的实例。

游戏社区

打开兴趣部落:

```
[TencentOAuth launchQQBuluoWithBid:@"15558"];
```

其中， 15558 为部落 bid，游戏接入方需要自行提供。

处理 QQ 业务的回调

最近更新时间：2017-11-02 15:51:45

在使用 QQApiInterface 的方法时需要设置回调才能正确调用。设置方法如下：

```
- (BOOL)application:(UIApplication *)application handleOpenURL:(NSURL *)url {
[QQApiInterface handleOpenURL:url delegate:qqApiDelegate];
return YES;
}
```

在 handleOpenURL 中添加 `QQApiInterface handleOpenURL:url delegate: qqApiDelegate` 代码，可以在 QQAPIDemoEntry 类中实现 QQApiInterfaceDelegate 的回调方法。

> **注意：**
> 添加关于 onReq 和 onResp 的说明，同时微信与 QQ 回调相同，提醒开发者注意采用不同的代理处理回调。

ChangeLog

最近更新时间：2017-11-02 15:52:31

Version 1.0.0：

- 增加游戏公会绑定 QQ 群接口。
- 增加查询公会是否绑定 QQ 群接口。
- 增加解绑公会接口。
- 增加游戏内一键加群功能。
- 增加查询用户是否加入公会接口。

OpenSDK中转CGI接口（后台接口）

最近更新时间：2018-01-15 14:35:42

公共接口说明

CGI 调用方法

- 方法1： http://proxy.vip.qq.com/cgi-bin/QQConnect.fcgi
- 方法2： https://proxy.vip.qq.com/cgi-bin/QQConnect.fcgi

公共请求参数

参数	是否必须	含义
oauth_consumer_key	是	申请 QQ 登录成功后，分配给应用的appid
openid	是	用户的 ID，与 QQ 号码一一对应。 可通过调用 https://graph.qq.com/oauth2.0/me?access_token=YOUR_ACCESS_TOKEN 来获取。
access_token	是	可通过使用 Authorization_Code 来获取。 access_token有 3 个月有效期。
cmd	是	cgi名称，此字段决定具体的接口功能
sign	是	签名
format	否	控制 cgi 的输出格式。 目前可选值为 json、jsonp，如果不传则默认为 json
callback	否	当 format 为 jsonp 时，可通过 callback 参数指定返回时的字符串,如果不传则默认为"_Callback"

注意：
所有参数都是放在 QUREY_STRING 中（即 url 的 ？后以 & 作为分隔符），不支持 POST 调用。

公共返回参数

参数	是否必须	含义
ret	是	返回值，0 表示成功，非 0 表示错误码
msg	是	错误描述
data	是	返回的数据。不同的 cgi，返回不同的格式

签名算法

签名值 sign 是将请求源串以及密钥根据一定签名方法生成的签名值，用来提高传输过程参数的防篡改性。 签名值的生成共有 3 个步骤：构造源请求串、构造待加密串和生成签名值。

使用示例

URL： http://proxy.vip.qq.com/cgi-bin/QQConnect.fcgi

请求参数：

```
oauth_consumer_key=100330589
access_token=1821FD577DD45526CA6EA2ADAC685508
openid=29439511BB5D23F36BD09F5B0DF94181
cmd=qc_set_summary_card
card_id=11746
style_id=1
version=7.2.5
platform=2
```

密钥：123456

请求串参数排序

将上面的请求，按照参数名升序排序之后拼接，格式：参数 1 名称参数 1 值参数 2 名称参数2值，示例如下：

```
$access_token$1821FD577DD45526CA6EA2ADAC685508$card_id$11746$cmd$qc_set_summary_card$oauth_consumer_key$100330589$openid$29439511BB5D23F36BD09F5B0DF94181$platform$2$style_id$1$version$7.2.5
```

构造待加密串

将上一步中准备好的参数和密钥拼接起来得到待加密串，格式：待加密串=密钥&参数串&密钥：

```
123456$access_token$1821FD577DD45526CA6EA2ADAC685508$card_id$11746$cmd$qc_set_summary_card$oauth_consumer_key$100330589$openid$29439511BB5D23F36BD09F5B0DF94181$platform$2$style_id$1$version$7.2.5123456
```

生成签名值

将上一步中拼接之后的待加密串进行md5编码（小写 16 进制字符串）

```
sign=tolower(md5("123456$access_token$1821FD577DD45526CA6EA2ADAC685508$card_id$11746$cmd$qc_set_summary_card$oauth_consumer_key$100330589$openid$29439511BB5D23F36BD09F5B0DF94181$platform$2$style_id$1$version$7.2.5123456"))
```

得到的签名值结果如下：
a61529eb51852c7bb6fe396bd62dadcd

最终得到 HTTP 请求串

```
http://proxy.vip.qq.com/cgi-bin/QQConnect.fcgi?oauth_consumer_key=100330589&access_token=1821FD577DD45526CA6EA2ADAC685508&openid=29439511BB5D23F36BD09F5B0DF94181&cmd=qc_set_summary_card&sign=a61529eb51852c7bb6fe396bd62dadcd&card_id=11746&style_id=1&version=7.2.5&platform=2
```

CGI 接口汇总

功能描述

游戏方按照约定上报特定意义的流水。

接入流程

1.游戏方与 opensdk 侧约定上报的字段，需要以下数据，经由 opensdk 确认登记后，才可以进行上报。

- 游戏 appid。
- 上报字段，具体见下面"type字段说明"。
- 报的请求量和请求峰值的评估。

2.游戏方有资源推广等导致上报量大幅上涨的情况，需要提前周知 opensdk 进行评估和扩容。
3.如需接入关怀，需要额外进行以下步骤。

- 需要游戏方提前申请好公众号（必须是服务号）。
- 向 opensNNdk 侧提供申请好的公众号，opensdk 侧添加相应的消息模板和权限。
- 提供需要发送关怀消息的场景，opensdk 侧检查场景是否可以实现，并负责录入场景规则。

CGI 名字

`gc_achieve_report`

调用方式

`POST`

输入参数

输入参数是一个 json 结构。

参数	是否必须	含义
appid	是	游戏 appid
param	是	是一个 json 字符串, 结构见下面说明

```
http://proxy.vip.qq.com/cgi-bin/QQConnect.fcgi?oauth_consumer_key=100330589&access_token=1821FD577DD45526CA6EA2ADAC685508&openid=29439511BB5D23F36BD09F5B0DF94181&cmd=gc_achieve_report&sign=ca3ba525d78fe5c883833dc8e8119021&param={"total":2,"list":[{"type":12,"data":"1","bcover":0,"expires":0},{"type":26,"data":"123","bcover":0,"expires":0},{"type":27,"data":"456","bcover":0,"expires":0},{"type":28,"data":"476","bcover":1,"expires":1510109015}]}
```

json 结构

参数	是否必须	类型	含义
total	是	number	数量，list 数组元素个数
List	是	vector	流水内容
type	是	number	流水类型，由opensdk统一分配，详细说明见下表
data	是	String	流水值
bcover	是	number	是否覆盖， 0：不覆盖, 1：覆盖

参数	是否必须	类型	含义
expires	是	number	0:不过期, 非 0:过期的时间点（unix 时间戳）

type 字段说明（必填 type）

type12、26、27、28 是必填的字段，所有流水上报必须夹带上报。

type	bcover	expire	描述
12	bcover=1	expire=0	平台类型:iOS(0)，安卓(1)
26	bcover=1	expire=0	大区信息:手 Q(1)，微信(2)
27	bcover=1	expire=0	区服 ID
28	bcover=1	expire=0	角色 ID

type 字段说明（常用 type）

常用 type 是选取的比较公用的字段，游戏方可以选择性上报，如果有其他字段，需要约定 type 值后进行上报。

常用 type 是选取的比较公用的字段，游戏方可以选择性上报，如果有其他字段，需要约定 type 值后进行上报。

type	bcover	expire	描述	备注
1	bcover=1	expire=0	等级	
2	bcover=1	expire=0	金钱（钻石、金币、点卷）	
3	bcover=1	与游戏结算时间一致	流水得分	
8	bcover=1	expire=0	角色 ID	
17	bcover=1	expire=0	战力	变化时上报
25	bcover=1	expire=0	用户注册时间	
29	bcover=1	expire=0	角色名称	创建角色时上报，且每条流水上报时同时上报此项
43	bcover=1	expire=0	累积充值金额	累计充值金额变动时进行上报
44	bcover=1	expire=0	单笔充值金额	有充值发生时进行上报

type	bcover	expire	描述	备注
45	bcover=1	expire=0	游戏内玩家VIP等级	变化时上报、登出上报
46	bcover=1	expire=0	充值时间	有充值发生时进行上报
46	bcover=1	expire=0	充值时间	有充值发生时进行上报
6000	bcover=1	expire=0	当天累计游戏时长	上报格式：unix 时间戳（单位必须为：秒）周期上报，每 5 分钟上报一次，上报用户当日的累计游戏时长。（例：用户前 5 分钟在线 1 分钟，上报一次 1 分钟；下一个5分钟在线 2 分钟，则上报一次 3 分钟。）

输出示例

```
{
"ret":0, //中转cgi返回码
"msg":"ok", //中转cgi返回信息
"data":
    {
"ret":0, //0-成功，其他-失败
"msg":"xx", //错误描述
    }
}
```

WPA 临时会话

最近更新时间：2018-01-15 14:49:44

iOS SDK 支持发起 QQ 临时会话，获取指定 QQ 帐号在线状态。使用 WPA 功能需要设置 QQ 业务回调，请参考 处理 QQ 业务的回调。

发起 QQ 临时会话

下面是向指定 QQ 号码发起临时会话的示例代码：

```
(void)onOpenWPA:(QElement *)sender {
    [self.view endEditing:YES];
    [self.root fetchValueUsingBindingsIntoObject:self];
    QQApiWPAObject *wpaObj = [QQApiWPAObject objectWithUin:self.binding_uin];
    SendMessageToQQReq *req = [SendMessageToQQReq reqWithContent:wpaObj];
    QQApiSendResultCode sent = [QQApiInterface sendReq:req];
    [self handleSendResult:sent];
}
```

获取指定 QQ 号码的在线状态

```
(void)getQQUinOnlineStatues:(QElement *)sender {
    [self.view endEditing:YES];
    [self.root fetchValueUsingBindingsIntoObject:self];
    NSArray *ARR = [NSArray arrayWithObjects:self.binding_uin, nil];
    [QQApiInterface getQQUinOnlineStatues:ARR delegate:self];
}
```

会员接口（后台接口）

最近更新时间：2018-01-18 19:56:42

会员包月活动

适用场景

用户在游戏 APP 内开通会员包月服务，可获得专属礼包，同时刺激双方用户相互活跃。

接入方式

步骤 1: 业务方将游戏互联 APPID，游戏名称，管理员 QQ 和回调接口统一提交给 QQ 会员侧，由会员侧提供回调密钥和 aid。
步骤 2: 业务方在游戏 APP 内根据 URL 参数规则拼凑 URL，通过 openUrl 接口打开内部浏览器跳转到支付页面。
其中 URL 为 `https://mq.vip.qq.com/m/connect/open` ，参数以 GET 参数形式传入。
`https://mq.vip.qq.com/m/connect/open?code=svip&month=1&aid=mvip.p.hz.demo`

URL参数说明：

参数名	是否必填	说明	备注
code	是	小写的业务代码	svip：超级会员 vip： QQ 会员
month	是	开通月份	整数
aid	是	业务渠道	由QQ会员侧分配，必须保证准确无误。

Android-游戏开放 SDK 调用样例：

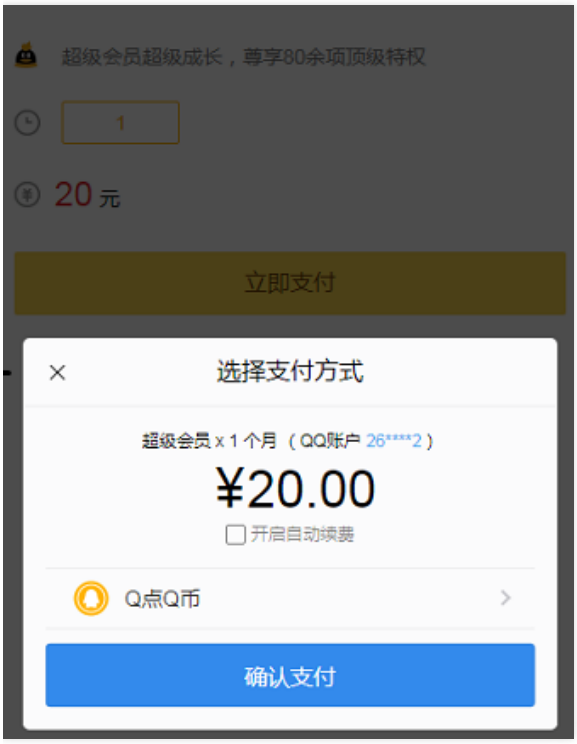
`openUrl(Context context, String url, int screenOrientation)` 接口通过传入 url 和屏幕方向，打开相应页面。

```
WebViewHelper.getInstance().setTencent(mTencent).openUrl(v.getContext(), "https://mq.vip.qq.com/m/connect/open?code=svip&month=1&aid=mvip.p.hz.demo", ActivityInfo.SCREEN_ORIENTATION_SENSOR);
```

效果展示

展示界面由业务方设计开发。

单击【开通超级会员】按钮弹出内部浏览器。



开通流水接入指引

管理端使用

对外开发的管理端可以提供给外部录入用户配置信息和开通 aid（开通来源标记，用户录入信息时由系统自动生成）标记信息。
其中用户配置信息标识游戏情况，开通 aid 配置信息标识该游戏在不同场景下的开通情况。

配置信息录入

用户配置信息如下：

名称	备注
appid	每款游戏的标识(用户申请，必填)
url	开通流水接收地址(比如一个 cgi 地址，必填)
userinfo	申请人的联系方式(例如 QQ 号码、手机号码等，必填)
appkey	转发流水过程中校验 key(与 appid 对应，接收方根据该 key 校验数据，系统生成)
desc	描述信息(例如游戏类别)

注意：
用户配置信息中，appid 为主键，每款游戏有一个对应的 appid；

针对新添加的 appid，系统负责生成 appkey，在以后信息修改的过程中不会修改 appkey；开通流水转发过程中使用 appkey+data 进行 md5 加密，流水接收方根据 data 和 appkey 校验数据。

用户配置信息标识了一款游戏的情况，根据用户开通情况以及游戏运营活动种类的不同，可以为该款游戏配置多种情况下的开通 aid 信息，aid 配置信息表如下：

名称	备注
aid	款游戏的标识(用户申请，系统生成)
appid	开通流水接收地址(比如一个 cgi 地址，必填)
openmonth	该 aid 对应的开通月份(必填)
desc	描述信息(例如开通页面入口等)

注意：
aid 配置信息表中 aid 为主键，每款游戏可以有多个 aid。

数据的录入默认都会提交到测试环境，如果配置信息需要发布现网，可以提交审核，审核通过 10 分钟内流水转发生效。

配置信息查询

- 用户配置信息的查询支持 appid、userinfo 字段的精确查询 。
- 开通 aid 信息的查询支持 aid 的模糊查询、appid 的精确查询。

开通流水转发

开通方式的接入可以参考 会员包月活动 接入文档。

开通流水的转发流程：

1. 根据用户开通流水中的 aid 判断是否属于合作方的开通。
2. 根据该 aid 在配置表中的转发地址转发开通数据，通过 HTTP/HTTPS 请求，报体数据格式为 json。

流水转发请求

流水转发使用 HTTP/HTTPS 请求转发的方式，其中 url 为管理端的用户配置信息内容。其他字段和含义下面举例说明：
某款游戏申请到的 appid：1111
流水接收地址为：`http://10.100.70.5/cgi-bin/srfentry.fcgi?`
数据校验 appkey：ffff
该款游戏下合法的 aid 为：aaaa

那么会员侧在接收到用户开通之后，根据用户的配置信息，流水转发的 url 为（data 为 urlencode）：

```
http://proxyvac.qq.com/cgi-bin/srfentry.fcgi?ts=1508210911065&data={"aid":"aaaaa","msg_time":"1507799551","open_months":"1","open_type":"vip","openid":"xjfdaldfjrefjfljljgfjg","order_id":"20171012-1712128958"}&sign=fefaf9ddd9d4dce23da3b664ec457413
```

请求参数

ulr 中各个字段的说明如下：

http://proxyvac.qq.com/cgi-bin/srfentry.fcgi //用户在配置表中的转发url
Ts=1508210911065 //流水开始转发的时间戳(ms)

data域为json格式数据(每个字段值都为string)

```
{
"aid": "aaaaa",
"msg_time": "1507799551",
"open_months": "1",
"open_type": "vip",
"openid": "xjfdaldfjrefjfljljgfjg",
"order_id": "20171012-1712128958"
}
sign=fefaf9ddd9d4dce23da3b664ec457413 //data域数据(在这里是上面json数据)appkey组成字符串后md5值，md5(data+appkey)
```

请求 data 域字段说明

字段名	类型	说明
aid	String	开通 aid
msg_time	String	开通时间戳
open_months	String	开通 aid
msg_time	String	开通时长(单位为月)
open_type	String	开通类型(vip：会员，svip：超级会员)
openid	String	用户 openid
order_id	String	开通订单 id

注意：
openid 和 order_id 可以唯一确定用户的开通记录，接收方需要处理流水去重。

流水转发回应

对应开通流水的转发结果，需要返回接收结果情况。

返回参数

返回包体约定为 json 数据格式，包含的字段及其含义如下所示：

```
{
"ret": 0,
"msg": "succ"
}
```

返回字段说明

字段名	类型	说明
ret	int	返回值，0 为正常
msg	String	返回信息

说明：
流水转发统一使用 get 请求，根据用户表中转发地址的配置，支持 HTTP 和 HTTPS 两种方式。

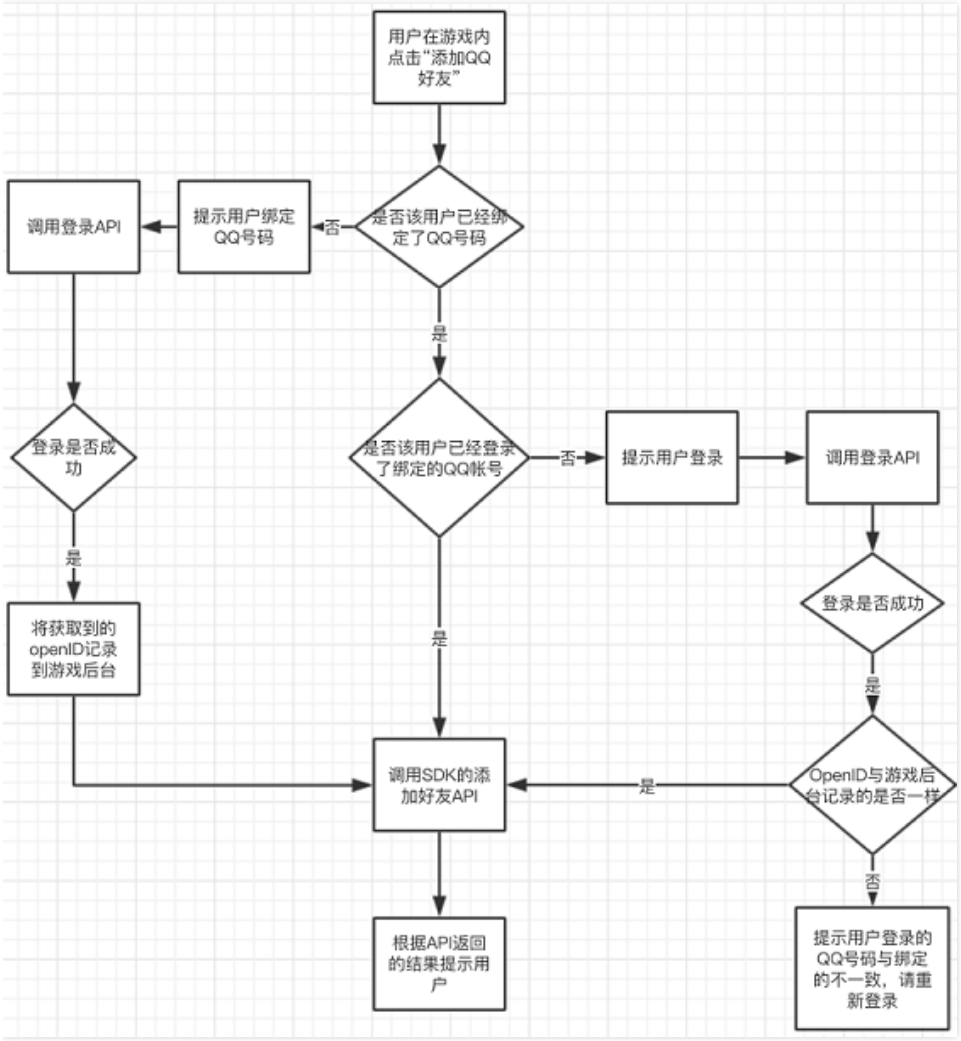
无登录状态使用说明

最近更新时间：2018-01-15 14:40:04

游戏 SDK 支持无登录态调用 API，游戏接入方无须使用手机 QQ 登录游戏，仅需要在调用某些需要登录态的 API 时候进行一次 QQ 帐号和游戏账号的绑定即可。

目前强制需要登录之后才能调用的 API 包括：游戏加群、绑群、解散群功能、创建 QZone相册、获取相册列表、上传图片、发表说说、验证空间粉丝、游戏 SDK 新增 API。

游戏接入方需要在后台数据库记录一个游戏账号绑定的 QQ OpenID 字段，当用户首次使用登录 API 的功能时，提示用户将此 QQ 帐号与当前登录的游戏账号绑定，用户确认之后再调用 SDK 内的登录接口，获取到 OpenID 之后记录到游戏后台。后续如果用户再次使用登录 API 功能，游戏接入方需要首先判断用户是否已经绑定了 QQ 帐号，然后进一步判断绑定的 QQ 帐号是否已经登录（通过是否 TencentOAuth 的 accessToken 值是否为空来判断）。具体的流程如下（简化版）：



游戏 SDK 功能

最近更新时间：2018-01-15 14:41:08

游戏 SDK 增加了 3 个新的 API，分别为：添加表情包到 QQ、设置资料卡名片以及样式，和发送公众号消息给指定 QQ 好友。

添加表情包到 QQ

下面是添加表情包到 QQ 的示例代码：

```
(void)onAddQQEmoji:(UIButton *)sender {
NSString *emojiID = @" 1234" ;
[_oauth addQQEmoji:emojiID];
}
```

这里的 emojiID 是表情包的 ID，详情请咨询游戏 SDK 的开发。 调用此接口后，TencentOAuth 对象的 delegate 会收到一个回调：

```
(void)tencentOAuth:(TencentOAuth *)tencentOAuth didCompleteAddQQEmojiWithError:(NSError *)error serverResponse:(NSDictionary * _Nullable)response
```

如果请求出现错误，这里的 error 对象描述了错误原因以及错误码等信息。
response 表示服务器返回的数据，如果是因为网络原因或者参数错误导致请求没有发出，则 response 可能为空。response 的示例如下：

```
{
"ret":0, //中转cgi返回码
"msg":"ok", //中转cgi返回信息
"data":
{
"ret":0, //0-成功，其他-失败
"msg":"xx", //错误描述，活动类型表情鉴权失败时，该值表示活动描述
"data":{"url":""}
}
}
```

当出现错误时，调用方可以参考 response 中的 msg 以及 ret 字段来检查错误原因。

设置资料卡名片和样式

下面是设置手机 QQ 资料卡的名片和样式的示例代码：

```
(void)onSetSummaryCard:( UIButton *)sender {
    NSString *cardID = @" 1234" ;
    NSString *styleID = @" 2414" ;
[_oauth setQQSummaryCardID:cardID styleID:styleID];
}
```

调用接口之后，TencentOAuth 对象的 delegate 会收到一个回调：

```
(void)tencentOAuth:(TencentOAuth *)tencentOAuth didCompleteSetQQSummaryCardWithError:(NSError *)error serverResponse:(NSDictionary * _Nullable)response
```

发送公众号消息给 QQ 好友

下面是发送公众号消息给指定好友的示例代码：

```
(void)onSendPublicAccountMessage:( UIButton *)sender {
    NSString * openIDS = @"  [{\"type\":0,\"openid\":\"962f1d0f4cfcdc2981c51bcc0a630057\"}]";
    NSString * openIDS = @"  MSG_SHARE_FRIEND_DOUBLE";
    NSString * previewURL = [NSURL URLWithString:@"  https://egame.gtimg.cn/club/pgg/v2.0/img/index/icon.png"];
    NSString * redirectingURL = [NSURL URLWithString: @"  https://egame.qq.com/"];
     [_oauth sendPublicAccountMessageToQQOpenIDs:openIDS withMessageTag:messageTag WithPreviewImageURL:previewURL andRedirectingURL:redirectingURL];
}
```

调用接口之后，TencentOAuth 对象的 delegate 会收到一个回调：

```
(void)tencentOAuth:(TencentOAuth *)tencentOAuth didCompleteSendPublicAccountMessageWithError:(NSError *)error serverResponse:(NSDictionary * _Nullable)response
```

注意：
调用此接口必须确保对方已经是自己的好友，而且关注了公众号，否则对方会收不到消息。

社交分享消息

最近更新时间：2019-03-06 20:01:30

功能描述

通过 GET 调用方式，实现点对点定向分享(分享消息给手机 QQ 好友，在公众账号中显示)。

CGI 名字

`send_share_for_inner`

输入参数

参数	是否必须	含义
fopenids	是	接收方信息
imageUrl	是	消息缩略图
targetUrl	是	跳转链接
game_tag	是	消息类型： MSG_INVITE(邀请) MSG_FRIEND_EXCEED(超越炫耀) MSG_HEART_SEND(送心) MSG_SHARE_FRIEND_PVP(PVP 对战)
dst	是	msf-手q(包括 iPhone, Android QQ 等)，目前只能填 1001
flag	是	漫游 (0：是；1：否，目前只能填 1)

实际示例

输入示例

```
http://proxy.vip.qq.com/cgi-bin/QQConnect.fcgi?cmd=send_share_for_inner&access_token=8d340568 2be0dd70510325272b495c5e&fopenids=[{"openid":"1983e77ac7517e997967168816c41867","type":0}]&imageUrl=http://imgcache.qq.com/club/gamecenter/cms/2016-04-01/1459504923_868_750x1182.jpg&targetUrl=http://gamecenter.qq.com/gcjump?appid=1104466820&oauth_consumer_key=100330589&openid=1983e77ac7517e997967168816c41867&dst=1001&flag=1&game_tag=MSG_SHARE_FRIEND_DOUBLE&sign=86ad69be5551c514b39693fc436c89e4
```

输出示例

```
{
"ret": 0, //中转cgi返回码
```

```
"msg": "ok", //中转cgi返回信息
"data": {
"ret": 0, //0-成功，其他-失败
"msg": "xx"    //错误描述
}
}
```

错误码

最近更新时间：2018-12-24 10:06:42

错误码	说明	处理措施
-1	client 的请求参数无效	检查请求参数准确性
-2	请求中的 APPID 不存在	请确认 APPID、appkey 的准确性，提交工单 联系客服处理
-3	client 请求中 App 到 API 访问无权限	请确认该 APPID 是否有权限，如有权限请 提交工单 联系客服处理
-4	请求中的 App IP 不允许	若失败请 提交工单 联系客服处理
-5	签名验证失败	重试一次，若失败请 提交工单 联系客服处理
-6	client 请求中 App 到 API 访问超限	请降低访问频率，检查是否正常请求，提交工单 联系客服处理
-7	请求协议非法，例如 HTTPS 搞成了 HTTP	请确认访问协议是否正确，提交工单 联系客服处理
-8	请求受限，通常是安全审计没通过	请 提交工单 联系客服处理
-9	API 不存在	请 提交工单 联系客服处理
-10	请求中的 App 内网 IP 不允许	请 提交工单 联系客服处理
-11	请求中的 App 外网 IP 不允许	请 提交工单 联系客服处理
-12	测试环境调试号码受限	请使用有权限的号码
-20	client 请求中 API 未经用户授权	请重新授权
-21	access_token 已废除	请重新授权
-22	openid 非法	请确认 APPID、openid 的准确性， 提交工单 联系客服处理
-23	openkey 非法	请重新授权
-24	openid openkey 验证失败	请重新授权
-25	0x71f 0x5b：timestamp 与系统当前时间相差超过10分钟	请确认 APPID、appkey、openkey 的一致性， 提交工单 联系客服处理
-26	0x71f 0x5a：重复的 nonce	openid 转换时出错，请确认 APPID、openid 准确性
-70	登录验证返回，验证 openkey 时 appid 非法	请确认 APPID、appkey、openkey 的一致性， 提交工单 联系客服处理
-71	openid 和 openkey 不匹配	确认 openid 与 openkey 的一致性
-72	appkey 和权限 tmem 中的 appkey 不一致	请确认 APPID、appkey 的准确性， 提交工单 联系客服处理
-73	0x47 access token 改密失效	请重新授权
100000	缺少或错误 response_type	请检查请求参数

错误码	说明	处理措施
100001	缺少参数 client_id	请检查请求参数
100002	缺少参数 client_secret	请检查请求参数
100003	HTTP head 中缺少 Authorization	请检查请求参数
100004	缺少或错误 grant_type	请检查请求参数
100005	缺少 code 参数	请检查请求参数
100006	缺少 refresh token	请检查请求参数
100007	缺少 access token	请检查请求参数
100008	该 APPID 不存在	请确认 APPID 的准确性， 提交工单 联系客服处理
100009	appkey（client_secret）不合法	请确认 APPID、appkey 的准确性， 提交工单 联系客服处理
100010	回调地址不合法	回调地址不合法，常见原因请查阅 腾讯开发平台
100011	App 不处于上线状态	请 提交工单 联系客服处理
100012	非 post 方式	请 提交工单 联系客服处理
100013	access token 不合法	需重新授权
100014	access token 过期	需重新授权
100015	access token 废除	需重新授权
100016	access token 验证失败，其它原因	需重新授权
251001	参数非法	按照参数列表检查参数是否正确
251002	access token 不合法	需重新授权
251003	获取好友 openid 失败	接口调用无权限，请申请权限
251004	获取好友缓存失败	请 提交工单 联系客服处理
251005	更新好友缓存失败	请 提交工单 联系客服处理