腾讯社区开放平台.NET SDK

“QQ登录”使用户能使用QQ账号一键登录接入网站，大大降低了注册、登录的门槛。借助庞大的QQ用户群，给第三方网站带来更多新用户。已登录用户还可以将在第三方网站发布、分享的信息即时同步到QQ空间，使网站内容通过好友关系得到进一步的传播。

腾讯社区开放平台.NET SDK是基于腾讯社区开放平台的[OAuth登录文档](http://wiki.opensns.qq.com/wiki/%E3%80%90QQ%E7%99%BB%E5%BD%95%E3%80%91OAuth%E7%99%BB%E5%BD%95%E6%96%87%E6%A1%A3)和[API文档](http://wiki.opensns.qq.com/wiki/%E3%80%90QQ%E7%99%BB%E5%BD%95%E3%80%91API%E6%96%87%E6%A1%A3) 使用C#开发的.NET 4.0的一个库，目前完整实现了腾讯社区开放平台的所有接口。可以加快你的社区接入到腾讯社区开放平台，免去大量的重复的调试开发的时间，所以我是以New BSD协议创建的开源项目，各位有同样需求的各位同仁参与建设着一个项目，项目的托管地址是<http://qqconnect.codeplex.com/>。

腾讯社区开放平台.NET SDK使得你调用腾讯社区开放平台的REST接口非常容易，你只需要到<http://connect.opensns.qq.com/apply> 申请到必须的consumer key和consumer secret，在你的申请通过了腾讯社区开放平台的审核后就可以得到。这个酷的主要特性如下：

* 封装好了腾讯社区开放平台的OAuth协议访问腾讯社区开放平台的REST接口
* 和腾讯社区开放平台的接口保持同步（目前只有一个接口，将来肯定会不断增加）

这个库使用Visual Studio 2010和C# 3.0构建的，主要扩展和使用到下面的开源库：

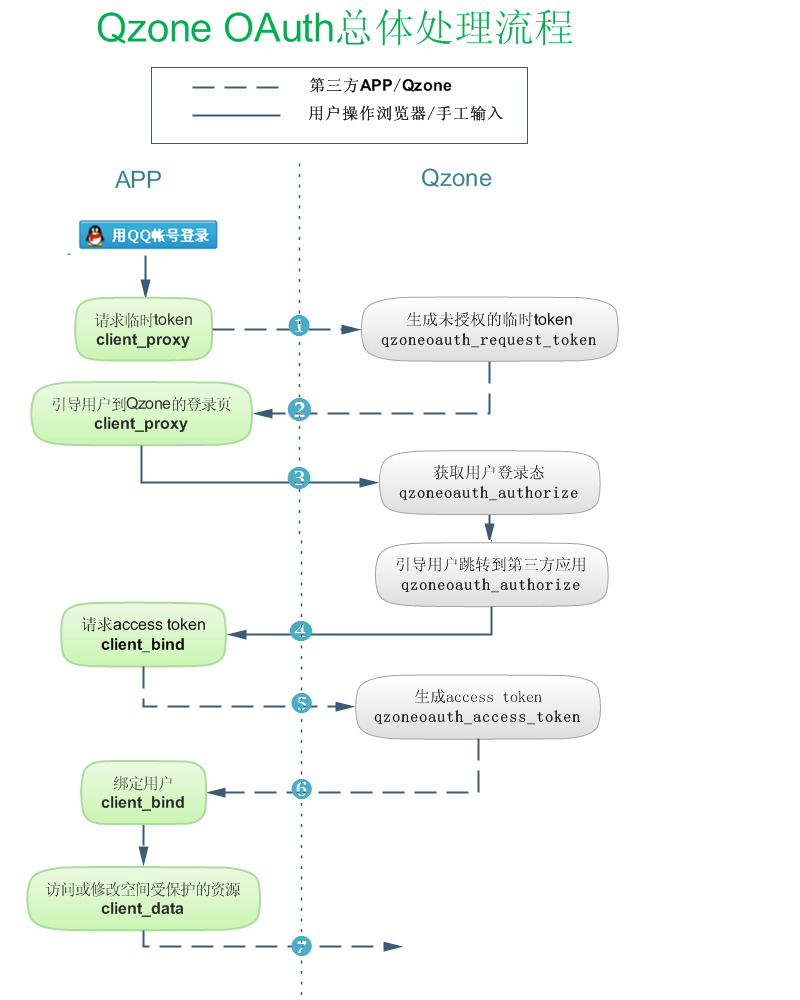
1. [DotnetOpenauth](http://www.dotnetopenauth.net/)：DotNetOpenAuth 是一个 C# 开发包，用以实现 OpenID 2.0 的提供者、中继方（Relying Party）、OAuth使用者和OAuth服务提供者；同时支持 InfoCard Selector 。OAuth协议客户端的实现上参考了这个项目的代码。

1. [Jayrock](http://jayrock.berlios.de/)：Jayrock是一个LGPL的开源的软件，实现了JSON和JSON-RPC，支持微软ASP.NET框架。具体可以参考文章：[Jayrock: JSON and JSON-RPC for .NET](http://www.cnblogs.com/shanyou/archive/2007/07/01/802213.html)

上面我们从高层次了解了这个SDK的实现原理，下面我们介绍如何使用这个SDK来和腾讯社区开放平台进行交互。SDK主要有两个需要关注的类QzoneSDK.Context.QzoneContext

和 QzoneSDK.Qzone。

首先我们结合[【QQ登录】OAuth登录文档](http://wiki.opensns.qq.com/wiki/%E3%80%90QQ%E7%99%BB%E5%BD%95%E3%80%91OAuth%E7%99%BB%E5%BD%95%E6%96%87%E6%A1%A3#1._QQ.E7.A9.BA.E9.97.B4OAuth.E7.AE.80.E4.BB.8B)来说明。Qzone OAuth总体处理流程里共有七个步骤，从下面的总体流程图中可以看到：



左侧：为第三方网站需要做的事情。在每个绿色的方框里，标明了第三方应该调用的接口，接口主要功能及逻辑见Qzone OAuth接口说明。

右侧：为Qzone接收到第三方应用发过来的请求后做的事情。第三方请求的发送地址以及应包含的参数在接下来的单步详解中进行了说明

通过类QzoneSDK.Context.QzoneContext带上你申请的oauth\_consumer\_key （对应Qzone的appid）和oauth\_consumer\_secret （对应Qzone的appkey）到腾讯社区开放平台获取到Request Token，然后把tokenKey和tokensecret保存到Session里，后续需要使用。示例代码如下：

string key = ConfigurationManager.AppSettings["ConsumerKey"];

string secret = ConfigurationManager.AppSettings["ConsumerSecret"];

var context = new QzoneSDK.Context.QzoneContext(key, secret);

var callbackUrl = "http://localhost/account/QQCallback";

var requestToken = context.GetRequestToken(callbackUrl);

Session["requesttokenkey"] = requestToken.TokenKey;

Session["requesttokensecret"] = requestToken.TokenSecret;

var authenticationUrl = context.GetAuthorizationUrl(requestToken, callbackUrl);

return Redirect(authenticationUrl);

用户在腾讯社区开放平台上授权使用后会导向到你的网站，也就是上图的第三步，在门上述的callback 的页面上就可以获取上图第四步的参数，利用上述获得的参数调用QzoneSDK.Qzone进行第五步调用，就可以获得AccessToken，这个AccessToken，目前access\_token(及其secret)是长期有效的，和某一个openid对应，目前可以支持线下获取该openid的信息。

if (Request.QueryString["oauth\_vericode"] != null)

{

var requestTokenKey = Session["requesttokenkey"].ToString();

var requestTokenSecret = Session["requesttokensecret"].ToString();

var verifier = Request.QueryString["oauth\_vericode"];

string key = ConfigurationManager.AppSettings["ConsumerKey"];

string secret = ConfigurationManager.AppSettings["ConsumerSecret"];

QzoneSDK.Qzone qzone = new QzoneSDK.Qzone(key, secret, requestTokenKey, requestTokenSecret, verifier);

var access = qzone.OAuthTokenKey;

var currentUser = qzone.GetCurrentUser();

var user = (BasicProfile)JsonConvert.Import(typeof(BasicProfile),currentUser);

}

上述代码还演示调用腾讯开放社区目前唯一的一个API （ get\_user\_info），通过Jayrock库将Json数据转换为库中定义的用户登录数据的实体BasicProfile。