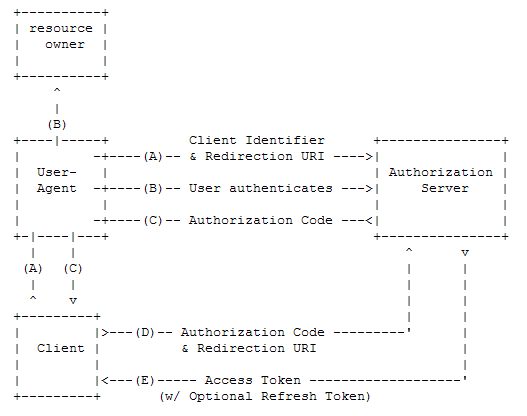
# 1、简介

开放平台使用server-side模式，是[OAuth2.0认证](http://wiki.open.qq.com/wiki/website/OAuth2.0%E7%AE%80%E4%BB%8B" \o "website/OAuth2.0简介)的一种模式，又称Web Server Flow；  
适用于需要从web server访问的应用，例如Web/wap网站。

其授权验证流程示意图如下（图片来源：[OAuth2.0协议草案V21的4.1节](http://tools.ietf.org/html/draft-ietf-oauth-v2-21" \t "_blank" \o "http://tools.ietf.org/html/draft-ietf-oauth-v2-21) ）   


对于应用而言，需要进行三步：

1. 获取Authorization Code；

2. 通过Authorization Code获取Access Token

3. 使用Access Token访问和修改用户数据

# 2、流程说明

## 2.1 用户通过移动学习平台登录第三方网站

以用户通过移动学习平台登录第三方网站为例：

1. 用户在第三方网站请求通过移动学习平台登录
2. 第三方网站引导用户请求移动学习平台的授权接口（同时带上回调URL）

/openapi/oauth/authorize

1. 移动学习平台验证用户登录，用户未登录则跳转登录页面
2. 用户登录成功后进入授权页面
3. 用户同意授权则重定向到客户端的回调URL，并带上授权码（Authorization Code）
4. 第三方网站后台通过授权码（Authorization Code）向移动学习平台获取授权令牌（Access Token）

/openapi/oauth/token

1. 第三方网站保存获取到的用户授权令牌（Access Token）
2. 第三方网站后台根据授权令牌（Access Token）向移动学习平台请求用户信息接口

/openapi/user/get\_user\_info

1. 第三方网站响应用户登录成功



## 2.2 用户从移动学习平台打开第三方网站（内嵌式）

1. 用户在移动学习平台请求打开第三方网站
2. 移动学习平台验证用户登录，用户未登录则跳转登录页面
3. 用户登录成功后进入授权页面（根据用户点击的第三方应用加上client\_id和redirect\_url等参数）

/openapi/oauth/authorize

1. 用户同意授权则重定向到客户端的回调URL，并带上授权码（Authorization Code）
2. 第三方网站后台通过授权码（Authorization Code）向移动学习平台获取授权令牌（Access Token）

/openapi/oauth/token

1. 第三方网站保存获取到的用户授权令牌（Access Token）
2. 第三方网站后台根据授权令牌（Access Token）向移动学习平台请求用户信息接口

/openapi/user/get\_user\_info

1. 第三方网站响应用户登录成功

# 3、过程详解

## Step1：获取Authorization Code

请求地址： /openapi/oauth/authorize

请求方法：GET

请求参数：

请求参数请包含如下内容：

参数 是否必须 含义

response\_type 必须 授权类型，此值固定为“code”。

client\_id 必须 分配给应用/网站的client\_id。

redirect\_uri 必须 成功授权后的回调地址，必须是注册应用/网站时填写的主域名下的地址，建议设置为网站首页或网站的用户中心。注意需要将url进行URLEncode。

state 必须 client端的状态值。用于第三方应用防止CSRF攻击，成功授权后回调时会原样带回。请务必严格按照流程检查用户与state参数状态的绑定。

scope 可选 请求用户授权时向用户显示的可进行授权的列表。

可填写的值是API文档中列出的接口，以及一些动作型的授权，如果要填写多个接口名称，请用逗号隔开。

例如：scope=get\_user\_info

不传则默认请求对接口get\_user\_info进行授权。

建议控制授权项的数量，只传入必要的接口名称，因为授权项越多，用户越可能拒绝进行任何授权。

返回说明：

如果用户成功登录并授权，则会跳转到指定的回调地址，并在redirect\_uri地址后带上Authorization Code和原始的state值。如：

http://www.test.com/my.php?code=MAh5kR\*\*\*\*\*\*&state=test

## Step2：通过Authorization Code获取Access Token

请求地址： /openapi/oauth/ token

请求方法：GET

请求参数：

请求参数请包含如下内容：

参数 是否必须 含义

grant\_type 必须 授权类型，此值固定为“authorization\_code”。

client\_id 必须 分配给应用/网站的client\_id。

client\_secret 必须 分配给应用/网站的client\_secret。

code 必须 上一步返回的authorization code。

如果用户成功登录并授权，则会跳转到指定的回调地址，并在URL中带上Authorization Code。

例如，回调地址为www.test.com/my.php，则跳转到：

http://www.test.com/my.php?code=MAh5kR\*\*\*\*\*\*

redirect\_uri 必须 与上面一步中传入的redirect\_uri保持一致。

返回说明：

如果成功返回，即可在返回包中获取到Access Token。

返回如下JSON串：{"access\_token":"0f1c7fc2-a937-4746-821e-8d59f292072d","token\_type":"bearer","refresh\_token":"8938f2f8-4685-4831-9195-abf1c8a820f2","expires\_in":43199,"scope":"read"}

说明：

expires\_in是该access token的有效期，单位为秒。

## Step3：使用Access Token来访问和修改用户数据

1. 建议网站在用户登录后，即调用get\_user\_info接口，获得该用户的头像、昵称并显示在网站上，使用户体验统一。

2. 调用其他OpenAPI，以访问和修改用户数据。所有OpenAPI详见API列表。

以调用get\_user\_info接口为例：

（1）发送请求到get\_user\_info的URL（请将access\_token，appid等参数值替换为你自己的）：

/openapi/user/get\_user\_info?access\_token=YOUR\_ACCESS\_TOKEN

（2）成功返回后，即可获取到用户数据：

{

"ret": 0,

"msg": "获取用户成功",

"body": {

"user": {

"nickName": "昵称",

…

}

}

}