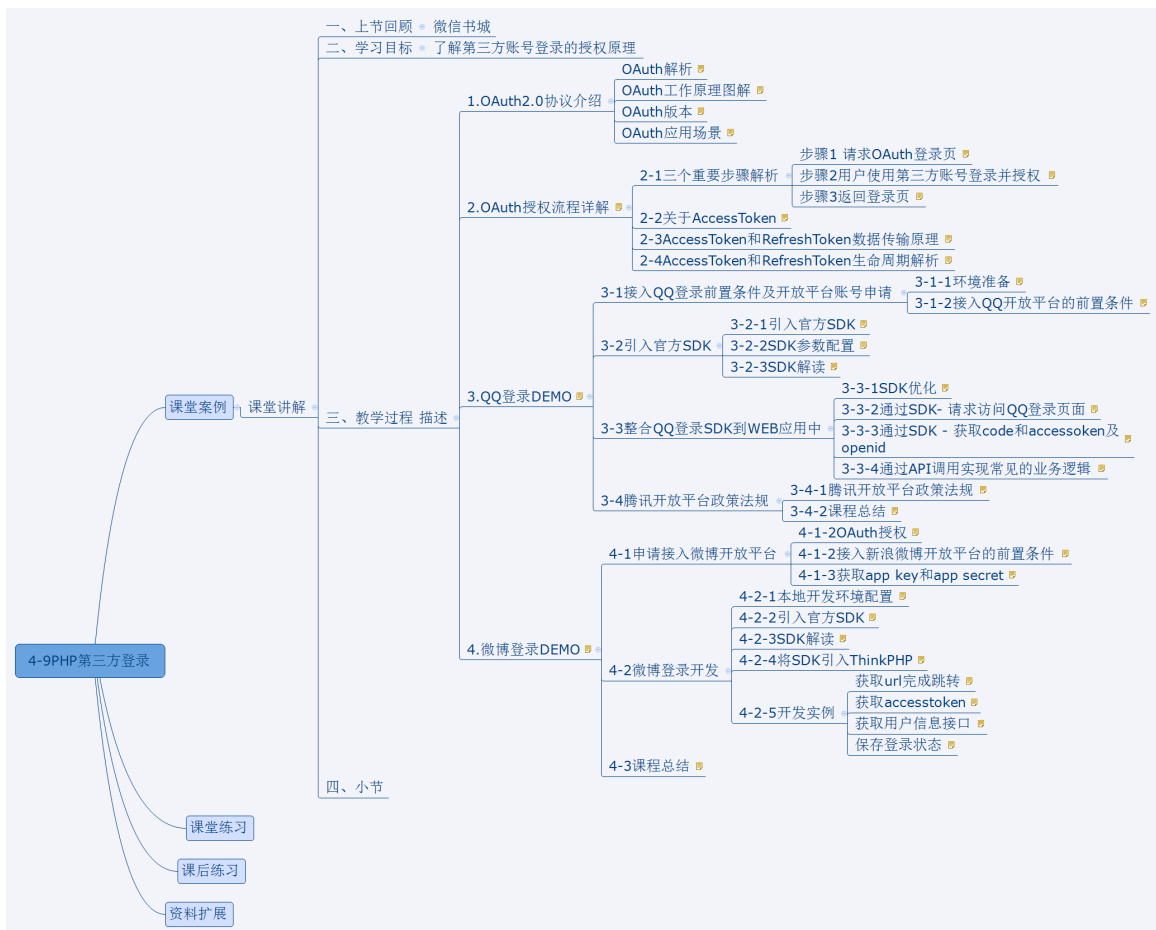


4-9PHP第三方登录

4-9PHP第三方登录	1
1. 课堂案例	2
课堂讲解	2
一、上节回顾	2
微信书城	2
二、学习目标	2
了解第三方账号登录的授权原理	2
三、教学过程 描述	2
1.OAuth2.0协议介绍.....	2
2.OAuth授权流程详解.....	4
3.QQ登录DEMO	7
4.微博登录DEMO.....	22
四、小节	42
2. 课堂练习	42
3. 课后练习	42
4. 资料扩展	42



1. 课堂案例

课堂讲解

一、上节回顾

微信书城

二、学习目标

了解第三方账号登录的授权原理

三、教学过程 描述

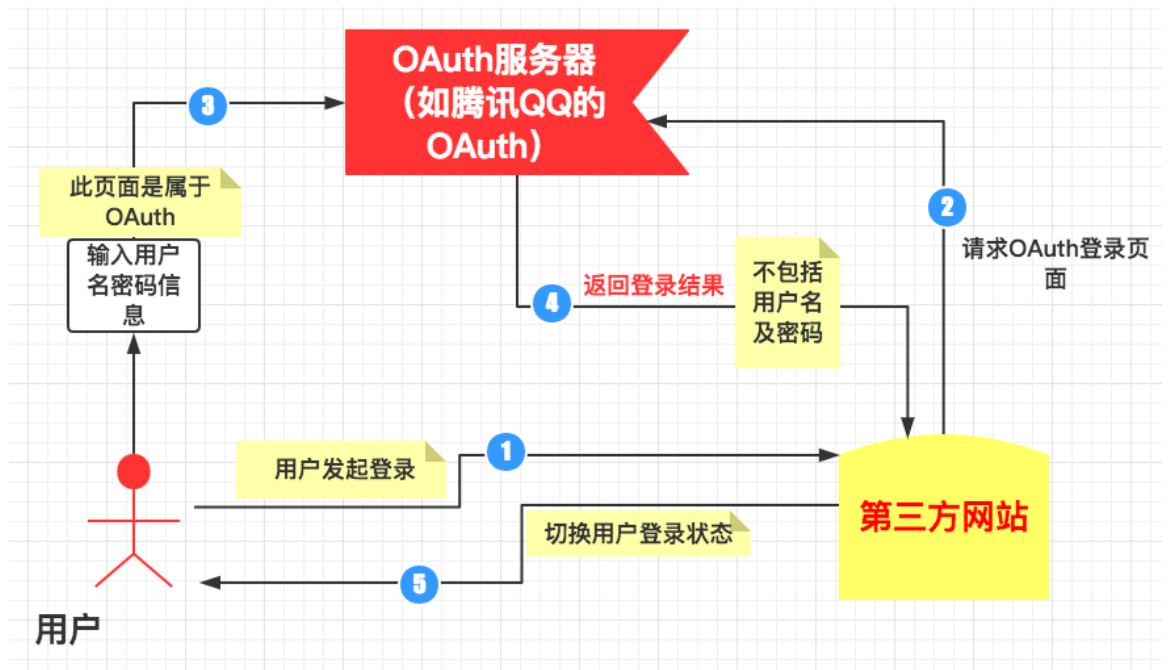
1.OAuth2.0协议介绍

OAuth解析

OAuth 开放授权 (Open Authorization)

是一个开放标准, 允许用户授权第三方网站访问他们存储在另外的服务提供者上的信息, 而不需要将用户名和密码提供给第三方网站或分享他们数据的所有内容

OAuth工作原理图解



OAuth版本

之前是基于OAuth1的版本, 由于之前造成过安全事故, 因此改版后出了一个OAuth2.0版本, 这也是我们讲解的版本

OAuth应用场景

QQ用户授权某网站使用其QQ账号想着的信息

获取授权后, 在符合有权限规则的情况下访问各种API

如, 阿里各平台都可以使用淘宝账号登录, 并且可以支付如分享这些功能都是基于OAuth来实现

分组名称	适用平台	API接口	功能说明	备注
访问用户资料	网站、移动	get_user_info	获取登录用户的昵称、头像、性别	
访问用户QQ会员信息	网站、移动	get_vip_info	获取QQ会员的基本信息	需要申请
		get_vip_rich_info	获取QQ会员的高级信息	
访问我的空间相册	网站、移动	list_album	获取用户QQ空间相册列表	需要申请
		upload_pic	上传一张照片到QQ空间相册	
		add_album	在用户的空间相册里，创建一个新的个人相册	
		list_photo	获取用户QQ空间相册中的照片列表	
访问我的腾讯微博资料	网站、移动	get_info	获取登录用户在腾讯微博详细资料	
分享内容到我的腾讯微博	网站、移动	add_t	发表一条微博	
		del_t	删除一条微博	
		add_pic_t	发表一条带图片的微博	
		get_repost_list	获取单条微博的转发或点评列表	
获得我的微博好友信息	网站、移动	get_other_info	获取他人微博资料	
		get_fanslist	我的微博粉丝列表	
		get_idollist	我的微博偶像列表	
		add_idol	收听某个微博用户	
		del_idol	取消收听某个微博用户	
访问我的财付通信息	网站、移动	get_tenpay_addr	在这个网站上将展现您财付通登记的收货地址	需要申请

2.OAuth授权流程详解

以QQ登录为例

QQ登录OAuth2.0总体处理流程如下：

Step1: 申请接入，获取appid和apikey；

Step2: 开发应用，并设置协作者帐号进行测试联调；

Step3: 放置QQ登录按钮；

Step4: 通过用户登录验证和授权，获取Access Token；

Step5: 通过Access Token获取用户的OpenID；

Step6: 调用OpenAPI，来请求访问或修改用户授权的资源。

2-1三个重要步骤解析

步骤1 请求OAuth登录页

如打开一个授权地址：

`https://graph.qq.com/oauth2.0/authorize?response_type=code&client_id=[YOUR_APPID]&redirect_uri=[YOUR_REDIRECT_URI]&scope=[THE_SCOPE]`



步骤2用户使用第三方账号登录并授权

OAuth服务器内部对登录页提交的参数进行权限，注意授权的API越多，越可能被用户拒绝，

用户同意授权后，即授权成功后



步骤3返回登录页

授权成功后, 回跳转到登录时提交的回调URL ,并且会带上 Authorization Code

如下:

http://www.qq.com/my.php?code=520DD95263C1CFEA0870FBB66E*****

2-2关于AccessToken

AccessToken 即访问令牌

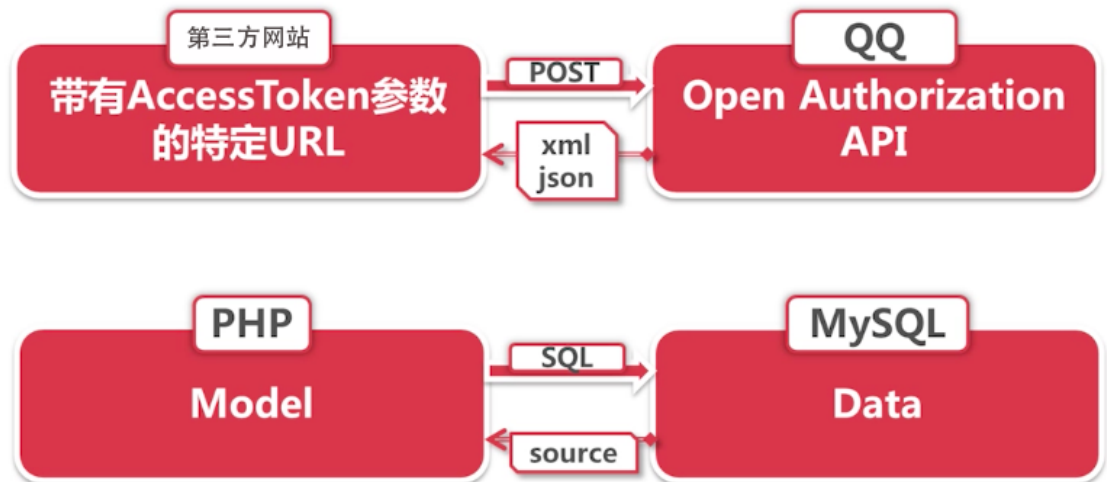
通过上面的三个步骤, 我们获得了Authorization Code ,通过 Authorization Code 可以调用API去获取AccessToken,

比如QQ:

[https://graph.qq.com/oauth2.0/token?grant_type=authorization_code&client_id=\[YOUR_APP_ID\]&client_secret=\[YOUR_APP_Key\]&code=\[The_AUTHORIZATION_CODE\]&state=\[The_CLIENT_STATE\]&redirect_uri=\[YOUR_REDIRECT_URI\]](https://graph.qq.com/oauth2.0/token?grant_type=authorization_code&client_id=[YOUR_APP_ID]&client_secret=[YOUR_APP_Key]&code=[The_AUTHORIZATION_CODE]&state=[The_CLIENT_STATE]&redirect_uri=[YOUR_REDIRECT_URI])

2-3AccessToken和RefreshToken数据传输原理

AccessToken与RefreshToken



2-4AccessToken和RefreshToken生命周期解析

获取到的access token具有3个月有效期, 用户再次登录时自动刷新。

3.QQ登录DEMO

http://wiki.connect.qq.com/%E5%BC%80%E5%8F%91%E6%94%BB%E7%95%A5_server-side

3-1接入QQ登录前置条件及开放平台账号申请

3-1-1环境准备

准备一个可以在线访问的网站, 比如前面我们所开发的网站, 另外此网站的域名必须是已经备案通过 的

3-1-2接入QQ开放平台的前置条件

1. 成为开发者:

<https://connect.qq.com/index.html> 登录后, 注册成为开发

 QQ 互联

首页 应用管理 文档资料

 罗慧华-多迪网络 ▾


资料填写


邮箱验证


完成注册

 罗慧华-多迪网络

☐ 公司接入 ☒ 个人接入


上传身份证照片

下一步

上面的开发者审核通过后, 进行如下第2部

2. 创建应用 要求创建一个WEB应该, 并且当前网站域名需在线且备案通过





罗慧华 (个人开发者)
联系方式: [REDACTED] | 验证邮箱: [REDACTED]
审核已通过

网站应用

移动应用

创建应用

应用名称	APPID	创建时间	审核状态	操作
 第三登陆测...	101408664	2017-06-12	审核中	--



创建网站应用



创建iOS应用



创建Android应用



资料填写



完善资料



完成注册

请选择



请选择



网站名称

网站简介，200字以内

☐ 我已阅读并已同意 [《开发者协议》](#)

创建应用

资料填写

完善资料

完成注册

网站域名:

网站回调域:

提供方:

网站备案号:

上传网站图标
64X64(px)

上传网站图标
100X100(px)

创建应用

创建应该审核通过后: 进行如下第3部

3. 获取 APPID 和 APP Key

网站应用		移动应用			创建应用
应用名称	APPID	创建时间	审核状态	操作	
第三登陆测...	101408664	2017-06-12	审核中	--	
内衣加盟网	101243581	2015-09-01	通过	查看	
衣柜加盟网	101238672	2015-07-28	通过	查看	



内衣加盟网

删除应用

APP ID:

APP Key:

基本信息

应用接口

基本信息

应用接口

基本信息

应用接口

网站名称:

内衣加盟网

网站类别:

其他

网站简介:

内衣品牌加盟咨询网站

3-2引入官方SDK

3-2-1引入官方SDK

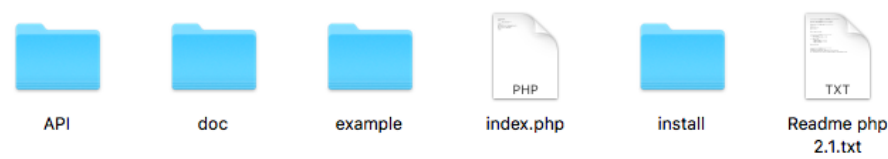
下载SDK:<http://wiki.connect.qq.com/sdk%E4%B8%8B%E8%BD%BD>

SDK for 网站接入

SDK类型	最近更新日期	文件大小	说明
JS SDK	2012-05-30		方便网站快速接入QQ登录，QQ互联提供了JS-widget。 JS-widget是JS封装的SDK，只需要在网站引入JS文件，就可以用封装好的方法访问对应的API，开发简单。使用说明：详见 JS SDK使用说明。 修改历史： V1.0.1 解决了与浏览器已知的兼容性问题。 V1.0.0 适用于基于OAuth2.0的JavaScript网站接入。
PHP SDK v2.1	2013-12-26	104 K	使用说明：详见下载包中doc文件夹中的文档。 修改历史：V2.1 修复了一个安全问题。 MD5: 1A180F106EF785AE4A04E91EA1882A69
Java SDK v2.0	2013-01-07	3990K	使用说明：1.直接引入Sdk4J.jar 包至项目工程内。 2.修改qqconnectconfig.properties 文件，在指定修改的地方填写自己app的相关信息和要获取的scope权限（前4行信息）注意不要试图修改api的请求地址，这里之所以暴露出来是为了方便兼容以后的变动。 3.将qqconnectconfig.properties文件放到自己的项目的context ClassLoader的可以加载的目录下，一般放在项目的src目录即可。 4.查看demo程序，结合api doc文档，进行接口的调用。 MD5: F0D2F0D7A09EC72FB27E46C0B0179470
Ruby On Rails SDK	2012-09-18	4K	本SDK由开发者046569贡献，如有疑问请联系作者(046569@046569.com)。 使用说明：详见【QQ登录】Ruby_On_Rails SDK使用说明。

下载完后，解压到本地网站根目录

3-2-2SDK参数配置



上图是解压后的目录结构：

我们通过访问本地开发环境下的 install 目录来配置SDK参数 如下图



填写完表单后, 点击

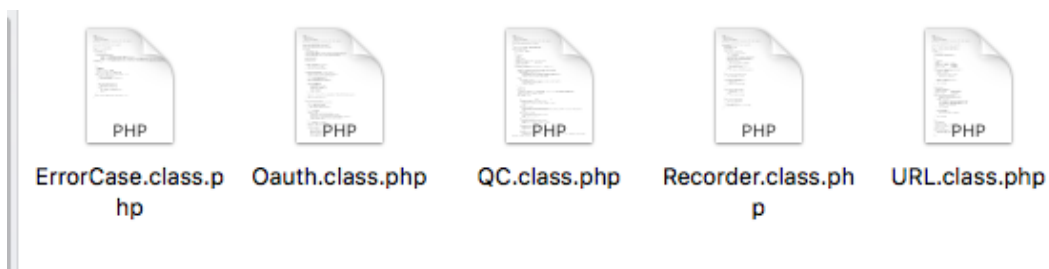
“配置”

按钮

完成配置, 完成后我们进行文件结构整理, 最终只保留API文件夹

3-2-3SDK解读

API 下的 class目录为类库文件, 如下:



ErrorCase.class.php 主要是错误处理

Oauth.class.php 主要是处理授权, 如 引导授权, 取 access_token 、openid

QC.class.php 主类库文件, 所有的接口都是通过此文件调取

Recorder.class.php 用于读取配置信息, 及 存储 access_token, 及 openid

URL.class.php 用于拼接请求URL参数, 及CURL get请求与post请求的封装

3-3整合QQ登录SDK到WEB应用中

3-3-1SDK优化

因为此SDK比较旧了, 很多结构都不是很合理, 也不利于我们将此SDK整合到现有项目中, 因此我们进行优化, 当然不是必须的, 但优化后可以避免一些问题;

优化1

修改: API/comm目录下的config.php 文件

```
define("ROOT",dirname(dirname(__FILE__))."/");  
define("CLASS_PATH",ROOT."class/");
```

这两个常量名(ROOT , CLASS_PATH), 经常被其它的一些框架使用, 很有可能造成常量重定义的错误 因此我们将此常量进行修改

比如: 将ROOT 改为 QQ_CONNECT_ROOT

将CLASS_PATH 改为 QQ_CONNECT_CLASS_PATH

注意: 要将整个API 目录下的这两个常量进行替换, 不只是当前文件

优化2

为了方便我们将SDK集成到ThinkPHP,我们将

API/comm/inc.php文件, 配置成我们TP的形式,

inc.php 文件内容如下: 共两行。

```
<?php die('forbidden'); ?>
{"appid":"101243581","appkey":"6f692cb7680b0592d5d9e24f3cc8843e","callback":"example/oauth/callback.php","scope":"get_user_info","errorReport":true,"storageType":"file","host":"localhost","user":"root","password":"root","database":"test"}
```

第一行是避免被用户通过URL进行访问，没有用

第二行是一个json的配置字符串，我们可以将其转为TP中数组的方式配置，其中

红色部份是没用的信息，可以不用配置

步骤如下：

1. 打一应该中的 config.php 文件 如：Application/Common/Conf/config.php

修改：'LOAD_EXT_CONFIG' => 'weixin,city'，这行配置，如果没有这个配置项，添加一条

将值改为 'LOAD_EXT_CONFIG' => 'weixin,city,qq'，其中的qq 表示加载 qq.php 这个配置文件

2. 在 Application/Common/Conf/ 下新建一个 qq.php 文件

内容如下：

```
<?php
//第三方登录QQ登录配置文件
return array(
    "QQ_CONNECT"=>array(
        "appid"=>"101243581", //APP ID
        "appkey"=>"6f692cb7680b0592d5d9e24f3cc8843e", //APP Key
        "callback"=>"example/oauth/callback.php", //回调地址
        "scope"=>"get_user_info", //需要授权的接口名称 可支持的接口为 http://
        wiki.connect.qq.com/api%E5%88%97%E8%A1%A8 多个接口以逗号分隔
        "errorReport"=>true, //开启错误报告
        "storageType"=>"file", //存储方式 （可以不定义，这个配置没有用）
    ),
);
```

3. 修改API/class/Recorder.class.php文件将如下红框内的代码进行修改


```

require_once(CLASS_PATH."ErrorCase.class.php");
class Recorder{
    private static $data;
    private $inc;
    private $error;

    public function __construct(){
        $this->error = new ErrorCase();

        //-----读取配置文件
        $incFileContents = file(ROOT."comm/inc.php");
        $incFileContents = $incFileContents[1];
        $this->inc = json_decode($incFileContents);
        if(empty($this->inc)){
            $this->error->showError("20001");
        }

        if(empty($_SESSION['QC_userData'])){
            self::$data = array();
        }else{
            self::$data = $_SESSION['QC_userData'];
        }
    }
}

```

将红框内的代码改为如下：

```

public function __construct(){
    $this->error = new ErrorCase();

    //-----读取配置文件

    /*$incFileContents = file(ROOT."comm/inc.php");
    $incFileContents = $incFileContents[1];
    $this->inc = json_decode($incFileContents);*/

    $qq_config = C('QQ_CONNECT'); //取得QQ配置
    if(!is_array($qq_config) || count($qq_config)==0){
        $this->error->showError("40001","请配置QQ信息"); //提示去配置信息
    }
    $this->inc->appid = $qq_config['appid']; //APP ID
    $this->inc->appkey = $qq_config['appkey']; //APP Key
    $this->inc->callback = $qq_config['callback']; //回调地址
    $this->inc->scope = $qq_config['scope']; //需要授权的接口名称
    $this->inc->errorReport = $qq_config['errorReport']; //开启错误报告
    $this->inc->storageType = $qq_config['storageType']; //存储方式

    if(empty($this->inc)){
        $this->error->showError("20001");
    }

    if(empty($_SESSION['QC_userData'])){
        self::$data = array();
    }else{
        self::$data = $_SESSION['QC_userData'];
    }
}

```

修改完后，删除没用的文件：如：

删除：API/comm/inc.php 此文件我们已用TP的方式获取了，因此没用了..

删除：API/comm/utlis.php 此文件应该是遗留文件，我们的类库并没有使用

4. 将API目录移到 Thinkphp/Library/Vendor 目录下，并将API文件夹名称改名为 QQConnect

到此，SDK改造完毕，之后我们可以在TP中使用 Vendor("QQConnect.qqConnectAPI"); 进行加载

3-3-2通过SDK- 请求访问QQ登录页面

```
Vendor("QQConnect.qqConnectAPI"); //加载QQ登录SDK

$qc = new \QC; //实例化QQ登录SDK
$qc->qq_login(); //执行引导授权
```

3-3-3通过SDK - 获取code和accessoken及openid

```
Vendor("QQConnect.qqConnectAPI"); //加载QQ登录SDK

$qc = new \QC; //实例化QQ登录SDK

//步骤1
$access_token = $qc->qq_callback(); //取得access_token

//步骤2
$openid = $qc->qq_openid(); //取得openid
```

3-3-4通过API调用实现常见的业务逻辑

做一个DEMO, 获取QQ用户的信息, 并保存用户登录信息到数据库及session

首先我们前面的步骤已经取到了openid 因此我们可以直接能过openid 取用户信息如下:

```
Vendor("QQConnect.qqConnectAPI"); //加载QQ登录SDK

$qc = new \QC; //实例化QQ登录SDK

//步骤1
$access_token = $qc->qq_callback(); //取得access_token

//步骤2
$openid = $qc->qq_openid(); //取得openid

//步骤3
$userinfo = $qc->get_user_info(); //取得用户信息
```

其次, 我们要将用户信息保存到数据库及session中

1. 创建一张数据表:

```
//用户登录表
CREATE TABLE `echo_demo`.`app_user_login` (
  `id` int(10) NOT NULL AUTO_INCREMENT COMMENT '用户id',
  `uname` varchar(100) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT '' COMMENT '用户名',
  `password` varchar(100) CHARACTER SET utf8 DEFAULT '' COMMENT '用户密码',
  `qq_openid` varchar(100) CHARACTER SET utf8 DEFAULT '' COMMENT '第三登录 QQ登录 OPENID',
  `qq_userinfo` varchar(1000) CHARACTER SET utf8 DEFAULT '' COMMENT 'QQ用户信息 (json字符串)
  用于保存QQ用户的用户信息',
  `login_type` tinyint(1) DEFAULT 0 COMMENT '登录类型: 0 本地账号登录; 1 QQ登录 ; 2 微博登录',
  `login_time` int(11) DEFAULT 0 COMMENT '登录时间',
  PRIMARY KEY (`id`)
) COMMENT='用户登录表 包括第三登录';
```

用户登录表

```
CREATE TABLE `echo_demo`.`app_user_login` (
  `id` int(10) NOT NULL AUTO_INCREMENT COMMENT '用户id',
  `uname` varchar(100) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT '' COMMENT
'用户名',
  `password` varchar(100) CHARACTER SET utf8 DEFAULT '' COMMENT '用户密码',
  `qq_openid` varchar(100) CHARACTER SET utf8 DEFAULT '' COMMENT '第三登录 QQ登录
OPENID',
  `qq_userinfo` varchar(1000) CHARACTER SET utf8 DEFAULT '' COMMENT 'QQ用户信息
(json字符串) 用于保存QQ用户的用户信息',
  `login_type` tinyint(1) DEFAULT 0 COMMENT '登录类型: 0 本地账号登录; 1 QQ登录 ; 2
微博登录',
  `login_time` int(11) DEFAULT 0 COMMENT '登录时间',
  PRIMARY KEY (`id`)
) COMMENT='用户登录表 包括第三登录';
```

2. 将用户信息保存到数据库

```

$qc = new \QC($access_token,$openid); //重新实例化,并将access_token 与 openid 传进去
//步骤3
$userinfo = $qc->get_user_info(); //取得用户信息,数组信息

$json_userinfo = json_encode($userinfo,JSON_UNESCAPED_UNICODE); //将用户信息转为json
//如何获取用户信息出错,则提示错误信息,将结束当前整个逻辑
if($userinfo['ret'] != 0){
    exit("登录失败,请联系管理员.错误原因:".$userinfo['msg']);
}

//步骤4,处理用户信息,并实现站点登录
//如: 将openid 与 $userinfo 的数据写入数据库中,将保存登录状态到session中,这样不用访问每个页面都进行授权。而是直接到session中取数据,跟我们处理正常的登录检查流程一样。如果
过期我们重新引导用户授权。。并更新QQ用户信息到数据库.... 演示代码为:

//检查用户信息是否存在,存在就更新用户数据,不存在添加用户数据
$m = M('UserLogin'); //实例用户登录数据模型
$curr_time = time(); //当前时间
$rs = $m->where(array('qq_openid'=>$openid))->find(); //查询用户是否存在

if($rs){
    //更新用户信息
    $m->where(array('qq_openid'=>$openid))->save(array('qq_userinfo'=>$json_userinfo,'login_time'=>$curr_time));
}else{
    //添加用户信息
    $m->add(array('qq_openid'=>$openid,'qq_userinfo'=>$json_userinfo,'login_type'=>1,'login_time'=>$curr_time));
}

//将用户状态写入session
session(array('expire'=>3600)); //设置过期时间为1个小时
session('is_login',$openid); //设置为已登录
$login_info = array(
    "login_type" => 1, //qq登录
    "user_info" => $userinfo, //用户信息
    "login_time" => $curr_time //登录时间
);
session('login_info',$login_info); //存储登录信息

//跳转到其它页面,如用户中心,或网首页
$this->redirect('userCenter'); //跳转到用户中心

```

3-4 腾讯开放平台政策法规

3-4-1 腾讯开放平台政策法规

平台政策与注意事项

- Appid申请之后3个月未申请上线将被回收
- 申请上线需要使用官网提供的QQ登录按钮素材
- 我们的站点要做到拒绝黄赌毒，遵守国家法律法规

3-4-2 课程总结

注册开发者(审核), 创建应用(审核)

需要在线网站, 及通过备案

授权过程:

1. 引导授权页
2. 用户输账号密码
3. 取code, 然后通过 code 取access_token
4. 通过 access_token 取openid
5. 通过 access_token 及 openid 调用API, 取数据

用到了, curl 官方的SDK,
以及带领到家将SDK集成到ThinkPHP

通过 tp 实现DEMO

4.微博登录DEMO

<http://open.weibo.com/wiki/%E7%BD%91%E7%AB%99%E6%8E%A5%E5%85%A5>

4-1申请接入微博开放平台

4-1-2OAuth授权

文档: <http://open.weibo.com/wiki/%E6%8E%88%E6%9D%83%E6%9C%BA%E5%88%B6%E8%AF%B4%E6%98%8E>

流程:

1. 请求授权: OAuth2/authorize 请求用户授权Token
2. 获取Access_token OAuth2/access_token 获取授权过的Access Token
3. 通过 Access_token 请求API接口

OAuth 所有可用接口:

开发者可以先浏览OAuth2.0的接口文档，熟悉OAuth2.0的接口及参数的含义，然后我们根据应用场景各自说明如何使用OAuth2.0。

接口文档

接口	说明
OAuth2/authorize	请求用户授权Token
OAuth2/access_token	获取授权过的Access Token
OAuth2/get_token_info 新	授权信息查询接口
OAuth2/revokeoauth2 新	授权回收接口
OAuth2/get_oauth2_token	OAuth1.0的Access Token更换至OAuth2.0的Access Token

1. OAuth2的authorize接口

接口地址：<https://api.weibo.com/oauth2/authorize>

请求参数

	必选	类型及范围	说明
client_id	true	string	申请应用时分配的AppKey。
redirect_uri	true	string	授权回调地址，站外应用需与设置的回调地址一致，站内应用需填写canvas page的地址。
scope	false	string	申请scope权限所需参数，可一次申请多个scope权限，用逗号分隔。 使用文档
state	false	string	用于保持请求和回调的状态，在回调时，会在Query Parameter中回传该参数。开发者可以用这个参数验证请求有效性，也可以记录用户请求授权页前的位置。这个参数可用于防止跨站请求伪造（CSRF）攻击
display	false	string	授权页面的终端类型，取值见下面的说明。
forcelogin	false	boolean	是否强制用户重新登录，true：是，false：否。默认false。
language	false	string	授权页语言，缺省为中文简体版，en为英文版。英文版测试中，开发者任何意见可反馈至 @微博API

display说明：

参数取值	类型说明
default	默认的授权页面，适用于web浏览器。
mobile	移动终端的授权页面，适用于支持html5的手机。注：使用此版授权页请用 https://open.weibo.cn/oauth2/authorize 授权接口
wap	wap版授权页面，适用于非智能手机。
client	客户端版本授权页面，适用于PC桌面应用。
apponweibo	默认的站内应用授权页，授权后不返回access_token，只刷新站内应用父框架。

返回数据

返回值字段	字段类型	字段说明
code	string	用于第二步调用oauth2/access_token接口，获取授权后的access token。
state	string	如果传递参数，会回传该参数。

示例

```
//请求  
https://api.weibo.com/oauth2/authorize?client_id=123050457758183&redirect_uri=http://www.example.com  
  
//同意授权后会重定向  
http://www.example.com/response?code=CODE
```

2. OAuth2的access_token接口

接口地址: https://api.weibo.com/oauth2/access_token

请求参数

	必选	类型及范围	说明
client_id	true	string	申请应用时分配的AppKey。
client_secret	true	string	申请应用时分配的AppSecret。
grant_type	true	string	请求的类型，填写authorization_code

grant_type为authorization_code时

	必选	类型及范围	说明
code	true	string	调用authorize获得的code值。
redirect_uri	true	string	回调地址，需需与注册应用里的回调地址一致。

返回数据

```
{
  "access_token": "ACCESS_TOKEN",
  "expires_in": 1234,
  "remind_in": "798114",
  "uid": "12341234"
}
```

返回值字段	字段类型	字段说明
access_token	string	用户授权的唯一票据，用于调用微博的开放接口，同时也是第三方应用验证微博用户登录的唯一票据，第三方应用应该用该票据和自己应用内的用户建立唯一映射关系，来识别登录状态，不能使用本返回值里的UID字段来做登录识别。
expires_in	string	access_token的生命周期，单位是秒数。
remind_in	string	access_token的生命周期（该参数即将废弃，开发者请使用expires_in）。
uid	string	授权用户的UID，本字段只是为了方便开发者，减少一次user/show接口调用而返回的，第三方应用不能用此字段作为用户登录状态的识别，只有access_token才是用户授权的唯一票据。

3. oauth2的get_token_info

查询用户access_token的授权相关信息，包括授权时间，过期时间和scope权限。

接口地址：https://api.weibo.com/oauth2/get_token_info

请求参数

access_token: 用户授权时生成的access_token。

返回数据

```
{
  "uid": 1073880650,
  "appkey": 1352222456,
  "scope": null,
  "create_at": 1352267591,
  "expire_in": 157679471
}
```

返回值字段	字段类型	字段说明
uid	string	授权用户的uid。
appkey	string	access_token所属的应用appkey。
scope	string	用户授权的scope权限。
create_at	string	access_token的创建时间，从1970年到创建时间的秒数。
expire_in	string	access_token的剩余时间，单位是秒数。

4. Oauth2的revokeoauth2 授权回收接口，帮助开发者主动取消用户的授权。

接口地址: <https://api.weibo.com/oauth2/revokeoauth2>

请求参数

access_token: 用户授权应用的access_token

返回数据

```
{
  "result": "true"
}
```

使用场景说明

- 1. 应用下线时，清空所有用户的授权
- 2. 应用新上线了功能，需要取得用户scope权限，可以回收后重新引导用户授权
- 3. 开发者调试应用，需要反复调试授权功能
- 4. 应用内实现类似登出微博帐号的功能

4-1-2接入新浪微博开放平台的前置条件



1. 成为开发者, <http://open.weibo.com/development> 登录成功就是开发者了...

2. 创建应用

微连接
Weibo Connect

微连接是微博开放平台对非商业第三方合作的统称。主要分为账号互通、微博组件、SDK与其他微连接应用，涵盖除轻应用外的所有应用类型。

立即创建微连接

- 整合微博资源**
快速融入微博，打造高度社交特性应用
- 增加新用户**
使用微博登录，降低注册门槛
- 品牌传播**
利用内容传播获取更多品牌曝光及口碑
- 传播内容**
利用微博传播渠道，促进用户分享内容

选择您要创建的应用类型



i 当前创建的是微连接应用，如需使用“粉服接口”或“轻应用框架”请到[微服务](#)页面选择需要的产品功能进行接入。



网站接入

部署微博组件，账号
互联互通

案例

音悦台 蚂蜂窝



移动应用

移动客户端应用

案例

啪啪 美拍



网页应用

搭建基于微博的站外
网页应用

案例

省电大师 KFC



其他

浏览器插件、车载系
统、智能设备等应用

案例

搜狗小飞机

1. 验证网站所有权 2. 部署微连接 3. 提交审核

网站名称: 宁静夏天 ✓

网站域名: yyw66.cn ✓

选择验证方式: ☐ 在您的网站首页添加<meta>标签

☒ 将验证文件上传到您的网站根目录
(WAP站请使用此方式验证)

wb_adda2a963b1ceae1.txt [点击下载]

注意: 请勿修改文件名

1.点击下载验证文件;
2.不要修改文件名, 将文件上传到网站根目录下;
3.点击"验证并添加"按钮。
适用场景: WAP站或首页有自动跳转的Web站;
注意事项: 使用此种验证方式通常需要您拥有网站的FTP管理权限。

☒ 我已阅读并接受《新浪微博开发者协议》

验证并添加

重新填写

技术支持邮箱: weibo_api_tech@sina.com

下载 第一个框的文件, 上传到网站要目录后, 点击“验证并添加”

添加新网站

1. 验证网站所有权

2. 部署微连接

3. 提交审核

✔ 恭喜！我们已经验证了您对网站的管理权！并成功获取 App Key 和 App Secret

App Key: [REDACTED]

App Sercet: [REDACTED]

您接下来可能要

快速部署: [微组件](#)

深度部署: [微博登录](#) | [JS-SDK或JS组件](#)

稍后部署: [先了解网站控制台](#)

下一步

技术支持邮箱: weibo_api_tech@sina.com

1. 验证网站所有权	2. 部署微连接	3. 提交审核
网站名称: 宁静的夏天		也用于发布微博的来源显示, 不超过10个汉字或20个字母。如果您填写的名称未在网站首页的title出现, 可能会无法通过审核。
网站域名: echo.yyw66.cn		
ICP备案号: 境内网站 粤ICP备16077655		✓
来源链接: http://echo.yyw66.cn		✓
网站分类: 个人站		
面向终端: <input checked="" type="radio"/> 个人电脑 <input type="radio"/> 移动终端		
网站简介: 个人信息展示		✓
网站介绍: 本站主要用于展示个人信息, 及网站功能测试用途, 如第三登录测试, 信息公众号测试		✓
网站图标: 上传图片		✓
		
<div>提交审核</div>		每天最多可提交3次

4-1-3 获取app key和app secret

审核通过后, 登录开发者中心

 微博·开放平台

微连接 ▾ 微服务 ▾ 文档 支持 推广 **我的应用**

 caomao_girl ▾

我的应用

open.weibo.com/apps

**caomao_girl** 广东, 广州

开发者名称: 罗慧华 开发者类型: 个人 基本信息: 已完善
身份认证: 已认证 级别: 未评级 NEW 验证邮箱: veiol@qq.com 已验证

 微博开放平台 **V**
开放平台 [+ 关注](#)

**宁静的夏天**
微连接 - 连接网站

上线

 微博·开放平台

微连接 ▾ 微服务 ▾ 文档 支持 推广 我的应用

 caomao_girl ▾

宁静的夏天

 网站控制台

 数据统计 >

 部署微连接 >

 网站信息 >

 接口管理 >

 管理网站 >

**宁静的夏天**
App Key: 3605148157

网站状态:
验证 **部署** **申请** **上线**

基本状态:
 **上线** **点击这里**

 昨日新增分享: **0**

 昨日新增用户: **0**

链接分享

☒ 新增链接 [?] ☐ 新增原创微博 [?] ☐ 新增转发微博 [?] ☐ 点击数 [?]

10							
8							



4-2 微博登录开发

4-2-1 本地开发环境配置

我们使用WAPM ThinkPHP 环境

4-2-2 引入官方SDK

下载地址：

<https://github.com/xiaosier/libweibo/archive/master.zip>

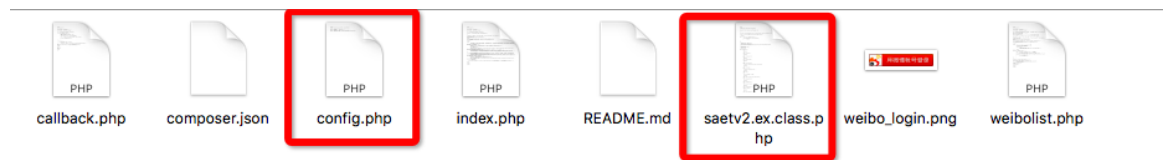
4-2-3 SDK解读

使用微博账号登录，将要用到的SDK 类和函数如下：

1. `new SaeTOAuthV2($appkey , $appsecret);` 实例化授权类
2. 获取授权引导地址函数: `getAuthorizeURL($url)` 参数为跳转地址
3. `getAccessToken($type, $keys);`
 - * @param string \$type 请求的类型,可以为:code, password, token
 - * @param array \$keys 其他参数:
 - * - 当\$type为code时: `array('code'=>..., 'redirect_uri'=>...)`
 - * - 当\$type为password时: `array('username'=>..., 'password'=>...)`
 - * - 当\$type为token时: `array('refresh_token'=>...)`
4. `new SaeTClientV2($appkey , $appsecret , $access_token);` 实例化API请求类
5. `get_uid();` 取用户ID
6. `show_user_by_id($uid)` 取得用户信息

4-2-4将SDK引入ThinkPHP

解压下载的SDK包, 结构为:



有用的文件只有两个, 一个是配置文件 `config.php` 我们将其个性为TP的配置方式,

`saetv2.ex.class.php` 此文件是主要的类库文件。

改造方式如下:

1. 修改 本地环境下 `Application/Common/Conf/config.php`

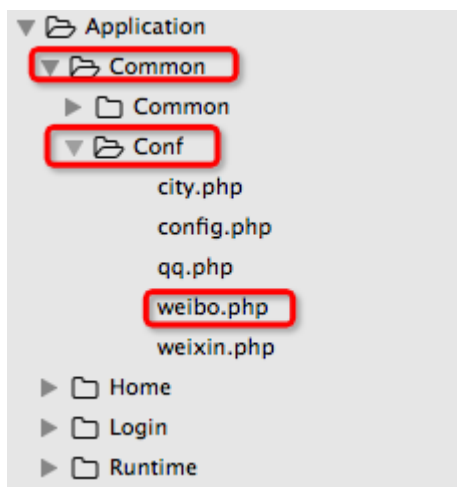
找到此行 : `'LOAD_EXT_CONFIG' => 'weixin,city,qq',`

添加一个配置项, 修改后为: 'LOAD_EXT_CONFIG'=> 'weixin,city,qq,weibo',

2. 在 Application/Common/Conf/ 目录下添加一个weibo.php的配置文件: 内容为

```
<?php
return array(
    "WEIBO_CONNECT"=>array(
        "app_key"=>"3605148157", //App Key
        "app_secret"=>"76945869cea6b1729034330ae0091d38", //App Secret
        "callback"=>"http://echo.yyw66.cn/login.php/Index/actWeiboLogin", //回调地
    ),
);
```

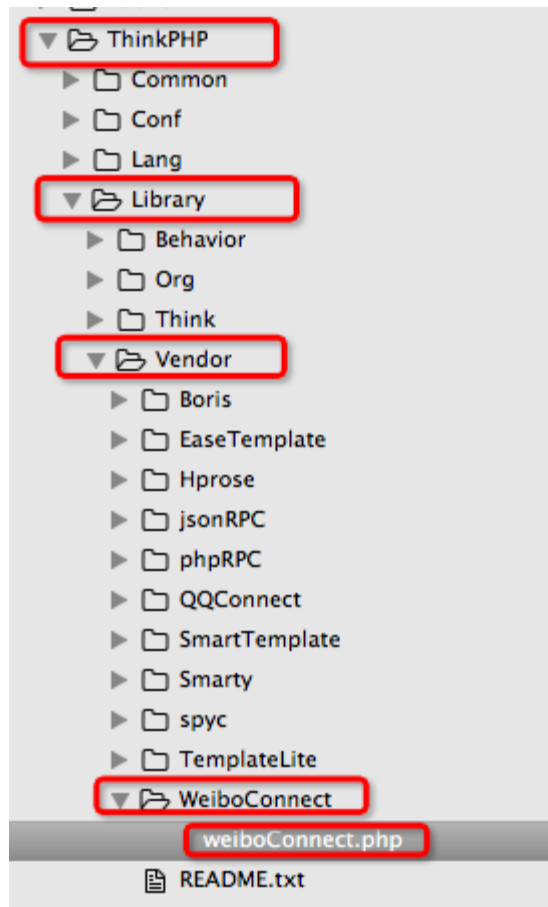
结构为:



3. 在 ThinkPHP/library/Vendor 下创建一个文件夹 名为: WeiboConnect

将上面解压的 seatv2.ex.class.php 复制到 ThinkPHP/library/Vendor/WeiboConnect 下并重命名为 weiboConnect.php

最终TP目录结构为:



4-2-5开发实例

获取url完成跳转

```
Vendor("WeiboConnect.weiboConnect"); //加载weibo登录SDK

$config_weibo = C('WEIBO_CONNECT'); //微博配置
$o = new \SaeT0AuthV2($config_weibo['appid'],$config_weibo['appkey']); //实例化授权类

$url = $o->getAuthorizeURL($config_weibo['callback']); //引导授权地址

redirect($url); //跳转到授权页
```

获取accesstoken

如下代码必须添加到回调函数页面，在TP中操作的话，就是要在对应方法中写如下代码：

比如，我们当前 weibo 配置文件

```

<?php
return array(
    "WEIBO_CONNECT"=>array(
        "appid"=>"微博开放平台获取", //appid
        "appkey"=>"微博开放平台获取", //appkey
        "callback"=>"http://echo.yyw66.cn/login.php/Index/actWeiboLogin", //回调地址
    ),
);

```

因此我们要在 `actWeiboLogin` 方法中处理 获取 `access_token` 的操作, 同时, 服务器会在地址中插入 `code` 参数

```

public function actWeiboLogin(){
    /*步骤
    * 1. 获取 code
    * 2. 通过 code 获取 access_token
    * 3. 获取 uid
    * 4. 获取用户信息
    * 5. 存储用户登录状态,
    * 6. 及将用户信息写入数据库
    */

    Vendor("WeiboConnect.weiboConnect"); //加载weibo登录SDK
    $config_weibo = C('WEIBO_CONNECT'); //微博配置
    $o = new \SaeT0AuthV2($config_weibo['app_key'],$config_weibo['app_secret']); //实例化授权类

    //获取access_token
    if (I('get.code')!="") {
        $keys = array();
        // 1. 获取 code
        $keys['code'] = I('get.code'); //取得服务器返回授权码 code 值
        $keys['redirect_uri'] = $config_weibo['callback']; //回调地址
        try {
            //2. 获取 access_token
            $token = $o->getAccessToken( 'code', $keys );
        } catch (OAuthException $e) {
        }
    }
}

```

获取用户信息接口

users/show 获取用户信息

接口地址: <https://api.weibo.com/2/users/show.json>

JSON示例

```

{
    "id": 1404376560,
    "screen_name": "zaku",

```

```

"name": "zaku",
"province": "11",
"city": "5",
"location": "北京 朝阳区",
"description": "人生五十年, 乃如梦如幻;有生斯有死, 壮士复何憾。",
"url": "http://blog.sina.com.cn/zaku"
"profile_image_url": "http://tp1.sinaimg.cn/1404376560/50/0/1"
"domain": "zaku",
"gender": "m",
"followers_count": 1204,
"friends_count": 447,
"statuses_count": 2908,
"favourites_count": 0,
"created_at": "Fri Aug 28 00:00:00 +0800 2009",
"following": false,
"allow_all_act_msg": false,
"geo_enabled": true,
"verified": false,
"status": {
  "created_at": "Tue May 24 18:04:53 +0800 2011",
  "id": 11142488790,
  "text": "我的相机到了。",
  "source": "<a href='\"http://weibo.com\"' rel='\"nofollow\"'>新浪微博</a>"
  "favorited": false,
  "truncated": false,
  "in_reply_to_status_id": "",
  "in_reply_to_user_id": "",
  "in_reply_to_screen_name": "",
  "geo": null,
  "mid": "5610221544300749636",
  "annotations": [],
  "reposts_count": 5,
  "comments_count": 8
},
"allow_all_comment": true,
"avatar_large": "http://tp1.sinaimg.cn/1404376560/180/0/1"
"verified_reason": "",
"follow_me": false,
"online_status": 0,
"bi_followers_count": 215
}

```

请求参数

	必选	类型及范围	说明
access_token	true	string	采用OAuth授权方式为必填参数，OAuth授权后获得。
uid	false	int64	需要查询的用户ID。
screen_name	false	string	需要查询的用户昵称。

注意事项

参数uid与screen_name二者必选其一，且只能选其一；
接口升级后，对未授权本应用的uid，将无法获取其个人简介、认证原因、粉丝数、关注数、微博数及最近一条微博内容。

返回字段说明

返回值字段	字段类型	字段说明
id	int64	用户UID
idstr	string	字符串型的用户UID
screen_name	string	用户昵称
name	string	友好显示名称
province	int	用户所在省级ID
city	int	用户所在城市ID
location	string	用户所在地
description	string	用户个人描述
url	string	用户博客地址
profile_image_url	string	用户头像地址（中图），50×50像素
profile_url	string	用户的微博统一URL地址
domain	string	用户的个性化域名
weihao	string	用户的微号
gender	string	性别，m：男、f：女、n：未知
followers_count	int	粉丝数
friends_count	int	关注数
statuses_count	int	微博数
favourites_count	int	收藏数
created_at	string	用户创建（注册）时间
following	boolean	暂未支持
allow_all_act_msg	boolean	是否允许所有人给我发私信，true：是，false：否
geo_enabled	boolean	是否允许标识用户的地理位置，true：是，false：否
verified	boolean	是否是微博认证用户，即加V用户，true：是，false：否
verified_type	int	暂未支持
remark	string	用户备注信息，只有在查询用户关系时才返回此字段
status	object	用户的最近一条微博信息字段 详细
allow_all_comment	boolean	是否允许所有人对我的微博进行评论，true：是，false：否
avatar_large	string	用户头像地址（大图），180×180像素
avatar_hd	string	用户头像地址（高清），高清头像原图
verified_reason	string	认证原因
follow_me	boolean	该用户是否关注当前登录用户，true：是，false：否
online_status	int	用户的在线状态，0：不在线、1：在线
bi_followers_count	int	用户的互粉数
lang	string	用户当前的语言版本，zh-cn：简体中文，zh-tw：繁体中文，en：英语


```
//获取uid
if ($token) {
    //dump($token);
    $access_token = $token['access_token']; // access_token

    $c = new \SaeTClientV2( $config_weibo['app_key'],$config_weibo['app_secret'], $access_token ); //实例化api请求类
    $arr_uid = $c->get_uid(); //获取uid数据
    //3. 获取uid
    $uid = $arr_uid['uid']; //获取uid 也可以直接从 $token['uid'] 获取
    $arr_user = $c->show_user_by_id( $uid); //根据UID获取用户等基本信息

    //4. 获取用户信息
    $json_userinfo = json_encode($arr_user,JSON_UNESCAPED_UNICODE); //将用户信息转为json字符

    //5. 存储登录状态信息
    $curr_time = time(); //当前时间
    session(array('expire'=>3600)); //初始session
    session('is_login',$uid); //设置为已登录
    $login_info = array(
        "login_type" => 2, //weibo登录
        "user_info" => $arr_user, //用户信息
        "login_time" => $curr_time //登录时间
    );
    session('login_info',$login_info); //存储登录信息

    //6. 及将用户信息写入数据库
    //user_login 数据库表 添加weibo_uid 和 weibo_userinfo 两个字段 用于存储用户信息
    $m = M('UserLogin');
    $rs = $m->where(array('weibo_uid'=>$uid))->find(); //检查用户是否存在
    if($rs){
        //更新登录信息
        $m->where(array('weibo_uid'=>$uid))->save(array("weibo_userinfo"=>$json_userinfo,'login_time'=>$curr_time));
    }else{
        //添加登录信息
        $m->add(array('weibo_uid'=>$uid,'weibo_userinfo'=>$json_userinfo,'login_type'=>2,'login_time'=>$curr_time));
    }

    //跳转到用户中心
    $this->redirect("userCenter");
}
```

获取用户信息

保存登录状态

```
//获取uid
if ($token) {
    //dump($token);
    $access_token = $token['access_token']; // access_token

    $c = new \SaeTClientV2( $config_weibo['app_key'],$config_weibo['app_secret'], $access_token ); //实例化api请求类
    $arr_uid = $c->get_uid(); //获取uid数据
    //3. 获取uid
    $uid = $arr_uid['uid']; //获取uid 也可以直接从 $token['uid'] 获取
    $arr_user = $c->show_user_by_id( $uid); //根据UID获取用户等基本信息

    //4. 获取用户信息
    $json_userinfo = json_encode($arr_user,JSON_UNESCAPED_UNICODE); //将用户信息转为json字符

    //5. 存储登录状态信息
    $curr_time = time(); //当前时间
    session(array('expire'=>3600)); //初始session
    session('is_login',$uid); //设置为已登录
    $login_info = array(
        "login_type" => 2, //weibo登录
        "user_info" => $arr_user, //用户信息
        "login_time" => $curr_time //登录时间
    );
    session('login_info',$login_info); //存储登录信息

    //6. 及将用户信息写入数据库
    //user_login 数据库表 添加weibo_uid 和 weibo_userinfo 两个字段 用于存储用户信息
    $m = M('UserLogin');
    $rs = $m->where(array('weibo_uid'=>$uid))->find(); //检查用户是否存在
    if($rs){
        //更新登录信息
        $m->where(array('weibo_uid'=>$uid))->save(array("weibo_userinfo"=>$json_userinfo,'login_time'=>$curr_time));
    }else{
        //添加登录信息
        $m->add(array('weibo_uid'=>$uid,'weibo_userinfo'=>$json_userinfo,'login_type'=>2,'login_time'=>$curr_time));
    }

    //跳转到用户中心
    $this->redirect("userCenter");
}
```

保存登录状态

写入数据库

4-3课程总结

微信登录跟QQ登录所使用的授权都是oauth2.0

原理都一样,

只是API接口不一样, 返回的数据不一样

从官方提供的SDK比较可以看出, 微博的SDK更精简, 而QQ的相对较臃肿, 原因是QQ的SDK比较旧, 没有更新, 具我估计, 这个SDK是实习生写的, 感觉比较乱, 耦合太高, 不利于使用, 因此有基础的同学可以课后自己重写QQ的SDK, 功能只要求登录使用, 其它的接口可以不必考虑

四、小节

2. 课堂练习

3. 课后练习

4. 资料扩展