

4-5 微信access_token与微信SDK

4-5 微信access_token与微信SDK.....	1
1. 课堂案例	2
课堂讲解	2
一、上节回顾	2
微信事件接受与消息回复	2
二、学习目标	2
获取access_token.....	2
创建自己的SDK.....	2
公众号实现天气查询	2
三、教学过程 描述	2
1.关于access_token.....	2
2.微信sdk的实现.....	8
3.第三方天气查询接口在微信中的使用	10
四、小节	12
2. 课后练习	12
3. 课堂练习	12
4. 资料扩展	12



1. 课堂案例

课堂讲解

一、上节回顾

微信事件接受与消息回复

二、学习目标

获取access_token

创建自己的SDK

公众号实现天气查询

三、教学过程 描述

1. 关于access_token

1-1access_token的特性

access_token 的概念

access_token 访问令牌 的意思，类似于我们的暗号，如果暗号正确表示身份确认，即可进行下面的流程。

access_token是公众号的全局唯一接口调用凭据，公众号调用各接口时都需使用access_token

特性

- 1. 唯一性
- 2. 时效性

1-2access_token说明

我们可以通过 接口来获取 access_token

备注:前面我们说过, access_token是调用接口的唯一凭证, 而access_token的获取也是通过接口, 这就意味着, 我们如果要使用任何接口, 都需要调用两个接口, 第一个接口是取得access_token的接口, 第二个接口是真实使用的那个接口。但是我们知道, access_token接口的获取是有限制的, 如下图

体验接口权限表				
类目	功能	接口	每日调用上限/次	操作
	基础支持	获取access_token	2000	
		获取微信服务器IP地址	无上限	

如图中我们的测试号, 每天只能拉取2000次, 要是我们的公众号有粉丝5000人, 显然有3000个粉丝是拉取不到 access_token, 因此, 出现一个概念, 叫作有效期, 每次拉取的access_token 都有 7200秒即两个小时的有效期, 这意味着, 理想状态下我们一天只要拉取12次就能满足要求。

不需要 用到 access_token 的地方

关注/取消关注公众号的事件, 二维码扫描的事件, 以能让消息的被回复等都不需要access_token ,因此这个都不要调用接口, 只是事件推送与回复

需要 用到 access_token 的地方

只要需要调用接口，如拉取服务器地址，菜单管理，部份消息管理，素材管理，用户管理，账号管理等，都是需要 access_token

access_token 获取说明

接口调用请求说明

http请求方式: GET
https://api.weixin.qq.com/cgi-bin/token?grant_type=client_credential&appid=APPID&secret=APPSECRET

参数说明

参数	是否必须	说明
grant_type	是	获取access_token填写client_credential
appid	是	第三方用户唯一凭证
secret	是	第三方用户唯一凭证密钥，即appsecret

返回说明

正常情况下，微信会返回下述JSON数据包给公众号：
{ "access_token": "ACCESS_TOKEN", "expires_in": 7200 }

参数	说明
access_token	获取到的凭证
expires_in	凭证有效时间，单位：秒

错误时微信会返回错误码等信息，JSON数据包示例如下(该示例为AppID无效错误)：
{ "errcode": 40013, "errmsg": "invalid appid" }

1-3curl的简单实例

通过CURL模拟 GET 或 POST 请求

关键代码：

CURL 模拟 GET 请求

```
/**
 * CURL 模拟 GET 请求
 * @param string $str_api      请求地址
 * @param array $arr_param     请求参数
 * @param string $str_returnType 数据返回类型, 默认为数组方式
 * @return array 或 string     返回数组或字符串
 */
function getRequest($str_api,$arr_param=array(),$str_returnType='array'){
    if(!$str_api){
        exit('request url is empty 请求地址不正确');
    }
    //url拼装
    if(is_array($arr_param) && count($arr_param)>0){
        $tmp_param = http_build_query($arr_param);
        if(strpos($str_api, '?') !== false){ $str_api .= "&".$tmp_param; }
        else{ $str_api .= "?" . $tmp_param; }
    }elseif (is_string($arr_param)){
        if(strpos($str_api, '?') !== false){ $str_api .= "&".$arr_param; }
        else{ $str_api .= "?" . $arr_param; }
    }
    //请求头
    $this_header = array("content-type: application/x-www-form-urlencoded; charset=UTF-8");

    $ch = curl_init(); //初始curl
    curl_setopt($ch,CURLOPT_URL,$str_api); //需要获取的 URL 地址
    curl_setopt($ch,CURLOPT_HEADER,0); //启用时会将头文件的信息作为数据流输出, 此处禁止输出头信息
    curl_setopt($ch,CURLOPT_RETURNTRANSFER,1); //获取的信息以字符串返回, 而不是直接输出
    curl_setopt($ch,CURLOPT_CONNECTTIMEOUT,30); //连接超时时间
    curl_setopt($ch,CURLOPT_HTTPHEADER,$this_header); //头信息
    curl_setopt($ch, CURLOPT_ENCODING, 'gzip');
    $res = curl_exec($ch); //执行curl请求
    $response_code = curl_getinfo($ch);
    if(curl_errno($ch)){ exit('Curl error: ' . curl_error($ch)); }
    //请求成功
    if($response_code['http_code'] == 200){
        if($str_returnType == 'array'){ return json_decode($res,true); }
        else{ return $res; }
    }else{
        $code = $response_code['http_code'];
        switch ($code) {
            case '404':
                exit('请求的页面不存在');
                break;
            default:
                # code...
                break;
        }
    }
}
```

CURL 模拟 POST 请求

```

/**
 * /发送POST数据
 * @param string $str_api      接口地址
 * @param string $post_data    要发送的post数据
 * @param string $str_returnType 返回的数据类型默认是数组
 * @return 默认返回数组
 */
public function postRequest($str_api,$post_data,$str_returnType='array'){
    if(!$str_api){
        exit('request url is empty 请求地址不正确');
    }

    $ch = curl_init(); //初始curl
    curl_setopt($ch,CURLOPT_URL,$str_api); //需要获取的 URL 地址
    curl_setopt($ch,CURLOPT_HEADER,0); //启用时会将头文件的信息作为数据流输出，此处禁止输出头信息
    curl_setopt($ch,CURLOPT_RETURNTRANSFER,1); //获取的信息以字符串返回，而不是直接输出
    curl_setopt($ch,CURLOPT_CONNECTTIMEOUT,30); //连接超时时间
    //curl_setopt($ch,CURLOPT_HTTPHEADER,$this_header); //头信息
    curl_setopt($ch, CURLOPT_ENCODING, 'gzip');
    curl_setopt($ch, CURLOPT_POST, 1); //post请求
    //curl_setopt($ch, CURLOPT_SAFE_UPLOAD, false); // PHP 5.6.0 后必须开启
    curl_setopt($ch, CURLOPT_POSTFIELDS, $post_data);
    $res = curl_exec($ch); //执行curl请求
    $response_code = curl_getinfo($ch);

    //请求出错
    if(curl_errno($ch)){
        echo 'Curl error: ' . curl_error($ch). "<br>";
        //echo $res;
        var_dump($response_code);
    }

    //请求成功
    if($response_code['http_code'] == 200){
        if($str_returnType == 'array'){
            return json_decode($res,true);
        }else{
            return $res;
        }
    }else{
        $code = $response_code['http_code'];
        switch ($code) {
            case '404':
                exit('请求的页面不存在');
                break;

            default:
                # code...
                break;
        }
    }
}
}

```

1-4获取access_token和微信服务器IP地址

获取access_token 关键代码

```

/**
 * 取得access_token
 * @return string 成功则返回 access_token 否则返回false
 */
function getToken($reset=false){
    //缓存初始化
    S(array('type'=>'file','prefix'=>'dodiapp_', 'expire'=>3600));
    // 获取access_token api接口: "https://api.weixin.qq.com/cgi-bin/token?grant_type=client_credential&appid=APPID&secret=APPSECRET";
    if($reset){
        S('access_token',null);
    }
    $access_token = S('access_token');
    if(!$access_token){
        $appid = 'wx02129124b3746f6'; //公众平台获取
        $appsecret = 'a68cea8d19234a3e41f5f34f97520f9c'; //公众平台获取
        $api_get_token = "https://api.weixin.qq.com/cgi-bin/token"; //获取access_token 的 api接口
        //传递的参数
        $param = array(
            'grant_type'=>'client_credential',
            'appid'=>$appid,
            'secret'=>$appsecret
        );
        //执行curl get请求
        $sarr = getRequest($api_get_token,$param);

        if($sarr['errcode'] != 0){
            $this->writeLog($sarr['errcode'],$sarr['errmsg']);//写错误日志
            return false;
        }else{
            $access_token = $sarr['access_token']; //取得access_token
            S('access_token',$sarr['access_token']); //存储access_token
        }
    }
    return $access_token; //返回access_token
}

```

获取微信服务器IP地址接口

接口调用请求说明

http请求方式: GET

https://api.weixin.qq.com/cgi-bin/getcallbackip?access_token=ACCESS_TOKEN

参数说明

参数	是否必须	说明
access_token	是	公众号的access_token

返回说明

正常情况下, 微信会返回下述JSON数据包给公众号:

```

{
  "ip_list": [
    "127.0.0.1",
    "127.0.0.2",
    "101.226.103.0/25"
  ]
}

```

参数	说明
ip_list	微信服务器IP地址列表

错误时微信会返回错误码等信息, JSON数据包示例如下(该示例为AppID无效错误):

```
{"errcode":40013,"errmsg":"invalid appid"}
```

关键代码

```
/**
 * 取得微信服务器IP地址列表
 * @return array 返回微信服务器的IP列表数组
 */
public function getWeixinIp(){
    $access_token = $this->getToken(); // 获取access_token
    $api = "https://api.weixin.qq.com/cgi-bin/getcallbackip"; //接口地址
    //请求参数
    $param = array(
        'access_token'=>$access_token
    );
    //发送get 请求
    $arr = $this->getRequest($api,$param);
    return $arr['ip_list'];
}
```

2.微信sdk的实现

2-1微信SDK介绍

什么是SDK

SDK (Software Development Kit) 软件开发工具包

SDK就是对一系统功能进行代码的封装, 成为一种工具

微信SDK, 就是我们将, 微信的一系列功能, 进行封装, 将微信开发的过程, 封装成工具, 使用方便复用, 和加快开发, 和方便维护。本节课将在TP中介绍 如何在用TP进行封装自己的微信 SDK

2-2改造之前代码为SDK

我们基本TP环境, 对之前的代码进行改造, 封装成SDK

改造原则：

1. 封装微信接入
2. 封装获取 access_token 事件
3. 封装获取微信服务器IP地址
1. 封装 关注, 与取消关注事件
2. 封装消息回复事件
3. 封装扫描事件

步骤：

1. 新建一个控制器层专门处理微信 (Weixin 层)
2. 新建 CommonWeixin.class.php文件, 用于处理微信公共事宜
3. 新建 MessageWeixin.class.php文件, 用于处理消息回复
4. 新建 EventWeixin.class.php文件, 用于处理推送事件

过程: 请看DEMO ,将代码转移到以上3个文件中

2-3微信SDK - 多图文的代码实现

使用SDK的方式回复多图文消息

1. 回复图文消息

```
<xml>
<ToUserName><![CDATA[toUser]]></ToUserName>
<FromUserName><![CDATA[fromUser]]></FromUserName>
<CreateTime>12345678</CreateTime>
<MsgType><![CDATA[news]]></MsgType>
<ArticleCount>2</ArticleCount>
<Articles>
<item>
<Title><![CDATA[title1]]></Title>
<Description><![CDATA[description1]]></Description>
<PicUrl><![CDATA[picurl]]></PicUrl>
<Url><![CDATA[url]]></Url>
```

```

</item>
<item>
<Title><![CDATA[title]]></Title>
<Description><![CDATA[description]]></Description>
<PicUrl><![CDATA[picurl]]></PicUrl>
<Url><![CDATA[url]]></Url>
</item>
</Articles>
</xml>

```

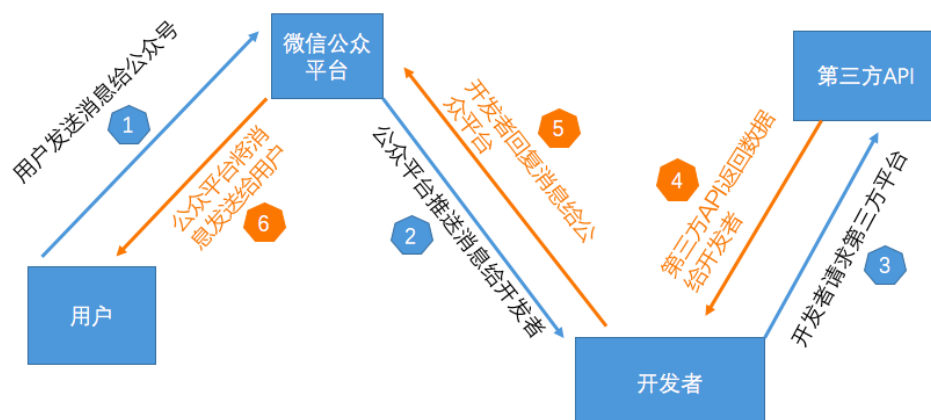
参数	是否必须	说明
ToUserName	是	接收方帐号（收到的OpenID）
FromUserName	是	开发者微信号
CreateTime	是	消息创建时间 （整型）
MsgType	是	news
ArticleCount	是	图文消息个数，限制为8条以内
Articles	是	多条图文消息信息，默认第一个item为大图,注意，如果图文数超过8，则将会无响应
Title	是	图文消息标题
Description	是	图文消息描述
PicUrl	是	图片链接，支持JPG、PNG格式，较好的效果为大图360*200，小图200*200
Url	是	点击图文消息跳转链接

3.第三方天气查询接口在微信中的使用

3-1第三方API在微信中的实现

实现原理：

第三方API在微信中的实现原理



3-2根据城市ID查询当地天气

实现步骤：

1. 接收用户发送的查询天气消息:比如: 天气/广州 或 天气/101280101
2. 服务器检查用户发送的内容格式: 如果符合第一步的格式, 就调用第三方的天气查询API, 如果不符合, 直接回复不能查询到用户的天气
3. 获取到天气信息后, 将天气信息整理后, 回复给用户

3-3根据城市名查询当地天气代码实现

实现效果



3-4根据城市名查询当地天气代码测试

演示

四、小节

2. 课后练习

3. 课堂练习

4. 资料扩展