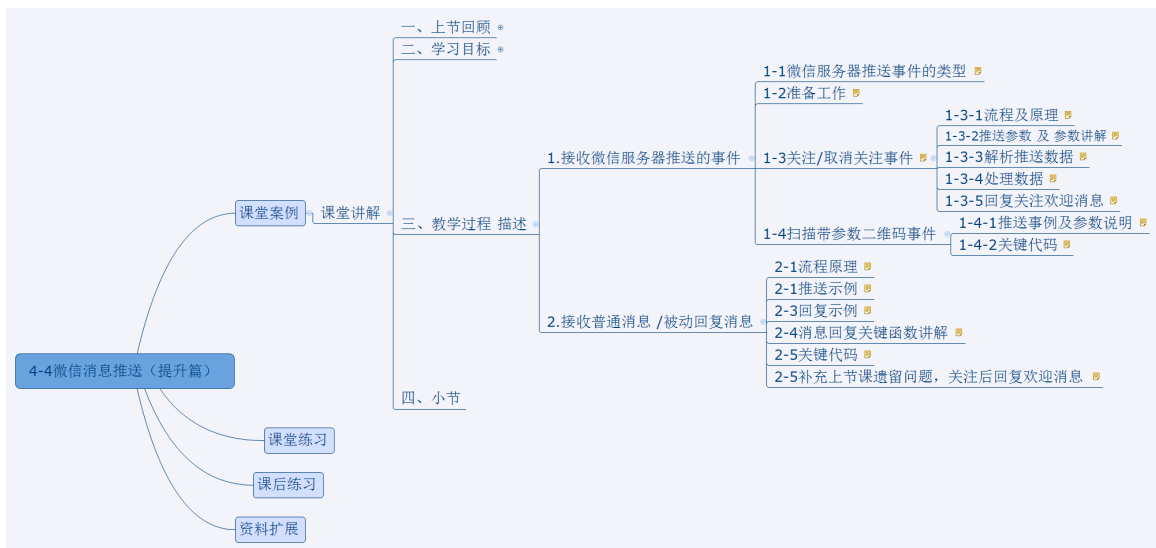


4-4 微信消息推送（提升篇）

4-4 微信消息推送（提升篇）	1
1. 课堂案例	2
课堂讲解	2
一、上节回顾	2
接入微信公众平台	2
二、学习目标	2
实现事件接收	2
实现文本消息回复	2
实现图文消息回复	2
三、教学过程 描述	2
1.接收微信服务器推送的事件	2
2.接收普通消息 /被动回复消息	15
四、小节	23
2. 课堂练习	23
3. 课后练习	23
4. 资料扩展	23



1. 课堂案例

课堂讲解

一、上节回顾

接入微信公众平台

二、学习目标

实现事件接收

实现文本消息回复

实现图文消息回复

三、教学过程 描述

1.接收微信服务器推送的事件

1-1微信服务器推送事件的类型

1 关注/取消关注事件

2 扫描带参数二维码事件

3 上报地理位置事件

4 自定义菜单事件

5 点击菜单拉取消息时的事件推送

6 点击菜单跳转链接时的事件推送

1-2准备工作

微信公众平台数据是怎么推送的？

以POST数据包的方式推送，获取推送内容：file_get_contents("php://input")

微信公众平台推送的内容是什么样的格式？

内容格式： 是以xml格式字符串的方式推送

如何解析微信公众平台推送的XML格式字符串？

解析函数：simplexml_load_string(); 返回一个对象

为了调试，我们准备了两个数据表(wixin_log, weixin_xml)分别用以保存服务器返回的错误日志、和微信服务器推送过来的数据，另外准备了两个函数，分别用来保存微信服务器的返回的错误日志和推送的数据

```

//推送数据表
CREATE TABLE `app_weixin_xml` (
  `id` int(10) NOT NULL AUTO_INCREMENT,
  `xml` text COMMENT '推送的代码',
  PRIMARY KEY (`id`)
) ENGINE=MyISAM AUTO_INCREMENT=27 DEFAULT CHARSET=utf8;

//错误日志表
CREATE TABLE `app_weixin_log` (
  `id` int(10) NOT NULL AUTO_INCREMENT,
  `errcode` int(10) DEFAULT '0' COMMENT '错误码',
  `errmsg` varchar(255) DEFAULT '' COMMENT '错误消息',
  `createtime` int(10) DEFAULT '0' COMMENT '错误创建时间',
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COMMENT='微信开发错误记录表';

```

两个函数

```

/**
 * 添加微信错误日志
 * @param inter $errcode 错误码
 * @param string $errmsg 错误消息
 */
public function writeLog($errcode,$errmsg){
    $weixin_log = M('WeixinLog');
    $data = array(
        'errcode' => $errcode,
        'errmsg' => $errmsg,
        'createtime'=> time()
    );
    $weixin_log->add($data);
}

//保存数据
public function saveXml(){
    $str = file_get_contents("php://input");
    M('WeixinXml')->add(array('xml'=>$str));
}

```

另外准备两个使用 CURL 模拟请求发送函数

```
public 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); //启用时会把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;
        }
    }
}
```

```

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 400:
                break;
        }
    }
}

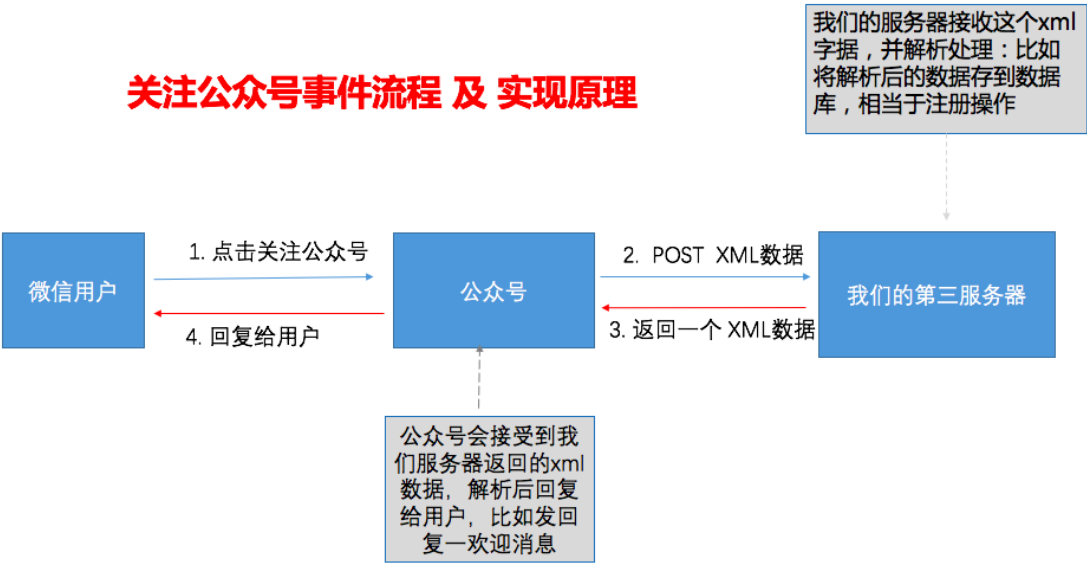
```

1-3关注/取消关注事件

步骤:

- 1.用户点击关注按钮
- 2.公众平台推送XML数据到我们填写的URL地址
3. 我们的服务器获取微信公众平台推送的post数据
2. 解析获取的数据
3. 处理数据
4. 返回一个欢迎消息给用户

1-3-1流程及原理



1-3-2推送参数 及 参数讲解

推送XML数据包示例：

```
<xml>
<ToUserName><![CDATA[toUser]]></ToUserName>
<FromUserName><![CDATA[FromUser]]></FromUserName>
<CreateTime>123456789</CreateTime>
<MsgType><![CDATA[event]]></MsgType>
<Event><![CDATA[subscribe]]></Event>
</xml>
```

参数说明：

参数	描述
ToUserName	开发者微信号
FromUserName	发送方帐号（一个OpenID）
CreateTime	消息创建时间 （整型）
MsgType	消息类型，event
Event	事件类型，subscribe(订阅)、unsubscribe(取消订阅)

1-3-3解析推送数据

```
<?php
$postxml = file_get_contents("php://input"); //获取公众平台推送的xml数据
$obj_xml = simplexml_load_string($postxml); //解析xml数据为一个数据对象
```

1-3-4处理数据

当发生关注事件, 我们将关注者的OPENID写到数据库

1. 创建表

```
//微信公众号用户表
CREATE TABLE `demo`.`app_weixin_user` (
  `id` int(10) NOT NULL AUTO_INCREMENT,
  `openid` varchar(50) CHARACTER SET utf8 NOT NULL DEFAULT '' COMMENT '微信用户openid',
  `create_time` int(10) NOT NULL DEFAULT 0 COMMENT '关注时间',
  PRIMARY KEY (`id`)
) COMMENT='微信用户关注公众号 记录表';|
```

当发生取消关注事件, 我们将关注者的OPENID从数据库删除

关键代码:


```

//获取公众号推送的post数据
$postxml = file_get_contents("php://input");
//解析xml数据 为 对象
$objxml = simplexml_load_string($postxml);
/*
将xml数据对象 转为数组
*/
$objxml = (array)$objxml;
$arrxml = array();
foreach ($objxml as $key => $value) {
    $arrxml[$key] = (string)$value;
}
//获取消息类型
$msgType= $arrxml['MsgType'];

if($msgType == 'event'){
    //获取事件类型
    $event_name = $arrxml['Event'];
    //关注事件处理
    if($event_name == 'subscribe'){
        //获取数据
        $server_uid = $arrxml['ToUserName']; //公众号ID
        $user_uid = $arrxml['FromUserName']; //微信ID
        $createtime = $arrxml['CreateTime']; //关注时间

        //保存关注用户的openid到数据库
        $weixin_user = M('WeixinUser');
        $arr = array();
        $arr['openid'] = $user_uid;
        $arr['create_time'] = $createtime;

        $weixin_user->where(array('openid'=>$user_uid))->delete(); //删除已存在的用户
        $weixin_user->add($arr); //添加用户
        echo ""; //回复一个空串给微信服务器
    }

    //取消关注事件处理
    if($event_name == 'unsubscribe'){
        $user_uid = $arrxml['FromUserName'];
        $weixin_user = M('WeixinUser');
        $weixin_user->where(array('openid'=>$user_uid))->delete(); //删除已关注的用户
        echo ""; //回复一个空串给微信服务器
    }
}
}

```

1-3-5回复关注欢迎消息

我们回复一个空字符串表示我们接收到了这个推送，下节课我们学完消息回复后，我们一起来补充下，如何回复一个欢迎消息

1-4扫描带参数二维码事件

1-4-1推送事例及参数说明

1. 用户未关注时, 进行关注后的事件推送

推送XML数据包示例:

```
<xml><ToUserName><![CDATA[toUser]]></ToUserName>
<FromUserName><![CDATA[FromUser]]></FromUserName>
<CreateTime>123456789</CreateTime>
<MsgType><![CDATA[event]]></MsgType>
<Event><![CDATA[subscribe]]></Event>
<EventKey><![CDATA[qrscene_123123]]></EventKey>
<Ticket><![CDATA[TICKET]]></Ticket>
</xml>
```

参数说明:

参数	描述
ToUserName	开发者微信号
FromUserName	发送方帐号（一个OpenID）
CreateTime	消息创建时间（整型）
MsgType	消息类型，event
Event	事件类型，subscribe
EventKey	事件KEY值，qrscene_为前缀，后面为二维码的参数值
Ticket	二维码的ticket，可用来换取二维码图片

备注：从上面的示例可以看出，这个推送数据跟 关注事件的差别是，只多了两个参数：一个是 EventKey 另一个是 Ticket 因此我们可以将此事件合并到 关注事件

2. 用户已关注时的事件推送

推送XML数据包示例:

```
<xml>
<ToUserName><![CDATA[toUser]]></ToUserName>
<FromUserName><![CDATA[FromUser]]></FromUserName>
<CreateTime>123456789</CreateTime>
<MsgType><![CDATA[event]]></MsgType>
<Event><![CDATA[SCAN]]></Event>
```

```
<EventKey><![CDATA[SCENE_VALUE]]></EventKey>
<Ticket><![CDATA[TICKET]]></Ticket>
</xml>
```

参数说明:

参数	描述
ToUserName	开发者微信号
FromUserName	发送方帐号（一个OpenID）
CreateTime	消息创建时间（整型）
MsgType	消息类型，event
Event	事件类型，SCAN
EventKey	事件KEY值，是一个32位无符号整数，即创建二维码时的二维码scene_id
Ticket	二维码的ticket，可用来换取二维码图片

000

1-4-2 关键代码

我们先给用户数据表添加几个字段，用于存扫描事件的参数

```
//修改用户表
ALTER TABLE `demo`.`app_weixin_user`
  ADD COLUMN `eventkey` varchar(100) CHARACTER SET utf8 DEFAULT '' COMMENT '扫描事件KEY值' AFTER `create_time`,
  ADD COLUMN `ticket` varchar(100) CHARACTER SET utf8 DEFAULT '' COMMENT '二维码Ticket 值 可用来换取二维码图片' AFTER `eventkey`;
```

我提前准备了一张带场景值的二维码图片



关键代码：

1».未关注的扫描关键代码

```

//获取公众号推送的post数据
$postxml = file_get_contents("php://input");
//解析xml数据 为 对象
$objxml = simplexml_load_string($postxml);
/*
将xml数据对象 转为数组
*/
$objxml = (array)$objxml;
$arrxml = array();
foreach ($objxml as $key => $value) {
    $arrxml[$key] = (string)$value;
}
//获取消息类型
$msgType= $arrxml['MsgType'];

if($msgType == 'event'){
    //获取事件类型
    $event_name = $arrxml['Event'];
    //关注事件处理
    if($event_name == 'subscribe'){
        //获取数据
        $server_uid = $arrxml['ToUserName']; //公众号ID
        $user_uid = $arrxml['FromUserName']; //微信ID
        $createtime = $arrxml['CreateTime']; //关注时间

        //保存关注用户的openid到数据库
        $weixin_user = M('WeixinUser');
        $arr = array();
        $arr['openid'] = $user_uid;
        $arr['create_time'] = $createtime;

        //扫描参数
        if($arrxml['EventKey']){
            $arr['eventkey'] = $arrxml['EventKey']; //场景值
            $arr['ticket'] = $arrxml['Ticket']; //二维码ticket值
        }

        $weixin_user->where(array('openid'=>$user_uid))->delete(); //删除已存在的用户
        $weixin_user->add($arr); //添加用户
        echo ""; //回复一个空串给微信服务器
    }

    //取消关注事件处理
    if($event_name == 'unsubscribe'){
        $user_uid = $arrxml['FromUserName'];
        $weixin_user = M('WeixinUser');
        $weixin_user->where(array('openid'=>$user_uid))->delete(); //删除已关注的用户
        echo ""; //回复一个空串给微信服务器
    }
}
}

```

2》已关注的扫描关键代码

这是一张带新场景值的二维码图片



关键代码

```

if($msgType == 'event'){
    //获取事件类型
    $event_name = $arr_xml['Event'];
    //关注事件处理
    if($event_name == 'subscribe'){
        //获取数据
        $server_uid = $arr_xml['ToUserName']; //公众号ID
        $user_uid = $arr_xml['FromUserName']; //微信ID
        $createtime = $arr_xml['CreateTime']; //关注时间

        //保存关注用户的openid到数据库
        $weixin_user = M('WeixinUser');
        $arr = array();
        $arr['openid'] = $user_uid;
        $arr['create_time'] = $createtime;

        //扫描参数
        if($arr_xml['EventKey']){
            $arr['eventkey'] = $arr_xml['EventKey']; //场景值
            $arr['ticket'] = $arr_xml['Ticket']; //二维码ticket值
        }

        $weixin_user->where(array('openid'=>$user_uid))->delete(); //删除已存在的用户
        $weixin_user->add($arr); //添加用户
        echo ""; //回复一个空串给微信服务器
    }

    //取消关注事件处理
    if($event_name == 'unsubscribe'){
        $user_uid = $arr_xml['FromUserName'];
        $weixin_user = M('WeixinUser');
        $weixin_user->where(array('openid'=>$user_uid))->delete(); //删除已关注的用户
        echo ""; //回复一个空串给微信服务器
    }

    //扫描事件(已关注的前提下扫描)
    if($event_name == 'SCAN'){
        //更新场景值
        $user_uid = $arr_xml['FromUserName']; //微信ID
        $createtime = $arr_xml['CreateTime']; //关注时间
        $eventkey = $arr_xml['EventKey']; //场景值
        $ticket = $arr_xml['Ticket']; //二维码ticket值

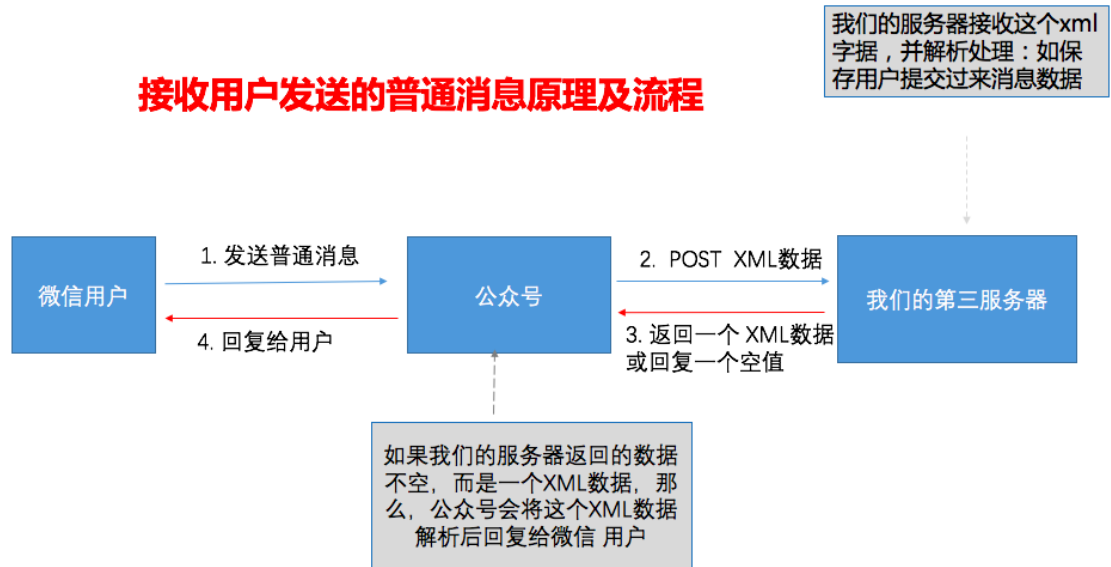
        $weixin_user = M('WeixinUser');
        //个性数据
        $arr = array(
            'openid'=>$user_uid,
            'createtime'=>$createtime,
            'eventkey'=>$eventkey,
            'ticket'=>$ticket
        );
        $weixin_user->where(array('openid'=>$user_uid))->save($arr);
        echo "";
    }
}

```

2.接收普通消息/被动回复消息

2-1流程原理

接收用户发送的普通消息原理及流程



2-1推送示例

微信服务器推送给我们的第三方服务器的POST数据格式：

1. 文本消息

```
<xml>
<ToUserName><![CDATA[toUser]]></ToUserName>
<FromUserName><![CDATA[fromUser]]></FromUserName>
<CreateTime>1348831860</CreateTime>
<MsgType><![CDATA[text]]></MsgType>
<Content><![CDATA[this is a test]]></Content>
<MsgId>1234567890123456</MsgId>
</xml>
```

参数

参数	描述
ToUserName	开发者微信号
FromUserName	发送方帐号（一个OpenID）
CreateTime	消息创建时间（整型）
MsgType	text
Content	文本消息内容
MsgId	消息id，64位整型

2.图片消息

```
<xml>
<ToUserName><![CDATA[toUser]]></ToUserName>
<FromUserName><![CDATA[fromUser]]></FromUserName>
<CreateTime>1348831860</CreateTime>
<MsgType><![CDATA[image]]></MsgType>
<PicUrl><![CDATA[this is a url]]></PicUrl>
<MediaId><![CDATA[media_id]]></MediaId>
<MsgId>1234567890123456</MsgId>
</xml>
```

参数	描述
ToUserName	开发者微信号
FromUserName	发送方帐号（一个OpenID）
CreateTime	消息创建时间（整型）
MsgType	image
PicUrl	图片链接（由系统生成）
MediaId	图片消息媒体id，可以调用多媒体文件下载接口拉取数据。
MsgId	消息id，64位整型

2-3回复示例

我们的第三方服务器回复给微信服务器的示例

1.回复文本消息

```

<xml>
<ToUserName><![CDATA[toUser]]></ToUserName>
<FromUserName><![CDATA[fromUser]]></FromUserName>
<CreateTime>12345678</CreateTime>
<MsgType><![CDATA[text]]></MsgType>
<Content><![CDATA[你好]]></Content>
</xml>

```

参数说明

参数	是否必须	描述
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	text
Content	是	回复的消息内容（换行：在content中能够换行，微信客户端就支持换行显示）

2.回复 图片消息

```

<xml>
<ToUserName><![CDATA[toUser]]></ToUserName>
<FromUserName><![CDATA[fromUser]]></FromUserName>
<CreateTime>12345678</CreateTime>
<MsgType><![CDATA[image]]></MsgType>
<Image>
  <MediaId><![CDATA[media_id]]></MediaId>
</Image>
</xml>

```

参数说明

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	image
MediaId	是	通过素材管理中的接口上传多媒体文件，得到的id。

2-4消息回复关键函数讲解

`sprintf(format,arg1,arg2,arg++)`

format 参数详解

参数必需。规定字符串以及如何格式化其中的变量。

可能的格式值：

%% - 返回一个百分号 %

%b - 二进制数

%c - ASCII 值对应的字符

%d - 包含正负号的十进制数(负数、0、正数)

%e - 使用小写的科学计数法(例如 1.2e+2)

%E - 使用大写的科学计数法(例如 1.2E+2)

%u - 不包含正负号的十进制数(大于等于 0)

%f - 浮点数(本地设置)

%F - 浮点数(非本地设置)

%g - 较短的 %e 和 %f

%G - 较短的 %E 和 %f

%o - 八进制数

%s - 字符串

%x - 十六进制数(小写字母)

%X - 十六进制数(大写字母)

附加的格式值。必需放置在 % 和字母之间(例如 %.2f)：

+(在数字前面加上 + 或 - 来定义数字的正负性。默认情况下，只有负数才做标记，正数不做标记)

' (规定使用什么作为填充，默认是空格。它必须与宽度指定器一起使用。例如：%x20s(使用 "x" 作为填充))

-(左调整变量值)

[0-9] (规定变量值的最小宽度)

. [0-9] (规定小数位数或最大字符串长度)

注释: 如果使用多个上述的格式值, 它们必须按照以上顺序使用。

arg1 必需。规定插到 format 字符串中第一个 % 符号处的参数。

arg2 可选。规定插到 format 字符串中第二个 % 符号处的参数。

arg++ 可选。规定插到 format 字符串中第三、四等 % 符号处的参数。

2-5 关键代码

案例1: 我们实现的一个功能是收集用户发送到公众平台的文本消息, 然后回复 hello + 消息内容
如: 用户发送: word 到公众平台, 我们回复 hello word

步骤:

1. 接收微信服务器推送到我们填写的URL地址XML数据
2. 解析XML数据并做处理
3. 回复XML数据到微信服务器

案例1 关键代码:

```

//获取公众号推送的post数据
$postxml = file_get_contents("php://input");
//解析xml数据
$objxml = simplexml_load_string($postxml);
//将xml数据对象转为数组
$objxml = (array)$objxml;
$arrxml = array();
foreach ($objxml as $key => $value) {
    $arrxml[$key] = (string)$value;
}
$arrxml = $this->arr_xml;

//消息类型
$msgType= $arrxml['MsgType'];

//如果是文本消息
if($msgType == 'text'){
    /*
    步骤
    1. 取得xml数据，上面的 $arr_xml 就是我们获取的数据
    2. 取得用户发送的文件内容
    3. 回复XML数据到微信服务器
    */

    //2.取得用户发送的内容
    $content = $arrxml['Content']; //用户发送的消息
    //取得回复消息需要的参数
    $server_uid = $arrxml['ToUserName']; //公众号ID
    $user_uid = $arrxml['FromUserName']; //微信ID
    $curr_time = time(); //当前时间
    $return_content = "Hello ".$content; //回复给用户的消息内容
    //回复给微信服务器的xml数据格式 【文本消息】
    $str = "<xml>
        <ToUserName><![CDATA[%s]]></ToUserName>
        <FromUserName><![CDATA[%s]]></FromUserName>
        <CreateTime>%d</CreateTime>
        <MsgType><![CDATA[text]]></MsgType>
        <Content><![CDATA[%s]]></Content>
    </xml>";

    //3. 回复xml数据给微信服务器
    echo sprintf($str,$user_uid,$server_uid,$curr_time,$return_content);
}

```

案例2:

我们实现的一个功能是，收集用户发送到公众平台的图片消息，然后又获取到的图片，原路返回

步骤:

1. 接收微信服务器推送到我们填写的URL地址XML数据
2. 解析XML数据并做处理

3. 回复XML数据到微信服务器

案例2关键代码

```
//获取公众号推送的post数据
$postxml = file_get_contents("php://input");
//解析xml数据
$objxml = simplexml_load_string($postxml);
//将xml数据对象转为数组
$objxml = (array)$objxml;
$arrxml = array();
foreach ($objxml as $key => $value) {
    $arrxml[$key] = (string)$value;
}
$arrxml = $this->arrxml;

//消息类型
$msgType= $arrxml['MsgType'];

//如果是文本消息
if($msgType == 'image'){
    /*
    步骤
    1. 取得xml数据，上面的 $arrxml 就是我们获取的数据
    2. 取得用户发送的图片
    3. 回复XML数据到微信服务器
    */

    //2. 取得用户发送的图片信息
    $mediaid = $arrxml['MediaId']; //图片id
    //取得回复消息需要的参数
    $server_uid = $arrxml['ToUserName']; //公众号ID
    $user_uid = $arrxml['FromUserName']; //微信ID
    $curr_time = time(); //当前时间
    //回复给微信服务器的xml数据格式 【图片消息】
    $str = "<xml>
    <ToUserName><![CDATA[%s]]></ToUserName>
    <FromUserName><![CDATA[%s]]></FromUserName>
    <CreateTime>%d</CreateTime>
    <MsgType><![CDATA[image]]></MsgType>
    <Image>
    <MediaId><![CDATA[%s]]></MediaId>
    </Image>
    </xml>";

    //3. 回复xml数据给微信服务器
    echo sprintf($str,$user_uid,$server_uid,$curr_time,$mediaid);
}
```

2-5补充上节课遗留问题，关注后回复欢迎消息

关键代码:

```

//获取数据
$server_uid = $arr_xml['ToUserName']; //公众号ID
$user_uid = $arr_xml['FromUserName']; //微信ID
$createtime = $arr_xml['CreateTime']; //关注时间

//保存关注用户的openid到数据库
$weixin_user = M('WeixinUser');
$arr = array();
$arr['openid'] = $user_uid;
$arr['create_time'] = $createtime;

//扫描参数
if($arr_xml['EventKey']){
    $arr['eventkey'] = $arr_xml['EventKey']; //场景值
    $arr['ticket'] = $arr_xml['Ticket']; //二维码ticket值
}

$weixin_user->where(array('openid'=>$user_uid))->delete(); //删除已存在的用户
$weixin_user->add($arr); //添加用户
//echo ""; //回复一个空串给微信服务器 不再回复空字符
//
//回复一个欢迎消息
$str = "<xml>
    <ToUserName><![CDATA[%s]]></ToUserName>
    <FromUserName><![CDATA[%s]]></FromUserName>
    <CreateTime>%d</CreateTime>
    <MsgType><![CDATA[text]]></MsgType>
    <Content><![CDATA[%s]]></Content>
</xml>";

//3. 回复xml数据给微信服务器
echo sprintf($str,$user_uid,$server_uid,time(),'欢迎关注多迪公众平台');

```

四、小节

2. 课堂练习
3. 课后练习
4. 资料扩展