

第3章 开发模式

由于编辑模式下仅能实现内容固定的自动回复，这远没有发挥微信公众平台的功能。如果想实现更强大、更精准的个性化功能的服务，就需要用到开发模式。本章将介绍启用开发模式以及一个简单的自动回复功能。

3.1 启用开发模式

3.1.1 关闭编辑模式

登录微信公众平台后台，依次进入菜单“功能”→“高级功能”页面，可以看到两种模式：编辑模式和开发模式，以及两种模式当前的开启状态，如图 3-1 所示。



图 3-1 编辑模式和开发模式

点击“编辑模式”下方的“进入”链接，可以看到右上侧滑动开关处在开启状态，如图 3-2 所示。



图 3-2 编辑模式已开启

点击右侧的滑动开关，将弹出确认操作框，如图 3-3 所示。
点击“确定”按钮，关闭编辑模式，滑动开关将滑动到关闭一侧。

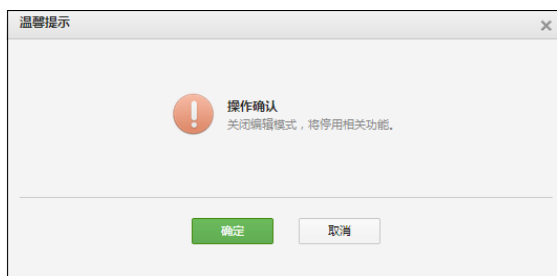


图 3-3 关闭操作确认

3.1.2 启用开发模式

点击开发模式页面，将看到尚未成为开发者的提示，如图 3-4 所示。

点击“成为开发者”按钮，将出现《微信公众平台开发者服务协议》，勾选同意协议并且点击“下一步”按钮后，将出现接口配置信息页面，如图 3-5 所示。

在此处需要填写 URL 和 Token。其中 URL 是开发者用来接收微信服务器数据的接口 URL。Token 可任意填写，用作生成签名（该 Token 会和接口 URL 中包含的 Token 进行比对，从而验证安全性）。

在上一章中，我们创建了新浪云计算平台应用，其域名 URL 为 `http://cctv15.sinaapp.com/`，而 Token 在代码中定义为 `weixin`。购买的虚拟主机 URL 为 `http://txw1958.gotoip3.com/`，上传程序中 Token 也定义为 `weixin`，我们使用 SAE 的 URL 和 Token 填入，提交成功，如图 3-6 所示。



图 3-4 尚未成为开发者

请填写接口配置信息，此信息需要你拥有自己的服务器资源。
填写的URL需要正确响应微信发送的Token验证，请阅读[接入指南](#)。

URL

Token

[什么是Token?](#)

图 3-5 接口配置信息

☒ 你已成为开发者

服务器配置（用于接收用户信息） [修改](#)

URL	<code>http://cctv15.sinaapp.com/</code>
Token	<code>weixin</code>

图 3-6 成为开发者

此时开发模式还处于关闭状态，点击滑动开关，滑动开关将处在开启状态，开发模式已经开启，如图 3-7 所示。



图 3-7 开发模式已开启

至此，开发模式已成功启用。

3.1.3 常见问题与解决方案

当我们在提交 URL 和 Token 时，有时会碰到提交不成功的情况，具体有以下几种。

1. 请求 URL 超时

这种情况一般是由于服务器网速或响应速度太慢。可以重试几次或者等一段时间再来试，如果还是这样，则需要考虑更换速度更快、性能更好的服务器。

2. 系统发生错误，请稍后重试

这种情况一般是由于微信服务器短时内的异常引起的，一般重试或者过一段时间尝试即可。

3. 服务器没有正确响应 Token 验证，请阅读“消息接口使用指南”

这种情况需要具体分析验证过程卡在哪一个环节了。可以通过调用变量 `$_SERVER` 来获取服务器和执行环境信息来分析。

`$_SERVER` 是一个包含了诸如头信息 (header)、路径 (path) 以及脚本位置 (script location) 等信息的数组。这个数组中的项目由 Web 服务器创建。欲了解更多关于 `$_SERVER` 的信息，请访问官方网站：

<http://www.php.net/manual/zh/reserved.variables.server.php>

这里需要使用以下两个元素：

❑ `$_SERVER['REMOTE_ADDR']` (来访者的 IP 地址，此处为微信服务器的 IP)

❑ `$_SERVER['QUERY_STRING']` (查询请求字符串，此处为微信服务器发过来的 GET 请求字符串)

将以上两个变量记录到日志中。函数定义如下：

```
function traceHttp()
{
    $content = date('Y-m-d H:i:s')."\\nREMOTE_ADDR:".$_SERVER["REMOTE_ADDR"]."\\nQUERY_STRING:".$_SERVER["QUERY_STRING"]."\\n\\n";

    if (isset($_SERVER['HTTP_APPNAME'])){ //SAE
        sae_set_display_errors(false);
        sae_debug(trim($content));
        sae_set_display_errors(true);
    }else {
        $max_size = 100000;
        $log_filename = "log.xml";
        if(file_exists($log_filename) and (abs(filesize($log_filename)) > $max_size)){unlink($log_filename);}
        file_put_contents($log_filename, $content, FILE_APPEND);
    }
}
```

上面代码中，当环境为 SAE 时，使用 SAE 的调试函数 `sae_debug()` 将内容记

录到日志中。而在具有读写权限的空间下,使用 `file_put_contents()` 函数把字符串写入文件。

然后在程序的数据处理之前调用该函数,记录信息,代码如下所示:

```
define("TOKEN", "weixin");
traceHttp();
$wechatObj = new wechatCallbackapiTest();
if (isset($_GET['echostr'])) {
    $wechatObj->valid();
}else{
    $wechatObj->responseMsg();
}
```

当提交 URL 和 Token 验证时,程序目录下应当生成一个 `log.txt` 文件。内容如下:

```
2014-01-10 11:03:21
REMOTE_ADDR:101.226.61.144
QUERY_STRING:signature=6e35c6f3d3279338781047dbffd09426b9ecdee3&echostr
=5979420653038092664&timestamp=1392001400&nonce=1392192345
```

综上可以得出初步结论:

- ❑ 如果没有生成日志文件:微信服务器没有访问你的服务器,需要先检查一下你的服务器是否可以通过公网访问,以及 URL 路径是否正确并且可以访问,如果公网能访问而微信服务器不能访问,那可能是防火墙拦截了 80 端口或微信服务器 IP,也可能服务器所在区域与微信服务器通信不畅,此时需要更换服务器。
- ❑ 如果已经生成日志文件:查看 `REMOTE_ADDR` 和 `QUERY_STRING` 内容是否与上述类似。确认 `signature`、`timestamp`、`nonce`、`echostr` 四个参数都有值。如果这些都没有问题,则检查程序中定义的 Token 值是否和提交的一致,然后检查程序流程及数据处理是否与官方文档描述的一致。

3.2 实现自动回复

在前面的例子中,我们已经嵌入了一个简单的时间查询功能,发送一个问号“?”就能回复当前的时间,如图 3-8 所示。

至此,你的微信公众平台账号已经成功使用了开发模式,并且已经能够实现自动回复了。

3.3 消息交互原理分析

接下来结合上一章的代码来分析一下微信公众平台的消息交互原理。下面的代码基于微信公众平台官

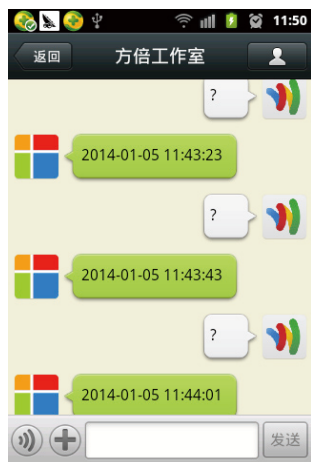


图 3-8 自动回复时间

方示例代码修改而成。

```
1 <?php
2 /*
3     方倍工作室 http://www.cnblogs.com/txw1958/
4     CopyRight 2013 www.doucube.com All Rights Reserved
5 */
6
7 define("TOKEN", "weixin");
8 $wechatObj = new wechatCallbackapiTest();
9 if (isset($_GET['echostr'])) {
10     $wechatObj->valid();
11 }else{
12     $wechatObj->responseMsg();
13 }
14
15 class wechatCallbackapiTest
16 {
17     public function valid()
18     {
19         $echoStr = $_GET["echostr"];
20         if($this->checkSignature()){
21             echo $echoStr;
22             exit;
23         }
24     }
25
26     private function checkSignature()
27     {
28         $signature = $_GET["signature"];
29         $timestamp = $_GET["timestamp"];
30         $nonce = $_GET["nonce"];
31
32         $token = TOKEN;
33         $tmpArr = array($token, $timestamp, $nonce);
34         sort($tmpArr);
35         $tmpStr = implode( $tmpArr );
36         $tmpStr = sha1( $tmpStr );
37
38         if( $tmpStr == $signature ){
39             return true;
40         }else{
41             return false;
42         }
43     }
44
45     public function responseMsg()
46     {
47         $postStr = $GLOBALS["HTTP_RAW_POST_DATA"];
48
49         if (!empty($postStr)){
50             $postObj = simplexml_load_string($postStr, 'SimpleXMLElement',
LIBXML_NOCDATA);
51             $fromUsername = $postObj->FromUserName;
52             $toUsername = $postObj->ToUserName;
53             $keyword = trim($postObj->Content);
```

```
54         $time = time();
55         $textTpl = "<xml>
56             <ToUserName><![CDATA[%s]]></ToUserName>
57             <FromUserName><![CDATA[%s]]></FromUserName>
58             <CreateTime>%s</CreateTime>
59             <MsgType><![CDATA[%s]]></MsgType>
60             <Content><![CDATA[%s]]></Content>
61             <FuncFlag>0</FuncFlag>
62             </xml>";
63         if($keyword == "?" || $keyword == "? ")
64         {
65             $msgType = "text";
66             $contentStr = date("Y-m-d H:i:s",time());
67             $resultStr = sprintf($textTpl, $fromUsername, $toUsername,
68             $time, $msgType, $contentStr);
69             echo $resultStr;
70         }
71         }else{
72             echo "";
73             exit;
74         }
75     }
76 }>
```

首先看一下代码的结构。

第2行~第5行：注释部分。

第7行：使用 define() 函数定义常量，常量名称为 TOKEN，常量的值为 weixin，这个值就是在启用开发模式时填写的 Token。

第15行~第75行：定义了一个类 wechatCallbackapiTest，并在类中定义了3个方法 valid()、checkSignature() 和 responseMsg()。

第8行~第13行：程序执行语句。第8行实例化了一个类对象。在第9行中，判断 GET 请求是否有 echostr 变量，如果有则执行 valid() 方法，否则执行 responseMsg() 方法。

其次分析微信消息交互流程。

在提交 URL 和 Token 申请验证时，微信服务器将发送 GET 请求到填写的 URL 上，并且带上4个参数（signature、timestamp、nonce 和 echostr），GET 请求如下：

```
signature=6e35c6f3d3279338781047dbffd09426b9ecdee3&echostr=597942065303
8092664&timestamp=1392001400&nonce=1392192345
```

上述请求参数说明如表 3-1 所示。

表 3-1 请求校验参数说明

参数	描述
signature	微信加密签名，signature 结合了开发者填写的 token 参数和请求中的 timestamp 参数、nonce 参数
timestamp	时间戳
nonce	随机数
echostr	随机字符串

由于这个 GET 请求包含 echostr 变量, 所以执行 valid() 方法, 在该方法中, 又调用了校验签名方法 checkSignature() 方法。如果签名校验为真, 则原样输出变量 \$echoStr 的值。

加密 / 校验流程如下:

- 1) 第 33 行~第 34 行: 将 token、timestamp、nonce 三个参数进行字典序排序。
- 2) 第 35 行~第 36 行: 将三个参数字符串拼接成一个字符串进行 sha1 加密。
- 3) 第 38 行~第 42 行: 开发者获得加密后的字符串可与 signature 对比, 标识该请求来源于微信。

而在发送问号的时候, 微信服务器将也会带上前面三个参数 (signature、timestamp 和 nonce) 访问开发者设置的 URL, 同时还会将消息的 XML 数据包 POST 到 URL 上。XML 格式如下:

```
<xml>
  <ToUserName><![CDATA[gh_ba6050bc0be7]]></ToUserName>
  <FromUserName><![CDATA[oDeOAjgSJUX10wvImSRMSwmyQAYa]]></FromUserName>
  <CreateTime>1392043637</CreateTime>
  <MsgType><![CDATA[text]]></MsgType>
  <Content><![CDATA[?]]></Content>
  <MsgId>5978781895719912033</MsgId>
</xml>
```

而消息请求不包含 echostr 变量, 所以将执行响应消息 responseMsg() 方法。

第 47 行: 响应消息方法首先接收上述原始 POST 数据。

第 50 行: 然后将数据载入对象中, 对象名为 SimpleXMLElement, LIBXML_NOCDATA 表示将 CDATA 合并为文本节点。

第 51 行~第 54 行: 取得 XML 类对象的值, 并赋给新的变量, 注意发送方变为接收方, 接收方变为发送方。

第 55 行~第 62 行: 构造要回复的 XML 数据包。

第 63 行: 判断发送过来的关键字是不是问号。

第 64 行~第 65 行: 设置回复的消息类型为 text, 内容为当前年月日时分秒。

第 66 行~第 67 行: 封装回复的 XML 数据包, 并且向微信服务器输出。XML 格式如下所示:

```
<xml>
  <ToUserName><![CDATA[oDeOAjgSJUX10wvImSRMSwmyQAYa]]></ToUserName>
  <FromUserName><![CDATA[gh_ba6050bc0be7]]></FromUserName>
  <CreateTime>1392043638</CreateTime>
  <MsgType><![CDATA[text]]></MsgType>
  <Content><![CDATA[2014-01-05 11:43:23]]></Content>
</xml>
```

这样用户就会收到回复的消息, 效果如图 3-8 所示。