# 微信公众号开发\_03

# 今日目标6

* 能够理解用户微信二维码的应用场景

作用：获取关注者的来源

场景：

推广

3级分销

加盟

活动

* 能够使用php代码获取微信临时与永久二维码

临时：30天

永久 10万

3步走

1、ticket

2、用ticket得到图片资源

3、把图片资源保存到服务器中

* 能够使用微信LBS接口实现相关功能

lbs 位置服务

1、在后台开启获取位置 5s上报一次

2、在被动接收事件中把位置保存到数据库中

3、在开放平台中注册账号，用开放平台提供接口，实现我们公众号中位置服务器功能

* 能够使用php代码调用微信语音识别接口

1、公众平台开启语音识别

2、接收接口关于msgType voice中就是多一个节点 Recognition 转义后的文字

* 能够了解网页授权开发流程

oauth2.0

0、在后台设置授权url地址

1、生成跳转得到code的url地址

2、通过get得到code

3、用code进行网页授权 access\_token和openid

4、可选，如果userinfo我们还可以得到用户的基本信息

* 能够掌握JSSDK的开发步骤

1、在后台设置授权url地址

2、引入js类库

3、生成签名

4、进行前台config配置

5、成功后就可以进行相关的业务实现

# 一、生成场景二维码.

## 1.1、场景二维概述

为了满足用户渠道推广分析和用户帐号绑定等场景的需要，公众平台提供了生成带参数二维码的接口。使用该接口可以获得多个带不同场景值的二维码，用户扫描后，公众号可以接收到事件推送。

**场景二维码作用：**

解决关注时来源问题

**使用的场景：**

* + 加盟
  + 活动场景
  + 分销系统

目前有2种类型的二维码：

* 临时二维码，是有过期时间的，最长可以设置为在二维码生成后的30天（即2592000秒）后过期，但能够生成较多数量。临时二维码主要用于帐号绑定等不要求二维码永久保存的业务场景
* 永久二维码，是无过期时间的，但数量较少（目前为最多10万个）。永久二维码主要用于适用于帐号绑定、用户来源统计等场景。

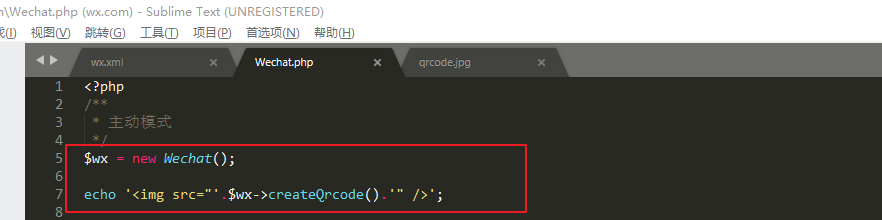
## 1.2、生成临时和永久二维码

<https://mp.weixin.qq.com/wiki?t=resource/res_main&id=mp1443433542>

步骤：

* 获取ticket(凭据)
* 通过ticket(凭据)换取二维码图片资源
* 保存二维码图片资源到本机



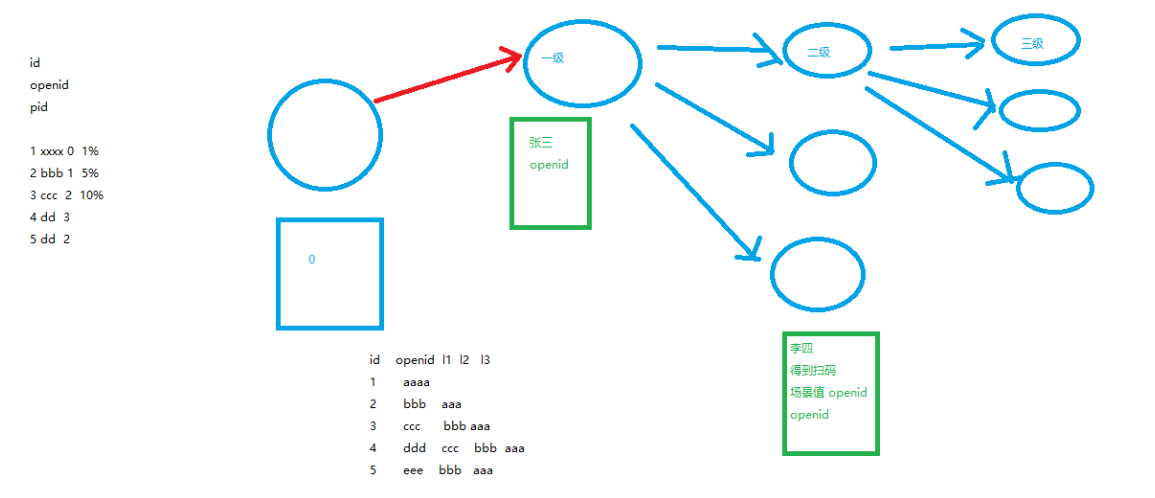


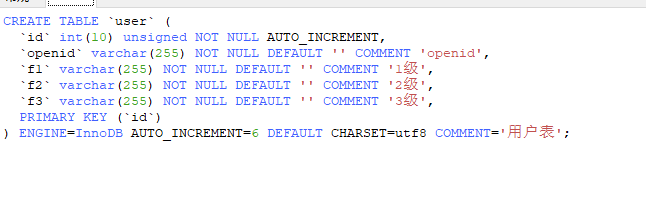




## 1.3、三级分销

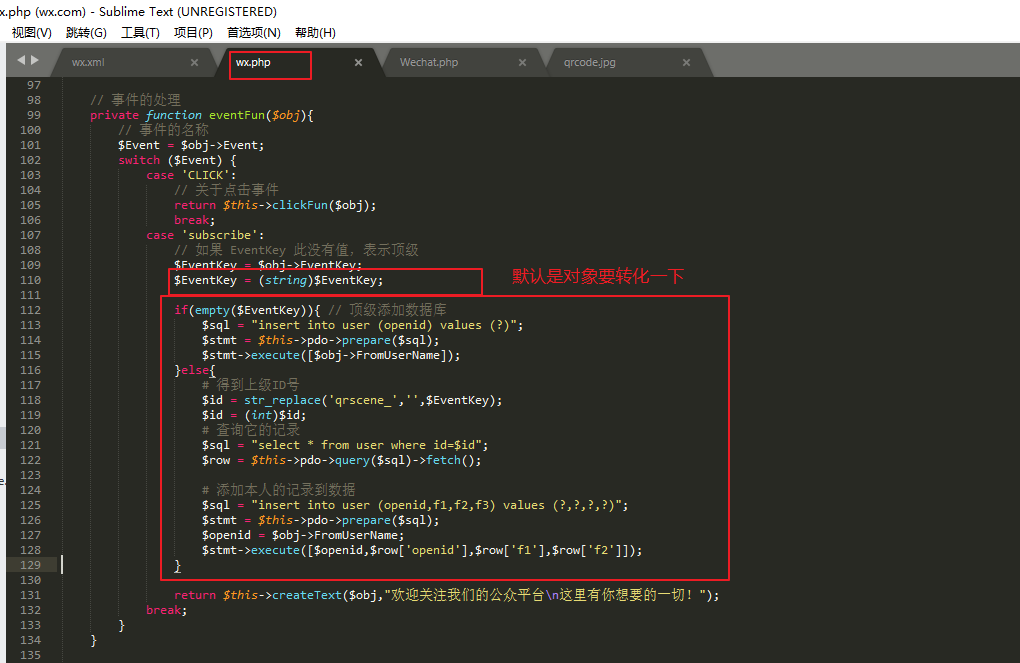
表设计



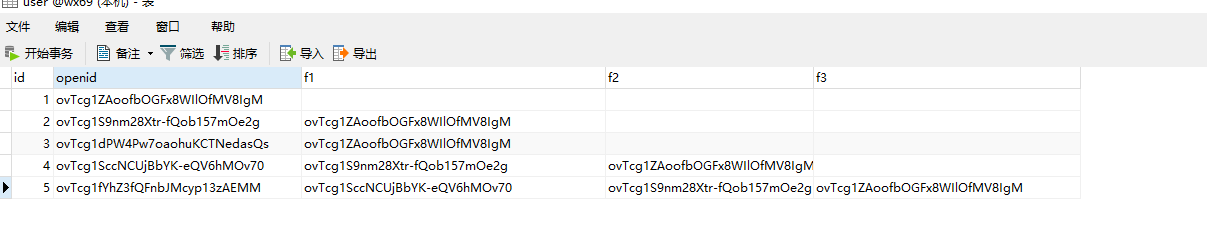


代码：

用户新关注时，入库，根据场景值来进行入库



数据库数据



# 二、微信语音识别

就在关注者在公众号里面说一句话语音，公众号接收到用户的语音并返回对应的文字。

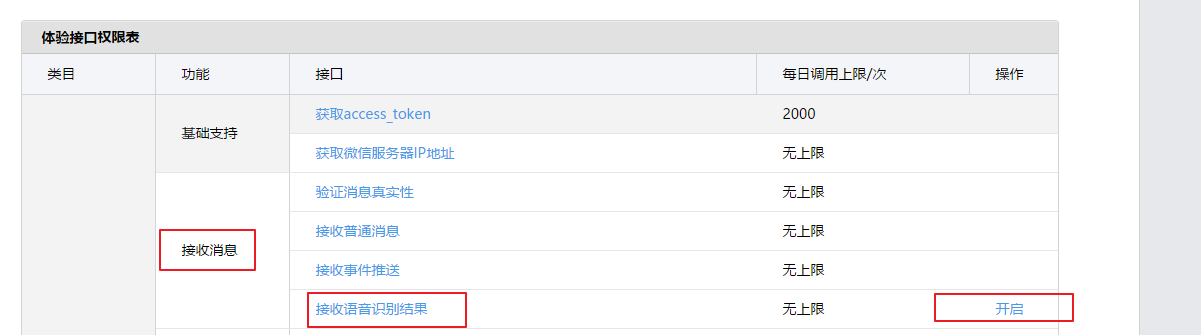
据测试，微信语音识别功能，准确率还是挺好的，当然针对的是普通话。

## 2.1、微信语音识别开启

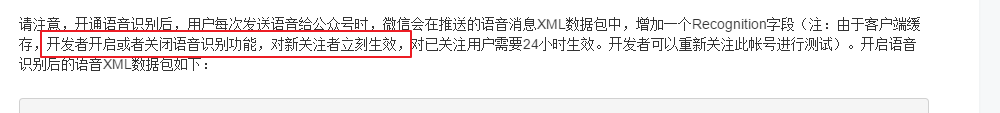
公众号后台对应的api列表中，点击相应的按钮，来开启语音识别功能。

注：对新关注者，立刻生效，对已关注者，24小时之内生效，对已关注都取消关注再次关注，也需要等待。

开启微信公众号的语音识别：

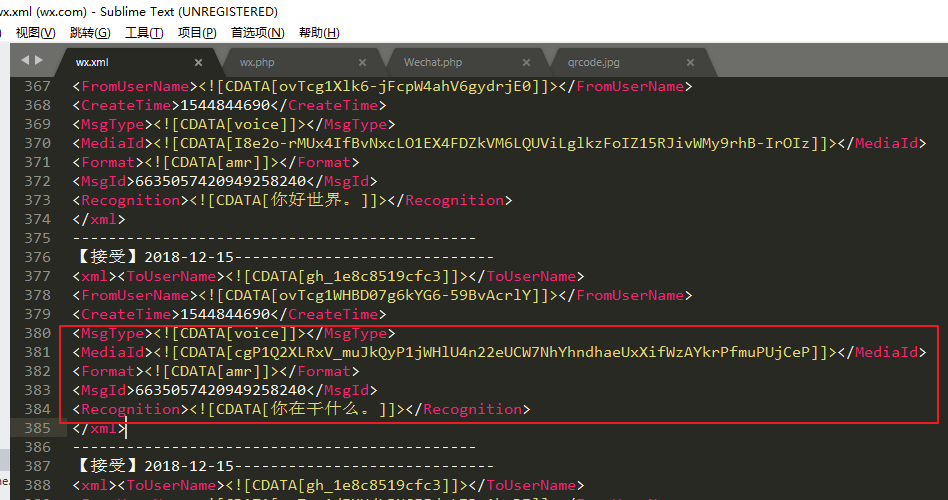
 

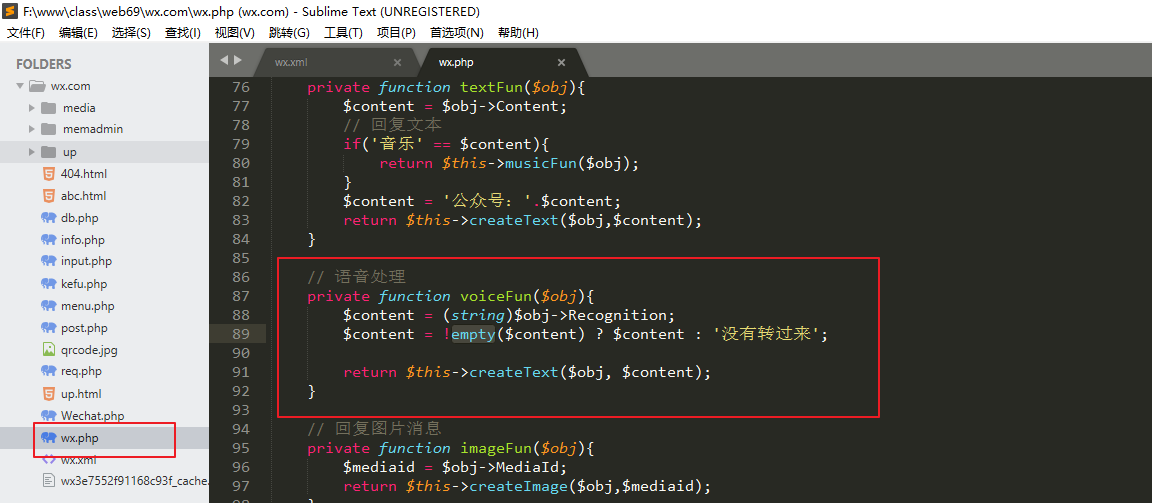
## 2.2、接受语音识别结果

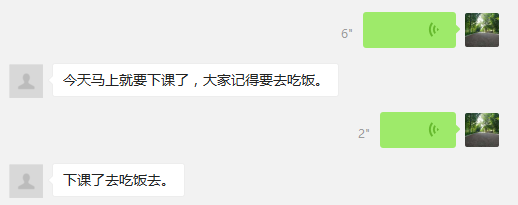


如果开启了语音识别结果，则在原有的语音接受消息 voice 事件中的xml节点将多一个节点: Recognition









# 三、微信LBS

LBS：基于**位置的服务**

它是通过网络定位或是外部定位方式(如GPS)获取终端用户的位置信息(经纬度)，根据位置(经纬度)为用户提供相应服务的一种增值业务。

## 3.1、微信LBS开启

开启微信获取用户位置服务

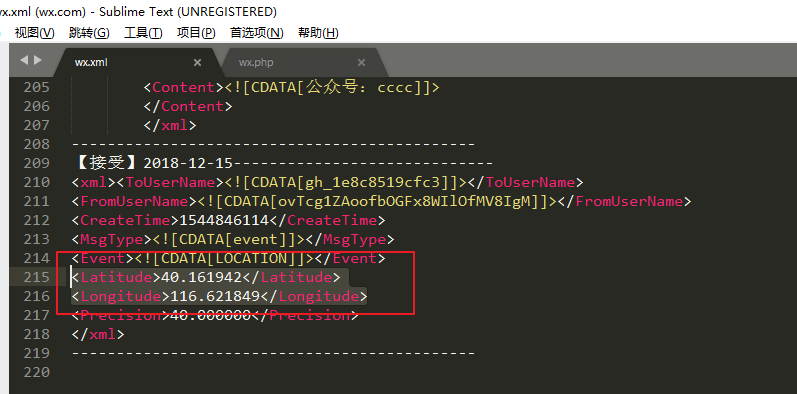




## 3.2、接受用户位置

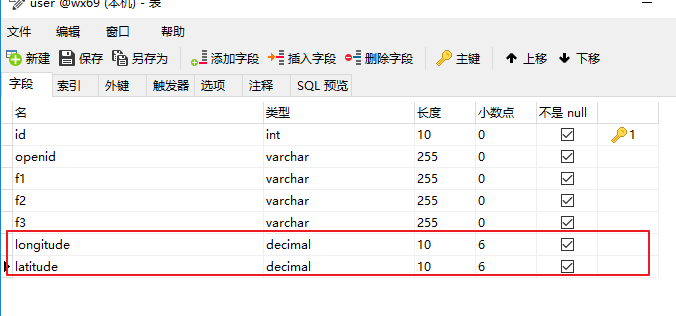


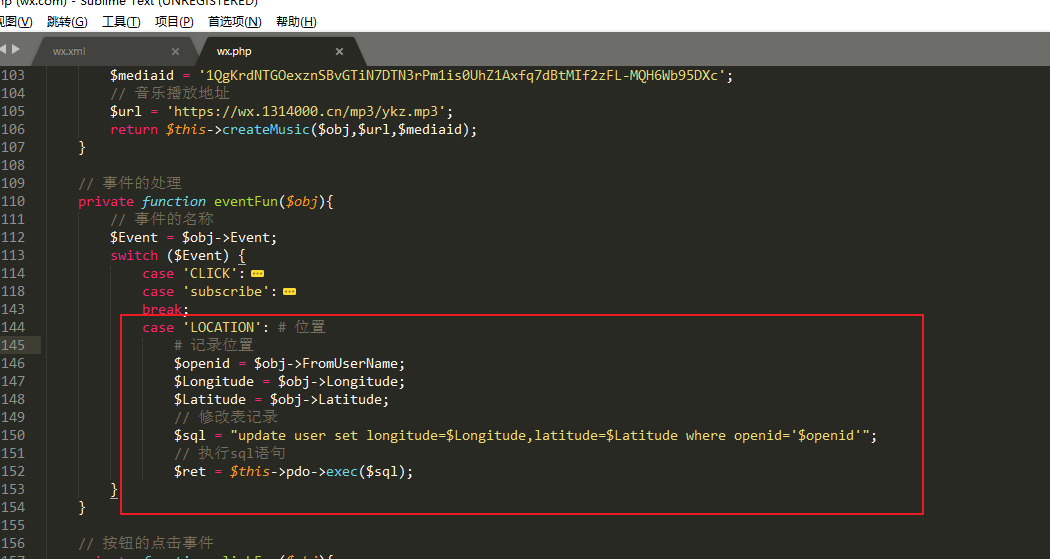


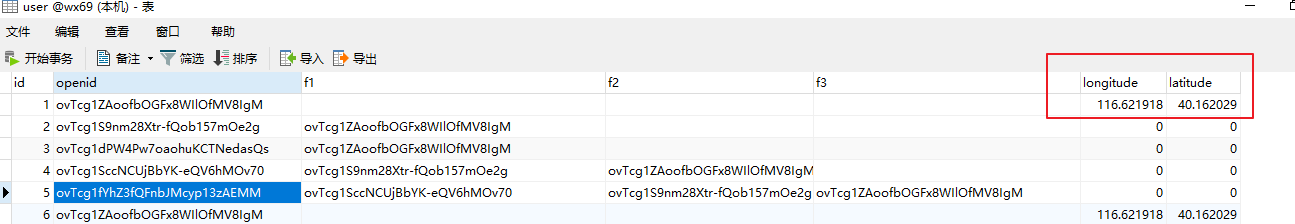


## 3.3、存储位置入库

第1步：修改用户表中的字段，添加经纬度字段







## 3.4、整合高德位置服务

基于高德开放平台，来基于位置的服务。

高德开发平台：<https://lbs.amap.com>



### 3.4.1、注册账号





根据开放平台，所提示消息，进行一步步的资料的完成。

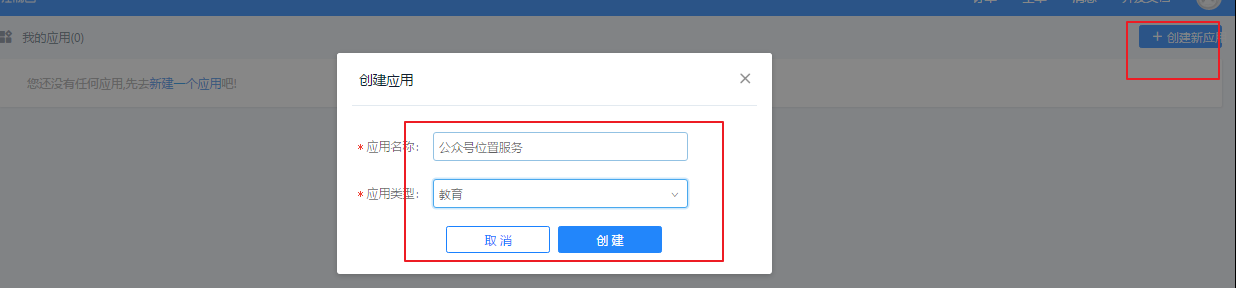
注册成功后，进行登录进入到高德开放平台控制台界面中



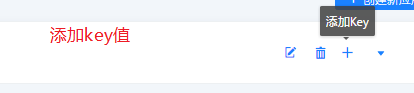
### 3.2.2、添加apikey

填写用户名和密码进行登录后，进入后台控制界面

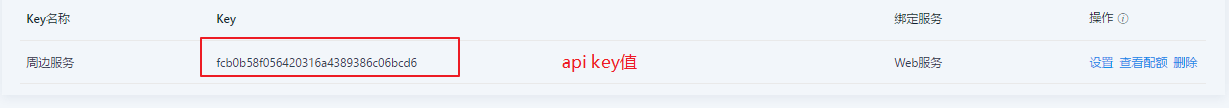








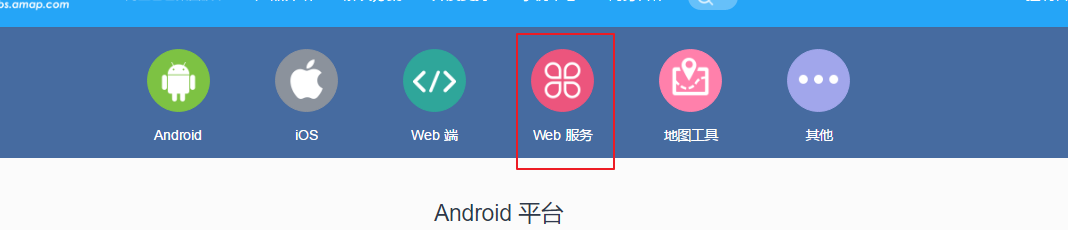




有了APIKEY值，我们就可以进行相关的业务代码的编写。

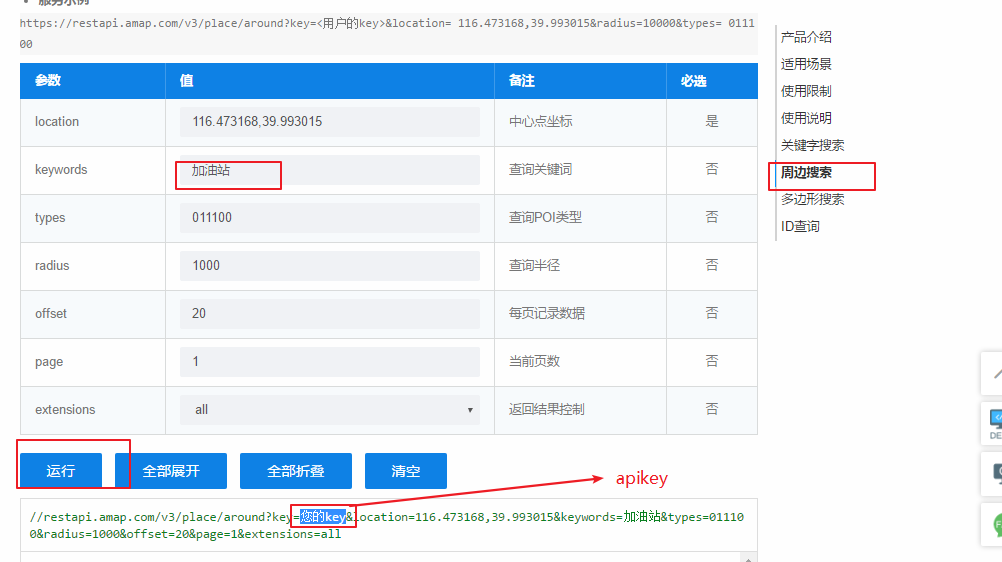
点击进入开发文档







根据测试的账号，我们得知需要一个apikey值



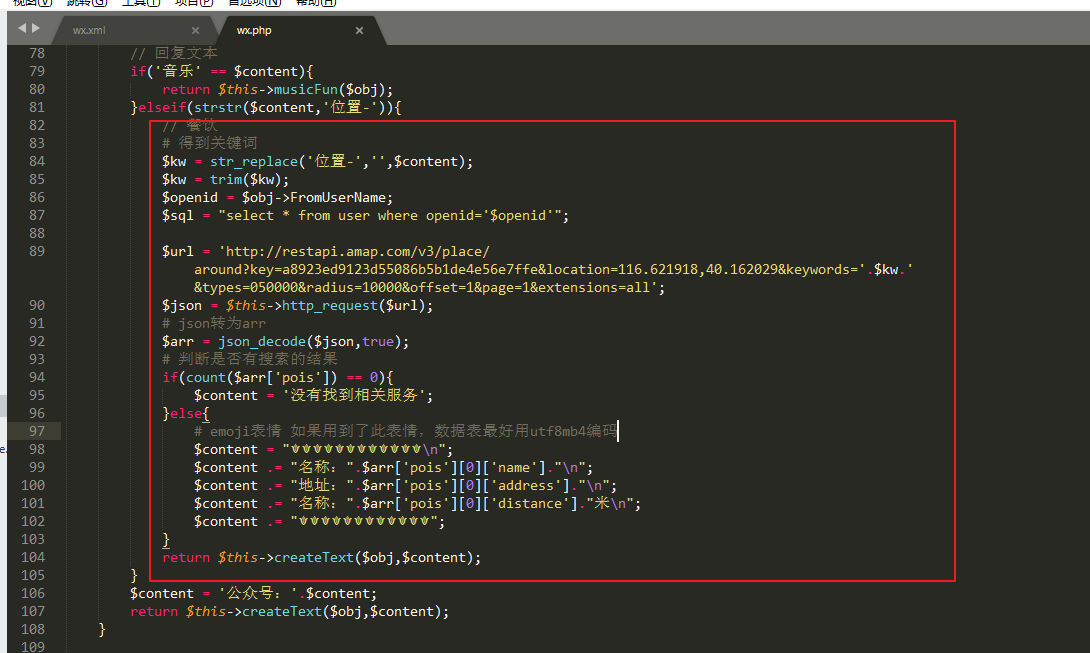
### 3.2.3、整合高德周边搜索到公众中

周边服务api地址：<https://lbs.amap.com/api/webservice/guide/api/search/#around>

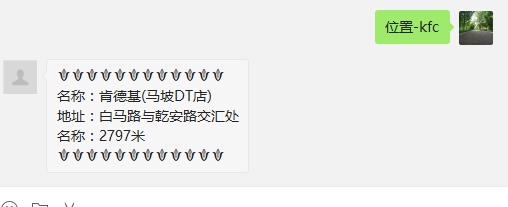
emoji表情：<http://www.sitemap-xml.org/editor/emoji/>



实现代码：



效果



小结：

* 在后台开启获取位置服务，选5s获取一次
* 在事件获取位置的接受处理中，把位置入库
* 注册位置提供的开放平台(高德),得到api接口地址和apikey
* 在公众号中进行关键词回复功能中添加相关的业务代码
* 可以让回复的文本更加的优美，用于emoji表情

# 四、网页授权

## 4.1、概述

公众号后台可以通过设置授权域名，然后通过此域名生成相应的链接，分享到朋友圈，好友等，后用户通过微信客户端来点击访问此链接，公众平台就可以获取用户基本信息，进而实现业务逻辑。

应用场景：

问卷调查

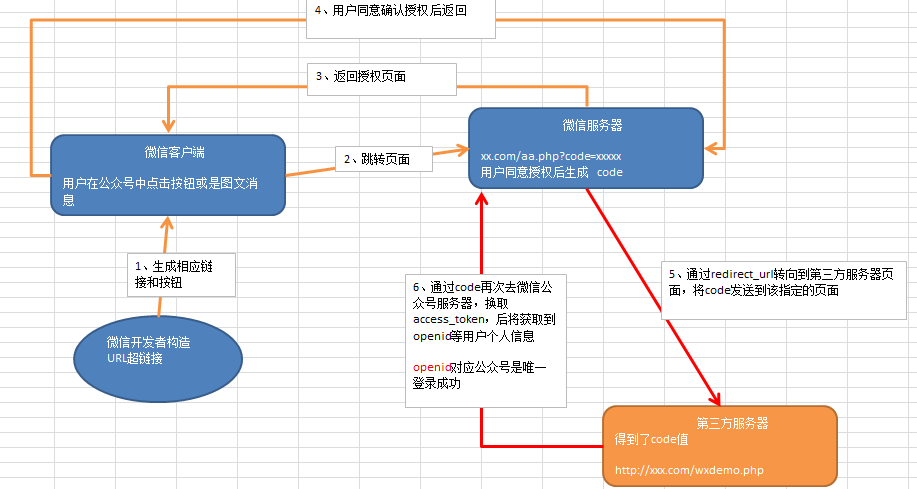
抽奖

购物车

如果openid和用户和密码，直接登陆

## 4.2、网页授权原理图

OAuth2.0鉴权



## 4.3、设置微信授权域名



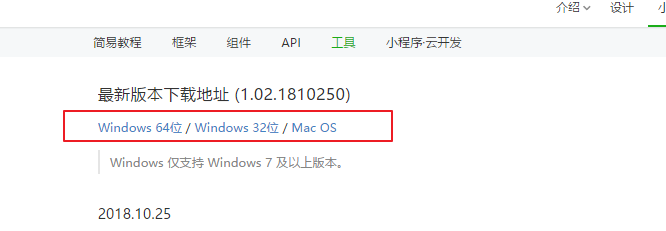


## 4.4、网页授权开发工具

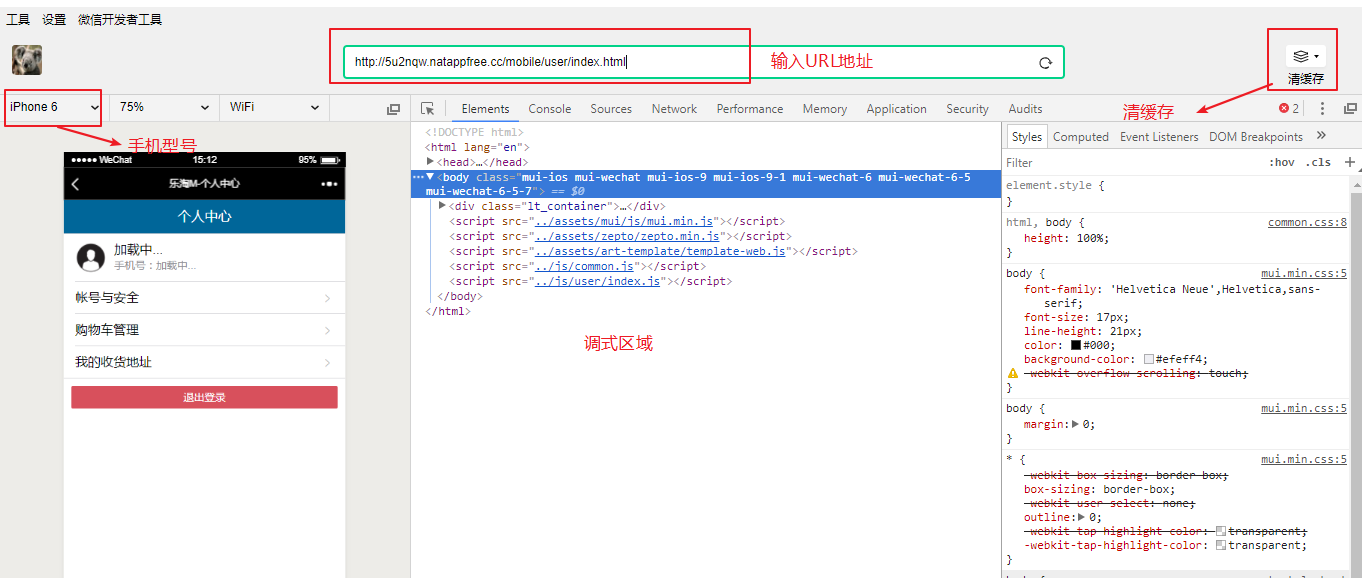
为了帮助开发者简单和高效地开发和调试微信公众号，推出了全新的 微信开发者工具，集成了公众号网页调试和小程序调试两种开发模式。

下载地址：

<https://developers.weixin.qq.com/miniprogram/dev/devtools/download.html>

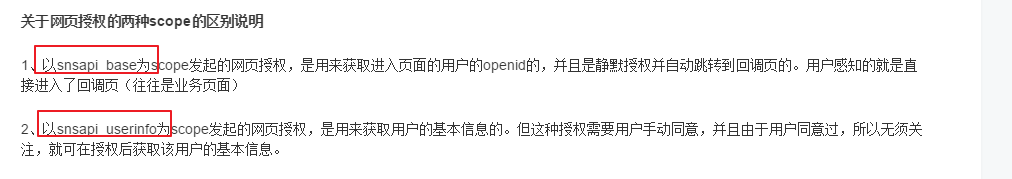






## 4.5、微信授权代码实现

<https://mp.weixin.qq.com/wiki?t=resource/res_main&id=mp1421140842>

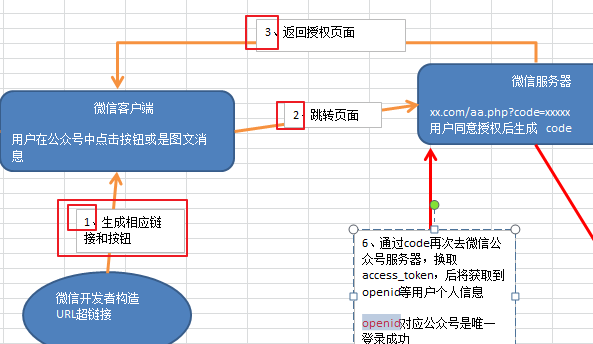


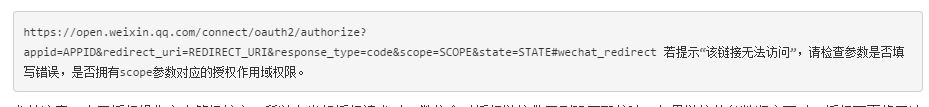
base此权限只能获取用户的openid而得不到用户的基本信息，授权是无感，不需要用户确认就可以完成授权，静默方式。

userinfo 可能获取openid和用户的基本信息，需要用户确认。



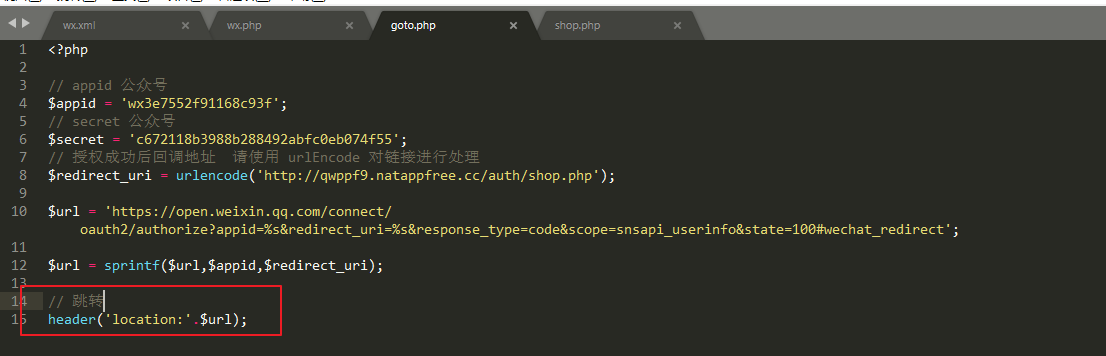
### 4.5.1、获取code值







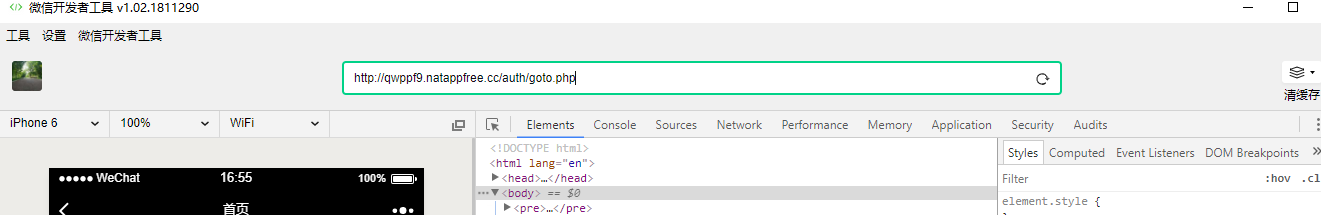
在服务器中编写生成跳转地址url程序



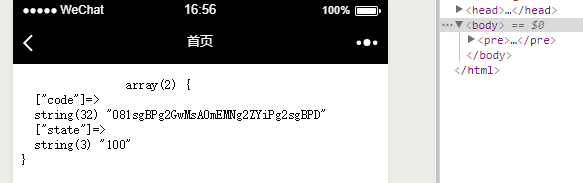
编写一个授权跳转过来的接受页面



把生成跳转页面的URL放入到微信中进行测试

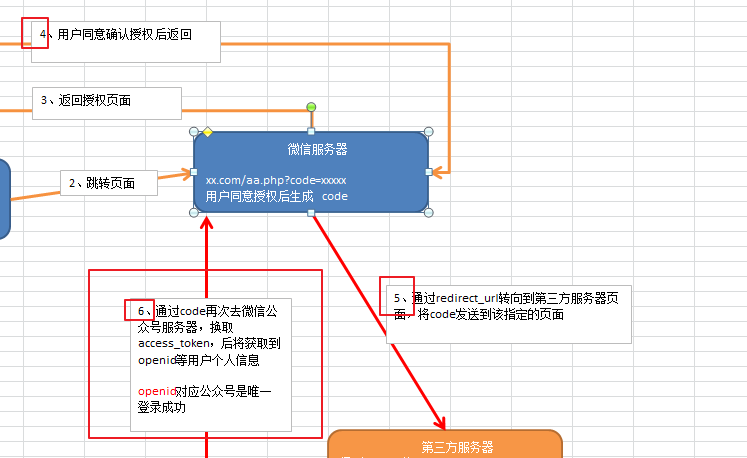




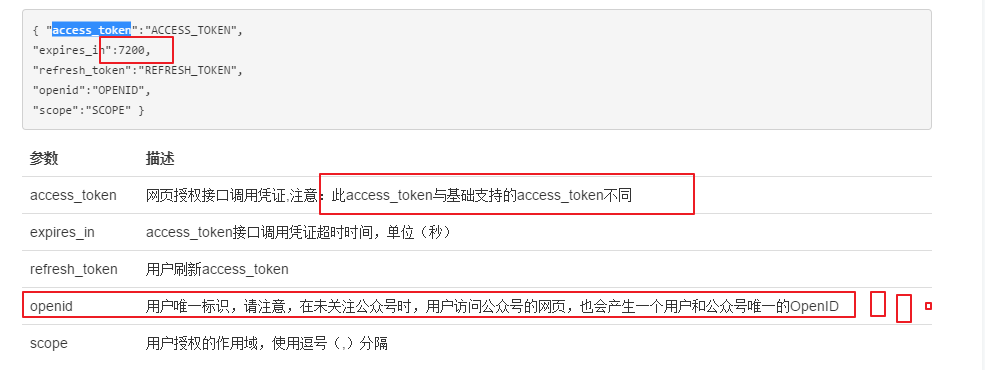


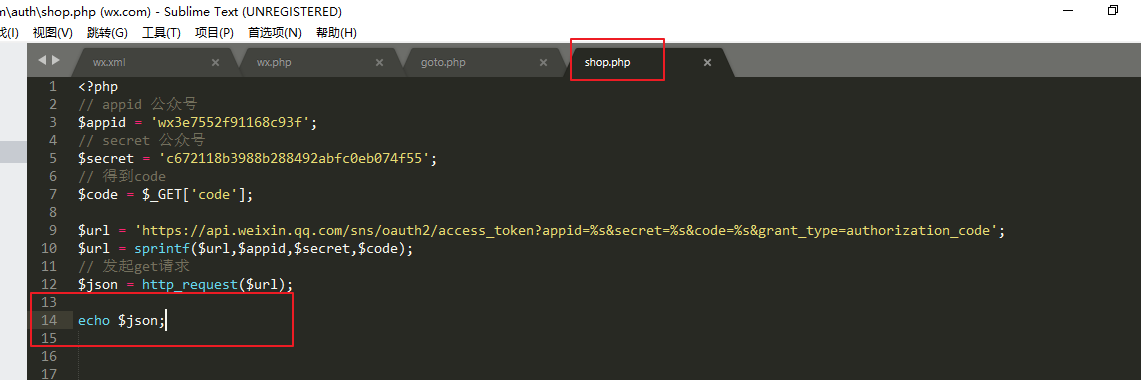
注：code有效期为5分钟

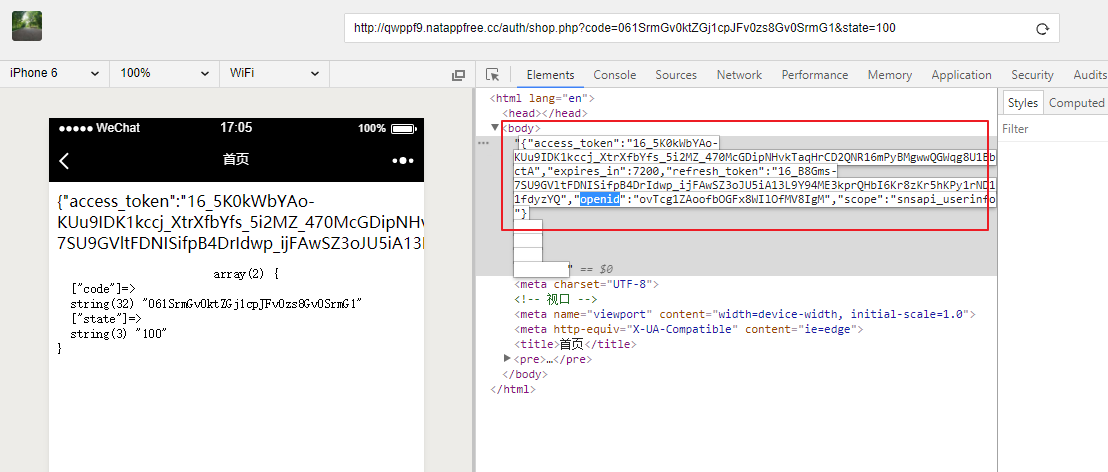
### 4.5.2、得到openid完成授权









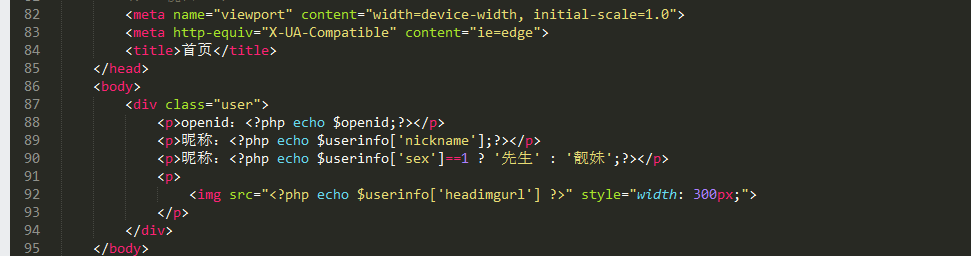


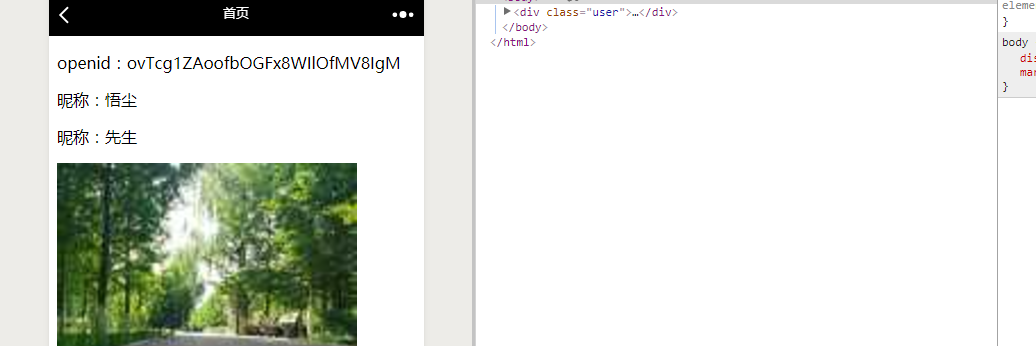
### 4.5.3、拉取用户信息











小结：

* 生成跳转的url地址，得到了code
* 用code换取access\_token和openid,完成授权
* 非必须的，如果是userinfo授权，则可以获取用户的基本信息

# 五、微信JS-SDK

## 5.1、什么是JSSDK

就是微信帮助我们完成的JS一个工具的封装，直接调用这些接口就可以完成相应的功能。

**作用：**

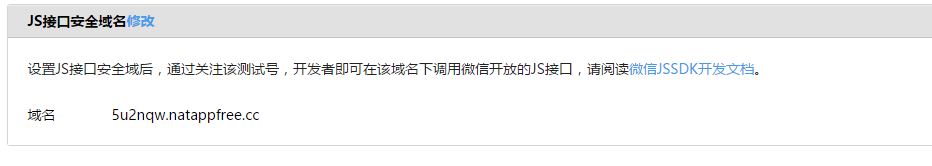
* 自定义分享的链接(此链接也是有规定，如公众后台设置有效域名一致)
* 调用一些手机底层应用，摄像头，获取网络制式

**开发5步骤：**

* 平台绑定授权域名
* 引入公众平台jssdk的js文件
* 根据官方签名算法编写出对就的签名字符串
* 设置前台script中的config接口配置注入权限验证
* 通过前台script中的接口提供的ready方法处理成功验证

## 5.2、jssdk绑定域名

在开发测试平台添加绑定域名

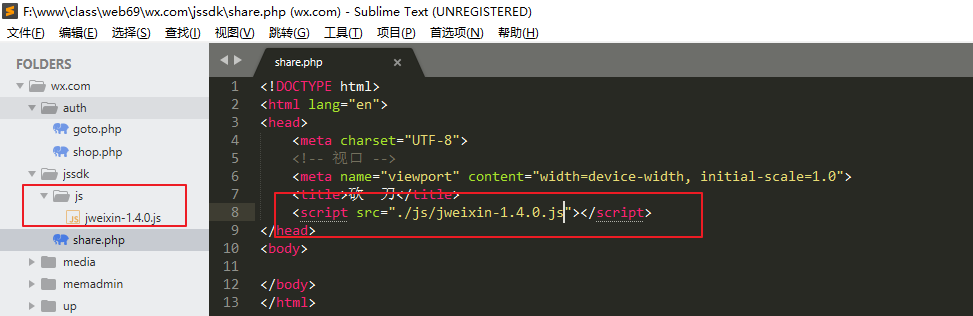


一定要添加上授权域名，不然不能进行下一步的操作。

## 5.3、JSSDK使用步骤

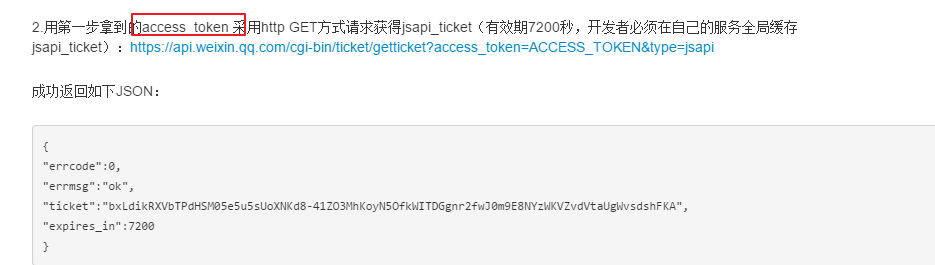
### 5.3.1、引入JS文件



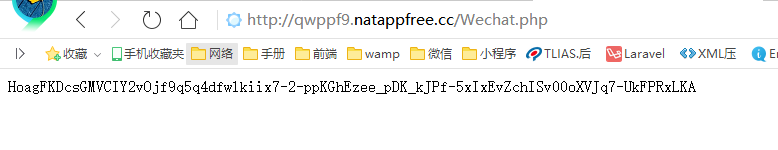


### 5.3.2、生成签名算法

#### 5.3.2.1、得到jsapi\_ticket





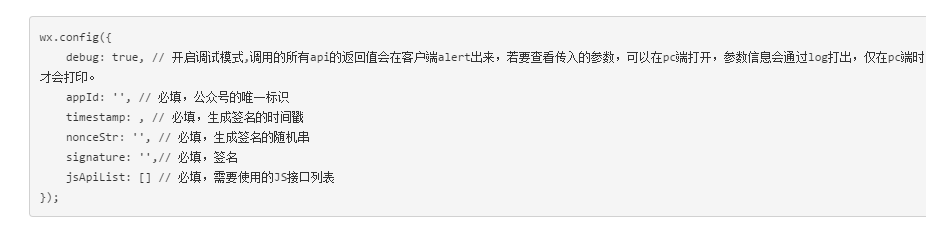


#### 5.3.2.2、生成签名

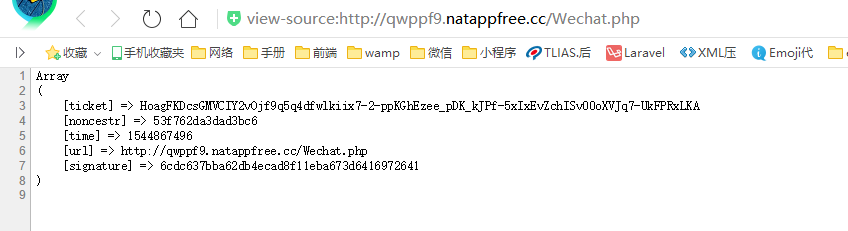


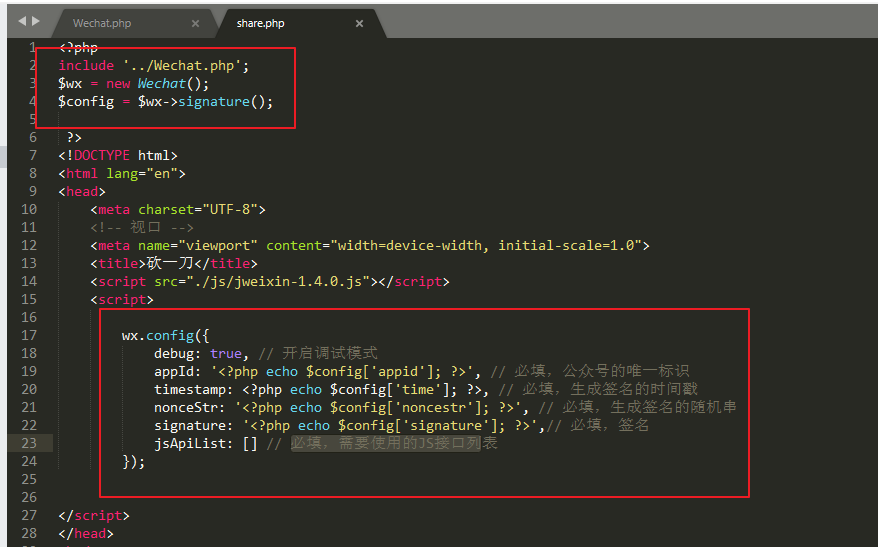


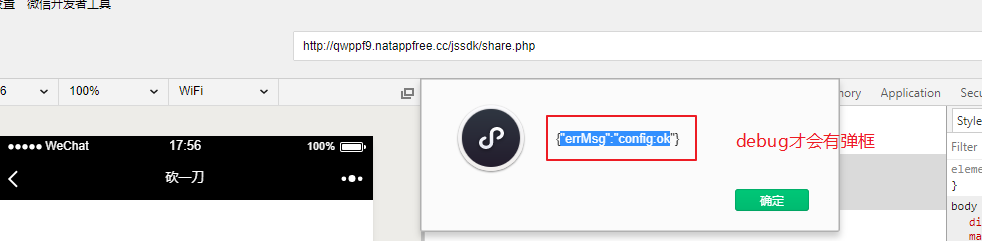
### 5.3.3、通过config接口注入权限验证配置

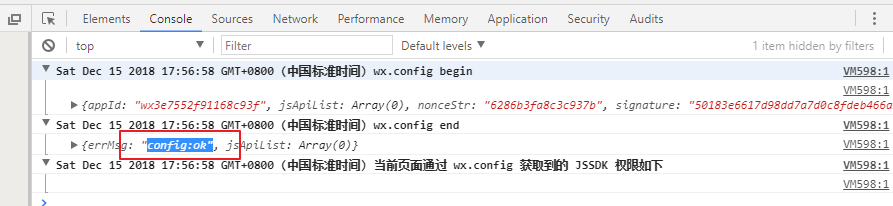












### 5.3.4、通过ready接口处理成功验证

