Wechat微信公众平台开发

目录

[Wechat微信公众平台开发 1](#_Toc475985050)

[一、微信的语音识别接口 2](#_Toc475985051)

[1、什么是语音识别 2](#_Toc475985052)

[2、微信的语音识别接口 2](#_Toc475985053)

[3、使用php代码获取微信的语音识别结果 2](#_Toc475985054)

[二、图灵机器人接口（第三方平台） 3](#_Toc475985055)

[1、什么是图灵机器人 3](#_Toc475985056)

[2、图灵机器人接口 3](#_Toc475985057)

[3、使用php代码对接图灵机器人 4](#_Toc475985058)

[4、在微信中整合图灵机器人 5](#_Toc475985059)

[三、网页授权接口（Oauth2.0） 5](#_Toc475985060)

[1、什么是网页授权接口（Oauth2.0） 5](#_Toc475985061)

[2、网页授权模式说明 6](#_Toc475985062)

[3、网页授权开发步骤 6](#_Toc475985063)

[4、实现网页授权功能 6](#_Toc475985064)

[四、微信JSSDK接口 9](#_Toc475985065)

[1、什么是JSSDK接口 9](#_Toc475985066)

[2、下载JSSDK官方接口 10](#_Toc475985067)

[3、微信JSSDK开发步骤 10](#_Toc475985068)

[4、实现JSSDK接口功能 10](#_Toc475985069)

[五、微信接口与MySQL数据库 13](#_Toc475985070)

[1、设置CentOS系统中MySQL的编码格式 13](#_Toc475985071)

[2、创建相应数据库 14](#_Toc475985072)

[3、在微信接口中调用数据库信息 14](#_Toc475985073)

[六、微信接口与Memcached 15](#_Toc475985074)

[1、Memcached概述 15](#_Toc475985075)

[2、微信中Memcached应用场景 15](#_Toc475985076)

[3、在CentOS系统中安装Memcached 15](#_Toc475985077)

[4、安装php memcached扩展 16](#_Toc475985078)

[5、使用Memcached缓存access\_token接口调用凭证 16](#_Toc475985079)

[6、使用Memcached实现微信多级菜单 17](#_Toc475985080)

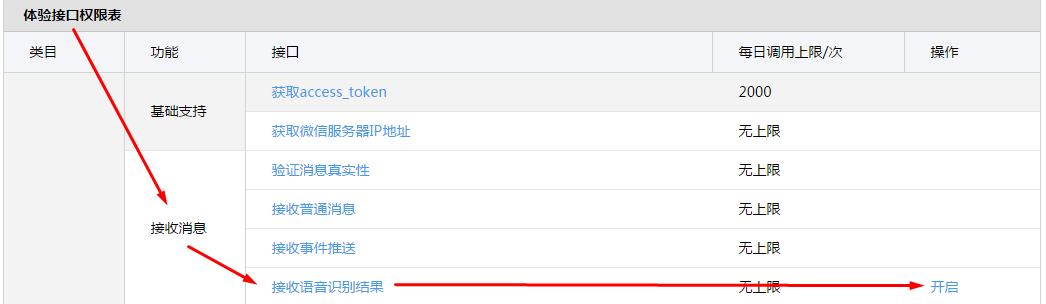
# 一、微信的语音识别接口

## 1、什么是语音识别

语音识别就是让机器通过识别和理解过程把语音信号转变为相应的文本内容，简言之就是让机器听懂人类的语言。

## 2、微信的语音识别接口

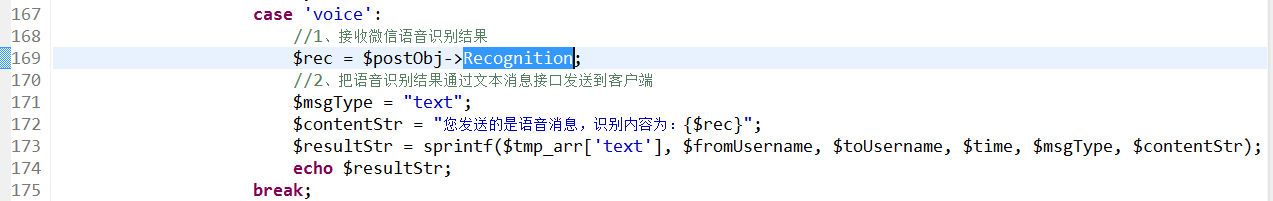
① 必须要手工开启语音识别接口



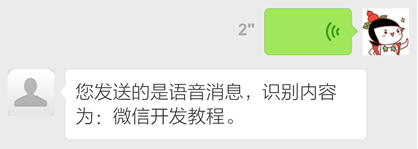
② 语音识别结果对应的XML模板



## 3、使用php代码获取微信的语音识别结果



保存代码，然后使用FlashFXP软件上传到阿里云服务器，测试结果如下图：



# 二、图灵机器人接口（第三方平台）

## 1、什么是图灵机器人

图灵机器人是中文语境下智能度最高的机器人。截至目前，图灵机器人对中文语义的理解准确率高达90%以上，累计开发者及合作伙伴超23万。

图灵机器人官网：http://www.tuling123.com/

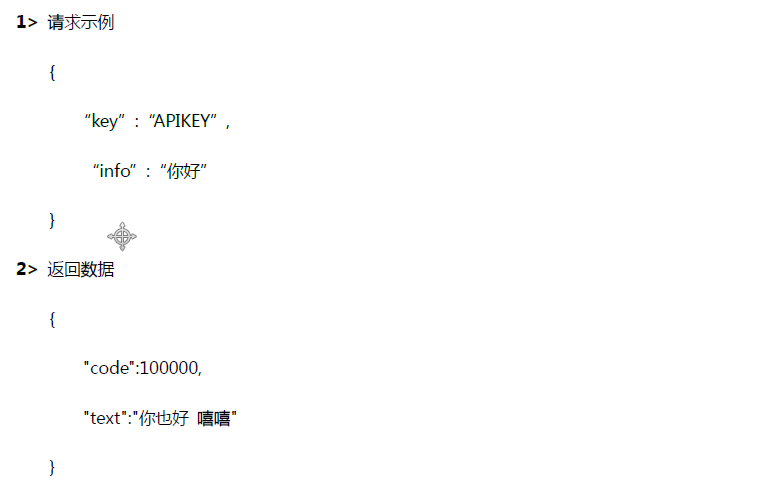
## 2、图灵机器人接口

接口地址：<http://www.tuling123.com/openapi/api>

应用key：9009fc44f168cfc7055c8a469821ce9b

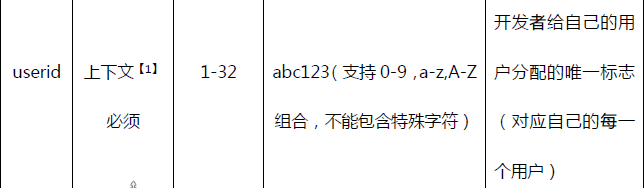
请求类型：HTTP POST

传输数据结构：



注意事项：

① 传输数据时必须携带userid



② 使用curl库向图灵发送请求时，必须设置头信息，否则服务器端无法接收到传递的数据。

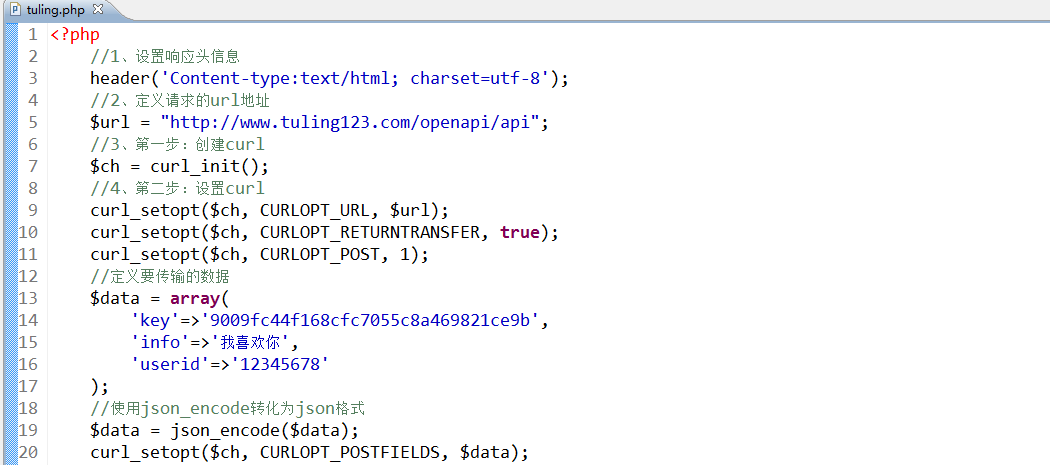
curl\_setopt($ch, CURLOPT\_HTTPHEADER, array(

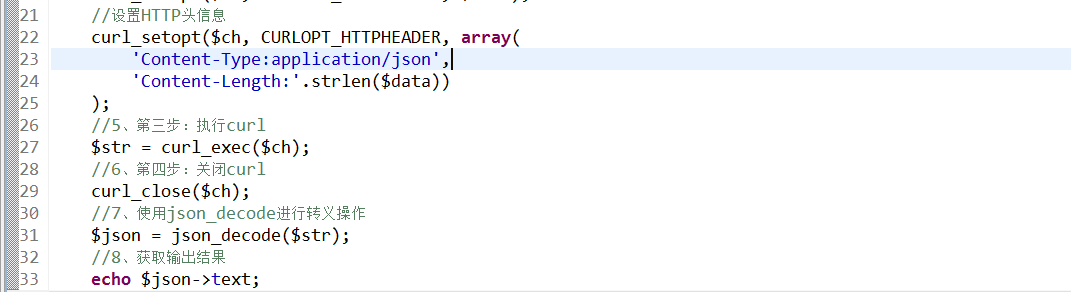
'Content-Type:application/json',

'Content-Length:'.strlen($data))

);

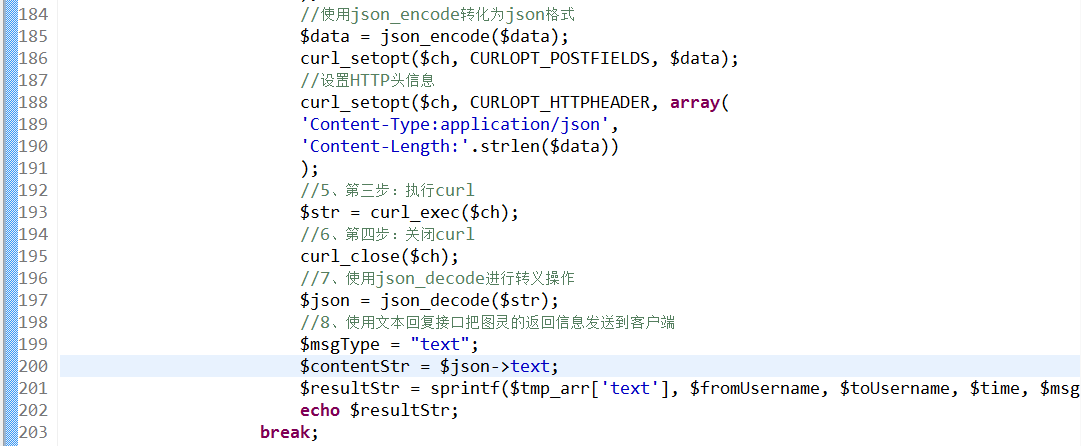
## 3、使用php代码对接图灵机器人



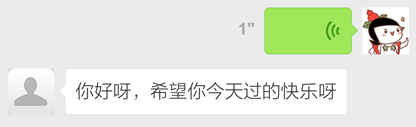


## 4、在微信中整合图灵机器人





保存代码，然后使用FlashFXP软件上传代码到阿里云服务器，测试结果如下图所示：



# 三、网页授权接口（Oauth2.0）

## 1、什么是网页授权接口（Oauth2.0）

如果用户在微信客户端中访问第三方网页，公众号可以通过微信网页授权机制，来获取用户基本信息，进而实现相关的业务逻辑功能，如账号绑定等等。

## 2、网页授权模式说明

1、以snsapi\_base为scope发起的网页授权，是用来获取进入页面的用户的openid的，并且是静默授权并自动跳转到回调页的。用户感知的就是直接进入了回调页（往往是业务页面）

2、以snsapi\_userinfo为scope发起的网页授权，是用来获取用户的基本信息的。但这种授权需要用户手动同意，并且由于用户同意过，所以无须关注，就可在授权后获取该用户的基本信息。



## 3、网页授权开发步骤

① 设置回调域名（非常重要）

② 使用客服接口向微信端发送授权url

③ 在回调redirect\_uri页面获取code

④ 通过code换取网页授权access\_token（基础支持中的access\_token完全不同，特别小心）

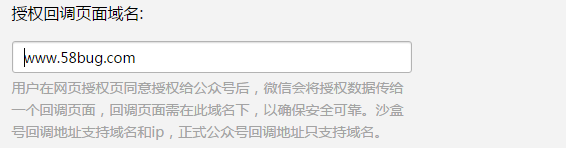
⑤ 使用access\_token + openid拉取用户信息

## 4、实现网页授权功能

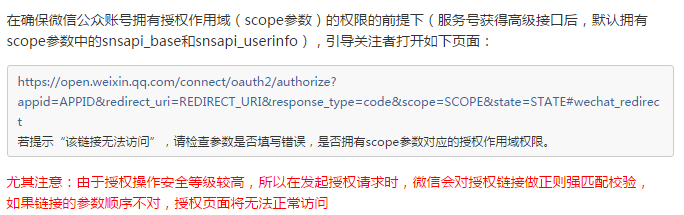
① 设置回调域名（测试号管理-体验接口权限表）



单击修改按钮，设置回调域名，如下图所示：



② 使用客服接口向微信端发送授权url



案例：获取授权url链接

<https://open.weixin.qq.com/connect/oauth2/authorize?appid=wx127080c82bf38e25&redirect_uri=http://www.58bug.com/userinfo.php&response_type=code&scope=snsapi_base&state=123#wechat_redirect>

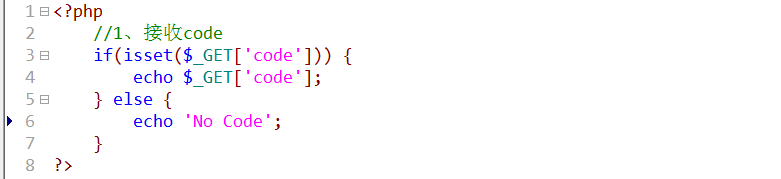
使用客服消息接口（create\_url.php）发送授权链接到客户端，示例代码：



运行结果如下图所示：



③ 在回调redirect\_uri页面获取code（userinfo.php编写代码）



运行结果：

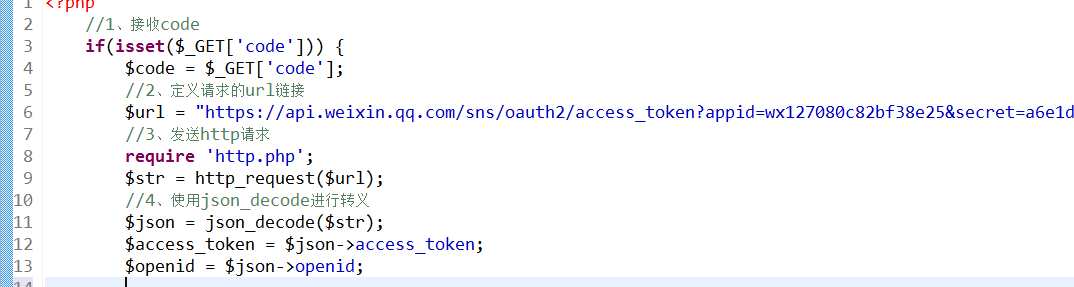


④ 通过code换取网页授权access\_token（基础支持中的access\_token完全不同，特别小心）

分析接口，如下图所示：



在userinfo.php页面中编写代码：

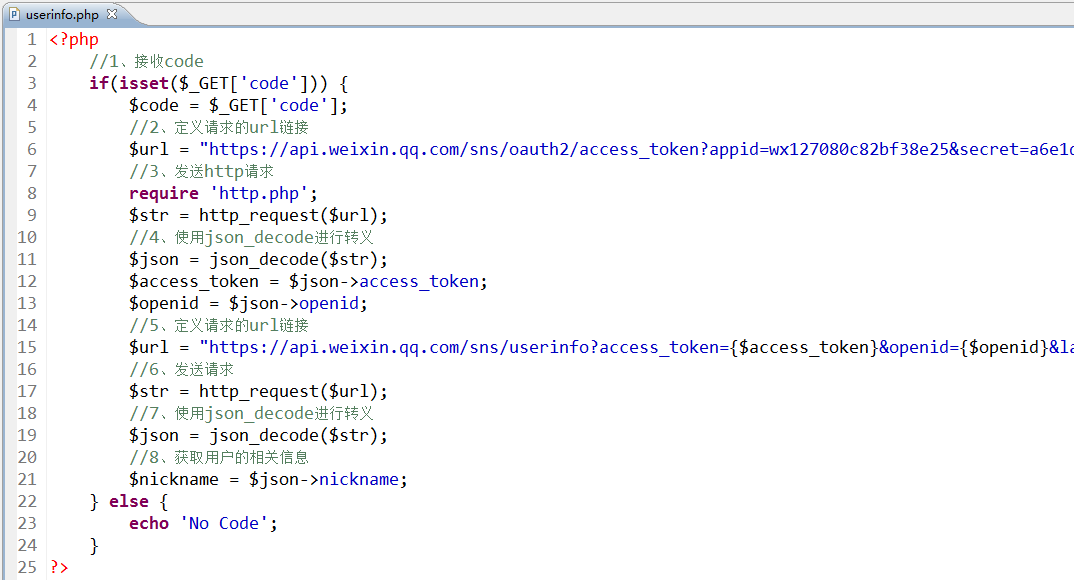


⑤ 使用access\_token + openid拉取用户信息

分析接口，如下图所示：



在userinfo.php页面编写代码，如下图所示：



运行结果：



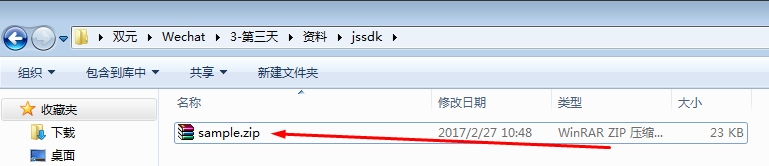
# 四、微信JSSDK接口

## 1、什么是JSSDK接口

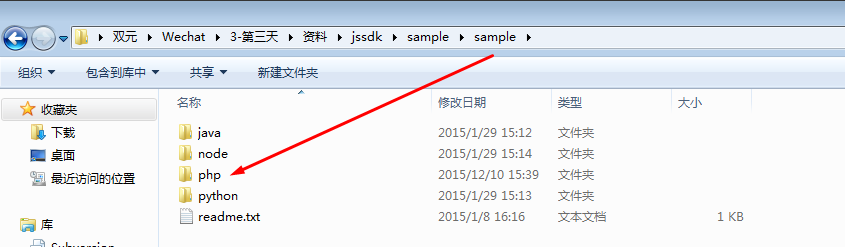
微信JS-SDK是微信公众平台面向网页开发者提供的基于微信内的网页开发工具包。通过使用微信JS-SDK，网页开发者可以调用微信中的拍照、分享、支付、摇一摇等特有的功能，为微信用户提供更优质的网页体验。

## 2、下载JSSDK官方接口

下载地址：<http://demo.open.weixin.qq.com/jssdk/sample.zip>



解压后，如下图所示：



## 3、微信JSSDK开发步骤

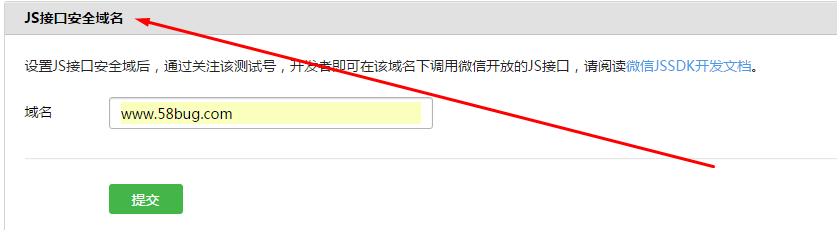
① 在php页面中配置appid与appsecret

② 在jsApiList中添加应用权限（如分享接口）

③ 在wx.ready中编写相关接口代码

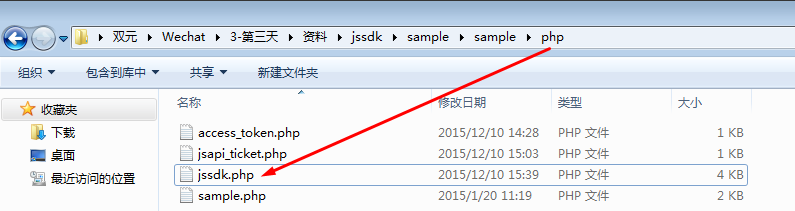
## 4、实现JSSDK接口功能

实现JSSDK接口功能必须有一个前提：设置JS接口安全域名，如下图所示：

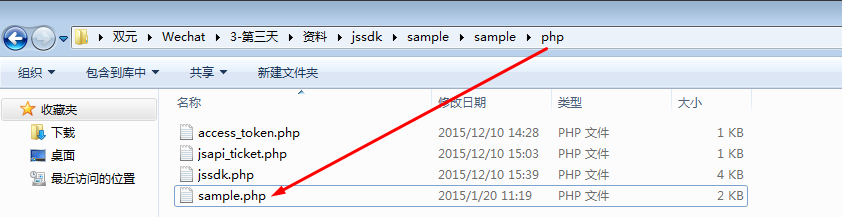


① 在php页面中配置appid与appsecret

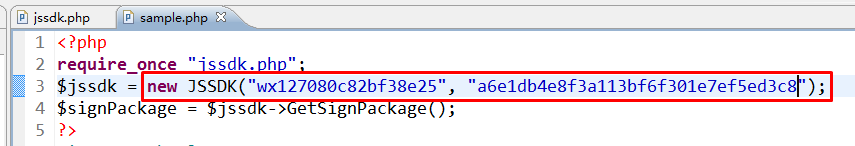
第一步：复制php接口中的jssdk.php页面到wechat项目中



第二步：复制sample.php页面到wechat项目中，JSSDK接口的案例



第三步：在sample.php示例代码中，配置appid与appsecret



② 在jsApiList中添加应用权限（如分享接口）

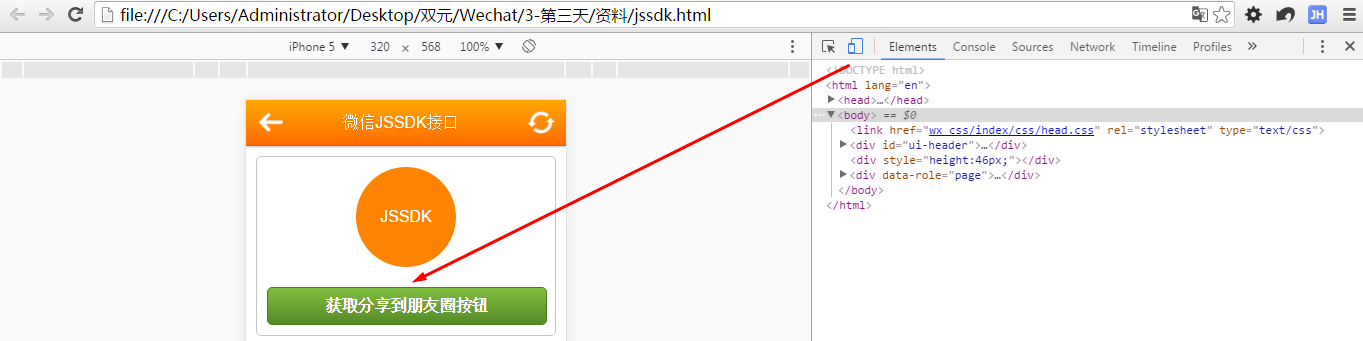


添加分享到朋友圈权限，如下图所示：

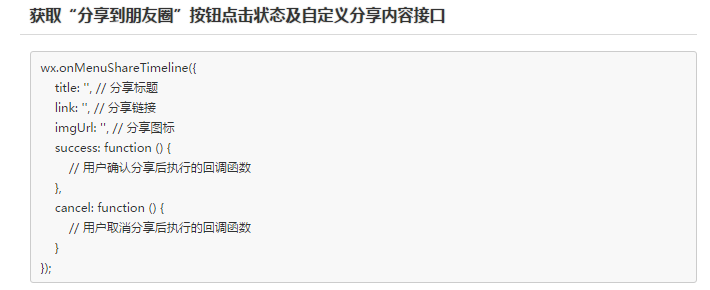


③ 在wx.ready中编写相关接口代码

第一步：编写相关的微网站页面，如下图所示：

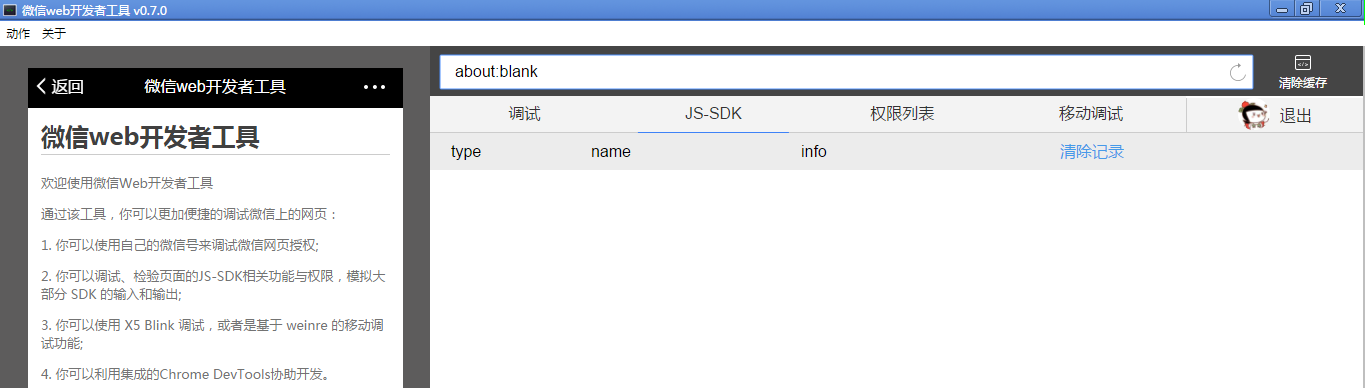


第二步：在wx.ready方法中，编写代码：





保存代码后，使用FlashFXP软件上传到服务器端，使用微信测试工具进行测试：



测试结果如下图所示：

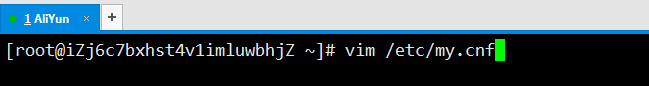


# 五、微信接口与MySQL数据库

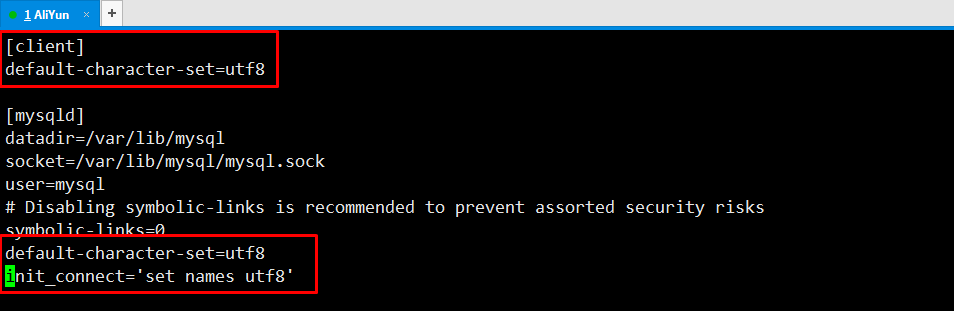
## 1、设置CentOS系统中MySQL的编码格式

在CentOS系统中，默认MySQL采用的是拉丁编码，我们必须对其进行更改操作，更改为utf8的编码格式。

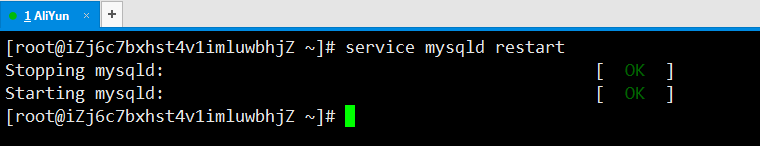
① 使用vim编辑器打开/etc目录my.cnf文件



② 设置编码格式如下：

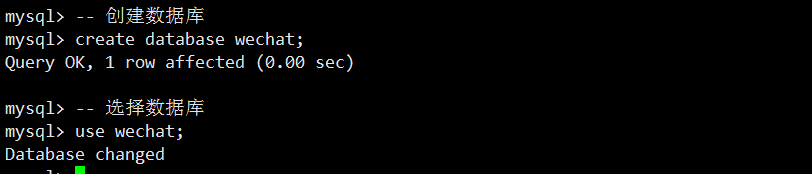


③ 配置完成后，保存，然后使用service指令对mysqld服务进行重启操作

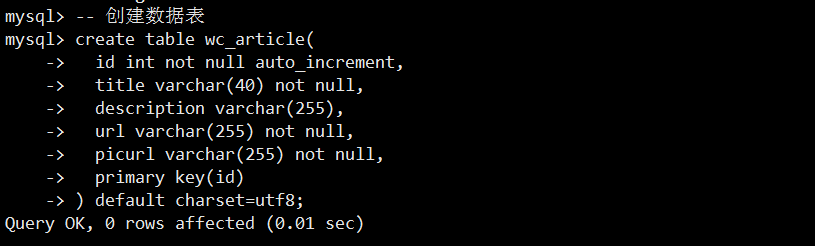


## 2、创建相应数据库

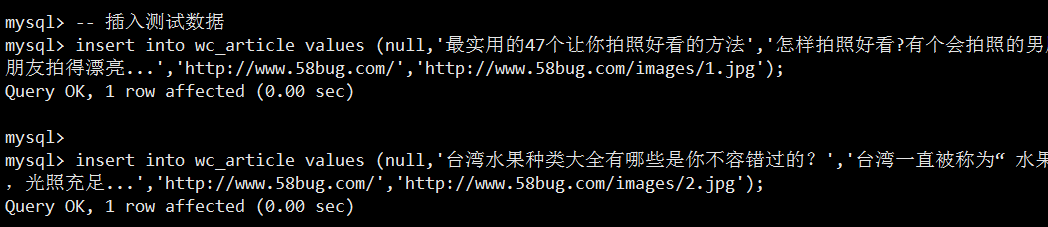
① 创建数据库wechat



② 在wechat数据库中创建wc\_article数据表



③ 插入测试数据，如下图所示：



## 3、在微信接口中调用数据库信息



代码保存完毕后，使用FlashFXP软件上传api.php代码到阿里云服务器，测试结果如下图所示：



# 六、微信接口与Memcached

## 1、Memcached概述

Memcached是一个高性能的分布式内存对象缓存系统，其主要通过在内存中缓存数据来减少数据库的读取次数，从而提高网站的访问速度。

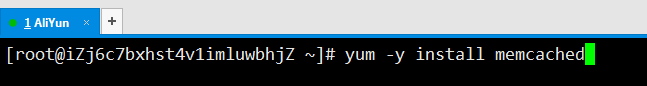
## 2、微信中Memcached应用场景

① 缓存access\_token接口调用凭证

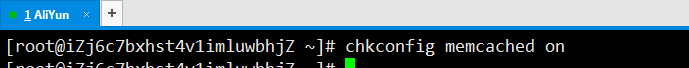
② 实现微信多级菜单

## 3、在CentOS系统中安装Memcached

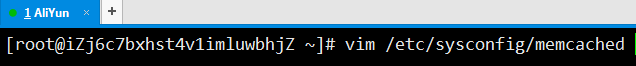
① 使用yum指令安装memcached



② 设置让memcached可以随开机启动

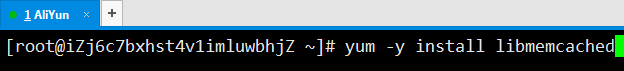


③ 设置memcached的配置文件

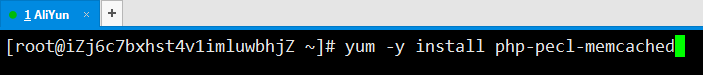


## 4、安装php memcached扩展

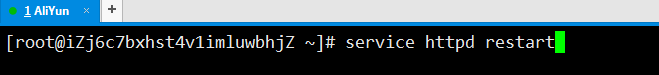
① 安装libmemcached依赖库



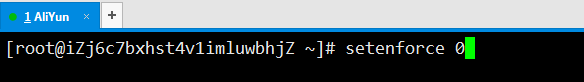
② 安装php-pecl-memcached扩展



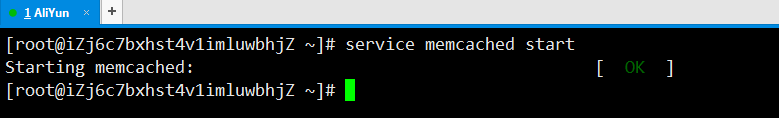
③ 重启httpd服务（重启apache）



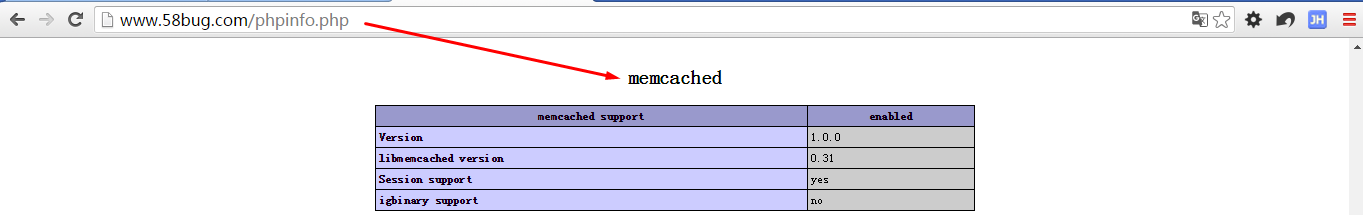
④ 关闭SELinux



⑤ 启动memcached

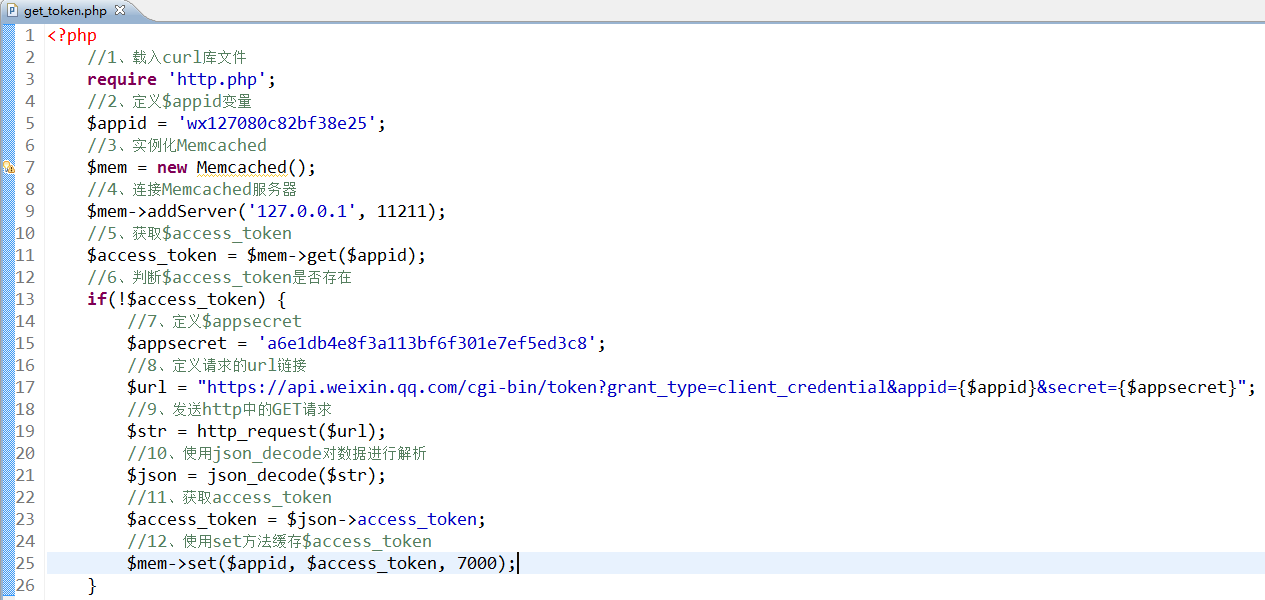


使用phpinfo.php测试扩展是否安装成功：



## 5、使用Memcached缓存access\_token接口调用凭证

重写get\_token.php页面，示例代码：



使用FlashFXP上传文件到服务器端，测试结果如下：



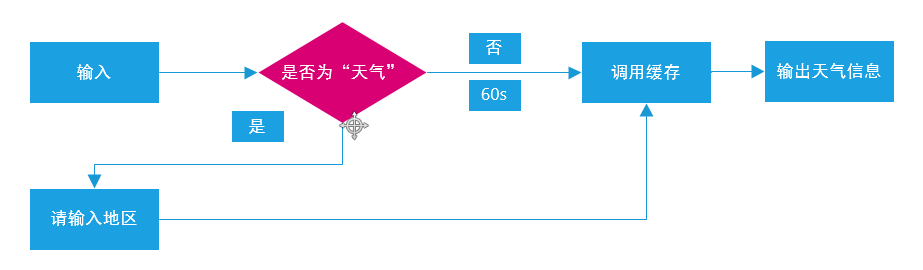
## 6、使用Memcached实现微信多级菜单

① 微信多级菜单说明

我们以“天气预报”为例：

首先提示用户先输入“天气”关键词，然后返回输入城市名称，1分钟内可以连续查询，1分钟后则需要重新输入“天气”关键词才能进行查询。

② 微信多级菜单原理图



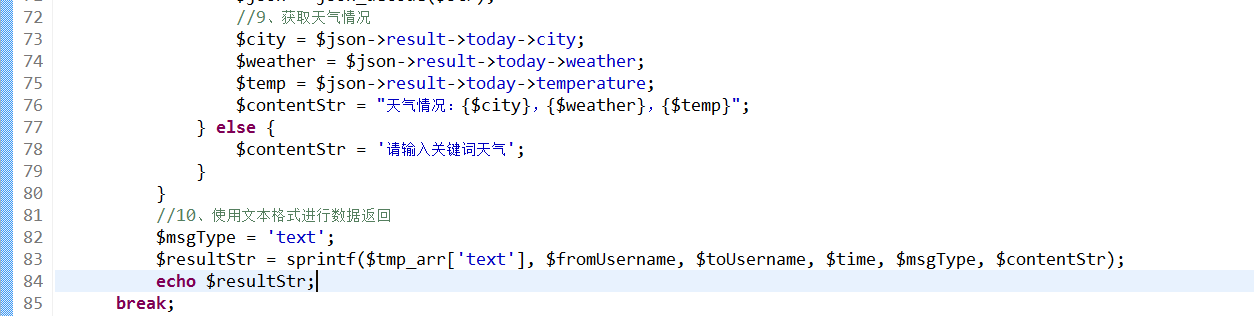
③ 使用php+memcached实现微信的多级菜单

要用到的知识点：聚合数据（<http://www.juhe.cn/>）



示例代码：





保存代码，然后使用FlashFXP软件上传到阿里云服务器，打开微信端测试结果如下图所示：

