### 一、环境是lnamp

###### 环境介绍：

###### 0、mykj编码都是gbk的

1、Apache-php7的容器必须支持：curl 命令，并且支持nds解析

下面的案例不支持：导致无法请求微信接口

root@ab2cab43cb8a:/var/www/html# curl api.weixin.qq.com

curl: (6) Could not resolve host: api.weixin.qq.com

**解决办法：**需要查看网络ip是不是有多余的删除

查看网络接口： [root@localhost ~]# nmcli device status

查看某个接口：[root@localhost ~]# nmcli device show ens33

删除某一个ip: nmcli conn delete ens33 ipv4.addresses 192.168.88.22

然后查看开启IP转发功能

[root@localhost sysconfig]# cd /etc/sysctl.d/ && ls

99-sysctl.conf

[root@localhost sysctl.d]# vim 99-sysctl.conf

设置net.ipv4.ip\_forward = 1

查看开启情况

[root@localhost sysctl.d]# sysctl -p

net.ipv4.ip\_forward = 1

[root@localhost sysctl.d]# sysctl net.ipv4.ip\_forward

net.ipv4.ip\_forward = 1

生产环境不要使用apt-get update命令，因为下面有介绍

最近领导让研究nagios，作为一名小白，以前从来没有接触过nagios，也是云里雾里的就开搞，终于，三天研究下来，基本监控项可以配置，等着部署客户端监控其他服务器了，但是服务器也有好几十台，一个个部署，有点太愚昧，所以，利用ansible批量执行本地脚本，自己在本地写完编译脚本，开始推送公钥，一切准备就绪，执行命令ansible all -m script 'shell.sh',结果。。。。悲剧了，邮箱开始炸了。

接二连三的apache服务down掉了，mysql服务down掉了，立刻终止批量执行命令，查看报警服务器，发现mysql已经由原来的5.5.34更新为5.5.54，不过还好，没有造成什么损失，产品没有报障，但是，有一台服务器，mysql没有起来，登上去手动启动，提示start job failed，这是为什么，查看error.log，里面并没有任何的日志信息，空文件，百度说是因为socket文件丢失，但别忘了，mysql不启动，socket文件肯定没有，接着查看各个文件权限，运行用户都是mysql，权限正常。

查找了大概半个小时，还是没有确定原因，决定替换一下配置文件试试，找了一台启动正常的mysql服务器，down下来他的配置文件，使用rz上传配置文件到有问题这台，但没有装这个命令，然后执行apt-get install lrzsz安装,执行过程中，奇迹发生了，显示mysql有一部分并没有更新完成，在装rz的过程中自动修复了未完成的一部分，查看mysql状态，启动正常了，误打误撞解决了无法启动问题。（出问题的这台服务器上面的数据库没有业务使用，调用的redis，所以并没有影响业务）

所以，这也是低级错误，也是自身技术欠缺，分享出来，希望大家在生产环境中，慎用apt-get update ，apt-get upgrade，yum update这些命令，同时批量执行脚本时，先在本地测试一下，然后找台线上的测试一下，确定没有问题，再批量执行，否则，一旦出问题，真的很难快速解决。

最后，希望各位大神看完别喷，因为我毕竟是小白一名，只是希望后面跟我一样的小白别犯这样的错误。

2、容器挂载目录时，宿主机的目录会清空对应容器的目录，所以必须把容器的配置文件复制到宿主机中；另外挂载时，目录对应目录，文件对应文件。

###### php.ini配置要求：

#session.save\_handler = user //用户自定义session, 默认是files

上面这句过时了，不用管

session.cookie\_lifetime = 86400 //24小时

session.cookie\_lifetime = 31536000 //一年

post\_max\_size = 26M

upload\_max\_filesize = 26M

###### 0.nginx,mysql,php7-apache这些都是默认安装了discuz

###### 1.mysql账号是root 密码000000

###### 2.论坛账号是admin 密码admin

###### 3.window电脑 host设置访问到虚拟机

###### 192.168.134.33 discuz.com

###### 192.168.134.33 bbs.discuz.com

###### 192.168.134.33 3g.discuz.com

###### 4.查看启动端口

###### yum install net-tools 安装netstat

###### netstat -lntpu #listen n端口 tcp udp pid进程

###### 5.防火墙janusec docker镜像自己安装

###### docker pull

###### registry.cn-shenzhen.aliyuncs.com/janusec/janusec:0.9.16

###### docker run -d --privileged=true -v /mnt/docker/janusec/conf:/tmp -p 80:80 -p 443:443 -p 9080:9080 -p 9443:9443 f6fb99e41213 /sbin/init

###### 注意：必须分配IP给 容器，还要互通才行

###### docker run -d --privileged=true -p 80:80 -p 443:443 -p 9080:9080 -p 9443:9443 这里改成镜像ID /sbin/init

###### 涉及到的口令均为 J@nusec123 woyehaizhishiwo@qq.com

###### http://192.168.134.33:9080/janusec-admin/

###### 防火墙的账号是admin

###### 进入防火墙的容器：

###### docker exec -it janusec /bin/bash

###### docker exec -it --user=root Container\_ID /bin/bash

###### docker exec -it --user=root eb94d40f7efe /bin/bash

###### PostgreSQL account: janusec/J@nusec123

###### 必须在容器内启动防火墙命令：/usr/local/janusec/janusec &

###### 这样用的话：如果用于生产环境，这样遇到异常中断服务时不会自动重启

###### 说明文档

###### https://doc.janusec.com/cn/diagnose/

###### 验证：http://discuz.com/phpinfo.php?id=1 and 1=1

###### 关闭seliux,防火墙容器才不会报错

###### 6.#将容器php7-fmp打包为swoole的镜像

###### docker commit php7-fpm swoole

###### 7.构建容器需要创建网络 docker network create --subnet=172.100.100.0/24 go

###### 8.关闭防火墙：

###### systemctl status firewalld

###### systemctl stop firewalld

###### 防火墙开机不启动systemctl disable firewalld

###### 9.php8容器里的shell界面按<——Backspace删除键时出现了^H^H^H^H^H^H^H^H^H^H^H^H

###### 解决办法：输入stty erase ^H后回车，就恢复正常

###### 10.php容器都安装了composer

###### 二、NGINX配置：

server {

listen 80;

server\_name wx.schoolxt.cn localhost discuz.com 3g.discuz.com bbs.discuz.com;

charset gbk;

#access\_log /var/log/nginx/host.access.log main;

error\_log /usr/share/nginx/mykj/error.log;

root /usr/share/nginx/mykj/cmd;#我的框架项目

# $uri包含请求的文件名和路径，不包含包含“?”或“#”等参数。所以这样代理过去的时候apache

中的php $\_GET\$\_POST是空的

# http://www.alipay.com/alipay/index.html => $uri：/alipay/index.html

#$request\_uri包含请求的文件名和路径及所有参数

# http://www.alipay.com/alipay/index.html# => $uri：/alipay/index.html#

#$query\_string表示 请求路径?后面的所有参数

#以某常规的字符串开头匹配,满足百度编辑器上传效果

location ^~ /static/admin/lib/editor/gbk/php/controller.php{

client\_max\_body\_size 26m;

#注意，如果不生效可以在http{} 中添加

#上视频大小 需要配合php.ini post\_max\_size = 26M || upload\_max\_filesize = 26M

proxy\_pass http://172.100.100.93/$uri?$query\_string; #http://discuz.com/static/admin/lib/editor/php/controller.php?id=1&prm=2

proxy\_set\_header Host $http\_host;

proxy\_set\_header REMOTE\_ADDR $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header Scheme $scheme;

proxy\_set\_header SERVER\_PORT $server\_port;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_http\_version 1.1;

proxy\_set\_header Connection "keep-alive";

}

index index.php index.html;

#autoindex on; #默认是关闭的，目录浏览功能

#php框架只有一个入口文件index.php

location / {

#rewrite ^/$ /Home/Index.html permanent;

#try\_files $uri $uri/ /index.php/Home/Index;#内部重定向循环,因为nginx与apache共享目

录

#nginx与apache共享静态目录(在构造容器的时候挂载同一个的apache上的项目目录)，找不到的文件都会

代理到apache

try\_files $uri $uri/ @apache;

# index index.html index.htm;

# 当输入域名的时候，转向http://xxx.com/index.php/Home/Index

# 标志位 => last 相当于 Apache 里的[L]标记，表示完成 rewrite

# last 如果在 location 下用的话，对于重写后的 URI 会对其所在的server{}重新匹配

location

# break 反之

#-f 如果文件存在，而且是文件

#if (!-f $request\_filename) {

# proxy\_pass http://127.0.0.1:9501;

# }

#如果不存在转向访问首页 index.php

#if (!-f $request\_filename){

# rewrite (.\*) /index.php/Home/Index last;

#}

}

#error\_page 404 /404.html;

# redirect server error pages to the static page /50x.html

#

error\_page 500 502 503 504 /50x.html;

#location = /50x.html {

# root /usr/share/nginx/html;

#}

#注意这里的匹配是apache , 不需要正则重写，只要访问浏览器，nginx不存在的资源都进来apache,

location @apache{

# $query\_string表示 请求路径?后面的所有参数

# $uri包含请求的文件名和路径，不包含包含“?”或“#”等参数。

#比如http://discuz.com/Home/Index.html?id=1&prm=2

#=> $uri：/Home/Home/Index.html

proxy\_pass http://172.100.100.93/index.php$uri?$query\_string;

#http://discuz.com/index.php/Home/Index.html?id=1&prm=2

# 当输入discuz.com 请求时nginx做处理

#转发为：http://172.100.100.93/index.php@php?no=1

#所以翻墙其实很容易，只要用nginx这样的代理就行了，前提是你能访问代理

proxy\_set\_header Host $http\_host;

#获取客户端真实IP

proxy\_set\_header REMOTE\_ADDR $remote\_addr;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header REMOTE-HOST $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

#proxy\_set\_header X-Forward-For $remote\_addr;

proxy\_set\_header Scheme $scheme;

proxy\_set\_header SERVER\_PORT $server\_port;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_http\_version 1.1;

proxy\_set\_header Connection "keep-alive";

# IF https

# proxy\_set\_header HTTPS "on";

#proxy\_set\_header Connection $connection\_upgrade;

# proxy\_set\_header X-Forward-For $remote\_addr;

# proxy\_redirect default;

# proxy\_set\_header Host $http\_host:$server\_port;

# proxy\_set\_header Host $host:$proxy\_port;

}

location ~\* ^.+\.(jpg|jpeg|gif|png|swf|rar|zip|css|js)$ {

access\_log off;

expires -1;

break;

}

# proxy the PHP scripts to Apache listening on 127.0.0.1:80

#区分大小写的正则匹配

# location ~ \/([A-Z][a-z]+)\{2\}(/\w\*)\*\.html$ {

# proxy\_pass http://172.100.100.93/index.php$uri;

# }

# pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000

#区分大小写的正则匹配

location ~ \.php$ { //当输入discuz.com时，nginx默认处理 index.php脚本这里会被触发到。这时请求的url资源也是空的。所以重写也匹配到空了又重新跳转为discuz.com/@php然后这样的请求，又触发了上面的配置转向@apache

rewrite ^ /@php; #跳转到一个网站根目录不存的文件，然后在转向@apache

# root /usr/share/nginx/html;

# fastcgi\_pass 172.100.100.90:9000;

# fastcgi\_index index.php;

# fastcgi\_param SCRIPT\_FILENAME /scripts$fastcgi\_script\_name;

# include fastcgi\_params;

}

# deny access to .htaccess files, if Apache's document root

# concurs with nginx's one

#

#location ~ /\.ht {

# deny all;

#}

}

**Apache配置：**

<VirtualHost \*:80>

ServerAdmin webmaster@localhost

DocumentRoot /var/www/html/mykj/cmd

注意：凡是ajax请求都会进入apache, 不管URL访问nginx上存在的静态资源

1. **MYKJ系统目录权限设置：**

**|mykj //根**

**|\_vendor //composer组件目录**

**|\_ cmd //网站根目录**

**|\_index.php //入口地址**

**|\_ght //参考miniui框架案例**

**|\_Home //前台模版**

**|\_Mykj //后台模版**

**|\_static //静态资源**

**|\_app**

**|\_controller**

**|\_Event //观察者、事件、 有队列**

**|\_Helpers //助手函数**

**|\_Middleware //中间件**

**|\_model //数据库操作层，进行数据交互**

**|\_Providers //服务提供者**

**|\_Listeners //监听事件时，需要处理的业务**

**|\_Service // 暂时没有 逻辑层，处理业务逻辑**

**|\_bootstrap //与业务独立,单独创建一个目录如bootstrap放框架核心文件.**

**|\_lib //存放路由等独立业务的功能库**

**|\_Config //配置目录**

**|\_mykj //组件私包仓库目录(开发可以参考这个目录)**

**A.上传目录 777权限 ;**

**B. 日志目录777权限：**cd /mnt/docker/apache-php/web/mykj/logs.txt

cd /mnt/docker/apache-php/web/mykj/cmd/static/upload/image

[root@localhost upload]# chmod 777 -R

### 路由规则：前台或后台与控制器和方法名，首字母必须大写(陀峰命名)

###### 注意url资源是走nginx，还是apache

A.静态资源 ， 走nginx资源， <http://discuz.com/Mykj/index.html> 进入后台

1. 动态资源，走apache资源：[http://xxx.com](http://xxx.com/f)/Mykj后台目录/后台控制器/方法名.

如: 后台ajax访问：http://discuz.com/Mykj/ColumnTree/save.html

动态资源,控制器参数与方法参数不存在的时候：前台 ， 默认控制器 Home ，默认方法Index ; 后台，默认控制器Mykj, 默认方法：Login;

###### 

###### 前台目录 :Home ； 后台目录：Mykj ；

不管前、后台 ，走静态资源都需要在静态页面中发ajax请求，来验证有没有权限。因为不走apache控制不了用户，到底有没有权限进来。

浏览器地址:http://discuz.com/Mykj/Login.html（直接请求nginx服务器上静态资源）

代码url:login.html 没有加/ 就是走浏览器地址的相对路径http://discuz.com/Mykj/

###### 2.GET访问 :方法名后面以HTML格式文件结束，问号后台跟参数

###### 如：/ 前台目录/ 控制器 /方法名

http://bbs.discuz.com/Home/gbk.html?dir=ght

如果方法名不以.html结束，都统一跳转index方法

如果控制器不存在都跳前台控制器Home,Index方法

避免微信拦截：可以加GET参数no ,因为我在中件间设置了

http://wx.schoolxt.cn/?no=1

###### 3.POST访问 :

控制器和方法名，首字母必须大写，方法名后面以HTML格式文件结束

###### 4.AJAX访问 :

A.相对路径不带控制器：方法名，首字母必须大写，方法名后面以HTML格式文件结束

比如浏览器地址：http://bbs.discuz.com/Mykj/Login.html

$.ajax({  
 url:'Login.html',

B..绝对路径带控制器：控制器和方法名，首字母必须大写，方法名后面以HTML格式文件结束

比如浏览器地址：http://bbs.discuz.com/Mykj/Login.html

$.ajax({  
 url:'/Home/Login.html',

注意：后台前端miniui提交的post方法中的key 为param， 如：$\_POST['param']

Param值都是JSON对象，所以都用控制器的基类PostJosnToArr方法统一转换一下，才能变成数组。基类转换了 $\_POST['param'] 转换的变量为：$this->arrPost;

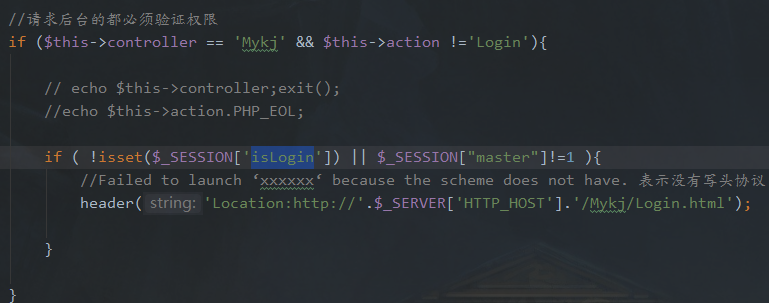
###### 四、访问控制:

0. 后台ajax权验证走的路由用绝对路径

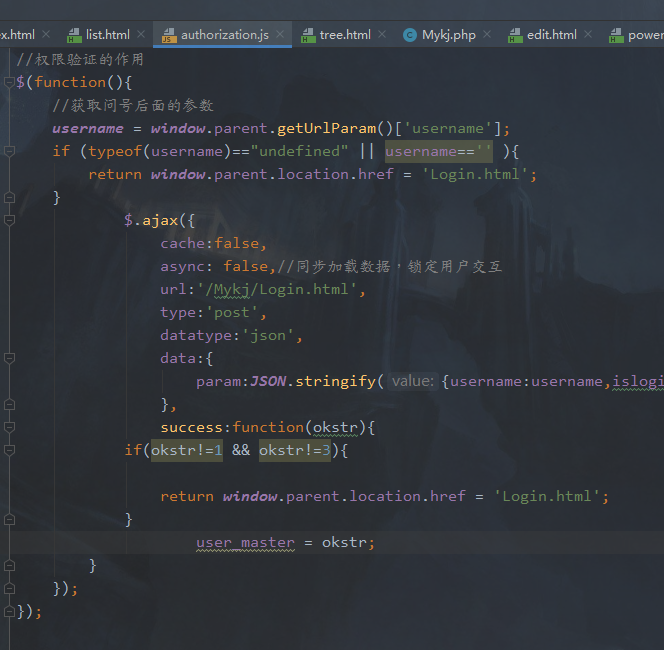
$.ajax({  
 url:'/Mykj/Login.html',

路由程序判断Mykj是后台，Login.html赋值给控制器变量，程序发现控制器‘Login.html’不存在，就会定义默的控制器是Mykj，默认的方法是Login。

1. 凡是请求后台Mykj的，都是在路由类中判断拦截



1. 后台nginx的静态文件都需要添加，ajax请求验证权限代码：



Js，user\_master = 1 代表是超级管理员，具有审核权限，user\_master = 3代表没有审核权限, 特别是在发布信息需不需要审核功能。

### 五、后台地址：

http://discuz.com/Mykj/Login.html

**账号：adminstro**

**密码：654321**

### 六、静态模版

1. 模板目录分为后台、前台目录：

后台模版目录：cmd/Mykj/

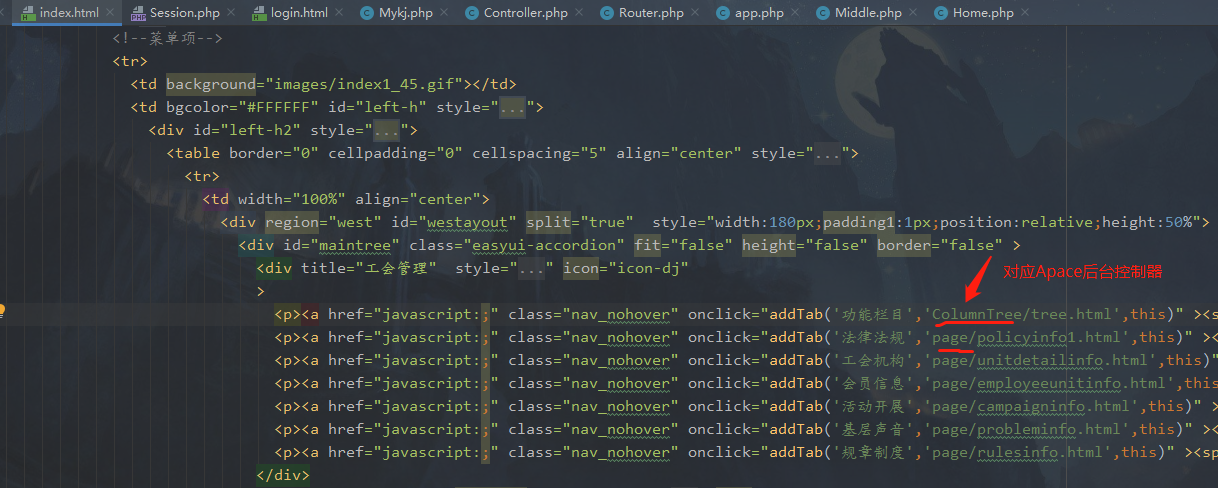
|

——css、images、js、public目录

|

\_\_\_\_xxxx.html 文件（之所以这样放，是因为静态文件方便相对引用图片 js css资源）

注意第一个Mykj是指后台程序文件目录，因要将就URL 的访问控制器到apache，所以请求nginx静态资源也必须定义访问控制器栏目：

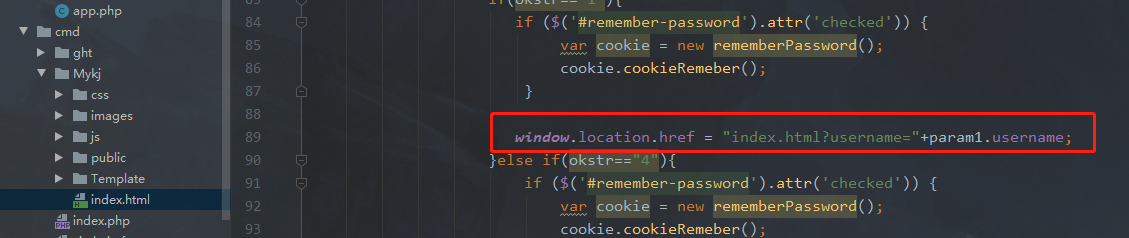


访问http://discuz.com/Mykj/index.html?username=adminstro点击上面菜单‘ColumnTree/tree.html’

下面的跳转，代码用相对’Mykj’路径，表示:

<http://discuz.com/Mykj/ColumnTree/tree.html>

Nginx找到cmd/Mykj/ColumnTree/tree.html资源，tree.html再用ajax请求,http://discuz.com/Mykj/ColumnTree/save.html(nginx不存在这个save.html资源就会去找代理apache)



第一次方问NGINX找不到login.html页面就找代理apache处理，apache执行PHP，导入了login.html页面，然后用户提交ajax处理响应完成后，跳转到nginx处理，nginx找到页面就直接转发给用户。

### 函数：

1. Controller类的PostJosnToArr() : 将前台提交的json数据转化为数组

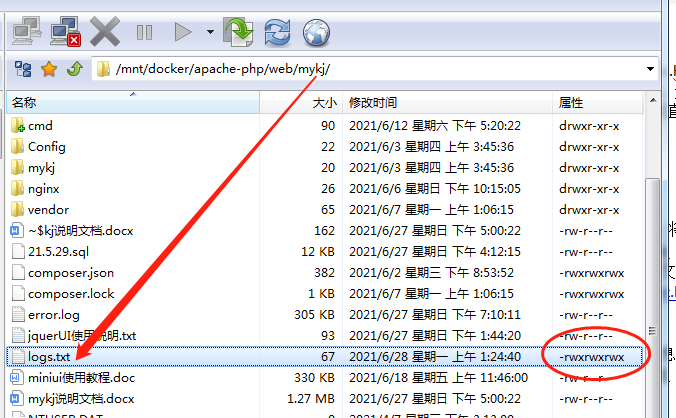
utf8-gbk.php：文件编码转换成gbk

例如：将cmd/ght下面的目录文件转成gbk

<http://bbs.discuz.com/Home/gbk.html?dir=ght>

2.//助手函数logs（），打印客户端信息写入logs.txt文件中

logs( client\_info()); //接收参数为数组



### 常量及全局变量

TIME 当前时间差

BASEDIR 网站入口index.php文件的上级目录

$\_SESSION 存入的键有：openid,uid,master(0普通会员，2律师，1超级管理员，3普通管理员)

### 九、AJAX请求编码问题

**交互过程前景知识了解：**

当我们使用表单按照传统方式post提交时候（非AJAX提交），浏览器会根据当前页面编码，encode一次，然后发送到服务端，服务端接收到表单，会自动dencode一次，通常这个过程是对程序是透明的，因此加上手动编码、解码，就变成上面所说的二次编码。

但当我们使用AJAX方式提交时候，浏览器并不会自动替我们encode，因此在jquery中有这样的一段代码：

1. jQuery.param=**function**( a ) {
2. **var** s = [ ];
3. **function** add( key, value ){
4. s[ s.length ] = encodeURIComponent(key) + '=' + encodeURIComponent(value);
5. };

//jQuery.ajax， jQuery.param方法中encodeURIComponent这方法，这是javascript内置的方法，对目标字符串执行utf-8 encode，因此，当页面使用gbk编码时候，服务端会使用gbk进行解码，但实际提交的数据是以utf-8编码的，所以造成接收到内容为乱码或者为问号。

没错，就是用 escape代替 encodeURIComponent方法，不过必须注意：

escape不编码字符有69个：\*，+，-，.，/，@，\_，0-9，a-z，A-Z

encodeURIComponent不编码字符有71个：!， '，(，)，\*，-，.，\_，~，0-9，a-z，A-Z

使用了 escape之后必须对加号进行编码，否则，当内容含有加号时候会被服务端翻译为空格。

1. 这样实现的：**var** encode=**function**(str){
2. str=escape(str);
3. str=str.replace(/\+/g,"%u002B");
4. **return** str;
5. };
6. **function** add( key, value ){
7. s[ s.length ] = encode(key) + '=' + encode(value);
8. };

**最终我还是采用二次编码(需要固定ajax提交的参数key为param,并且控制器要继承基类控制器Controller)，**将AJAX post的请求都转换成gbk编码：

$arrPost = json\_decode($\_POST['param'],true);//+true的作用是变成数组，否则是json对象

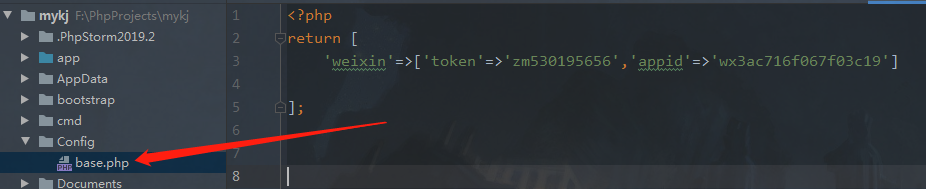
foreach ($arrPost as $k => $v) {

$arrPost[$k] = iconv('utf-8','gbk',$v);

}

### 十、配置文件获取

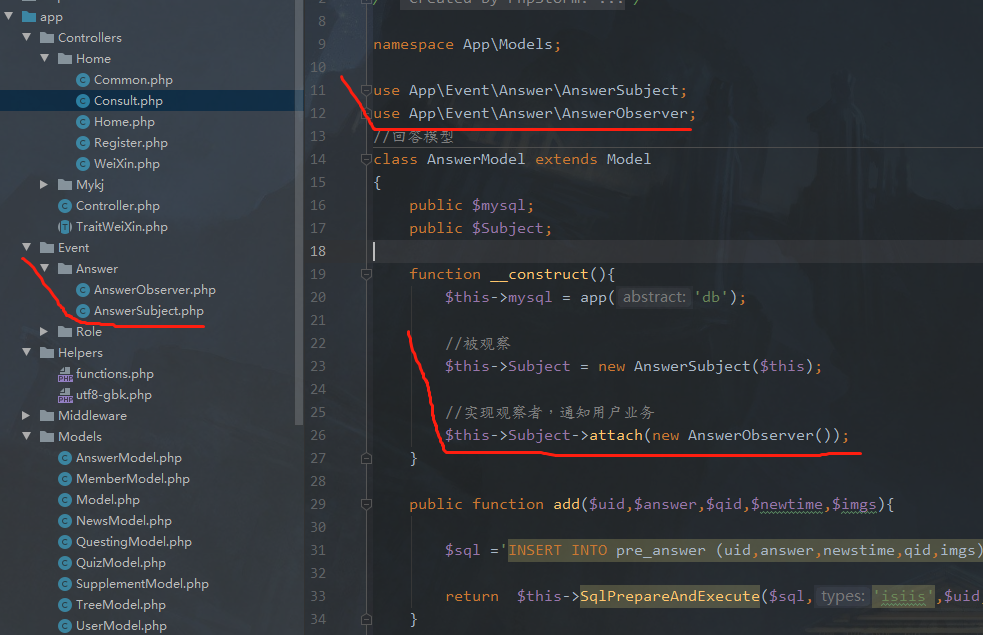
设置：在Config目录中



获取格式：‘文件名.一维数组键名’或 ‘文件名.一维数组键名.二维数组键名’

如：$token = app('config')->get('base.weixin.token');

1. **观察者事件：一般用在表模型的更新前或更新后触发**

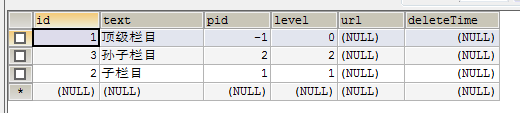


观察者的队列公共类在：

namespace App\Event\Beanstalkd.php;

**数据字典**

功能栏目表

CREATE TABLE `pre\_column` (  
  `id` smallint unsigned NOT NULL AUTO\_INCREMENT,  
  `text` char(8) NOT NULL,  
  `pid` smallint NOT NULL,  
  `level` tinyint NOT NULL,  
  `url` varchar(55) CHARACTER SET gbk COLLATE gbk\_chinese\_ci DEFAULT NULL,  
  `deleteTime` int DEFAULT NULL,  
  PRIMARY KEY (`text`),  
  UNIQUE KEY `id` (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=gbk

**十三、测试**

no=1代表不限制PC访问，否则取掉no=1代理只能在微信端访问。

前端测试：

<http://192.168.40.128/Home/Video?no=1>

<http://discuz.com/?no=1>