# 惠付券部署文档

1. **环境**

**1. 部署环境**

**该项目部署在阿里云服务器上，服务器相关参数如下：**

**内存：2GB**

**带宽：1Mbps**

**IP地址：39.107.254.229**

1. **软件列表**

**Mysql： 5.6.25 64位**

**Nginx： 1.9.9**

**PHP： 5.6.21**

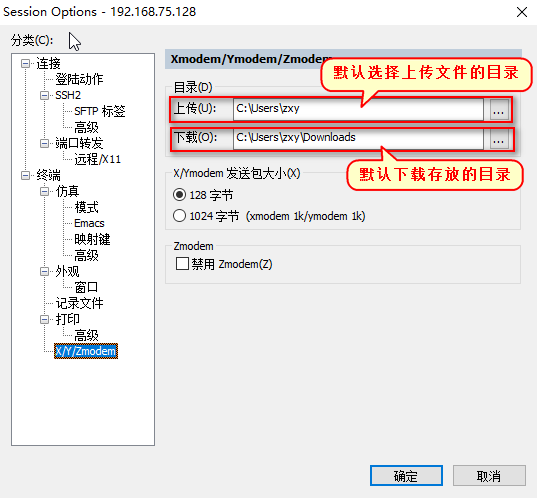
**Memacache : 2.2.0**

1. **安装相关软件**

**1.安装lrzsz**

**在linux上安装lrzsz工具，安装方式如下：安装： yum install lrzsz**

**安装完成之后需要设置windows上传和下载的目录：CTR--->选项--->会话选项**



# **2.安装MYSQL**

1 yum remove mysql mysql-server mysql-libs mysql-common 卸载mysql

rm -rf /var/lib/mysql

rm /etc/my.cnf

/\*[root@pc ~]# cd /usr/local/src/

[root@pc src]# yum -y install bison expat expat-devel ncurses ncurses-devel libaio libaio-devel libc.so.6

[root@pc src]# wget https://cmake.org/files/v3.5/cmake-3.5.2.tar.gz

[root@pc src]# tar -xzf cmake-3.5.2.tar.gz

[root@pc src]# cd cmake-3.5.2

[root@pc cmake-3.5.2]# ./configure

// 在 ./configure 后 可能会报 “cannot find appropriate C++ compiler on this system” 原因是缺少C++编辑器

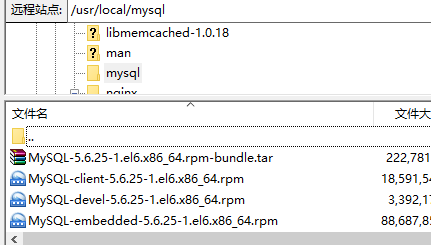
你可以直接输入命令安装：

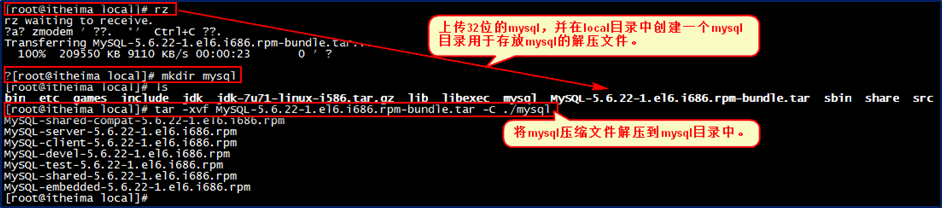
yum install gcc-c++ //

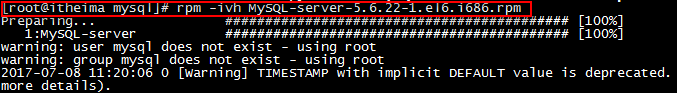
[root@pc cmake-3.5.2]# make

[root@pc cmake-3.5.2]# make install\*/

上传并解压我们的mysql







**安装完mysql服务端之后,开启mysql服务:**

**Service mysql start**

【安装mysql客户端】

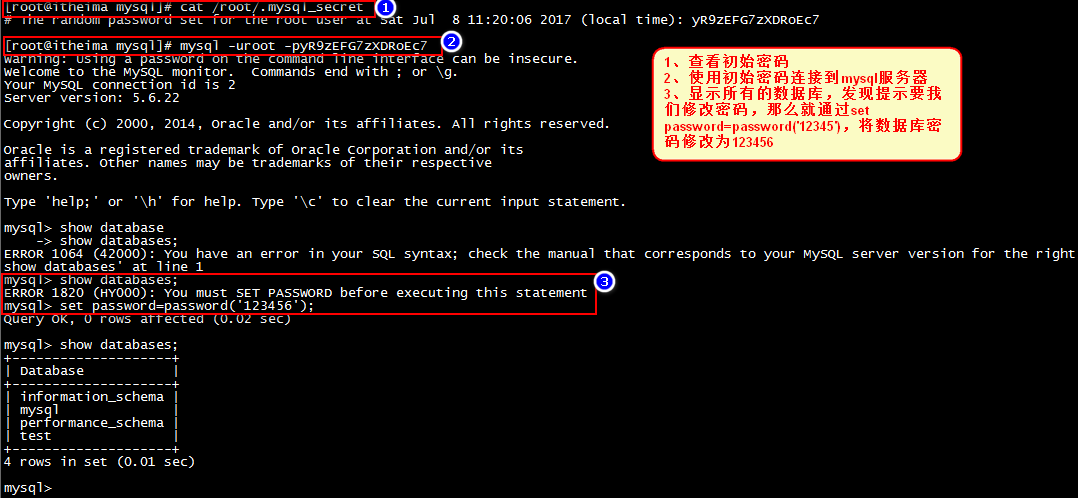


查看初始密码，访问mysql数据库：

cat /root/.mysql\_secret

进入数据库 mysql -uroot -pR30NJZwqCvMvGN35

修改密码 set password=password('123456');



**linux下软件安装-mysql远程访问**

**将mysql服务添加到系统服务并开机启动：**

加入到系统服务：

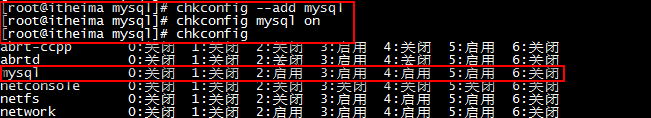
chkconfig --add mysql

自动启动：

chkconfig mysql on

查询列表：

chkconfig



**mysql远程访问：**

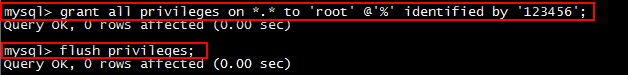
**登录：mysql -uroot –p123456**

**mysql -u root –p**

**设置远程访问（使用root密码）：**

grant all privileges on \*.\* to 'root' @'%' identified by '123456'; //后面的root是你数据库的密码

**flush privileges;**

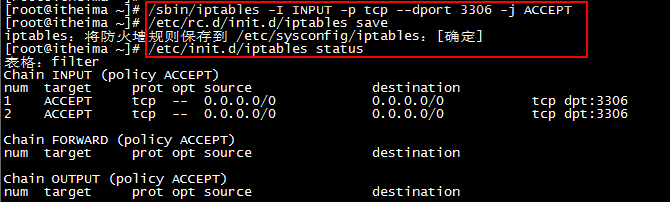


**防火墙打开3306端口：**

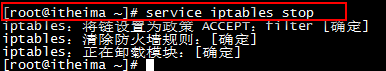
/sbin/iptables -I INPUT -p tcp --dport 3306 -j ACCEPT

/etc/rc.d/init.d/iptables save

/etc/init.d/iptables status



**或者关闭防火墙：service iptables stop**



**打开SQLyog进行远程连接测试：**

chkconfig iptables off 禁止防火墙自启

# **3.安装NGINX**

在安装nginx前首先要确认系统中安装了**gcc、pcre-devel、zlib-devel、openssl-devel**。

yum -y install gcc pcre-devel zlib-devel openssl openssl-devel

[root@pc ~]# cd /usr/local/src/

root@pc src]# wget http://nginx.org/download/nginx-1.9.9.tar.gz

[root@pc src]# tar -xzf nginx-1.9.9.tar.gz

[IMG_256](http://www.cnblogs.com/xxoome/p/javascript:void(0);)

## 解压

tar -zxvf nginx-1.9.9.tar.gz

##进入nginx目录

cd nginx-1.9.9

## 配置  
./configure --prefix=/usr/local/nginx  
  
# make  
make  
make install

[IMG_257](http://www.cnblogs.com/xxoome/p/javascript:void(0);)

OK，现在可以执行make 了。

执行make、make install命令

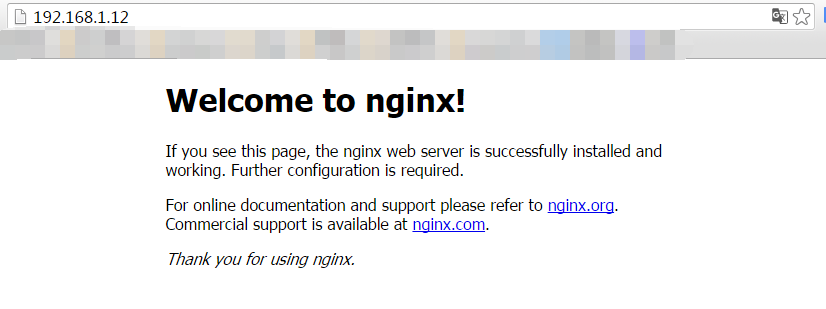
启动nginx

cd /usr/local/nginx/sbin

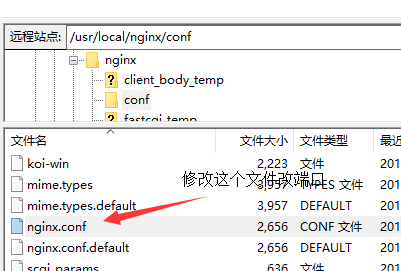
./nginx //启动nginx

在浏览器中输入服务器的ip地址，如：192.168.1.12

刷新浏览器



注意 nginx 默认的是80 端口 改端口需要在



# **4.安装PHP**

安装插件，防止PHP报错

yum -y install libxml2-devel -y

yum -y install libxml2

yum -y install curl-devel

yum -y install libjpeg-devel

yum -y install libpng-devel

yum -y install libpng

yum -y install libXpm-devel

yum install bzip2 bzip2-devel //

yum -y install freetype-devel

yum -y install libxslt-devel

如果已经安装了可能会进行升级，版本完全一致则不会进行任何操作。

yum -y install libjpeg libjpeg-devel libpng libpng-devel libvpx libvpx-devel libXpm libXpm-devel libxml2 libxml2-devel fontconfig fontconfig-devel freetype freetype-devel zlib zlib-devel bzip2 bzip2-devel curl curl-devel openssl openssl-devel

yum -y install bzip2-devel curl-devel freetype-devel gcc libjpeg-devel libpng-devel libxslt-devel libxml2-devel openssl-devel pcre-devel pcre-devel zlib-devel

1. **安装php**

[root@pc ~]# cd /usr/local/src/

[root@pc src]# wget ftp://mcrypt.hellug.gr/pub/crypto/mcrypt/libmcrypt/libmcrypt-2.5.7.tar.gz

[root@pc src]# tar -xzf libmcrypt-2.5.7.tar.gz

[root@pc src]# cd libmcrypt-2.5.7

[root@pc libmcrypt-2.5.7]# ./configure

[root@pc libmcrypt-2.5.7]# make

[root@pc libmcrypt-2.5.7]# make install

[root@pc libmcrypt-2.5.7]# ldconfig

[root@pc libmcrypt-2.5.7]# cd libltdl/

[root@pc libltdl]# ./configure --enable-ltdl-install

[root@pc libltdl]# make

[root@pc libltdl]# make install

[root@pc libltdl]# cd ../../

[root@pc src]# wget http://jaist.dl.sourceforge.net/project/mhash/mhash/0.9.9.9/mhash-0.9.9.9.tar.gz

[root@pc src]# tar -zxf mhash-0.9.9.9.tar.gz

[root@pc src]# cd mhash-0.9.9.9

[root@pc mhash-0.9.9.9]# ./configure

[root@pc mhash-0.9.9.9]# make

[root@pc mhash-0.9.9.9]# make install

[root@pc mhash-0.9.9.9]# cd ../

[root@pc src]# wget http://jaist.dl.sourceforge.net/project/lnmpaio/web/mcrypt/mcrypt-2.6.8.tar.gz

[root@pc src]# tar -xzf mcrypt-2.6.8.tar.gz

[root@pc src]# cd mcrypt-2.6.8

[root@pc mcrypt-2.6.8]# ./configure LD\_LIBRARY\_PATH=/usr/local/lib

[root@pc mcrypt-2.6.8]# make

[root@pc mcrypt-2.6.8]# make install

[root@pc mcrypt-2.6.8]# cd ../

[root@pc src]# wget http://mirror.centos.org/centos/6/os/x86\_64/Packages/libzip-0.9-3.1.el6.x86\_64.rpm

[root@pc src]# rpm -ivh libzip-0.9-3.1.el6.x86\_64.rpm

[root@pc src]# wget http://cn2.php.net/distributions/php-5.6.21.tar.gz

[root@pc src]# tar -xzf php-5.6.21.tar.gz

[root@pc src]# cd php-5.6.21

[root@pc php-5.6.21]# echo '/usr/local/lib' >> /etc/ld.so.conf.d/local.conf

[root@pc php-5.6.21]# echo '/usr/local/lib64' >> /etc/ld.so.conf.d/local.conf

[root@pc php-5.6.21]# ldconfig

**3、配置安装变量**

./configure \--prefix=/usr/local/php-5.6.21 \--with-config-file-path=/usr/local/php-5.6.21/etc \--enable-fpm \--enable-mysqlnd \--with-mysqli\--with-pdo-mysql\--with-libxml-dir \--enable-zip \--with-zlib \--with-zlib-dir \--with-curl \--with-mcrypt \--with-gd \--enable-gd-native-ttf \--with-openssl \--with-mhash \--with-xmlrpc \--with-jpeg-dir \--with-png-dir \--with-xpm-dir \--with-freetype-dir \--enable-shared \--enable-xml \--disable-rpath \--enable-bcmath \--enable-shmop \--enable-sysvsem \--enable-inline-optimization \--enable-mbregex \--enable-mbstring \--enable-pcntl \--enable-sockets \--enable-soap

“/usr/local/php-5.6.21”是安装路径，可以改成自己喜欢的安装路径。

**4、编译源码**

# 在解压目录执行编译命令：make  
cd php-5.6.21

make

**5、安装php**

make install

**6、配置PHP**

[root@pc php-5.6.21]# cd /usr/local/php-5.6.21/etc

[root@pc etc]# cp /usr/local/src/php-5.6.21/php.ini-development php.ini

[root@pc etc]# cp php-fpm.conf.default php-fpm.conf

root@pc etc]# vi php-fpm.conf

pm.max\_children = 40

pm.start\_servers = 20

pm.min\_spare\_servers = 5

pm.max\_spare\_servers = 35

root@pc etc]# vi php.ini

date.timezone = 'PRC'

error\_log = /data/logs/php\_errors.log

[root@pc etc]# /usr/local/php-5.6.21/sbin/php-fpm

[root@pc etc]# /usr/local/php-5.6.21/bin/php -v

PHP 5.6.21 (cli) (built: May 7 2016 15:29:00)

Copyright (c) 1997-2016 The PHP Group

Zend Engine v2.6.0, Copyright (c) 1998-2016 Zend Technologies

配置php.ini “cgi.fix\_pathinfo=0” 如果后续有问题改为1

启动php-fpm服务：

/usr/local/php/sbin/php-fpm

//惠圈启动php-fpm

（/usr/local/php-5.6.21/sbin/php-fpm -c /usr/local/php-5.6.21/etc/php.ini -y /usr/local/php-5.6.21/etc/php-fpm.conf）

接下来编辑一个测试的php程序，在nginx下的html目录下创建test.php文件，打印一下php配置：

<?php

phpinfo();

?>

　　然后打开浏览器输入对应的地址进行访问，看到输出页面，说明nginx和php都配置成功了：

/\*在/usr/local/nginx/conf/添加一个

**pathinfo.conf配置文件内容如下**

set $real\_script\_name $fastcgi\_script\_name;

if ($fastcgi\_script\_name ~ "(.+?\.php)(/.\*)") {

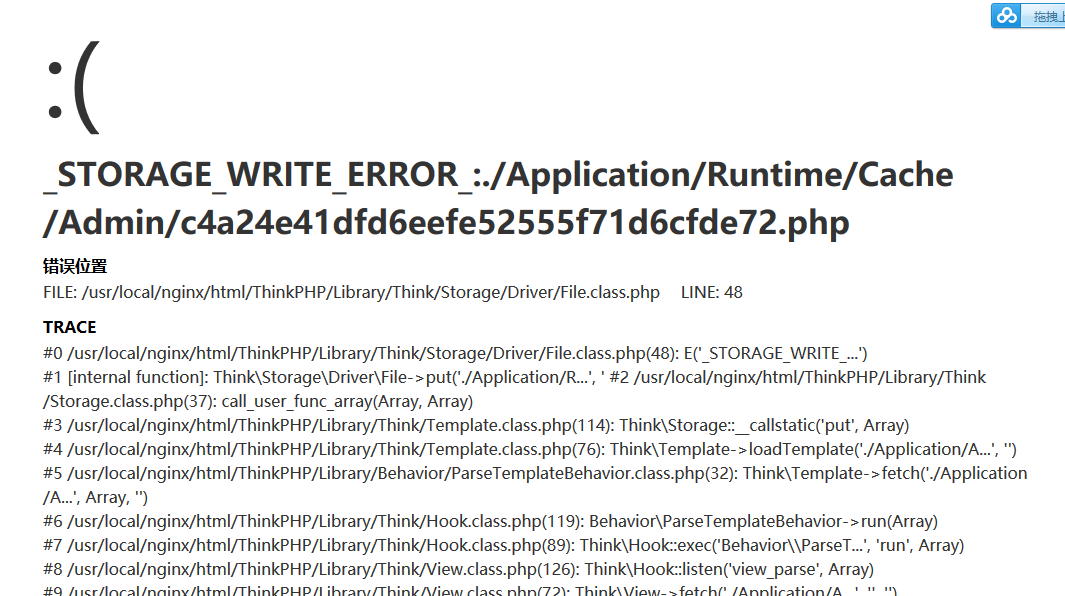
set $real\_script\_name $1;

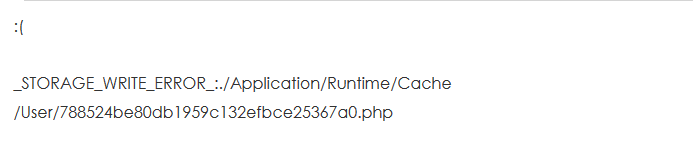
set $path\_info $2;

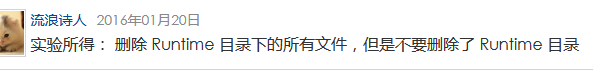
}

fastcgi\_param SCRIPT\_NAME $fastcgi\_script\_name;

fastcgi\_param PATH\_INFO $path\_info;\*/ 自己配置没用上









Nginx.conf配置参考以前惠圈配置

如果进入项目出现缺少pdo\_mysql

**在PHP源码包中进入pdo\_mysql**

cd ext/pdo\_mysql

执行/usr/local/php-5.6.21/bin/phpize



make

Make install

成功则出现  
Installing shared extensions:     /usr/local/php/lib/php/extensions/no-debug-non-zts-20060613/  
说明的/usr/local/php//lib/php/extensions/no-debug-non-zts-20060613/目录下生成了pdo\_mysql.so文件  
12.在php配置文件中加上  
extension=”/usr/local/php/lib/php/extensions/no-debug-non-zts-20060613/pdo\_mysql.so”

重启nginx php-fpm

# **4.Memcache配置**

[root@pc ~]# cd /usr/local/src/

root@pc src]# wget https://launchpad.net/libmemcached/1.0/1.0.18/+download/libmemcached-1.0.18.tar.gz

[root@pc src]# tar -xzf libmemcached-1.0.18.tar.gz

[root@pc src]# cd libmemcached-1.0.18

[root@pc libmemcached-1.0.18]# ./configure --prefix=/usr/local/libmemcached-1.0.18

[root@pc libmemcached-1.0.18]# make

[root@pc libmemcached-1.0.18]# make install

[root@pc libmemcached-1.0.18]# cd ../

[root@pc src]# wget http://pecl.php.net/get/memcached-2.2.0.tgz

[root@pc src]# tar -xzf memcached-2.2.0.tgz

[root@pc src]# cd memcached-2.2.0

[root@pc memcached-2.2.0]# /usr/local/php-5.6.21/bin/phpize

[root@pc memcached-2.2.0]# ./configure --enable-memcached \

--with-php-config=/usr/local/php-5.6.21/bin/php-config \

--with-libmemcached-dir=/usr/local/libmemcached-1.0.18

checking sasl/sasl.h usability... no

checking sasl/sasl.h presence... no

checking for sasl/sasl.h... no

configure: error: no, sasl.h is not available. Run configure with --disable-memcached-sasl to disable this check

[root@pc memcached-2.2.0]# yum -y install cyrus-sasl-devel

[root@pc memcached-2.2.0]# /usr/local/php-5.6.21/bin/phpize

[root@pc memcached-2.2.0]# ./configure --enable-memcached --disable-memcached-sasl \

--with-php-config=/usr/local/php-5.6.21/bin/php-config \

--with-libmemcached-dir=/usr/local/libmemcached-1.0.18

[root@pc memcached-2.2.0]# make

[root@pc memcached-2.2.0]# make install

[root@pc memcached-2.2.0]# vi /usr/local/php-5.6.21/etc/php.ini

extension=memcached.so

[root@pc memcached-2.2.0]# service php-fpm restart

[root@pc memcached-2.2.0]# /usr/local/php-5.6.21/bin/php -i |grep memcached

[root@pc memcached-2.2.0]# cd ../

[root@pc src]# wget http://pecl.php.net/get/memcache-2.2.7.tgz

[root@pc src]# tar -xzf memcache-2.2.7.tgz

[root@pc src]# cd memcache-2.2.7

[root@pc memcache-2.2.7]# /usr/local/php-5.6.21/bin/phpize

[root@pc memcache-2.2.7]# ./configure --with-php-config=/usr/local/php-5.6.21/bin/php-config

[root@pc memcache-2.2.7]# make

[root@pc memcache-2.2.7]# make install

[root@pc memcache-2.2.7]# vi /usr/local/php-5.6.21/etc/php.ini

extension=memcache.so

[root@pc ~]# cd /usr/local/src/

[root@pc src]# wget https://github.com/downloads/libevent/libevent/libevent-2.0.21-stable.tar.gz

[root@pc src]# tar -xzf libevent-2.0.21-stable.tar.gz

[root@pc src]# cd libevent-2.0.21-stable

[root@pc libevent-2.0.21-stable]# ./configure -prefix=/usr

[root@pc libevent-2.0.21-stable]# make

[root@pc libevent-2.0.21-stable]# make install

[root@pc libevent-2.0.21-stable]# ls -al /usr/lib | grep libevent

[root@pc libevent-2.0.21-stable]# cd ../

[root@pc src]# wget http://memcached.org/files/memcached-1.4.25.tar.gz

[root@pc src]# tar -xzf memcached-1.4.25.tar.gz

[root@pc src]# cd memcached-1.4.25

[root@pc memcached-1.4.25]# ./configure -with-libevent=/usr

[root@pc memcached-1.4.25]# make

[root@pc memcached-1.4.25]# autoreconf -ivf

[root@pc memcached-1.4.25]# make

[root@pc memcached-1.4.25]# make install

[root@pc memcached-1.4.25]# ls -al /usr/local/bin/mem\*

[root@pc memcached-1.4.25]# /usr/local/bin/memcached -d -m 512 -u root -l 127.0.0.1 -p 11211 -c 256 -P /data/logs/memcached.pid //启动memcache

如果出现access deniy 设置php.ini里面

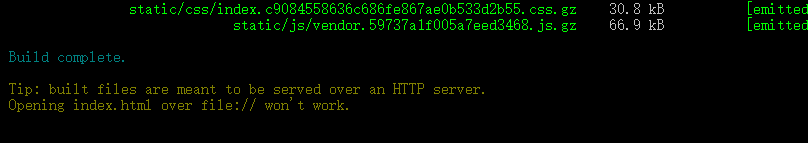
always\_populate\_raw\_post\_data = -1

cgi.fix\_pathinfo=1

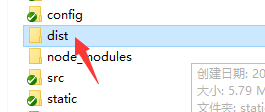
# **前端项目部署**

1.将原NPM项目打包

C:\Users\hkp19\Desktop\repository\huiquan\Development\_library\05Coding\coupon-vue-demo>**cnpm run build**



完成

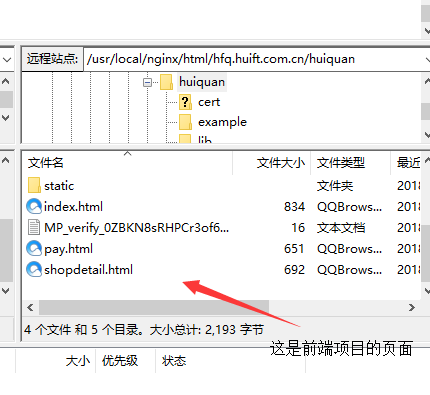




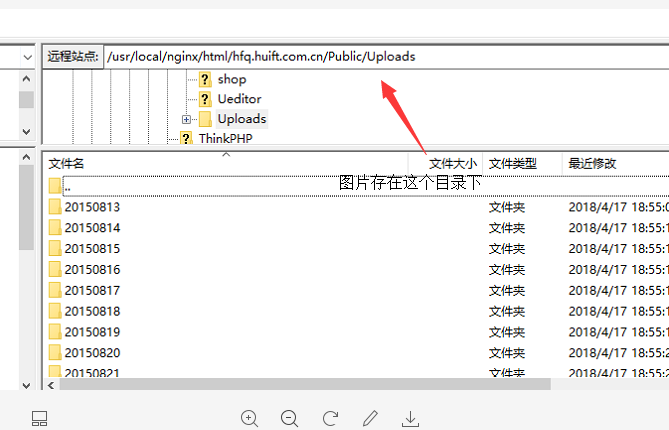
Dist文件夹中有页面和static资源

这是打包完成后的包

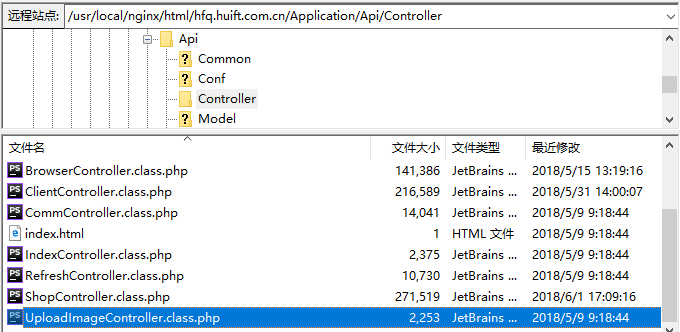
将其丢入nginx中 完成

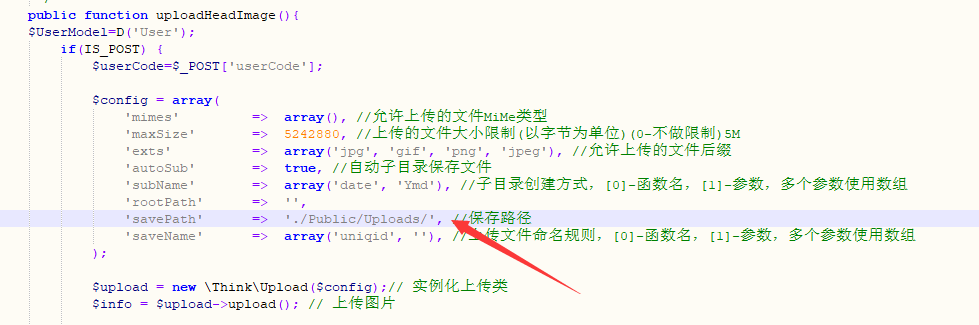


图片的存储路径



在代码中 通过以下配置





通过savePath 参数来配置