1. **环境搭建**

本节内容已写博客： <http://blog.csdn.net/zzy2011266/article/details/40399043>

有网的话直接看博客吧，直接粘过来的排版无法直视。。下面内容是粘贴博客，并修改特定数据为本项目所需。

**一、 本地windows 版 环境搭建**

1. 安装mysql , 这个没啥问题，上官网下一个然后一直下一步直到finish;

2. 下载php, windows 版本的在：<http://windows.php.net/download/> ，根据需要自行选择稳定的版本，注意一个问题是同一版本有不同的型号，5.4之后的都是用VC11编译的，之前的有VC9 和VC6, 这个对windows来说我感觉没多大影响，因为windows下不用自己去编译安装。真正的区别是Thread Safe or Non Thread Safe, 每个版本都有这俩型号，对于Nginx 需要下载的是Non Thread Safe 的版本，对于Apache, IIS 等服 务器是下载Thread Safe版本的。 因为我们项目用到了ThinkPHP3.1 所以至少要下载PHP5.3或以上版本，我下的是php-5.4.33-nts。ThinkPHP3.2则是要求PHP5.4及以上。

3. 下载Nginx 服务器[,http://nginx.org/en/download.html](http://nginx.org/en/download.html" \t "_blank) , 下载一个稳定版本，我下的是nginx-1.6.2。

       上两个文件我都解压到了D:/

 4. php.ini 配置

       PHP安装目录下有俩配置文件：php.ini.devlelopment 和 php.ini.production,我用的是第一个，把名字中的development 去掉，变成默认的配置文件php.ini. 这个文件里面分号——“;”表示注释。所以你要开启某个选项只要去掉前面的分号就行，我主要改了几个必须的选项：

error\_reporting = E\_ALL  ;//开发过程中让他开启，生产环境中注释掉

display\_errors = On

extension\_dir = "D:/php-5.4.33-nts/ext" ;//这个是必须要的，因为有些extension 的dll 文件得上这找

cgi.fix\_pathinfo=1  ; //因为我们使用nginx 是以FastCGI 的方式，所以得开启这个选项

extension=php\_mysql.dll;//用到了mysql ， 这俩动态扩展必须添加

extension=php\_mysqli.dll；

extension=php\_gd2.dll；//用到了GD库来产生验证码，这个得加上

;//以下是常用的一些extension, 都打开

extension=php\_pdo\_mysql.dll

extension=php\_pdo\_sqlite.dll

extension=php\_bz2.dll

extension=php\_curl.dll

extension=php\_fileinfo.dll

extension=php\_gd2.dll

extension=php\_gettext.dll

extension=php\_gmp.dll

extension=php\_intl.dll

extension=php\_imap.dll

extension=php\_sockets.dll

extension=php\_soap.dll

5. nginx.conf 配置

进入nginx 安装目录的conf 文件夹（D:\nginx-1.6.2\conf）找到这个配置文件，#是注释符，如果你没有复杂的需求，官网上的文档基本就可以满足需求：<http://nginx.org/en/docs/beginners_guide.html> 。

我们这个项目比较复杂，一个服务器里跑了两种不同URL Mapping的应用，所以多写了一些东西。配置文件的主体结构是这样的：

http {

server {

}

。。。 #可以有多个server, 监听不同的端口的应用

}

我们改的基本都在Server 模块中

server {  
        listen       80;  
        server\_name  localhost;  
 access\_log    logs/xx.access.log;  
        error\_log logs/xx.error.log;  
        rewrite ^(.+)/$ $1;  
  
 location / {  
            root   D:/nginx-1.6.2/data/www;  
            index  index.php;  
  
 if (!-e $request\_filename) {  
                  rewrite ^(.\*)$ /index.php$1 last;  
                  break;  
              }  
         }  
   
 location ~ .+\.php($|/) {  
             root D:/nginx-1.6.2/data/www/;  
 set $script    $uri;  
               
    if ($uri ~ "^(.+\.php)(/.+)") {  
                set $script $1;  
                set $path\_info $2;  
             }  
             fastcgi\_pass   127.0.0.1:9001;  
             fastcgi\_index  index.php;  
             fastcgi\_param  PATH\_INFO   $path\_info;  
             fastcgi\_param  SCRIPT\_FILENAME  $document\_root$fastcgi\_script\_name;  
             include        fastcgi\_params;  
             break;  
         }  
}

要注意的几个地方：

1. 日志， nginx 在页面上基本显示的东西不够详细，访问日志和错误日志可以详细的看到错的地方；

2. rewrite 加上url 的重写规则的正则表达式

3. location / {

root  //代表服务器的根目录，自己找个地儿吧，index 设置成index.php

server name  //代表服务器的地址，如果写的是域名的话，通过IP是不能访问的

4. if(!-e $request\_filename)重写规则

5. localtion ~ .+\.php { // 所有的php 脚本文件的请求都使用php-cgi来代理解释执行，所以运行时得启动php-cgi.exe

   fastcgi\_pass   127.0.0.1:9000;  // FastCGI默认监听9000端口

   fastcgi\_index  index.php;

   fastcgi\_param  SCRIPT\_FILENAME  $document\_root$fastcgi\_script\_name; //把 //$script改成这个，前面不用加'/'

6. 启动服务器和FastCGI

命令行进入PHP根目录,输入：php-cgi.exe -b 127.0.0.1:9000 -c php.ini

 进入nginx 根目录点击 nginx. exe , 启动nginx

在URL中输入项目的访问地址即可。

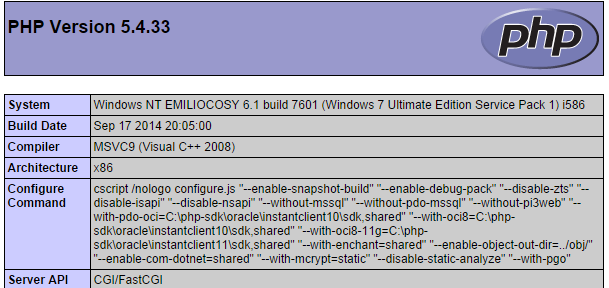
也可以在服务器根目录下创建一个phpinfo.php文件，写上

<?php

phpinfo();//显示PHP各种信息

?>

http://localhost/phpinfo.php能出来这个页面就是搞定了：



为了让今后启动和关闭服务器更加便捷，写两个bat文件:start.bat, stop.bat,需要下载RunHiddenConsole.zip包解压到nginx目录内，RunHiddenConsole.exe的作用是在执行完命令行脚本后可以自动关闭脚本。

***start.bat:***

@echo off

REM Windows 下无效

REM set PHP\_FCGI\_CHILDREN=5

REM 每个进程处理的最大请求数，或设置为 Windows 环境变量

set PHP\_FCGI\_MAX\_REQUESTS=1000

echo Starting PHP FastCGI...

RunHiddenConsole D:/php-5.4.33-nts/php-cgi.exe -b 127.0.0.1:9001 -c D:/php-5.4.33-nts/php.ini

echo Starting nginx...

RunHiddenConsole D:/nginx-1.6.2/nginx.exe -p D:/nginx-1.6.2

***stop.bat:***

@echo off

echo Stopping nginx...

taskkill /F /IM nginx.exe > nul

echo Stopping PHP FastCGI...

taskkill /F /IM php-cgi.exe > nul

exit

以后双击这俩文件就行了。

**二、 服务器端linux 版 环境搭建**

 服务器Linux系统是CentOS的，yum库比较旧，php5.3是最新， 所以得找其他的源来安装php5.4以上。此处要注意一个问题，**最好最好先安装mysql, 然后再安装php,** 不然要很悲剧的要再编译生成msql.so (等价于windows 中的php\_mysql.dll，如果不幸你丫已经先装了php 请先看：[http://www.cnblogs.com/wangtao\_20/archive/2011/03/07/1973621.html](http://www.cnblogs.com/wangtao_20/archive/2011/03/07/1973621.html" \t "_blank))

**1. 源码安装mysql 5.6**

--先下载mysql5.6的源码：

下载地址：[http://dev.mysql.com/downloads/mysql/](http://dev.mysql.com/downloads/mysql/" \t "_blank)  选择source code我们用的是Generic Linux   31.5M那个。

-- 源码安装

教程一：

<http://blog.163.com/liyinhui20080527@126/blog/static/815232582013885310900/>

这个帖子是找到的最佳的解决方案，注意在Linux下先添加mysql 用户，然后在数据库中使用该用户。

粘贴该贴内容，以便无网查看：

Linux操作系统：[CentOS](http://www.linuxidc.com/topicnews.aspx?tid=14" \o "CentOS" \t "_blank) 6.3

1：下载：当前mysql版本到了5.6.10  
   
下载地址：[http://dev.mysql.com/downloads/mysql/5.6.html#downloads](http://dev.mysql.com/downloads/mysql/5.6.html" \l "downloads" \t "_blank)  
   
选择“Source Code”  
   




在此之前最好注册一个[Oracle](http://www.linuxidc.com/topicnews.aspx?tid=12" \o "Oracle" \t "_blank)账号  
   
2：必要软件包

1. yum -y install  gcc gcc-c++ gcc-g77 autoconf automake zlib\* fiex\* libxml\* ncurses-devel libmcrypt\* libtool-ltdl-devel\* make cmake

3:编译安装

1. [root@server182 ~]# groupadd mysql
2. [root@server182 ~]# useradd -r -g mysql mysql
3. [root@server182 ~]# tar -zxvf mysql-5.6.10.tar.gz
4. [root@server182 ~]# cd mysql-5.6.10
5. [root@server182 mysql-5.6.10]# cmake .
6. [root@server182 mysql-5.6.10]# make && make install
7. -------------------------默认情况下是安装在/usr/local/mysql
8. [root@server182 ~]# chown -R mysql.mysql /usr/local/mysql
9. [root@server182 ~]# cd /usr/local/mysql/scripts
10. [root@server182 ~]# ./mysql\_install\_db --user=mysql --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data
11. [root@server182 ~]# cd /usr/local/mysql/support-files
12. [root@server182 support-files]# cp mysql.server /etc/rc.d/init.d/mysql
13. [root@server182 support-files]# cp my-default.cnf /etc/my.cnf
14. [root@server182 ~]# chkconfig -add mysql
15. [root@server182 ~]# chkconfig mysql on
16. [root@server182 ~]# service mysql start
17. Starting MySQL SUCCESS!
18. [root@server182 support-files]# mysql

-- 导入数据

CREATE USER 'mysql'@'%' INDENTIFIED BY 'mysql'; //创建用户

CREATE DATABASE bell;//创建数据库

USE bell;

SOURCE bell.sql; //导入数据

GRANT ALL ON bell.\* TO 'mysql'@'%' ;//授权该用户在任何机器都可以访问数据库bell.

**2.编译安装PHP5.5 & Nginx**

这个教程帖子写的好的我已经无以复加了，直接贴上来吧：

教程二： <http://www.woyaohuijia.cn/show/58.html>

以下为粘贴内容：

1.安装依赖包  
  
yum -y install gcc make gd-devel libjpeg-devel libpng-devel libxml2-devel bzip2-devel libcurl-devel libaio  
  
2.安装php5.5  
wget http://cn2.php.net/get/php-5.5.10.tar.gz/from/cn2.php.net/mirror  
tar -zxvf mirror  
  
cd php-5.5.10  
  
./configure \  
--prefix=/usr/local/php5 \  
--with-config-file-path=/usr/local/php5/etc \  
--with-bz2 \  
--with-curl \  
--enable-ftp \  
--enable-sockets \  
--disable-ipv6 \  
--with-gd \  
--with-jpeg-dir=/usr/local \  
--with-png-dir=/usr/local \  
--with-freetype-dir=/usr/local \  
--enable-gd-native-ttf \  
--with-iconv-dir=/usr/local \  
--enable-mbstring \  
--enable-calendar \  
--with-gettext \  
--with-libxml-dir=/usr/local \  
--with-zlib \  
--with-pdo-mysql=mysqlnd \  
--with-mysqli=mysqlnd \  
--with-mysql=mysqlnd \  
--enable-dom \  
--enable-xml \  
--with-libdir=lib64 \  
--enable-pdo \  
--enable-fpm  
  
编译过程有可能报如下错误：  
  
checking for known struct flock definition… configure: error: Don’t know how to define struct flock on this system, set –enable-opcache=no  
  
解决方法：  
  
yum groupinstall "Development tools"  
  
  
  
说明：以上配置根据自己需要选择就行，如果使用nginx+php，就必须使用php-fpm。  
  
make  
  
make install  
  
  
  
3.配置php5.5  
  
cp php.ini-production /usr/local/php5/etc/php.ini  #php配置文件  
  
cp /usr/local/php5/etc/php-fpm.conf.default /usr/local/php5/etc/php-fpm.conf #php-fpm配置文件  
  
  
  
4.启动php-fpm  
  
/usr/local/php5/sbin/php-fpm  
  
查看是否启动成功：  
   
netstat -lnt | grep 9000  
  
tcp 0 0 127.0.0.1:9000 0.0.0.0:\* LISTEN  
  
或者使用如下命令，查看是否9000端口被php-fpm占用：  
  
netstat -tunpl | grep 9000  
  
tcp        0      0 127.0.0.1:9000              0.0.0.0:\*                   LISTEN      2124/php-fpm  
  
  
  
vi /usr/local/php5/etc/php-fpm.conf  
  
打开pid = run/php-fpm.pid前的注释，这样fpm的进程就会被写入这个文件：/usr/local/php5/var/run/php-fpm.pid  
  
  
  
php-fpm 关闭：  
  
kill -INT `cat /usr/local/php5/var/run/php-fpm.pid`  
  
或者：pkill php-fpm

**3. 配置php.ini 与nginx.conf**

  -- php.ini基本跟windows类似，只是所有的.\*dll文件在linux 下是\*.so 文件基本集成到了php中，不用手动添加到自启动

error\_reporting = E\_ALL  ;//开发过程中让他开启，生产环境中注释掉

display\_errors = On

extension\_dir = **"/usr/local/php5/lib/php/extensions/no-debug-non-zts-20121212"**;//这个跟windows区别一下

cgi.fix\_pathinfo=1; //因为我们使用nginx 是以FastCGI 的方式，所以得开启这个选项

-- nginx.conf

跟windows版的配置一模一样即可

4. phpMyAdmin 加入

phpMyAdmin可以在web页面管理数据库，非常方便，配置也非常简单。

--先下载最新的稳定版本PhpMyAdmin；

--然后配置config.inc.php这个文件，修改几个参数

$cfg['Servers'][$i]['auth\_type']= 'http';  //http not cookie

$cfg['Servers'][$i]['host']= 'localhost';

$cfg['Servers'][$i]['connect\_type']= 'tcp';

$cfg['Servers'][$i]['compress']= false;

$cfg['Servers'][$i]['AllowNoPassword']= true; //allow no password;

--将项目放置到nginx服务器目录下，我是放在了/nginx/html的目录下：

然后配置nginx.conf 将phpmyadmin 加入到服务器中运行，因为项目bell有重写规则，跟phpmyadmin的URL访问冲突，故我在配置文件中新加入一个server, 端口改成8888，让服务器也监听这个端口，而不受另一个bell URL重写的影响。

server {  
        listen      8888;  
        server\_name  localhost;  
        access\_log /var/log/phpmyadmin\_access.log;  
        error\_log /var/log/phpmyadmin\_error.log;

        location / {  
           root  /usr/share/nginx/html;  
           index  index.php index.html index.htm;  
        }  
  
        #particular setting for phpmyadmin visiting  
        location /phpmyadmin {  
            root /usr/share/nginx/html;  
            index index.php;  
  
           location ~ \.php$ {  
             fastcgi\_pass 127.0.0.1:9000;  
             fastcgi\_index index.php;  
       include fastcgi.conf;  
         
         }  
       }  
      error\_page   500 502 503 504  /50x.html;  
      location = /50x.html {  
           root  /usr/share/nginx/html;  
      }

}

**5. 启动服务**

service nginx start

/usr/local/php5/sbin/php-fpm

查看端口是否在监听

netstat -lntp | grep 9000 //看php-cgi 是否启动， nginx 监听的是80端口

每次修改完配置文件都要重启服务：

service nginx restart

pkill php-fpm

/usr/local/php5/sbin/php-fpm

1. **本项目配置相关**
2. 如果修改了数据库用户名和密码，则bell 和bellAdmin的下列文件要修改：

vi /data/www/servers/bell/bellAdmin/System/Conf/conf.php

修改

'DB\_USER' => 'mysql',

'DB\_PWD' => 'xxx',

vi /data/www/servers/bell/mysql.php

修改

'DB\_USER' => 'mysql',

'DB\_PWD' => 'xxx',

1. 各个配置文件及日志文件在测试服务器中的位置：

//nginx.conf 配置文件

vi /etc/nginx/nginx.conf

//php & php-fpm 配置文件

cd /usr/local/php5/etc/

php-fpm.conf 和php.ini 都在这个目录下

//phpMyAdmin配置文件

vi /etc/phpMyAdmin/config.inc.php

//日志文件

nginx日志：

cd /var/log/nginx

phpMyAdmin日志:

vi /var/log/phpmyadmin\_access.log

vi /var/log/phpmyadmin\_error.log

php-fpm日志:

cd /usr/local/php5/var/log

1. 安装目录

mysql: cd /usr/local/mysql

nginx: cd /usr/share/nginx

php: cd /usr/local/php5

phpMyAdmin: cd /usr/share/nginx/html/phpmyadmin

1. 项目目录

闪铃app后台接口项目bell: cd /data/www/servers/bell/bell

闪铃后台运营平台bellAdmin: cd /data/www/servers/bell/bellAdmin

***附录一： 服务器上nignx.conf***

user nginx;

worker\_processes 1;

error\_log /var/log/nginx/error.log;

#error\_log /var/log/nginx/error.log notice;

#error\_log /var/log/nginx/error.log info;

pid /var/run/nginx.pid;

events {

worker\_connections 1024;

}

http {

include /etc/nginx/mime.types;

default\_type application/octet-stream;

# log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '

# '$status $body\_bytes\_sent "$http\_referer" '

# '"$http\_user\_agent" "$http\_x\_forwarded\_for"';

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

sendfile on;

#tcp\_nopush on;

#keepalive\_timeout 0;

keepalive\_timeout 65;

client\_max\_body\_size 10M;

gzip on;

server {

listen 8888;

server\_name localhost;

access\_log /var/log/phpmyadmin\_access.log;

error\_log /var/log/phpmyadmin\_error.log;

location / {

root /usr/share/nginx/html;

index index.php index.html index.htm;

}

#particular setting for phpmyadmin visiting

location /phpmyadmin {

root /usr/share/nginx/html;

index index.php;

location ~ \.php$ {

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php;

#fastcgi\_param PATH\_INFO $path\_info;

#fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

#include fastcgi\_params;

include fastcgi.conf;

#break;

}

}

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

location = /50x.html {

root /usr/share/nginx/html;

}

}

server {

listen 80;

server\_name localhost;

access\_log /var/log/nginx/bell\_access.log;

error\_log /var/log/nginx/bell\_error.log;

#add rewrite rules

rewrite ^(.+)/$ $1;

#basic location setting for projects under server root:

location / {

root /data/www/servers/bell;

index index.php;

if (!-e $request\_filename) {

rewrite ^(.\*)$ /index.php$1 last;

break;

}

location ~ .+\.php($|/) {

root /data/www/servers/bell;

set $script $uri;

if ($uri ~ "^(.+\.php)(/.+)") {

set $script $1;

set $path\_info $2;

}

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php;

fastcgi\_param PATH\_INFO $path\_info;

fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

include fastcgi\_params;

break;

}

}

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

location = /50x.html {

root /usr/share/nginx/html;

}

}

}

***附录二：服务器上php.ini，只改了几个地方***

display\_startup\_errors = Off

log\_errors = On

include\_path = "/usr/local/php5/lib/php"

cgi.fix\_pathinfo=1

zend\_extension =opcache.so