**项目后台框架配置**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 作者 | 日期 | 修改背景色 |
| V1.0 | 黄明 | 2013-8-15 | 无 |
| V1.1 | 黄明 | 2013-8-28 | 蓝色 |
| V1.2 | 黄明 | 2013-9-3 | 无，在最后添加在apache上部署django项目的方法 |
| V1.3 | 黄明 | 2013-10-22 | 无，在最后增加crontab相关内容 |

代码地址：<https://github.com/kobe6672823/appServer.git>

下载后搭建步骤如下（注意要装有django和mysql server和south）：

South的安装方法：sudo pip install south（pip的一般用法：<http://alexliyu.blog.163.com/blog/static/16275449620126239949478/>）  
1：首先要修改mysql的默认字符集，防止字符集问题，这一步一定要在创建database之前完成！步骤如下：  
      进入mysql查看当前的编码：  
      mysql> show variables like ‘%server%’;  
     +———————--------+——————---------+  
      | Variable\_name        | Value             |  
     +—————-------------+——————---------+  
      | character\_set\_server| latin1            |  
      | collation\_server     | latin1\_swedish\_ci  |  
      | server\_id            | 0                 |   
     +———————---------+——————---------+  
  
      如果和上面一样，不是utf-8就需要做以下操作  
      先暂停mysql服务：  
      $sudo /etc/init.d/mysql stop  
      修改 /etc/mysql/my.cnf（或者是/etc/my.cnf，如果是centos自带的mysql，应该是/etc/my.cnf这个问题）  
      sudo vi /etc/mysql/my.cnf  
  
      在mysqld下面加上：  
      default-character-set=utf8  
      init\_connect='SET NAMES utf8'  
      保存，退出wq  
  
      重启mysql服务：  
      $sudo /etc/init.d/mysql start （如果出错就修改my.cnf权限755）  
  
      再次打开mysql查看编码：  
      mysql> show variables like '%server%';  
      +----------------------+-----------------+  
       | Variable\_name        | Value           |  
      +----------------------+-----------------+  
       | character\_set\_server | utf8            |  
       | collation\_server     | utf8\_general\_ci |  
       | server\_id            | 0               |  
      +----------------------+-----------------+  
       mysql的字符集问题就解决了  
  
2：在mysql中创建‘appServerDB’这个database（命令：create database appServerDB），注意要用root用户登录mysql，而且  
      项目的配置文件（/appServer/appServer/setting.py）中，mysql用户是root，密码为空，如果需要做修改，可以到  
      /appServer/appServer/setting.py中修改  
3：在目录“appServer/”下面，执行：python manage.py syncdb(生成项目相关的表) (若要出现相关提示，则按照提示来进行相应的操作。)

5：在目录“appServer/”下面，执行：python manage.py convert\_to\_south game (在game目录下创建migrations目录以及第一次迁移)

测试建立工程是否成功：  
1：在目录“appServer/”下面，执行: python manage.py runserver  
2：访问 <http://127.0.0.1:8000/admin/>  ，如果上述数据库建立没有问题，在game那一项那里，可以对User, Story,

若出现错误ImportError: No module named requests。则运行sudo pip install requests來安装requests module。

     Chapter进行增删查改，并且尝试一下输入中文，插入中文看有没有问题，如果上面mysql的字符集问题解决了，插入和查询中文测  
     试过是没有问题的  
     开发时需要测试数据的话，也可以用这个admin页面来自己添加测试数据  
~~3：访问~~[~~http://127.0.0.1:8000/qqlogin/~~](http://127.0.0.1:8000/qqlogin/)~~这类“后台接口文档”中规定的接口，看能不能返回正确的信息  
      这些接口都没具体的完成，只是简单返回现在访问的这个接口的名称和参数~~

3：到appServer/mytest目录下，执行那些python，看各个接口能不能正常运行，返回“后台接口”文档中所规定的各项返回值

在apache用mod\_wsgi部署项目的步骤如下：

一、安装mod\_wsgi(参考官方文档：<http://code.google.com/p/modwsgi/wiki/QuickInstallationGuide>)

1：到这里下载源码<http://code.google.com/p/modwsgi/downloads/list>

2：tar xvfz mod\_wsgi-3.4.tar.gz（版本号以实际下到的为准）

3：cd到解压后的文件夹，按顺序执行如下三个命令：./configure， make， make install

4：修改/etc/httpd/conf/httpd.conf（apache的配置文件），添加这句话（LoadModule wsgi\_module modules/mod\_wsgi.so

），需要特别注意的是， 需要在150行左右的地方添加这句话（加动态库都在这部分，加在别的部分无效），可以搜索“LoadModule”的位置，然后在这附近添加

5：sudo service httpd restart，重启apache就可以了

二、部署django项目（参考官方文档：<https://docs.djangoproject.com/en/1.5/howto/deployment/wsgi/modwsgi/>）

需要特别注意，将项目文件夹拷贝到apache的DocumentRoot下（即/var/www/html）

下面所做的修改，都是针对DocumentRoot目录下的项目的！

对应于我们这个项目，很简单，修改apache配置文件：/etc/httpd/conf/httpd.conf

在line245（即section1的结尾处），添加如下代码：

|  |
| --- |
| WSGIScriptAlias / /var/www/html/appServer/appServer/wsgi.py  WSGIPythonPath /var/www/html/appServer  <Directory /var/www/html/appServer>  #<Files wsgi.py>  Order allow,deny  Allow from all  #</Files>  </Directory>  Alias /static /var/www/html/appServer/appServer/static/  <Location "/static">  SetHandler None  </Location> |

然后sudo service httpd restart，重启apache就可以了。（重启apache可能有问题，如果有问题就完成下面步骤三的内容，就没问题了，因为上面贴出来的添加的代码是完成步骤三后的apache配置文件）

三：部署在apache后，admin页面没有css相关的内容

参考：<http://obroll.com/solving-admin-css-not-loaded-or-not-found-in-django/>

原因：django不会自动提供css，图片这类静态文件，需要做一些设置，才能让apache这些prodution server提供这类静态文件。照着参考网址里面的步骤做就能让admin页面正常了。

需要添加3个crontab任务：

1：每天晚上3点修改数据库（修改表game\_coauthorsstatistics）

2：每周星期二晚上那个3点备份数据库

3：logrotate

添加这3个crontab任务，需要

vim /etc/crontab

然后添加这两句任务：

~~0 3 \* \* \* root python /home/hm/appServer/scripts/adjustCoauthorStat.py~~

0 3 \* \* \* root python /home/hm/appServer/scripts/adjustCoauthorNum.py

0 3 \* \* 2 root mysqldump -uroot -p --default-character-set=utf8 --opt --extended-insert=false -x appServerDB > /home/hm/appServer/DBbackup/appServerDB\_backup.`date +\%y\%m\%d`.sql

注意路径需要改成脚本在服务器的绝对路径

由于crontab是读到内存当中的，所以修改完/etc/crontab后可能不会马上执行，重新启动一下crond就行（/etc/init.d/crondrestart）

数据库回复方法：

mysql -uroot -p appServerDB < appServerDB\_backup.131217.sql

logrotate用python的logging模块实现了