

蜗牛项目部署方法

蜗牛项目部署方法

蜗牛进销存项目

Linux环境下

正式版1.4部署

旧版本1.0部署

几个版本的配置信息

中文乱码问题解决

AgileOne项目

windows环境下

Linux环境下

Guest项目（Python的Django项目）

蜗牛Boss项目

2.0版（老版本）

3.5版本

DVWA工具配置

蜗牛进销存项目

Linux环境下

#

正式版1.4部署

1. 该项目是基于Java开发的WEB应用，部署运行项目需要首先部署JDK、Tomcat、MySQL等系统环境。
2. 部署完上述环境后需通过浏览器访问 <http://host:host-port>，如果能够成功打开Tomcat欢迎画面，说明系统环境已经部署成功。
3. 将获得的项目文件WoniuSales-20180508-V1.4-bin.war和数据文件woniusales-20180508-V1.4.sql统统复制到Tomcat的webapps目录下。
4. 重启tomcat服务。
5. 在Tomcat的webapps目录下会出现WoniuSales-20180508-V1.4-bin目录，进入该目录下WEB-INF/classes/下，打开db.properties文件编辑，修改内容如下

```
1 db_url=jdbc:mysql://localhost:3306/woniusales?useUnicode=true&characterEncoding=utf8
  # 如果配置数据库时修改了默认端口3306，那么此处需要做相应修改
2 db_username=root # 此处修改为实际有效的数据库用户名
3 db_password= # 此处添加数据库的密码
4 db_driver=com.mysql.jdbc.Driver
```

6. 保存上面的修改后，在终端中首先启动并进入mysql数据库，创建woniusales数据库

```
1 mysql -uroot -p
```

```
1 create database woniusales;
```

7. 输入 `\q` 退出mysql，然后将sql文件导入到数据库中

```
1 mysql -u root -p woniusales < /usr/local/apache-tomcat-8.0.52/webapps/woniusales-20180508-V1.4.sql
```

8. 数据导入成功后，重启Tomcat服务
9. 通过浏览器访问 <http://host:host-port/WoniuSales-20180508-V1.4-bin> 即可。

旧版本1.0部署

1. 确保服务器环境部署成功
2. 将获得的WoniuChild文件夹和数据文件XXX.sql统统复制到Tomcat的Webapps目录下
3. 命令行启动mysql数据库
4. 添加新的mysql用户并授权

```
1 create user denny identified by '123456';
2 grant all privileges on *.* to 'denny'@'localhost' identified by '123456';
```

5. 退出命令行mysql数据库
6. 重新用新创建的denny用户身份命令行启动mysql数据库
7. 创建新的woniuchild数据库

```
1 create database woniuchild;
```

8. 再次退出命令行mysql数据库
9. 执行sql导入命令还原数据库

```
1 mysql -udenny -p woniuchild < /usr/local/tomcat目录名/webapps/XXX.sql # 注意替换tomcat目录名和XXX为实际对应的真实名字
```

10. 数据导入成功后，重启mysql和tomcat服务
11. 通过浏览器即可正常访问

几个版本的配置信息

#

- 1.0版 数据库名字woniuchild，数据库的用户名denny，密码123456；
- 1.1版 数据库名字milor，数据库的用户名denny，密码123456；
- 1.3版 数据库名字milor，数据库的用户名root，密码123456；
- 1.4版 数据库名字woniusales，数据库的用户名root，密码可以自己设置，这个版本可以通过db.properties来配置。

中文乱码问题解决

#

使用XAMPP集成环境安装的mysql数据库，由于使用的版本较低。数据库默认的字符集不是utf8，在导入sql数据文件后，会发现远程访问数据库中文显示为乱码。这种情况需要修改数据库配置文件才能解决。

1. 首先登录数据库用下面的命令查询一下当前数据库的字符集，确认一下是否默认字符集都是latin1

```
1 mysql -uroot -p
```

登录数据库后执行下面的sql语句

```
1 show variables like '%char%';
```

2. 如果查询发现字符集不是utf8，那么需要检查当前数据库的配置文件读取顺序

```
1 mysql --help | grep 'my.cnf'
```

显示结果中cnf文件的读取优先顺序为 由左至右

3. 然后利用查找命令查询当前系统中所有位置的“my.cnf”文件

```
1 find / -name 'my.cnf'
```

4. 用vi命令打开并编辑第二步找到的并且在第一步结果中位置相对最左边的“my.cnf”文件，在文件中**添加**下面的内容

```
1 [client]
2 default-character-set=utf8
3 [mysqld]
4 character-set-server=utf8
5 [mysql]
6 default-character-set=utf8
```

5. 保存文件后，重启mysql服务

```
1 service mysqld start
```

6. 之后重建数据库并导入sql文件，中文即可正确显示。

AgileOne项目

windows环境下

#

1. 首先安装XAMPP环境，注意由于agileone自身的原因，XAMPP必须使用1.6.8版本。
2. 安装完成后将agileone项目文件包复制到xampp目录下的htdocs目录中。
3. 双击桌面的XAMPP启动，勾选apache和mysql服务并启动。
4. 浏览器首先访问 <http://hostname/agileone/install>，点击页面下方的创建数据库和表按钮。
5. 如果提示“AgileOne已经成功安装, 为确保数据安全, 请勿两次安装 ...”，注意检查 agileone\Common\Config\Global.conf 文件中的db.dbname的值，值若不是空，请清空。
6. 正常情况应该显示“已成功创建数据库及表. [点此访问首页](#) (默认登录: admin/admin)”。
7. 至此agileone项目已经配置成功，每次访问 <http://hostname/agileone> 即可。

Linux环境下

#

1. 首先下载并配置XAMPP环境，注意由于agileone自身的原因，XAMPP必须使用1.6.8版本。
2. 配置并成功启动XAMPP后将agileone项目文件包复制到lampp目录下的htdocs目录中。
3. 首先给agileone项目下的Attachment、Cache、Cache/Config、Cache/Data、Cache/Lang、Common/Config/Global.conf等授权写权限。

```
1  chmod 777 /opt/lampp/htdocs/agileone/路径 路径替换为上述几个目录或者文件即可
```

4. 授权完毕需重启XAMPP服务。
5. 浏览器首先访问 <http://hostname/agileone/install> , 点击页面下方的创建数据库和表按钮。
6. 如果提示“AgileOne已经成功安装, 为确保数据安全, 请勿两次安装 ...”, 注意检查 agileone\Common\Config\Global.conf文件中的db.dbname的值, 值若不是空, 请清空。
7. 正常情况应该显示“已成功创建数据库及表. [点此访问首页](#) (默认登录: admin/admin)”。
8. 至此agileone项目已经配置成功, 每次访问 <http://hostname/agileone> 即可。

Guest项目 (Python的Django项目)

1. 配置好python版本后执行pip命令安装Django环境

```
1  pip install Django
```

2. 成功安装后需要安装django-bootstrap3包

```
1  pip install django-bootstrap3
```

3. 继续安装pymysql包

```
1  pip install pymysql
```

4. 解压guest.zip包到python脚本的工作目录下

- 如果本机Django安装的是2.0版本以上的包的话, 还需要配合修改guest中下面两个文件
- 修改guest\sign\models.py

```
1  event = models.ForeignKey(Event)
2  # 修改为
3  event = models.ForeignKey(Event, on_delete=models.CASCADE)
```

- 修改guest\sign\urls.py

```
1  # 文件末尾增加
2  app_name = 'sign'
```

5. 在guest目录位置, 执行下面的命令

```
1  python manage.py runserver
```

6. 运行成功将显示下面的信息

```
1  Django version 2.0.6, using settings 'guest.settings'
2  Starting development server at http://127.0.0.1:8000/
3  Quit the server with CTRL-BREAK.
```

7. 通过浏览器访问 <http://localhost:8000> 即可

蜗牛Boss项目

本项目是基于Java开发的B/S架构系统，因此项目配置前需首先确保服务器中已经配置好了JRE、Tomcat 8以及MySQL环境。

2.0版（老版本）

#

1. 复制WoniuBoss.war和woniuboss2.0.sql到Tomcat的webapps目录
2. 设置服务器的MySQL数据库root用户密码为root（因为WoniuBoss项目代码写死使用该用户名和密码访问数据库woniuboss2.0）。

```
1 mysqladmin -uroot -p旧密码 password 新密码
```

3. MySQL数据库中创建名为woniuboss2.0的数据库（注意名字拼写必须一致，原因同上）。

```
1 create database `woniuboss2.0`;
```

4. 授权root用户的远程访问权限（原因同上）。

```
1 grant all privileges on `woniuboss2.0`.* to 'root'@'%' identified by 'root';
```

5. 退出数据库，导入sql文件的数据到woniuboss2.0数据库。

```
1 mysql -uroot -p woniuboss2.0 < /sql文件路径/woniuboss2.0.sql
```

6. 重启数据库以及Tomcat服务。

```
1 service mysqld restart
2 service tomcat stop
3 service tomcat start
```

7. 通过浏览器访问WoniuBoss项目。

3.5版本

#

sql文件只有表结构，除有一个管理员账号WNCD056之外，没有其他任何数据，需要自行手工添加。

DVWA工具配置

1. 官网 [下载地址](#)；
2. 下载DVWA_1.9.zip压缩包，解压后放到系统的Xampp的htdocs目录下；
3. 修改..\xampp\htdocs\DVWA-1.9\config\config.inc.php 文件中的数据库配置信息

```
1 $_DVWA[ 'db_server' ] = 'localhost';
2 $_DVWA[ 'db_database' ] = 'dvwa';
3 $_DVWA[ 'db_user' ] = 'root';           // 本地数据库的用户名
4 $_DVWA[ 'db_password' ] = 'root';      // 本地数据库的密码
```

4. 重启Xampp的服务
5. 访问 <http://localhost/DVWA-1.9/setup.php>，点击 create/reset database 即可

6. 默认用户名、密码为admin、password。
7. 更改安全级别，选择DVWA Security项，选择设置Low，然后点击Submit。
8. 解决两个权限问题。

```
1 Writable folder /opt/lampp/htdocs/DVWA-1.9/hackable/uploads/  
2 Writable file /opt/lampp/htdocs/DVWA-1.9/external/phpids/0.6/lib/IDS/tmp/phpids_log.txt
```

需要分别授权可写权限

```
1 chmod 763 /opt/lampp/htdocs/DVWA-1.9/hackable/uploads  
2 chmod 666 /opt/lampp/htdocs/DVWA-1.9/external/phpids/0.6/lib/IDS/tmp/phpids_log.txt
```