# 环境部署

1. 安装JDK1.8

略

1. 安装maven

下载apache-maven-3.5.0

进入：<http://maven.apache.org/download.cgi>， 选择apache-maven-3.5.0-bin.zip下载

解压文件至D:/

设置环境变量:MAVEN\_HOME=D:/ apache-maven-3.5.0

添加至PATH变量末尾：; %MAVEN\_HOME%\bin

mvaen官方的库在国外经常连不上，阿里云的一个中央仓库

<mirror>

<id>alimaven</id>

<mirrorOf>central</mirrorOf>

<name>aliyun maven</name>

<url>http://maven.aliyun.com/nexus/content/repositories/central/</url>

</mirror>

1. 安装Eclipse

下载Eclipse，解压至D:/

* 安装SVN插件：<http://subclipse.tigris.org/update_1.8.x>
* 安装maven插件：

1. <http://download.eclipse.org/releases/juno>

juno为eclipse版本

在filter框中输入maven便能定位要安装的插件

选择 “Collaboration”-“m2e - Maven Integration for Eclipse"，

重启Eclipse

1. 将features和plugins目录下的jar文件分别拷贝到eclipse/features和eclipse/plugins目录下，然后重启eclipse，启动后会download依赖包

* Installations配置：

为了使得Eclipse中安装的Maven插件，同windows中安装的那个相同，需要让eclipse中的maven重新定位一下

Window -> preferences -> type filter text -> maven -> Installations -> Add

选择D:/ apache-maven-3.5.0目录

* 设置settings目录

Window -> preferences -> type filter text -> maven -> User Settings ->

选择D:\apache-maven-3.5.0\conf\settings.xml文件

* 关闭Validation验证

Window -> preferences -> Validation -> Disable All -> OK

* 关闭JavaScript验证

Window -> preferences -> type filter text -> JavaScript -> Error/Warnings

去掉勾选 Enable JavaScript semantic validation

* Jetty插件安装

1. 游戏生产环境使用的是Resin，但是开发经常要编译后代码才能生效，所以开发环境建议使用Jetty

Help->Eclipse Marketplace，在Find后面的框中输入jetty，选择Eclipse Jetty -> Install即可。

1. 本地资源包插件下载地址：http://pan.baidu.com/s/1sjNP5Id

或者是地址：http://pan.baidu.com/s/1bpobHDl

使用方法：解压缩，直接复制到eclipse安装目录的dropins文件夹下面，重启就可以了

* 启动参数

Run-> Run Configurations->Arguments->VM arguments->添加参数-Dprofile=development

* Tomcat运行参数设置：

1. Windows环境下在startup.bat文件最前添加

set JAVA\_HOME=C:\Program Files\Java\jdk1.7.0\_67

set profile=development

1. Linux环境下在startup.sh文件最前添加

export profile=production

* Tomcat 项目部署

1. 创建目录{TOMCAT\_HOME}\conf\Catalina\localhost

在目录{TOMCAT\_HOME}\conf\Catalina\localhost下创建jx.hall.logic.xml文件

<Context docBase="D:\project\jx.hall.logic" reloadable="true" />

jx.hall.logic.xml文件名即为访问的根目录地址

D:\project\jx.hall.logic即为工程文件物理地址

访问地址：<http://localhost:8080/jx.hall.logic/>

1. 打包：Eclipse->选择jx.hall.logic工程文件->单击右键->Run as ->Maven install

生成jx.hall.logic.war文件，存放目录./target/jx.hall.logic.war

1. 将jx.hall.logic.war解压缩至D:\project\jx.hall.logic，该目录为WEB-INF文件夹的上一层目录

* svn服务器部署

1. 安装

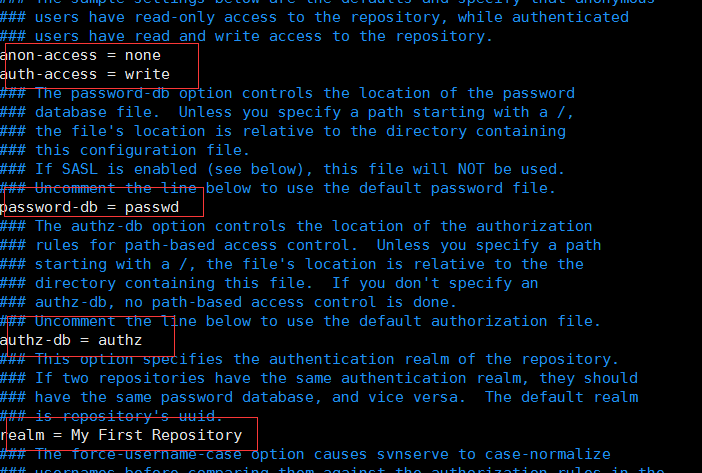
# yum -y install subversion

# mkdir /data/svn

# svnadmin create /data/svn/project

2. 配置

# vim /data/svn/project/conf/svnserve.conf



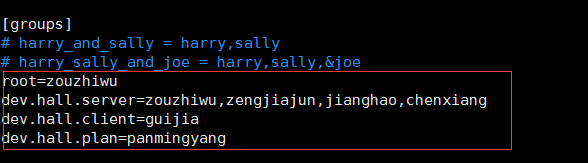
anno-access = none：没有登录的用户不能访问

auth-access = write：登录的用户可以写入

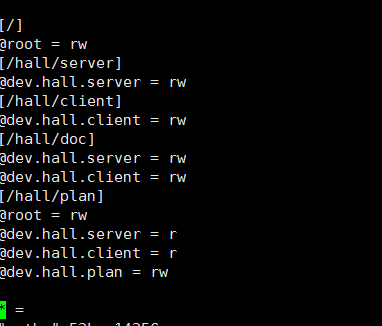
password-db = passwd：密码配置文件

authz-db = authz：权限配置文件

# vim /data/svn/project/conf/authz



设置角色组，并分配用户



权限角色配置

注意：\* = 一定要加

* mysql远程访问授权

#mysql -uroot -p123456;

mysql>use mysql;

mysql>GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'%'WITH GRANT OPTION;

mysql>FLUSH PRIVILEGES;

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

#/etc/init.d/iptables save

#service iptables restart

* 启动SVN服务器：

# svnserve -d -r /data/svn/project

* ftp服务器安装

# rpm -qa |grep vsftpd 检查安装vsftpd软件

vsftpd-2.0.5-16.el5\_5.1

# rpm -e vsftpd-2.0.5-16.el5\_5.1

# yum install vsftpd -y

# systemctl start vsftpd.service

# useradd jx.hall.ftpd

# passwd jx.hall.ftpd

新的 密码：

重新输入新的 密码：

启动： service vsftpd start

* manager部署

在eclipse里面maven.install，生成的war在工程目录的target目录下

# cd /project/jx.njmj.manager/

# rm -rf \*

# rz /jx.njmj.manager/target/jx.njmj.manager.war

# unzip jx.njmj.manager.war

# sh /data/tomcat/bin/shutdown.sh

# sh /data/tomcat/bin/startup.sh

# rm -rf /project/jx.njmj.manager/jx.njmj.manager.war

* logic部署

在eclipse里面maven.install，生成的jar在工程目录的target目录下

# cd /project/jx.njmj.logic/

# rm -rf jx.njmj.logic.jar

# rz /jx.njmj.logic/target/jx.njmj.logic.jar

# cd /project/jx.njmj.logic/lib/

# rm -rf jx.njmj.common.jar

# rz /jx.njmj.common/target/jx.njmj.common.jar

# cd /project/jx.njmj.logic/

# ps -ef | grep njmj

找到对应的线程id，并删除

# kill id

# sh ./shartup.sh

* 把默认的CentOS yum源修改成国内的aliyun yum源

# ll /etc/yum.repos.d/ // 查询当前源

# mkdir /opt/centos-yum.bak // 创建备份目录

# mv /etc/yum.repos.d/\* /opt/centos-yum.bak/ // 备份当前源

# cd /etc/yum.repos.d // 进入目录/etc/yum.repos.d

// 各系统版本repo文件对应的下载操作

# wget <http://mirrors.aliyun.com/repo/Centos-5.repo> // centos 5

# wget <http://mirrors.aliyun.com/repo/Centos-6.repo> // centos 6

# wget <http://mirrors.aliyun.com/repo/Centos-7.repo> // centos 7

# cat /etc/redhat-release // 查看系统的版本

# yum clean all

Loaded plugins: fastestmirror

Cleaning repos: base extras updates

Cleaning up everything

Cleaning up list of fastest mirrors

# yum makecache

# yum list

* java工程部署

1. 修改src/main/resources目录下配置，对应开发环境/生产环境/测试环境

2. 修改pom.xml的lib引用文件, classpathPrefix和outputDirectory分别是引用jar包目录和编译导出文件目录

<build>

<finalName>qy.times.logic</finalName>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-jar-plugin</artifactId>

<version>2.6</version>

<configuration>

<archive>

<manifest>

<mainClass>com.logic.factory.Server</mainClass>

<addClasspath>true</addClasspath>

<classpathPrefix>/root/project/qy.times.logic/lib/</classpathPrefix>

</manifest>

<manifestEntries>

<Class-Path>.</Class-Path>

</manifestEntries>

</archive>

</configuration>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-dependency-plugin</artifactId>

<executions>

<execution>

<id>copy-dependencies</id>

<phase>prepare-package</phase>

<goals>

<goal>copy-dependencies</goal>

</goals>

<configuration>

<outputDirectory>/root/project/qy.times.logic/lib</outputDirectory>

<overWriteReleases>false</overWriteReleases>

<overWriteSnapshots>false</overWriteSnapshots>

<overWriteIfNewer>true</overWriteIfNewer>

</configuration>

</execution>

</executions>

</plugin>

</plugins>

</build>