**监控中心开发环境搭建**

1. **工具&框架**
2. JDK : 1.6
3. 容器：tomcat 7
4. 开发工具：Eclipse IDE for Java EE Developers
5. 管理、构建工具：Maven (m2 eclipse插件)
6. 数据库 Mysql
7. 框架 ：spring MVC与Mybatis
8. **环境搭建步骤**
9. 下载并安装JDK，配置环境变量

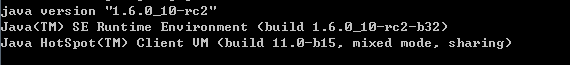
下载地址：<http://www.cr173.com/soft/2457.html>（官网需要注册才能下载，方便起见在第三方网站下载即可）

环境变量配置：查找变量名为“JAVA\_HOME”的系统变量，如果没有，则点击【新建】按钮，在弹出的对话框里，变量名填入“JAVA\_HOME”，在变量值里面填入jdk的安装路径，如“d:\java\jdk\”

查找变量名为“Path”的环境变量，然后点击【编辑】按钮，在变量值的最后面加上“ ;d:\java\jdk\bin”， 记得别忘了最前面的那个分号

查找变量名为“CLASSPATH”的环境变量，如果没有找到该环境变量则点击【新建】按钮，在变量名里填入“CLASSPATH”，在变量值里面填入“ d:\java\jdk\lib\dt.jar;d:\java\jdk\lib\tools.jar;.”特别要注意的就是最后那个点,一定要写上

Cmd -> java –version回车会显示版本信息，则jdk配置成功



1. tomcat 7下载安装

下载地址: <http://tomcat.apache.org/download-70.cgi>

下载apache-tomcat-7.0.42-windows-x64.zip或apache-tomcat-7.0.42-windows-x86.zip文件之后直接解压到某个目录下，比如：D:\javahost\apache-tomcat-7.0.42。然后在我的电脑->属性->高级->环境变量->系统变量中添加以下环境变量：新建系统变量名: TOMCAT\_HOME, 值(value)为：D:\javahost\apache-tomcat-7.0.42

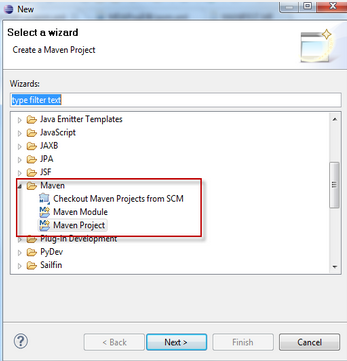
新建系统变量，变量名：CATALINA\_HOME  变量值：D:\java\apache-tomcat-7.0.42 (Tomcat 解压到的目录)。

查找系统变量名: classpath, 增加值为：%TOMCAT\_HOME%\bin

查找系统变量名: path, 增加值为：;%CATALINA\_HOME%\lib;%CATALINA\_HOME%\lib\servlet-api.jar;%CATALINA\_HOME%\lib\jsp-api.jar;      增加后的classpath的值为：;%JAVA\_HOME%\lib;%JAVA\_HOME%\lib\tools.jar;%JAVA\_HOME%\lib\dt.jar;%TOMCAT\_HOME%\bin

1. Eclipse+m2 Eclipse插件
2. 下载并安装Eclipse <http://www.eclipse.org/downloads/> （Eclipse IDE for Java EE Developers）
3. 安装m2 eclipse插件 ：点击eclipse菜单栏Help->Eclipse Marketplace搜索到插件Maven Integration for Eclipse 并点击安装即可

安装成功之后在Eclipse菜单栏中点击File->New->Other,在弹出的对话框中会看到如下图所示



1. 数据库

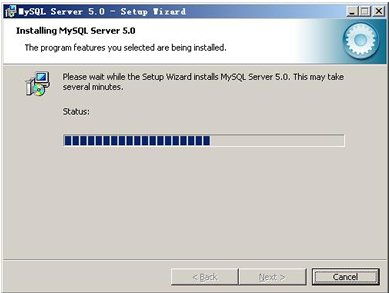
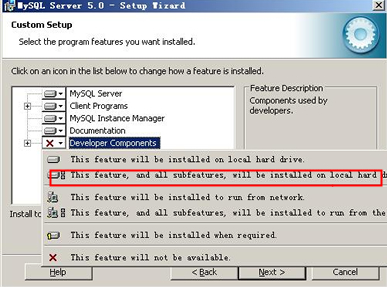
下载：<http://www.jb51.net/softs/88657.html#down>

安装：双击解压缩，运行安装文件

有“Typical（默认）”、“Complete（完全）”、“Custom（用户自定义）”三个选项，我们选择“Custom”

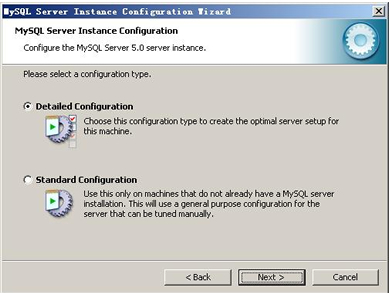
在“Developer Components（开发者部分）”上左键单击，选择“This feature, and all subfeatures, will be installed on local hard drive.”，即“此部分，及下属子部分内容，全部安装在本地硬盘上”。

点选“Change...”，手动指定安装目录。D:\Mysql 返回刚才的界面，按“Next”继续，按“Install”开始安装。



直到出现下面的界面，这里是询问你是否要注册一个mysql.com的账号，或是使用已有的账号登陆mysql.com，一般不需要了，点选“Skip Sign-Up”，按“Next”略过此步骤。

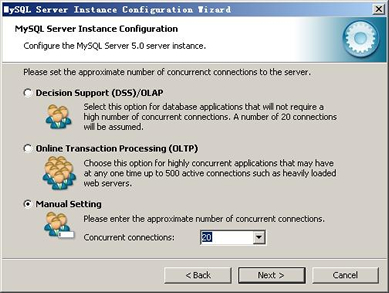
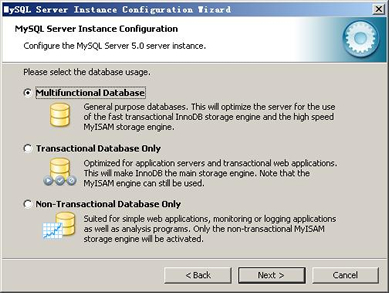
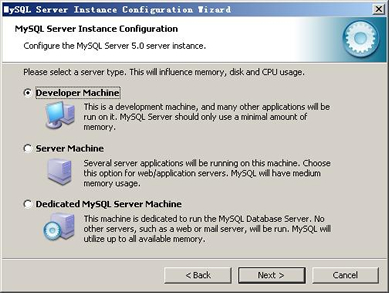
mysql配置向导，将 “Configure the Mysql Server now”前面的勾打上，点“Finish”结束软件的安装并启动mysql配置向导。



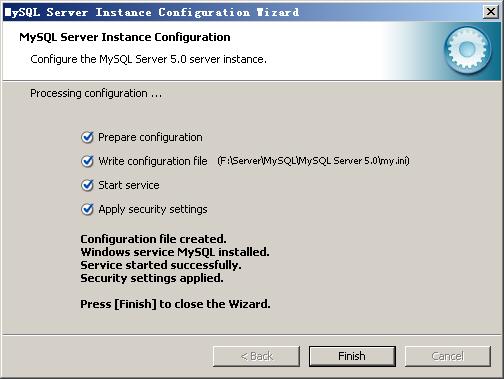
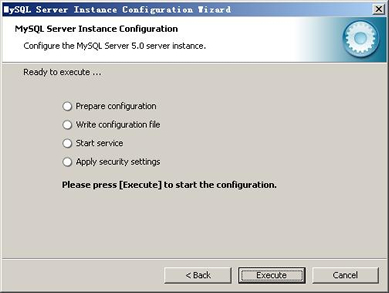
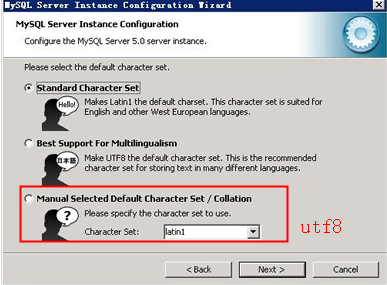
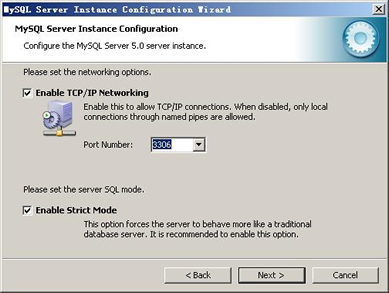
Detailed Configuration（手动精确配置）”、“Standard Configuration（标准配置）”，我们选择“Detailed Configuration”，方便熟悉配置过程。

选择服务器类型，“Developer Machine（开发测试类，mysql占用很少资源）”、“Server Machine（服务器类型，mysql占用较多资源）”、“Dedicated MySQL Server Machine（专门的数据库服务器，mysql占用所有可用资源）”，大家根据自己的类型选择了，一般选“Server Machine”，不会太少，也不会占满。

择mysql数据库的大致用途，“Multifunctional Database（通用多功能型，好）”、“Transactional Database Only（服务器类型，专注于事务处理，一般）”、“Non-Transactional Database Only（非事务处理型，较简单，主要做一些监控、记数用，对MyISAM数据类型的支持仅限于non-transactional），随自己的用途而选择了，我这里选择“Transactional Database Only”，按“Next”继续。



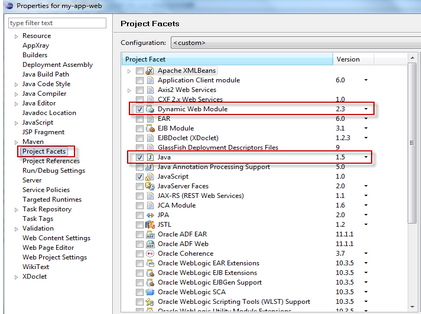
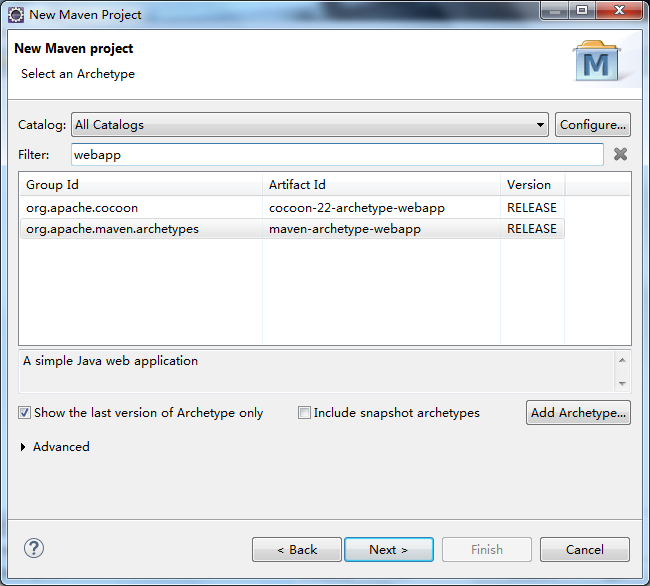
是否启用TCP/IP连接，设定端口，如果不启用，就只能在自己的机器上访问mysql数据库了，我这里启用，把前面的勾打上，Port Number：3306，在这个页面上，可以选择“启用标准模式”（Enable Strict Mode），这样MySQL就不会允许细小的语法错误。按“Next”继续,配置字符编码



设置完毕，按“Finish”结束mysql的安装与配置——这里有一个比较常见的错误，就是不能“Start service”，一般出现在以前有安装mysql的服务器上，解决的办法，先保证以前安装的mysql服务器彻底卸载掉了；不行的话，检查是否按上面一步所说，之前的密码是否有修改，照上面的操作；如果依然不行，将mysql安装目录下的data文件夹备份，然后删除，在安装完成后，将安装生成的 data文件夹删除，备份的data文件夹移回来，再重启mysql服务就可以了，这种情况下，可能需要将数据库检查一下，然后修复一次，防止数据出错。

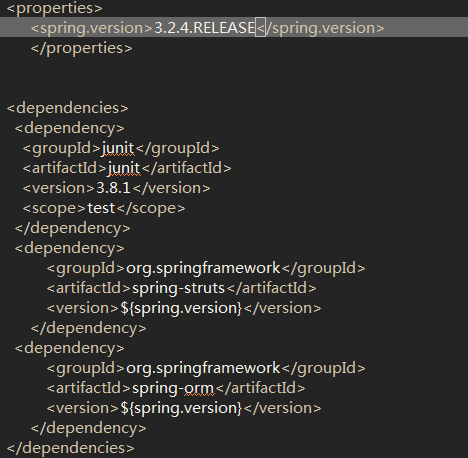
1. **框架搭建**
2. 新建一个maven工程

点击Eclipse菜单栏File->New->Ohter->Maven，在弹出的对话框中选择Maven Project并点击Next，选择maven-archetype-quickstart（创建的是maven的web工程）

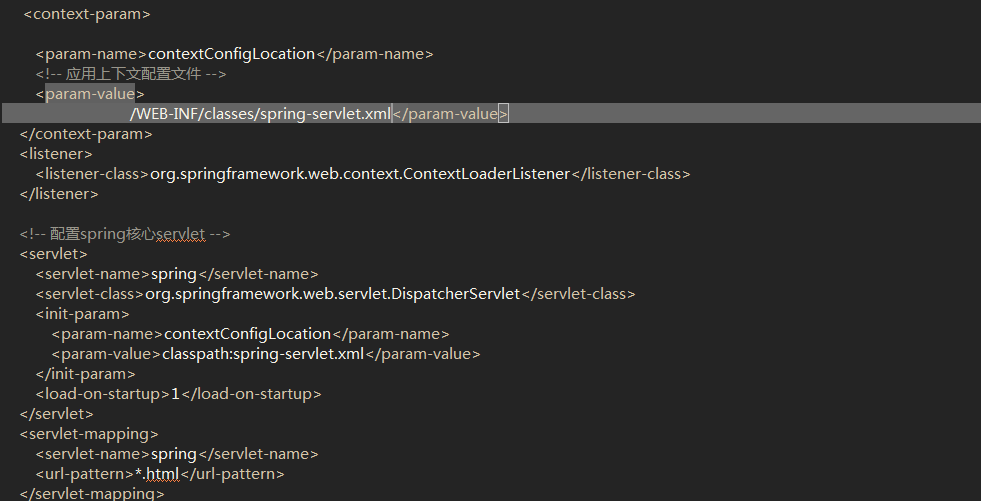


按图示填写好groupId, artfactId,version等信息(groupId：项目组织的逆向域名，比如Apache的项目可以org.apache开头,artfactId：表示groupId下单个项目的唯一标识version：发布项目的版本，版本标识加上SNAPSHOT标记)，点击Finish。右击项目，点击Properties->Project Facets（JAVA 1.6） ,maven web工程建立完毕

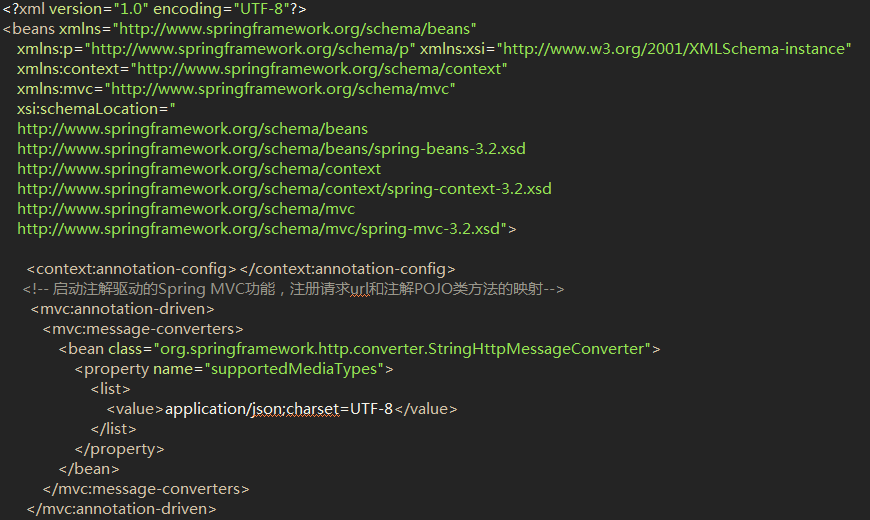
1. 整合spring MVC
2. 项目pom.xml文件中加入对spring包的依赖



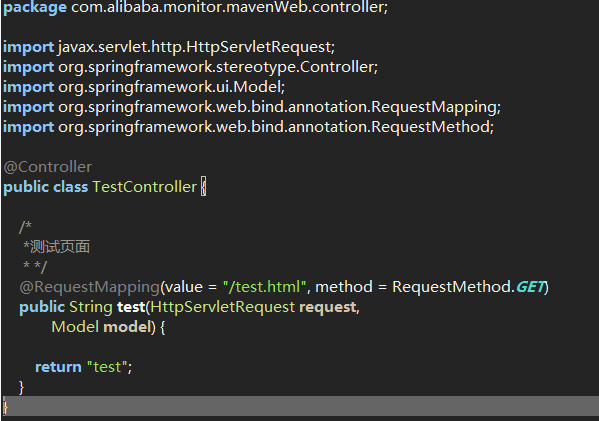
② web.xml配置



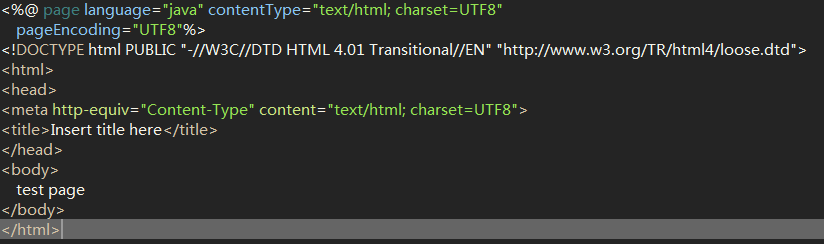
③ resources目录下创建名为spring-servlet.xml的配置文件



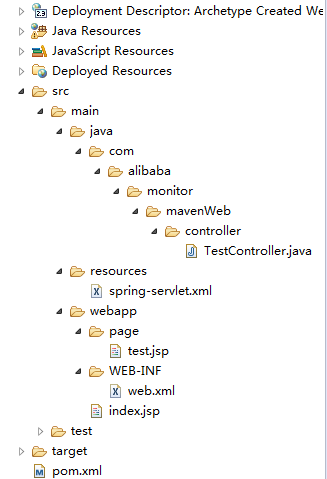
④ src\main\java\com.alibaba.monitor.mavenWeb.controller下创建TestController.java



⑤ src\main\webapp\page下创建test.jsp的文件

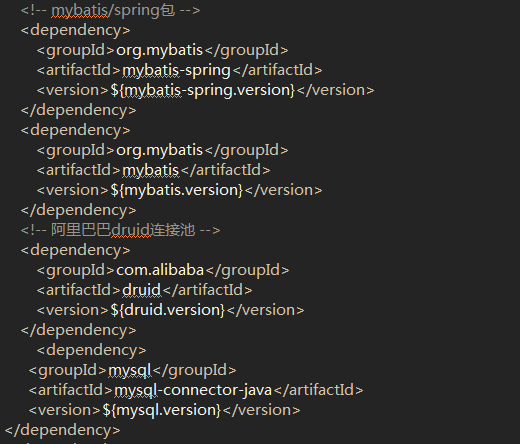
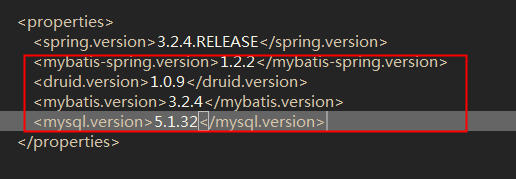


最后的文件结构为

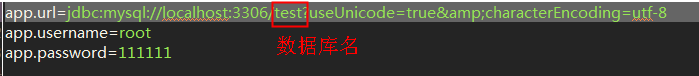


右击项目名称，run as - > run on server启动tomcat , 在浏览器打开 <http://localhost:8080/你的工程名/test.html>显示test page即为成功

1. 整合mybatis
2. 在pom.xml加入包依赖

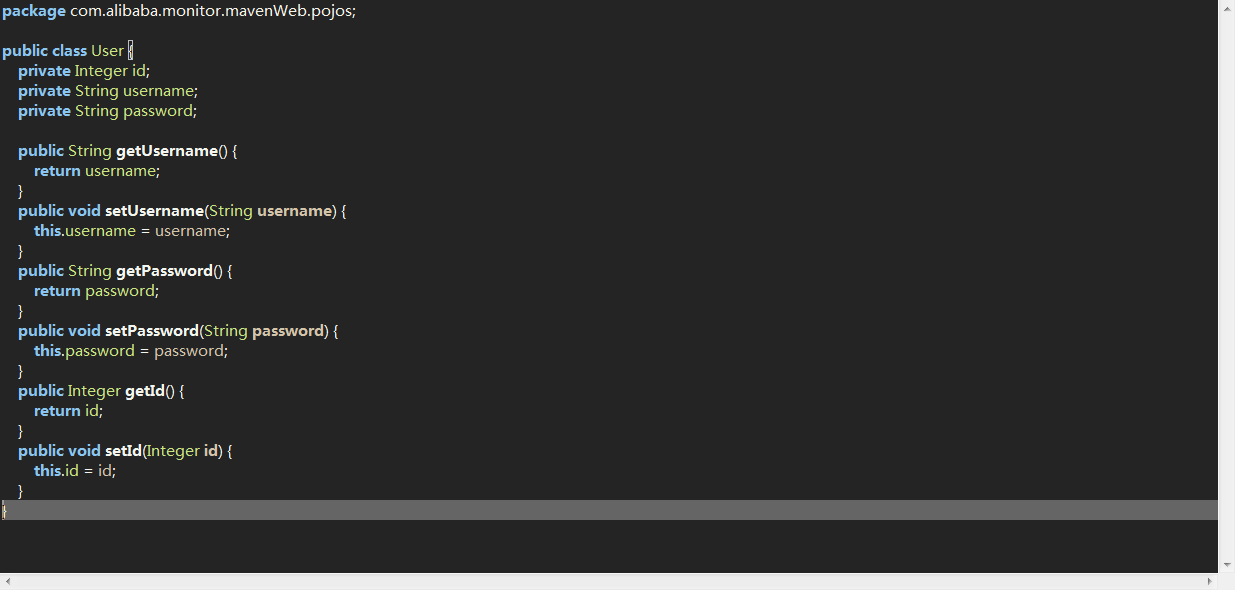


1. resources目录下创建名为jdbc.properties的配置文件，管理数据库的地址，用户名和密码



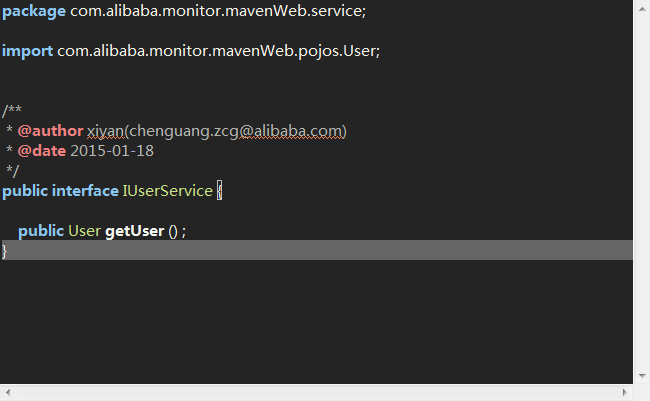
③ 创建com\alibaba\monitor\mavenWeb\pojos目录

在pojos目录下创建User.java

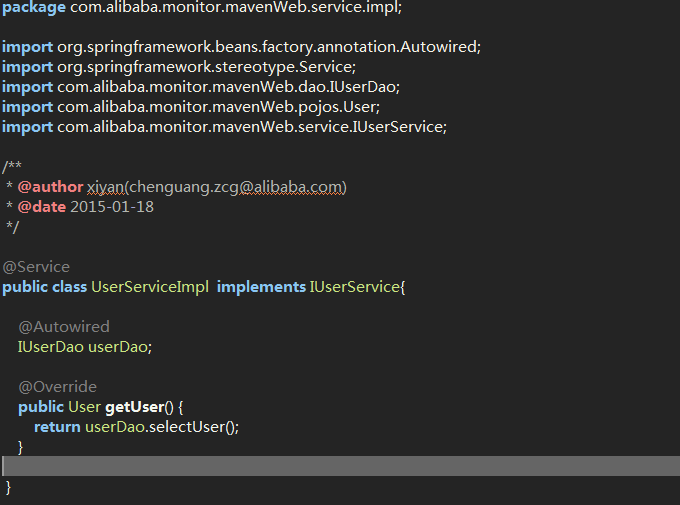


④：创建com\alibaba\monitor\mavenWeb\service目录和com\alibaba\monitor\mavenWeb\pojos\service\impl目录

Service下创建IUserService.java

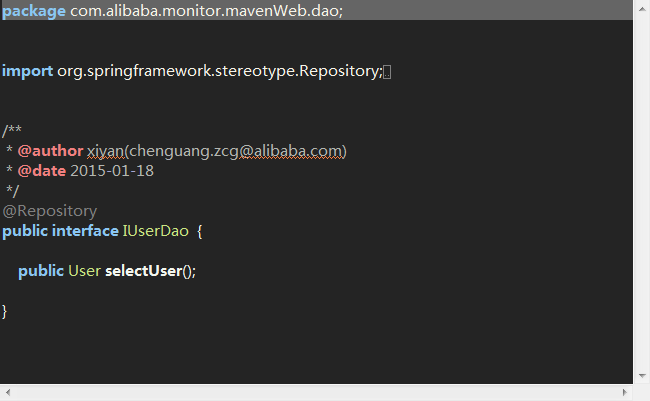


impl下创建UserServiceImpl.java



⑤创建com\alibaba\monitor\mavenWeb\dao目录

创建IUserDao.java



⑥：resources目录下创建名为spring-mybatis.xml配置文件

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:p="http://www.springframework.org/schema/p"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-3.2.xsd">

<bean id="JDBCConfig"

class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">

<property name="location">

<value>classpath:jdbc.properties</value>

</property>

</bean>

<!-- 数据源 one -->

<bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource"

init-method="init" destroy-method="close">

<property name="url">

<value>${app.url}</value>

</property>

<property name="username">

<value>${app.username}</value>

</property>

<property name="password">

<value>${app.password}</value>

</property>

</bean>

<!-- mybatis文件配置，扫描所有mapper文件 -->

<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean"

p:dataSource-ref="dataSource" p:configLocation="classpath:mybatis-config.xml"

p:mapperLocations="classpath:mapper/\*.xml"/>

<!-- configLocation为mybatis属性

mapperLocations为所有mapper -->

<!-- spring与mybatis整合配置，扫描所有dao -->

<bean class="org.mybatis.spring.mapper.MapperScannerConfigurer"

p:basePackage="com.alibaba.monitor.mavenWeb.dao" p:sqlSessionFactoryBeanName="sqlSessionFactory"/>

<!-- 对数据源进行事务管理 -->

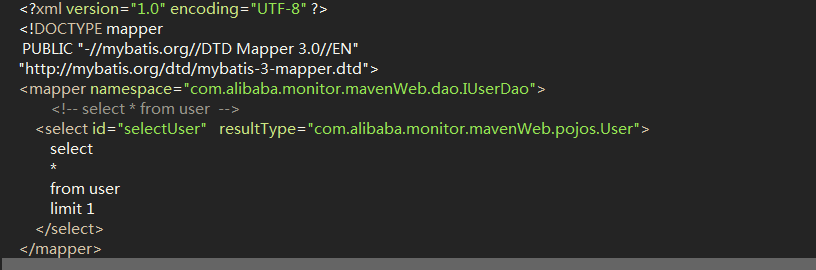
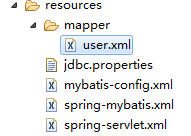
<bean id="transactionManager"

class="org.springframework.jdbc.datasource.DataSourceTransactionManager"

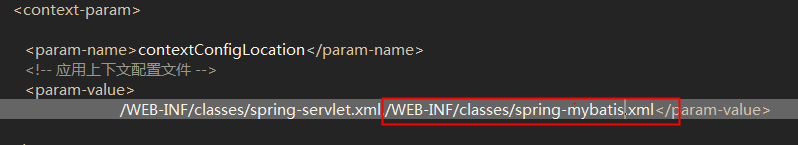
p:dataSource-ref="dataSource"/>

</beans>

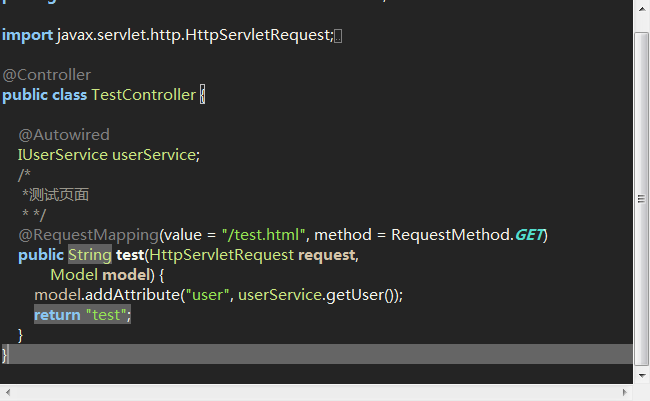
⑦：resources下创建mapper文件夹，在mapper下创建user.xml(具体sql语句)



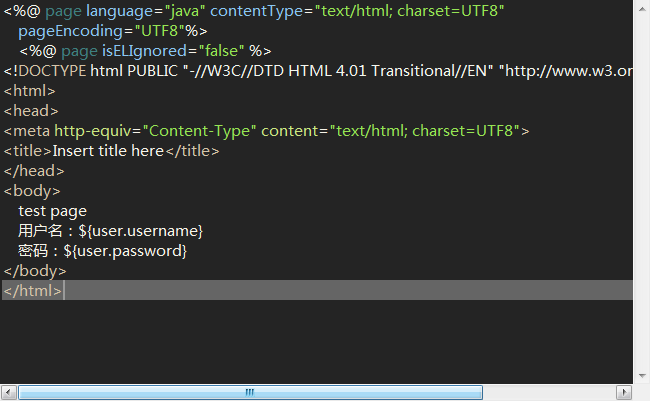
⑧修改web.xml



⑨修改TestController.java



⑩修改test.jsp

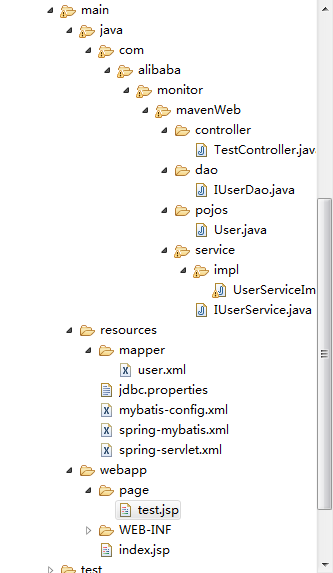


重启tomcat，访问<http://localhost:8080/你的工程名/test.html>

用户名和密码为数据库读出的数据



最终的目录结构



框架基本搭建完成