**maven+springMVC+mybatis+junit详细搭建过程**

发表于1年前(2014-02-12 17:53) 阅读（10303） | 评论（[30](http://my.oschina.net/u/1011897/blog/199172" \l "comments)） *144*人收藏此文章, [我要收藏](javascript:add_to_favor(199172,3))

*赞22*

[**慕课网，程序员升职加薪神器，点击免费学习**](http://www.imooc.com/course/list?from=oschina)

*摘要*在做javaweb的过程中，搭建框架式比较头疼事情了，每次重新去搭建太浪费时间了，现在把框架搭建完成，可以在以后的项目中运用，节约开发成本。

[maven](http://www.oschina.net/search?scope=blog&q=maven)[springMVC](http://www.oschina.net/search?scope=blog&q=springMVC)[mybatis](http://www.oschina.net/search?scope=blog&q=mybatis)[log4j](http://www.oschina.net/search?scope=blog&q=log4j)[javaweb项目框架搭建](http://www.oschina.net/search?scope=blog&q=javaweb%E9%A1%B9%E7%9B%AE%E6%A1%86%E6%9E%B6%E6%90%AD%E5%BB%BA)

*目录[-]*

 [springMVC+mybatis框架搭建](http://my.oschina.net/u/1011897/blog/199172" \l "OSC_h1_1)

 [1. 工程目录结构整理清楚](http://my.oschina.net/u/1011897/blog/199172" \l "OSC_h2_2)

 [2. 引入依赖包](http://my.oschina.net/u/1011897/blog/199172" \l "OSC_h2_3)

 [3. 配置数据库连接属性](http://my.oschina.net/u/1011897/blog/199172" \l "OSC_h2_4)

 [4. 配置spring配置文件](http://my.oschina.net/u/1011897/blog/199172" \l "OSC_h2_5)

 [5. java代码编写（model，dao，service层代码）](http://my.oschina.net/u/1011897/blog/199172" \l "OSC_h2_6)

 [6. mybatis配置](http://my.oschina.net/u/1011897/blog/199172" \l "OSC_h2_7)

 [7. junit测试插入功能](http://my.oschina.net/u/1011897/blog/199172" \l "OSC_h2_8)

 [8. springMVC模块搭建](http://my.oschina.net/u/1011897/blog/199172" \l "OSC_h2_9)

 [9. log4j日志记录搭建](http://my.oschina.net/u/1011897/blog/199172" \l "OSC_h2_10)

 [10. 测试运行](http://my.oschina.net/u/1011897/blog/199172" \l "OSC_h2_11)

 [整体包下载地址：](http://my.oschina.net/u/1011897/blog/199172" \l "OSC_h2_12)

**springMVC+mybatis框架搭建**

在上一遍博客中以及讲诉了新建maven项目的流程，现在紧跟上一遍文章，接着搭建spring项目

首先我们先要弄清搭建项目的一般流程，需要注意哪些方面，想要什么样的效果，自己的功能有哪些？

（假设效果：项目目录结构清晰，能够查询到本地数据库中的内容。。）

**1. 工程目录结构整理清楚**

在src/main/java文件夹中，新建包cn.springmvc.model(存放javabean),

cn.springmvc.dao（存放spring与mybatis连接接口）,

cn.springmvc.service（service接口）,

cn.springmvc.service.impl（service接口的实现），

cn.springmvc.controller（存放控制层controller）

在src/main/resource文件夹中，新建包conf（存放配置文件），

mapper（mybatis的mapper文件）

在src/test/java文件夹中，新建包cn.springmvc.test(存放测试文件)

在WEB-INF文件夹下新建jsp文件夹（存放jsp文件）

这样项目结构基本完成了

**2. 引入依赖包**

打开maven的pom文件，对本次开发所需使用的架包依次导入（maven项目管理的优势）

查找依赖结构有个不错的网站，<http://search.maven.org/> 只要输入包名即可查找引来关系

pom.xml(包依赖)

[?](http://my.oschina.net/u/1011897/blog/199172)

|  |  |
| --- | --- |
| 33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144 | <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd">  <modelVersion>4.0.0</modelVersion>  <groupId>eyas.springmvc</groupId>  <artifactId>springmvc</artifactId>  <packaging>war</packaging>  <version>0.0.1-SNAPSHOT</version>  <name>springmvc Maven Webapp</name>  <url>http://maven.apache.org</url>  <properties>  <!-- spring版本号 -->  <spring.version>3.2.4.RELEASE</spring.version>  <!-- mybatis版本号 -->  <mybatis.version>3.2.4</mybatis.version>  <!-- log4j日志文件管理包版本 -->  <slf4j.version>1.6.6</slf4j.version>  <log4j.version>1.2.9</log4j.version>  </properties>  <dependencies>  <!-- spring核心包 -->  <!-- springframe start -->  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-core</artifactId>  <version>${spring.version}</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-web</artifactId>  <version>${spring.version}</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-oxm</artifactId>  <version>${spring.version}</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-tx</artifactId>  <version>${spring.version}</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-jdbc</artifactId>  <version>${spring.version}</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-webmvc</artifactId>  <version>${spring.version}</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-aop</artifactId>  <version>${spring.version}</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-context-support</artifactId>  <version>${spring.version}</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-aop</artifactId>  <version>${spring.version}</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-test</artifactId>  <version>${spring.version}</version>  </dependency>  <!-- springframe end -->  <!-- mybatis核心包 -->  <dependency>  <groupId>org.mybatis</groupId>  <artifactId>mybatis</artifactId>  <version>${mybatis.version}</version>  </dependency>  <!-- mybatis/spring包 -->  <dependency>  <groupId>org.mybatis</groupId>  <artifactId>mybatis-spring</artifactId>  <version>1.2.2</version>  </dependency>  <!-- mysql驱动包 -->  <dependency>  <groupId>mysql</groupId>  <artifactId>mysql-connector-java</artifactId>  <version>5.1.29</version>  </dependency>  <!-- junit测试包 -->  <dependency>  <groupId>junit</groupId>  <artifactId>junit</artifactId>  <version>4.11</version>  <scope>test</scope>  </dependency>  <!-- 阿里巴巴数据源包 -->  <dependency>  <groupId>com.alibaba</groupId>  <artifactId>druid</artifactId>  <version>1.0.2</version>  </dependency>  <!-- json数据 -->  <dependency>  <groupId>org.codehaus.jackson</groupId>  <artifactId>jackson-mapper-asl</artifactId>  <version>1.9.13</version>  </dependency>  <!-- 日志文件管理包 -->  <!-- log start -->  <dependency>  <groupId>log4j</groupId>  <artifactId>log4j</artifactId>  <version>${log4j.version}</version>  </dependency>  <dependency>  <groupId>org.slf4j</groupId>  <artifactId>slf4j-api</artifactId>  <version>${slf4j.version}</version>  </dependency>  <dependency>  <groupId>org.slf4j</groupId>  <artifactId>slf4j-log4j12</artifactId>  <version>${slf4j.version}</version>  </dependency>  <!-- log end -->  </dependencies>  <build>  <finalName>springmvc</finalName>  </build>  </project> |

**3. 配置数据库连接属性**

conf/ jdbc.properties（jdbc配置文件)

[?](http://my.oschina.net/u/1011897/blog/199172)

|  |  |
| --- | --- |
| 1  2  3  4 | jdbc\_driverClassName=com.mysql.jdbc.Driver  jdbc\_url=jdbc:mysql://localhost:3306/mydays?useUnicode=true&amp;characterEncoding=utf-8  jdbc\_username=root  jdbc\_password=root |

**4. 配置spring配置文件**

conf/spring.xml(spring配置文件的扫描)

[?](http://my.oschina.net/u/1011897/blog/199172)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | <?xml version="1.0" encoding="UTF-8"?>  <beans xmlns="http://www.springframework.org/schema/beans"  xmlns:context="http://www.springframework.org/schema/context"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans-3.0.xsd  http://www.springframework.org/schema/context  http://www.springframework.org/schema/context/spring-context.xsd">  <!-- 引入jdbc配置文件 -->  <context:property-placeholder location="classpath:conf/jdbc.properties"/>  <!-- 扫描文件（自动将servicec层注入） -->  <context:component-scan base-package="cn.springmvc.service"/>  </beans> |

conf/spring-mybatis.xml（spring与mybatis连接属性）

[?](http://my.oschina.net/u/1011897/blog/199172)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84 | <?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"  xmlns:context="http://www.springframework.org/schema/context"  xmlns:aop="http://www.springframework.org/schema/aop"  xmlns:tx="http://www.springframework.org/schema/tx"  xmlns:util="http://www.springframework.org/schema/util"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans-3.2.xsd  http://www.springframework.org/schema/context  http://www.springframework.org/schema/context/spring-context-3.2.xsd  http://www.springframework.org/schema/tx  http://www.springframework.org/schema/tx/spring-tx-3.2.xsd  http://www.springframework.org/schema/aop  http://www.springframework.org/schema/aop/spring-aop-3.2.xsd  http://www.springframework.org/schema/util  http://www.springframework.org/schema/util/spring-util-3.2.xsd">  <bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource" init-method="init"  destroy-method="close" >  <property name="driverClassName">  <value>${jdbc\_driverClassName}</value>  </property>  <property name="url">  <value>${jdbc\_url}</value>  </property>  <property name="username">  <value>${jdbc\_username}</value>  </property>  <property name="password">  <value>${jdbc\_password}</value>  </property>  <!-- 连接池最大使用连接数 -->  <property name="maxActive">  <value>20</value>  </property>  <!-- 初始化连接大小 -->  <property name="initialSize">  <value>1</value>  </property>  <!-- 获取连接最大等待时间 -->  <property name="maxWait">  <value>60000</value>  </property>  <!-- 连接池最大空闲 -->  <property name="maxIdle">  <value>20</value>  </property>  <!-- 连接池最小空闲 -->  <property name="minIdle">  <value>3</value>  </property>  <!-- 自动清除无用连接 -->  <property name="removeAbandoned">  <value>true</value>  </property>  <!-- 清除无用连接的等待时间 -->  <property name="removeAbandonedTimeout">  <value>180</value>  </property>  <!-- 连接属性 -->  <property name="connectionProperties">  <value>clientEncoding=UTF-8</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:conf/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="cn.springmvc.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> |

**5. java代码编写（model，dao，service层代码）**

cn.springmvc.model/User.java(用户基本信息)

[?](http://my.oschina.net/u/1011897/blog/199172)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | package cn.springmvc.model;  /\*\*  \* 用户表  \*/  public class User {  private int id;  private int state;  private String nickname;  public int getId() {  return id;  }  public void setId(int id) {  this.id = id;  }  public int getState() {  return state;  }  public void setState(int state) {  this.state = state;  }  public String getNickname() {  return nickname;  }  public void setNickname(String nickname) {  this.nickname = nickname;  }  } |

cn.springmvc.dao/UserDAO.java(dao操作接口)

[?](http://my.oschina.net/u/1011897/blog/199172)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | package cn.springmvc.dao;  import cn.springmvc.model.User;  public interface UserDAO {  /\*\*  \* 添加新用户  \* @param user  \* @return  \*/  public int insertUser(User user);  } |

cn.springmvc.service/UserService.java(service层接口)

[?](http://my.oschina.net/u/1011897/blog/199172)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | package cn.springmvc.service;  import cn.springmvc.model.User;  public interface UserService {  public int insertUser(User user);  } |

cn.springmvc.service.impl/UserServiceImpl.java(service层接口实现)

[?](http://my.oschina.net/u/1011897/blog/199172)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | package cn.springmvc.service.impl;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Service;  import cn.springmvc.dao.UserDAO;  import cn.springmvc.model.User;  import cn.springmvc.service.UserService;  @Service  public class UserServiceImpl implements UserService{  @Autowired  private UserDAO userDAO;  @Override  public int insertUser(User user) {  // TODO Auto-generated method stub  return userDAO.insertUser(user);  }  } |

**6. mybatis配置**

conf/mybatis-config.xml(mybatis配置的基本文件)

[?](http://my.oschina.net/u/1011897/blog/199172)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | <?xml version="1.0" encoding="UTF-8" ?>  <!DOCTYPE configuration  PUBLIC "-//mybatis.org//DTD Config 3.0//EN"  "http://mybatis.org/dtd/mybatis-3-config.dtd">  <configuration>  <!-- 命名空间 -->  <typeAliases>  <typeAlias alias="User" type="cn.springmvc.model.User"/>  </typeAliases>  <!-- 映射map -->  <mappers>  </mappers>  </configuration> |

mapper/UserMapper.xml(mybatis的实现)

[?](http://my.oschina.net/u/1011897/blog/199172)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | <?xml version="1.0" encoding="UTF-8" ?>  <!DOCTYPE mapper  PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  "http://mybatis.org/dtd/mybatis-3-mapper.dtd">  <mapper namespace="cn.springmvc.dao.UserDAO">  <insert id="insertUser" parameterType="User" keyProperty="id">  insert into days\_user(  state,  nickname)  values  (  #{state},  #{nickname})  </insert>  </mapper> |

这样就完成了基本插入用户的功能！

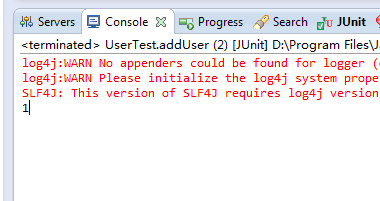
**7. junit测试插入功能**

cn.springmvc.test/UserTest.java(用户测试模块)

[?](http://my.oschina.net/u/1011897/blog/199172)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32 | package cn.springmvc.test;  import org.junit.Before;  import org.junit.Test;  import org.springframework.context.ApplicationContext;  import org.springframework.context.support.ClassPathXmlApplicationContext;  import cn.springmvc.model.User;  import cn.springmvc.service.UserService;  public class UserTest {  private UserService userService;  @Before  public void before(){  @SuppressWarnings("resource")  ApplicationContext context = new ClassPathXmlApplicationContext(new String[]{"classpath:conf/spring.xml"  ,"classpath:conf/spring-mybatis.xml"});  userService = (UserService) context.getBean("userServiceImpl");  }  @Test  public void addUser(){  User user = new User();  user.setNickname("你好");  user.setState(2);  System.out.println(userService.insertUser(user));  }  } |

测试结果如图：

[](http://static.oschina.net/uploads/space/2014/0212/170052_OpC7_1011897.png)

即完成插入功能！

**8. springMVC模块搭建**

web.xml（web功能配置）

[?](http://my.oschina.net/u/1011897/blog/199172)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94 | <?xml version="1.0" encoding="UTF-8"?>  <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"  id="WebApp\_ID" version="2.5">  <display-name>Archetype Created Web Application</display-name>  <!-- 读取spring配置文件 -->  <context-param>  <param-name>contextConfigLocation</param-name>  <param-value>classpath:conf/spring.xml;  classpath:conf/spring-mybatis.xml  </param-value>  </context-param>  <!-- 设计路径变量值 -->  <context-param>  <param-name>webAppRootKey</param-name>  <param-value>springmvc.root</param-value>  </context-param>  <!-- Spring字符集过滤器 -->  <filter>  <filter-name>SpringEncodingFilter</filter-name>  <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>  <init-param>  <param-name>encoding</param-name>  <param-value>UTF-8</param-value>  </init-param>  <init-param>  <param-name>forceEncoding</param-name>  <param-value>true</param-value>  </init-param>  </filter>  <filter-mapping>  <filter-name>SpringEncodingFilter</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping>  <!-- 日志记录 -->  <context-param>  <!-- 日志配置文件路径 -->  <param-name>log4jConfigLocation</param-name>  <param-value>classpath:conf/log4j.properties</param-value>  </context-param>  <context-param>  <!-- 日志页面的刷新间隔 -->  <param-name>log4jRefreshInterval</param-name>  <param-value>6000</param-value>  </context-param>  <listener>  <listener-class>org.springframework.web.util.Log4jConfigListener</listener-class>  </listener>  <listener>  <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>  </listener>  <!-- springMVC核心配置 -->  <servlet>  <servlet-name>spring</servlet-name>  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>  <init-param>  <param-name>contextConfigLocation</param-name>  <param-value>classpath:conf/spring-mvc.xml</param-value>  </init-param>  <load-on-startup>2</load-on-startup>  </servlet>  <servlet-mapping>  <servlet-name>spring</servlet-name>  <url-pattern>\*.do</url-pattern>  </servlet-mapping>  <welcome-file-list>  <welcome-file>index.jsp</welcome-file>  </welcome-file-list>  <!-- 错误跳转页面 -->  <error-page>  <!-- 路径不正确 -->  <error-code>404</error-code>  <location>/WEB-INF/errorpage/404.jsp</location>  </error-page>  <error-page>  <!-- 没有访问权限，访问被禁止 -->  <error-code>405</error-code>  <location>/WEB-INF/errorpage/405.jsp</location>  </error-page>  <error-page>  <!-- 内部错误 -->  <error-code>500</error-code>  <location>/WEB-INF/errorpage/500.jsp</location>  </error-page>  </web-app> |

conf/spring-mvc.xml(mvc配置文件)

[?](http://my.oschina.net/u/1011897/blog/199172)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | <?xml version="1.0" encoding="UTF-8"?>  <beans xmlns="http://www.springframework.org/schema/beans"  xmlns:p="http://www.springframework.org/schema/p"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:context="http://www.springframework.org/schema/context"  xmlns:mvc="http://www.springframework.org/schema/mvc"  xsi:schemaLocation="  http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans-3.2.xsd  http://www.springframework.org/schema/context  http://www.springframework.org/schema/context/spring-context-3.2.xsd  http://www.springframework.org/schema/mvc  http://www.springframework.org/schema/mvc/spring-mvc-3.2.xsd">  <!-- 扫描controller（controller层注入） -->  <context:component-scan base-package="cn.springmvc.controller"/>  <!-- 避免IE在ajax请求时，返回json出现下载 -->  <bean id="jacksonMessageConverter" class="org.springframework.http.converter.json.MappingJacksonHttpMessageConverter">  <property name="supportedMediaTypes">  <list>  <value>text/html;charset=UTF-8</value>  </list>  </property>  </bean>  <!-- 对模型视图添加前后缀 -->  <bean id="viewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver"  p:prefix="/WEB-INF/jsp/" p:suffix=".jsp"/>  </beans> |

**9. log4j日志记录搭建**

conf/log4j.properties(日志记录的配置文件)

[?](http://my.oschina.net/u/1011897/blog/199172)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26 | ### set log levels ###  #log4j.rootLogger = debug , stdout , D , E  log4j.rootLogger = debug , stdout , D  ### output to the console ###  log4j.appender.stdout = org.apache.log4j.ConsoleAppender  log4j.appender.stdout.Target = System.out  log4j.appender.stdout.layout = org.apache.log4j.PatternLayout  #log4j.appender.stdout.layout.ConversionPattern = %d{ABSOLUTE} %5p %c{ 1 }:%L - %m%n  log4j.appender.stdout.layout.ConversionPattern = %-d{yyyy-MM-dd HH:mm:ss} [%c]-[%p] %m%n  ### Output to the log file ###  log4j.appender.D = org.apache.log4j.DailyRollingFileAppender  log4j.appender.D.File = ${springmvc.root}/WEB-INF/logs/log.log  log4j.appender.D.Append = true  log4j.appender.D.Threshold = DEBUG  log4j.appender.D.layout = org.apache.log4j.PatternLayout  log4j.appender.D.layout.ConversionPattern = %-d{yyyy-MM-dd HH:mm:ss} [ %t:%r ] - [ %p ] %m%n  ### Save exception information to separate file ###  log4j.appender.D = org.apache.log4j.DailyRollingFileAppender  log4j.appender.D.File = ${springmvc.root}/WEB-INF/logs/error.log  log4j.appender.D.Append = true  log4j.appender.D.Threshold = ERROR  log4j.appender.D.layout = org.apache.log4j.PatternLayout  log4j.appender.D.layout.ConversionPattern = %-d{yyyy-MM-dd HH:mm:ss} [ %t:%r ] - [ %p ] %m%n |

**10. 测试运行**

WEB-INF/jsp/index.jsp(测试文件)

[?](http://my.oschina.net/u/1011897/blog/199172)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | <%@ page language="java" contentType="text/html; charset=UTF-8"  pageEncoding="UTF-8"%>  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  <title>Insert title here</title>  </head>  <body>  <h1>欢迎你！！！</h1>  </body>  </html> |

cn.springmvc.controller/UserComtroller.java(controller层控制)

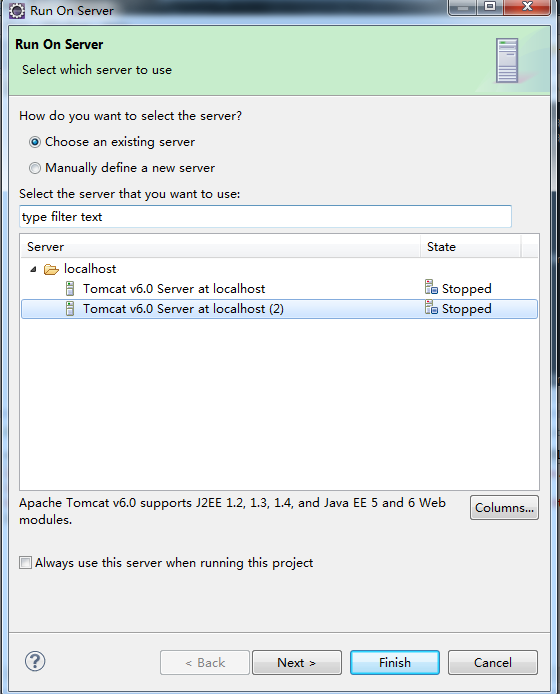
[?](http://my.oschina.net/u/1011897/blog/199172)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | package cn.springmvc.controller;  import org.springframework.stereotype.Controller;  import org.springframework.web.bind.annotation.RequestMapping;  @Controller  @RequestMapping("/")  public class UserController {  @RequestMapping("index")  public String index(){  return "index";  }  } |

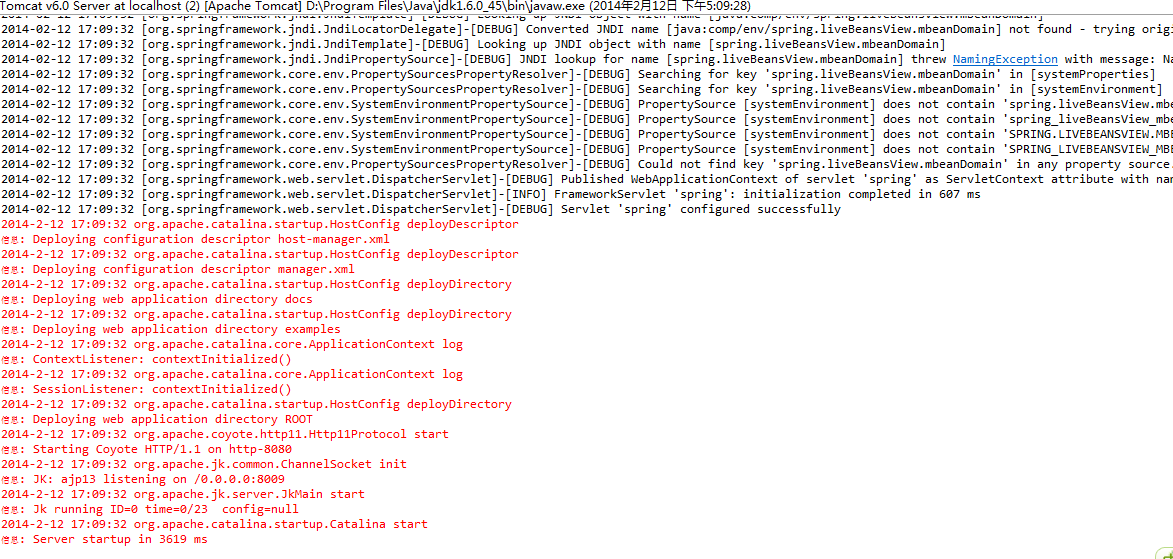
将该项目部署到本地tomcat6中

右击该项目选择Run as，选择Run on Server

选择自己建立的tomcat6运行如图：

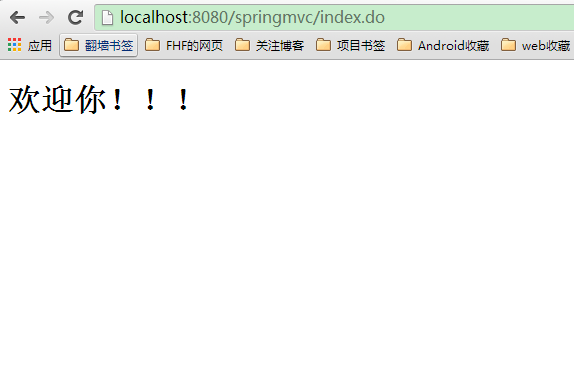
[](http://static.oschina.net/uploads/space/2014/0212/171404_Blt0_1011897.png)

控制台打印出日志如图（启动成功）：

[](http://static.oschina.net/uploads/space/2014/0212/171444_DCnP_1011897.png)

在浏览器中输入：http://localhost:8080/springmvc/index.do 进行测试

效果如图（返回成功）：

[](http://static.oschina.net/uploads/space/2014/0212/171552_cloQ_1011897.png)

**整体包下载地址：**

http://download.csdn.net/detail/fengerfhf/6916831