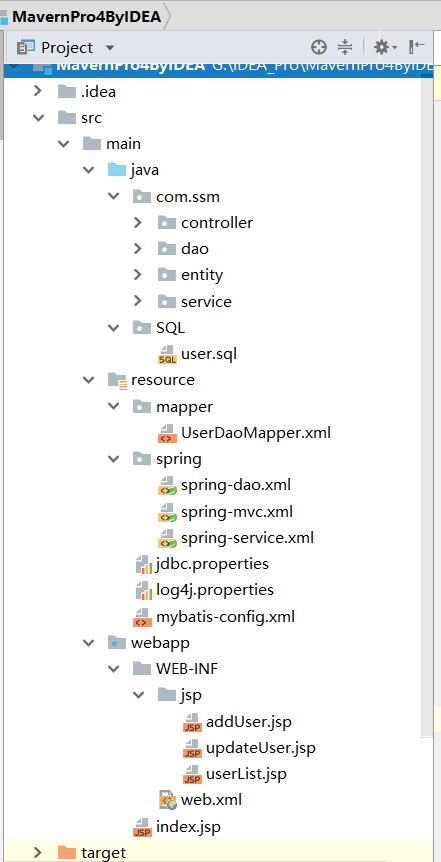
1. 用户表User

|  |
| --- |
| **SET** FOREIGN\_KEY\_CHECKS=0;**DROP TABLE** IF **EXISTS** `user`; **CREATE TABLE** `user` (  **`id` int**(11) **NOT NULL** auto\_increment,  **`name` varchar**(100) **NOT NULL**,  **`address` varchar**(255) **NOT NULL**,  **`age` int**(11) **NOT NULL**,  **`phone` varchar**(20) **NOT NULL**,  **PRIMARY KEY** (**`id`**) ) ENGINE=InnoDB AUTO\_INCREMENT=2 **DEFAULT** CHARSET=utf8;**INSERT INTO** `user` **VALUES** (**'1'**, **'张三'**, **'安徽芜湖'**, **'27'**, **'153413543532'**); |

2、项目完整结构图



3、pom.xml文件配置

**Spring配置：（spring核心、spring Dao、spring MVC）**

**Servlet web的配置：（servlet-api、jsp-api、jstl）**

**数据库和mybatis：（mysql驱动、c3p0、mybatis、mybatis-spring）**

**日志配置：log4j**

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?> <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/xsd/maven-4.0.0.xsd">  <modelVersion>4.0.0</modelVersion>  <groupId>com.ssm</groupId>  <artifactId>MavernPro4ByIDEA</artifactId>  <version>1.0-SNAPSHOT</version>  <packaging>war</packaging>  <name>MavernPro4ByIDEA Maven Webapp</name>  <!-- FIXME change it to the project's website -->  <url>http://www.example.com</url>  <properties>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>  <spring.version>5.0.3.RELEASE</spring.version>  <mybatis.version>3.4.4</mybatis.version>  <maven.compiler.source>1.7</maven.compiler.source>  <maven.compiler.target>1.7</maven.compiler.target>  </properties>   <dependencies>  <dependency>  <groupId>junit</groupId>  <artifactId>junit</artifactId>  <version>4.11</version>  <scope>test</scope>  </dependency>  <!-- 第一部分：Spring 配置-->  <!-- Spring core -->  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-core</artifactId>  <version>${spring.version}</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-beans</artifactId>  <version>${spring.version}</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-context</artifactId>  <version>${spring.version}</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-context-support</artifactId>  <version>${spring.version}</version>  </dependency>  <!-- Spring DAO -->  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-jdbc</artifactId>  <version>${spring.version}</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-tx</artifactId>  <version>${spring.version}</version>  </dependency>  <!-- Spring mvc -->  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-web</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-test</artifactId>  <version>${spring.version}</version>  </dependency>  <!-- 第二部分：Servlet web -->  <dependency>  <groupId>javax.servlet</groupId>  <artifactId>javax.servlet-api</artifactId>  <version>3.0.1</version>  <scope>provided</scope>  </dependency>  <dependency>  <groupId>javax.servlet.jsp</groupId>  <artifactId>jsp-api</artifactId>  <version>2.2</version>  <scope>provided</scope>  </dependency>  <dependency>  <groupId>javax.servlet</groupId>  <artifactId>jstl</artifactId>  <version>1.2</version>  </dependency>  <dependency>  <groupId>com.fasterxml.jackson.core</groupId>  <artifactId>jackson-databind</artifactId>  <version>2.9.4</version>  </dependency>  <!-- 第三部分：数据库和mybatis -->  <!-- 数据库 -->  <dependency>  <groupId>mysql</groupId>  <artifactId>mysql-connector-java</artifactId>  <version>5.1.38</version>  </dependency>  <!-- 数据库连接池 -->  <dependency>  <groupId>com.mchange</groupId>  <artifactId>c3p0</artifactId>  <version>0.9.5.2</version>  </dependency>  <!-- MyBatis -->  <dependency>  <groupId>org.mybatis</groupId>  <artifactId>mybatis</artifactId>  <version>${mybatis.version}</version>  </dependency>  <!-- mybatis-spring整合包 -->  <dependency>  <groupId>org.mybatis</groupId>  <artifactId>mybatis-spring</artifactId>  <version>1.3.1</version>  </dependency>  <!-- 第四部分：日志 -->  <!-- 实现slf4j接口并整合 -->  <dependency>  <groupId>log4j</groupId>  <artifactId>log4j</artifactId>  <version>1.2.17</version>  </dependency>  </dependencies>  <build>  <finalName>MavernPro4ByIDEA</finalName>  <pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->  <plugins>  <plugin>  <artifactId>maven-clean-plugin</artifactId>  <version>3.1.0</version>  </plugin>  <!-- see http://maven.apache.org/ref/current/maven-core/default-bindings.html#Plugin\_bindings\_for\_war\_packaging -->  <plugin>  <artifactId>maven-resources-plugin</artifactId>  <version>3.0.2</version>  </plugin>  <plugin>  <artifactId>maven-compiler-plugin</artifactId>  <version>3.8.0</version>  </plugin>  <plugin>  <artifactId>maven-surefire-plugin</artifactId>  <version>2.22.1</version>  </plugin>  <plugin>  <artifactId>maven-war-plugin</artifactId>  <version>3.2.2</version>  </plugin>  <plugin>  <artifactId>maven-install-plugin</artifactId>  <version>2.5.2</version>  </plugin>  <plugin>  <artifactId>maven-deploy-plugin</artifactId>  <version>2.8.2</version>  </plugin>  </plugins>  </pluginManagement>  </build> </project> |

4、Web.xml配置（SpringMVC的拦截、加载spring配置文件、编码过滤、欢迎页）

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?> <web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee  http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"  version="3.1" metadata-complete="true">  <display-name>Archetype Created Web Application</display-name>   <welcome-file-list>  <welcome-file>index.jsp</welcome-file>  </welcome-file-list>   <servlet>  <servlet-name>dispatcher</servlet-name>  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>  <!-- 配置springMVC需要加载的配置文件  spring-dao.xml,spring-service.xml,spring-mvc.xml  Mybatis - > spring -> springmvc  -->  <init-param>  <param-name>contextConfigLocation</param-name>  <param-value>classpath:spring/spring-\*.xml</param-value>  </init-param>  </servlet>   <servlet-mapping>  <servlet-name>dispatcher</servlet-name>  <!-- 默认匹配所有的请求 -->  <url-pattern>/</url-pattern>  </servlet-mapping>   <filter>  <filter-name>encodingFilter</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>  </filter>   <filter-mapping>  <filter-name>encodingFilter</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping> </web-app> |

5、mybatis-config配置（使用别名、驼峰命令转换、主键自增）

|  |
| --- |
| <?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>  <!-- 配置全局属性 -->  <settings>  <!-- 使用jdbc的getGeneratedKeys获取数据库自增主键值 -->  <setting name="useGeneratedKeys" value="true" />   <!-- 使用列别名替换列名 默认:true -->  <setting name="useColumnLabel" value="true" />   <!-- 开启驼峰命名转换:Table{create\_time} -> Entity{createTime} -->  <setting name="mapUnderscoreToCamelCase" value="true" />   </settings> </configuration> |

6、spring-dao.xml配置（加载properties文件、数据源、SqlSession工厂、Dao接口）

|  |
| --- |
| <?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:context="http://www.springframework.org/schema/context"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd  http://www.springframework.org/schema/context  http://www.springframework.org/schema/context/spring-context.xsd">  <!-- 配置整合mybatis过程 -->  <!-- 1.配置数据库相关参数properties的属性：${url} -->  <context:property-placeholder location="classpath:jdbc.properties"/>  <!-- 2.数据库连接池 -->  <bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource">  <!-- 配置连接池属性 -->  <property name="driverClass" value="${jdbc.driver}"/>  <property name="jdbcUrl" value="${jdbc.url}"/>  <property name="user" value="${jdbc.username}"/>  <property name="password" value="${jdbc.password}"/>  <!-- c3p0连接池的私有属性 -->  <property name="maxPoolSize" value="30"/>  <property name="minPoolSize" value="10"/>  <!-- 关闭连接后不自动commit -->  <property name="autoCommitOnClose" value="false"/>  <!-- 获取连接超时时间 -->  <property name="checkoutTimeout" value="10000"/>  <!-- 当获取连接失败重试次数 -->  <property name="acquireRetryAttempts" value="2"/>  </bean>  <!-- 3.配置SqlSessionFactory对象 -->  <bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">  <!-- 注入数据库连接池 -->  <property name="dataSource" ref="dataSource"/>  <!-- 配置MyBaties全局配置文件:mybatis-config.xml -->  <property name="configLocation" value="classpath:mybatis-config.xml"/>  <!-- 扫描pojo包 使用别名 -->  <property name="typeAliasesPackage" value="com.ssm.entity"/>  <!-- 扫描sql配置文件:mapper需要的xml文件 -->  <property name="mapperLocations" value="classpath:mapper/\*.xml"/>  </bean>  <!-- 4.配置扫描Dao接口包，动态实现Dao接口，注入到spring容器中 -->  <bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">  <!-- 注入sqlSessionFactory -->  <property name="sqlSessionFactoryBeanName" value="sqlSessionFactory"/>  <!-- 给出需要扫描Dao接口包 -->  <property name="basePackage" value="com.ssm.dao"/>  </bean> </beans> |

7、spring-service.xml配置（事务管理器、事务注解扫描、service层扫描）

|  |
| --- |
| <?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:context="http://www.springframework.org/schema/context"  xmlns:tx="http://www.springframework.org/schema/tx"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd  http://www.springframework.org/schema/context  http://www.springframework.org/schema/context/spring-context.xsd  http://www.springframework.org/schema/tx  http://www.springframework.org/schema/tx/spring-tx.xsd">  <!-- 扫描service包下所有使用注解的类型 -->  <context:component-scan base-package="com.ssm.service" />  <!-- 配置事务管理器 -->  <bean id="transactionManager"  class="org.springframework.jdbc.datasource.DataSourceTransactionManager">  <!-- 注入数据库连接池 -->  <property name="dataSource" ref="dataSource" />  </bean>  <!-- 配置基于注解的声明式事务 -->  <tx:annotation-driven transaction-manager="transactionManager" /> </beans> |

8、spring-mvc.xml配置（controller层扫描、驱动注解、静态资源处理、视图解析）

|  |
| --- |
| <?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: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.xsd  http://www.springframework.org/schema/context  http://www.springframework.org/schema/context/spring-context.xsd  http://www.springframework.org/schema/mvc  http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd">  <!-- 配置SpringMVC -->  <!-- 1.开启SpringMVC注解模式 -->  <!-- 简化配置：  (1)自动注册DefaultAnootationHandlerMapping,AnotationMethodHandlerAdapter  (2)提供一些列：数据绑定，数字和日期的format @NumberFormat, @DateTimeFormat, xml,json默认读写支持  -->  <mvc:annotation-driven />  <!-- 2.静态资源默认servlet配置  (1)加入对静态资源的处理：js,gif,png  (2)允许使用"/"做整体映射  -->  <mvc:default-servlet-handler/>  <!-- 3.配置jsp 显示ViewResolver -->  <bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">  <property name="viewClass" value="org.springframework.web.servlet.view.JstlView" />  <property name="prefix" value="/WEB-INF/jsp/" />  <property name="suffix" value=".jsp" />  </bean>  <!-- 4.扫描web相关的bean -->  <context:component-scan base-package="com.ssm.controller" /> </beans> |

9、jdbc.properties配置（mysql的表、用户、密码等）

|  |
| --- |
| jdbc.driver=com.mysql.jdbc.Driver jdbc.url=jdbc:mysql://127.0.0.1:3306/db\_news?useUnicode=true&characterEncoding=utf8 jdbc.username=root jdbc.password=123456 |

10、Log4j.properties配置（控制台输出格式、输出每日的日志文件）

|  |
| --- |
| log4j.rootLogger=DEBUG, Console ,File ,DailyRollingFile  #Console log4j.appender.Console=org.apache.log4j.ConsoleAppender  log4j.appender.Console.layout=org.apache.log4j.PatternLayout  log4j.appender.Console.layout.ConversionPattern=%d [%t] %-5p [%c] - %m%n #File log4j.appender.File = org.apache.log4j.FileAppender log4j.appender.File.File = G://IDEA\_Pro//MavernPro4ByIDEA//log4j1.log log4j.appender.File.layout = org.apache.log4j.PatternLayout log4j.appender.File.layout.ConversionPattern =%d [%t] %-5p [%c] - %m%n #DailyRollingFile log4j.appender.DailyRollingFile = org.apache.log4j.DailyRollingFileAppender log4j.appender.DailyRollingFile.File = G://IDEA\_Pro//MavernPro4ByIDEA//log4j2.log log4j.appender.DailyRollingFile.layout = org.apache.log4j.PatternLayout log4j.appender.DailyRollingFile.layout.ConversionPattern =%d [%t] %-5p [%c] - %m%n |

1. User实体类

|  |
| --- |
| package com.ssm.entity; /\*\*  \* 说明：User实体类定义  \* @author 真好  \* @company 京东商城（上海）  \* @create 2019-01-09 12:41  \*/ public class User {  private int id;//用户ID  private String name;//用户姓名  private String address;//用户地址  private int age;//用户年龄  private String phone;//用户手机号码  get和set方法。。。。。  } |

1. userDao接口定义

|  |
| --- |
| package com.ssm.dao; import com.ssm.entity.User; import java.util.List; /\*\*  \* 说明：用户User的dao接口定义  \* @author 真好  \* @company 京东商城（上海）  \* @create 2019-01-09 12:43  \*/ public interface UserDao {  /\*\*  \* 添加用户  \* @param user  \* @return  \*/  int addUser(User user);  /\*\*  \* 根据用户ID删除用户  \* @param id  \* @return  \*/  int deleteUser(int id);   /\*\*  \* 根据用户Id查找用户  \* @param id  \* @return  \*/  User getUserById(int id);   /\*\*  \* 查询所有用户  \* @return  \*/  List<User> getUserList();   /\*\*  \* 更新用户信息  \* @param user  \* @return  \*/  int updateUser(User user); } |

1. userDaoMapper.xml映射文件

|  |
| --- |
| <?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="com.ssm.dao.UserDao">   <resultMap type="User" id="UserResultMap" >  <id property="id" column="id"/>  <result property="name" column="name"/>  <result property="address" column="address"/>  <result property="age" column="age"/>  <result property="phone" column="phone"/>  </resultMap>   <insert id="addUser" parameterType="User">  INSERT INTO user VALUES (null ,#{name}, #{address},#{age},#{phone})  </insert>   <delete id="deleteUser" parameterType="int">  DELETE FROM user WHERE id=#{id}  </delete>   <update id="updateUser" parameterType="User">  UPDATE user  SET name = #{name},address = #{address},age = #{age},phone=#{phone}  WHERE id = #{id}  </update>   <select id="getUserById" resultType="User" parameterType="int">  SELECT \*  FROM user  WHERE id=#{id}  </select>  <select id="getUserList" resultMap="UserResultMap">  SELECT \* FROM user  </select> </mapper> |

1. UserService接口定义

|  |
| --- |
| package com.ssm.service; import com.ssm.entity.User; import java.util.List;  /\*\*  \* 说明：用户UserService的接口定义  \* @author 真好  \* @company 京东商城（上海）  \* @create 2019-01-09 13:00  \*/ public interface UserService {   /\*\*  \* 添加用户  \* @param user  \* @return  \*/  int addUser(User user);   /\*\*  \* 根据用户ID删除用户  \* @param id  \* @return  \*/  int deleteUser(int id);   /\*\*  \* 根据用户Id查找用户  \* @param id  \* @return  \*/  User getUserById(int id);   /\*\*  \* 查询所有用户  \* @return  \*/  List<User> getUserList();   /\*\*  \* 更新用户信息  \* @param user  \* @return  \*/  int updateUser(User user); } |

1. UserServiceImpl实现类

|  |
| --- |
| package com.ssm.service.impl;  import com.ssm.dao.UserDao; import com.ssm.entity.User; import com.ssm.service.UserService; import org.springframework.beans.factory.annotation.Autowired; import org.springframework.stereotype.Service;  import java.util.List;  /\*\*  \* 说明：  \*  \* @author 真好  \* @company 京东商城（上海）  \* @create 2019-01-09 13:01  \*/ @Service("userService") public class UserServiceImpl implements UserService {   @Autowired  UserDao userDao;   @Override  public int addUser(User user) {  return userDao.addUser(user);  }   @Override  public int deleteUser(int id) {  return userDao.deleteUser(id);  }   @Override  public User getUserById(int id) {  return userDao.getUserById(id);  }   @Override  public List<User> getUserList() {  return userDao.getUserList();  }   @Override  public int updateUser(User user) {  return userDao.updateUser(user);  } } |

1. UserController控制类

|  |
| --- |
| package com.ssm.controller;  import com.ssm.entity.User; import com.ssm.service.UserService; import org.springframework.beans.factory.annotation.Autowired; import org.springframework.stereotype.Controller; import org.springframework.ui.Model; import org.springframework.web.bind.annotation.RequestMapping; import org.springframework.web.bind.annotation.RequestParam;  import java.util.List;  /\*\*  \* 说明：用户controller类定义  \* @author 真好  \* @company 京东商城（上海）  \* @create 2019-01-09 13:05  \*/ @Controller @RequestMapping("/user") public class UserController {   @Autowired  UserService userService;   /\*\*  \* 根据请求是否有id，判断是“跳转到addUser.jsp”，还是跳转到“updateUser.jsp”  \* @param id  \* @param model  \* @return  \*/  @RequestMapping("/toAddUser")  public String toAddUser(@RequestParam(value = "id" , required = false) Integer id , Model model){  if(id == null){ // 是增加的请求  return "addUser";  }else{  User ResultUser = userService.getUserById(id);  model.addAttribute("currentUser" , ResultUser);  return "updateUser";  }  }   /\*\*  \* 显示所有用户信息  \* @param model  \* @return  \*/  @RequestMapping("/userList")  public String userList(Model model){  List<User> users = userService.getUserList();  model.addAttribute("userList",users);  return "userList";  }   /\*\*  \* 根据id是否为空，判断是添加还是更新操作，并重定向到用户列表  \* @param user  \* @param id  \* @return  \*/  @RequestMapping("/doAddUser")  public String doAddUser(User user ,@RequestParam(value = "id" , required = false) Integer id ){  if(id == null){ //添加用户到数据库中  userService.addUser(user);  }else{ //更新用户到数据库中  userService.updateUser(user);  }  return "redirect:/user/userList";  }   /\*\*  \* 根据用户Id，删除用户  \* @param id  \* @return  \*/  @RequestMapping("/deleteUser")  public String deleteUser(@RequestParam("id") Integer id){  userService.deleteUser(id);  return "redirect:/user/userList";  } } |

1. index.jsp

|  |
| --- |
| <%--  User: 23237  Date: 2019/1/9  Time: 13:34 --%> <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %> <!--引入jstl标签--> <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %> <%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %> <%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %> <%  String path = request.getContextPath();  String basePath = request.getScheme() + "://"  + request.getServerName() + ":" + request.getServerPort() + path + "/"; %> <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"> <html> <head>  <!--使用basePath——项目名/-->  <base href="<%=basePath%>">  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  <title>Title</title>  <!--用于seo优化-->  <meta name="Author" content="Java1234\_小锋老师">  <meta name="keywords" content="Java1234"/>  <meta name="description" content="Java1234"/>  <!-- 引入 Bootstrap -->  <link href="https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/css/bootstrap.min.css" rel="stylesheet"> </head> <body>  <div class="container">  <div class="jumbotron">  <div class="container">  <h1>欢迎使用“用户管理系统”</h1>  <p>本系统采用IDEA开发工具，使用maven管理项目依赖，采用当前比较流行的Spring+SpringMVC+Mybatis框架，实现了用户的增删查改的基本功能</p>  <p><a class="btn btn-primary btn-lg" href="user/userList" role="button">显示所有用户</a></p>  </div>  </div> </div>  <script src="https://cdn.jsdelivr.net/npm/jquery@1.12.4/dist/jquery.min.js"></script> <script src="https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/js/bootstrap.min.js"></script> </body> </html> |

1. userList.jsp

|  |
| --- |
| <%--  User: 23237  Date: 2019/1/9  Time: 13:39 --%> <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %> <!--引入jstl标签--> <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %> <%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %> <%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %> <%  String path = request.getContextPath();  String basePath = request.getScheme() + "://"  + request.getServerName() + ":" + request.getServerPort() + path + "/"; %> <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"> <html> <head>  <!--使用basePath——项目名/-->  <base href="<%=basePath%>">  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  <title>Title</title>  <!--用于seo优化-->  <meta name="Author" content="Java1234\_小锋老师">  <meta name="keywords" content="Java1234"/>  <meta name="description" content="Java1234"/>  <!-- 引入 Bootstrap -->  <link href="https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/css/bootstrap.min.css" rel="stylesheet"> </head> <body>  <div class="container">  <a class="btn btn-success" href="user/toAddUser" role="button">添加用户</a>  <table class="table table-hover">  <tr>  <td >用户ID</td>  <td >姓名</td>  <td >地址</td>  <td >年龄</td>  <td >手机号码</td>  <td >操作</td>  </tr>   <c:forEach items="${userList}" var="user">  <tr>  <td >${user.id}</td>  <td >${user.name}</td>  <td >${user.address}</td>  <td >${user.age}</td>  <td >${user.phone}</td>  <td >  <a href = "user/deleteUser?id=${user.id}">[删除]&nbsp;&nbsp;</a>  <a href="user/toAddUser?id=${user.id}">[更新]</a>  </td>  </tr>  </c:forEach>   </table> </div>  <script src="https://cdn.jsdelivr.net/npm/jquery@1.12.4/dist/jquery.min.js"></script> <script src="https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/js/bootstrap.min.js"></script> </body> </html> |

1. addUser.jsp

|  |
| --- |
| <%--  User: 23237  Date: 2019/1/9  Time: 13:39 --%> <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>  <%  String path = request.getContextPath();  String basePath = request.getScheme() + "://"  + request.getServerName() + ":" + request.getServerPort() + path + "/"; %> <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"> <html> <head>  <!--使用basePath——项目名/-->  <base href="<%=basePath%>">  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  <title>Title</title>  <!--用于seo优化-->  <meta name="Author" content="Java1234\_小锋老师">  <meta name="keywords" content="Java1234"/>  <meta name="description" content="Java1234"/>  <!-- 引入 Bootstrap -->  <link href="https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/css/bootstrap.min.css" rel="stylesheet"> </head> <body> <div class="container">  <div class="panel panel-default">  <div class="panel-heading text-center">  <h3 class="panel-title">添加用户</h3>  </div>  <div class="panel-body">   <form action="user/doAddUser" method="post">  <div class="form-group">  <label for="name">姓名：</label>  <input type="text" class="form-control" name = "name" id="name" placeholder="请输入姓名...">  </div>  <div class="form-group">  <label for="address">地址：</label>  <input type="text" class="form-control" name = "address" id="address" placeholder="请输入地址...">  </div>  <div class="form-group">  <label for="age">年龄：</label>  <input type="text" class="form-control" name = "age" id="age" placeholder="请输入年龄...">  </div>  <div class="form-group">  <label for="phone">手机号码：</label>  <input type="text" class="form-control" name = "phone" id="phone" placeholder="请输入手机号码...">  </div>   <button type="submit" class="btn btn-default">添加</button>  </form>   </div>  </div> </div> <script src="https://cdn.jsdelivr.net/npm/jquery@1.12.4/dist/jquery.min.js"></script> <script src="https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/js/bootstrap.min.js"></script> </body> </html> |

1. updateUser.jsp

|  |
| --- |
| <%--  User: 23237  Date: 2019/1/9  Time: 13:39 --%> <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %> <!--引入jstl标签--> <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %> <%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %> <%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %> <%  String path = request.getContextPath();  String basePath = request.getScheme() + "://"  + request.getServerName() + ":" + request.getServerPort() + path + "/"; %> <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"> <html> <head>  <!--使用basePath——项目名/-->  <base href="<%=basePath%>">  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  <title>Title</title>  <!--用于seo优化-->  <meta name="Author" content="Java1234\_小锋老师">  <meta name="keywords" content="Java1234"/>  <meta name="description" content="Java1234"/>  <!-- 引入 Bootstrap -->  <link href="https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/css/bootstrap.min.css" rel="stylesheet"> </head> <body>  <div class="container">  <div class="panel panel-default">  <div class="panel-heading text-center">  <h3 class="panel-title">修改用户</h3>  </div>  <div class="panel-body">   <form action="user/doAddUser?id=${currentUser.id}" method="post">  <div class="form-group">  <label for="name">姓名：</label>  <input type="text" class="form-control" name = "name" value="${currentUser.name}" id="name">  </div>  <div class="form-group">  <label for="address">地址：</label>  <input type="text" class="form-control" name = "address" value="${currentUser.address}" id="address">  </div>  <div class="form-group">  <label for="age">年龄：</label>  <input type="text" class="form-control" name = "age" value="${currentUser.age}" id="age">  </div>  <div class="form-group">  <label for="phone">手机号码：</label>  <input type="text" class="form-control" name = "phone" value="${currentUser.phone}" id="phone">  </div>   <button type="submit" class="btn btn-default">更新</button>  </form>   </div>  </div> </div> <script src="https://cdn.jsdelivr.net/npm/jquery@1.12.4/dist/jquery.min.js"></script> <script src="https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/js/bootstrap.min.js"></script> </body> </html> |