实验11 Spring MVC框架：SSM框架整合

**【实验目的及要求】**

1. 掌握SSM框架整合方法；
2. 完成成绩管理系统的课程管理模块；
3. 基于SSM框架的系统设计方法；
4. 要求所有回答的文本格式：**五号，宋体、1.5倍行距，保留段单元格背景。**

**【实验步骤】**

1. **基于配置文件的SSM框架整合（学生信息管理模块）**

（1）在IDEA中新建Maven项目gms（Grade Management System），启动MySQL数据库服务，采用本地Tomcat配置项目的Web服务环境。

（2）登录MySQL数据库，执行下列SQL脚本。

|  |
| --- |
| -- student数据库创建示例  create database student character set gbk;  use student;  -- 设计客户端默认字符集  SET @saved\_cs\_client = @@character\_set\_client;  SET @@character\_set\_client = gbk;  -- 含指定引擎和默认字符集  create table stu(  sno char(9) primary key,  sname varchar(30) not null,  ssex char(2) not null,  snative varchar(30),  mno int  )ENGINE=InnoDB DEFAULT CHARSET=gbk;  set names gbk;  insert into stu values('100000001','尚小云','女','广东广州',1);  insert into stu values('100000002','廖时飞','男','广东梅州',1);  insert into stu values('100000003','宋凌枫','男','湖南郴州',2);  insert into stu values('100000004','刘小纳', '女','广东佛山',2);  create table course(  cno int primary key,  cname varchar(50) not null,  period int  )ENGINE=InnoDB DEFAULT CHARSET=gbk;  set names gbk;  insert into course values(1,'高等数学',80);  insert into course values(2,'大学英语',70);  insert into course values(3,'数据结构',70);  insert into course values(4,'数据库原理与应用',70);  create table teacher(  tno int primary key,  tname varchar(24) not null,  tsex char(2) not null,  tel varchar(20)  )ENGINE=InnoDB DEFAULT CHARSET=gbk;  set names gbk;  insert into teacher values(1,'袁怀斌','男','13590909090');  insert into teacher values(2,'杨文远','男','13588888888');  insert into teacher values(3,'周云霞','女','13511118888');  create table sc(  sno char(9),  cno int,  tno int,  participation float,  final float,  total float,  constraint fk\_sc\_stu foreign key(sno) references stu(sno),  constraint fk\_sc\_course foreign key(cno) references course(cno),  constraint fk\_sc\_teacher foreign key(tno) references teacher(tno),  constraint pk\_sc primary key(sno,cno,tno)  )ENGINE=InnoDB DEFAULT CHARSET=gbk;  set names gbk;  insert into sc values('100000001',1,1,80,87,85);  insert into sc values('100000001',2,2,85,75,81);  insert into sc values('100000002',1,1,78,82,80);  -- 恢复客户端默认字符集  SET @@character\_set\_client = @saved\_cs\_client; |

（3）根据三个框架的要求，在pom文件中配置好依赖。

（4）在webapp目录中修改系统首页index.jsp，提供进入学生信息管理的超链接，内容如下。

|  |
| --- |
| <%@ 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>成绩管理系统（GMS）</title>  </head>  <body>  <a href=*"studentlist"*>学生信息管理</a>  </body>  </html> |

（5）在/WEB-INF/jsp/student目录下，新建studentlist.jsp文件，提供学生信息的列表显示，以及学生信息的添加、删除和更新操作，内容如下。

|  |
| --- |
| <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"* %>  <!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>学生信息列表</title>  <script>  **function** frmSubmit(){  document.form1.submit();  }  </script>  </head>  <body>  <center>学生信息</center>  <form id=*"form1"* name=*"form1"* method=*"post"* action=*"studentdelete"*>  <table align=*"center"* width=*"500"* border=*"1"* cellspacing=*"0"* cellpadding=*"0"* style="border-collapse:*collapse*" bordercolor=*"#0099FF"*>  <tr>  <td>选择</td>  <td>学号</td>  <td>姓名</td>  <td>性别</td>  <td>籍贯</td>  <td>专业号</td>  <td>操作&nbsp;&nbsp;<a href=*"studentpreinsert"*>添加</a>&nbsp;&nbsp;<a href=*"#"* onclick="javascript:frmSubmit();">删除</a></td>  </tr>  <c:forEach items=*"*${studentList}*"* var=*"student"*>  <tr>  <td><input type=*"checkbox"* name=*"snoArray"* value=*"*${student.sno}*"*></td>  <td>${student.sno}</td>  <td>${student.sname}</td>  <td>${student.ssex}</td>  <td>${student.snative}</td>  <td>${student.mno}</td>  <td>  <a href=*"studentpreupdate?sno=*${student.sno}*&sname=*${student.sname}*&ssex=*${student.ssex}*&snative=*${student.snative}*&mno=*${student.mno}*"*>修改</a>  </td>  </tr>  </c:forEach>  </table>  </form>  </body>  </html> |

**说明**：该网页中用到了JSTL，因此需要使用标签<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"* %>导入标签库。

（6）在/WEB-INF/jsp/student目录下，新建studentupdate.jsp文件，提供学生信息的修改表单，内容如下。

|  |
| --- |
| <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>修改学生信息</title>  </head>  <body>  <form id=*"form1"* name=*"form1"* method=*"post"* action=*"studentupdate"*>  <table align=*"center"* width=*"500"* border=*"1"* cellspacing=*"0"* cellpadding=*"0"* style="border-collapse:*collapse*" bordercolor=*"#0099FF"*>  <tr>  <td width=*"116"* height=*"30"* align=*"right"* valign=*"middle"*>学号：</td>  <td width=*"378"* align=*"left"* valign=*"middle"*>  <input type=*"text"* name=*"sno"* readonly=*"readonly"* id=*"sno"* value=*"*${student.sno}*"*/></td>  </tr>  <tr>  <td width=*"116"* height=*"30"* align=*"right"* valign=*"middle"*>姓名：</td>  <td width=*"378"* align=*"left"* valign=*"middle"*>  <input type=*"text"* name=*"sname"* id=*"sname"* value=*"*${student.sname}*"*/></td>  </tr>  <tr>  <td width=*"116"* height=*"30"* align=*"right"* valign=*"middle"*>性别：</td>  <td width=*"378"* align=*"left"* valign=*"middle"*>  <input type=*"text"* name=*"ssex"* id=*"ssex"* value=*"*${student.ssex}*"* /></td>  </tr>  <tr>  <td width=*"116"* height=*"30"* align=*"right"* valign=*"middle"*>籍贯：</td>  <td width=*"378"* align=*"left"* valign=*"middle"*>  <input type=*"text"* name=*"snative"* id=*"snative"* value=*"*${student.snative}*"* /></td>  </tr>  <tr>  <td width=*"116"* height=*"30"* align=*"right"* valign=*"middle"*>专业：</td>  <td width=*"378"* align=*"left"* valign=*"middle"*>  <input type=*"text"* name=*"mno"* id=*"mno"* value=*"*${student.mno}*"* /></td>  </tr>  <tr>  <td height=*"30"* align=*"right"* valign=*"middle"*>&nbsp;</td>  <td align=*"left"* valign=*"middle"*><input type=*"submit"* name=*"button"* id=*"button"* value=*"提交"* />  <input type=*"reset"* name=*"button2"* id=*"button2"* value=*"重置"* /></td>  </tr>  </table>  </form>  </body>  </html> |

（7）在/WEB-INF/jsp/student目录下，新建studentadd.jsp文件，提供添加学生信息的表单，内容如下。

|  |
| --- |
| <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>添加学生信息</title>  </head>  <body>  <form id=*"form1"* name=*"form1"* method=*"post"* action=*"studentinsert"*>  <table align=*"center"* width=*"500"* border=*"1"* cellspacing=*"0"* cellpadding=*"0"* style="border-collapse:*collapse*" bordercolor=*"#0099FF"*>  <tr>  <td width=*"116"* height=*"30"* align=*"right"* valign=*"middle"*>学号：</td>  <td width=*"378"* align=*"left"* valign=*"middle"*>  <input type=*"text"* name=*"sno"* id=*"sno"* /></td>  </tr>  <tr>  <td width=*"116"* height=*"30"* align=*"right"* valign=*"middle"*>姓名：</td>  <td width=*"378"* align=*"left"* valign=*"middle"*>  <input type=*"text"* name=*"sname"* id=*"sname"* /></td>  </tr>  <tr>  <td width=*"116"* height=*"30"* align=*"right"* valign=*"middle"*>性别：</td>  <td width=*"378"* align=*"left"* valign=*"middle"*>  <input type=*"text"* name=*"ssex"* id=*"ssex"* /></td>  </tr>  <tr>  <td width=*"116"* height=*"30"* align=*"right"* valign=*"middle"*>籍贯：</td>  <td width=*"378"* align=*"left"* valign=*"middle"*>  <input type=*"text"* name=*"snative"* id=*"snative"* /></td>  </tr>  <tr>  <td width=*"116"* height=*"30"* align=*"right"* valign=*"middle"*>专业：</td>  <td width=*"378"* align=*"left"* valign=*"middle"*>  <input type=*"text"* name=*"mno"* id=*"mno"* /></td>  </tr>  <tr>  <td height=*"30"* align=*"right"* valign=*"middle"*>&nbsp;</td>  <td align=*"left"* valign=*"middle"*><input type=*"submit"* name=*"button"* id=*"button"* value=*"提交"* />  <input type=*"reset"* name=*"button2"* id=*"button2"* value=*"重置"* /></td>  </tr>  </table>  </form>  </body>  </html> |

（8）在/WEB-INF目录中，修改web.xml，提供创建Spring容器、编码过滤器及Spring MVC前端控制器的配置信息，内容如下。

|  |
| --- |
| <?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"* xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"* id=*"WebApp\_ID"* version=*"3.0"*>  <display-name>gms</display-name>  <welcome-file-list>  <welcome-file>index.jsp</welcome-file>  </welcome-file-list>  <!-- 配置加载Spring文件的监听器-->  <context-param>  <param-name>contextConfigLocation</param-name>  <param-value>classpath:config/applicationContext.xml</param-value>  </context-param>  <listener>  <listener-class>  org.springframework.web.context.ContextLoaderListener  </listener-class>  </listener>  <!-- 编码过滤器 -->  <filter>  <filter-name>encoding</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>encoding</filter-name>  <url-pattern>\*</url-pattern>  </filter-mapping>  <!--Spring MVC前端器 -->  <servlet>  <servlet-name>gms</servlet-name>  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>  <init-param>  <param-name>contextConfigLocation</param-name>  <param-value>classpath:config/springmvc-config.xml</param-value>  </init-param>  <load-on-startup>1</load-on-startup>  </servlet>  <servlet-mapping>  <servlet-name>gms</servlet-name>  <url-pattern>/</url-pattern>  </servlet-mapping>  </web-app> |

（9）在/main/resouces/config目录中，添加Spring框架的配置文件applicationContext.xml，提供数据源、事务、Mybatis会话工厂、mapper扫描器及bean扫描器等配置信息，内容如下。

|  |
| --- |
| *<?*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:tx="http://www.springframework.org/schema/tx"  xmlns:context="http://www.springframework.org/schema/context"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans-4.3.xsd  http://www.springframework.org/schema/tx  http://www.springframework.org/schema/tx/spring-tx-4.3.xsd  http://www.springframework.org/schema/context  http://www.springframework.org/schema/context/spring-context-4.3.xsd">  *<!-- 读取db.properties -->* <context:property-placeholder location="classpath:config/db.properties"/>  *<!-- 配置数据源 -->* <bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource">  *<!--数据库驱动 -->* <property name="driverClassName" value="${jdbc.driver}" />  *<!--连接数据库的url -->* <property name="url" value="${jdbc.url}" />  *<!--连接数据库的用户名 -->* <property name="username" value="${jdbc.username}" />  *<!--连接数据库的密码 -->* <property name="password" value="${jdbc.password}" /> *<!-- &lt;!&ndash;最大连接数 &ndash;&gt;--> <!-- <property name="maxTotal" value="${jdbc.maxTotal}" />--> <!-- &lt;!&ndash;最大空闲连接 &ndash;&gt;--> <!-- <property name="maxIdle" value="${jdbc.maxIdle}" />--> <!-- &lt;!&ndash;初始化连接数 &ndash;&gt;--> <!-- <property name="initialSize" value="${jdbc.initialSize}" />-->* </bean>  *<!-- 事务管理器，依赖于数据源 -->* <bean id="transactionManager" class=  "org.springframework.jdbc.datasource.DataSourceTransactionManager">  <property name="dataSource" ref="dataSource" />  </bean>  *<!-- 开启事务注解 -->* <tx:annotation-driven transaction-manager="transactionManager"/>  *<!-- 配置MyBatis工厂SqlSessionFactory -->* <bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">  *<!--注入数据源 -->* <property name="dataSource" ref="dataSource" />  *<!--指定核MyBatis心配置文件位置 -->* <property name="configLocation" value="classpath:config/mybatis-config.xml" />  </bean>  *<!-- 配置mapper扫描器 -->* <bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">  <property name="basePackage" value="com.mapper"/>  </bean>  *<!-- 扫描Service -->* <context:component-scan base-package="com.service" /> </beans> |

说明：上述配置文件中引用的db.properties文件也要放在/main/resouces/config目录中，内容如下。

|  |
| --- |
| jdbc.driver=com.mysql.jdbc.Driver  jdbc.url=jdbc:mysql://localhost:3306/student?characterEncoding=GBK  jdbc.username=root  jdbc.password=root  jdbc.maxTotal=30  jdbc.maxIdle=10  jdbc.initialSize=5 |

（10）在/main/resouces/config目录中，添加Mybatis配置文件mybatis-config.xml，添加别名配置信息，内容如下。

|  |
| --- |
| <?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>  <!-- 别名定义 -->  <typeAliases>  <package name=*"com.po"* />  </typeAliases>  </configuration> |

（11）在/main/resouces/config目录中，添加Spring MVC配置文件springmvc-config.xml，添加控制器扫描及视图解析器配置信息，内容如下。

|  |
| --- |
| <?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:mvc=*"http://www.springframework.org/schema/mvc"*  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-4.3.xsd*  *http://www.springframework.org/schema/mvc*  *http://www.springframework.org/schema/mvc/spring-mvc-4.3.xsd*  *http://www.springframework.org/schema/context*  *http://www.springframework.org/schema/context/spring-context-4.3.xsd"*>  <!-- 配置包扫描器，扫描@Controller注解的类 -->  <context:component-scan base-package=*"com.controller"* />  <!-- 加载注解驱动 -->  <mvc:annotation-driven />  <!-- 配置视图解析器 -->  <bean class=  *"org.springframework.web.servlet.view.InternalResourceViewResolver"*>  <property name=*"prefix"* value=*"/WEB-INF/jsp/"* />  <property name=*"suffix"* value=*".jsp"* />  </bean>  </beans> |

（12）在com.po包中，添加实体类Student，代码如下。

|  |
| --- |
| **package** com.po;  **public** **class** Student {  **private** String sno;  **private** String sname;  **private** String ssex;  **private** String snative;  **private** **int** mno;  // 省略了Getter和Setter方法  } |

（13）在com.service包中，添加接口StudentService，其方法用于实现学生信息管理。

|  |
| --- |
| **package** com.service;  **import** java.util.List;  **import** com.po.Student;  **public** **interface** StudentService {  List<Student> findAllStudent();  Student findStudentBySno(String sno);  **void** addStudent(Student student);  **void** deleteStudents(String[] snoArray);  **void** updateStudent(Student student);  } |

（14）在com.service.impl包中，添加接口StudentService的实现类，代码如下。

|  |
| --- |
| package com.service.impl;  import java.util.List;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Service;  import org.springframework.transaction.annotation.Transactional;  import com.mapper.StudentMapper;  import com.po.Student;  import com.service.StudentService;  @Service  @Transactional  public class StudentServiceImpl implements StudentService{  @Autowired  private StudentMapper studentMapper;    public List<Student> findAllStudent(){  return studentMapper.selectStudents();  }  public Student findStudentBySno(String sno){  return studentMapper.selectStudentBySno(sno);  }  public void addStudent(Student student){  studentMapper.insertStudent(student);  }  public void deleteStudents(String[] snoArray){  for (int i = 0; i < snoArray.length; i ++){  studentMapper.deleteStudentBySno(snoArray[i]);  }  }  public void updateStudent(Student student){  studentMapper.updateStudent(student);  }  } |

（15）在com.mapper包中，添加接口StudentMapper，提示对数据表stu的基本访问方法。

|  |
| --- |
| package com.mapper;  import java.util.List;  import com.po.Student;  public interface StudentMapper {  Student selectStudentBySno(String sno);  List<Student> selectStudentBySname(String sname);  List<Student> selectStudents();  void insertStudent(Student student);  void updateStudent(Student student);  void deleteStudentBySno(String sno);  } |

（16）在resources/com/mapper文件夹中，添加stu表的映射文件StudentMapper.xml，内容如下。

|  |
| --- |
| <?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.mapper.StudentMapper"*>  <!--按学号查询学生信息 -->  <select id=*"selectStudentBySno"* parameterType=*"String"* resultType=*"Student"*>  select \* from stu where sno=#{sno};  </select>  <!-- 按姓名查询学生信息 -->  <select id=*"selectStudentBySname"* parameterType=*"String"* resultType=*"Student"*>  <bind name=*"sname"* value=*"'%'+sname+'%'"* />  select \* from stu  <where>  <if test=*"sname!=null and sname!=''"*>  and sname like #{sname}  </if>  </where>  </select>  <!--按所有学生信息 -->  <select id=*"selectStudents"* resultType=*"Student"*>  select \* from stu;  </select>  <insert id=*"insertStudent"* parameterType=*"Student"*>  insert into stu(sno,sname,ssex,snative,mno) values(#{sno},#{sname},#{ssex},#{snative},#{mno});  </insert>  <delete id=*"deleteStudentBySno"* parameterType=*"String"*>  delete from stu where sno=#{sno};  </delete>  <update id=*"updateStudent"* parameterType=*"Student"*>  update stu  <set>  <if test=*"sname!=null and sname!=''"*>  sname=#{sname},  </if>  <if test=*"ssex!=null and ssex!=''"*>  ssex=#{ssex},  </if>  <if test=*"snative!=null"*>  snative=#{snative},  </if>  <if test=*"mno!=null and mno!=''"*>  mno=#{mno},  </if>  </set>  where sno=#{sno};  </update>  </mapper> |

（17）在com.controller包中，添加控制器类StudentController，实现学生信息的添加、删除和更新操作，内容如下。

|  |
| --- |
| package com.controller;  import java.util.List;  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.RequestMethod;  import com.po.Student;  import com.service.StudentService;  @Controller  public class StudentController {  @Autowired  private StudentService studentService;    @RequestMapping("/studentlist")  public String studentList(Model model) {  List<Student> list = studentService.findAllStudent();  model.addAttribute("studentList", list);  return "student/studentlist";  }    @RequestMapping(value="/studentpreinsert",method=RequestMethod.GET)  public String studentPreinsert() {  return "student/studentadd";  }    @RequestMapping(value="/studentinsert", method=RequestMethod.POST)  public String studentInsert(Student student){  studentService.addStudent(student);  return "redirect:studentlist";  }    @RequestMapping(value="/studentdelete", method=RequestMethod.POST)  public String studentDelete(String[] snoArray) {  studentService.deleteStudents(snoArray);  return "redirect:studentlist";  }    @RequestMapping(value="/studentpreupdate", method=RequestMethod.GET)  public String studentPreupdate(Student student, Model model) {  model.addAttribute("student", student);  return "student/studentupdate";  }    @RequestMapping(value="/studentupdate", method=RequestMethod.POST)  public String studentUpdate(Student student) {  studentService.updateStudent(student);  return "redirect:studentlist";  }  } |

（18）在Web环境中测试程序的有效性。

（19）模仿上述过程，在gms项目中，添加课程信息管理模块，主要内容如下。

|  |
| --- |
| courselist.jsp文件内容： |
| Courseadd.jsp文件内容： |
| Courseupdate.jsp文件内容： |
| 实体类Course代码： |
| 业务层接口CourseService代码： |
| 接口CourseService的实现类CourseServiceImpl代码： |
| 映射文件CourseMapper.xml内容： |
| 接口CourseMapper代码： |
| 控制器CourseController代码： |

（20）根据上述实验步骤，归纳总结在SSM框架中，实现Web系统各模块功能的基本思路。

|  |
| --- |
|  |

1. **基于配置类的SSM框架整合（学生信息管理模块）**
2. 以配置类方式整合SSM框架，要求实现“学生信息管理模块”，即完成student表的查询、插入、删除和更新操作。
3. **要求：项目名称以自己姓名拼音的首字母取名，如张晓峰的项目名为zxf；该项目要求提交源码的压缩文件，以“学号\_姓名”命名。**