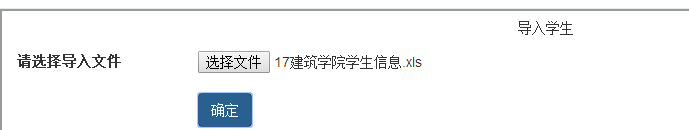
# 添加学生信息

1.导入学生对应的响应事件是数据库里对应的importStudents.jsp：



1. 添加importstudents.jsp页面



|  |
| --- |
| 1. .这里用了bootstrap里面集成好的文件样式 让排版更好看了   例子：  导入外部样式文件：  <link rel="stylesheet" href="https://cdn.bootcss.com/bootstrap/3.3.7/css/bootstrap.min.css">  <div class="form-group">  <label for="inputfile">文件输入</label>  <input type="file" id="inputfile">  </div>  效果：     1. .原作者在这里用了栅格系统，让网页更好的适应不同大小屏幕的显示器 2. .核心实现代码   <div class=*"ibox-content"*>  <form id=*"file\_form"* action=*"importExcleStudent"*  method=*"post"*>  <div class=*"form-group"*>  <label class=*"col-sm-2 control-label"*>请选择导入文件</label>  <div class=*"col-sm-10"*>  <input type=*"file"* name="importStudent" id=*"file\_input"*  class=*"filename"* /> <br />  <input type=*"submit"* class=*"btn btn-primary"*  value=*"确定"* id=*'upFile-btn'*>  </div>  </div>  </form>  </div>  Form里面指定了对应的action：importExcleStudent.action 然后就要通过这个action实现exl表格向数据库的导入  这里对界面还需要完善的是 判断是否添加了文件 判断添加的文件格式是否正确 判断文件是否上传成功   1. 添加bootstrap外部依赖文件   <script type=*"text/javascript"* src=*"jules/js/jquery.min.js"*></script>  <script type=*"text/javascript"* src=*"js/bootstrap.min.js"*></script>  <script type=*"text/javascript"* src=*"js/bootstrap-multiselect.js"*></script>  注：jquery文件一定要添加在bootstrap之前，否则没有作用  添加script方法：  <script type=*"text/javascript"*>  $(**function**() {  $("#file\_form").submit(  **function**() {  //首先验证文件格式  **var** fileName = $('#file\_input').val();  **if** (fileName === '') {  alert('请选择文件');  **return false**;  }  **var** fileType = (fileName.substring(fileName  .lastIndexOf(".") + 1, fileName.length))  .toLowerCase();  **if** (fileType !== 'xls') {  alert('文件格式不正确，excel文件！');  **return false**;  }  $("#file\_form").ajaxSubmit({  dataType : "json",  success : **function**(data, textStatus) {  **if** (data['s'] === 'OK') {  alert("文件上传成功");  // window.location.href = "https://localhost:8080/SDSYw/excleFile/importExcleStudent.log";  $('#file\_input').html("");  console.log('上传文件成功');  } **else** {  console.log('文件格式错误');  }    **return false**;  }  });  **return false**;  });  });  </script> |

**新知识：（表单提交）**

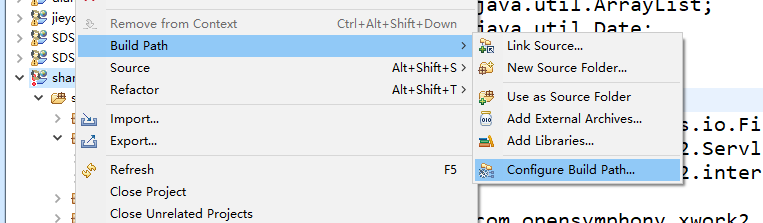
|  |
| --- |
| 1. $(selector).submit(function)这样提交表单的话：提交后当前页面直接跳转到form的action所指向的页面。 2. $(selector).ajaxSubbmit(obj)这样提交表单会等到表单提交完后在再触发事件   3 . 我们这个表单的功能是希望：点了提交后 判断是否选取了文件 和 判断文件格式是否正确 然后这些都正确了后提交到action 然后action里的方法执行完了后网页再判断文件是否上成功。所以用subbmit是无法实现第二个回调功能。所以我们要用 ajaxSubmit(obj)来提交数据  注意：这两个方法提交完后都要返回 false  // 阻止表单自动提交事件，必须返回false，否则表单会自己再做一次提交操作，并且页面跳转  // 必须返回false，否则表单会自己再做一次提交操作，并且页面跳转 |

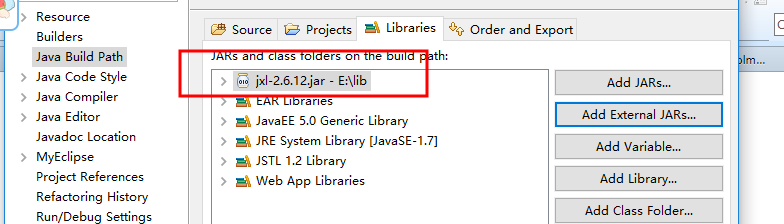
1. struts.xml添加对应的action：*importExcleStudent.action*

|  |
| --- |
| <action name=*"importExcleStudent"* method=*"importExcleStudent"* class=*"excle"*>  <result type=*"json"* name=*"success"*>  <param name=*"root"*>s</param>  </result>  </action> |

1. 新建对excel表格操作的类ExcleAction.java
2. ）因为要对excel表格操作，要先导入架包 jxl架包

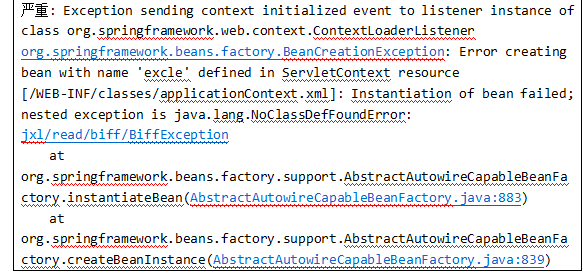
（可以在这网站下载http://jareye.com/）







注意：光这样导入时没用的，会报下面的错误



我们需要把jxk架包放在lib里面才行

1. 在ExcelAction类添加方法importExcleStudent()方法

/// 导入，更新学生数据

**public** String importExcleStudent() **throws** IOException {

}

1. 新建对学生的操作接口StudentDao.java

|  |
| --- |
| 新建方法：  **public** **interface** StudentDao {  **public** Students isexist(String stuId);//判断是否有这学生  **public** **void** updatestudent(Students stu);//更新学生信息  **public** **void** addstudent(Students stu);//添加学生  } |

1. 新建类StudentDaoImp.java类实现接口

|  |
| --- |
| **public** **class** StudentDaoImp **extends** HibernateDaoSupport **implements** StudentDao {  @Override  **public** Students isexist(String stuId) {  // **TODO** Auto-generated method stub  List<Students> list = getHibernateTemplate().find(  "from Students s where s.id.SStudentId=?", stuId);  **if** (list.size() > 0) {  System.*out*.println("存在"+list.get(0).getSName());  **return** list.get(0);  }**else**  **return** **null**;  }  @Override  **public** **void** updatestudent(Students stu) {  **this**.getHibernateTemplate().update(stu);  }  @Override  **public** **void** addstudent(Students stu) {  **this**.getHibernateTemplate().save(stu);  }  } |

1. 在application里面添加studentdaoimp的映射实现自动new

|  |
| --- |
| <bean id=*"studentDaoImp"* class=*"web.dao.imp.StudentDaoImp"*>  <property name=*"sessionFactory"* ref=*"mysessionFactory"*></property>  </bean> |

1. 定义变量。并生成get set方法

|  |
| --- |
| **private** Timestamp createTime;  **private** File importStudent;  **private** String importStudentFileName;//这里一定是要对应的  **private** JSONObject s;  **private** StudentDao studentDao; |

1. StudentDao需要在application.xml添加响应

|  |
| --- |
| <bean id=*"excle"* class=*"web.action.ExcleAction"*>  <property name=*"studentDao"* ref=*"studentDaoImp"*></property>  </bean> |

1. 添加学生信息修改的的时间信息

|  |
| --- |
| **private** **static** Timestamp getTime() {  SimpleDateFormat df = **new** SimpleDateFormat("yyyy-MM-dd HH:mm:ss");  **try** {  Date time = df.parse(df.format(**new** Date()));  //parse()返回的是一个Date类型数据，format返回的是一个StringBuffer类型的数据 new Date()获取的是现在的时间  **return** **new** Timestamp(time.getTime());  //Date.getTime()的到的是1970年01月01日8点中以来的毫秒数  } **catch** (ParseException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  **return** **null**;  }  } |

1. 导入学生资源

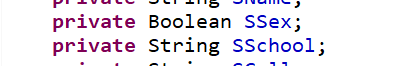
|  |
| --- |
| **public** String importExcleStudent() **throws** IOException {  createTime = *getTime*();//获取操作的时间  String path = "/excleFile";  (在文件夹里新建了个exleFile的临时文件)    //获取excleFile的绝对路径  String realpath = ServletActionContext.*getServletContext*().getRealPath(path);  System.*out*.println(realpath); //D:\myseclipseworkplace\.metadata\apache-tomcat-8.0.29\webapps\SDSYw\excleFile    //realpath是父路径 importstudentfilename是文件名 也就在realpath下新建了个文件importStudentFIleName文件  File target = **new** File(realpath, importStudentFileName);  System.*out*.println(target+" "+ importStudentFileName);//17建筑学院学生信息.xls  //D:\myseclipseworkplace\.metadata\apache-tomcat-8.0.29\webapps\SDSYw\excleFile\17建筑学院学生信息.xls  //向路径realpath写入日志  FileWriter writer =createFileWirter(realpath+"/importExcleStudent.log");  System.*out*.println(realpath+"/importExcleStudent.log");  //D:\myseclipseworkplace\.metadata\apache-tomcat-8.0.29\webapps\SDSYw\excleFile/importExcleStudent.log  （可以看到这两步骤的结果）  2345截图20180803212853  **if** (target.exists()) {  target.delete();  }    **try** {  //创建文件字节读取流对象时，必须明确与之关联的数据源。  FileInputStream fi = **new** FileInputStream(target);    Workbook rwb = Workbook.*getWorkbook*(fi);////打开表1  // Workbook bookResource = Workbook.getWorkbook(new File("resource.xls"));//打开表1  Sheet rs = rwb.getSheet(0);//获得第一个工作表对象  **int** rows = rs.getRows();// 所有行  **int** clos = rs.getColumns();// 所有列  // System.out.println(rows+","+clos);  Boolean sex = **false**;  String stuId, stuName = **null**, stuSex = **null**, stuSchool = **null**, stuCollege = **null**, stuProfession = **null**,  stuClass = **null**, stuClassType = **null**, stuJobstatus = **null**, stuTutor = **null**, stuPhone = **null**,  stuIdcard = **null**;  StudentsId stuI = **null**;  Students student = **null**;  Students stu = **null**;  **int** number=1,j=0;  **for** (**int** i = 1; i < rows; i++) {  //System.out.println(i+",,"+rows);  j = 0;  stuId = rs.getCell(j++, i).getContents().trim();//取出第j i位置的内容  stuName = rs.getCell(j++, i).getContents();  stuSex = rs.getCell(j++, i).getContents();  stuSchool = rs.getCell(j++, i).getContents();  stuCollege = rs.getCell(j++, i).getContents();  stuProfession = rs.getCell(j++, i).getContents();  stuClass = rs.getCell(j++, i).getContents();  stuClassType = rs.getCell(j++, i).getContents();  stuJobstatus = rs.getCell(j++, i).getContents();  stuTutor = rs.getCell(j++, i).getContents();  stuPhone = rs.getCell(j++, i).getContents();  stuIdcard = rs.getCell(j++, i).getContents();  **if** (stuSex.equals("男")) {  sex = **false**;  } **else** **if** (stuSex.equals("女")) {  sex = **true**;  }  stuI = **new** StudentsId();  stuI.setSStudentId(stuId);  student = **null**;  student = studentDao.isexist(stuId);  **if** (student != **null**) {// 表中存在该学生，则更新  stuI.setSId(student.getId().getSId());  stu = **new** Students(stuI, stuName, sex, stuSchool, stuCollege, stuProfession, stuClass,  stuClassType, stuJobstatus, stuTutor, stuPhone, stuIdcard,  MD5Util.*MD5*(stuId.substring(stuId.length() - 6)));  // System.out.println(stuId + "--" + stuId.length() + ">>" + stuId.substring(stuId.length() - 6));  stu.setSCreateTime(student.getSCreateTime());  stu.setSChangeTime(*getTime*());  studentDao.updatestudent(stu);  writer.write("第"+(number++)+"记录"+stuId+"更新成功\r\n");  } **else** {// 表中不存在该学生，则添加  System.*out*.println(stuId + "--" + stuId.length() + ">>" + stuId.substring(stuId.length() - 6));  stu = **new** Students(stuI, stuName, sex, stuSchool, stuCollege, stuProfession, stuClass,  stuClassType, stuJobstatus, stuTutor, stuPhone, stuIdcard,  MD5Util.*MD5*(stuId.length()>6?stuId.substring(stuId.length() - 6):("1234567"+stuId).substring(("1234567"+stuId).length() - 6)));  stu.setSCreateTime(createTime);  stu.setSChangeTime(createTime);  stu.setSPicture("https://sdsy.zzjc.edu.cn/SDSYw/image/1.jpg");  studentDao.addstudent(stu);  writer.write("第"+(number++)+"记录"+stuId+"添加成功\r\n");  }  writer.flush();  }  writer.close();  fi.close();  }  **catch** (FileNotFoundException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  } **catch** (BiffException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  } **catch** (IOException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  }**catch**(Exception e){  System.*out*.println(e);  }**finally**{    }  s = JSONTools.*createJsonObject*("s", "OK");  **return** "success";  } |
| Finished |

注意：在修改

Students的属性类型的时候 它对应的students.hbn.xml对应的属性也要修改



例子：

把ssex的属性修改成了boolaen然后对应的.hbm.xml也要修改



# 添加老师信息

1. Jsp和学导入学生的一样，就是form提交的action改成了*importExcleTeacher.action*
2. 在struts.xml天啊及对应的action

<action name=*"importExcleTeacher"* method=*"importExcleTeacher"* class=*"excle"*>

<result type=*"json"* name=*"success"*>

<param name=*"root"*>s</param>

</result>

</action>

1. 添加importEcelTeacher方法

|  |
| --- |
| // 老师  **private** File importTeacher;  **private** String importTeacherFileName;  **private** TeacherDao teacherDao;  **public** String importExcleTeacher() **throws** IOException {  createTime = getTime();  System.*out*.println("导入教师");  String path = "/excleFile";  String realpath = ServletActionContext.*getServletContext*().getRealPath(path);  FileWriter writer =createFileWirter(realpath+"/importExcleTeacher.log");  File target = **new** File(realpath, importTeacherFileName);  **if** (target.exists()) {  target.delete();  }  **try** {  FileUtils.*copyFile*(importTeacher, target);  System.*out*.println("教师信息文件上传成功");  } **catch** (IOException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  }  **try** {  FileInputStream fi = **new** FileInputStream(target);  Workbook wb = Workbook.*getWorkbook*(fi);  Sheet rs = wb.getSheet(0);  **int** rows = rs.getRows();// 行  **int** clos = rs.getColumns();// 列  System.*out*.println(rows + "and" + clos);  String tName = **null**, tId = **null**, tAddress = **null**, tTelePhone = **null**, tEmail = **null**, tCollege = **null**,  tPhone = **null**, tType = **null**, tdesc = **null**, tp = **null**, themeall;  TeacherId teaId = **null**;  Teacher tea = **null**;  Teacher teacher = **null**;  **int** number=1;  **for** (**int** i = 1; i < rows; i++) {  **for** (**int** j = 0; j < clos; j += 255) {  tName = rs.getCell(j++, i).getContents();  tId = rs.getCell(j++, i).getContents();  tAddress = rs.getCell(j++, i).getContents();  tTelePhone = rs.getCell(j++, i).getContents();  tEmail = rs.getCell(j++, i).getContents();  tCollege = rs.getCell(j++, i).getContents();  tPhone = rs.getCell(j++, i).getContents();  tType = rs.getCell(j++, i).getContents();  tdesc = rs.getCell(j++, i).getContents();  tp = rs.getCell(j++, i).getContents();  String[] t = tType.split(",|，");  themeall = "";  tId = tId.trim();  **for** (**int** k = 0; k < t.length; k++) {  **switch** (t[k]) {  **case** "学习":  themeall += "1,";  **break**;  **case** "就业":  themeall += "2,";  **break**;  **case** "创业":  themeall += "3,";  **break**;  **case** "情感":  themeall += "4,";  **break**;  **case** "心理":  themeall += "5,";  **break**;  **case** "生活":  themeall += "6,";  **break**;  **case** "其他":  themeall += "7,";  **break**;  **default**:  **break**;  }  teaId = **new** TeacherId();  teaId.setTTeacherId(tId);  tea = **null**;// 可以不添加，只是为保证tea判断正确  tea = teacherDao.isExist(tId);  **if** (tea != **null**) {  // System.out.println("修改");  // System.out.println(tea.getId().getTTeacherId());  teaId.setTId(tea.getId().getTId());  teacher = **new** Teacher(teaId, MD5Util.*MD5*(tId), tName, tPhone, Integer.*parseInt*(tTelePhone),  tEmail, tCollege, tAddress, themeall, tp, tdesc);  teacher.setTFrequency(tea.getTFrequency());  teacher.setTSearchnum(tea.getTSearchnum());  teacher.setTPower(tea.getTPower());  teacher.setTCreateTime(tea.getTCreateTime());  teacher.setTType(tea.getTType());  teacher.setTChangeTime(getTime());  teacherDao.updatepsd(teacher);  writer.write("第"+(number++)+"记录"+tId+"更新成功\r\n");  } **else** {  teacher = **new** Teacher(teaId, MD5Util.*MD5*(tId), tName, tPhone, Integer.*parseInt*(tTelePhone),  tEmail, tCollege, tAddress, themeall, tp, tdesc);  teacher.setTFrequency(0);  teacher.setTSearchnum(0);  // teacher.setTPicture("https://sdsy.zzjc.edu.cn/SDSYw/image/1.jpg");  teacher.setTPower(1);  teacher.setTType(2);  teacher.setTCreateTime(createTime);  teacher.setTChangeTime(createTime);  teacherDao.addTeacher(teacher);  writer.write("第"+(number++)+"记录"+tId+"添加成功\r\n");  }  writer.flush();  }  }  }  writer.close();  fi.close();  } **catch** (FileNotFoundException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  } **catch** (BiffException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  } **catch** (IOException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  }  s = JSONTools.*createJsonObject*("s", "OK");  **return** "success";  }  注意 file和string定义两个一定要按这种格式才行 |

1. 给teacherdao添加相应的接口：

|  |
| --- |
| **public** Teacher isExist(String tId);  **public** **void** updatepsd(Teacher teacher);  **public** **void** addTeacher(Teacher teacher); |

1. 实现这些接口

|  |
| --- |
| @Override  **public** Teacher isExist(String tId) {  List<Teacher> list=**this**.getHibernateTemplate().find("from Teacher s where s.id.TTeacherId=?",tId);  **if** (list.size()>0) {  **return** list.get(0);  }**else**{  **return** **null**;  }  }  @Override  **public** **void** updatepsd(Teacher teacher) {  **this**.getHibernateTemplate().update(teacher);    }  @Override  **public** **void** addTeacher(Teacher teacher) {  // **TODO** Auto-generated method stub  **this**.getHibernateTemplate().save(teacher);  } |

1. 在application添加响应的事件

<bean id=*"excle"* class=*"web.action.ExcleAction"*>

<property name=*"studentDao"* ref=*"studentDaoImp"*></property>

<property name=*"teacherDao"* ref=*"teacherDaoImp"*></property>

</bean>