# 尚德书院登录界面的制作

目录

**[1. 数据库信息的录入](#_Toc2536_WPSOffice_Level1)** **[1](#_Toc2536_WPSOffice_Level1)**

[（1） 新建数据库 teacher并导入数据](#_Toc28278_WPSOffice_Level2) [1](#_Toc28278_WPSOffice_Level2)

**[2. 导入web工程，并反向工程](#_Toc12909_WPSOffice_Level1)** **[1](#_Toc12909_WPSOffice_Level1)**

**[3. 新建对teacher表实现的操作接口 TeacherDao.java](#_Toc11922_WPSOffice_Level1)** **[2](#_Toc11922_WPSOffice_Level1)**

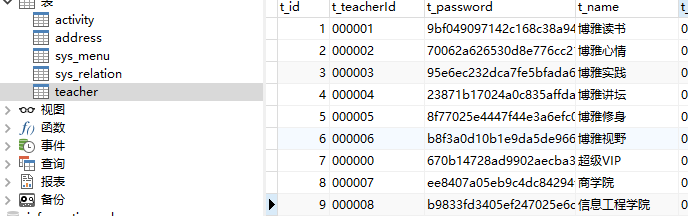
**[4. TeacherDaoImp.java接口功能的实现](#_Toc19257_WPSOffice_Level1)** **[2](#_Toc19257_WPSOffice_Level1)**

**[5. 建立login.jsp](#_Toc18034_WPSOffice_Level1)** **[2](#_Toc18034_WPSOffice_Level1)**

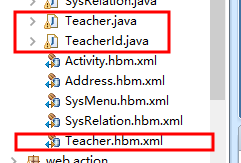
**[6. 在给login.jsp的提交按钮添加AJAX的方法](#_Toc5104_WPSOffice_Level1)** **[3](#_Toc5104_WPSOffice_Level1)**

**[登录后：](#_Toc5609_WPSOffice_Level1)** **[5](#_Toc5609_WPSOffice_Level1)**

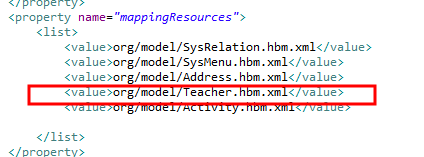
1. 数据库信息的录入
2. 新建数据库 teacher并导入数据

（部分数据）

1. 导入web工程，并反向工程
2. 反向工程后得到如下两张表

（原因：因为teacher是双主键）

1. 反向完后还有个关键的操作，要在applicationContext.xml里面添加它的映射



1. 新建对teacher表实现的操作接口 TeacherDao.java

|  |
| --- |
| **public** **interface** TeacherDao {  **public** Teacher login(String name,String psd);  } |

1. TeacherDaoImp.java接口功能的实现

|  |
| --- |
| **public** **class** TeacherDaoImp **extends** HibernateDaoSupport **implements** TeacherDao {  @SuppressWarnings("unchecked")  **public** Teacher login(String name, String psd) {  String[] parm = {name, psd};  System.*out*.println(name+""+psd);  List<Teacher> l = getHibernateTemplate()  .find("from Teacher t where t.id.TTeacherId = ? and t.TPassword = ?",  parm);  **if** (l.size() > 0) {  **return** l.get(0);  } **else**  **return** **null**;  }  } |

1. 建立login.jsp

|  |
| --- |
| 写入表单：  <form method=*"Get"* id=*"loginform"* class=*"form-vertical"* action=*"login.action"*>  <div class=*"control-group"*>  <label class=*"control-label"*>用户名</label>  <div class=*"controls"*>  <div class=*"main\_input\_box"*>  <span class=*"add-on bg\_lg"*><i class=*"icon-user"*  style="font-size:*16px*;"></i></span><input name=*"name"* type=*"text"* />  </div>  </div>  </div>  <div class=*"control-group2"*>  <label class=*"control-label"*>登陆密码</label>  <div class=*"controls"*>  <div class=*"main\_input\_box"*>  <span class=*"add-on bg\_ly"*><i class=*"icon-lock"*  style="font-size:*16px*;"></i></span><input name=*"psdMd5"* type=*"password"* />  </div>  </div>  </div>  <div class=*"control-group2"* style="margin: *0px 0px 0px 21px*">  &nbsp;&nbsp;&nbsp;&nbsp;  <div id=*"div\_id\_embed"*></div>  </div>  </form> |

1. 因为表单提交的地方时：login.action则要在struts.xml里面建立联系

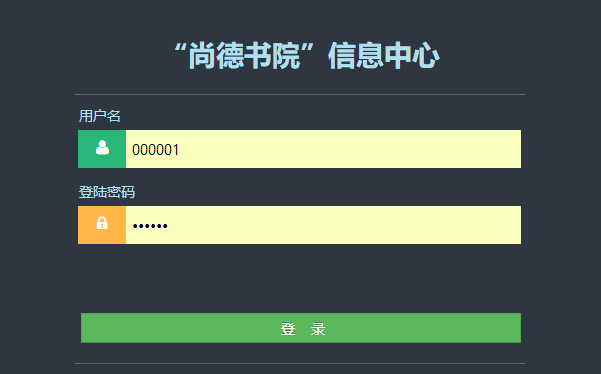
|  |
| --- |
| <action name=*"login"* method=*"login"* class=*"webAction"*>  <result type=*"json"* name=*"success"*>  <param name=*"root"*>s</param>  </result>  </action>  这里要注意的是：在Webaction.java里面传递值得时候，这里的json一定要写清楚 |

1. 表单里 用户名的input的name属性是name ，密码的input属性是：psdMd5，所以在WedAction.xml要定义相应的变量值

|  |
| --- |
| 并生成get set方法 |

1. 在给login.jsp的提交按钮添加AJAX的方法

|  |  |
| --- | --- |
| <input type=*"button"* id=*"loginBtn"*class=*"btn btn-success"*style="width:*440px*;"value=*" 登&nbsp;&nbsp;&nbsp;&nbsp;录"* />  <script type=*"text/javascript"*>  window.onload = **function**() {  $('#loginBtn').click(**function**() {  $.ajax({  type: "get",  url: "login.action",  dataType: "json",  data: $('#loginform').serialize(),  success: **function**(data) {  **if**(data.result == 0) {  window.location.href = 'index.jsp'  } **else** {  alert("用户名或密码错误");  }  },  error: **function**() {  alert("失败!");  }});});};  </script>  点完提交的时候开始执行WebAction的login方法   |  | | --- | | **public** String login() **throws** Exception {  System.*out*.println("开始登陆" );  System.*out*.println(name+" " +psdMd5);  Teacher t = teacherDao.login(name, MD5Util.*MD5*(psdMd5));  **if** (t == **null**) {  // System.out.println("空");  result = 1;  } **else** {  result = 0;  // System.out.println("登陆" + name + ":" + psdMd5);  Integer p = 1;  session.put("user", t);  session.put("power", t.getTPower());  session.put("loginname", t.getTName());  session.put("loginid", t.getId().getTTeacherId());  session.put("id", t.getId().getTId());//这个id时领导表示列的id刚好是18  session.put("actpag", p);// 用于活动分页  session.put("itempag", p);// 用于活动类别分页1  session.put("recordpag", p);// 学生成绩分页  session.put("addresspag", p);// 地址分页2  session.put("announcepag", p);// 公告分页3  }  s = JSONTools.*createJsonObject*("result", result);  **return** "success";  }  //这个“result”与login.jsp里的要一样 | |

效果:

登录后：

# 活动界面

目录

[1.分析](#_Toc30002_WPSOffice_Level1) [1](#_Toc30002_WPSOffice_Level1)

[2. 数据库的建立](#_Toc18426_WPSOffice_Level1) [1](#_Toc18426_WPSOffice_Level1)

[3. 对数据库方向工程](#_Toc2054_WPSOffice_Level1) [1](#_Toc2054_WPSOffice_Level1)

[4. 新建接口 ActivityDao.java](#_Toc4634_WPSOffice_Level1) [2](#_Toc4634_WPSOffice_Level1)

[5. 新建一个实现接口的类ActivityDaoImp.java：](#_Toc21346_WPSOffice_Level1) [2](#_Toc21346_WPSOffice_Level1)

[6. 在WebAction.java实现方法 allActivity()](#_Toc30321_WPSOffice_Level1) [4](#_Toc30321_WPSOffice_Level1)

[7. 实现：](#_Toc2108_WPSOffice_Level1) [4](#_Toc2108_WPSOffice_Level1)

[8. jsp页面的实现](#_Toc4838_WPSOffice_Level1) [5](#_Toc4838_WPSOffice_Level1)

[9. Jsp总代码](#_Toc24571_WPSOffice_Level1) [6](#_Toc24571_WPSOffice_Level1)

[10. 删除功能的实现](#_Toc27122_WPSOffice_Level1) [9](#_Toc27122_WPSOffice_Level1)

[11. 活动详情的实现](#_Toc3868_WPSOffice_Level1) [11](#_Toc3868_WPSOffice_Level1)

1.分析

|  |
| --- |
| （1）活动界面：activitilist用来获取所有的数据，所以我要先在数据库先查找。取出所有数据。    （2）一些需要权限的操作 sessionScope.power    这里的 session.power 的用法和sessionScope.power效果一样。  这个power是在登录的时候就已经赋值了 |

1. 数据库的建立

|  |
| --- |
| （1）新建数据库 activity     1. 新建数据库activity\_type用来放活动属性 |

1. 对数据库方向工程



然后重要的一步骤 在applicationContext.xml建立联系：



1. 新建接口 ActivityDao.java

|  |
| --- |
| **//写一个查找所有活动的方法**  **public** **interface** ActivityDao {  **public** List<Activity> allActivity(**int** i, **int** j);//用来查找所有活动  **public** **double** pageSum(**int** i);//用来查看总数  } |

1. 新建一个实现接口的类ActivityDaoImp.java：

|  |
| --- |
| **//用来查看活动总量**  **public** **double** pageSum(**int** i) {  Session session = getSession();  Query query;  **if** (i == 0) {  query = session  .createQuery("select count(\*) from Activity where r\_status!=-1");  } **else** {  query = session  .createQuery("select count(\*) from Activity where r\_status!=-1 and r\_type=?");  query.setParameter(0, ++i);  }  Long b = (Long) query.uniqueResult();  **double** a = b;  session.close();  **return** a;  }  // 查询所有活动  **public** List<Activity> allActivity(**int** i, **int** k) {  List<Teacher> list = **null**;  List<ActivityType> list2 = **null**;  List<Address> list3 = **null**;  Session ssn = getSession();  Query query;  **if** (k == 0) {  query = ssn  .createQuery("from Activity where RStatus!=-1 order by RId desc");  } **else** {  query = ssn  .createQuery("from Activity where RStatus!=-1 and RType=? order by RId desc");  query.setParameter(0, ++k);  }  // System.out.println(i);  query.setFirstResult(i);  query.setMaxResults(7);  List<Activity> l = query.list();  **int** len = l.size();  **if** (len != 0) {  String[] values1 = **new** String[len];  Object[] values2 = **new** Object[len];  Object[] values3 = **new** Object[len];  StringBuilder sb = **new** StringBuilder();  StringBuilder sb2 = **new** StringBuilder();  StringBuilder sb3 = **new** StringBuilder();  // System.out.println(len+","+values1.length);  **for** (**int** j = 0; j < len; j++) {  values1[j] = l.get(j).getRTId();  values2[j] = l.get(j).getRType();  values3[j] = l.get(j).getRAddress();  **if** (j != len - 1) {  sb.append("?,");  sb2.append("?,");  sb3.append("?,");  } **else** {  sb.append("?");  sb2.append("?");  sb3.append("?");  }  }  // 这里的list查询出来的应该包含TeacherId吧  list = getHibernateTemplate().find(  "FROM Teacher t WHERE t.id.TTeacherId IN (" + sb.toString()  + ")", values1);  list2 = getHibernateTemplate().find(  "FROM ActivityType t WHERE t.aid IN (" + sb2.toString()  + ")", values2);  list3 = getHibernateTemplate().find(  "FROM Address t WHERE t.id IN (" + sb3.toString() + ")",  values3);  ssn.close();  **for** (Activity a : l) {  **for** (Teacher entity : list)  **if** (a.getRTId().equals(entity.getId().getTTeacherId())) {  a.setTeacher(entity);  **break**;  }  **for** (ActivityType entity1 : list2)  **if** (a.getRType() == entity1.getAid()) {  a.setActtype(entity1);  **break**;  }  **for** (Address entity2 : list3)  **if** (a.getRAddress().equals(entity2.getId())) {  a.setAddress(entity2);  **break**;  }}}**return** l; } |

1. 在WebAction.java实现方法 allActivity()



写入方法allActivity()

|  |
| --- |
| **public** String allActivity() **throws** Exception {  } |

1. 实现：

|  |
| --- |
| 1. 定义变量   ActivityDao activityDao;  **private** List<Activity> activitylist =**new** ArrayList<Activity>();//所有活动信息  并生成get set方法 |
| 2.//后台数据管理  **public** String allActivity() **throws** Exception {  String loginid = (String) session.get("loginid");  **int** i = 0;  **switch** (loginid) {  //用来判断登录方是谁  **case** "000001":  i = 1;  **break**;  **case** "000002":  i = 2;  **break**;  **case** "000003":  i = 3;  **break**;  **case** "000004":  i = 4;  **break**;  **case** "000005":  i = 5;  **break**;  **case** "000006":  i = 6;  **break**;  }  **double** x = activityDao.pageSum(i);  **if** (x != 0) {  Integer pa = (Integer) session.get("actpag");  pageSum = (**int**) (activityDao.pageSum(i) / 7 + 0.99);  **if** (pageSum == 0)  pageSum = 1;  **if** (n == 0)  pa = 1;  **else** **if** (n == 1 && pa > 0)  pa = pa - 1;  **else** **if** (n == 2 && pa < pageSum)  pa = pa + 1;  **else** **if** (n == 3 && pa < pageSum)  pa = pageSum;  n = 4;// 保持原位置  page = pa;  activitylist = activityDao.allActivity((pa - 1) \* 7, i);  session.put("actpag", pa);  } **else** {  activitylist = **null**;  }  **return** "success";  } |

1. jsp页面的实现

|  |
| --- |
| 1. 作用：<link href=*"jules/css/style.min862f.css?v=4.1.0"* rel=*"stylesheet"*>   有：  没有：   1. 作用：让页面切换变得柔和   <link href=*"jules/css/animate.min.css"* rel=*"stylesheet"*>   1. 作用：改变了样式   <link href=*"jules/css/bootstrap.min14ed.css?v=3.3.6"* rel=*"stylesheet"*>  没有：  有： |

1. Jsp总代码

|  |
| --- |
| <%@page contentType=*"text/html; charset=UTF-8"*%>  <%@taglib uri=*"/struts-tags"* prefix=*"s"*%>  <%@ taglib uri=*"http://java.sun.com/jsp/jstl/core"* prefix=*"c"*%>  <%@ page isELIgnored=*"false"*%>  <%@taglib uri=*"http://java.sun.com/jsp/jstl/fmt"* prefix=*"fmt"*%>  <!DOCTYPE html>  <html>  <head>  <meta charset=*"utf-8"*>  <meta name=*"viewport"* content=*"width=device-width, initial-scale=1.0"*>  <!-- 后台主页 -->  <title>活动列表</title>  <link href=*"jules/css/bootstrap.min14ed.css?v=3.3.6"* rel=*"stylesheet"*>  <link href=*"jules/css/animate.min.css"* rel=*"stylesheet"*>  <link href=*"jules/css/style.min862f.css?v=4.1.0"* rel=*"stylesheet"*>  </head>  <body class=*"gray-bg"*>  <div class=*"wrapper wrapper-content animated fadeInRight"*>  <div class=*"row"*><div class=*"col-sm-12"*><div class=*"ibox"*>  <div class=*"ibox-content"*><h2>所有信息列表</h2><div class=*"clients-list"*>  <div class=*"tab-content"*><div id=*"tab-1"* class=*"tab-pane active"*>  <div class=*"full-height-scroll"*><div class=*"table-responsive"*>  <table class=*"table table-striped table-hover"*>  <tbody>  <c:forEach items=*"*${activitylist}*"* var=*"res"*>  <tr style="height:*80px*">  <td><img src=*"*${res.RPicture}*"* width=*"60px;"*  height=*"60px;"* /></td>  <td style="width: *160px*">活动号： <br /> <small>  ${res.RId} </small>  </td>  <td style="width: *160px*">活动名称： <br /> <small>  ${res.RName} </small>  </td>  <td class=*"project-title"*>活动负责人： <br /> <small>  ${res.teacher.TName}&nbsp; </small>  </td>  <td class=*"project-title"*>活动地址： <br /> <small>  ${res.address.name}&nbsp; </small>  </td>  <td class=*"project-title"* ><small >举办日期：</small> <br />  <fmt:formatDate value=*"*${res.RStartTime}*"*  pattern=*"yyyy-MM-dd HH:mm"* />&nbsp;  </small>  </td>  <td>人数：${res.RMaximum}&nbsp;</td>  <td>活动类型： <c:choose>  <c:when test=*"*${res.RType == 2 }*"*>  <span>博雅读书</span>  </c:when>  <c:when test=*"*${res.RType == 3 }*"*>  <span>博雅"心晴"</span>  </c:when>  <c:when test=*"*${res.RType == 4 }*"*>  <span>博雅实践</span>  </c:when>  <c:when test=*"*${res.RType == 5 }*"*>  <span>博雅讲坛</span>  </c:when>  <c:when test=*"*${res.RType == 6 }*"*>  <span>博雅修身</span>  </c:when>  <c:when test=*"*${res.RType == 7 }*"*>  <span>博雅视野</span>  </c:when>  </c:choose>  </td>  </tr>  </c:forEach>  </tbody>  </table>  <div style="text-align: *center*;">  <s:if test=*"page!=1"*>  <a href=*"allActivity.action?n=0"*>首页</a>  &nbsp;  <a href=*"allActivity.action?n=1"*>上一页</a>  </s:if>  <s:if test=*"page!=PageSum"*>  &nbsp;  <a href=*"allActivity.action?n=2"*>下一页</a>  &nbsp;  <a href=*"allActivity.action?n=3"*>尾页</a>  </s:if>  &nbsp;第${page}页/共${PageSum}页  </div></div></div></div></div></div></div></div></div></div></div>  </body>  </html> |

效果：



1. 删除功能的实现

|  |
| --- |
| 1.<td style="text-align:*center*;width: *50px*;">  <c:if test=*"*${sessionScope.power == 0}*"*>  <button onClick=*"delActivity(*${res.RId}*)"* type=*"button"* class=*"btn btn-danger btn-xs"*>  <span>删除</span>  </button>  </c:if>  </td>   1. 添加onClick（）方法   <script type=*"text/javascript"*>  // 删除  **function** delActivity(id) {  **if** (confirm("确认删除?")) {  **var** url = "deleteActivity.action?id=" + id;  window.location.href = url;  }  }  </script>   1. 在struts.xml添加deleteActivity.action   <action name=*"deleteActivity"* class=*"webAction"* method=*"deleteActivity"*>  <result name=*"success"* type=*"chain"*>allActivity</result>  <result name=*"faile"*>faile.jsp</result>  </action>   1. 在ActivityDao新建接口deleteActivity(Activity activity)和查找指定活动的接口findoneActivity（）; 2. 在ActivityDaoImp实现就口   // 删除活动  **public** **void** deleteActivity(Activity activity) {  **this**.getHibernateTemplate().update(activity);  }  // 查询一个活动详情  **public** Activity findoneActivity(**int** id) {  List<Activity> list = getHibernateTemplate().find(  "from Activity a where a.RId=?", id);  **if** (list.size() > 0) {  List<Teacher> t = getHibernateTemplate().find(  "from Teacher t where id.TTeacherId=?",  list.get(0).getRTId());  **if**(t.size()>0)  list.get(0).setTeacher(t.get(0));  **return** list.get(0);  }  **return** **null**;  }   1. 在WebActiob添加deleteActivity方法   **public** **void** updateActTime(Activity detailactivity) {  // **TODO** Auto-generated method stub  **this**.getHibernateTemplate().update(detailactivity);  }   1. 在WebAction.java添加deleteActivity（）实现方法   // 删除活动 将活动状态改变  **public** String deleteActivity() **throws** Exception {  Activity activity = activityDao.findoneActivity(id);  **if** (activity != **null**) {  activity.setRStatus(-1);  activityDao.deleteActivity(activity);  **return** "success";  } **else**  **return** "faile";  } |

效果：

1. 活动详情的实现

|  |
| --- |
| 1.<td style="text-align:*center*;width: *50px*;">  <button onClick=*"detailActivity(*${res.RId}*)"*  type=*"button"* class=*"btn btn-primary btn-xs"*>  <span>详情</span>  </button>  </td>   1. 添加onclick（）方法   // 详情  **function** detailActivity(id) {  **var** url = "oneActivity.action?id=" + id;  window.location.href = url;  }   1. 在structs.xml添加oneActivity.action   <action name=*"oneActivity"* class=*"webAction"* method=*"oneActivity"*>  <result name=*"success"* >/article.jsp</result>  </action>   1. 新建article.jsp   <%@page contentType=*"text/html; charset=UTF-8"*%>  <%@taglib uri=*"/struts-tags"* prefix=*"s"*%>  <%@ taglib uri=*"http://java.sun.com/jsp/jstl/core"* prefix=*"c"*%>  <%@ page isELIgnored=*"false"*%>  <!DOCTYPE html>  <html>  <head>  <meta charset=*"utf-8"*>  <meta name=*"viewport"* content=*"width=device-width, initial-scale=1.0"*>  <title>活动详情</title>  <link href=*"jules/css/bootstrap.min14ed.css?v=3.3.6"* rel=*"stylesheet"*>  <link href=*"jules/css/font-awesome.min93e3.css?v=4.4.0"* rel=*"stylesheet"*>  <link href=*"jules/css/animate.min.css"* rel=*"stylesheet"*>  <link href=*"jules/css/style.min862f.css?v=4.1.0"* rel=*"stylesheet"*>  <script type=*"text/javascript"* src=*"jules/js/jquery.min.js"*></script>  <script src=*"jules/js/qrcode.min.js"*></script><!-- 二維碼顯示 -->    <script type=*"text/javascript"*>  $(document).ready(**function**() {  **var** act\_id = ${  detailactivity.RId  };  **var** json = '{"act\_id":' + act\_id + '}';  console.log("act\_id=" + json);  **var** qrcode = **new** QRCode('qrcode\_2', {  text: json, //放入你想传的json串  width: 256,  height: 256,  colorDark: '#000000',  colorLight: '#ffffff',  correctLevel: QRCode.CorrectLevel.H  });  });  </script>  </head>  <body class=*"gray-bg"*>  <div id=*"showLargeImage"* style="display:*none*;z-index:*9999*;width:*500px*;height:*400px*;position:*absolute*;left:*50%*;">  <img id=*"ShowImageBox"* src=*"http://img.52fuqing.com/upload/news/20160125/201601251216541264.jpg"* width=*"500px;"* height=*"380px;"* />  </div>  <div class=*"wrapper wrapper-content animated fadeInRight article"*>  <div class=*"row"*>  <div class=*"col-lg-10 col-lg-offset-1"*>  <div class=*"ibox"*>  <div class=*"ibox-content"*>  <div class=*"text-center "*>  <h1>${detailactivity.RName}</h1>  <div class=*"ibox float-e-margins"*>  <div class=*"ibox-title"*>  <h5>申请活动详情</h5>  </div>  <div class=*"ibox-content"*>  <div class=*"form-group"* style="height: *95px*">  <label class=*"col-sm-2"* style="a">活动简介</label>  <textarea disabled=*"disabled"* rows=*"4"* type=*"text"* name=*"feature"* class=*"form-control"* style="width:*450px*">${detailactivity.RIntroduction}</textarea>  </div>  <div class=*"form-group"* style="height: *95px*">  <label class=*"col-sm-2"* style="a">第一次活动时间</label>  ${detailactivity.RStartTime}&nbsp;>>&nbsp;${detailactivity.REndTime}  </div><div class=*"form-group"* style="height: *95px*">  <label class=*"col-sm-2"* style="a">第二次活动时间</label>  ${detailactivity.RStartTimeTwo}&nbsp;>>&nbsp;${detailactivity.REndTimeTwo}  </div>  <div class=*"col-sm-10"* >  <div class=*"form-group"* style="height: *80px*">  <label class=*"col-sm-2 control-label"* style="width:*50px*;height:*50px*">活动二维码</label>  </div>  <div class=*"col-sm-10"* id=*"qrcode\_2"*><button onclick="print()">打印二维码</button></div>  </div>  </div>  <div class=*"ibox float-e-margins"*>  <div class=*"ibox-title"*>  <h5>导师详情</h5>  </div>  <div class=*"ibox-content"*>  <div class=*"form-group"* style="height: *95px*;">  <label class=*"col-sm-2"* style="a">导师姓名</label>  <span>${detailactivity.teacher.TName}</span>  </div>  <div class=*"hr-line-dashed"*></div>  <div class=*"form-group"* style="height: *95px*">  <label class=*"col-sm-2 control-label"*>所属分院</label>  <span>${detailactivity.teacher.TCollege}</span>  </div>  <div class=*"hr-line-dashed"*></div>  <div class=*"form-group"* style="height: *80px*;">  <label class=*"col-sm-2 control-label"*>导师简介</label>  <textarea disabled=*"disabled"* rows=*"4"* type=*"text"* name=*"feature"* class=*"form-control"* style="width:*450px*">${detailactivity.teacher.TIntroduction}</textarea> </div></div></div></div></div></div></div></div></div></div>  </body>  </html>   1. 新建detailactivity变量，并生成get set 方法 2. 添加oneActivity方法   // 活动详情  **public** String oneActivity() **throws** Exception {  detailactivity = activityDao.findoneActivity(id);  **if** (detailactivity != **null**) {  **return** "success";  } **else**  **return** "faile";  } |

效果