

图1：员工列表

图1的jsp页面代码web\teacher\employeeList.jsp如下：

在页面上生成链接：

<http://localhost:8080/employeeAction5.action?employeeId=1>

然后根据struts+spring文件的指向跳转。

<%@ taglib prefix="s" uri="/struts-tags"%>

<s:iterator value="#session.employees1" var="list">  
<tr align="center">  
 <td><s:property value="#list.employeeId"/></td>  
 …

<td><a href="employeeAction5.action?employeeId=

<s:property value="#list.employeeId"/>">编辑</a></td>

</tr>  
</s:iterator>

struts文件config\struts\struts-employee.xml部分代码如下：

<?xml version="1.0" encoding="UTF-8" ?>  
<struts>  
 <package name="employee" extends="struts-global" >

点击编辑按钮之后，首先执行EmployeeListAction.java中的函数editEmployeeByID()，成功之后，打开/teacher/employeeEdit.jsp页面

http://localhost:8080/employeeAction5.action?employeeId=1

<!--分页展示员工信息的Action-->  
 <action name="employeeAction2" class="employees" >

<result name="success">/teacher/employeeList.jsp</result>

</action>  
 <!--点击编辑之后跳转到编辑员工信息页面的Action-->  
 <action name="employeeAction5"

class="employees" method="editEmployeeById" >  
 <result name="success">/teacher/employeeEdit.jsp</result>  
 </action>

<!--编辑之后处理提交的表单的Action-->

<action name="employeeAction6"

class="employees2" method="editEmployee" >

<result name="success" type="redirectAction">employeeAction2</result>

<result name="input">/teacher/employeeEdit.jsp</result>

</action>  
 </package>  
</struts>

spring文件/config/spring/applicationContext-employee.xml部分内容如下：

<?xml version="1.0" encoding="UTF-8"?>  
<beans …>  
 <bean id="employees" class="com.panda.action.EmployeeListAction" scope="prototype"/>  
 <bean id="employees2" class="com.panda.action.EditEmployeeAction" scope="prototype"/>  
 </beans>

EmployeeListAction.java部分代码如下：

public String editEmployeeById() throws Exception {

System.out.println("到达编辑的Action,编辑员工1：editEmployeeById()");

if (employeeId == 0) {

throw new Exception("员工ID不能为0");

}

EmployeesEntity employeesEntity = employeeService.queryEmployeeByEmployeeId(employeeId);

if (employeesEntity.getEmployeesByManagerId() != null) {

employeesEntity.setManagerId(employeesEntity.getEmployeesByManagerId().getEmployeeId());

}

employeesEntity.setDepartmentId(employeesEntity.getDepartmentsByDepartmentId().getDepartmentId());

HttpServletRequest request = (HttpServletRequest) actionContext.get(ServletActionContext.HTTP\_REQUEST);

request.getSession().setAttribute("employee2",employeesEntity);

List<EmployeesEntity> lists = (List<EmployeesEntity>)request.getSession().getAttribute("employees");

List<String> ListEmployeesID = new ArrayList<String>();

ListEmployeesID.add("");//允许员工没有上司

for(EmployeesEntity data : lists) {

//员工的上司不能是

if (data.getEmployeeId()!=employeesEntity.getEmployeeId())

新增SESSION：ListEmployeesID，被employeeEdit.jsp引用。

ListEmployeesID.add(String.valueOf(data.getEmployeeId()));

}

request.getSession().setAttribute("ListEmployeesID",ListEmployeesID);

return SUCCESS ;

}

EmployeeListAction.editEmployeeById()的作用是为即将打开的员工编辑页面准备SESSION变量"employee2"，employee2的值是编辑之前的员工信息employeesEntity，上面的EmployeeListAction.editEmployeeById()执行过后，返回SUCCESS，就打开图2。

图2的JSP页面代码是：web\teacher\employeeEdit.jsp，其中员工上司ID下拉框的代码如下：

EmployeeListAction.java中的

request.getSession().setAttribute("ListEmployeesID",ListEmployeesID);

<tr>  
<td>员工上司ID</td>  
 <td>  
 <select name="employeesEntitys.managerId">  
 <s:iterator value="#session.ListEmployeesID" id="**list**">  
 <s:if test="#session.employee2.managerId==**#list**">  
 <option selected><s:property value="#session.employee2.managerId"/> </option>  
 </s:if>  
 <s:else>  
 <option><s:property value="#list"/></option>  
 </s:else>  
 </s:iterator>  
 </select>  
 </td>  
 </tr>

填充下拉框，并设置默认选项

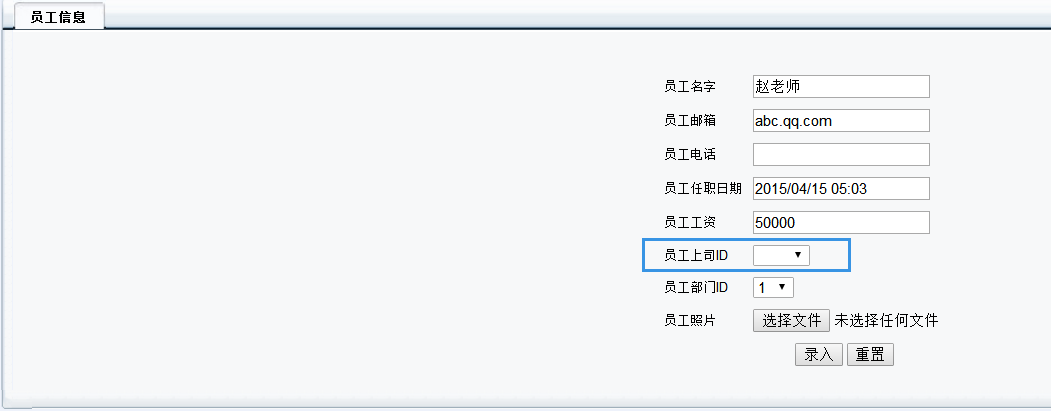


图2：修改员工信息页面