使用hibernate和struts2实现分页功能 - xiaoluo501395377 - 博客园

想着每天能学个新东西，今天准备了这个hibernate+struts2实现页面的分页功能，在这里，首先要感谢 http://www.blogjava.net/rongxh7/archive/2008/11/29/243456.html 这位博友发表的博文，通过对他代码的阅读，从而自己实现了网页分页功能，以下是源代码，如大家有需要，可以回复我，我会放出源代码供大家下载！

1.DAO层接口的设计，定义一个PersonDAO接口，里面声明了两个方法：

```
publicinterface PersonDAO{    public List queryByPage(String hql, int offset, int pageSize);        publicint
getAllRowCount(String hql);}
```

2.DAO层接口的实现类PersonDAOImpl类，将其两个方法实现出来：

复制代码

```
publicclass PersonDAOImpl implements PersonDAO{    /**     * 通过hql语句得到数据库中记录总数     */    @Override
publicint getAllRowCount(String hql)    {        Session session = HibernateUtil.openSession();        Transaction tx =
null;        int allRows = 0;        try        {            tx = session.beginTransaction();
Query query = session.createQuery(hql);                        allRows = query.list().size();
tx.commit();                    }        catch (Exception e)        {            if(tx != null)            {
tx.rollback();            }                        e.printStackTrace();        }        finally        {
HibernateUtil.closeSession(session);        }                return allRows;    }    /**     * 使用hibernate提供的分页功
能，得到分页显示的数据     */    @SuppressWarnings("unchecked")    @Override    public List queryByPage(String hql, int
offset, int pageSize)    {        Session session = HibernateUtil.openSession();        Transaction tx = null;
List list = null;                try        {            tx = session.beginTransaction();                        Query
query = session.createQuery(hql).setFirstResult(offset).setMaxResults(pageSize);                        list =
query.list();                        tx.commit();                    }        catch (Exception e)        {
if(tx != null)            {                tx.rollback();            }                        e.printStackTrace();
}        finally        {            HibernateUtil.closeSession(session);        }                return list;
}}
```

复制代码

3.定义了一个PageBean(每一页所需要的内容都存放在这个PageBean里面)，里面用来存放网页每一页显示的内容：

复制代码

```
publicclass PageBean{    private List list; //通过hql从数据库分页查询出来的list集合privateint allRows; //总记录数
privateint totalPage; //总页数privateint currentPage; //当前页public List getList()    {        return list;    }
publicvoid setList(List list)    {        this.list = list;    }    publicint getAllRows()    {        return allRows;
}    publicvoid setAllRows(int allRows)    {        this.allRows = allRows;    }    publicint getTotalPage()    {
return totalPage;    }    publicvoid setTotalPage(int totalPage)    {        this.totalPage = totalPage;    }
publicint getCurrentPage()    {        return currentPage;    }    publicvoid setCurrentPage(int currentPage)    {
this.currentPage = currentPage;    }        /**     * 得到总页数     * @param pageSize 每页记录数     * @param allRows
总记录数     * @return 总页数     */publicint getTotalPages(int pageSize, int allRows)    {        int totalPage =
(allRows % pageSize == 0)? (allRows / pageSize): (allRows / pageSize) + 1;                return totalPage;    }
/**     * 得到当前开始记录号     * @param pageSize 每页记录数     * @param currentPage 当前页     * @return*/publicint
getCurrentPageOffset(int pageSize, int currentPage)    {        int offset = pageSize * (currentPage - 1);
return offset;    }        /**     * 得到当前页， 如果为0 则开始第一页，否则为当前页     * @param page     *
@return*/publicint getCurPage(int page)    {        int currentPage = (page == 0)? 1: page;                return
currentPage;    }    }
```

复制代码

4.Service层接口设计，定义一个PersonService接口，里面声明了一个方法，返回一个PageBean：

```
publicinterface PersonService{    public PageBean getPageBean(int pageSize, int page);}
```

5.Service层接口实现类PersonServiceImpl类，实现唯一的方法：

复制代码

```
publicclass PersonServiceImpl implements PersonService{    private PersonDAO personDAO = new PersonDAOImpl();        /**
* pageSize为每页显示的记录数     * page为当前显示的网页     */    @Override    public PageBean getPageBean(int pageSize,
int page)    {        PageBean pageBean = new PageBean();               String hql = "from Person";               int
allRows = personDAO.getAllRowCount(hql);              int totalPage = pageBean.getTotalPages(pageSize, allRows);
int currentPage = pageBean.getCurPage(page);              int offset = pageBean.getCurrentPageOffset(pageSize,
currentPage);              List list = personDAO.queryByPage(hql, offset, pageSize);
pageBean.setList(list);        pageBean.setAllRows(allRows);        pageBean.setCurrentPage(currentPage);
pageBean.setTotalPage(totalPage);              return pageBean;    }}
```

复制代码

6.Action层设计，定义一个PersonAction：

复制代码

```
publicclass PersonAction extends ActionSupport{    private PersonService personService = new PersonServiceImpl();
privateint page;        publicint getPage()    {        return page;    }    publicvoid setPage(int page)    {
this.page = page;    }    @Override    public String execute() throws Exception    {        //表示每页显示5条记录，page
表示当前网页        PageBean pageBean = personService.getPageBean(5, page);              HttpServletRequest request =
ServletActionContext.getRequest();              request.setAttribute("pageBean", pageBean);              return
SUCCESS;    }}
```

复制代码

7.辅助类设计，HibernateUtil:

复制代码

```
publicclass HibernateUtil{    privatestatic SessionFactory sessionFactory;        static    {        sessionFactory =
new Configuration().configure().buildSessionFactory();    }        publicstatic Session openSession()    {
Session session = sessionFactory.openSession();              return session;    }        publicstaticvoid
closeSession(Session session)    {        if(session != null)        {            session.close();        }    }    }
```

复制代码

8.最后也就是分页页面显示pagePerson.jsp：

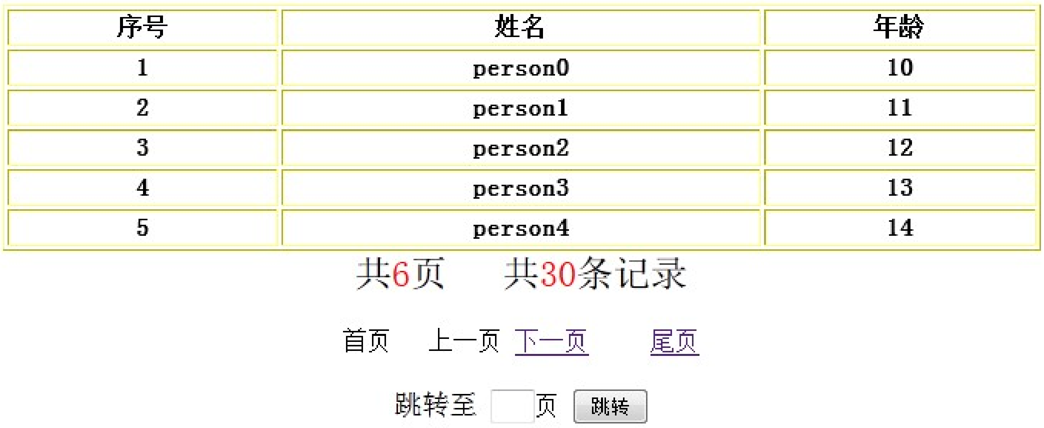
复制代码

```
<%@ page language="java" import="java.util.*" pageEncoding="UTF-8"%><%@ taglib prefix="s" uri="/struts-tags"%><html>
<head><base href="<%=basePath%>"><title>My JSP 'pagePerson.jsp' starting pagetitle><meta http-equiv="pragma"
content="no-cache"><meta http-equiv="cache-control" content="no-cache"><meta http-equiv="expires" content="0"><meta
http-equiv="keywords" content="keyword1,keyword2,keyword3"><meta http-equiv="description" content="This is my page">
<script type="text/javascript">function validate()        {            var page = document.getElementsByName("page")
[0].value;                          if(page ><s:property value="#request.pageBean.totalPage"/>)            {
alert("你输入的页数大于最大页数，页面将跳转到首页！");                              window.document.location.href
="personAction";                              returnfalse;            }                      returntrue;        }
script>head><body><h1><font color="blue">分页查询font>h1><hr><table border="1" align="center" bordercolor="yellow"
width="50%"><tr><th>序号th><th>姓名th><th>年龄th>tr><s:iterator value="#request.pageBean.list" id="person"><tr><th>
<s:property value="#person.id"/>th><th><s:property value="#person.name"/>th><th><s:property
value="#person.age"/>th>tr>s:iterator>table><center><font size="5">共<font color="red"><s:property
value="#request.pageBean.totalPage"/>font>页 font>  <font size="5">共<font color="red"><s:property
value="#request.pageBean.allRows"/>font>条记录font><br><br><s:if test="#request.pageBean.currentPage == 1">
首页    上一页        s:if><s:else><a href="personAction.action">首页a>   <a href="personAction.action?page=/>">上一页
a>s:else><s:if test="#request.pageBean.currentPage != #request.pageBean.totalPage"><a href="personAction.action?
page=/>">下一页a>   <a href="personAction.action?page=/>">尾页a>s:if><s:else>            下一页    尾页
s:else>center><br><center><form action="personAction" onsubmit="return validate();"><font size="4">跳转至font><input
type="text" size="2" name="page">页            <input type="submit" value="跳转">form>center>body>html>
```

复制代码

至此，hibernate+struts2实现网页分页功能代码部分就完毕了，像hibernate与struts的配置文件就不列出来了，那些都不是重点！

页面效果如下:



序号	姓名	年龄
1	person0	10
2	person1	11
3	person2	12
4	person3	13
5	person4	14

共6页　　共30条记录

首页　上一页 下一页　　尾页

跳转至　页