实验 11 用户信息管理系统中的分页查询功能实现

胡祝华　2020.06.11

一、实验目的

1. 掌握删除选中、分页查询、复杂条件查询等复杂功能的实现方法。
2. 掌握添加功能模块时，三层架构不同层次的修改顺序和思路。
3. 进一步巩固 java web 开发中的相关技术。

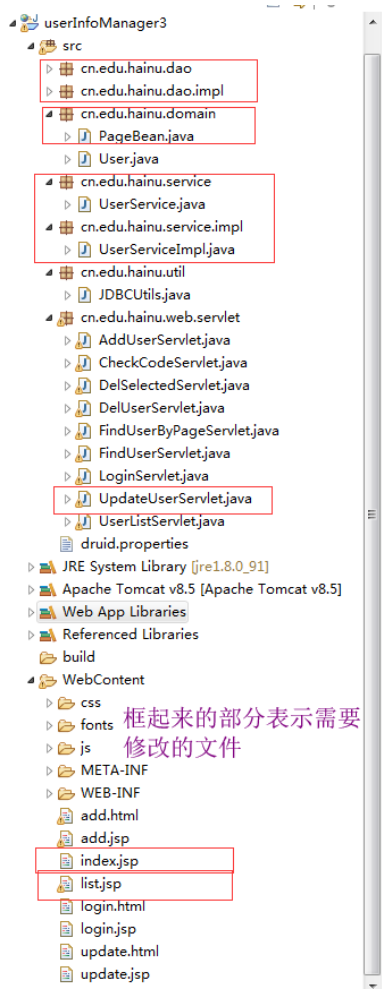
二、实验环境要求

1. java EE eclipse Version: 2018-09 (4.9.0)
2. JDK 1.8.0_91
3. TOMCAT 8.5
4. 相关依赖的 jar 包。

三、实验步骤

说明:实验 6（查）、实验 8（增）、实验 9（删、改）、实验 10（删除选中）完成了简单的增、删、改、查、删除选中的操作。

1. 本次实验在实验 10 的基础上进行功能扩展和升级，实现用户列表页面的分页查询功能。与实验10 有差异的代码都会高亮显示出来。 本次实验完成后的项目结构：



2. 分页查询的好处：

1）减轻服务器内存的开销；

2）提升用户体验。

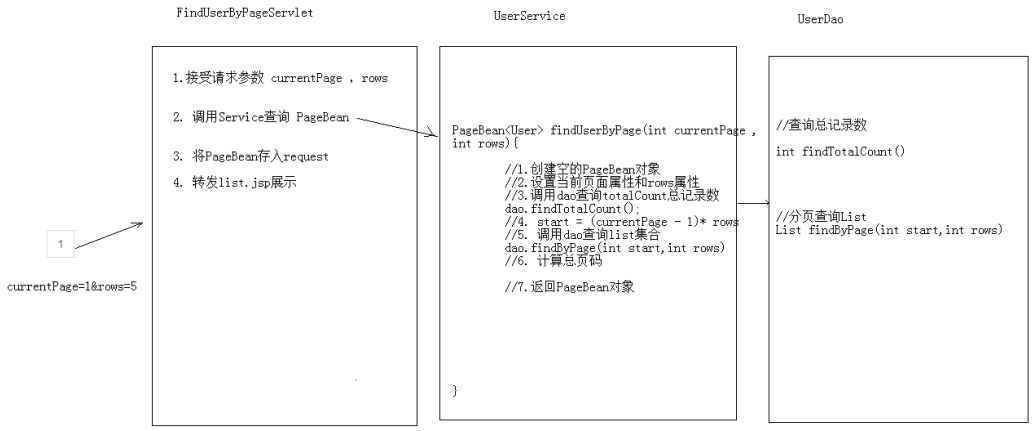
3. 分页功能的实现：

3.1 分析、详细设计流程如下：

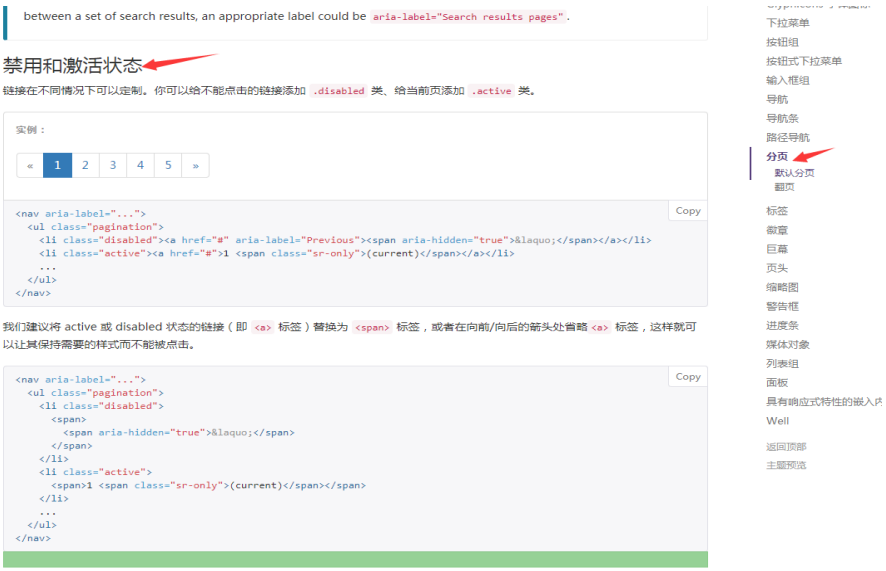
分析过程：

因为 list 页面到底需要分成多少页，需要通过 service 层和 dao 层查询数据库 user 表才能知道，因此，需要一个 servlet 来处理该需求，这个 servlet 负责获取一个承载了这些参数信息的 pagebean 对象，并将该对象存入到 requet 共享作用域中，list.jsp 再获取这些参数数据。进一步的详细设计如下图。

设计流程：



3.2 ***分析：***查看百度的分页的功能，发现选中页面和样式和其他未选中的样式不一样。这个功能我们可以在 bootstrap 的主页中去找到解决方案。https://v3.bootcss.com/components/#pagination ，在页面中找到主件→分页。给出的方案如下：



3.3 第一步，根据上面的分析过程，我们发现在实体域 domain 中需要建立一个 javabean，即 PageBean 的类，用来承载与分页相关的一些参数。***注意：***因为不仅仅用户列表需要分页，其他的列表可能也需要分页，因此，这里我们定义每页的显示数据时，给 PageBean 和 List 添加一个自定义的泛型 T，**保证 PageBean 组件的通用性**。当我们需要真正创建列表对象时，则将知道的列表数据的类型传给 T 即可。对应的包路径为：cn.edu.hainu.domain。Bean 的代码如下：

```
package cn.edu.hainu.domain;

import java.util.List;

/**
 * 分页对象
 */
public class PageBean<T> {
    private int totalCount; // 总记录数
    private int totalPage ; // 总页码
    private List<T> list ; // 每页的数据
    private int currentPage ; //当前页码
    private int rows;//每页显示的记录数

    public int getTotalCount() {
        return totalCount;
    }

    public void setTotalCount(int totalCount) {
        this.totalCount = totalCount;
    }

    public int getTotalPage() {
        return totalPage;
    }

    public void setTotalPage(int totalPage) {
        this.totalPage = totalPage;
    }

    public List<T> getList() {
        return list;
    }

    public void setList(List<T> list) {
```

```
        this.list = list;
    }

    public int getCurrentPage() {
        return currentPage;
    }

    public void setCurrentPage(int currentPage) {
        this.currentPage = currentPage;
    }

    public int getRows() {
        return rows;
    }

    public void setRows(int rows) {
        this.rows = rows;
    }

    @Override
    public String toString() {
        return "PageBean{" +
                "totalCount=" + totalCount +
                ", totalPage=" + totalPage +
                ", list=" + list +
                ", currentPage=" + currentPage +
                ", rows=" + rows +
                '}';
    }
}
```

3.4 第二步，开始实现真正的逻辑代码的编写工作。根据 3.1 中的设计流程图发现我们需要在 web 层、service 层和 dao 层都需要修改代码。这一步需要把设计流程图设计出来，并且理清楚。

3.5 第三步，修改 web 层的代码。按照设计流程的详细设计，首先在 cn.edu.hainu.web.servlet 包中创建一个 FindUserByPageServlet 的 servlet。这个 servlet 创建了之后，原来的那个用于显示用户列表的 servlet（UserListServlet）就可以不用了。

由于 service 层和 dao 层的代码还没有编写。因此，在这个新建的 servlet 中会出现错误提示。我们先不用理会。等我们编写完成 service 层和 dao 层的代码后错误就会自动消失。**这样的一个编写代码的逻辑是**：根据详细设计流程，从 web 层向 dao 层逐步完善代码，通过 web 层的设计实现去驱动 service 层和 dao 层代码的编写。因为有可能详细设计流程考虑的不是非常全面，需要在模块测试中进行迭代完善。

FindUserByPageServlet.java:

```
package cn.edu.hainu.web.servlet;

import cn.edu.hainu.domain.PageBean;
import cn.edu.hainu.domain.User;
import cn.edu.hainu.service.UserService;
import cn.edu.hainu.service.impl.UserServiceImpl;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.util.Map;

@WebServlet("/findUserByPageServlet")
```

```java
public class FindUserByPageServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        request.setCharacterEncoding("utf-8");

        //1.获取参数
        String currentPage = request.getParameter("currentPage");//当前页码
        String rows = request.getParameter("rows");//每页显示条数

        //健壮性检查，即如果当前页面为空或null时，设定为1
        if(currentPage == null || "".equals(currentPage)){
            currentPage = "1";
        }

        //健壮性检查，即如果页面显示的行数为空或null时，设定为5
        if(rows == null || "".equals(rows)){
            rows = "5";
        }

        //2.调用service查询，通过调用service层的方法获得PageBean对象。该对象在jsp中要用。
        UserService service = new UserServiceImpl();
        //指定泛型T的类型为User。
        PageBean<User> pb = service.findUserByPage(currentPage,rows);

        System.out.println(pb);

        //3.将PageBean作为共享数据存入request作用域中
        request.setAttribute("pb",pb);

        //4.转发到list.jsp
        request.getRequestDispatcher("/list.jsp").forward(request,response);
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        this.doPost(request, response);
    }
}
```

3.6 第四步，根据详细设计流程修改 service 层的接口和实现类代码。

扩充接口代码 UserService.java

```java
package cn.edu.hainu.service;

import java.util.List;

import cn.edu.hainu.domain.PageBean;
import cn.edu.hainu.domain.User;

/**
 * 用户管理的业务接口
 */
public interface UserService {

    /**
     * 查询所有用户信息
     * @return
     */
    public List<User> findAll();

    /**
     * 登录方法
     * @param user
     * @return
     */
```

```
    User login(User user);

    /**
     * 保存User
     * @param user
     */
    void addUser(User user);

    /**
     * 根据id删除User
     * @param id
     */
    void deleteUser(String id);

    /**
     * 根据id查询
     * @param id
     * @return
     */
    User findUserById(String id);

    /**
     * 修改用户信息
     * @param user
     */
    void updateUser(User user);

    /**
     * 批量删除用户
     * @param ids
     */
    void delSelectedUser(String[] ids);

    /**
     * 分页查询
     * @param currentPage
     * @param rows
     * @return
     */
    PageBean<User> findUserByPage(String currentPage, String rows);
}
```

根据详细设计扩充实现类 UserServiceImpl.java，注意代码中的详细注释说明。

```
package cn.edu.hainu.service.impl;

import java.util.List;

import cn.edu.hainu.dao.UserDao;
import cn.edu.hainu.dao.impl.UserDaoImpl;
import cn.edu.hainu.domain.PageBean;
import cn.edu.hainu.domain.User;
import cn.edu.hainu.service.UserService;

public class UserServiceImpl implements UserService {
    private UserDao dao = new UserDaoImpl();

    @Override
    public List<User> findAll() {
        //调用Dao完成查询
        return dao.findAll();
    }

    @Override
    /**
     * 调用dao层的代码实现查询。
```

```
     */
    public User login(User user) {
        return
dao.findUserByUsernameAndPassword(user.getUsername(),user.getPassword());
    }

    @Override
    public void addUser(User user) {
        dao.add(user);
    }

     @Override
     public void deleteUser(String id)
     {
          dao.delete(Integer.parseInt(id));
     }

    @Override
    public User findUserById(String id) {
        return dao.findById(Integer.parseInt(id));
    }

    @Override
    public void updateUser(User user) {
        dao.update(user);
    }

    @Override
    public void delSelectedUser(String[] ids) {
        if(ids != null && ids.length > 0){
            //1.遍历数组
            for (String id : ids) {
                //2.调用dao删除，复用了dao中的delete方法。
                dao.delete(Integer.parseInt(id));
            }
        }

    }

    @Override
    /**
     * 在service层的业务逻辑中填充pagebean的参数。并返回PageBean的对象。
     */
    public PageBean<User> findUserByPage(String _currentPage, String _rows) {

        int currentPage = Integer.parseInt(_currentPage); // 类型转换。
        int rows = Integer.parseInt(_rows); // 类型转换。

        if(currentPage <=0) {//健壮性检查
            currentPage = 1;
        }
        //1.创建空的PageBean对象
        PageBean<User> pb = new PageBean<User>();
        //2.设置参数
        pb.setCurrentPage(currentPage); //添加进pb对象中
        pb.setRows(rows); //添加进pb对象中

        //3.调用dao查询总记录数
        int totalCount = dao.findTotalCount();
        pb.setTotalCount(totalCount); //添加进pb对象中
        //4.调用dao查询List集合
        //计算开始的记录索引
        int start = (currentPage - 1) * rows;
        List<User> list = dao.findByPage(start,rows); //从dao从获取要展示的分页数据
        pb.setList(list); //添加进pb对象中
```

```
        //5.计算总页码，并设置进pb对象中
        int totalPage = (totalCount % rows)  == 0 ? (totalCount/rows) :
(totalCount/rows) + 1;
        pb.setTotalPage(totalPage); //添加进pb对象中

        return pb; //返回给web层
    }
}
```

3.7 上一步完成后，FindUserByPageServlet 中的错误会消失。因为还没有对 dao 层的代码进行编程，因此 service 层的代码会产生错误，先不理会。继续第五步，修改 dao 层的代码。根据详细设计分析，修改 UserDao.java。

```
package cn.edu.hainu.dao;

import java.util.List;

import cn.edu.hainu.domain.User;

/**
 * 用户操作的 DAO
 */
public interface UserDao {
    public List<User> findAll();
    User findUserByUsernameAndPassword(String username, String password);

    void add(User user);

    void delete(int id);

    User findById(int i);//用户回显信息。

    void update(User user);

    /**
     * 查询总记录数
     * @return
     * @param condition
     */
    int findTotalCount();

    /**
     * 分页查询每页记录
     * @param start
     * @param rows
     * @param condition
     * @return
     */
    List<User> findByPage(int start, int rows);
}
```

根据详细设计修改 UserDaoImpl.java:

```
package cn.edu.hainu.dao.impl;

import java.util.List;

import org.springframework.jdbc.core.BeanPropertyRowMapper;
import org.springframework.jdbc.core.JdbcTemplate;

import cn.edu.hainu.dao.UserDao;
import cn.edu.hainu.domain.User;
import cn.edu.hainu.util.JDBCUtils;
```

```
public class UserDaoImpl implements UserDao {

    private JdbcTemplate template = new JdbcTemplate(JDBCUtils.getDataSource());

    @Override
    public List<User> findAll() {
        //使用 JDBC 操作数据库...
        //1.定义 sql
        String sql = "select * from user";
        List<User> users = template.query(sql, new BeanPropertyRowMapper<User>(User.class));

        return users;
    }

    @Override
    /**
     * 利用 spring 框架中的 template，实现数据库的用户名和密码的查询，并将结果封装成 user 对象。
     */
    public User findUserByUsernameAndPassword(String username, String password) {
        try {
            String sql = "select * from user where username = ? and password = ?";
            User user = template.queryForObject(sql, new BeanPropertyRowMapper<User>(User.class),
username, password);
            return user;
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }

    }

    @Override
    public void add(User user) {
        //1.定义 sql
        String sql = "insert into user values(null,?,?,?,?,?,?,null,null)";
        //2.执行 sql
        template.update(sql, user.getName(), user.getGender(), user.getAge(), user.getAddress(), user.getQq(),
user.getEmail());
    }

    @Override
    public void delete(int id) {
        //1.定义 sql
        String sql = "delete from user where id = ?";
        //2.执行 sql
        template.update(sql, id);
    }

    @Override
    public User findById(int id) {//用于用户信息回显
        String sql = "select * from user where id = ?";
        return template.queryForObject(sql, new BeanPropertyRowMapper<User>(User.class), id);
    }

    @Override
    public void update(User user) {
        String sql = "update user set name = ?,gender = ? ,age = ? , address = ? , qq = ?, email = ? where id = ?";
        template.update(sql, user.getName(), user.getGender(), user.getAge(), user.getAddress(), user.getQq(),
user.getEmail(), user.getId());
    }

    @Override
    /**
     *从数据库获取总的记录数。
     */
    public int findTotalCount() {
```

```
        //1.定义模板初始化 sql
        String sql = "select count(*) from user";
        return template.queryForObject(sql,Integer.class);
    }

    @Override
    /**
     * 从数据库获得在list页面要展示的数据列表对象
     */
    public List<User> findByPage(int start, int rows) {
        String sql = "select * from user limit ?, ?";
        return template.query(sql,new BeanPropertyRowMapper<User>(User.class), start , rows);
    }
}
```

3.8 至此，后台代码阶段性的编写任务完成。项目也不会出错了。接下来进行阶段性的测试。看下代码是否有考虑不完善的问题。

1）首先我们检查 pagebean 中承载的参数是否正确。

启动 tomcat 服务器部署项目。直接访问 **FindUserByPageServlet，并给它传递两个参数。CurrentPage 和 rows。**

在 firefox 浏览器输入：

http://localhost:8080/userInfoManager3/findUserByPageServlet?currentPage=1&rows=5

因为我们在 servlet 中将 pb 对象打印在控制台，因此我们可以查看控制台输出。

```
//2.调用service查询，通过调用service层的方法获得PageBean对象。该对象在js
UserService service = new UserServiceImpl();
//指定泛型T的类型为User。
PageBean<User> pb = service.findUserByPage(currentPage,rows);

System.out.println(pb);
```

控制台的输出如下：

```
信息: {dataSource-1} inited
PageBean{totalCount=6, totalPage=2, list=[User{id=14, name='张三', gender='男', age=13, address='陕西'
```

我们从控制台拷贝出来显示如下：

```
PageBean{totalCount=6, totalPage=2, list=[User{id=14, name='张三', gender='男', age=13,
address='陕西', qq='12345', email='zhangsan@qq.com', username='null', password='null'},
User{id=13, name='李四', gender='女', age=15, address='北京', qq='88888',
email='ls@qq.com', username='null', password='null'}, User{id=12, name='张三', gender='
男', age=13, address='陕西', qq='12345', email='zhangsan@qq.com', username='null',
password='null'}, User{id=2, name='李四', gender='女', age=15, address='北京',
qq='88888', email='ls@qq.com', username='lisi', password='123'}, User{id=1, name='张
三', gender='男', age=13, address='陕西', qq='12345', email='zhangsan@qq.com',
username='zhangsan', password='123'}], currentPage=1, rows=5}
```

通过分析控制台输出，我们发现总的记录数，总的分页数，当前要展示的 5 条记录数，当前要展示的页码，及要展示的记录条数都获取和计算正确。

3.9 测试成功后，我们开始前台显示页面的编写工作。

因为分页是显示在 list.jsp 页面中，所以需要改造 list.jsp 页面。把相关的数据从 pagebean 中取出来。因为之前在 list 页面中是显示 user 表中的所有记录，改造后应该要显示 pb 中的 list 容器中获取的 5 条记录。同时修改总的记录数和页码，这些信息都在共享的 pb 对象中。修改代码高亮如下。

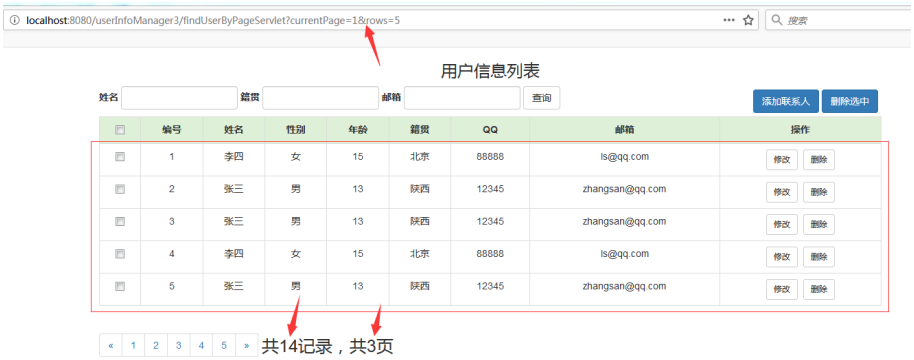
list.jsp:

```jsp
<%@ page contentType="text/html;charset=UTF-8" language="java"%>

<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<!DOCTYPE html>
<!-- 网页使用的语言 -->
<html lang="zh-CN">
<head>
<!-- 指定字符集 -->
<meta charset="utf-8">
<!-- 使用Edge最新的浏览器的渲染方式 -->
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<!-- viewport视口：网页可以根据设置的宽度自动进行适配，在浏览器的内部虚拟一个容器，容器的宽度与设备的宽度相同。
    width: 默认宽度与设备的宽度相同
    initial-scale: 初始的缩放比，为1:1 -->
<meta name="viewport" content="width=device-width, initial-scale=1">
<!-- 上述3个meta标签*必须*放在最前面，任何其他内容都*必须*跟随其后！ -->
<title>用户信息管理系统</title>

<!-- 1. 导入CSS的全局样式 -->
<link href="css/bootstrap.min.css" rel="stylesheet">
<!-- 2. jQuery导入，建议使用1.9以上的版本 -->
<script src="js/jquery-2.1.0.min.js"></script>
<!-- 3. 导入bootstrap的js文件 -->
<script src="js/bootstrap.min.js"></script>
<style type="text/css">
td, th {
    text-align: center;
}
</style>

<script>
    function deleteUser(id, name) {
        //用户安全提示
        if (confirm("您确定要删除" + name + "吗？")) {
            //访问路径
            location.href = "${pageContext.request.contextPath}/delUserServlet?id="
                    + id;
        }
    }

    window.onload = function() {//window.onload的功能是让页面加载完后再获取按钮id。
        //给删除选中按钮添加单击事件
        document.getElementById("delSelected").onclick = function() {
          var flag = false;
          //判断是否有选中条目
          var cbs = document.getElementsByName("uid");
          for (var i = 0; i < cbs.length; i++) {
              if(cbs[i].checked){
                  //有一个条目选中了
                  flag = true;
                  break;
              }
          }
          if(!flag)//说明没有条目被选中，则不执行任何操作，直接返回。
            return;

            if (confirm("您确定要删除选中条目吗？")) {
                    //表单提交
                    document.getElementById("form").submit();
            }
        }
```

```
        //1.获取第一个复选框cb
        document.getElementById("firstCb").onclick = function() {
            //2.获取下边列表中所有的cb
            var cbs = document.getElementsByName("uid");
            //3.遍历
            for (var i = 0; i < cbs.length; i++) {
                //4.设置这些cbs[i]的checked状态 = firstCb.checked
                cbs[i].checked = this.checked;
            }
        }
    }
</script>
</head>
<body>
    <div class="container">
        <h3 style="text-align: center">用户信息列表</h3>

        <div style="float: left;">

            <form class="form-inline">
                <div class="form-group">
                    <label for="exampleInputName2">姓名</label> <input type="text"
                        class="form-control" id="exampleInputName2">
                </div>
                <div class="form-group">
                    <label for="exampleInputName3">籍贯</label> <input type="text"
                        class="form-control" id="exampleInputName3">
                </div>

                <div class="form-group">
                    <label for="exampleInputEmail2">邮箱</label> <input type="email"
                        class="form-control" id="exampleInputEmail2">
                </div>
                <button type="submit" class="btn btn-default">查询</button>
            </form>

        </div>

        <div style="float: right; margin: 5px;">
            <a class="btn btn-primary"
                href="${pageContext.request.contextPath}/add.jsp">添加联系人</a> <a
                class="btn btn-primary" href="javascript:void(0);" id="delSelected">
删除选中</a>
        </div>

        <form id="form"
            action="${pageContext.request.contextPath}/delSelectedServlet"
            method="post">
            <table border="1" class="table table-bordered table-hover">
                <tr class="success">
                    <th><input type="checkbox" id="firstCb"></th>
                    <th>编号</th>
                    <th>姓名</th>
                    <th>性别</th>
                    <th>年龄</th>
                    <th>籍贯</th>
                    <th>QQ</th>
                    <th>邮箱</th>
                    <th>操作</th>
                </tr>

                <c:forEach items="${pb.list}" var="user" varStatus="s">
                    <tr>
```

```
                        <td><input type="checkbox" name="uid"
value="${user.id}"></td>
                        <td>${s.count}</td>
                        <td>${user.name}</td>
                        <td>${user.gender}</td>
                        <td>${user.age}</td>
                        <td>${user.address}</td>
                        <td>${user.qq}</td>
                        <td>${user.email}</td>
                        <td><a class="btn btn-default btn-sm"

    href="${pageContext.request.contextPath}/findUserServlet?id=${user.id}">修改
</a> 
                            <a class="btn btn-default btn-sm"

    href="javascript:deleteUser(${user.id},'${user.name}');">删除</a></td>
                    </tr>

                </c:forEach>
            </table>
        </form>

        <div>
            <nav aria-label="Page navigation">
                <ul class="pagination">
                    <li><a href="#" aria-label="Previous"> <span
                            aria-hidden="true">&laquo;</span>
                    </a></li>
                    <li><a href="#">1</a></li>
                    <li><a href="#">2</a></li>
                    <li><a href="#">3</a></li>
                    <li><a href="#">4</a></li>
                    <li><a href="#">5</a></li>
                    <li><a href="#" aria-label="Next"> <span
                            aria-hidden="true">&raquo;</span>
                    </a></li>
                    <span style="font-size: 25px; margin-left: 5px;">
                        共${pb.totalCount}记录，共${pb.totalPage}页
                    </span>
                </ul>
            </nav>
        </div>
    </div>
</body>
</html>
```

为了方便测试，往数据库中多添加几条记录。此时刷新刚才的 list.jsp 页面，则会出现从数据库找到的 5 条记录。因为总共 14 条记录，因此分成了 3 页。如下图：



3.10 下一步，把分页组件中假的分页页码改成真正的页码。把 list.jsp 中的分页组件代码中

的数据删除，采用 jstl 技术中的 foreach 进行遍历产生分页页码，此处采用 foreach 中的第一种遍历方式。同时迭代的次数、记录数据从 pb 对象中获取，编号的显示通过迭代的次数来标记。给定超链接转发的 servlet，当前页码和每页显示的记录条数。如下：

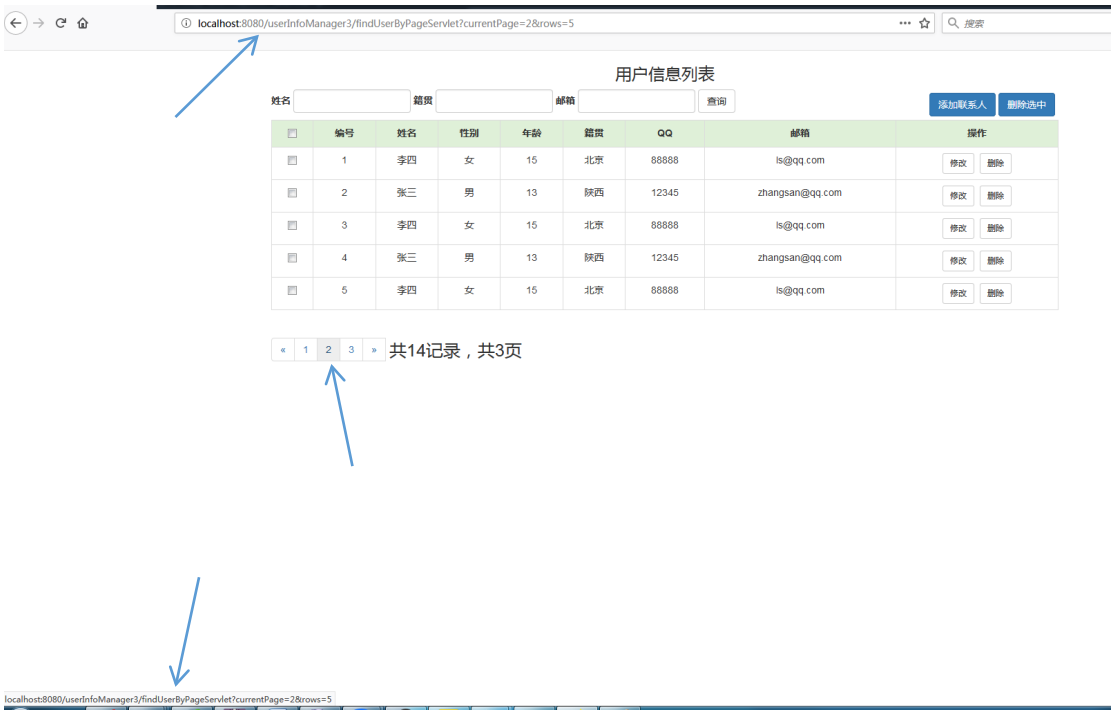
list.jsp

```
<%@ page contentType="text/html;charset=UTF-8" language="java"%>

<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<!DOCTYPE html>
<!-- 网页使用的语言 -->
<html lang="zh-CN">
<head>
<!-- 指定字符集 -->
<meta charset="utf-8">
<!-- 使用Edge最新的浏览器的渲染方式 -->
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<!-- viewport视口：网页可以根据设置的宽度自动进行适配，在浏览器的内部虚拟一个容器，容器的宽度与设备的宽度相同。
    width: 默认宽度与设备的宽度相同
    initial-scale: 初始的缩放比，为1:1 -->
<meta name="viewport" content="width=device-width, initial-scale=1">
<!-- 上述3个meta标签*必须*放在最前面，任何其他内容都*必须*跟随其后！ -->
<title>用户信息管理系统</title>

<!-- 1. 导入CSS的全局样式 -->
<link href="css/bootstrap.min.css" rel="stylesheet">
<!-- 2. jQuery导入，建议使用1.9以上的版本 -->
<script src="js/jquery-2.1.0.min.js"></script>
<!-- 3. 导入bootstrap的js文件 -->
<script src="js/bootstrap.min.js"></script>
<style type="text/css">
td, th {
    text-align: center;
}
</style>

<script>
    function deleteUser(id, name) {
        //用户安全提示
        if (confirm("您确定要删除" + name + "吗？")) {
            //访问路径
            location.href = "${pageContext.request.contextPath}/delUserServlet?id="
                    + id;
        }
    }

    window.onload = function() {//window.onload的功能是让页面加载完后再获取按钮id。
        //给删除选中按钮添加单击事件
        document.getElementById("delSelected").onclick = function() {
            var flag = false;
            //判断是否有选中条目
            var cbs = document.getElementsByName("uid");
            for (var i = 0; i < cbs.length; i++) {
                if (cbs[i].checked) {
                    //有一个条目选中了
                    flag = true;
                    break;
                }
            }
            if (!flag)//说明没有条目被选中，则不执行任何操作，直接返回。
                return;
```

```
            if (confirm("您确定要删除选中条目吗？")) {
                //表单提交
                document.getElementById("form").submit();
            }
        }

        //1.获取第一个复选框cb
        document.getElementById("firstCb").onclick = function() {
            //2.获取下边列表中所有的cb
            var cbs = document.getElementsByName("uid");
            //3.遍历
            for (var i = 0; i < cbs.length; i++) {
                //4.设置这些cbs[i]的checked状态 = firstCb.checked
                cbs[i].checked = this.checked;
            }
        }
    }
</script>
</head>
<body>
    <div class="container">
        <h3 style="text-align: center">用户信息列表</h3>

        <div style="float: left;">

            <form class="form-inline">
                <div class="form-group">
                    <label for="exampleInputName2">姓名</label> <input type="text"
                        class="form-control" id="exampleInputName2">
                </div>
                <div class="form-group">
                    <label for="exampleInputName3">籍贯</label> <input type="text"
                        class="form-control" id="exampleInputName3">
                </div>

                <div class="form-group">
                    <label for="exampleInputEmail2">邮箱</label> <input type="email"
                        class="form-control" id="exampleInputEmail2">
                </div>
                <button type="submit" class="btn btn-default">查询</button>
            </form>

        </div>

        <div style="float: right; margin: 5px;">
            <a class="btn btn-primary"
                href="${pageContext.request.contextPath}/add.jsp">添加联系人</a> <a
                class="btn btn-primary" href="javascript:void(0);" id="delSelected">
删除选中</a>
        </div>

        <form id="form"
            action="${pageContext.request.contextPath}/delSelectedServlet"
            method="post">
            <table border="1" class="table table-bordered table-hover">
                <tr class="success">
                    <th><input type="checkbox" id="firstCb"></th>
                    <th>编号</th>
                    <th>姓名</th>
                    <th>性别</th>
                    <th>年龄</th>
                    <th>籍贯</th>
                    <th>QQ</th>
                    <th>邮箱</th>
```

```
                    <th>操作</th>
                </tr>

                <c:forEach items="${pb.list}" var="user" varStatus="s">
                    <tr>
                        <td><input type="checkbox" name="uid"
value="${user.id}"></td>
                        <td>${s.count}</td>
                        <td>${user.name}</td>
                        <td>${user.gender}</td>
                        <td>${user.age}</td>
                        <td>${user.address}</td>
                        <td>${user.qq}</td>
                        <td>${user.email}</td>
                        <td><a class="btn btn-default btn-sm"

    href="${pageContext.request.contextPath}/findUserServlet?id=${user.id}">修改
</a> 
                            <a class="btn btn-default btn-sm"

    href="javascript:deleteUser(${user.id},'${user.name}');">删除</a></td>
                    </tr>

                </c:forEach>
            </table>
        </form>

        <div>
            <nav aria-label="Page navigation">
                <ul class="pagination">
                    <li><a href="#" aria-label="Previous"> <span
                            aria-hidden="true">&laquo;</span>
                    </a></li>

                    <c:forEach begin="1" end="${pb.totalPage}" var="i">
                        <li><a
    href="${pageContext.request.contextPath}/findUserByPageServlet?currentPage=${i}
&rows=5">${i}</a></li>
                    </c:forEach>

                    <li><a href="#" aria-label="Next"> <span
                            aria-hidden="true">&raquo;</span>
                    </a></li>
                    <span style="font-size: 25px; margin-left: 5px;">
                        共${pb.totalCount}记录，共${pb.totalPage}页
                    </span>
                </ul>
            </nav>
        </div>
    </div>
</body>
</html>
```

测试下，刷新 list.jsp 页面。如下图。



3.11 上面这步完成后，就可以分页显示用户记录信息了。但是还不够完善。我们继续完善代码。需要对一些特殊情况进行优化处理。另外，我们不能一直在浏览器地址栏中去请求 servlet 并给出请求参数。我们需要在 index.jsp 中点击超链接去获取用户列表。所以首先需要改造 index.jsp 页面。修改超链接的跳转 servlet 为 findUserByPageServlet。同时，不传递任何参数，因为在 *findUserByPageServlet* 中已经做了健壮性处理。

index.jsp

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html lang="zh-CN">
<head>
  <meta charset="utf-8"/>
  <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
  <meta name="viewport" content="width=device-width, initial-scale=1"/>
  <title>首页</title>

  <!-- 1. 导入CSS的全局样式 -->
  <link href="css/bootstrap.min.css" rel="stylesheet">
  <!-- 2. jQuery导入，建议使用1.9以上的版本 -->
  <script src="js/jquery-2.1.0.min.js"></script>
  <!-- 3. 导入bootstrap的js文件 -->
  <script src="js/bootstrap.min.js"></script>
  <script type="text/javascript">
  </script>
</head>
<body>
<div >${user.name},欢迎您</div> <!--测试用 hzh -->

<div align="center">
  <a
          href="${pageContext.request.contextPath}/findUserByPageServlet"
style="text-decoration:none;font-size:33px">查询所有用户信息
  </a>
</div>
</body>
</html>
```

3.12 分页页码选中后样式的改变处理。如 3.2 图中的显示，可以到 bootstrap 主页中查询对应的解决方案。操作：需要将当前选中的页码变成激活状态，而未选中的页码为不激活状态，因此需要作出判断。

另外，上一页和下一页按钮功能的实现。上一页：当前页码-1， 下一页：当前页码+1。因此需要继续改造 list.jsp 页面的分页模块代码。

list.jsp:

```
<%@ page contentType="text/html;charset=UTF-8" language="java"%>

<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<!DOCTYPE html>
<!-- 网页使用的语言 -->
<html lang="zh-CN">
<head>
<!-- 指定字符集 -->
<meta charset="utf-8">
<!-- 使用Edge最新的浏览器的渲染方式 -->
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<!-- viewport视口：网页可以根据设置的宽度自动进行适配，在浏览器的内部虚拟一个容器，容器的宽度与设备的宽度相同。
    width: 默认宽度与设备的宽度相同
    initial-scale: 初始的缩放比，为1:1 -->
<meta name="viewport" content="width=device-width, initial-scale=1">
<!-- 上述3个meta标签*必须*放在最前面，任何其他内容都*必须*跟随其后！ -->
<title>用户信息管理系统</title>

<!-- 1. 导入CSS的全局样式 -->
<link href="css/bootstrap.min.css" rel="stylesheet">
<!-- 2. jQuery导入，建议使用1.9以上的版本 -->
<script src="js/jquery-2.1.0.min.js"></script>
<!-- 3. 导入bootstrap的js文件 -->
<script src="js/bootstrap.min.js"></script>
<style type="text/css">
td, th {
    text-align: center;
}
</style>

<script>
    function deleteUser(id, name) {
        //用户安全提示
        if (confirm("您确定要删除" + name + "吗？")) {
            //访问路径
            location.href = "${pageContext.request.contextPath}/delUserServlet?id="
                    + id;
        }
    }

    window.onload = function() {//window.onload的功能是让页面加载完后再获取按钮id。
        //给删除选中按钮添加单击事件
        document.getElementById("delSelected").onclick = function() {
            var flag = false;
            //判断是否有选中条目
            var cbs = document.getElementsByName("uid");
            for (var i = 0; i < cbs.length; i++) {
                if (cbs[i].checked) {
                    //有一个条目选中了
                    flag = true;
                    break;
```

```
                }
            }
            if (!flag)//说明没有条目被选中，则不执行任何操作，直接返回。
                return;

            if (confirm("您确定要删除选中条目吗？")) {
                //表单提交
                document.getElementById("form").submit();
            }
        }

        //1.获取第一个复选框cb
        document.getElementById("firstCb").onclick = function() {
            //2.获取下边列表中所有的cb
            var cbs = document.getElementsByName("uid");
            //3.遍历
            for (var i = 0; i < cbs.length; i++) {
                //4.设置这些cbs[i]的checked状态 = firstCb.checked
                cbs[i].checked = this.checked;
            }
        }
    }
</script>
</head>
<body>
    <div class="container">
        <h3 style="text-align: center">用户信息列表</h3>

        <div style="float: left;">

            <form class="form-inline">
                <div class="form-group">
                    <label for="exampleInputName2">姓名</label> <input type="text"
                        class="form-control" id="exampleInputName2">
                </div>
                <div class="form-group">
                    <label for="exampleInputName3">籍贯</label> <input type="text"
                        class="form-control" id="exampleInputName3">
                </div>

                <div class="form-group">
                    <label for="exampleInputEmail2">邮箱</label> <input type="email"
                        class="form-control" id="exampleInputEmail2">
                </div>
                <button type="submit" class="btn btn-default">查询</button>
            </form>

        </div>

        <div style="float: right; margin: 5px;">
            <a class="btn btn-primary"
                href="${pageContext.request.contextPath}/add.jsp">添加联系人</a> <a
                class="btn btn-primary" href="javascript:void(0);" id="delSelected">
删除选中</a>
        </div>

        <form id="form"
            action="${pageContext.request.contextPath}/delSelectedServlet"
            method="post">
            <table border="1" class="table table-bordered table-hover">
                <tr class="success">
                    <th><input type="checkbox" id="firstCb"></th>
                    <th>编号</th>
                    <th>姓名</th>
```

```
                    <th>性别</th>
                    <th>年龄</th>
                    <th>籍贯</th>
                    <th>QQ</th>
                    <th>邮箱</th>
                    <th>操作</th>
                </tr>

                <c:forEach items="${pb.list}" var="user" varStatus="s">
                    <tr>
                        <td><input type="checkbox" name="uid"
value="${user.id}"></td>
                        <td>${s.count}</td>
                        <td>${user.name}</td>
                        <td>${user.gender}</td>
                        <td>${user.age}</td>
                        <td>${user.address}</td>
                        <td>${user.qq}</td>
                        <td>${user.email}</td>
                        <td><a class="btn btn-default btn-sm"

    href="${pageContext.request.contextPath}/findUserServlet?id=${user.id}">修改
</a> 
                            <a class="btn btn-default btn-sm"

    href="javascript:deleteUser(${user.id},'${user.name}');">删除</a></td>
                    </tr>

                </c:forEach>
            </table>
        </form>

        <div>
            <nav aria-label="Page navigation">
                <ul class="pagination">
                    <li><a
href="${pageContext.request.contextPath}/findUserByPageServlet?currentPage=${pb.cur
rentPage - 1}&rows=5" aria-label="Previous">  <!--上一页，修改href-->
                    <span aria-hidden="true">&laquo;</span>
                    </a></li>

                    <c:forEach begin="1" end="${pb.totalPage}" var="i">
                     <c:if test="${pb.currentPage == i}">   <!--选中的页码显示样式变
为激活状态-->
                        <li class="active"><a
href="${pageContext.request.contextPath}/findUserByPageServlet?currentPage=${i}&row
s=5">${i}</a></li>
                    </c:if>
                    <c:if test="${pb.currentPage != i}">  <!--未选中的页码-->
                        <li><a
href="${pageContext.request.contextPath}/findUserByPageServlet?currentPage=${i}&row
s=5">${i}</a></li>
                    </c:if>
                    </c:forEach>

                    <li><a
href="${pageContext.request.contextPath}/findUserByPageServlet?currentPage=${pb.cur
rentPage + 1}&rows=5" aria-label="Next"> <!--下一页，修改href-->
                    <span aria-hidden="true">&raquo;</span>
                    </a></li>
                    <span style="font-size: 25px; margin-left: 5px;">
                        共${pb.totalCount}记录，共${pb.totalPage}页
                    </span>
                </ul>
```

```
            </nav>
        </div>
    </div>
</body>
</html>
```

测试一下修改。

1） 刷新用户列表页面，发现激活的页码的状态已经发生了改变。

2） 按上一页和下一页也可以正常操作。

3.13 不过上面的操作会引发一个 bug，即当点击上一页到第 1 页的时候，如果再按上一页则还可以继续点击刷新页面，如果都最后一页后还可点击下一页按钮。现在要解决掉这个 bug。解决的思路：当前页面是第 1 页的时候，则将上一页的按钮隐藏。如果当前页面是最后一页的时候，则隐藏下一页按钮。继续改造 list.jsp。

list.jsp：

```
<%@ page contentType="text/html;charset=UTF-8" language="java"%>

<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<!DOCTYPE html>
<!-- 网页使用的语言 -->
<html lang="zh-CN">
<head>
<!-- 指定字符集 -->
<meta charset="utf-8">
<!-- 使用Edge最新的浏览器的渲染方式 -->
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<!-- viewport视口：网页可以根据设置的宽度自动进行适配，在浏览器的内部虚拟一个容器，容器的宽度与设备的宽度相同。
    width: 默认宽度与设备的宽度相同
    initial-scale: 初始的缩放比，为1:1 -->
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<!-- 上述3个meta标签*必须*放在最前面，任何其他内容都*必须*跟随其后！ -->
<title>用户信息管理系统</title>

<!-- 1. 导入CSS的全局样式 -->
<link href="css/bootstrap.min.css" rel="stylesheet">
<!-- 2. jQuery导入，建议使用1.9以上的版本 -->
<script src="js/jquery-2.1.0.min.js"></script>
<!-- 3. 导入bootstrap的js文件 -->
<script src="js/bootstrap.min.js"></script>
<style type="text/css">
td, th {
    text-align: center;
}
</style>

<script>
    function deleteUser(id, name) {
        //用户安全提示
        if (confirm("您确定要删除" + name + "吗？")) {
            //访问路径
            location.href = "${pageContext.request.contextPath}/delUserServlet?id="
                    + id;
        }
    }

    window.onload = function() {//window.onload的功能是让页面加载完后再获取按钮id。
        //给删除选中按钮添加单击事件
        document.getElementById("delSelected").onclick = function() {
            var flag = false;
            //判断是否有选中条目
            var cbs = document.getElementsByName("uid");
            for (var i = 0; i < cbs.length; i++) {
                if (cbs[i].checked) {
                    //有一个条目选中了
                    flag = true;
                    break;
                }
            }
            if (!flag)//说明没有条目被选中，则不执行任何操作，直接返回。
                return;

            if (confirm("您确定要删除选中条目吗？")) {
                //表单提交
                document.getElementById("form").submit();
            }
        }

        //1.获取第一个复选框cb
        document.getElementById("firstCb").onclick = function() {
            //2.获取下边列表中所有的cb
            var cbs = document.getElementsByName("uid");
            //3.遍历
            for (var i = 0; i < cbs.length; i++) {
                //4.设置这些cbs[i]的checked状态 = firstCb.checked
                cbs[i].checked = this.checked;
            }
        }
    }
</script>
</head>
<body>
    <div class="container">
        <h3 style="text-align: center">用户信息列表</h3>
```

```
        <div style="float: left;">

            <form class="form-inline">
                <div class="form-group">
                    <label for="exampleInputName2">姓名</label> <input type="text"
                        class="form-control" id="exampleInputName2">
                </div>
                <div class="form-group">
                    <label for="exampleInputName3">籍贯</label> <input type="text"
                        class="form-control" id="exampleInputName3">
                </div>

                <div class="form-group">
                    <label for="exampleInputEmail2">邮箱</label> <input type="email"
                        class="form-control" id="exampleInputEmail2">
                </div>
                <button type="submit" class="btn btn-default">查询</button>
            </form>

        </div>

        <div style="float: right; margin: 5px;">
            <a class="btn btn-primary"
                href="${pageContext.request.contextPath}/add.jsp">添加联系人</a> <a
                class="btn btn-primary" href="javascript:void(0);" id="delSelected">
删除选中</a>
        </div>

        <form id="form"
            action="${pageContext.request.contextPath}/delSelectedServlet"
            method="post">
            <table border="1" class="table table-bordered table-hover">
                <tr class="success">
                    <th><input type="checkbox" id="firstCb"></th>
                    <th>编号</th>
                    <th>姓名</th>
                    <th>性别</th>
                    <th>年龄</th>
                    <th>籍贯</th>
                    <th>QQ</th>
                    <th>邮箱</th>
                    <th>操作</th>
                </tr>

                <c:forEach items="${pb.list}" var="user" varStatus="s">
                    <tr>
                        <td><input type="checkbox" name="uid"
value="${user.id}"></td>
                        <td>${s.count}</td>
                        <td>${user.name}</td>
                        <td>${user.gender}</td>
                        <td>${user.age}</td>
                        <td>${user.address}</td>
                        <td>${user.qq}</td>
                        <td>${user.email}</td>
                        <td><a class="btn btn-default btn-sm"

    href="${pageContext.request.contextPath}/findUserServlet?id=${user.id}">修改
</a> 
                            <a class="btn btn-default btn-sm"

    href="javascript:deleteUser(${user.id},'${user.name}');">删除</a></td>
                    </tr>
```

```
                </c:forEach>
            </table>
        </form>

        <div>
            <nav aria-label="Page navigation">
                <ul class="pagination">
                <c:if test="${pb.currentPage == 1}">
                    <li class="hide">
                </c:if>
                <c:if test="${pb.currentPage != 1}">
                    <li>
                </c:if>
                        <a
href="${pageContext.request.contextPath}/findUserByPageServlet?currentPage=${pb.cur
rentPage - 1}&rows=5" aria-label="Previous">  <!--上一页，修改href-->
                        <span aria-hidden="true">&laquo;</span>
                        </a></li>

                        <c:forEach begin="1" end="${pb.totalPage}" var="i">
                         <c:if test="${pb.currentPage == i}">   <!--选中的页码显示样式变
为激活状态-->
                            <li class="active"><a
href="${pageContext.request.contextPath}/findUserByPageServlet?currentPage=${i}&row
s=5">${i}</a></li>
                        </c:if>
                        <c:if test="${pb.currentPage != i}">  <!--未选中的页码-->
                            <li><a
href="${pageContext.request.contextPath}/findUserByPageServlet?currentPage=${i}&row
s=5">${i}</a></li>
                        </c:if>
                        </c:forEach>

                <c:if test="${pb.currentPage==pb.totalPage}">
                    <li class="hide">
                </c:if>
                <c:if test="${pb.currentPage!=pb.totalPage}">
                    <li>
                </c:if>
                        <a
href="${pageContext.request.contextPath}/findUserByPageServlet?currentPage=${pb.cur
rentPage + 1}&rows=5" aria-label="Next"> <!--下一页，修改href-->
                            <span aria-hidden="true">&raquo;</span>
                            </a>
                        </li>

                        <span style="font-size: 25px; margin-left: 5px;">
                            共${pb.totalCount}记录，共${pb.totalPage}页
                        </span>
                    </ul>
                </nav>
            </div>
        </div>
    </body>
    </html>
```

测试分页页面的最终效果：

自此，本次分页查询功能正式编写完毕。

四、思考

1. 尝试将分页查询功能封装成一个可复用的模块。