## day18\_综合练习

### 0.1学习目标

* 完成登录功能
* 完成增加功能
* 完成删除功能
* 完成修改功能
* 完成选中项删除功能
* 完成分页查询功能
* 完成复杂条件查询功能

### 0.2 学习指南

* 简单增删改查（重要）
* 选中项删除（重要）
* 分页条件查询（重要）

### 第一堂课

本节知识点

页面调整

登录功能\_验证码显示

登录功能\_代码实现

本节目标

完成登录功能

##### 1.1.1知识概述

调整案例所需页面的样式以及内容

##### 1.1.2视频详情



##### 1.1.3总结与补充

无

##### 1.1.4课堂提问与练习

无

##### 1.1.5习题答案

无

##### 1.1.6视频缺陷

无

##### 1.1.7视频扩展

无

#### 登录功能\_验证码显示

##### 1.2.1知识概述

登录的时候要先校验验证码是否正确，正确再去校验账号密码，密码正确放入session中保存用户信息

##### 1.2.2视频详情



##### 1.2.3总结与补充

需要注意的是用户验证成功后把登录信息存入session中。

##### 1.2.4课堂提问与练习

##### 1.2.5习题答案

##### 1.2.6视频缺陷

无

##### 1.2.7视频扩展

无

#### 登录功能\_代码实现

##### 1.3.1知识概述

request.setCharacterEncoding("utf-8");

//2.获取数据

//2.1获取用户填写验证码

String verifycode = request.getParameter("verifycode");

//3.验证码校验

HttpSession session = request.getSession();

String checkcode\_server = (String) session.getAttribute("CHECKCODE\_SERVER");

session.removeAttribute("CHECKCODE\_SERVER");//确保验证码一次性

if(!checkcode\_server.equalsIgnoreCase(verifycode)){

//验证码不正确

//提示信息

request.setAttribute("login\_msg","验证码错误！");

//跳转登录页面

request.getRequestDispatcher("/login.jsp").forward(request,response);

return;

}

Map<String, String[]> map = request.getParameterMap();

//4.封装User对象

User user = new User();

try {

BeanUtils.populate(user,map);

} catch (IllegalAccessException e) {

e.printStackTrace();

} catch (InvocationTargetException e) {

e.printStackTrace();

}

//5.调用Service查询

UserService service = new UserServiceImpl();

User loginUser = service.login(user);

//6.判断是否登录成功

if(loginUser != null){

//登录成功

//将用户存入session

session.setAttribute("user",loginUser);

//跳转页面

response.sendRedirect(request.getContextPath()+"/index.jsp");

}else{

//登录失败

//提示信息

request.setAttribute("login\_msg","用户名或密码错误！");

//跳转登录页面

request.getRequestDispatcher("/login.jsp").forward(request,response);

}

##### 1.3.2视频详情



##### 1.3.3总结与补充

##### 1.3.4课堂提问与练习

##### 1.3.5习题答案

##### 1.3.6视频缺陷

无

##### 1.3.7视频扩展

无

### 第二堂课

本节知识点：

添加功能\_分析

添加功能\_实现

删除功能\_分析

删除功能\_代码实现

修改功能\_分析

本节目标 ：

完成添加与修改功能

#### 添加功能\_分析

##### 2.1.1知识概述

添加页面，输入数据一定要有属性名称，提交form表单到后台，后台接收数据封装到user中，并编写sql语句

##### 2.1.2视频详情



##### 2.1.3总结与补充

无

##### 2.1.4课堂提问与练习

无

##### 2.1.5习题答案

无

##### 2.1.6视频缺陷

无

##### 2.1.7视频扩展

无

#### 添加功能\_实现

##### 2.2.1知识概述

//1.设置编码

request.setCharacterEncoding("utf-8");

//2.获取参数

Map<String, String[]> map = request.getParameterMap();

//3.封装对象

User user = new User();

try {

BeanUtils.populate(user,map);

} catch (IllegalAccessException e) {

e.printStackTrace();

} catch (InvocationTargetException e) {

e.printStackTrace();

}

//4.调用Service保存

UserService service = new UserServiceImpl();

service.addUser(user);

//5.跳转到userListServlet

response.sendRedirect(request.getContextPath()+"/userListServlet");

##### 2.2.2视频详情



##### 2.2.3总结与补充

##### 2.2.4课堂提问与练习

无

##### 2.2.5习题答案

无

##### 2.2.6视频缺陷

无

##### 2.2.7视频扩展

无

#### 删除功能\_分析

##### 2.3.1知识概述

通过id删除数据，编写sql语句。

##### 2.3.2视频详情



##### 2.3.3总结与补充

##### 2.3.4课堂提问与练习

无

##### 2.3.5习题答案

无

##### 2.3.6视频缺陷

无

##### 2.3.7视频扩展

无

#### 2.4删除功能\_代码实现

##### 2.4.1知识概述

代码：  
                    //1.获取id

String id = request.getParameter("id");

//2.调用service删除

UserService service = new UserServiceImpl();

service.deleteUser(id);

//3.跳转到查询所有Servlet

response.sendRedirect(request.getContextPath()+"/userListServlet");

##### 2.4.2视频详情



##### 2.4.3总结与补充

##### 2.4.4课堂提问与练习

##### 2.4.5习题答案

##### 2.4.6视频缺陷

无

##### 2.4.7视频扩展

无

#### 2.5修改功能\_分析

##### 2.5.1知识概述

首先要进行数据回显，然后提交修改后数据到后台，进行保存。

##### 2.5.2视频详情



##### 2.5.3总结与补充

##### 2.5.4课堂提问与练习

##### 2.5.5习题答案

##### 2.5.6视频缺陷

无

##### 2.5.7视频扩展

无

### 第三堂课

本节知识点

修改功能\_代码实现1\_用户信息回显

修改功能\_代码实现2\_用户修改

删除选中功能\_分析

删除选中功能\_代码实现\_获取选中条目id

删除选中功能\_代码实现\_删除选中服务器代码编写

本节目标

能够完成修改代码

能够完成删除选中条目代码

#### 修改功能\_代码实现1\_用户信息回显

##### 3.1.1知识概述

//1.获取id

String id = request.getParameter("id");

//2.调用Service查询

UserService service = new UserServiceImpl();

User user = service.findUserById(id);

//3.将user存入request

request.setAttribute("user",user);

//4.转发到update.jsp

request.getRequestDispatcher("/update.jsp").forward(request,response);

##### 3.1.2视频详情



##### 3.1.3总结与补充

##### 3.1.4课堂提问与练习

##### 3.1.5习题答案

##### 3.1.6视频缺陷

无

##### 3.1.7视频扩展

无

#### 3.2修改功能\_代码实现2\_用户修改

##### 3.2.1知识概述

//1.设置编码

request.setCharacterEncoding("utf-8");

//2.获取map

Map<String, String[]> map = request.getParameterMap();

//3.封装对象

User user = new User();

try {

BeanUtils.populate(user,map);

} catch (IllegalAccessException e) {

e.printStackTrace();

} catch (InvocationTargetException e) {

e.printStackTrace();

}

//4.调用Service修改

UserService service = new UserServiceImpl();

service.updateUser(user);

//5.跳转到查询所有Servlet

response.sendRedirect(request.getContextPath()+"/userListServlet");

##### 3.2.2视频详情



##### 3.2.3总结与补充

##### 3.2.4课堂提问与练习

##### 3.2.5习题答案

##### 3.2.6视频缺陷

无

##### 3.2.7视频扩展

无

#### 3.3删除选中功能\_分析

##### 3.3.1知识概述

首先要获取选中条目的id，然后根据id进行删除。

##### 3.3.2视频详情

##### 3.3.3总结与补充

##### 3.3.4课堂提问与练习

##### 3.3.5习题答案

##### 3.3.6视频缺陷

无

##### 3.3.7视频扩展

无

#### 3.4删除选中功能\_代码实现\_获取选中条目id

##### 3.4.1知识概述

//给删除选中按钮添加单击事件

document.getElementById("delSelected").onclick = function(){

if(confirm("您确定要删除选中条目吗？")){

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){//有条目被选中

//表单提交

document.getElementById("form").submit();

}

}

##### 3.4.2视频详情



##### 3.4.3总结与补充

##### 3.4.4课堂提问与练习

##### 3.4.5习题答

##### 3.4.6视频缺陷

无

##### 3.4.7视频扩展

无

#### 3.5删除选中功能\_代码实现\_删除选中服务器代码编写

##### 3.5.1知识概述

//1.获取所有id

String[] ids = request.getParameterValues("uid");

//2.调用service删除

UserService service = new UserServiceImpl();

service.delSelectedUser(ids);

//3.跳转查询所有Servlet

response.sendRedirect(request.getContextPath()+"/userListServlet");

##### 3.5.2视频详情



##### 3.5.3总结与补充

##### 3.5.4课堂提问与练习

##### 3.5.5习题答

##### 3.5.6视频缺陷

无

##### 3.5.7视频扩展

无

### 第四堂课

本节知识点：

删除选中功能\_代码实现\_删除选中细节处理

分页查询功能\_分析

分页查询功能\_代码实现\_分析具体步骤

本节目标 ：

完成删除选中功能的细节处理

完成分页查询功能

#### 删除选中功能\_代码实现\_删除选中细节处理

##### 4.1.1知识概述

判断前台页面传到后台的数据是否为空。避免出现空指针异常。

##### 4.1.2视频详情



##### 4.1.3总结与补充

##### 4.1.4课堂提问与练习

##### 4.1.5习题答案

##### 4.1.6视频缺陷

无

##### 4.1.7视频扩展

无

#### 4.2分页查询功能\_分析

##### 4.2.1知识概述

前台页面需要把当前页与每页显示条数传入后台。

##### 4.2.2视频详情



##### 4.2.3总结与补充

##### 4.2.4课堂提问与练习

##### 4.2.5习题答案

##### 4.2.6视频缺陷

无

##### 4.2.7视频扩展

无

#### 4.3分页查询功能\_代码实现\_分析具体步骤

##### 4.3.1知识概述

//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

if(currentPage == null || "".equals(currentPage)){

currentPage = "1";

}

if(rows == null || "".equals(rows)){

rows = "5";

}

//获取条件查询参数

Map<String, String[]> condition = request.getParameterMap();

//2.调用service查询

UserService service = new UserServiceImpl();

PageBean<User> pb = service.findUserByPage(currentPage,rows,condition);

System.out.println(pb);

//3.将PageBean存入request

request.setAttribute("pb",pb);

request.setAttribute("condition",condition);//将查询条件存入request

//4.转发到list.jsp

request.getRequestDispatcher("/list.jsp").forward(request,response);

}

##### 4.3.2视频详情



##### 4.3.3总结与补充

##### 4.3.4课堂提问与练习

##### 4.3.5习题答案

##### 4.3.6视频缺陷

无

##### 4.3.7视频扩展

无

### 第五堂课

本节知识点：

分页查询功能\_代码实现\_后台代码实现

分页查询功能\_代码实现\_前台代码实现

本节目标 ：

1、能够完成分页查询功能\_代码实现\_后台代码实现

2、能够完成分页查询功能\_代码实现\_前台代码实现

#### 5.1分页查询功能\_代码实现\_后台代码实现

##### 5.1.1知识概述

//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

if(currentPage == null || "".equals(currentPage)){

currentPage = "1";

}

if(rows == null || "".equals(rows)){

rows = "5";

}

//获取条件查询参数

Map<String, String[]> condition = request.getParameterMap();

//2.调用service查询

UserService service = new UserServiceImpl();

PageBean<User> pb = service.findUserByPage(currentPage,rows,condition);

System.out.println(pb);

//3.将PageBean存入request

request.setAttribute("pb",pb);

request.setAttribute("condition",condition);//将查询条件存入request

//4.转发到list.jsp

request.getRequestDispatcher("/list.jsp").forward(request,response);

}

##### 5.1.2视频详情



##### 5.1.3总结与补充

##### 5.1.4课堂提问与练习

##### 5.1.5习题答案

##### 5.1.6视频缺陷

无

##### 5.1.7视频扩展

无

#### 5.2分页查询功能\_代码实现\_前台代码实现

##### 5.2.1知识概述

                 <nav aria-label="Page navigation">

<ul class="pagination">

<c:if test="${pb.currentPage == 1}">

<li class="disabled">

</c:if>

<c:if test="${pb.currentPage != 1}">

<li>

</c:if>

<a href="${pageContext.request.contextPath}/findUserByPageServlet?currentPage=${pb.currentPage - 1}&rows=5&name=${condition.name[0]}&address=${condition.address[0]}&email=${condition.email[0]}" aria-label="Previous">

<span aria-hidden="true">&laquo;</span>

</a>

</li>

<c:forEach begin="1" end="${pb.totalPage}" var="i" >

<c:if test="${pb.currentPage == i}">

<li class="active"><a href="${pageContext.request.contextPath}/findUserByPageServlet?currentPage=${i}&rows=5&name=${condition.name[0]}&address=${condition.address[0]}&email=${condition.email[0]}">${i}</a></li>

</c:if>

<c:if test="${pb.currentPage != i}">

<li><a href="${pageContext.request.contextPath}/findUserByPageServlet?currentPage=${i}&rows=5&name=${condition.name[0]}&address=${condition.address[0]}&email=${condition.email[0]}">${i}</a></li>

</c:if>

</c:forEach>

<li>

<a href="${pageContext.request.contextPath}/findUserByPageServlet?currentPage=${pb.currentPage + 1}&rows=5&name=${condition.name[0]}&address=${condition.address[0]}&email=${condition.email[0]}" aria-label="Next">

<span aria-hidden="true">&raquo;</span>

</a>

</li>

<span style="font-size: 25px;margin-left: 5px;">

共${pb.totalCount}条记录，共${pb.totalPage}页

</span>

##### 5.2.2视频详情



##### 5.2.3总结与补充

##### 5.2.4课堂提问与练习

##### 5.2.5习题答案

##### 5.2.6视频缺陷

无

##### 5.2.7视频扩展

无

### 第六堂课

本节知识点：

1复杂条件查询功能\_分析

2复杂条件查询功能\_代码实现1\_总记录数统计

3复杂条件查询功能\_代码实现2\_每页数据条件查询

本节目标 ：

完成复杂条件查询功能

#### 复杂条件查询功能\_分析

##### 6.1.1知识概述

             把查询条件发送到后台，后台根据条件进行查询

##### 6.1.2视频详情



##### 6.1.3总结与补充

##### 6.1.4课堂提问与练习

##### 6.1.5习题答案

##### 6.1.6视频缺陷

无

##### 6.1.7视频扩展

无

#### 复杂条件查询功能\_代码实现1\_总记录数统计J

##### 6.2.1知识概述

//获取条件查询参数

Map<String, String[]> condition = request.getParameterMap();

//2.调用service查询

UserService service = new UserServiceImpl();

PageBean<User> pb = service.findUserByPage(currentPage,rows,condition);

//1.定义模板初始化sql

String sql = "select count(\*) from user where 1 = 1 ";

StringBuilder sb = new StringBuilder(sql);

//2.遍历map

Set<String> keySet = condition.keySet();

//定义参数的集合

List<Object> params = new ArrayList<Object>();

for (String key : keySet) {

//排除分页条件参数

if("currentPage".equals(key) || "rows".equals(key)){

continue;

}

//获取value

String value = condition.get(key)[0];

//判断value是否有值

if(value != null && !"".equals(value)){

//有值

sb.append(" and "+key+" like ? ");

params.add("%"+value+"%");//？条件的值

}

}

System.out.println(sb.toString());

System.out.println(params);

return template.queryForObject(sb.toString(),Integer.class,params.toArray());

##### 6.2.2视频详情



##### 6.2.3总结与补充

##### 6.2.4课堂提问与练习

无

##### 6.2.5习题答案

无

##### 6.2.6视频缺陷

无

##### 6.2.7视频扩展

无

#### 复杂条件查询功能\_代码实现2\_每页数据条件查询

##### 6.3.1知识概述

String sql = "select \* from user where 1 = 1 ";

StringBuilder sb = new StringBuilder(sql);

//2.遍历map

Set<String> keySet = condition.keySet();

//定义参数的集合

List<Object> params = new ArrayList<Object>();

for (String key : keySet) {

//排除分页条件参数

if("currentPage".equals(key) || "rows".equals(key)){

continue;

}

//获取value

String value = condition.get(key)[0];

//判断value是否有值

if(value != null && !"".equals(value)){

//有值

sb.append(" and "+key+" like ? ");

params.add("%"+value+"%");//？条件的值

}

}

//添加分页查询

sb.append(" limit ?,? ");

//添加分页查询参数值

params.add(start);

params.add(rows);

sql = sb.toString();

System.out.println(sql);

System.out.println(params);

return template.query(sql,new BeanPropertyRowMapper<User>(User.class),params.toArray());

##### 6.3.2视频详情



##### 6.3.3总结与补充

##### 6.3.4课堂提问与练习

##### 6.3.5习题答案

##### 6.3.6视频缺陷

无

##### 6.3.7视频扩展

无