

第13章：综合案例

第1节：功能介绍

学生管理系统是基于JSP+Servlet+JDBC实现的后台管理系统，该系统主要用于高校管理在校学生的信息。主要功能有：系统登录、学生信息的添加、查看、修改、删除功能。本章主要将JavaWeb目前所学内容总结使用。

第2节：准备

1. 功能 系统登录、 添加学生 查询学生(显示所有、显示单个) 修改 删除商品(单个、多个、全部)
2. 分析库和表 系统用户User、学生Student
3. 创建数据库和表

1. 创建库

```
create database if not exists students;
```

2. 选中库

```
use students;
```

3. 创建表

```
create table if not exists user(  
    id int primary key auto_increment comment '编号',  
    username varchar(32) not null comment '账户',  
    pwd varchar(64) comment '密码',  
    role int comment '角色'  
) comment '管理员表';
```

```
CREATE TABLE stu(  
    sid INT PRIMARY KEY AUTO_INCREMENT comment '学号',  
    sname VARCHAR(50) comment '姓名',  
    gender VARCHAR(2) comment '性别',  
    birthday DATE comment '出生日期',  
    hobby VARCHAR(200) comment '爱好',  
    photo VARCHAR(200) comment '照片'  
) comment '学生表';
```

4. 搭建应用（基于MVC+三层结构）
5. 引入依赖（第三方jar包、配置文件、封装工具类和通用dao）

第3节：功能实现

3.1 系统登录

登录页login.jsp点击登录按钮 ----> LoginServlet中（获取登录账户和密码） ----> 调用业务层处理登录业务 ----> 调用Dao校验账户密码是否正确 返回信息 ----> LoginServlet中根据返回信息 跳转页面

3.2 首页数据

登录成功之后 -----> 请求后台HomeServlet(获取首页数据) -----> 调用业务层处理首页业务 -----> 调用Dao 查询数据 返回结果 -----> HomeServlet根据返回信息 跳转首页 home.jsp展示数据

3.3 添加功能

home.jsp中点击添加按钮 -----> AddStuUIServlet(转发或者重定向add.jsp页面) -----> 在添加页面中点击添加 -----> AddStuServlet(获取要添加数据) -----> 调用业务层处理添加业务 -----> 调用Dao 保存数据库 返回结果 -----> AddStuServlet根据返回信息 跳转首页 home.jsp 展示数据

3.4 修改功能

home.jsp中点击修改 -----> UpdateStuUIServlet(当前修改的商品查询出来, 转发或者重定向update.jsp 回显老数据) -----> 在修改页面点击修改按钮 -----> UpdateStuServlet(获取修改后的数据) -----> 调用业务层处理修改业务 -----> 调用Dao 修改数据库 返回结果 -----> UpdateStuServlet根据返回信息 跳转首页 home.jsp 展示数据

3.5 删除功能

1. 单删

home.jsp中点击删除 -----> DeleteStuServlet(获取要删除的学生id) -----> 调用业务层处理删除业务 -----> 调用Dao 修改数据库 返回结果 -----> DeleteStuServlet根据返回信息 跳转首页 home.jsp 展示数据

2. 多删

home.jsp中选中要删除信息点击多删除 -----> DeleteStuMoreServlet(获取要删除的学生id) -----> 调用业务层处理多删业务 -----> 调用Dao 修改数据库 返回结果 -----> DeleteStuMoreServlet根据返回信息 跳转首页 home.jsp 展示数据

3.6 分页功能

1. 分页需要的数据:

1. 默认第一页
2. 当前页码
3. 当前页的数据 6条数据 从数据库查询
4. 一共多少页 数据个数(数据库查询) / 一页显示的个数(自定义) 30 / 6

2. 将以上信息封装PageBean

```
public class PageBean {  
  
    private int currentPage;    // 当前页的页码  
  
    private int pageSize = 6;    // 一页显示的个数  
  
    private int totalCount;    // 一共有多少条数据  
  
    private int totalPage;    // 一共多少页
```

```
private List<Product> list; // 当前页需要的数据
```

```
private int startPage; // 开始页面
```

```
private int endPage; // 结束页面
```

3. 将数据保存到一个 PageBean 实体类对象

4. 将有数据的 pageBean 保存在共享域中 转发页面