实习三 EL、JSTL及MVC

1. 实习要求与目的
2. 掌握标签库的使用
3. 掌握EL及JSTL常用标签及方法
4. 掌握MVC的编程模式，完成一个完整的信息管理系统
5. 实习内容

使用EL、JSTL及MVC完成一个图书管理系统，要求完成图书信息的添加、修改、删除、查询与显示；并添加用户管理功能，只有登录用户才能完成图书的管理操作。

1. 扩展

试着使用过滤器作为控制器完成实习内容。

**实习报告提交地址：[www.zqldm.top:81](http://www.zqldm.top:81)**

项目目录结构：

E:.

├───.idea //存放idea配置

└───src //存放java源代码

├───main

│ ├───java

│ │ └───com

│ │ └───dlct

│ │ ├───controller //存放servlet文件

│ │ ├───dao //存放dao接口

│ │ ├───Filter //存放过滤器

│ │ ├───pojo //存放java bean

│ │ ├───service //存放service类用于处理业务

│ │ └───utils //存放工具类

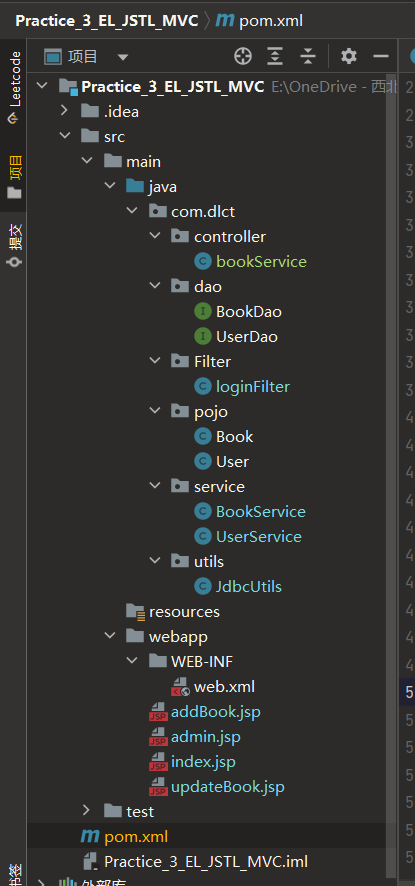
│ ├───resources

│ └───webapp //存放jsp文件

│ └───WEB-INF

└───test

└───java



Sql：

创建两张表：User表与Book表。User表存放用户名及其密码，Book表存放书籍信息

User表

/\*

 Navicat Premium Data Transfer

 Source Server         : MySQL8

 Source Server Type    : MySQL

 Source Server Version : 80030

 Source Host           : localhost:3306

 Source Schema         : ssm

 Target Server Type    : MySQL

 Target Server Version : 80030

 File Encoding         : 65001

 Date: 14/10/2022 15:31:41

\*/

SET NAMES utf8mb4;

SET FOREIGN\_KEY\_CHECKS = 0;

-- ----------------------------

-- Table structure for user

-- ----------------------------

DROP TABLE IF EXISTS `user`;

CREATE TABLE `user`  (

  `name` varchar(30) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,

  `pwd` varchar(30) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL

) ENGINE = InnoDB CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = Dynamic;

-- ----------------------------

-- Records of user

-- ----------------------------

INSERT INTO `user` VALUES ('123456', '123456');

SET FOREIGN\_KEY\_CHECKS = 1;

**Book表**

/\*

 Navicat Premium Data Transfer

 Source Server         : MySQL8

 Source Server Type    : MySQL

 Source Server Version : 80030

 Source Host           : localhost:3306

 Source Schema         : ssm

 Target Server Type    : MySQL

 Target Server Version : 80030

 File Encoding         : 65001

 Date: 14/10/2022 15:31:31

\*/

SET NAMES utf8mb4;

SET FOREIGN\_KEY\_CHECKS = 0;

-- ----------------------------

-- Table structure for books

-- ----------------------------

DROP TABLE IF EXISTS `books`;

CREATE TABLE `books`  (

  `bookID` int NOT NULL AUTO\_INCREMENT COMMENT '书id',

  `bookName` varchar(100) CHARACTER SET utf8mb3 COLLATE utf8mb3\_general\_ci NOT NULL COMMENT '书名',

  `bookCounts` int NOT NULL COMMENT '数量',

  `detail` varchar(200) CHARACTER SET utf8mb3 COLLATE utf8mb3\_general\_ci NOT NULL COMMENT '描述',

  PRIMARY KEY (`bookID`) USING BTREE

) ENGINE = InnoDB AUTO\_INCREMENT = 7 CHARACTER SET = utf8mb3 COLLATE = utf8mb3\_general\_ci ROW\_FORMAT = Dynamic;

-- ----------------------------

-- Records of books

-- ----------------------------

INSERT INTO `books` VALUES (1, 'java', 10, '从入门到放弃');

INSERT INTO `books` VALUES (2, 'MySQL', 10, '从删库到跑路');

INSERT INTO `books` VALUES (3, 'Linux', 5, '从进门到坐牢');

SET FOREIGN\_KEY\_CHECKS = 1;

**POJO类**

**Book.java**

**package** com.dlct.pojo;  
  
**public class** Book {  
 **private int bookID**;  
 **private** String **bookName**;  
 **private int bookCounts**;  
 **private** String **detail**;  
  
 **public int** getBookID() {  
 **return bookID**;  
 }  
  
 **public void** setBookID(**int** bookID) {  
 **this**.**bookID** = bookID;  
 }  
  
 **public** String getBookName() {  
 **return bookName**;  
 }  
  
 **public void** setBookName(String bookName) {  
 **this**.**bookName** = bookName;  
 }  
  
 **public int** getBookCounts() {  
 **return bookCounts**;  
 }  
  
 **public void** setBookCounts(**int** bookCounts) {  
 **this**.**bookCounts** = bookCounts;  
 }  
  
 **public** String getDetail() {  
 **return detail**;  
 }  
  
 **public void** setDetail(String detail) {  
 **this**.**detail** = detail;  
 }  
  
 **public** Book() {  
 }  
  
 **public** Book(**int** bookID, String bookName, **int** bookCounts, String detail) {  
 **this**.**bookID** = bookID;  
 **this**.**bookName** = bookName;  
 **this**.**bookCounts** = bookCounts;  
 **this**.**detail** = detail;  
 }  
  
 @Override  
 **public** String toString() {  
 **return "Book{"** +  
 **"bookID="** + **bookID** +  
 **", bookName='"** + **bookName** + **'\''** +  
 **", bookCounts="** + **bookCounts** +  
 **", detail='"** + **detail** + **'\''** +  
 **'}'**;  
 }  
}

**User.java**

**package** com.dlct.pojo;  
  
**public class** User {  
 **private** String **name**;  
 **private** String **pwd**;  
  
 **public** String getName() {  
 **return name**;  
 }  
  
 **public** User(String name) {  
 **this**.**name** = name;  
 }  
  
 **public** User() {  
 }  
  
 **public void** setName(String name) {  
 **this**.**name** = name;  
 }  
  
 **public** String getPwd() {  
 **return pwd**;  
 }  
  
 **public void** setPwd(String pwd) {  
 **this**.**pwd** = pwd;  
 }  
  
 @Override  
 **public** String toString() {  
 **return "User{"** +  
 **"name='"** + **name** + **'\''** +  
 **", pwd='"** + **pwd** + **'\''** +  
 **'}'**;  
 }  
}

**DAO层：**

**UserDao.java**

**package** com.dlct.dao;  
  
**import** com.dlct.pojo.User;  
  
**import** java.sql.SQLException;  
**import** java.util.List;  
  
**public interface** UserDao {  
 List<User> queryAllUser() **throws** SQLException;  
 User queryUserByName(String name) **throws** SQLException;  
}

**BookDao.java**

**package** com.dlct.dao;  
  
**import** com.dlct.pojo.Book;  
  
**import** java.sql.SQLException;  
**import** java.util.List;  
  
**public interface** BookDao {  
  
 List<Book> queryAllBook() **throws** SQLException;  
  
  
 List<Book> queryBookByName(String name) **throws** SQLException;  
  
 Book queryBookById(**int** id) **throws** SQLException;  
  
 **int** updateBook(Book book) **throws** SQLException;  
  
 **int** deleteBook(**int** id) **throws** SQLException;  
  
 **int** insertBook(Book book) **throws** SQLException;  
}

**JDBC工具类:**

**package** com.dlct.utils;  
  
**import** java.sql.\*;  
  
**public class** JdbcUtils {  
 **private static** String *driver* = **"com.mysql.cj.jdbc.Driver"**;  
 **private static** String *url* = **"jdbc:mysql://localhost:3306/ssm?useSSL=true&characterEncoding=UTF-8&useUnicode=true&serverTimezone=GMT"**;  
 **private static** String *userName* = **"root"**;  
 **private static** String *passWord* = **"123456"**;  
  
 *//获取连接* **public static** Connection getConnection(){  
 Connection connection = **null**;  
 **try** {  
 *//注册驱动* Class.*forName*(*driver*);  
 *//获取连接* connection = DriverManager.*getConnection*(*url*, *userName*, *passWord*);  
 } **catch** (ClassNotFoundException | SQLException e) {  
 e.printStackTrace();  
 }  
 **return** connection;  
 }  
  
 *//关闭连接* **public static boolean** close(Connection connection,Statement statement,ResultSet resultSet){  
 **boolean** flag = **true**;  
 **try** {  
 **if** (connection!=**null**){  
 connection.close();  
 }  
 **if** (statement!=**null**){  
 statement.close();  
 }  
 **if** (resultSet!=**null**){  
 resultSet.close();  
 }  
 }**catch** (SQLException e){  
 e.printStackTrace();  
 flag = **false**;  
 }  
 **return** flag;  
 }  
}

**Service类：**

**UserService.java**

**package** com.dlct.service;  
  
**import** com.dlct.dao.UserDao;  
**import** com.dlct.pojo.User;  
**import** com.dlct.utils.JdbcUtils;  
  
**import** java.sql.\*;  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
**public class** UserService **implements** UserDao{  
  
 @Override  
 **public** List<User> queryAllUser() **throws** SQLException {  
 *//查询所有用户* Connection connection = JdbcUtils.*getConnection*();  
 Statement stat = connection.createStatement();  
 String sql = **"select \* from ssm.user"**;  
 ResultSet res = stat.executeQuery(sql);  
  
 ArrayList<User> users = **new** ArrayList<>();  
 User user = **new** User();  
 **while**(res.next()){  
 user.setName(res.getString(**"name"**));  
 user.setPwd(res.getString(**"pwd"**));  
 users.add(user);  
 }  
  
 JdbcUtils.*close*(connection, stat, res);  
 **return** users;  
 }  
  
 @Override  
 **public** User queryUserByName(String name) **throws** SQLException {  
 *//查询所有用户* Connection connection = JdbcUtils.*getConnection*();  
 Statement stat = connection.createStatement();  
 String sql = **"select \* from javaee\_practice.user where name="** + name;  
 ResultSet res = stat.executeQuery(sql);  
  
 User user = **new** User();  
 **while**(res.next()){  
 user.setName(res.getString(**"name"**));  
 user.setPwd(res.getString(**"pwd"**));  
 }  
  
 JdbcUtils.*close*(connection, stat, res);  
 **return** user;  
 }  
}

**BookService.java**

**package** com.dlct.service;  
  
**import** com.dlct.dao.BookDao;  
**import** com.dlct.pojo.Book;  
**import** com.dlct.pojo.User;  
**import** com.dlct.utils.JdbcUtils;  
  
**import** java.sql.Connection;  
**import** java.sql.ResultSet;  
**import** java.sql.SQLException;  
**import** java.sql.Statement;  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
**public class** BookService **implements** BookDao {  
 @Override  
 **public** List<Book> queryAllBook() **throws** SQLException {  
 Connection connection = JdbcUtils.*getConnection*();  
 Statement stat = connection.createStatement();  
 String sql = **"select \* from ssm.books"**;  
 ResultSet res = stat.executeQuery(sql);  
  
 ArrayList<Book> books = **new** ArrayList<>();  
 **while**(res.next()){  
 Book book = **new** Book();  
 book.setBookID(res.getInt(**"bookID"**));  
 book.setBookName(res.getString(**"bookName"**));  
 book.setBookCounts(res.getInt(**"bookCounts"**));  
 book.setDetail(res.getString(**"detail"**));  
 books.add(book);  
 }  
 JdbcUtils.*close*(connection, stat, res);  
 **return** books;  
 }  
  
 @Override  
 **public** List<Book> queryBookByName(String name) **throws** SQLException {  
 Connection connection = JdbcUtils.*getConnection*();  
 Statement stat = connection.createStatement();  
 String sql = **"select \* from ssm.books where bookName ='"** + name + **"'"**;  
 ResultSet res = stat.executeQuery(sql);  
  
  
 ArrayList<Book> books = **new** ArrayList<>();  
 **while**(res.next()){  
 Book book = **new** Book();  
 book.setBookID(res.getInt(**"bookID"**));  
 book.setBookName(res.getString(**"bookName"**));  
 book.setBookCounts(res.getInt(**"bookCounts"**));  
 book.setDetail(res.getString(**"detail"**));  
 books.add(book);  
 }  
  
 JdbcUtils.*close*(connection, stat, res);  
 **return** books;  
 }  
  
 @Override  
 **public** Book queryBookById(**int** id) **throws** SQLException {  
 Connection connection = JdbcUtils.*getConnection*();  
 Statement stat = connection.createStatement();  
 String sql = **"select \* from ssm.books where bookID="** + id;  
 ResultSet res = stat.executeQuery(sql);  
  
 Book book = **new** Book();  
 **while**(res.next()){  
  
 book.setBookID(res.getInt(**"bookID"**));  
 book.setBookName(res.getString(**"bookName"**));  
 book.setBookCounts(res.getInt(**"bookCounts"**));  
 book.setDetail(res.getString(**"detail"**));  
 }  
  
 JdbcUtils.*close*(connection, stat, res);  
 **return** book;  
 }  
  
 @Override  
 **public int** updateBook(Book book) **throws** SQLException {  
 Connection connection = JdbcUtils.*getConnection*();  
 Statement stat = connection.createStatement();  
 String sql = **"update ssm.books "** +  
 **"set bookName='"** + book.getBookName()+  
 **"', bookCounts="** + book.getBookCounts() +  
 **", detail='"** + book.getDetail() +  
 **"' where bookID="** + book.getBookID();  
 **int** i = stat.executeUpdate(sql);  
 JdbcUtils.*close*(connection, stat, **null**);  
 **return** i;  
 }  
  
 @Override  
 **public int** deleteBook(**int** id) **throws** SQLException {  
 Connection connection = JdbcUtils.*getConnection*();  
 Statement stat = connection.createStatement();  
 String sql = **"delete from ssm.books where bookID="** + id;  
 **int** i = stat.executeUpdate(sql);  
 JdbcUtils.*close*(connection, stat, **null**);  
 **return** i;  
 }  
  
 @Override  
 **public int** insertBook(Book book) **throws** SQLException {  
 Connection connection = JdbcUtils.*getConnection*();  
 Statement stat = connection.createStatement();  
 String sql = **"insert into ssm.books(bookName, bookCounts, detail) "** +  
 **"VALUE ('"** + book.getBookName() + **"', "** + book.getBookCounts() +  
 **", '"** + book.getDetail() +**"')"**;  
 **int** i = stat.executeUpdate(sql);  
 JdbcUtils.*close*(connection, stat, **null**);  
 **return** i;  
 }  
}

**过滤器**

**LoginFilter.java**

**package** com.dlct.Filter;  
  
**import** com.dlct.dao.BookDao;  
**import** com.dlct.dao.UserDao;  
**import** com.dlct.pojo.Book;  
**import** com.dlct.pojo.User;  
**import** com.dlct.service.BookService;  
**import** com.dlct.service.UserService;  
**import** org.apache.ibatis.session.SqlSession;  
  
**import** javax.servlet.\*;  
**import** javax.servlet.http.Cookie;  
**import** javax.servlet.http.HttpServletRequest;  
**import** javax.servlet.http.HttpServletResponse;  
**import** javax.servlet.http.HttpSession;  
**import** java.io.IOException;  
**import** java.sql.SQLException;  
**import** java.util.List;  
  
**public class** loginFilter **implements** Filter {  
 **public void** doFilter(ServletRequest req, ServletResponse resp, FilterChain filterChain) **throws** IOException, ServletException {  
 UserDao userService = **new** UserService();  
 HttpServletRequest request = (HttpServletRequest) req;  
 HttpServletResponse response = (HttpServletResponse) resp;  
 HttpSession session = request.getSession();  
  
 **if** (session.getAttribute(**"rem"**) != **null**){  
 *//第二次登录直接放行* filterChain.doFilter(req, resp);  
  
 **return**;  
 }  
  
 String error = **null**;  
 String account = (String) req.getParameter(**"account"**);  
 String password = (String) req.getParameter(**"password"**);  
  
 User user = **null**;  
 **try** {  
 user = userService.queryUserByName(account);  
 } **catch** (SQLException e) {  
 **throw new** RuntimeException(e);  
 }  
  
 **if** (user == **null**) {  
 error = **"账号"**;  
 } **else if** (!user.getPwd().equals(password)) {  
 error = **"密码"**;  
 }  
  
 **if** (error != **null**) {  
 *//有错* session.setAttribute(**"error\_my"**, error);  
 System.***out***.println(**"有错"**);  
 response.sendRedirect(**"login.jsp"**);  
 } **else** {  
 *//放行* Cookie cookie1 = **new** Cookie(**"account"**, account);  
 Cookie cookie2 = **new** Cookie(**"password"**, password);  
 response.addCookie(cookie1);  
 response.addCookie(cookie2);  
  
 session.setAttribute(**"rem"**, **true**);  
 session.removeAttribute(**"error\_my"**);  
  
 filterChain.doFilter(req, resp);  
 System.***out***.println(**"无错"**);  
 }  
 }  
  
  
 @Override  
 **public void** init(FilterConfig filterConfig) **throws** ServletException {  
  
 }  
  
 @Override  
 **public void** destroy() {  
 Filter.**super**.destroy();  
 }  
}

**Servlet：bookServlet.java**

**package** com.dlct.controller;  
  
**import** com.dlct.dao.BookDao;  
**import** com.dlct.pojo.Book;  
**import** com.dlct.service.BookService;  
  
**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.sql.SQLException;  
**import** java.util.List;  
  
@WebServlet(urlPatterns = **"/book"**)  
**public class** bookService **extends** HttpServlet {  
 @Override  
 **protected void** doGet(HttpServletRequest req, HttpServletResponse resp) **throws** ServletException, IOException {  
 String type = req.getParameter(**"type"**);  
 **if**(type.equals(**"addBook"**)) {  
 *//添加书籍* BookDao bookService = **new** BookService();  
 String bookName = req.getParameter(**"bookName"**);  
 **int** bookCounts = Integer.*parseInt*(req.getParameter(**"bookCounts"**));  
 String detail = req.getParameter(**"detail"**);  
  
 Book book = **new** Book(1, bookName, bookCounts, detail);  
 **try** {  
 bookService.insertBook(book);  
 } **catch** (SQLException e) {  
 **throw new** RuntimeException(e);  
 }  
 resp.sendRedirect(**"/book?type=allBook"**);  
 }**else if** (type.equals(**"allBook"**)){  
 *//获取所有书籍* BookDao bookService = **new** BookService();  
 List<Book> books = **null**;  
 **try** {  
 books = bookService.queryAllBook();  
 } **catch** (SQLException e) {  
 **throw new** RuntimeException(e);  
 }  
  
 req.setAttribute(**"list"**, books);  
 req.getRequestDispatcher(**"/admin.jsp"**).forward(req, resp);  
 }  
 **else if**(type.equals(**"deleteBook"**)){  
 *//删除书籍* **int** id = Integer.*parseInt*(req.getParameter(**"id"**));  
 BookDao bookService = **new** BookService();  
 **try** {  
 **int** i = bookService.deleteBook(id);  
 } **catch** (SQLException e) {  
 **throw new** RuntimeException(e);  
 }  
 resp.sendRedirect(**"/book?type=allBook"**);  
 }  
 **else if**(type.equals(**"selectBook"**)){  
 *//查询书籍* String bookName = req.getParameter(**"bookName"**);  
 BookDao bookService = **new** BookService();  
 List<Book> book = **null**;  
 **try** {  
 book = bookService.queryBookByName(bookName);  
 } **catch** (SQLException e) {  
 **throw new** RuntimeException(e);  
 }  
 **if** (book.isEmpty()){  
 **try** {  
 book = bookService.queryAllBook();  
 } **catch** (SQLException e) {  
 **throw new** RuntimeException(e);  
 }  
 req.setAttribute(**"errorr"**, **"未找到"**);  
 System.***out***.println(**"未找到"**);  
 }  
 System.***out***.println(book);  
 *//req.removeAttribute("list");* req.setAttribute(**"list"**, book);  
 req.getRequestDispatcher(**"/admin.jsp"**).forward(req, resp);  
 }**else if**(type.equals(**"updateBook"**)){  
 *//更新书籍* BookDao bookService = **new** BookService();  
 **int** Id = Integer.*parseInt*(req.getParameter(**"bookID"**));  
 String bookName = req.getParameter(**"bookName"**);  
 **int** bookCounts = Integer.*parseInt*(req.getParameter(**"bookCounts"**));  
 String detail = req.getParameter(**"detail"**);  
  
 Book book = **new** Book(Id, bookName, bookCounts, detail);  
 **try** {  
 bookService.updateBook(book);  
 } **catch** (SQLException e) {  
 **throw new** RuntimeException(e);  
 }  
 resp.sendRedirect(**"/book?type=allBook"**);  
 }  
 }  
  
 @Override  
 **protected void** doPost(HttpServletRequest req, HttpServletResponse resp) **throws** ServletException, IOException {  
 doGet(req, resp);  
 }  
}

**JSP文件**

**Login.jsp**

*<%--  
 Created by IntelliJ IDEA.  
 User: 18109  
 Date: 2022/9/27  
 Time: 20:31  
 To change this template use File | Settings | File Templates.  
--%>*<%@ **page contentType**="**text/html;charset=UTF-8**" **language**="**java**" %>  
<**html**>  
<**head**>  
 <**title**>登录界面</**title**>  
 **<%** String account = **""**;  
 String password = **""**;  
 String error = (String) session.getAttribute(**"error\_my"**);  
 **if** (session.getAttribute(**"rem"**) != **null**) {  
 Cookie[] cookies = request.getCookies();  
 **if** (cookies != **null**) {  
 **for** (Cookie cookie : cookies) {  
 **if** (**"account"**.equals(cookie.getName())) {  
 account = (String) cookie.getValue();  
 }  
 **if** (**"password"**.equals(cookie.getName())) {  
 password = (String) cookie.getValue();  
 }  
 }  
 }  
**%>** <**script**>  
 **var** ju = **true**;  
 </**script**>  
**<%** }  
 **if**(error != **null**){  
 **%>** <**script type="text/javascript"**>  
 **var** ju = **false**;  
 alert(**"<%=**error**%>错误"**);  
 </**script**>  
 **<%** }  
 **%>**</**head**>  
<**body**>  
 <**form action="${**pageContext.request.contextPath**}/book/allBook" method="post"**>  
 账号：<**input type="text" name="account" value="<%=**account**%>"**> <**br**>  
 密码：<**input type="password" name="password" value="<%=**password**%>"**> <**br**>  
 自动登录：<**input type="checkbox" name="remember" value="true"**> <**br**>  
 <**input type="submit" value="提交"**>  
 </**form**>  
 <**script**>  
 **if** (ju === **true**){  
 document.forms[0].submit();  
 }  
 </**script**>  
</**body**>  
</**html**>

**Admin.jsp**

<%@ **taglib prefix**="**c**" **uri**="**http://java.sun.com/jsp/jstl/core**" %>  
*<%--  
 Created by IntelliJ IDEA.  
 User: 18109  
 Date: 2022/10/13  
 Time: 11:17  
 To change this template use File | Settings | File Templates.  
--%>*<%@ **page contentType**="**text/html;charset=UTF-8**" **language**="**java**" %>  
<**html**>  
<**head**>  
 <**title**>数据展示</**title**>  
 <**link href="https://cdn.staticfile.org/twitter-bootstrap/3.3.7/css/bootstrap.min.css" rel="stylesheet"**>  
</**head**>  
<**body**>  
<**div class="container"**>  
 <**div class="row clearfix"**>  
 <**div class="col-md-12 column"**>  
 <**div class="page-header"**>  
 <**h1**>  
 <**small**>书籍列表 ----- 显示所有书籍</**small**>  
 </**h1**>  
 </**div**>  
 </**div**>  
  
 <**div class="row"**>  
 <**div class="col-md-4 column"**>  
 <**a class="btn btn-primary" href="${**pageContext.request.contextPath**}/addBook.jsp"**>新增书籍</**a**>  
 </**div**>  
 <**div class="col-md-4 column"**></**div**>  
 <**div class="col-md-4 column"**>  
 <**form action="${**pageContext.request.contextPath**}/book/selectBookByName" method="post" class="form-inline"**>  
 <**span style="color**: **red**; **font-weight**: **bold"**>**${**errorr**}**</**span**>  
 <**input type="text" name="bookName" class="form-control" placeholder="请输入要查询的书籍名称"**>  
 <**input type="submit" value="查询" class="btn btn-primary"**>  
 </**form**>  
 </**div**>  
 </**div**>  
  
 </**div**>  
 <**div class="row clearfix"**>  
 <**div class="col-md-12 column"**>  
 <**table class="table table-hover table-striped"**>  
 <**thead**>  
 <**tr**>  
 <**th**>书籍编号</**th**>  
 <**th**>书籍名称</**th**>  
 <**th**>书籍数量</**th**>  
 <**th**>书籍详情</**th**>  
 <**th**>操作</**th**>  
 </**tr**>  
 </**thead**>  
  
 <**tbody**>  
 <**c:forEach var="book" items="${**list**}"**>  
 <**tr**>  
 <**td**>**${**book.bookID**}**</**td**>  
 <**td**>**${**book.bookName**}**</**td**>  
 <**td**>**${**book.bookCounts**}**</**td**>  
 <**td**>**${**book.detail**}**</**td**>  
 <**td**>  
 <**a href="${**pageContext.request.contextPath**}/updateBook.jsp?id=${**book.bookID**}"**>修改</**a**>  
 **&nbsp;** | **&nbsp;** <**a href="${**pageContext.request.contextPath**}/book/deleteBook?id=${**book.bookID**}"**>删除</**a**>  
 </**td**>  
 </**tr**>  
 </**c:forEach**>  
 </**tbody**>  
 </**table**>  
 </**div**>  
 </**div**>  
</**div**>  
  
</**body**>  
</**html**>

**addBook.jsp**

*<%--  
 Created by IntelliJ IDEA.  
 User: 18109  
 Date: 2022/10/13  
 Time: 19:45  
 To change this template use File | Settings | File Templates.  
--%>*<%@ **page contentType**="**text/html;charset=UTF-8**" **language**="**java**" %>  
<**html**>  
<**head**>  
 <**title**>Title</**title**>  
 <**link href="https://cdn.staticfile.org/twitter-bootstrap/3.3.7/css/bootstrap.min.css" rel="stylesheet"**>  
</**head**>  
<**body**>  
  
<**div class="container"**>  
 <**div class="row clearfix"**>  
 <**div class="col-md-12 column"**>  
 <**div class="page-header"**>  
 <**h1**>  
 <**small**>新增书籍</**small**>  
 </**h1**>  
 </**div**>  
 </**div**>  
 </**div**>  
  
 <**form action="${**pageContext.request.contextPath**}/book/addBook" method="post"**>  
 <**div class="form-group"**>  
 <**label**>书籍名称</**label**>  
 <**input type="text" class="form-control" name="bookName" placeholder="书籍名称" required**>  
 </**div**>  
 <**div class="form-group"**>  
 <**label**>书籍数量</**label**>  
 <**input type="text" class="form-control" name="bookCounts" placeholder="书籍数量" required**>  
 </**div**>  
 <**div class="form-group"**>  
 <**label**>书籍描述</**label**>  
 <**input type="text" class="form-control" name="detail" placeholder="书籍描述" required**>  
 </**div**>  
 <**button type="submit" class="btn btn-default"**>添加</**button**>  
 </**form**>  
  
</**div**>  
  
</**body**>  
</**html**>

**updateBook.jsp**

<%@ **page import**="**com.dlct.dao.BookDao**" %>  
<%@ **page import**="**com.dlct.service.BookService**" %>  
<%@ **page import**="**com.dlct.pojo.Book**" %>  
<%@ **page import**="**java.sql.SQLException**" %>*<%--  
 Created by IntelliJ IDEA.  
 User: 18109  
 Date: 2022/10/13  
 Time: 20:03  
 To change this template use File | Settings | File Templates.  
--%>*<%@ **page contentType**="**text/html;charset=UTF-8**" **language**="**java**" %>  
<**html**>  
<**head**>  
 <**title**>修改书籍</**title**>  
 <**link href="https://cdn.staticfile.org/twitter-bootstrap/3.3.7/css/bootstrap.min.css" rel="stylesheet"**>  
</**head**>  
<**body**>  
**<%  
 int** id = Integer.*parseInt*(request.getParameter(**"id"**));  
 BookDao bookDao = **new** BookService();  
 Book book = **null**;  
 **try** {  
 book = bookDao.queryBookById(id);  
 } **catch** (SQLException e) {  
 **throw new** RuntimeException(e);  
 }  
 request.setAttribute(**"book"**, book);  
**%>**<**div class="container"**>  
 <**div class="row clearfix"**>  
 <**div class="col-md-12 column"**>  
 <**div class="page-header"**>  
 <**h1**>  
 <**small**>修改书籍</**small**>  
 </**h1**>  
 </**div**>  
 </**div**>  
 </**div**>  
  
 <**form action="${**pageContext.request.contextPath**}/book/updateBook" method="post"**>  
 <**div class="form-group"**>  
 <**label**>书籍编号</**label**>  
 <**input type="text" class="form-control" name="bookID" value="${**book.bookID**}" readonly**>  
 </**div**>  
 <**div class="form-group"**>  
 <**label**>书籍名称</**label**>  
 <**input type="text" class="form-control" name="bookName" value="${**book.bookName**}" required**>  
 </**div**>  
 <**div class="form-group"**>  
 <**label**>书籍数量</**label**>  
 <**input type="text" class="form-control" name="bookCounts" value="${**book.bookCounts**}" required**>  
 </**div**>  
 <**div class="form-group"**>  
 <**label**>书籍描述</**label**>  
 <**input type="text" class="form-control" name="detail" value="${**book.detail**}" required**>  
 </**div**>  
 <**button type="submit" class="btn btn-default"**>确认修改</**button**>  
 </**form**>  
  
</**div**>  
  
</**body**>  
</**html**>

展示

主界面：



添加书籍：

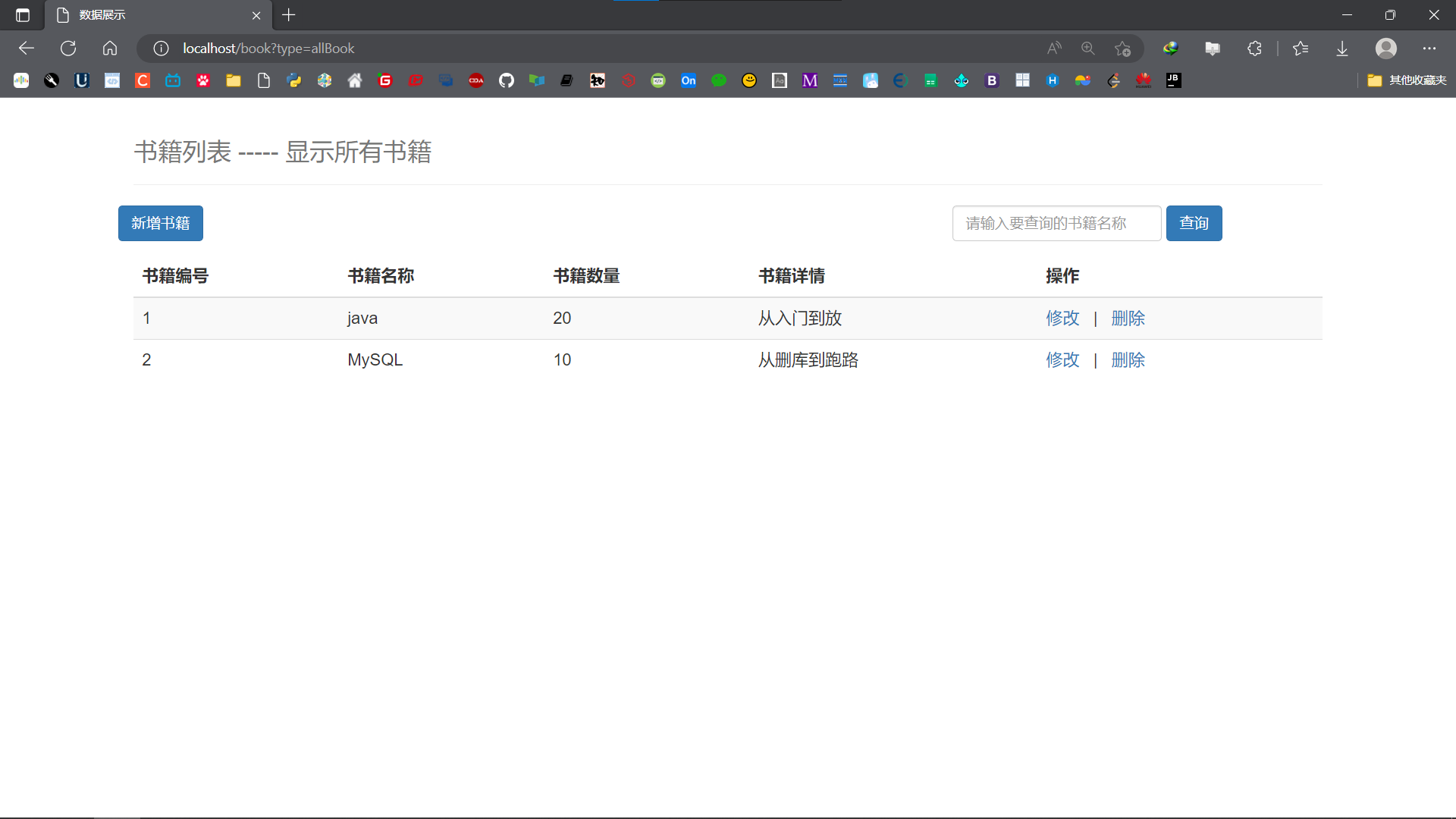




修改图书：



删除图书：



搜索图书：搜到展示书籍，未搜到展示所有书籍，并提示

