

## 实验 8 综合性实验

### 一、实验目的

- 1、掌握基于 MVC 的三层架构;
- 2、掌握 HTML+JSP+Servlet+JDBC 完成简单的 Web 应用。

### 二、实验题目

#### 1、简单的图书管理系统

通过数据源和 DAO 对象访问数据库。其中 JavaBeans 实现模型，访问数据库，Servlet 实现控制器，JSP 页面实现视图。

- 模型包括 2 个 JavaBean: BookBean 用于存放图书信息, BookDAO 用于访问数据库。
- 控制器包括 2 个 Servlet: BookQueryServlet 根据请求参数查询图书信息、BookInsertServlet 用来向数据库中插入数据。
- 视图包括 4 个 JSP 页面: bookQuery.jsp 显示查询页面、bookInsert.jsp 显示插入页面、display.jsp 显示查询结果页面和 errorPage.jsp 显示错误页面。

#### 2、题目自拟。

### 三、参考实现

实验 1 采用 MVC 设计模式，其主要代码如下：

【步骤 1】存放图书信息的 JavaBeans 代码 BookBean.java，它也是一个传输对象。

```
package com.beans;  
import java.io.*;
```

```
public class BookBean implements Serializable{  
    private String bookid = null;  
    private String title = null;  
    private String author = null;  
    private String publisher = null;  
    private float price = 0.0F;  
  
    public BookBean(){}  
    public BookBean(String bookId, String author,  
                    String title, String publisher, float price) {  
        this.bookid = bookId;  
        this.title = title;  
        this.author = author;  
        this.publisher = publisher;  
        this.price = price;  
    }  
    public String getBookid() { return this.bookid; }  
    public String getTitle() { return title; }  
    public String getAuthor() { return this.author; }  
    public float getPrice() { return price; }  
    public String getPublisher () { return publisher; }  
  
    public void setBookid(String bookid){ this.bookid=bookid; }  
    public void setTitle(String title){this.title=title; }
```

```

        public void setAuthor(String author){ this.author = author;    }
        public void setPrice(float price){this.price=price;    }
        public void setPublisher (String publisher){ this.publisher = publisher;}
    }

```

**【步骤 2】** 下面的 BookDAO 是一个简单的 JavaBeans，它实现数据库的访问。

```

package com.beans;
import java.sql.*;
import javax.sql.*;
import javax.naming.*;
import java.util.ArrayList;
import com.beans.BookBean;

public class BookDAO{
    private static InitialContext context= null;
    private DataSource dataSource = null;

    public BookDAO(){
        try{
            if(context == null){
                context = new InitialContext();
            }
            dataSource = (DataSource)context.lookup("java:comp/env/jdbc/bookDS");
        }catch(NamingException e2){
        }
    }
    // 根据书号查询图书信息
    public BookBean searchBook(String bookid){
        Connection conn = null;
        PreparedStatement pstmt = null;
        ResultSet rst = null;
        BookBean book = new BookBean();
        try{
            conn = dataSource.getConnection();
            pstmt = conn.prepareStatement("SELECT * FROM books WHERE bookid=?");
            pstmt.setString(1,bookid);
            rst = pstmt.executeQuery();
            if(rst.next()){
                book.setBookid(rst.getString("bookid"));
                book.setTitle(rst.getString("title"));
                book.setAuthor(rst.getString("author"));
                book.setPublisher(rst.getString("publisher"));
                book.setPrice(rst.getFloat("price"));
                return book;
            }else{

```

```

        return null;
    }
} catch(SQLException se){
    return null;
} finally{
    try{
        conn.close();
    } catch(SQLException se){
    }
}
}
}
// 插入一本图书记录
public boolean insertBook(BookBean book){
    Connection conn = null;
    PreparedStatement pstmt = null;
    try{
        conn = dataSource.getConnection();
        pstmt = conn.prepareStatement(
            "INSERT INTO books VALUES(?,?,?,?)");
        pstmt.setString(1,book.getBookid());
        pstmt.setString(2,book.getTitle());
        pstmt.setString(3,book.getAuthor());
        pstmt.setString(4,book.getPublisher());
        pstmt.setFloat(3,book.getPrice());

        pstmt.executeUpdate();
        pstmt.close();
        return true;
    } catch(SQLException se){
        return false;
    } finally{
        try{
            conn.close();
        } catch(SQLException se){ }
    }
}
}
}

```

【步骤 3】下面的 JSP 页面 bookQuery.jsp 实现根据书号查询图书信息

```

<%@ page contentType="text/html; charset=gb2312" %>
<html><head> <title>Book Query</title>
</head>
<body>
请输入一个书号:<br>
<form action="bookquery.do" method = "post">
<input type="text" name="bookid"><br>
<input type="submit" value="提交">

```

```

</form>
</body>
</html>

```

【步骤 4】下面的 JSP 页面 bookInsert.jsp 实现向数据库中插入数据

```

<%@ page contentType="text/html; charset=gb2312" %>
<html><head> <title>Book Insert</title>
</head>
<body>
<h3>请输入图书信息: </h3>
<% if(request.getAttribute("result")!=null)
    out.print(request.getAttribute("result"));
%>
<form action = "bookinsert.do" method = "post">
<table>
<tr><td>书号</td> <td><input type="text" name="bookid" ></td></tr>
<tr><td>书名</td><td><input type="text" name="title"></td></tr>
<tr><td>作者</td><td><input type="text" name="author" ></td></tr>
<tr><td>出版社</td><td><input type="text" name="publisher" ></td></tr>
<tr><td>单价</td><td><input type="text" name="price" ></td></tr>
<tr><td><input type="submit" value="确定" ></td>
<td><input type="reset" value="重置" ></td>
</tr>
</table>
</form>
</body></html>

```

【步骤 5】显示查询结果的 JSP 页面 display.jsp:

```

<%@ page contentType="text/html;charset=gb2312"%>
<jsp:useBean id="book" class="com.beans.BookBean" scope="request"/>
<html><body>
    书号: <jsp:getProperty name="book" property="bookid"/>
    书名: <jsp:getProperty name="book" property="title"/>
    作者: <jsp:getProperty name="book" property="author"/>
    出版社: <jsp:getProperty name="book" property="publisher"/>
    价格: <jsp:getProperty name="book" property="price"/>
</body></html>

```

【步骤 6】错误页面 errorPage.jsp 代码如下:

```

<%@ page contentType="text/html;charset=gb2312"%>
<html><body>
    对不起, 您查的图书不存在!
</body></html>

```

【步骤 7】下面的 Servlet 实现从请求参数获得书号, 然后从数据库中查找该书, 最后根据查询结果将请求转发到显示页面(display.jsp)或错误页面(errorPage.jsp)。

```

package com.control;

import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
import com.beans.BookBean;
import com.beans.BookDAO;

public class BookQueryServlet extends HttpServlet{
    public void doPost(HttpServletRequest request,HttpServletResponse response)

```

```

        throws ServletException,IOException{
String bookid = request.getParameter("bookid");
BookDAO bookdao = new BookDAO();
BookBean book = bookdao.searchBook(bookid);

if(book!=null){
    request.getSession().setAttribute("book", book);
    RequestDispatcher view = request.getRequestDispatcher("/display.jsp");
    view.forward(request, response);
} else {
    RequestDispatcher view = request.getRequestDispatcher("/errorPage.jsp");
    view.forward(request, response);
}
}
}

```

【步骤 8】下面的 Servlet 实现向数据库插入数据，并将控制请求的转发到 bookInsert.jsp 页面。

```

package com.control;

import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
import com.beans.BookBean;
import com.beans.BookDAO;

public class BookInsertServlet extends HttpServlet{
    public void doPost(HttpServletRequest request,HttpServletResponse response)
        throws ServletException,IOException{

        request.setCharacterEncoding("gb2312");
        String message = null;
        BookBean book = new BookBean(
            request.getParameter("bookid"),request.getParameter("title"),
            request.getParameter("author"),request.getParameter("publisher"),
            Float.parseFloat(request.getParameter("price")))
        );
        BookDAO bookdao = new BookDAO();
        boolean success = bookdao.insertBook(book);
        if(success){
            message = "成功插入一条记录！ ";
        } else {
            message = "插入记录错误！ ";
        }
        request.setAttribute("result",message);
        RequestDispatcher view = request.getRequestDispatcher("/bookInsert.jsp");
        view.forward(request, response);
    }
}

```

}

【步骤 9】在 DD 文件中部署 Servlet，将下列代码添加到 DD 文件的适当位置：

```
<servlet>
    <servlet-name>bookQuery</servlet-name>
    <servlet-class>com.control.BookQueryServlet</servlet-class>
</servlet>
<servlet>
    <servlet-name>bookInsert</servlet-name>
    <servlet-class>com.control.BookInsertServlet</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>bookQuery</servlet-name>
    <url-pattern>/bookquery.do</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>bookInsert</servlet-name>
    <url-pattern>/bookinsert.do</url-pattern>
</servlet-mapping>
```

#### 四、要求上交实验报告。