

第一阶段 注册页面的验证

第二阶段 完成登录和注册的功能

软件架构

饭店的架构

服务员 厨师 采购

软件的三层架构

表示层（表现层、web 层）

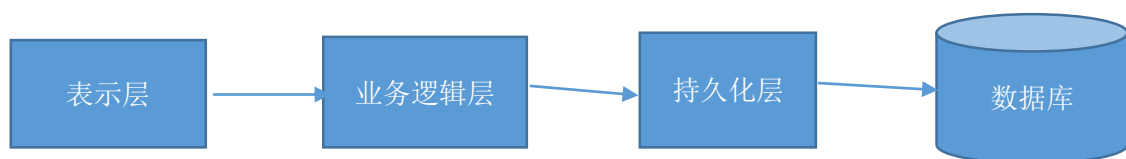
- 用户可以直接访问到的内容
- HTML 页面 Servlet

业务逻辑层

- 业务逻辑层负责处理网站的一些业务逻辑（登录、注册、买书、结账）
- Service 层

持久化层

- 负责操作数据库做增删改查等基本操作
- DAO 层



分层的目的就是解耦!!!

由于业务逻辑层和持久化层在开发过程中非常容易发生改变的,所以这两层我们采用一个面向接口的编程方式。

项目结构



开发步骤

1. 创建一个新的动态 web 工程。
2. 创建数据库 bookstore_0816
3. 创建表 bs_user
4. 创建类 User
 - a) Integer id
 - b) String username
 - c) String password
 - d) String email
5. 导入第三方的 jar 包
 - a) c3p0-0.9.1.2.jar
 - b) commons-dbutils-1.3.jar
 - c) mysql-connector-java-5.1.7-bin.jar
6. 导入 c3p0 配置文件
7. 创建 JDBCUtils 工具类
 - a) Connection getConnection()
 - b) void releaseConnection(Connection conn)
8. 创建 BaseDao<T>
 - a) int update()
 - b) List<T> getBeanList()
 - c) T getbBean()

9. 创建 UserDao 接口及实现类
 - a) User getUserByUsernameAndPassword(User user)
 - b) int saveUser(User user)
10. 创建 UserService 接口及实现类
 - a) User login(User user)
 - b) boolean regist(User user)
11. 创建 LoginServlet 和 RegistServlet
12. 导入已经写好的页面
 - a) 在页面中加入 base 标签
 - b) 修改页面中的路径
 - c) 修改表单的 method 和 action 属性

第三阶段 项目优化（修改 html 为 jsp）

1. 将所有的 html 修改为 jsp

- a) 新建一个 jsp 文件，并将 html 页面复制进新页面（主要 page 指令要保留）
- b) 在 html 页面中加入一个 page 指令，然后修改文件扩展名为 jsp
- c) 提取出页面中重复的部分，在通过 include 指令将这些内容分别引入网页

2. 登录和注册页面的错误消息显示

- 登录或注册失败以后再页面中显示错误消息
- 登录
 - 登录失败时也就是用户名或密码输错了，我们需要在域中设置一个错误消息然后在页面显示出来。

3. 表单的回显

- 登录或注册失败以后，在表单中显示用户上次输入的信息

4. 项目优化

- 问题 1：目前我们的项目中是一个功能对应一个 Servlet，登录有 LoginServlet、注册有 RegistServlet、删除用户有 DelUserServlet、修改用户有 UpdateUserServlet、添加图书有 AddBookServlet。。。这样会导致我们的项目被淹没在 Servlet 海洋中。
- 问题 1 思路：一个 Servlet 可以处理多个请求，比如有一个 UserServlet 那么用户相

关的请求都可交给该 Servlet，比如登录对应一个 login 方法，注册对应一个 regist 方法。

■ 步骤：

- ◆ 1.先创建一个 UserServlet
- ◆ 2.在 Servlet 中创建两个方法，一个是 login 一个是 regist
- ◆ 3.当用户要登录时调用 login 注册时调用 regist
- ◆ 4.我们希望用户可以告诉 Servlet 要掉 login 还是调 regist
- ◆ 5.我们通过用户传递 method 属性来判断要调用的方法。

■ 新问题：

- ◆ 在这个 UserServlet 中每添加一个方法，就要添加一个与之对应 if else 语句，这样会导致 if else 很多。

■ 解决：

- ◆ 如果可以直接根据方法名，获取到方法，我们就可以直接去调用方法，而不需要再手动调用
- ◆ 通过反射动态的获取用户要调用的方法对象，然后在去调用这个方法，这样做的好处是，当在添加新的方法进 Servlet 时，不需要写 if else，用户只需传递相应的方法名即可调用方法。
- ◆ 注意：我们所定义的方法要和 doGet 和 doPost 方法结构一样，可以通过重写 doGet 方法然后改名的形式创建方法。

■ 新问题：

- ◆ 我们将反射这一系列代码编写到了当前 Servlet，那么我们每新建一个 Servlet 这一堆代码就得在写一遍，我们的代码不能复用。

◆ 解决：

- 创建一个 BaseServlet，将这些重复的代码在 BaseServlet 中编写，其他 Servlet 只需继承 BaseServlet 而不用再次编写重复的代码。

■ 注意：

- ◆ 当我们使用 get 请求时，get 请求会覆盖 action 中的请求参数
- ◆ 当使用 get 请求时，可以在表单中设置一个表单隐藏域，这个域的 name 属性是 method，值是要调用的方法

```
<input type="hidden" name="method" value="login" />
```

➤ 问题 2：在大部分的 Servlet 中都需要先获取用户发送的请求参数，然后将这些请求参数封装为对象，像这些操作都是一些比较简单，但是又比较繁琐（尤其是当我们需要做转型操作）

➤ 问题 2 解决：写一个通用工具类来替咱们做这件事（BeanUtils）。

➤ JavaBean

- 属性私有化，通过 getter 和 setter 方法对属性进行读取和设置
- 有一个无参的构造器
- JavaBean 的属性指的是 getter 方法中属性名，属性名指的 getter 或 setter 方法将 get 或 set 去掉，然后首字母小写。getName()属性名叫 name，如果有一个 getAge()属性名叫 age。

➤ 使用 BeanUtils 可以直接将一个 Map 转换一个 JavaBean 的对象。

- 我们想可不可以获取到请求参数的 map，然后将这个 map 直接转换为我们想要的对象。
- 我们通过 BeanUtils 的 populate 方法将请求参数的 map 直接封装为对象，但是这块有一个问题需要注意表单中 name 属性要和类中的属性名一样，否则将不能进行封装。

第四阶段 使用 EL 替换所有 jsp 表达式

- login.jsp
- regist.jsp
- base.jsp