



【编程新技术实务】实验二 HTML以及J2EE简单编程（Java Web工程 JSP+Servlet+Mysql）

原创

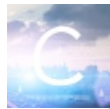
就算过了一载春秋  于 2019-11-21 01:30:03 发布  4115  收藏 17

分类专栏: [VV编程新技术实务](#) 文章标签: [Java](#) [Web](#) [Servlet](#) [JDBC](#) [Tomcat](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/qq_40889820/article/details/103127637

版权



[VV编程新技术实务](#) 专栏收录该内容

5 篇文章 1 订阅

订阅专栏

目录

实验目的

实验对应知识点

实验前任务

实验要求及步骤

一、Tomcat应用服务器安装配置

二、编写web页面, 完成对实验一数据库的操作

具体实现

项目主体结构

主要实现思想

效果展示

实现代码

实验感言

主要参考资料

实验目的

安装配置好Tomcat应用服务器, 使用Java进行JSP、Servlet的编写web应用并将其部署到Tomcat上

实验对应知识点

结合HTML进行JavaEE框架中的JSP、Servlet编程

实验前任务

学习Java的基本语法以及Java数据库编程

实验要求及步骤

一、Tomcat应用服务器安装配置

官网：<https://tomcat.apache.org>

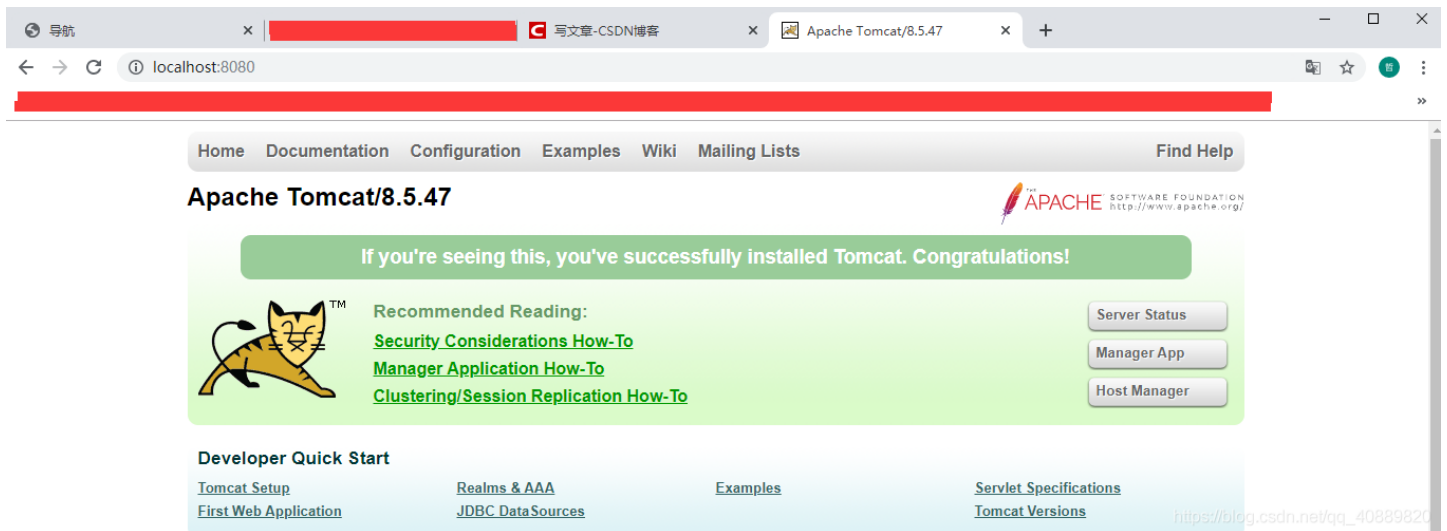
我下载的版本是8.5.47。

启动startup.bat效果如图：

```
Tomcat
18-Nov-2019 18:48:39.505 信息 [main] org.apache.tomcat.util.net.NioSelectorPool.getSharedSelector Using a shared selector for servlet write/read
18-Nov-2019 18:48:39.507 信息 [main] org.apache.catalina.startup.Catalina.load Initialization processed in 5167 ms
18-Nov-2019 18:48:39.651 信息 [main] org.apache.catalina.core.StandardService.startInternal Starting service [Catalina]
18-Nov-2019 18:48:39.652 信息 [main] org.apache.catalina.core.StandardEngine.startInternal Starting Servlet Engine: Apache Tomcat/8.5.47
18-Nov-2019 18:48:39.725 信息 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory 把web 应用程序部署到目录 [D:\Program Files\apache-tomcat-8.5.47\webapps\docs]
18-Nov-2019 18:48:40.274 信息 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [D:\Program Files\apache-tomcat-8.5.47\webapps\docs] has finished in [522] ms
18-Nov-2019 18:48:40.274 信息 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory 把web 应用程序部署到目录 [D:\Program Files\apache-tomcat-8.5.47\webapps\examples]
18-Nov-2019 18:48:41.895 信息 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [D:\Program Files\apache-tomcat-8.5.47\webapps\examples] has finished in [1,621] ms
18-Nov-2019 18:48:41.896 信息 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory 把web 应用程序部署到目录 [D:\Program Files\apache-tomcat-8.5.47\webapps\host-manager]
18-Nov-2019 18:48:41.994 信息 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [D:\Program Files\apache-tomcat-8.5.47\webapps\host-manager] has finished in [98] ms
18-Nov-2019 18:48:41.994 信息 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory 把web 应用程序部署到目录 [D:\Program Files\apache-tomcat-8.5.47\webapps\manager]
18-Nov-2019 18:48:42.041 信息 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [D:\Program Files\apache-tomcat-8.5.47\webapps\manager] has finished in [47] ms
18-Nov-2019 18:48:42.042 信息 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory 把web 应用程序部署到目录 [D:\Program Files\apache-tomcat-8.5.47\webapps\ROOT]
18-Nov-2019 18:48:42.105 信息 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [D:\Program Files\apache-tomcat-8.5.47\webapps\ROOT] has finished in [63] ms
18-Nov-2019 18:48:42.112 信息 [main] org.apache.coyote.AbstractProtocol.start 开始协议处理句柄["http-nio-8080"]
18-Nov-2019 18:48:42.130 信息 [main] org.apache.coyote.AbstractProtocol.start 开始协议处理句柄["ajp-nio-8009"]
18-Nov-2019 18:48:42.136 信息 [main] org.apache.catalina.startup.Catalina.start Server startup in 2627 ms
```

若出现乱码问题，将conf/logging.properties中java.util.logging.ConsoleHandler.encoding的值改为GBK即可。

在localhost:8080页面可看到如下界面：



在Idea中配置好Tomcat环境，略。

二、编写web页面，完成对实验一数据库的操作

1、编写页面如图2.6所示

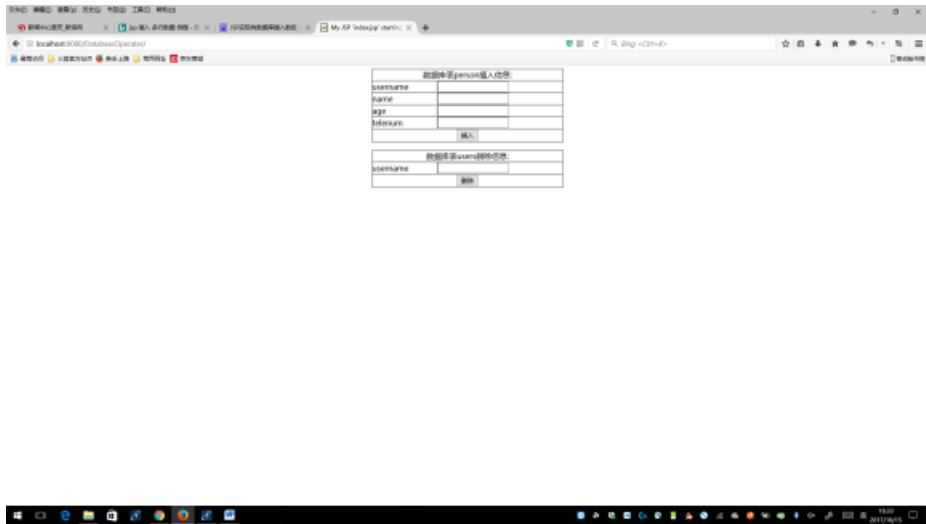


图2.6 数据库操作页面图

2、分别对应于2个表单编写后台处理逻辑（可以用JSP或Servlet进行处理），然后返回一个页面处理结果的页面，告知用户成功完成什么操作（如是进行了删除操作则信息相应改变），页面内有链接进行数据库数据的查询，如图2.7所示。



图2.7 数据库操作页面图

3、点击“查看数据库数据”则跳转到如图2.8所示页面，将数据库数据在页面中呈现出来，页面内有返回数据库操作页面的链接，点击即可跳转到数据库操作页面。



数据库表users信息:

username	password
ly	123456
test	11111

数据库表person信息:

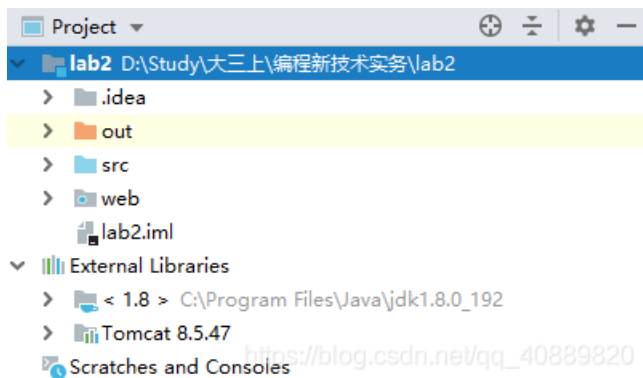
username	name	age	telenium
ly	雷力	15	
test	测试用户	25	13388009966

[返回数据库操作页面](#)

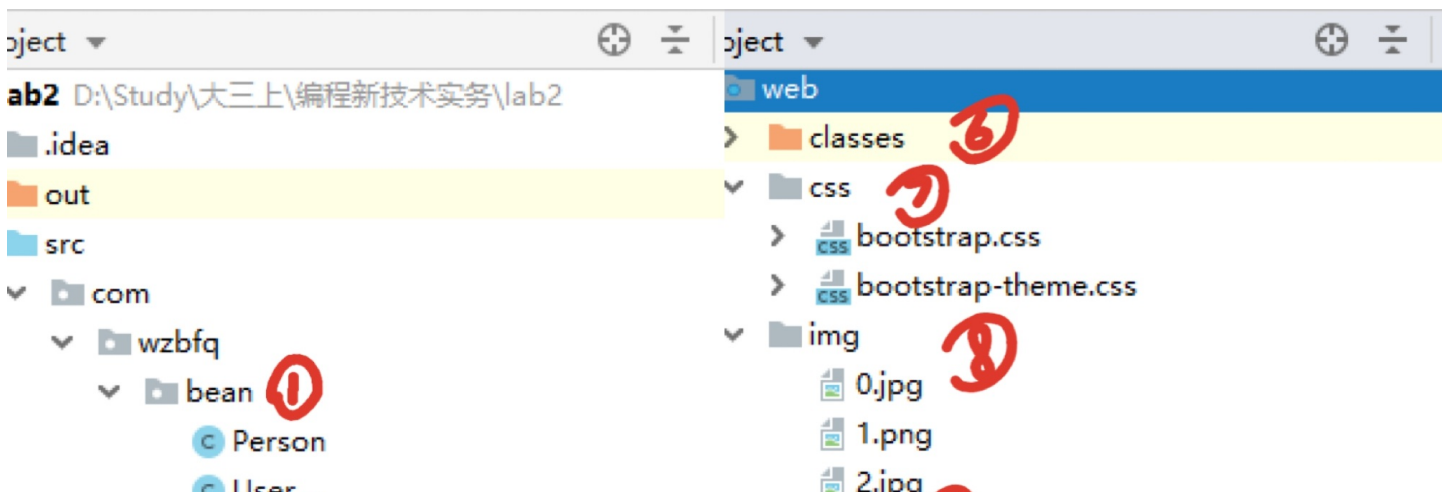
图2.8 数据库数据查询页面图

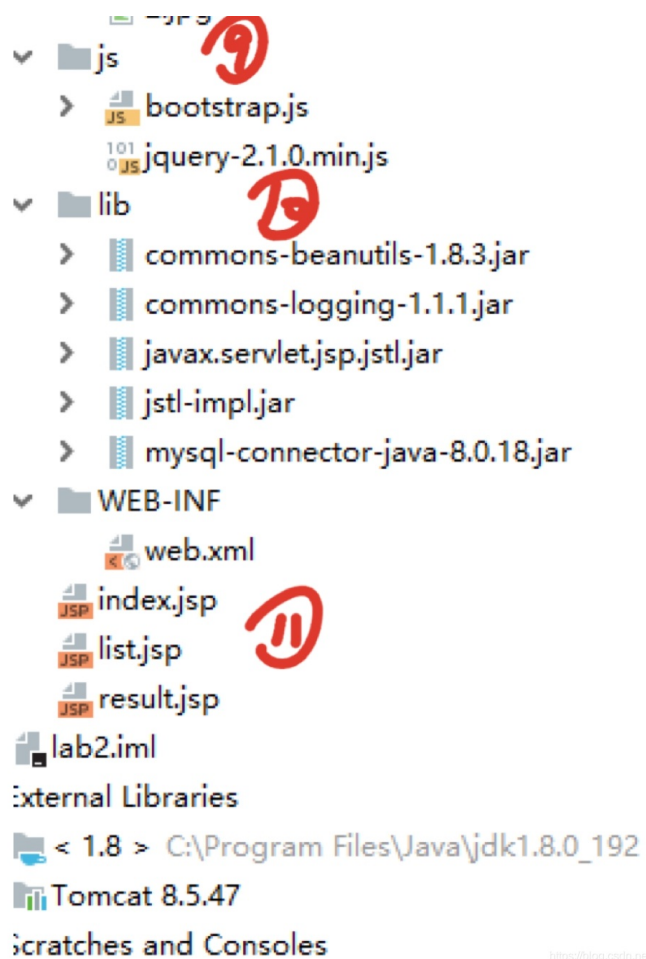
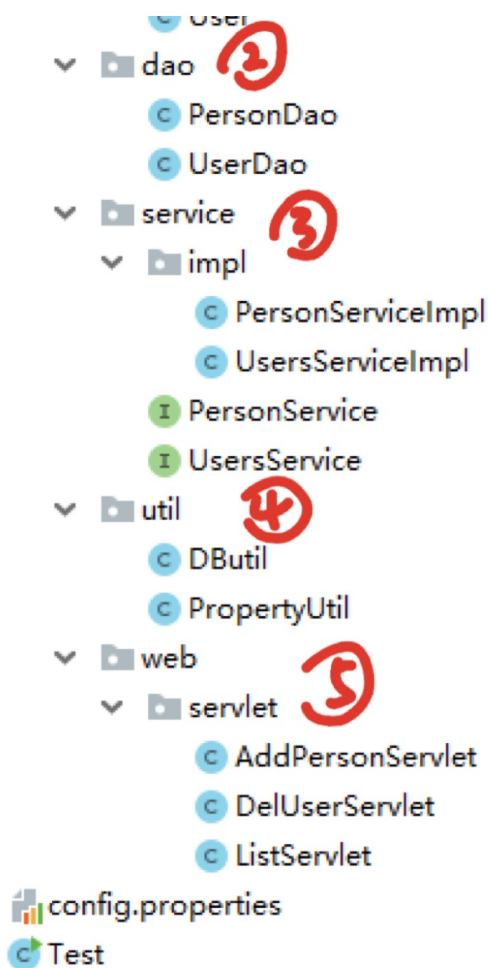
具体实现

项目主体结构



具体地:





https://blog.csdn.net/qz_40889820

- 1、bean层实现数据的封装，是数据库中数据的映射
- 2、dao层实现数据的持久化操作，如增删改查
- 3、service层是业务逻辑的实现
- 4、util层是工具，这里只用了读取配置文件的和连接数据库的
- 5、web层实现前后台的控制
- 6、classes文件夹存的是编译后的class文件
- 7、css
- 8、要用到的图片
- 9、js
- 10、lib文件夹存的是第三方包，这里的5个包中，第1、2个是和beanUtils有关的包，第3、4个是jstl标签库相关的包，第5个是jdbc驱动包。
- 11、jsp，建个jsp文件夹貌似更好看
(css和js可以在bootstrap上找<https://v3.bootcss.com/>)

主要实现思想

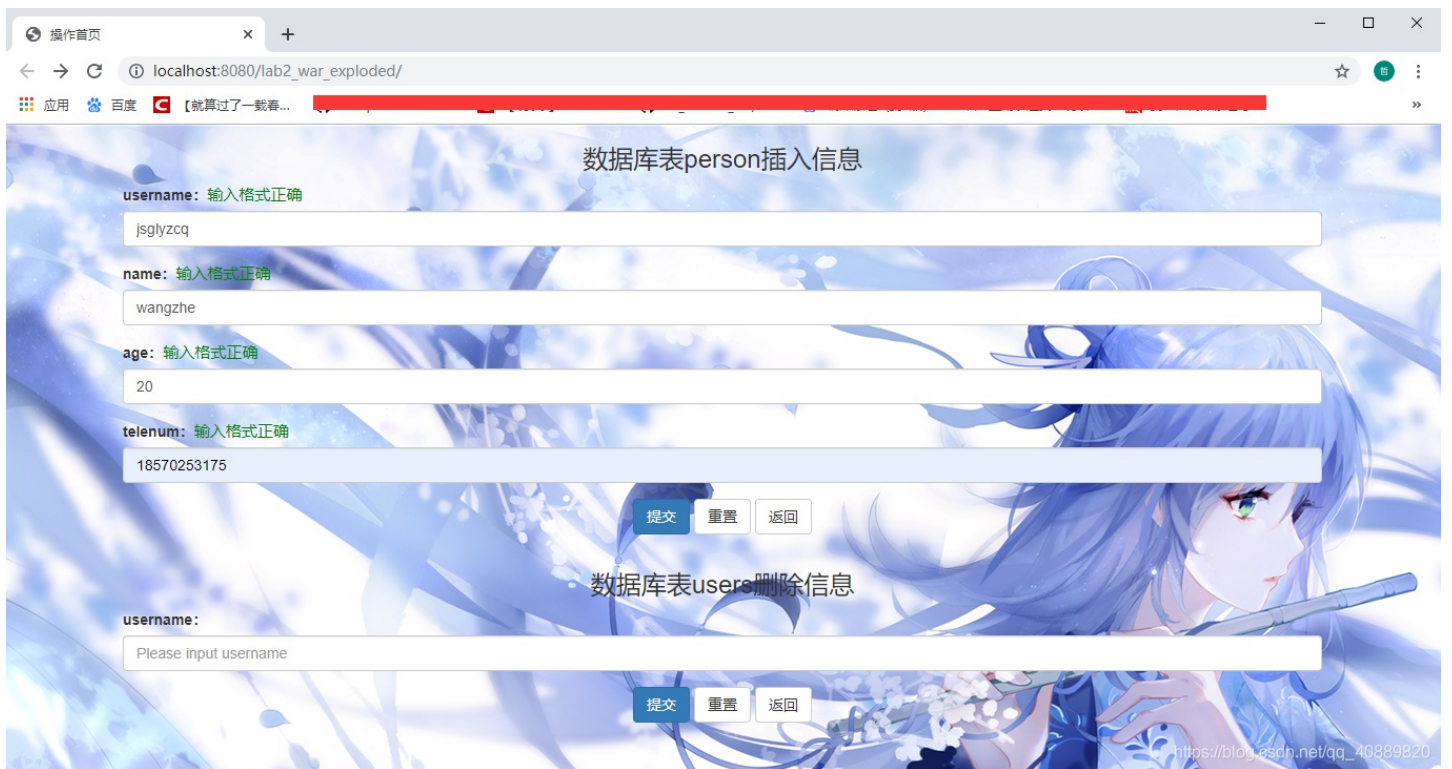
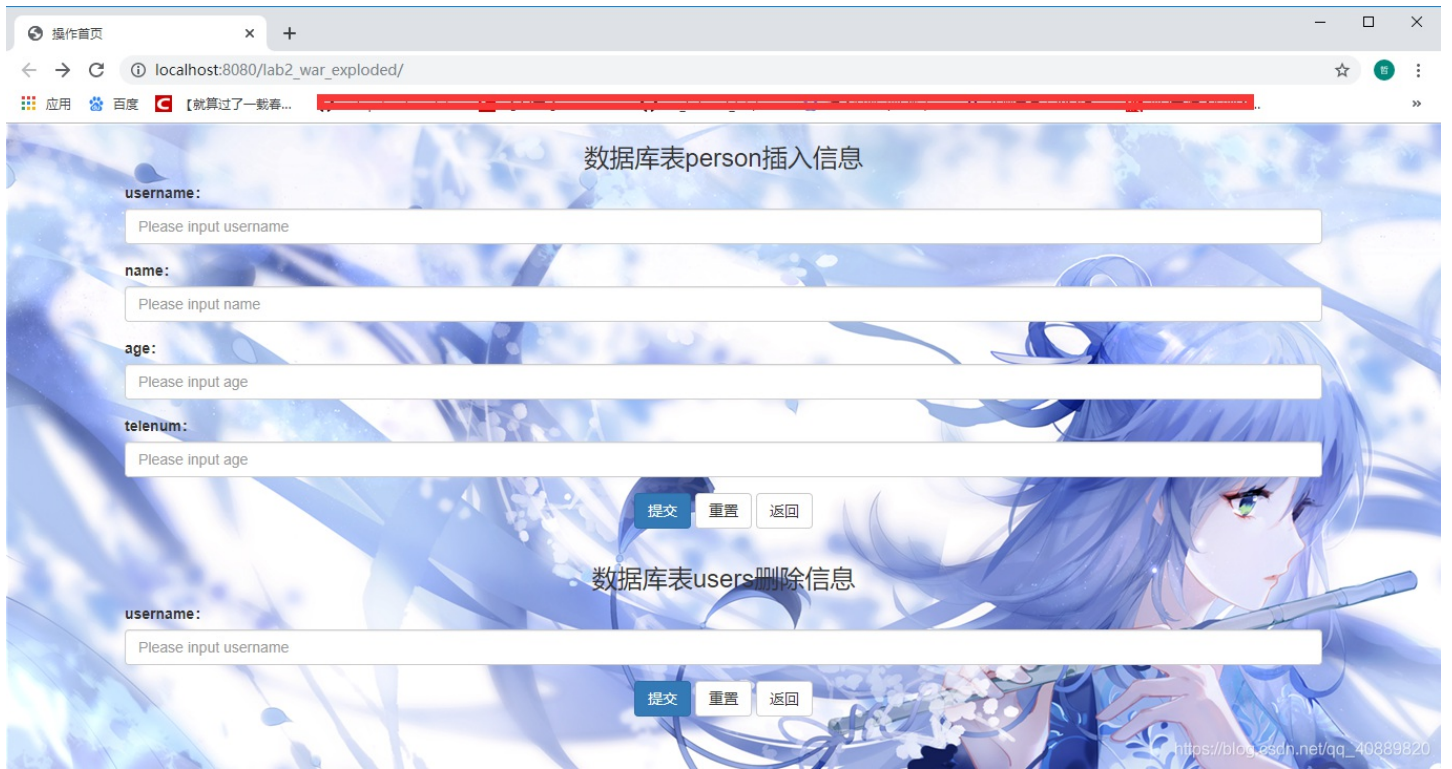
三层架构(3-tier architecture) 通常意义上的三层架构就是将整个业务应用划分为：界面层（User Interface layer）、业务逻辑层（Business Logic Layer）、数据访问层（Data access layer）。区分层次的目的即为了“高内聚低耦合”的思想。在软件体系架构设计中，分层式结构是最常见，也是最重要的一种结构。微软推荐的分层式结构一般分为三层，从下至上分别为：数据访问层、业务逻辑层（又或称为领域层）、表示层。

以上引用来自百度百科——三层架构

具体细节在此不赘述，仔细看看代码就会有更深的理解。

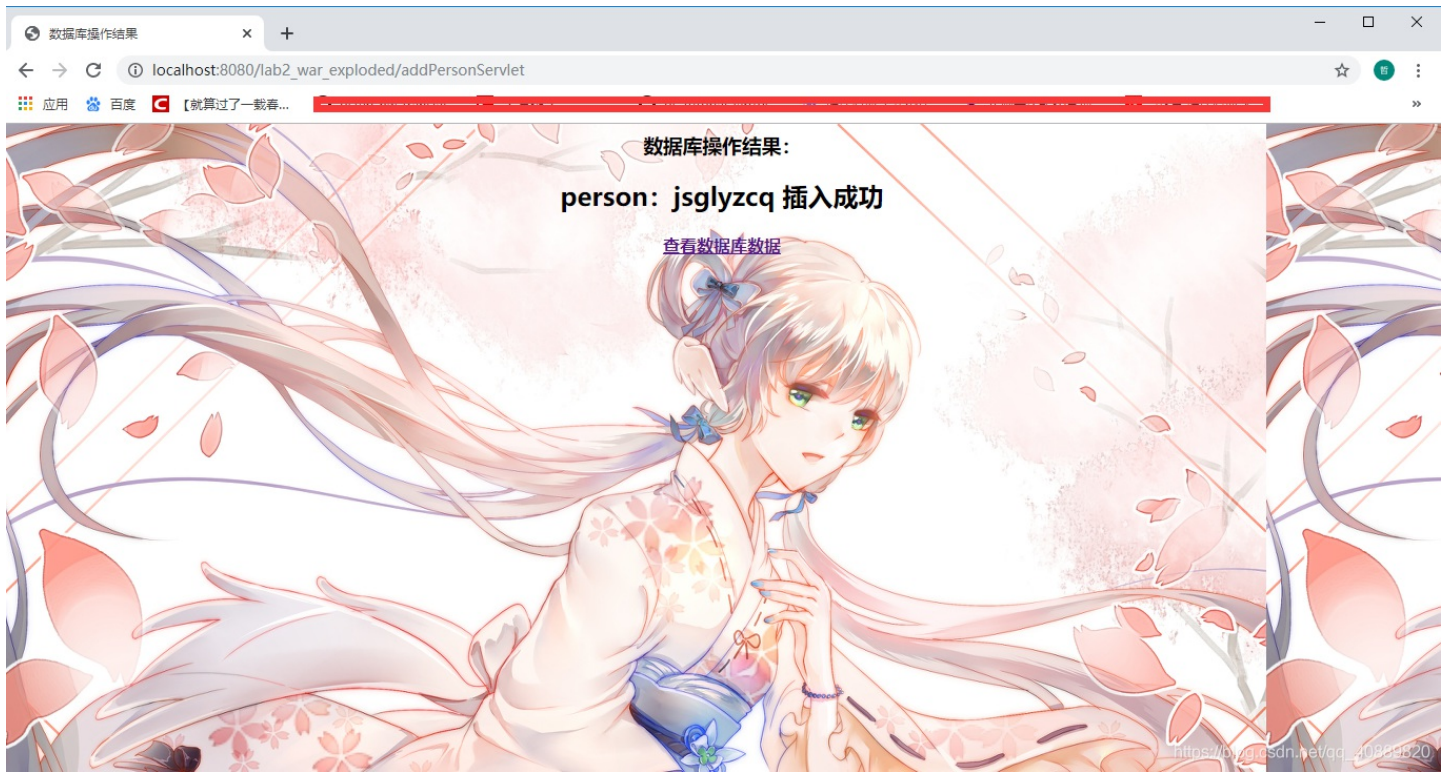
效果展示

1、



在数据提交至后台前，会对数据的合法性进行一个初步的检查。还提供了重置操作。返回操作可以忽略。

2、



提示操作的合法性

3、



(要不是避免数据看不清，才不用这个背景图)

(有些数据是完成整个项目之前插入的，所以可能会有若干(一个，Person表中的一个电话号码)数据不符合要求)

实现代码

代码有点多，上传到了github。这里给出我做的该系列实验的github地址：[编程新技术实务](#)。相关细节以后可能会更新。

实验感言

这个实验前前后后花了三天时间了，可以说配环境、调bug就占了一半的时间，但是调bug是个很好的学习过程。

感触有几点：

- 1、**重视编码方式**，jsp发送表单数据到servlet时编码方式不一样都有可能造成无法接收数据的情况。
 - 2、**重视路径**，很多问题的产生来源于路径设置不对，比如配环境的过程中。这里必须指出，WEB-INF/lib下的包最好也复制到Tomcat的lib目录下。
 - 3、写js这种前端类语言，有错误运行后又不提醒哪儿错，写的时候一定要细心。
- 做了这个简单的Java Web工程，对前后端有了进一步的了解，挺有意思的，比起算法竞赛不费脑一点（这就是我连续三天一直做这个实验的原因了吧）。

就不该找yg老师验，页面都没看就说行了，打了个A...

知识浅薄，若有错误、不足之处欢迎私信或留言。

主要参考资料

[1][CSDN——Javaweb实现增删改查操作操作](#)

[2][CSDN——Java Web idea中各个包所放类的作用说明](#)

[3][bilibili——JSP_增删改查分页小案例](#)