# 实验二. 使用JavaBean+Servlet+JSP实现MVC模式的留言板系统

## 实验目的：

1. 掌握在IntelliJ IDEA平台上开发Web项目的方法与过程；
2. 掌握在IntelliJ IDEA平台上开发Servlet组件的方法。
3. MVC模式的基本实现

## 二、 实验要求：

在IntelliJ IDEA平台上开发一个简单的Web项目，在项目中开发一个Servlet组件，并在使用它。

掌握用JavaBean+JSP+Servle实现MVC的基本方法。

## 三、 需求分析：

留言系统包括用户注册、用户登录，增加留言、查看自己留言的功能。

用户成功登录后，显示已有的留言，并允许添加新的留言，添加成功之后返回成功页面，从成功页面可以跳转到主页面也就是显示已有留言并允许添加新留言的页面。

为完成所述业务逻辑，需要一个用户表（userTable），字段包括：id、username和password。其中id设为自动增长的int型，并设为主键。username和password都设为varchar型。

还需要有一个留言表（lyTable）。字段包括：id、userId、date、title、content。其中id设为自动增长的int型，并设为主键。userId是user表中的id，在该表爪子外键，表明该条留言是该用户留的。

## 四、 实验步骤：

1. **准备数据库（使用MySQL）**

用户名和密码均为root

数据库liuyan

create database liuyan;

use liuyan;

create table userTable (

id int(11) primary key auto\_increment,

Username varchar (20) ,

Password varchar (20));

create table lyTable (

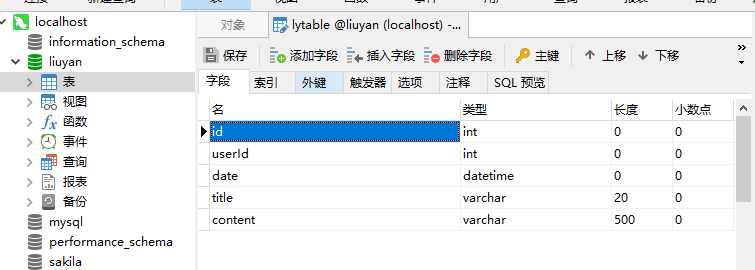
id int(11) primary key auto\_increment,

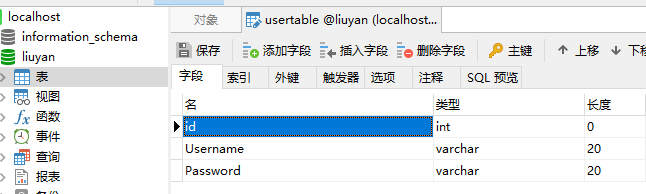
userId int(11) references userTable(id),

date datetime,

title varchar (20) ,

content varchar(500));





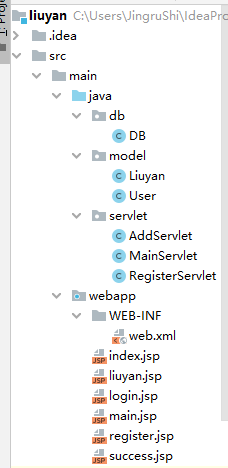
1. **新建工程**

从archetype创建Maven管理的web应用项目liuyan。pom.xml中添加的依赖包坐标为：



分析表是MySQL数据库的驱动包和Servlet的包。

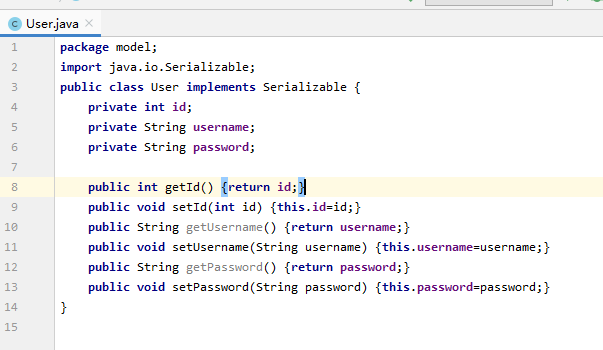
并创建项目的文件结构如下：

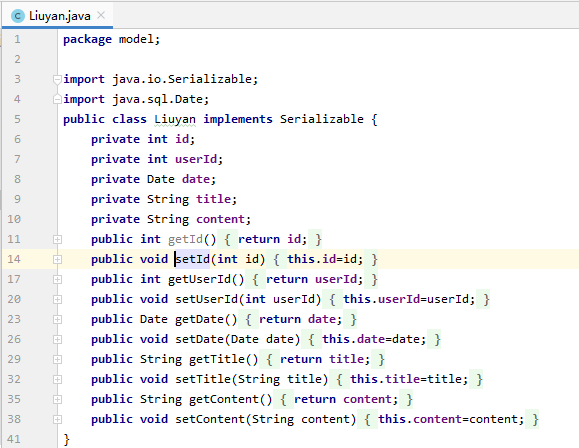


在db中存放数据库的操作代码，在model中存放数据库表的持久化类，在servlet中存放Servlet，所有JSP页面存放在webapp 中。

1. **编写代码**
   1. **开发Model**
2. 分别创建操作两张表的JavaBean

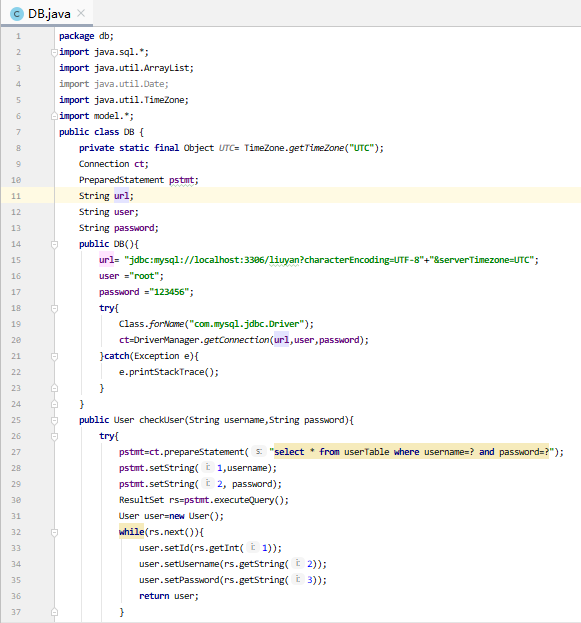
为了在下面的DB类中方便实现业务逻辑，这里先定义访问userTable表和lyTable中数据的JavaBean。



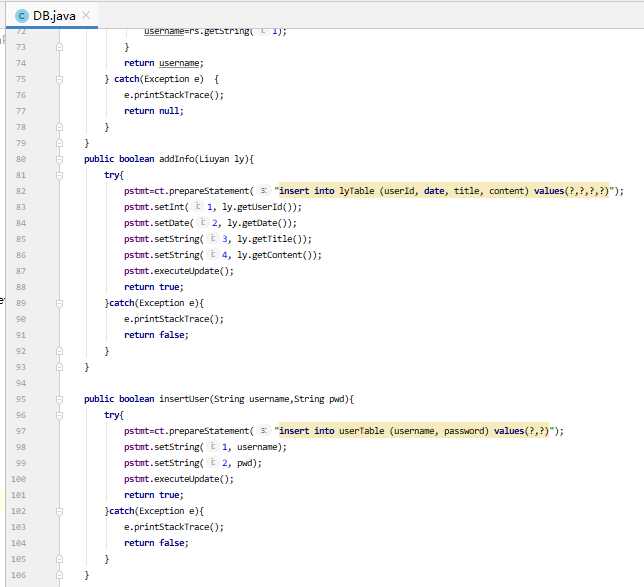


1. 访问数据库和实现业务逻辑的操作代码

数据库连接和各种数据操作的代码集中在以下的DB类中（结合SQL语句与JavaBean实现业务逻辑，暂时使用SQL语句查询数据，将查询结果转换成对象；或者反之，将对象中的数据用SQL语句插入到数据库）：



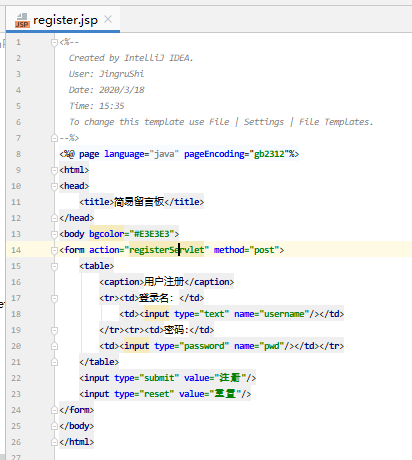




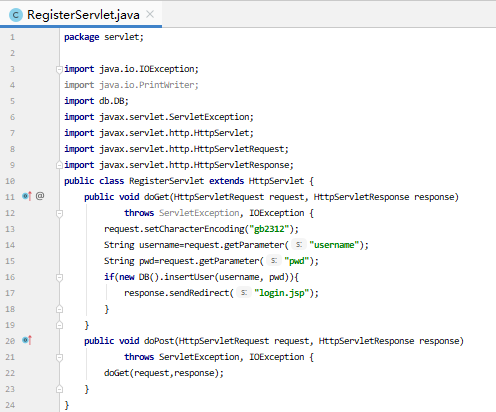
* 1. **开发View和Controller**

1. 创建注册页面和相应的控制器

先创建注册界面Register.jsp，其代码为：

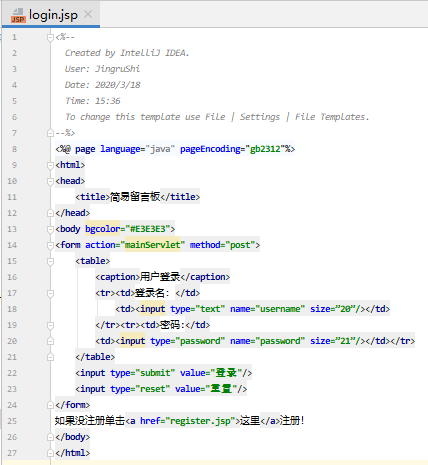


对应的控制类为：



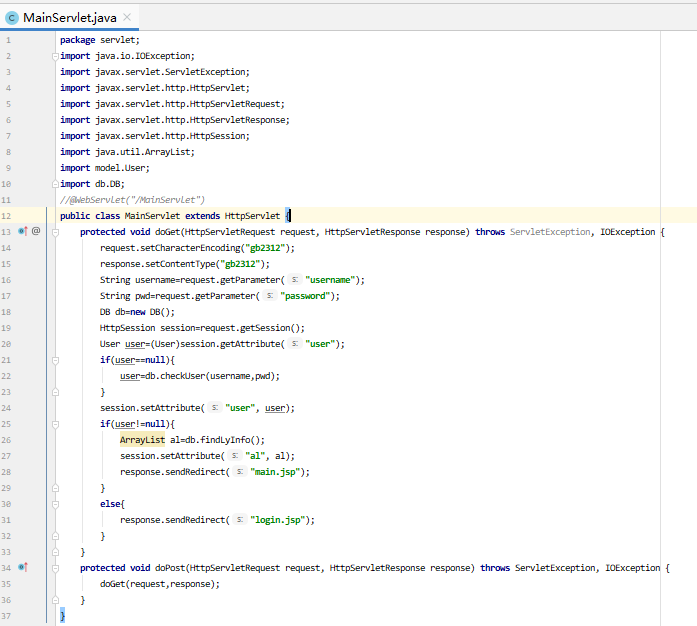
1. 创建登录页面和相应的控制器

登录界面login.jsp的代码为：



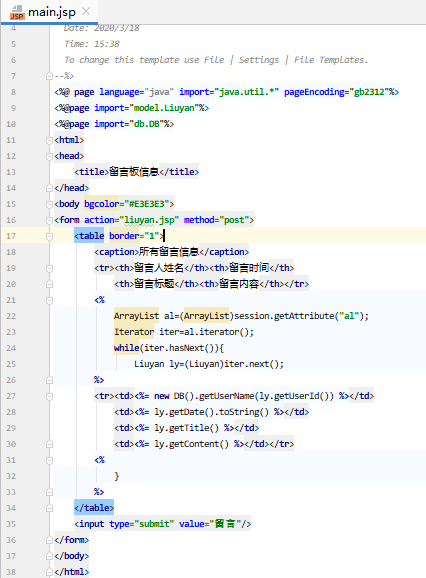
对应的控制类MainServlet类

当在登录页面中输入登录名和密码后，单击【登录】按钮，请求提交给了一个Servlet，其URL 为“mainServlet”，Servlet类命名为“MainServlet”，其代码：



1. 创建主页面main.jsp

MainServlet中验证成功后转到main.jsp页面，在main.jsp页面中显示所有留言信息，下面是main.jsp文件的代码：

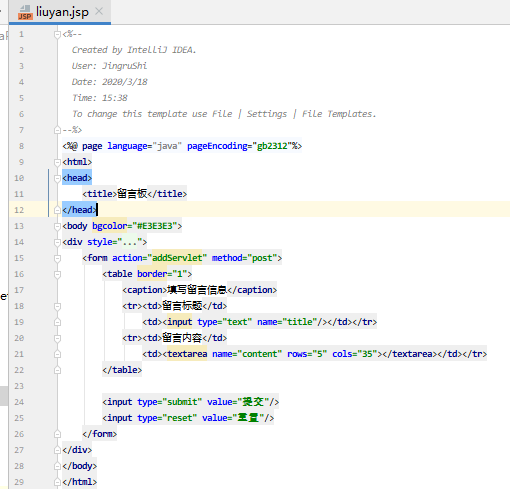


当按下“留言”按钮时，进入添加留言的页面。

1. 创建添加留言的页面和相应的控制器

建填写留言信息的页面liuyan.jsp

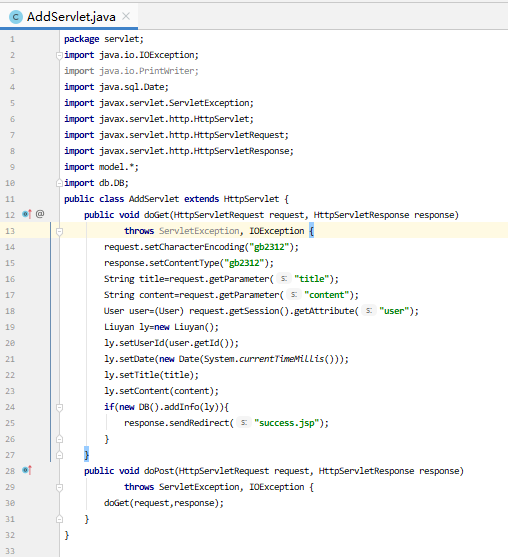
前用户信息已经保存在session中，所以留言时不用填写；留言时间就应该是当时的时间，也可以直接获得，所以用户需要填写的就是留言的标题及内容。其代码为：



创建添加留言的控制器

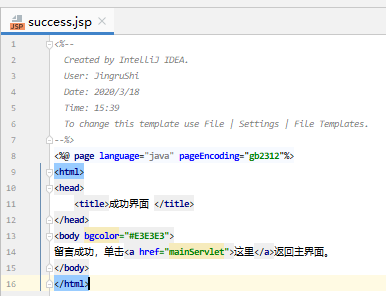
当用户填写完留言并提交时，由该Servlet控制用户的提交，处理用户提交的留言，这个servlet为AddServlet，其代码为：

在这个Servlet类中，调用了DB类的插入留言信息方法。



1. 创建成功页面

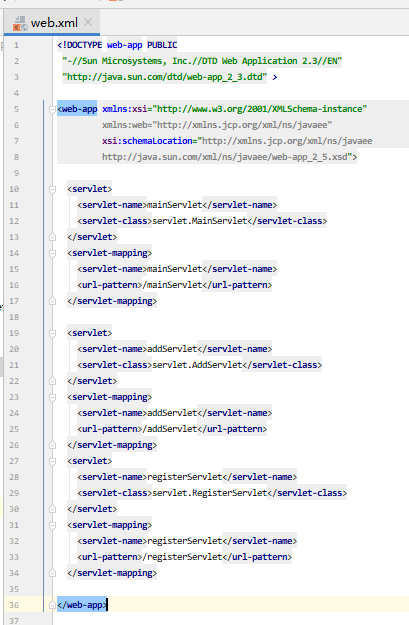
AddServlet控制器中，成功添加留言后转到了一个告诉用户成功的success.jsp页面，该页面代码为：



页面中有一个超链接，该链接是跳转到mainServlet，这里应该是第二次访问这个Servlet了，所以在该Servlet中可以从session取到当前用户的对象，不用再一次查询数据库，而只是把留言表中的信息查出保存，然后跳转到main.jsp。

1. **配置Servlet类**

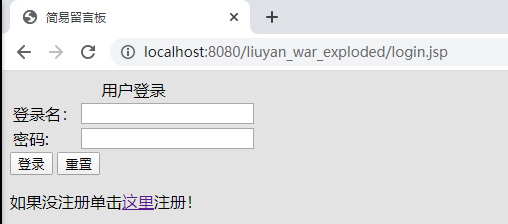
配置所有的Servlet类，这里集中在配置描述符web.xml中配置



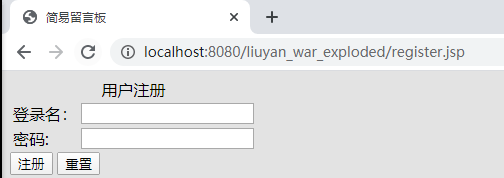
到此按前面设计的需求，已完成一个留言系统的基本功能。每一个功能都包括模型、视图和控制器，视图总是由JSP来完成，Servlet来实现控制器功能，访问数据库和业务逻辑用JavaBean实现，有些接单的业务逻辑直接包含在控制器中了而没有用单独的JavaBean来实现。

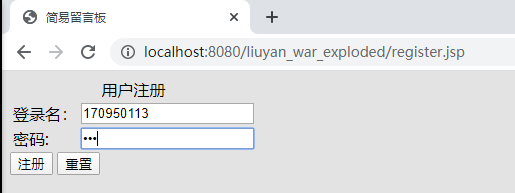
## 五、 运行效果：

1. 登录界面

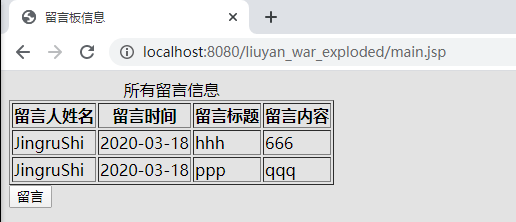


1. 注册界面

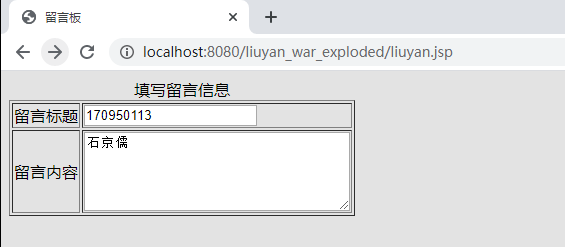


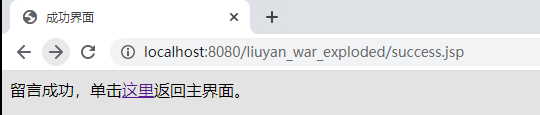


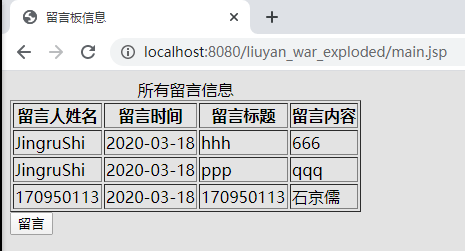
1. 登录成功后的留言板界面



1. 留言成功后的留言板界面







1. 上传到Github仓库链接

<https://github.com/TTopoo/Message-Board>