

Lab 9

Part 1: 配置服务器与数据库

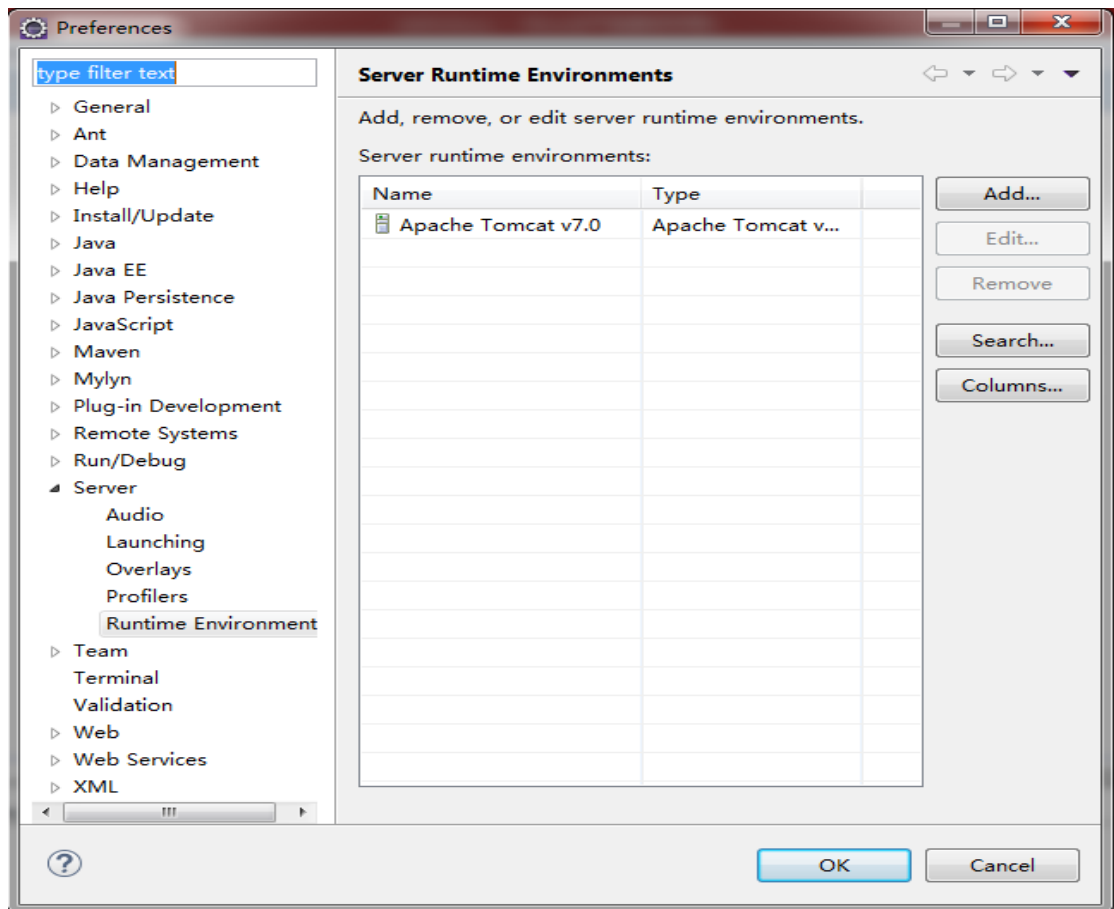
这一步只是 TA 推荐的一些工具，如果之前有开发经验比较熟悉其他的服务器，数据库可以跳转到 part2。另外，lab 不强制使用 java 完成，使用其他语言都可以。

=====分割线=====

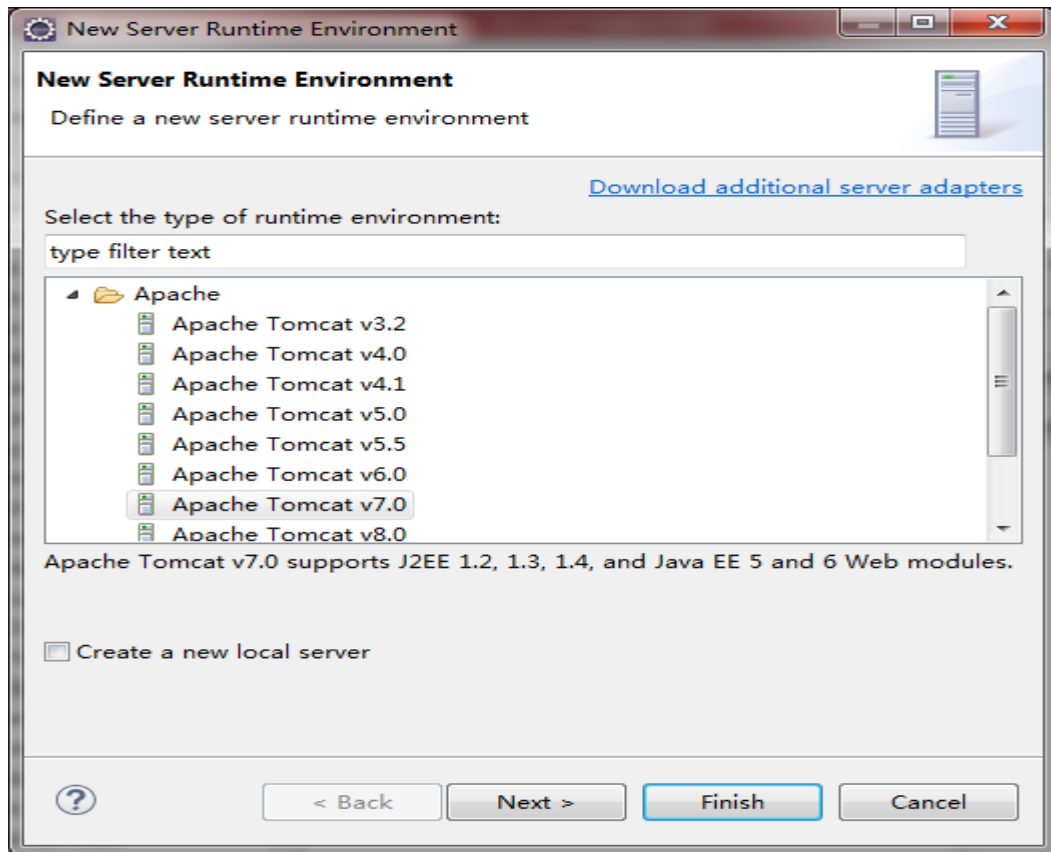
从今天的 lab 开始，你们将从单纯的 web 前端开发过度到 web 后端开发。首先，我们提供一些开发工具：Eclipse for J2EE，Mysql 以及 Tomcat。

首先，请先从 ftp 中下载 Eclipse for J2EE，这个 Eclipse 中有开发 Web 的插件，可以方便同学们进行 web 中各类文件。下载完成后，我们打开 Eclipse，先把我们第一个 lab 课上下载的 Tomcat 配置到 Eclipse 上面，这样可以直接从 Eclipse 里面控制 tomcat，Eclipse 也会自动帮你把项目部署到 tomcat 的目录下，方便开发。配置步骤如下：

1. 点击导航栏中 windows -> Preference 效果如下：

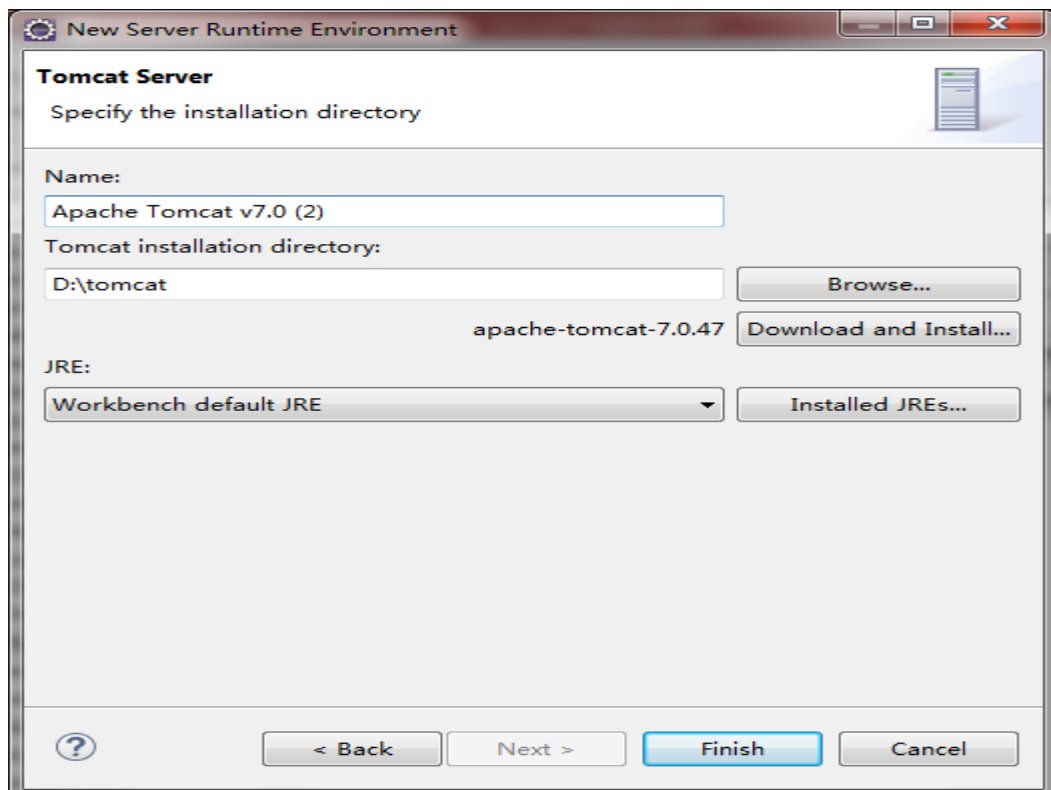


2. 找到 server -> Runtime Environment 也就是上面图片的样子，点击右边的 add 按钮。出现下面这个界面：



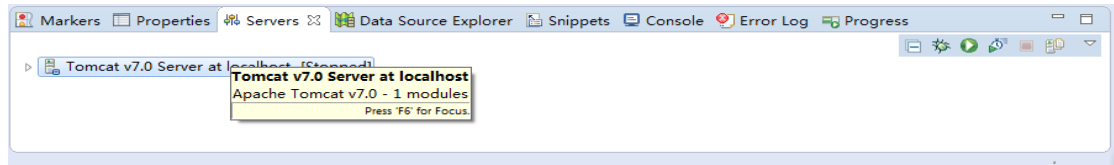
3.

4. 选中 Apache 选择你的 tomcat 的版本，如果是上次 lab 课装好的话是 tomcat v7.0 点击 Next 出现下面这个界面：



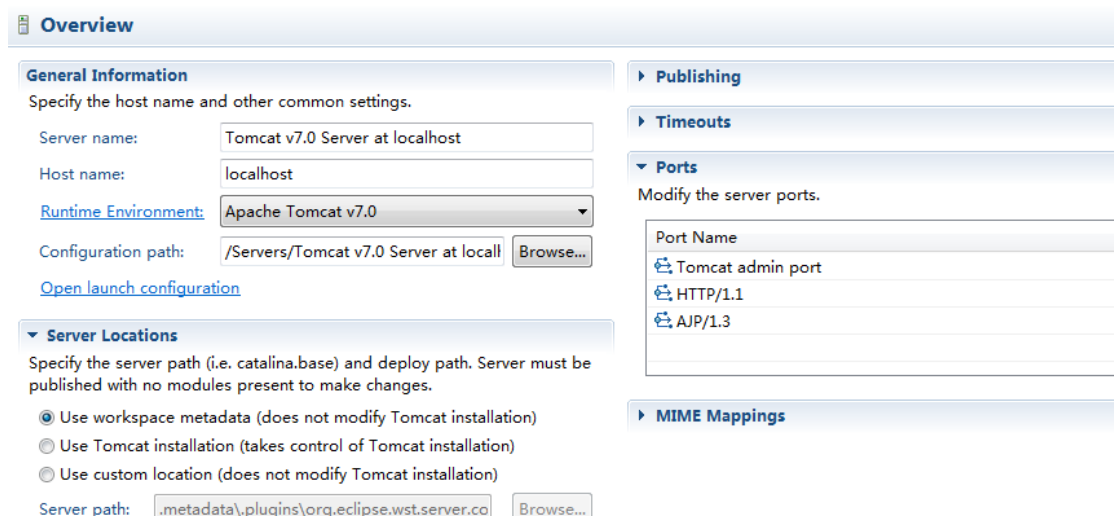
5.

6. 上面的 Name 随便输入一个, 下面的路劲只需要找到你 tomcat 的路径, 完成后点击 finish。Tomcat 就配置好了。这个时候我们回到主界面, 在主界面下边应该会有一个 Servers 的框, 如下所示:

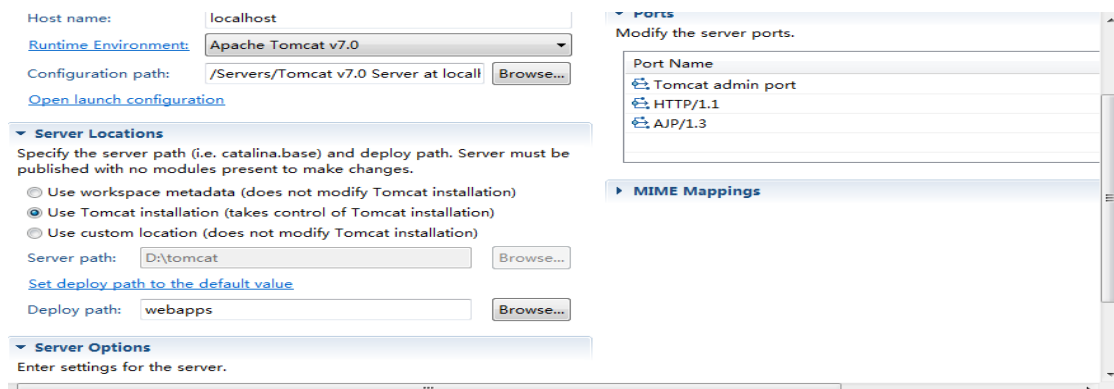


如果没有的话, 可以点击 windows -> show view-> other->Servers 把这个 view 调出来。

7. 我这上面是已经添加好 server 后的样子, 可以在 Servers 这个 view 中点击鼠标右键。New 一个 server, 选择你装好的 tomcat 版本, 而后的界面如上图。这个时候, 有些人可能还需要对这个 server 进行配置, 步骤如下: 双击上图的 Tomcat V7.0



8. 出现上面这个界面, 这个时候你需要在 Server Locations 下面选择第二个选项, Use Tomcat installation, 并且把 Deploy Path 改成 webapps。保存一下, 如下图。这样 Server 就配置好了。

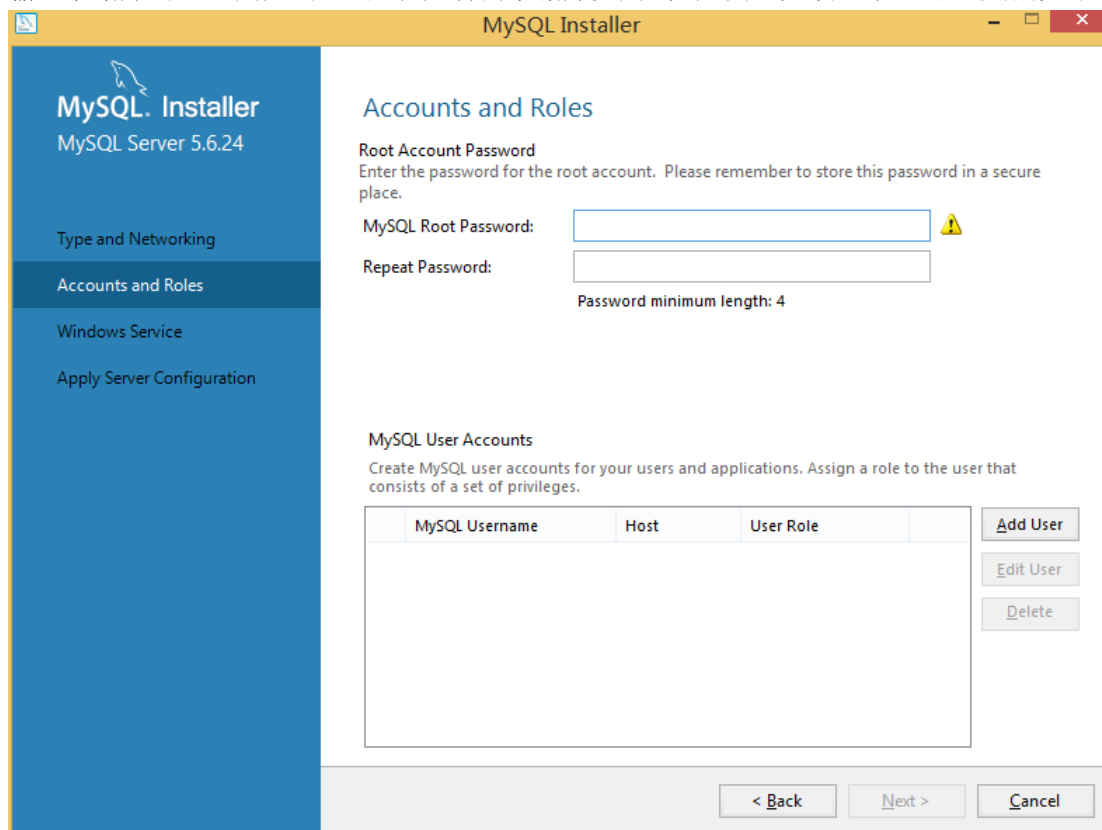


你可以启动一下 Server, 用浏览器进入 localhost:8080, 看是否出现 tomcat 的界面。如果成功可以继续往下进行, 如果不成功请联系助教。

第二步：下载 Mysql 数据库，一系列关于 Mysql 安装的介绍。

双击 lab 中 mysql-installer-community.msi 文件，跟着向导一直走，在 settype 页面就选择 Developer Default 吧！也可以就选择 server，这样的话你可以自己去下载一个类似 workbench 的数据库管理工具，推荐破解版的 navicat。Mysql 具体安装过程还是自行百度吧，装了一遍，删除太麻烦了要去注册表把所有和 mysql 相关的字段都删除掉，所以千万不要出错，=。=！

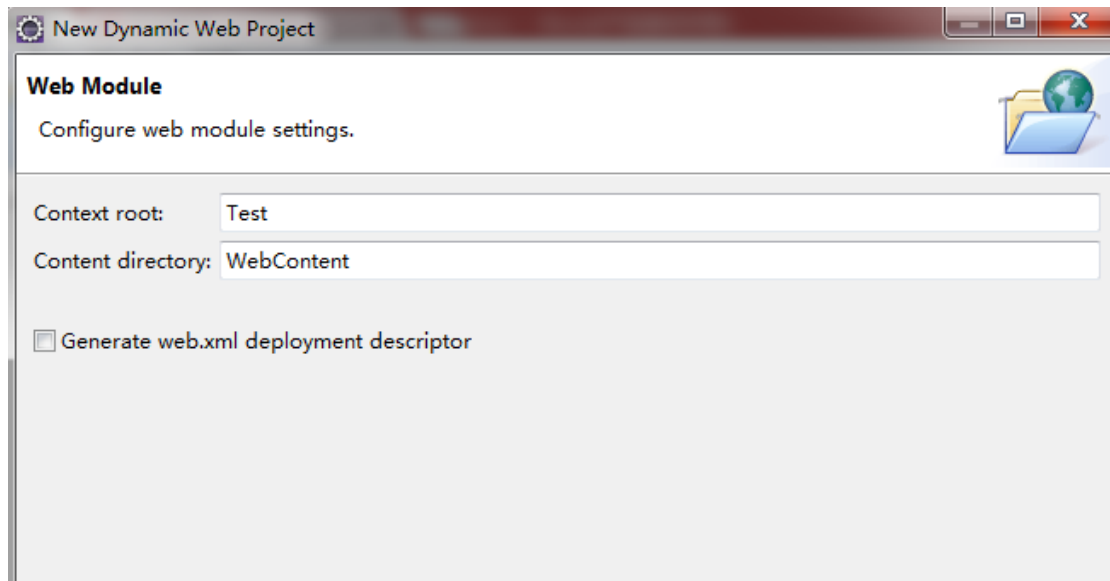
输入数据库的 root 用户的密码，以后访问数据库要用到，后面想要改这个密码还挺麻烦的：



点击 next，后一切按照默认配置就好。

第三步：在 Eclipse 上面开发 web 项目：

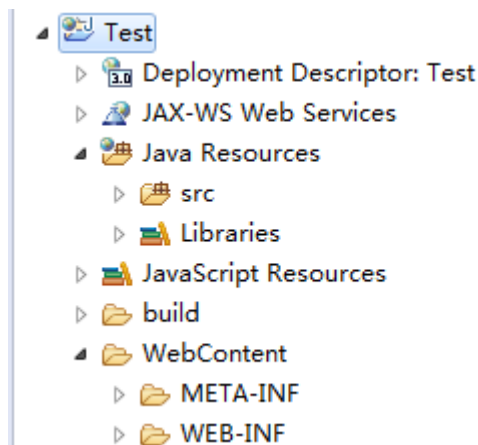
打开 ftp 上提供的 eclipse for javaEE，eclipse 大家都用过很长时间，比较熟悉，我们直接到主界面。在左边 Explorer 中右键，new->other->web->Dynamic Web Project，输入你的 projectName，一直 next 直到下面这个界面。



这里暂时不要勾选 Generate web.xml，如果勾选的话，需要你通过 web.xml 去配置 servlet 的监听，具体配置样式如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  id="WebApp_ID" version="2.5">
  <display-name>Lab9</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>
  <servlet>
    <description></description>
    <display-name>LoginServlet</display-name>
    <servlet-name>LoginServlet</servlet-name>
    <servlet-class>servlet.LoginServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>LoginServlet</servlet-name>
    <url-pattern>/login</url-pattern>
  </servlet-mapping>
</web-app>
```

为了操作方便，这里就不勾选。Servlet 的配置可以放置在后面在创建 servlet 的时候进行勾选（后面有详细介绍）。创建完成之后，展开创建好的工程可以看到如下目录：



我们的代码一般是放置在 Java Resources src 目录下，html, jsp 页面放置在 WebContent 下，然后在 WebContent 下创建 css, js, img 等存放相应文件。当我们要运行的时候就可以在我之前配置好的 server 下右键，选择 add and remove，将相应的工程加入，并且启动 tomcat，那么我们就可以通过浏览器 <http://localhost:8080/projectName/> 访问你该 project 的 WebContent 下面的页面。不过这次不需要你们创建，直接把 lab 下面的 ChatBoardLab 导入到你们 Eclipse 当中就可以了

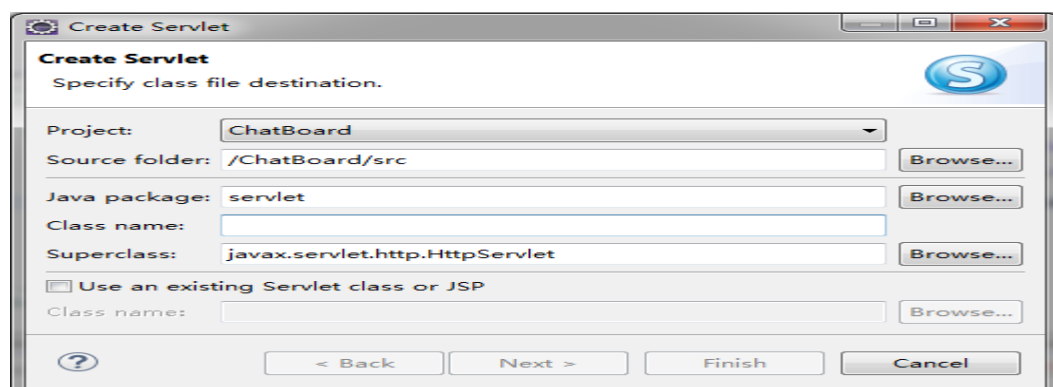
Part 2: 完成 Session 管理，前后台数据交互以及一些 Sql 操作

还记得 lab4 的那个单机版留言板吗，我们在学习到 web 后端以后可以利用 web 后端存储这样之前的留言。看看我们提供给你们文件。

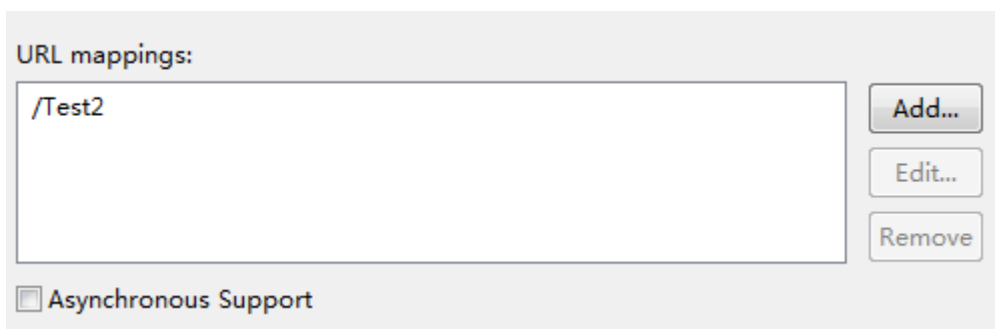
首先 data.sql 是助教这边统一的一个数据库，你需要在之前配置好的数据库中新建一个 note 的 schema，然后导入 data.sql 执行一下，会产生两张表：User 和 message（这一系列操作可在 workbench 等数据库管理工具中完成）。User 里面有 userID, username, password。第一个是主键，不过这里已经设置为自增的主键，所以存入的时候是不需要管它的。第二个表为 message，messageID 为主键，同样设置了自增长，userID 用来关联用户和信息，你可以通过 userID 定位到是哪个用户，得到用户名，message 字段为留言内容，最后的 note_time 字段为时间字段，你可以理解为 (yyyy-mm-dd hh:mm:ss) 这样一个时间段，你可以用 `SimpleDateFormat` 类进行 `string` 和 `date` 的转化。而后看另外一个 `ChatBoardLab` 这个文件夹，里面是助教实现的一个简单的登陆，有一些基本的数据库操作以及前后台数据交互的方式，你可以用作后面需要完成的任务的参考。其中注意一点，`java` 连接 `mysql` 数据库是需要相应的 `jar` 包的，在 `webcontent/WEB-INF/Lib` 下面有一个 `mysql-connector` 的 `jar` 包，以后你们需要连接 `mysql` 数据库的时候，也需要导入这个包，另外建议你们把开发中需要的其他 `jar` 包也放在这样一个目录下面。接下来介绍，我们这次的留言板需要实现哪样一些功能。

1. 你需要一个登陆界面，输入用户名和密码，如果用户名不存在则将这一套用户名和密码存入数据库，并且进入登录状态。如果用户名存在则需要验证用户的密码是否正确，如果正确则进入登入状态。这里需要用 session 存储你的登录状态。对于第一部分，你们不用担心，助教已经给出了相应的代码，主要有 `dao.Dao`, `servlet.LoginServlet` 以及一个 `login.jsp` 三个文件。特别强调一下，你们打开的数据库连接一定要记得关，比如 `Connection`, `Statement`, `ResultSet`，你们在 `finally` 中把连接关了。如果对这套流程还不熟悉的同学可以多看看，后面需要你们照着这个实现一些更为复杂的功能。这里特别介绍一下创建一个 `servlet`。

在相应的包点击鼠标右键就可以创建一个 `servlet`，点击后出现下面的界面：



Class Name 就是这个 servlet 的类名，填好点击 next。出现下面这个界面：



这里的 URL mappings 就是 mapping 我们前台 JSP 页面中 form 的 action，比如我们现在这个是 /Test2，那么他就接收到 action=“Test2”的表单提交。之后下一个页面让它默认设置就好。这个时候就会生成一个这样的界面：

```
*/
public Test() {
    super();
    // TODO Auto-generated constructor stub
}

/**
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
 */
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
}

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
}
```

在这里大家就可以通过 doPost 函数处理 post 请求。具体如何获取参数，以及保证中文不乱码，可以看 LoginServlet 的代码。

2. 接下来的功能就需要你们自行实现 •

1. 从登陆页面跳转到主界面后，要从数据库读出原来的所有发言。另外，为了体现 session 管理了登陆状态，如果你直接通过 url 访问你后面留言板的 jsp 页面是会自动跳转到 Login.jsp 上面的。
2. 你每次提交发言不是简单的通过 JS 代码放到上面留言板上，也需要你提交到后台，存入到数据库中。里面的表情，时间都要正确。另外，发言人的名称已经改成了登陆时候所使用的用户名了，不要再是一成不变的囧囧先生。
3. 我们原来的转发按钮已经改成了删除按钮，请实现删除功能，一定要删除数据库里面的值，而不是简单的删除 html 的元素。

简单的效果图：

[illegible]

这次助教提供的代码可以不用，你可以选择自己熟悉的框架，技术。
最好不要用 `nodejs` 和 `PHP` 以及 `JAVA` 之外的语言完成，不然我这里验证起来太麻烦，还要去配运行环境。

最好不要用 `nodejs` 和 `PHP` 以及 `JAVA` 之外的语言完成，不然我这里验证起来太麻烦，还要去配运行环境。

提示：1. 建议你们使用隐藏表单来完成这些功能。

2. 数据库中会用两个单引号来表示单引号，所以在存入单引号的时候需要用两个单引号替换一下

本次 Lab deadline: 5.17 (星期日)