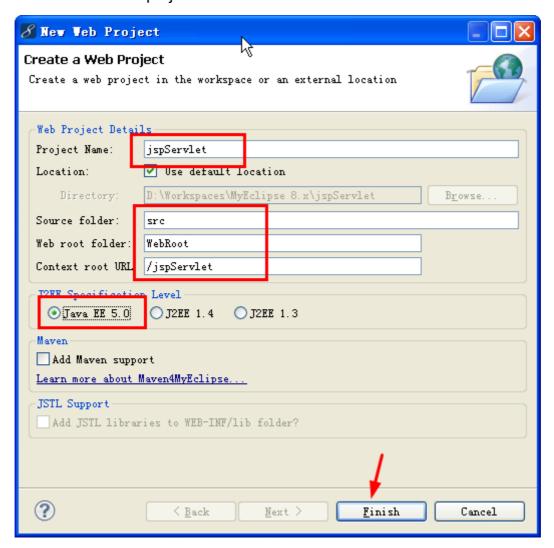
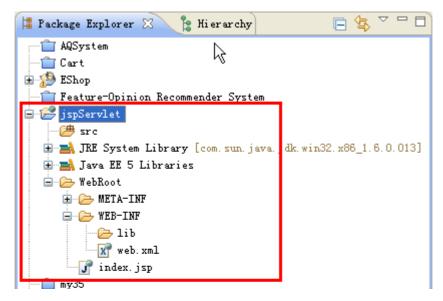
# 构建一个简单的 web 工程——不连接数据库

This web project is based on jsp and servlet. We will explain how to build a web project through a simple login example. A user log on with username and password, the system will welcome the user if input name is the same as information we set in the program; otherwise the system will return an error page.

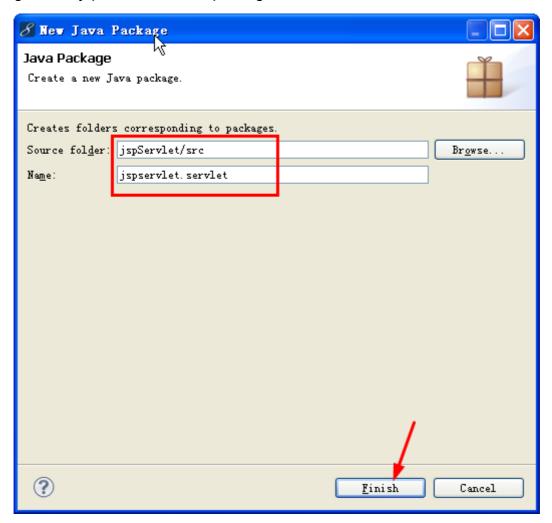
Create web project named 'jspServlet'.
 File->new->web project



Pay attention to the left *package explorer* window, if you see the following, go to the next step.



2. We need a reasonable package structure. Right click 'jspServlet'->new->package



The same way to create other packages

## jspservlet.dao, jspservlet.dao.impl

The packages will include interface and implementation about DataBase

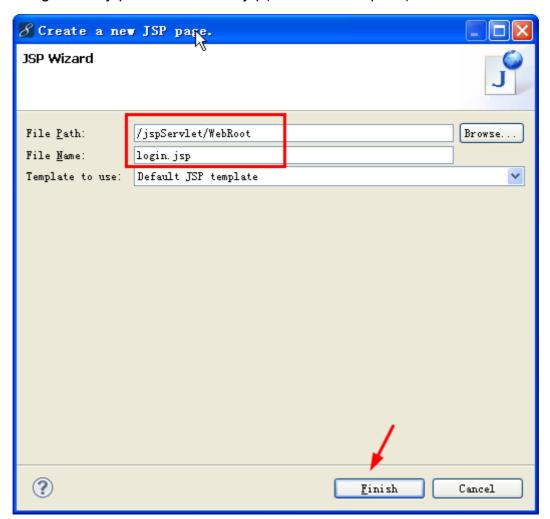
operation. First example does not connect to database, so some simple codes will replace codes that connect to database.

## jspservlet.vo

This package will include value object to encapsulate information.

3. Create login.jsp.

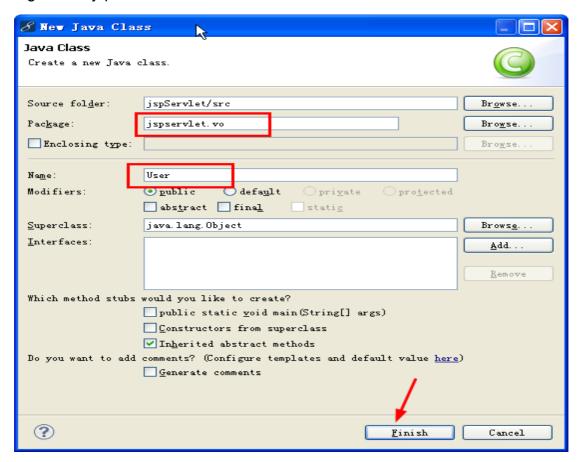
Right click 'jspServlet'->new-> jsp(advanced template)



## Edit login.jsp.

A web form contains two input item named 'username' and 'password', its request should be sent to an servlet whose path is 'login' through 'post' method.

4. Create a value object to encapsulate information. Right click 'jspServlet'->new->class.

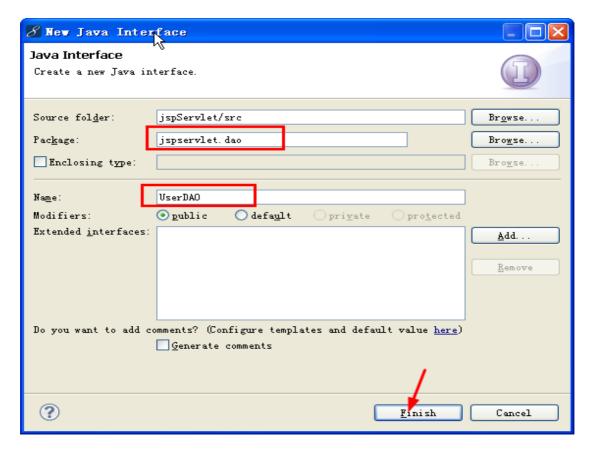


This object should include two private properties 'username' and 'password' and also the getter sand setters methods for them.

```
publit(jspServlet/src/jspservlet/vo/User.java)
private String username;
private String password;
public String getUsername() {
    return username;
}
public void setUsername(String username) {
    this.username = username;
}
public String getPassword() {
    return password;
}
public void setPassword(String password) {
    this.password = password;
}
```

5. Now we need some logic to process information user submits. Create UserDAO and UserdaoImpl.

Right click 'jspServlet'->new->interface.



Add a line of code like following:

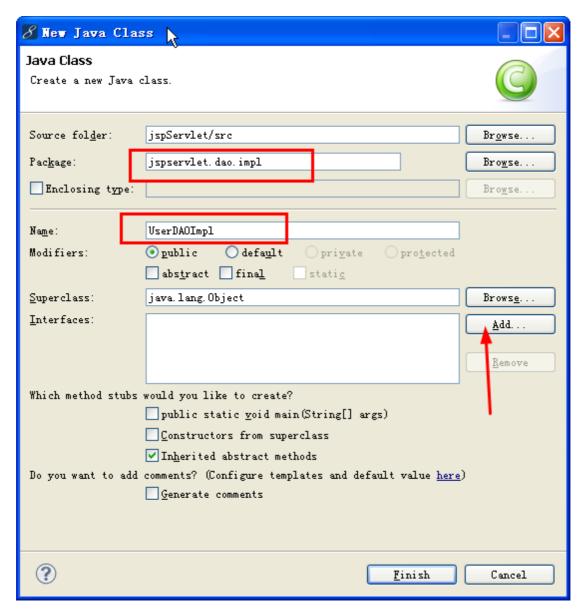
```
package jspservlet.dao;

import jspservlet.vo.User;

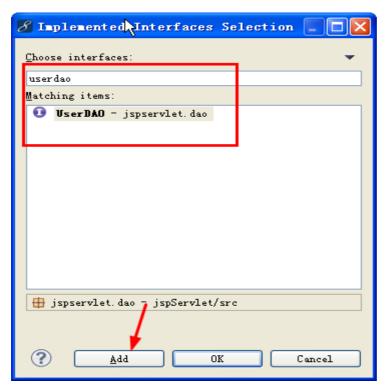
public interface UserDAO {

   public int queryByUsername(User user) throws Exception;
}
```

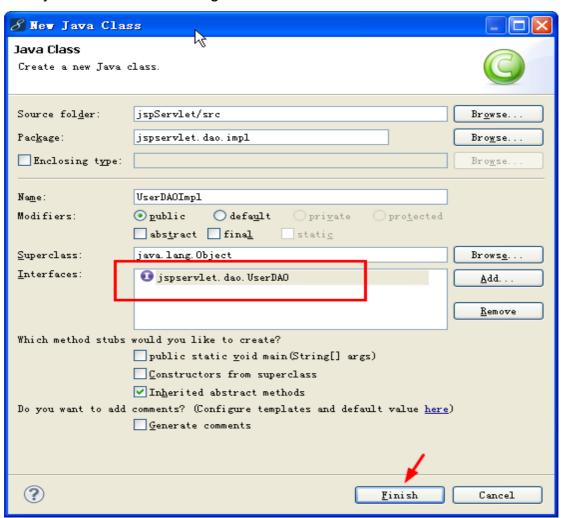
Right click 'jspServlet'->new->class.



Click **add** button, input 'UserDAO' into the text area then you will see the prompt items, select the right one and add it.

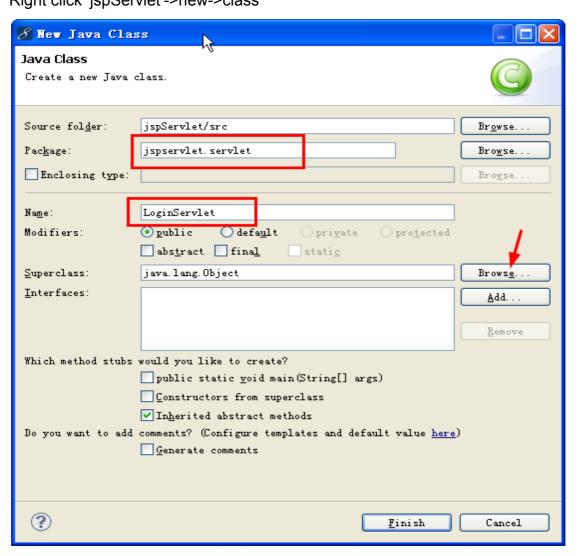


Then you will see the following:



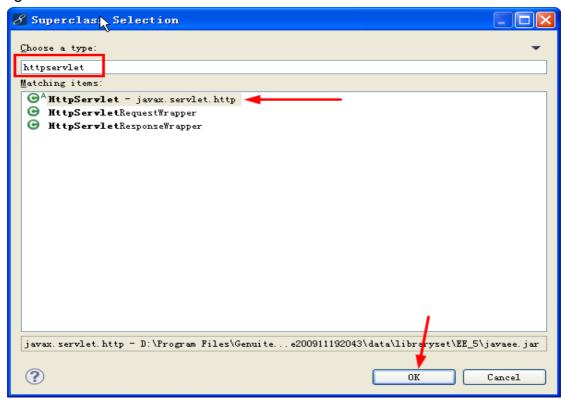
### Edit UserDAOImpl.java

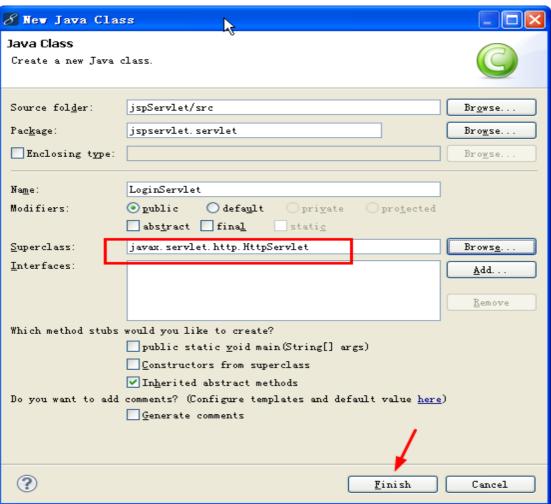
6. Create a servlet to accept the user's request. Right click 'jspServlet'->new->class



This class should extends from HttpServlet. Click **Browse** and input 'HttpServlet' into the text area then you will see the prompt items, select the

#### right one and add it.





#### Edit LoginServlet.java.

```
🕡 LoginServle java 🖂 🚺 UserDAOImpl. java
   public class LoginServlet extends HttpServlet {
                                                                                                                          ^ =
        public void doGet(HttpServletRequest req, HttpServletResponse res)
            throws IOException, ServletException(
        public void doPost(HttpServletRequest req, HttpServletResponse res)
            throws IOException, ServletException(
             User user = new User();
             user.setUsername(req.getParameter("username"));
             user.setPassword(req.getParameter("password"));
             UserDAO dao = new UserDAOImpl();
             int flag = 0;
                    flag = dao.queryByUsername(user);
                } catch (Exception e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
            if(flag == 1){
                HttpSession session=req.getSession();
                session.setAttribute("username", user.getUsername());
res.sendRedirect("./welcome.jsp");
                 res.sendRedirect("./error.isp");
```

#### Pay attention:

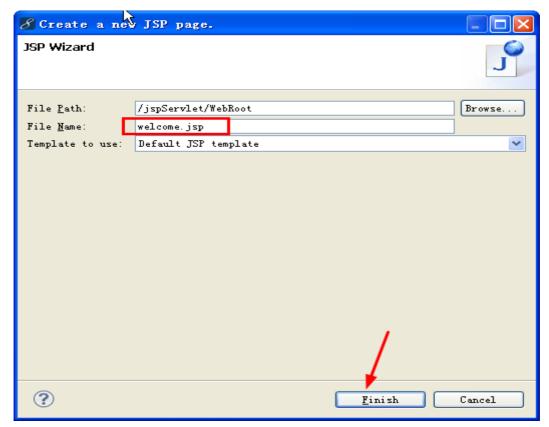
- This class should include two methods: doGet() and doPost(). If the web form is submitted by get method, servlet will call doGet() to process the request, otherwise call doPost(). Here we use doPost().
- Process the user entered information in the login.jsp.
  UserDAO dao = new UserDAOImpl();
- Use method provided by UserDAOImpl to query if the information is right or not.

```
flag = dao.queryByUsername(user);
```

➤ If the user information is right, then put his username in session object and give him a welcome redirect.

```
HttpSession session=req.getSession();
session.setAttribute("username", user.getUsername());
res.sendRedirect("./welcome.jsp");
```

- Else give him an error redirect. res.sendRedirect("./error.jsp");
- 7. So we should create two pages welcome.jsp and error.jsp. Right click 'jspServlet'->new->jsp(advanced template)



Edit welcome.jsp to show username, the following code should be added:

```
| Twicome.jsp \( \text{ } \)
| < \text{ } page language="java" contentType="text/html;charset=ISO-8859-1" pageEncoding="ISO-8859-1" \\
| < \text{ } \)
| < \text{ } \]
|
```

The same way to create error.jsp.

8. Edit web.xml.

```
x w b. xml
    ??xml version="1.0" encoding="UTF-8"?>
   <web-app version="2.5"</pre>
       xmlns="http://java.sun.com/xml/ns/javaee"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
       http://java.sun.com/xml/ns/javaee/web-app 2 5.xsd">
       <servlet-name>loginServlet/servlet-name>
       <servlet-class>jspservlet.servlet.LoginServlet/servlet-class>
      </servlet>
     <welcome-file-list>
       <welcome-file>/login.jsp</welcome-file>
      </welcome-file-list>
     <servlet-mapping>
       <servlet-name>loginServlet</servlet-name>
       <url-pattern>/login</url-pattern>
      </servlet-mapping>
```

Pay attention:

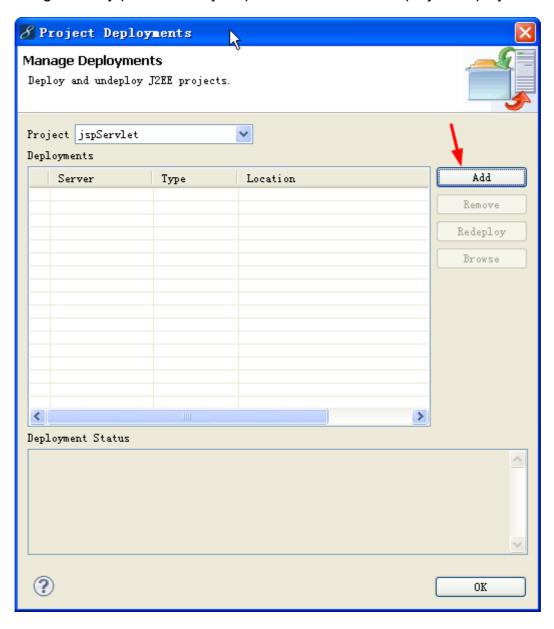
Tag <servlet> and <servlet-mapping> should appear at the same time.

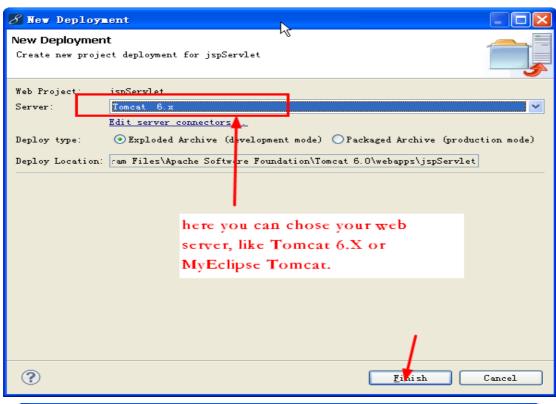
<servlet> is configuration about servlet's name and its class path.

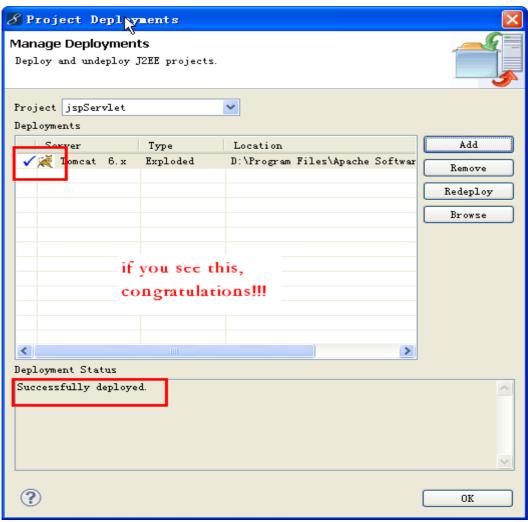
<servlet-mapping> is configuration about URL pattern that should be processed by the corresponding servlet.

Now, you have finished a simple project connect to database. Then try to deploy it.

9. Right click 'jspServlet'->MyEclipse->add and remove project deployments.

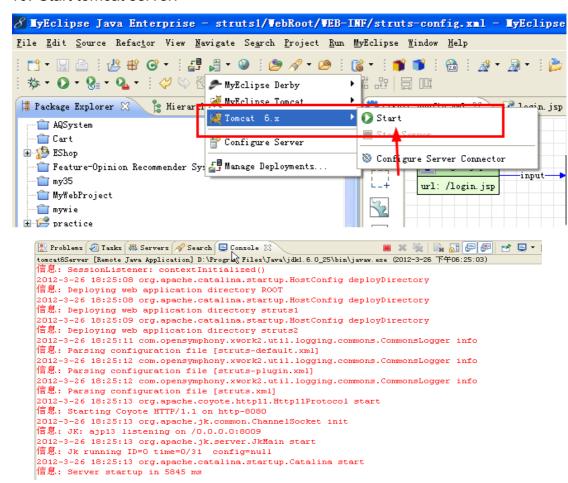






Now you have successfully deployed your project.

10. Start tomcat server.



#### 11. Window->show view->web browser

Enter: http://localhost:8080/jspServlet

Ŵ MyEclipse Neb Browser ⊠		
http://localhost:8080/jspServlet/		
username : password : Submit		

Input username 'tom' and password '123', then submit. You will see this page.

⊕ MyEclipse Web Browser ⊠	
http <mark>MyEclipse Web Browser</mark> ervlet/welcome.jsp	
welcome tom	

Input other names or let it empty, then you will see this page.



Congratulations! You have completed a simple web application. Next time we will add framework to the project.