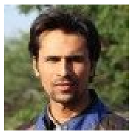
JAVA EE, SERVLETS AND JSP

MVC architecture with servlets and jsp

by Prasad Kharkar • August 11, 2013 • 146 Comments

Bio | Latest Posts



Prasad Kharkar

Prasad Kharkar is a java enthusiast and always keen to explore and learn java technologies. He is SCJP,OCPWCD, OCEJPAD and aspires to be java architect.

In this tutorial we are going to learn how to create a simple MVC application using servlets and jsp.

MVC i.e. Model-View-Controller is a pattern helpful separation of concerns.

- Model represents a POJO object that carries data.
- View is the layer in which the data is presented in visual format.
- Controller is the component which is responsible for communication between model and view.

A user always sees the view and communicates with the controller. We will understand this using a sample login application which will display a welcome username message and if the login fails, it will redirect to an error page. Here is what we are going to create.

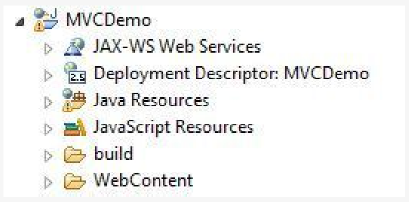
- login.jsp :- this will input username and password
- success.jsp :- If login is successful, then this page is displayed
- error.jsp :- If login is not successful then this page is displayed.
- LoginController.java :- This is controller part of the application which communicates with model
- Authenticator.java :- Has business logic for authentication
- User.java :- Stores username and password for the user.

Requirements:

- Eclipse IDE

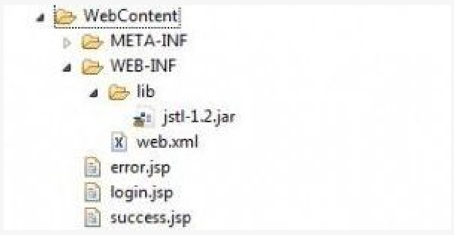
- Apache tomcat server
- JSTL jar

Create a new Dynamic web project in eclipse by clicking File -> New -> Dynamic Web Project. Fill the details i.e. project name, the server. Enter your project name as “MVCDemo”. You will get the following directory structure for the project.



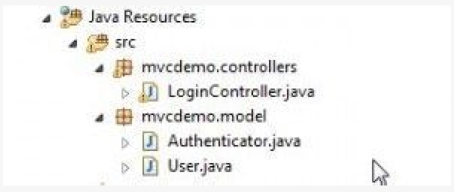
Initial Project Structure

Create success.jsp, error.jsp and login.jsp and LoginController servlet, Authenticator class, User class in the packages as shown in the images. Put the jstl.jar in WEB-INF/lib folder.



File Structure

Now that we have file structure, put this code in corresponding files.



Package Structure

```java
package mvcdemo.controllers;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import mvcdemo.model.Authenticator;
import mvcdemo.model.User;

import sun.text.normalizer.ICUBinary.Authenticate;

```

```java
public class LoginController extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public LoginController() {
        super();
    }

    protected void doPost(HttpServletRequest request,
            HttpServletResponse response) throws ServletExc

        String username = request.getParameter("username");
        String password = request.getParameter("password");
        RequestDispatcher rd = null;

        Authenticator authenticator = new Authenticator();
        String result = authenticator.authenticate(username
        if (result.equals("success")) {
            rd = request.getRequestDispatcher("/success.jsp
            User user = new User(username, password);
            request.setAttribute("user", user);
        } else {
            rd = request.getRequestDispatcher("/error.jsp")
        }
        rd.forward(request, response);
    }

}
```

Authenticator.java

```java
package mvcdemo.model;

public class Authenticator {

    public String authenticate(String username, String pass
        if (("prasad".equalsIgnoreCase(username))
                && ("password".equals(password))) {
            return "success";
        } else {
            return "failure";
        }
    }
}
```

User.java

```java
package mvcdemo.model;

public class User {

    private String username;
    private String password;

    public User(String username, String password){
        this.username = username;
        this.password = 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;
    }

}
```

error.jsp

```
<%@ page language="java" contentType="text/html; charset=IS
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset
<title>Insert title here</title>
</head>
<body>

Login failed, please try again.

</body>
</html>
```

login.jsp

```
<%@ page language="java" contentType="text/html; charset=IS
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset
<title>Insert title here</title>
</head>
<body>
    <form action="LoginController" method="post">
        Enter username : <input type="text" name="username"
        Enter password : <input type="password" name="passw
        <input type="submit" />
    </form>
</body>
</html>
```

success.jsp

```
<%@ page language="java" contentType="text/html; charset=IS
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset
<%@ taglib uri="http://java.sun.com/jstl/core" prefix="c" %
<title>Insert title here</title>
</head>
<body>

    Welcome ${requestScope['user'].username}.

</body>
</html>
```

and the web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instan
  <display-name>MVCDemo</display-name>
   <servlet>
    <description></description>
    <display-name>LoginController</display-name>
    <servlet-name>LoginController</servlet-name>
    <servlet-class>mvcdemo.controllers.LoginController</ser
  </servlet>
  <servlet-mapping>
    <servlet-name>LoginController</servlet-name>
    <url-pattern>/LoginController</url-pattern>
  </servlet-mapping>
</web-app>
```

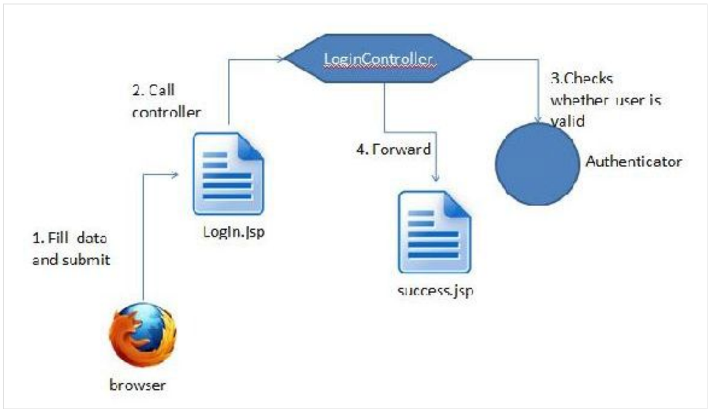
We are done with the code. Let us try and run it.

Start your tomcat server and hit url http://localhost:8080/MVCDemo /login.jsp.

You should be able to view this page.

Enter username as “prasad” and password as “password”. You will see the message “Welcome prasad”.

Let us understand what happens under the hood with the help of diagram.



1. First, user visits login.jsp page and fills out data and submits form.
2. This causes `LoginController` to be invoked because of `form action="LoginController"` .
3. In `LoginController` , following code causes the model to be invoked and set the `User` properties.

```
Authenticator authenticator = new Authenticator();
String result = authenticator.authenticate(username, pa
```

   and `User user = new User(username, password);` .
4. `rd.forward(request, response);` causes the servlet to forward to jsp page.

success.jsp page displays the username with expression language `Welcome ${requestScope['user'].username}.`

Notable advantages of mvc pattern are:

- This separate presentation layer from business layer.
- The controller performs action of invoking the model and sending data to view.
- Model is not even aware that it is used by some web application or desktop application. Authenticator class can be used by desktop applications also. Thus separation helps in re-usability.

Hope this helps in understanding how to create MVC application using servlets and jsp.