

MVC architecture with servlets and jsp

by [Prasad Kharkar](#) • August 11, 2013 • [178 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.

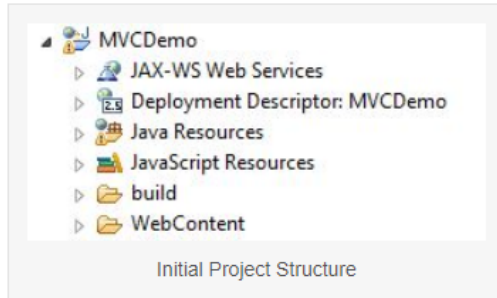
- login.jsp :- this will input username and password
- success.jsp :- If login is successful, then this page is displayed
- error.jsp :- If login is not successful then this page is displayed.
- LoginController.java :- This is controller part of the application which communicates with model
- Authenticator.java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

Requirements:

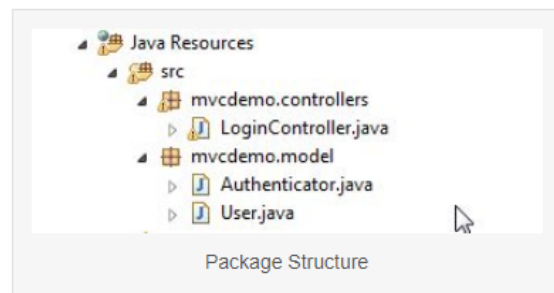
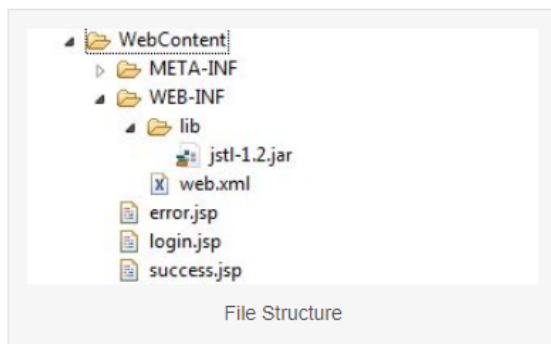
- Eclipse IDE
- Apache tomcat server
- JSTL jar

<http://www.thejavageek.com/2013/08/11/mvc-architecture-with-servlets-and-jsp/>

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.



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.



No
w
tha
t

we have file structure, put this code in corresponding

files.

LoginController Servlet

```
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;

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

    public LoginController() {
        super();
    }

    protected void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {

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

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

```

package mvcdemo.model;

3 public class Authenticator {
4
5     public String authenticate(String username, String password) {
6         if (("prasad".equalsIgnoreCase(username))
7             && ("password".equals(password))) {
8             return "success";
9         } else {
10            return "failure";
11        }
12    }
13 }

```

User.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