**Login Form using JSP + Servlet + JDBC + MySQL Example**

**author:**[**Ramesh Fadatare**](https://www.blogger.com/profile/14691512106162803120)

[**JDBC 4.2**](https://www.javaguides.net/search/label/JDBC%204.2?&max-results=10)[**JSP**](https://www.javaguides.net/search/label/JSP?&max-results=10)[**MYSQL**](https://www.javaguides.net/search/label/MySQL?&max-results=10)[**SERVLET TUTORIAL**](https://www.javaguides.net/search/label/Servlet%20Tutorial?&max-results=10)

In this article, we will build a simple Login Form using [**JSP**](https://www.javaguides.net/p/jsp-tutorial.html), [**Servlet**](https://www.javaguides.net/p/servlet-tutorial.html), [**JDBC**](https://www.javaguides.net/p/jdbc-tutorial.html) and [**MySQL**](https://www.javaguides.net/p/jdbc-tutorial.html) database.

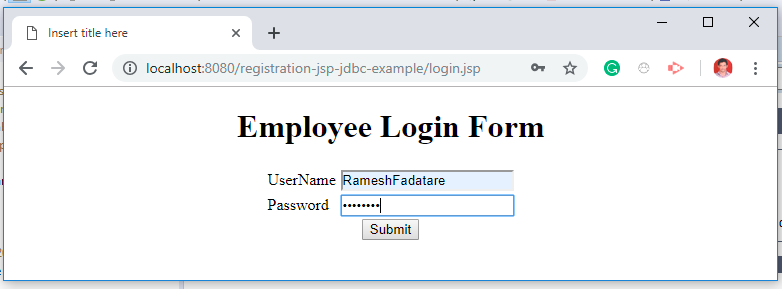
In this example, we will create an **Employee Login Form** and we will validate employee *username* and *password* with the database.

**Video**

This tutorial is explained in the below Youtube Video. Subscribe to my youtube channel at [**Java Guides - YouTube Channel**](https://www.youtube.com/channel/UC1Be9fnFTlcsUlejgfqag0g).

Get source code of this tutorial on my [**GitHub Repository**](https://github.com/RameshMF/servlet-tutorial).

Below diagram shows our Employee Login Form build using JSP:

**[](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEhqcDrMmozrKG8d1mEPj-6MyDlXeuGJlfDoUl0My-CdpswGP5UWDuIdGDW8uBUgeBC3SxD83LqUTi3TpKFlCs3OaL1rTokJpRCurMk9tfo4I7DLfdAxuY79jvbEkZLj3FMCpOyVJwpYkUM/s1600/jsp-servlet-login-form-example.PNG)**

You can download the source code of this article from my GitHub repository. The link has given at end of this article.  
  
Let me list out the tools and technologies that I have used to develop this application.

**Tools and technologies used**

* JSP - 2.2 +
* IDE - STS/Eclipse Neon.3
* JDK - 1.8 or later
* Apache Tomcat - 8.5
* JSTL - 1.2.1
* Servlet API - 2.5
* MySQL - mysql-connector-java-8.0.13.jar

**Development Steps**

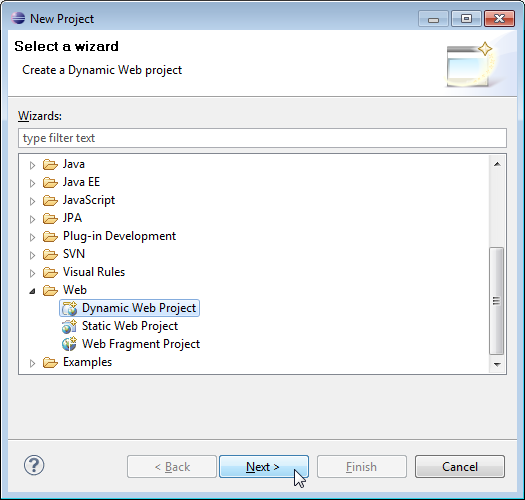
1. Create Eclipse Dynamic web project
2. Add Dependencies
3. Project Structure
4. MySQL Database Setup
5. Create a JavaBean - Login.java
6. Create a LoginDao.java
7. Create a LoginServlet.java
8. Create a login.jsp
9. Create a loginsuccess.jsp
10. Demo

**1. Create an Eclipse Dynamic Web Project**

To create a new dynamic Web project in Eclipse:

1. On the main menu select **File > New > Project....**

2. In the upcoming wizard choose **Web > Dynamic Web Project.**

**[](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEgVJm78O_nMxTOlCL-tLfLQQYSSCTIACRFgeGiWDWoGZvzRnPavDM3HLqdAbipTs4mFpAEdZ2A1R502L7Csv6aFz6c_ybO99z1j-eOXO52sKjW2VKXLC8CSWKO60UCDEbo0_xHWrePZMgc/s1600/create-web-proj-1.png)**

3. Click **Next**.

4. Enter project name as "login-jsp-servlet-jdbc-example";  
5. Make sure that the target runtime is set to Apache Tomcat with the currently supported version.

**2. Add Dependencies**

Add the latest release of below jar files to the lib folder.

- jsp-api.2.3.1.jar

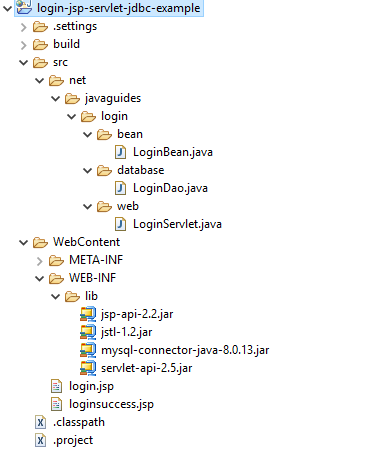
- servlet-api.2.3.jar

- mysql-connector-java-8.0.13.jar

In Eclipse, paste these JAR files to your project directory: **WebContent/WEB-INF/lib**

**3. Project Structure**

Standard project structure for your reference -

**[](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEiLlxYcqreiTrrItvM-WaSGWQ6Gdr0oGpZfkaLFkmO-dbyManDUWN9rqML2r_CVWI_bu6vnUSnG3Q16cc3pLLqfXXNIzg_JN71oQsThkE1R9MppZlR2l1lqBvOtZYIb0wZeoh0CXQaKNbg/s1600/project-structure.PNG)**

**4. MySQL Database Setup**

Let's create a database named "mysql\_database" in MySQL. Now, create a login table using below DDL script:

CREATE TABLE `login` (

`username` varchar(45) NOT NULL,

`password` varchar(45) DEFAULT NULL,

PRIMARY KEY (`username`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

Here is an Insert SQL statement:

INSERT INTO `mysql\_database`.`login` (`username`, `password`) VALUES ("Ramesh", "Ramesh");

**5. Create a JavaBean - LoginBean.java**

Let's create a *LoginBean* class which we will use in JSP action tags.

package net.javaguides.login.bean;

import java.io.Serializable;

public class LoginBean implements Serializable {

/\*\*

\*

\*/

private static final long serialVersionUID = 1 L;

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;

}

}

**6. Create a LoginDao.java**

Let's create a LoginDao class which contains JDBC code to connect with MySQL database. Add the following code to a LoginDao class:

package net.javaguides.login.database;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import net.javaguides.login.bean.LoginBean;

public class LoginDao {

public boolean validate(LoginBean loginBean) throws ClassNotFoundException {

boolean status = false;

Class.forName("com.mysql.jdbc.Driver");

try (Connection connection = DriverManager

.getConnection("jdbc:mysql://localhost:3306/mysql\_database?useSSL=false", "root", "root");

// Step 2:Create a statement using connection object

PreparedStatement preparedStatement = connection

.prepareStatement("select \* from login where username = ? and password = ? ")) {

preparedStatement.setString(1, loginBean.getUsername());

preparedStatement.setString(2, loginBean.getPassword());

System.out.println(preparedStatement);

ResultSet rs = preparedStatement.executeQuery();

status = rs.next();

} catch (SQLException e) {

// process sql exception

printSQLException(e);

}

return status;

}

private void printSQLException(SQLException ex) {

for (Throwable e: ex) {

if (e instanceof SQLException) {

e.printStackTrace(System.err);

System.err.println("SQLState: " + ((SQLException) e).getSQLState());

System.err.println("Error Code: " + ((SQLException) e).getErrorCode());

System.err.println("Message: " + e.getMessage());

Throwable t = ex.getCause();

while (t != null) {

System.out.println("Cause: " + t);

t = t.getCause();

}

}

}

}

}

**7. Create a LoginServlet.java**

Let's create *LoginServlet* to process HTTP request parameters and redirect to the appropriate JSP page based on the employee login status. If login successfully validated with the database then redirect to *loginsuccess.jsp* page otherwise redirect to *login.jsp* page.

package net.javaguides.login.web;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import net.javaguides.login.bean.LoginBean;

import net.javaguides.login.database.LoginDao;

/\*\*

\* @email Ramesh Fadatare

\*/

@WebServlet("/login")

public class LoginServlet extends HttpServlet {

private static final long serialVersionUID = 1 L;

private LoginDao loginDao;

public void init() {

loginDao = new LoginDao();

}

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

String username = request.getParameter("username");

String password = request.getParameter("password");

LoginBean loginBean = new LoginBean();

loginBean.setUsername(username);

loginBean.setPassword(password);

try {

if (loginDao.validate(loginBean)) {

//HttpSession session = request.getSession();

// session.setAttribute("username",username);

response.sendRedirect("loginsuccess.jsp");

} else {

HttpSession session = request.getSession();

//session.setAttribute("user", username);

//response.sendRedirect("login.jsp");

}

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

}

}

**8. Create a login.jsp**

Let's design login HTML form with following fields:

* username
* password

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<div align="center">

<h1>Employee Login Form</h1>

<form action="<%=request.getContextPath()%>/login" method="post">

<table style="with: 100%">

<tr>

<td>UserName</td>

<td><input type="text" name="username" /></td>

</tr>

<tr>

<td>Password</td>

<td><input type="password" name="password" /></td>

</tr>

</table>

<input type="submit" value="Submit" />

</form>

</div>

</body>

</html>

**9. Create a loginsuccess.jsp**

After an employee successfully login then this page shows a successful message on screen:

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<%@page import="net.javaguides.login.database.\*"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<div align="center">

<h1>You have logined successfully</h1>

</div>

</body>

</html>

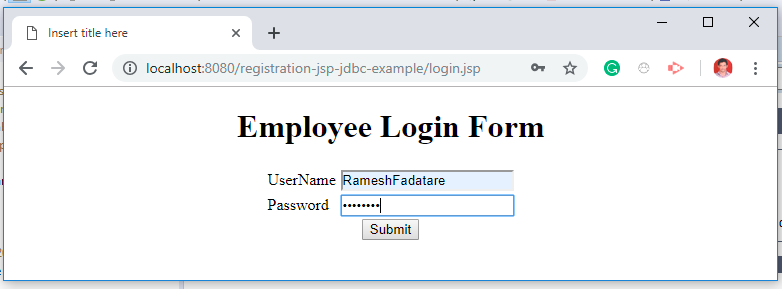
Note that in the above page, we have used JSP action tags. Read more about action tags here.

**9. Demo**

It's time to see a demo of the above development. Deploy this web application in tomcat server.

**Employee Login Form**

Once you deploy this application successfully then hit this link into a browser - [**http://localhost:8080/login-jsp-jdbc-mysql-example/login.jsp**](http://localhost:8080/login-jsp-jdbc-mysql-example/login.jsp)

**[](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEgjNxtDn2UvqYjItgft4LnzFBm3VTPWk3fuJERMO2awKYwmu24EOoUtS-DYlgIfj5kHYW1qOJZ0IMKPbbCLnMbrSeN_hj-5wXt33fSN6Iw3Wf0SxHyWhT7xwwvxk-m5YsHUhtDxPUho2Rs/s1600/jsp-servlet-login-form-example.PNG)**

**Login Success Page**

**[](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEiGW9TE_ONfyp9Weri6Hel6ChdimSmNSVpKvVZQYGFPcMuENZBvd_isyvD2CgmuaR_gn__6fWsLbxer6eiVck8hedwJY2fQ7lydCAS3uNpkB-Diu2o4tiAyyD1rVB-FJoM3NeRQSt9l2m0/s1600/login-success-page.PNG)**