Ejercicio de persistencia de datos STRUTS 2 con base de datos MySQL por JDBC

Lo primero es crear la base de datos

```
CREATE TABLE `struts_tutorial`.`login` (
   `user` VARCHAR( 10 ) NOT NULL ,
   `password` VARCHAR( 10 ) NOT NULL ,
   `name` VARCHAR( 20 ) NOT NULL ,
   PRIMARY KEY ( `user` )
) ENGINE = InnoDB;

INSERT INTO `struts_tutorial`.`login` (`user`, `password`, `name`)
 VALUES ('scott', 'navy', 'Scott Burgemott');
```

El siguiente paso es descargar el **MySQL Connector** archivo jar y la colocación de este archivo en el directorio WEB-INF \ lib de su proyecto. Después de haber hecho esto, ahora estamos listos para crear la clase de acción.

Crear acción

La clase de acción tiene las propiedades correspondientes a las columnas de la tabla de base de datos. Tenemos **de user, password** y **name** como de atributos de la clase. En el método de la acción, se utilizan los parámetros de **user** y **password** para comprobar si el **user** existe, de ser así, se muestra el **name** de **user** en la siguiente pantalla. Si el **user** ha introducido información incorrecta, los enviamos a la pantalla de inicio de sesión de nuevo. El siguiente es el contenido de **LoginAction.java** archivo:

```
package com.tutorialspoint.struts2;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

import com.opensymphony.xwork2.ActionSupport;
```

```
public class LoginAction extends ActionSupport {

   private String user;
   private String password;
   private String name;

   public String execute() {
      String ret = ERROR;
      Connection conn = null;

      try {
         String URL = "jdbc:mysql://localhost/struts_tutorial";
         Class.forName("com.mysql.jdbc.Driver");
         conn = DriverManager.getConnection(URL, "root", "root123");
         String sql = "SELECT name FROM login WHERE";
         sql+=" user = ? AND password = ?";
         PreparedStatement ps = conn.prepareStatement(sql);
         ps.setString(1, user);
         ps.setString(2, password);
         ResultSet rs = ps.executeQuery();

         while (rs.next()) {
            name = rs.getString(1);
            ret = SUCCESS;
         }
      } catch (Exception e) {
         ret = ERROR;
      } finally {
         if (conn != null) {
            try {
               conn.close();
            } catch (Exception e) {
            }
         }
      }
      return ret;
```

```
   }

   public String getUser() {
      return user;
   }

   public void setUser(String user) {
      this.user = user;
   }

   public String getPassword() {
      return password;
   }

   public void setPassword(String password) {
      this.password = password;
   }

   public String getName() {
      return name;
   }

   public void setName(String name) {
      this.name = name;
   }
}
```

Crear página principal

Ahora, vamos a crear un archivo JSP **index.jsp** para recoger el **name** de **user** y **password**. Este **name** de **user** y la **password** se cotejan con la base de datos.

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="s" uri="/struts-tags"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<html>
<head>
<title>Login</title>
</head>
<body>
   <form action="loginaction" method="post">
      User:<br/><input type="text" name="user"/><br/>
      Password:<br/><input type="password" name="password"/><br/>
      <input type="submit" value="Login"/>
   </form>
</body>
</html>
```

Crear Vistas:

Ahora vamos a crear **success.jsp** archivo que se invoca en la acción caso devuelve SUCCESS, pero vamos a tener otro archivo de vista en caso de error se devuelve de la acción.

```
<%@ page contentType="text/html; charset=UTF-8" %>
<%@ taglib prefix="s" uri="/struts-tags" %>
<html>
<head>
<title>Successful Login</title>
</head>
<body>
   Hello World, <s:property value="name"/>
</body>
</html>
```

Después será la vista de archivos **error.jsp** en caso de un error es devuelto por la acción.

```
<%@ page contentType="text/html; charset=UTF-8" %>
<%@ taglib prefix="s" uri="/struts-tags" %>
<html>
<head>
<title>Invalid User Name or Password</title>
</head>
```

```
<body>
   Wrong user name or password provided.
</body>
</html>
```

Archivos de configuración

Por último, vamos a poner todo junto con el archivo de configuración struts.xml de la siguiente manera:

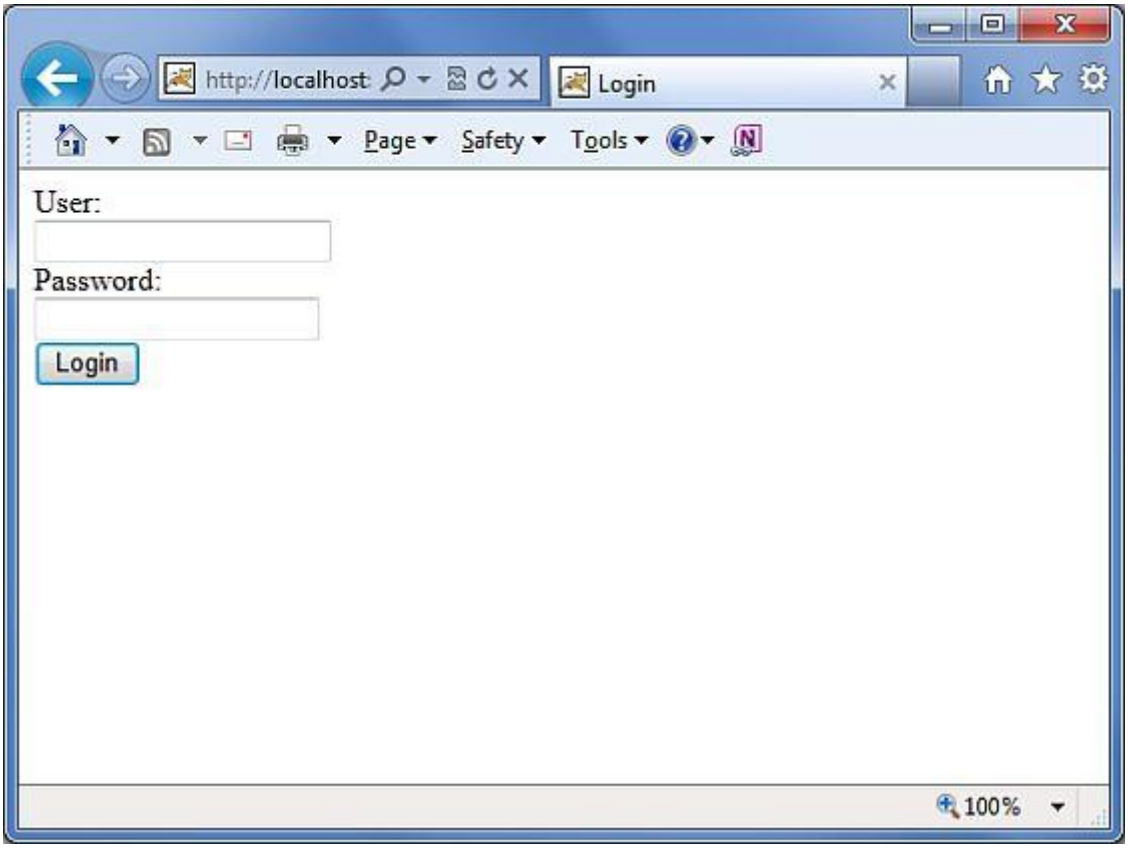
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
"-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
"http://struts.apache.org/dtds/struts-2.0.dtd">

<struts>
   <constant name="struts.devMode" value="true" />
   <package name="helloworld" extends="struts-default">

      <action name="loginaction"
         class="com.tutorialspoint.struts2.LoginAction"
         method="execute">
         <result name="success">/success.jsp</result>
         <result name="error">/error.jsp</result>
      </action>

   </package>

</struts>
```

El siguiente es el contenido de **web.xml** archivo:

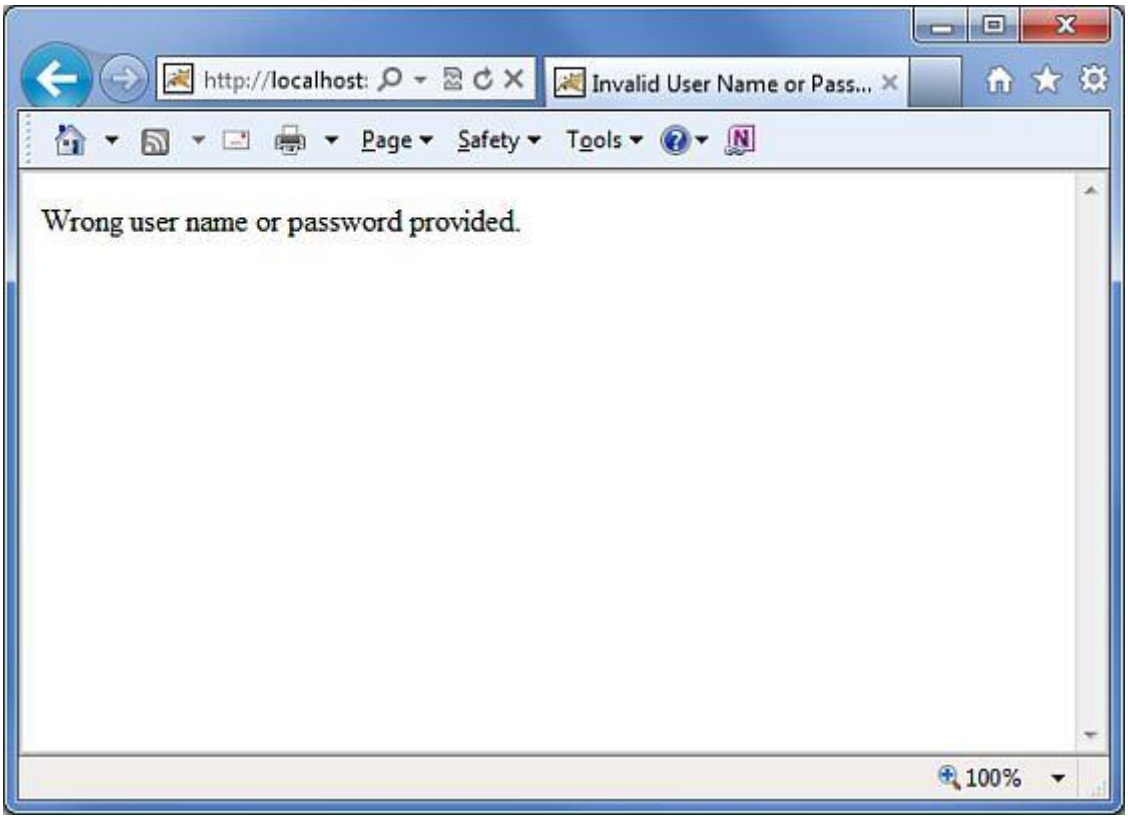
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns="http://java.sun.com/xml/ns/javaee"
   xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
   xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
   http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
```

```
   id="WebApp_ID" version="3.0">


   <display-name>Struts 2</display-name>
   <welcome-file-list>
      <welcome-file>index.jsp</welcome-file>
   </welcome-file-list>
   <filter>
      <filter-name>struts2</filter-name>
      <filter-class>
         org.apache.struts2.dispatcher.FilterDispatcher
      </filter-class>
   </filter>

   <filter-mapping>
      <filter-name>struts2</filter-name>
      <url-pattern>/*</url-pattern>
   </filter-mapping>
</web-app>
```

Ahora, haga clic derecho sobre el **name** del proyecto y haga clic en **Exportar> WAR File**para crear un archivo de Guerra. Entonces implementar esta WAR en el directorio webapps del Tomcat. Por último, iniciar el servidor Tomcat y tratar de acceso URL http: // localhost: 8080 / HelloWorldStruts2 / index.jsp. Esto le dará la siguiente pantalla:



Introduzca un **name** de **user** y una **password** incorrecta. Usted debe ver la página de la red



Ahora introduzca **scott** como **name** de **user** y **la marina** como **password**. Usted debe ver la página de la red