

STRUTS 2 - DATABASE ACCESS

http://www.tutorialspoint.com/struts_2/struts_database_access.htm

Copyright © tutorialspoint.com

This chapter will teach you how to access a database using Struts 2 in simple steps. Struts is a MVC framework and not a database framework but it provides excellent support for JPA/Hibernate integration. We shall look at the hibernate integration in a later chapter, but in this chapter we shall use plain old JDBC to access the database.

The first step in this chapter is to setup and prime our database. I am using MySQL as my database for this example. I have MySQL installed on my machine and I have created a new database called "struts_tutorial". I have created a table called **login** and populated it with some values. Below is the script I used to create and populate the table.

My MYSQL database has the default username "root" and "root123" password

```
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');
```

Next step is to download the [MySQL Connector](#) jar file and placing this file in the WEB-INF\lib folder of your project. After we have done this, we are now ready to create the action class.

Create Action

The action class has the properties corresponding to the columns in the database table. We have **user**, **password** and **name** as String attributes. In the action method, we use the user and password parameters to check if the user exists, if so , we display the user name in the next screen. If the user has entered wrong information, we send them to the login screen again. Following is the content of **LoginAction.java** file:

```
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;
}
}

```

Create main page

Now, let us create a JSP file **index.jsp** to collect the username and password. This username and password will be checked against the database.

```

<%@ 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>

```

Create Views:

Now let us create **success.jsp** file which will be invoked in case action returns SUCCESS, but we will have another view file in case of an ERROR is returned from the action.

```
<%@ 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>
```

Following will be the view file **error.jsp** in case of an ERROR is returned from the action.

```
<%@ 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>
```

Configuration Files

Finally, let us put everything together using the struts.xml configuration file as follows:

```
<?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"

            method="execute">
                <result name="success">/success.jsp</result>
                <result name="error">/error.jsp</result>
            </action>

    </package>
</struts>
```

Following is the content of **web.xml** file:

```
<?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"
    >

    <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>
```

Now, right click on the project name and click **Export > WAR File** to create a War file. Then deploy this WAR in the Tomcat's webapps directory. Finally, start Tomcat server and try to access URL <http://localhost:8080/HelloWorldStruts2/index.jsp>. This will give you following screen:

Enter a wrong user name and password. You should see the net page

Now enter **scott** as user name and **navy** as password. You should see the net page