

<Struts2 を使ったプログラム>

## JSP～Action クラス～DAO クラス～DTO クラス (List 使用) 編

---

<動的 Web プロジェクトを作成する>

まず、動的プロジェクト「WebProj4」を作成しましょう。

---

### *WebProj4*

---

<Struts2 を使うための準備>

---

#### <web.xml の記述>

web.xml は画面を表示する為の設定ファイル。

※必要に応じて画面中央あたりの「ソース」タブをクリックする。

---

WebContent\WEB-INF の web.xml をプログラミングしましょう。

---

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd" id="WebApp_ID" version="3.1">

    <display-name>webproj4</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.ng.filter.StrutsPrepareAndExecuteFilter</filter-class>
    </filter>
    <filter-mapping>
        <filter-name>struts2</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>
</web-app>
```

---

---

### < jar ファイルの格納 >

web.xml は画面を表示する為の設定ファイル。

※必要に応じて画面中央あたりの「ソース」タブをクリックする。

---

WebContent¥WEB-INF¥lib に以下の Jar ファイルを貼り付けましょう。

---

※struts2 に必要な Jar ファイル群（基本構成）

commons-fileupload-1.3.2.jar

commons-io-2.2.jar

commons-lang3-3.2.jar

commons-logging-1.1.3.jar

freemarker-2.3.22.jar

javassist-3.11.0.GA.jar

ognl-3.0.19.jar

struts2-core-2.3.32.jar

xwork-core-2.3.32.jar

※MySQL との接続に必要な Jar ファイル

mysql-connector-java-5.1.42-bin.jar

---

ここまでの作業でフレームワーク<Struts2>を使用するための事前準備は完了です。

---

---

## データベースの作成

---

<SQL ファイルの作成>

動的 Web プロジェクトフォルダ (WebProj4) を右クリックします。

新規→フォルダを選択します。

「sql」と入力して「完了」ボタンを押下。

sql フォルダを右クリック

「その他」を選択

ウィザードに「sql」と入力

SQL ファイルを選択

「testdb.sql」と入力。

「完了」ボタンを押下。

---

testdb.sql に以下をプログラミングしましょう。

---

```
drop database if exists testdb;
```

```
create database testdb;
```

```
use testdb;
```

```
create table users(
```

```
user_id int primary key auto_increment,
```

```
user_name varchar(255),
```

```
password varchar(255)
```

```
);
```

```
insert into users values(1,"taro","123");
```

```
insert into users values(2,"jiro","456");
```

```
insert into users values(3,"hanako","789");
```

```
create table inquiry(
```

```
name varchar(255),
```

```
qtype varchar(255),
```

```
body varchar(255)
```

```
);
```

---

<SQL ファイルを使って、データベースを作成する>

※初めて操作する場合と、2回目以降の場合では方法が変わります。

(初めて操作する場合)

SQL ファイルを選択して、

1. 右クリック

2. SQL ファイルの実行

3. データベース・サーバー・タイプ : MySql\_5.1 を選択してください。

接続プロファイル名 : 「作成」 ボタンを押下してください。

4. MySQL を選択し、名前欄に接続するデータベース名を入力してください。

「次へ」 ボタンを押下します。

5. ドライバー : 右から 2 番目の四角に+がついたマークを選択し、MySQL JDBC ドライバー MySQL 5.1 を選択してください。

「Jar リスト」 のタブを選択してください。

ドライバー欄に mysql-connector-java-5.1.0-bin.jar があるが サンプルとしてダミー設定されています。

その為、適切なものに置き換えます。

まず、「JAR/Zip の除去」 ボタンを押下し削除します。

「JAR/Zip の追加」 のボタンを押下します。

6. mysql-connector-java-x.x.xx-bin.jar を選択し、「開く」を押下します。

7. データベース : データベース名を入力します。

8. URL: jdbc:mysql://localhost:3306/database を jdbc:mysql://localhost:3306/mysql に変更します。

これは自分の PC に作成されたデータベースを指定することで、接続を実現する為です。

9. パスワード : mysql と入力します。

「次へ」 ボタンを押下します。

10. 「次へ」 ボタンを押下する。

11. 前の画面にて入力した情報が表示されます。

「完了」 ボタンを押下します。

---

(2回目以降の場合)

SQL ファイルを選択して、

- 1. 右クリック
- 2. SQL ファイルの実行
- 3. 接続プロファイル名をプルダウンして、作成した「新規 MySQL」を選択する
- 4. データベース名をプルダウンして、作成したデータベースを選択する
- 5. OK ボタンを押す

上記の操作でデータベースが作成されたことを確認しましょう。

- 1. コマンドプロンプトを開く（cmd）
- 2. MySQL にログインする。  

```
mysql -u root -p
```

```
mysql
```
- 3. show databases;
- 4. testdb が存在することを確認する。
- 5. use testdb;
- 6. show tables;
- 7. select \* from users;
- 8. test\_table に情報が登録されていることを確認する。

※これで、以下のデータベースとテーブルが作成できています。

データベース名 : testdb

テーブル名 : users

テーブル定義 :

Field	Type	Null	Key	Default	Extra
user_id	int(11)	NO	PRIMARY	NULL	
user_name	varchar(255 )	YES		NULL	
Password	varchar(255 )	YES		NULL	

user_id	user_name	password
1	taro	123
2	jiro	456
3	hanako	789

テーブル名 : inquiry

テーブル定義 :

Field	Type	Null	Key	Default	Extra
name	varchar(255 )	YES		NULL	
qtype	varchar(255 )	YES		NULL	
body	varchar(255 )	YES		NULL	

## CSS の作成

---

<CSS ファイルの作成>

動的 Web プロジェクトフォルダ (WebProj4) の WebContent フォルダを右クリックします。

新規->フォルダを選択します。

「css」と入力して「完了」ボタンを押下。

css フォルダを右クリック

「その他」を選択

ウィザードに「css」と入力

CSS ファイルを選択

「style.css」と入力。

「完了」ボタンを押下。

---

style.css に以下をプログラミングしましょう。

---

```
table {
    width: auto;
    border-spacing: 0;
    font-size: 14px;
}
table th {
    color: #fff;
    padding: 8px 15px;
    background: #258;
    background: -moz-linear-gradient(rgba(34, 85, 136, 0.7), rgba(34, 85, 136, 0.9) 50%);
    background: -webkit-gradient(linear, 100% 0%, 100% 50%, from(rgba(34, 85, 136, 0.7)), to(rgba(34, 85, 136, 0.9)));
    font-weight: bold;
    border-left: 1px solid #258;
    border-top: 1px solid #258;
    border-bottom: 1px solid #258;
    line-height: 120%;
    text-align: center;
    text-shadow: 0 -1px 0 rgba(34, 85, 136, 0.9);
    box-shadow: 0px 1px 1px rgba(255, 255, 255, 0.3) inset;
}
table th:first-child {
    border-radius: 5px 0 0 0;
}
table th:last-child {
    border-radius: 0 5px 0 0;
```

```
border-right:1px solid #258;
box-shadow: 2px 2px 1px rgba(0,0,0,0.1),0px 1px 1px rgba(255,255,255,0.3) inset;
}
table tr td {
padding: 8px 15px;
border-bottom: 1px solid #84b2e0;
border-left: 1px solid #84b2e0;
text-align: center;
}
table tr td:last-child {
border-right: 1px solid #84b2e0;
box-shadow: 2px 2px 1px rgba(0,0,0,0.1);
}
table tr {
background: #fff;
}
table tr:nth-child(2n+1) {
background: #f1f6fc;
}
table tr:last-child td {
box-shadow: 2px 2px 1px rgba(0,0,0,0.1);
}
table tr:last-child td:first-child {
border-radius: 0 0 0 5px;
}
table tr:last-child td:last-child {
border-radius: 0 0 5px 0;
}
table tr:hover {
background: #bbd4ee;
cursor:pointer;
}
```

---



---

## HelloStruts 機能

---

index.jsp

---

①WebContent に index.jsp を作成し、以下をプログラミングしましょう。

---

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="s" uri="/struts-tags" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>INDEX</title>
</head>
<body>
    <s:form action="HelloStrutsAction">
        <s:submit value="HelloStruts"/>
    </s:form>
</body>
</html>
```

---

パッケージの作成

---

②Java リソース¥src に以下のパッケージを作成しましょう。

---

```
com.internousdev.webproj4.action
com.internousdev.webproj4.dao
com.internousdev.webproj4.dto
com.internousdev.webproj4.util
```

---

```
DBConnector.java
(DBConnector.java)
```

---

③Java リソース¥src¥com.internousdev.webproj4.util パッケージに DBConnector.java を作成し、以下をプログラミングしましょう。

---

```
package com.internousdev.webproj4.util;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConnector {
    /**
     * JDBC ドライバー名
     */
    private static String driverName = "com.mysql.jdbc.Driver";
    /**
     * データベース接続 URL
     */
    private static String url = "jdbc:mysql://localhost/testdb";
    /**
     * データベース接続ユーザ名
     */
    private static String user = "root";
    /**
     * データベース接続パスワード
     */
    private static String password = "mysql";

    public Connection getConnection() {
        Connection con = null;
        try{
            Class.forName(driverName);
            con = DriverManager.getConnection(url, user, password);
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return con;
    }
}
```

---

---

HelloStrutsDTO.java  
(HelloStrutsDTO.java)

---

④Java リソース¥src¥com.internousdev.webproj4.dto パッケージ  
に HelloStrutsDTO.java を作成し、以下をプログラミングしましょう。

---

```
package com.internousdev.webproj4.dto;

public class HelloStrutsDTO {
    private int userId;
    private String userName;
    private String password;
    private String result;

    public int getUserId() {
        return userId;
    }

    public void setUserId(int userId) {
        this.userId = userId;
    }

    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;
    }
}
```

```
    public String getResult() {  
        return result;  
    }  
  
    public void setResult(String result) {  
        this.result = result;  
    }  
}
```

---

HelloStrutsDAO.java  
(HelloStrutsDAO.java)

---

⑤Java リソース¥src¥com.internousdev.webproj4.dao パッケージ  
に HelloStrutsDAO.java を作成し、以下をプログラミングしましょう。

---

```
package com.internousdev.webproj4.dao;  
  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.util.ArrayList;  
import java.util.List;  
  
import com.internousdev.webproj4.dto.HelloStrutsDTO;  
import com.internousdev.webproj4.util.DBConnector;  
  
public class HelloStrutsDAO {  
  
    List<HelloStrutsDTO> helloStrutsDTOList = new ArrayList<HelloStrutsDTO>();  
  
    public List<HelloStrutsDTO> select() {  
        DBConnector db = new DBConnector();  
        Connection con = db.getConnection();  
  
        String sql = "select * from users";  
        try {
```

---

```

        PreparedStatement ps = con.prepareStatement(sql);
        ResultSet rs = ps.executeQuery();

        while(rs.next()) {
            HelloStrutsDTO dto=new HelloStrutsDTO();
            dto.setUserId(rs.getInt("user_id"));
            dto.setUserName(rs.getString("user_name"));
            dto.setPassword(rs.getString("password"));
            dto.setResult("MySQL と接続できます。");
            helloStrutsDTOList.add(dto);
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    try {
        con.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return helloStrutsDTOList;
}
}

```

---

struts.xml

---

⑥Java リソース¥src に struts.xml を作成し、以下をプログラミングしましょう。

---

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
"-//Apache Software Foundation//DTD Struts Configuration 2.3//EN"
"http://struts.apache.org/dtds/struts-2.3.dtd">
<struts>
<constant name="struts.devMode" value="true" />

<package name="com.internousdev.webproj4.action" extends="struts-default">

<action name="HelloStrutsAction" class="com.internousdev.webproj4.action.HelloStrutsAction"
method="execute">
<result name="success">hello.jsp</result>

```

```
<result name="error">helloStrutsError.jsp</result>
</action>
</package>
</struts>
```

---

HelloStrutsAction  
(HelloStrutsAction.java)

---

⑦Java リソース¥src¥com.internousdev.webproj4.action パッケージ  
に HelloStrutsAction.java を作成し、以下をプログラミングしましょう。

---

```
package com.internousdev.webproj4.action;

import java.util.ArrayList;
import java.util.List;

import com.internousdev.webproj4.dao.HelloStrutsDAO;
import com.internousdev.webproj4.dto.HelloStrutsDTO;
import com.opensymphony.xwork2.ActionSupport;

public class HelloStrutsAction extends ActionSupport {

    private List<HelloStrutsDTO> HelloStrutsDTOList = new ArrayList<HelloStrutsDTO>();

    public String execute() {
        String ret=ERROR;
        HelloStrutsDAO dao=new HelloStrutsDAO();

        HelloStrutsDTOList=dao.select();

        if(HelloStrutsDTOList.size() > 0){
            ret=SUCCESS;
        }else{
            ret=ERROR;
        }
        return ret;
    }
}
```

---

```
public List<HelloStrutsDTO> getHelloStrutsDTOList() {  
    return HelloStrutsDTOList;  
}  
  
public void setHelloStrutsDTOList(List<HelloStrutsDTO> helloStrutsDTOList) {  
    HelloStrutsDTOList = helloStrutsDTOList;  
}  
}
```

---

hello.jsp

---

⑧WebContent に hello.jsp を作成し、以下をプログラミングしましょう。

---

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  
<%@ taglib prefix="s" uri="/struts-tags"%>  
<!DOCTYPE html>  
<html>  
<head>  
<meta charset="utf-8">  
<link rel="stylesheet" type="text/css" href="/css/style.css">  
<title>HelloStruts</title>  
</head>  
<body>  
<h1>HelloStruts2!</h1>  
<br>  
  
<table>  
    <tbody>  
        <tr>  
            <th>USERID</th>  
            <th>USERNAME</th>  
            <th>PASSWORD</th>  
            <th>RESULT</th>  
        </tr>  
  
        <s:iterator value="helloStrutsDTOList">  
            <tr>  
                <td><s:property value="userId"/></td>
```

```
<td><s:property value="userName"/></td>
<td><s:property value="password"/></td>
<td><s:property value="result"/></td>
</tr>
</s:iterator>
</tbody>
</table>

</body>
</html>
```

---

helloStrutsError.jsp

---

⑨WebContent に helloStrutsError.jsp を作成し、以下をプログラミングしましょう。

---

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="s" uri="/struts-tags"%>
<!DOCTYPE>
<html>
<head>
<meta charset="UTF-8">
<title>HelloStrutsError</title>
</head>
<body>
<h1>エラーが発生しました</h1>
<br>
</body>
</html>
```

---

⑩WebProj4 プロジェクトを実行してみましょう。

---



← → 🚫 🔗 http://localhost:8080/WebProj4/

HelloStruts

← → 🚫 🔗 http://localhost:8080/WebProj4/HelloStrutsAction.action

# HelloStruts2!

USERID	USERNAME	PASSWORD	RESULT
1	taro	123	MySQLと接続できます。
2	jiro	456	MySQLと接続できます。
3	hanako	789	MySQLと接続できます。

← → 🚫 🔗 http://localhost:8080/WebProj4/HelloStrutsAction.action

# エラーが発生しました

---

---

## Welcome 機能(1)

---

index.jsp

---

①WebContent の index.jsp に以下を追加しましょう。

```
<s:form action="WelcomeAction">
    <s:submit value="Welcome"/>
</s:form>
```

---

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="s" uri="/struts-tags" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>INDEX</title>
</head>
<body>
    <s:form action="HelloStrutsAction">
        <s:submit value="HelloStruts"/>
    </s:form>

    <s:form action="WelcomeAction">
        <s:submit value="Welcome"/>
    </s:form>
</body>
</html>
```

---

struts.xml

---

②Java リソース¥src の struts.xml に以下を追加しましょう。

```
<action name="WelcomeAction" class="com.internousdev.webproj4.action.WelcomeAction" method="execute">
<result name="success">welcome.jsp</result>
</action>
```

---

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
"-//Apache Software Foundation//DTD Struts Configuration 2.3//EN"
"http://struts.apache.org/dtds/struts-2.3.dtd">
<struts>
<constant name="struts.devMode" value="true" />

<package name="com.internousdev.webproj3.action" extends="struts-default">

<action name="HelloStrutsAction" class="com.internousdev.webproj4.action.HelloStrutsAction"
method="execute">
<result name="success">hello.jsp</result>
</action>

<action name="WelcomeAction" class="com.internousdev.webproj4.action.WelcomeAction" method="execute">
<result name="success">welcome.jsp</result>
</action>

</package>
</struts>
```

---

WelcomeAction  
(WelcomeAction.java)

---

③ Java リソース ¥src¥com.internousdev.webproj4.action パッケージに  
WelcomeAction.java を作成し、以下をプログラミングしましょう。

---

```
package com.internousdev.webproj4.action;

import com.opensymphony.xwork2.ActionSupport;

public class WelcomeAction extends ActionSupport{
    public String execute() {
        return SUCCESS;
    }
}
```

---

welcome.jsp

---

④WebContent に welcome.jsp を作成し、以下をプログラミングしましょう。

---

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="s" uri="/struts-tags" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>WELCOME</title>
</head>
<body>
名前とパスワードを入力してください。
<s:form action="LoginAction">
  <s:textfield name="username" label="ユーザー名" />
  <s:password name="password" label="パスワード" />
  <s:submit value="送信"/>
</s:form>
</body>
</html>
```

---

---

## Welcome 機能 (2)

---

struts.xml

---

①Java リソース¥src の struts.xml に以下を追加しましょう。

```
<action name="LoginAction" class="com.internousdev.webproj4.action.LoginAction" method="execute">
<result name="success">login.jsp</result>
<result name="error">loginError.jsp</result>
</action>

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
"-//Apache Software Foundation//DTD Struts Configuration 2.3//EN"
"http://struts.apache.org/dtds/struts-2.3.dtd">
<struts>
<constant name="struts.devMode" value="true" />

<package name="com.internousdev.webproj4.action" extends="struts-default">

<action name="HelloStrutsAction" class="com.internousdev.webproj4.action.HelloStrutsAction"
method="execute">
<result name="success">hello.jsp</result>
<result name="error">helloStrutsError.jsp</result>
</action>

<action name="WelcomeAction" class="com.internousdev.webproj4.action.WelcomeAction" method="execute">
<result name="success">welcome.jsp</result>
</action>

<action name="LoginAction" class="com.internousdev.webproj4.action.LoginAction" method="execute">
<result name="success">login.jsp</result>
<result name="error">loginError.jsp</result>
</action>

</package>
</struts>
```

---

LoginDTO.java  
(LoginDTO.java)

---

②Java リソース¥src¥com.internousdev.webproj4.dto パッケージ  
に LoginDTO.java を作成し、以下をプログラミングしましょう。

---

```
package com.internousdev.webproj4.dto;

public class LoginDTO {
    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;
    }
}
```

---

LoginDAO.java  
(LoginDAO.java)

---

③Java リソース¥src¥com.internousdev.webproj4.dao パッケージ  
に LoginDAO.java を作成し、以下をプログラミングしましょう。

---

```
package com.internousdev.webproj4.dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
```

---

```
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import com.internousdev.webproj4.dto.LoginDTO;
import com.internousdev.webproj4.util.DBConnector;

public class LoginDAO {

    public String username;
    public String password;

    public List<LoginDTO> loginDTOList = new ArrayList<LoginDTO>();

    public List<LoginDTO> select(String username,String password) {

        DBConnector db = new DBConnector();
        Connection con = db.getConnection();

        String sql = "select * from users where user_name=? and password=?";
        try {
            PreparedStatement ps = con.prepareStatement(sql);
            ps.setString(1, username);
            ps.setString(2, password);
            ResultSet rs = ps.executeQuery();

            while (rs.next()) {
                LoginDTO dto=new LoginDTO();
                dto.setUsername(rs.getString("user_name"));
                dto.setPassword(rs.getString("password"));
                loginDTOList.add(dto);
            }

            if(loginDTOList.size()<=0){
                LoginDTO dto=new LoginDTO();
                dto.setUsername("該当なし");
                dto.setPassword("該当なし");
                loginDTOList.add(dto);
            }
        }
    }
}
```

---

```

        } catch (SQLException e) {
            e.printStackTrace();
        }
        try {
            con.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return loginDTOList;
    }
}

```

---

LoginAction  
(LoginAction.java)

---

④Java リソース¥src¥com.internousdev.webproj4.action パッケージに LoginAction.java を作成し、以下をプログラミングしましょう。

---

```

package com.internousdev.webproj4.action;

import java.util.ArrayList;
import java.util.List;

import com.internousdev.webproj4.dao.LoginDAO;
import com.internousdev.webproj4.dto.LoginDTO;
import com.opensymphony.xwork2.ActionSupport;

public class LoginAction extends ActionSupport {
    private String username;
    private String password;
    private List<LoginDTO> loginDTOList = new ArrayList<LoginDTO>();

    public String execute() {
        String ret=ERROR;
        System.out.println(username);
        System.out.println(password);
        LoginDAO dao=new LoginDAO();
    }
}

```



```
        LoginDTOList=dao.select(username, password);

        if(this.username.equals(LoginDTOList.get(0).getUsername()) &&
this.password.equals(LoginDTOList.get(0).getPassword())){

            ret= SUCCESS;

        }else{

            ret=ERROR;

        }

        return ret;

    }

    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;

    }

    public List<LoginDTO> getLoginDTOList() {

        return LoginDTOList;

    }

    public void setLoginDTOList(List<LoginDTO> loginDTOList) {

        LoginDTOList = loginDTOList;

    }

}
```

---

---

login.jsp

---

⑤WebContent に login.jsp を作成し、以下をプログラミングしましょう。

---

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="s" uri="/struts-tags" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<link rel="stylesheet" type="text/css" href="/css/style.css">
<title>LOGIN</title>
</head>
<body>

<s:property value="LoginDTOList.get(0).username"/>さん、ようこそ！

<br>
<table>
<tbody>
<tr>
<th>USERNAME</th>
<th>PASSWORD</th>
</tr>

<s:iterator value="loginDTOList">
<tr>
<td><s:property value="username"/></td>
<td><s:property value="password"/></td>
</tr>
</s:iterator>

</tbody>
</table>

</body>
</html>
```

---

---

loginError.jsp

---

⑥WebContent に loginError.jsp を作成し、以下をプログラミングしましょう。

---

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="s" uri="/struts-tags" %>
<!DOCTYPE>
<html>
<head>
<meta charset="UTF-8">
<link rel="stylesheet" type="text/css" href="/css/style.css">
<title>LoginError</title>
</head>
<body>
<h1>エラーが発生しました</h1>
<br>
<h3>ログインできません。</h3>

<s:if test="(loginDTOList.get(0).username)=="該当なし"">
  該当者はいませんでした。
</s:if>

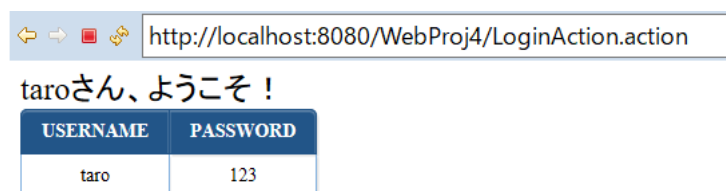
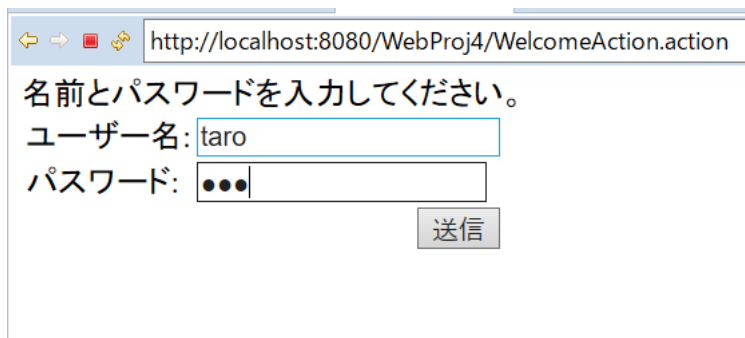
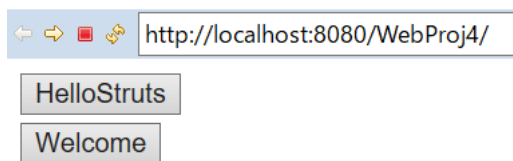
<br>
<table>
<tbody>
<tr>
  <th>USERNAME</th>
  <th>PASSWORD</th>
</tr>

<s:iterator value="loginDTOList">
<tr>
  <td><s:property value="username"/></td>
  <td><s:property value="password"/></td>
</tr>
</s:iterator>
</tbody>
</table>
```

</body>

</html>

⑦WebProj4 プロジェクトを実行してみましょう。



---

## Inquiry (お問い合わせ) 機能(1)

---

index.jsp

---

①WebContent の index.jsp に以下を追加しましょう。

```
<s:form action="InquiryAction">
    <s:submit value="問い合わせ"/>
</s:form>
```

---

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="s" uri="/struts-tags" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>INDEX</title>
</head>
<body>
    <s:form action="HelloStrutsAction">
        <s:submit value="HelloStruts"/>
    </s:form>

    <s:form action="WelcomeAction">
        <s:submit value="Welcome"/>
    </s:form>

    <s:form action="InquiryAction">
        <s:submit value="問い合わせ"/>
    </s:form>
</body>
</html>
```

---

---

struts.xml

---

②Java リソース¥src の struts.xml に以下を追加しましょう。

```
<action name="InquiryAction" class="com.internousdev.webproj4.action.InquiryAction" method="execute">
<result name="success">inquiry.jsp</result>
</action>
```

---

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
"-//Apache Software Foundation//DTD Struts Configuration 2.3//EN"
"http://struts.apache.org/dtds/struts-2.3.dtd">
<struts>
<constant name="struts.devMode" value="true" />

<package name="com.internousdev.webproj4.action" extends="struts-default">

<action name="HelloStrutsAction" class="com.internousdev.webproj4.action.HelloStrutsAction"
method="execute">
<result name="success">hello.jsp</result>
</action>

<action name="WelcomeAction" class="com.internousdev.webproj4.action.WelcomeAction" method="execute">
<result name="success">welcome.jsp</result>
</action>

<action name="LoginAction" class="com.internousdev.webproj4.action.LoginAction" method="execute">
<result name="success">login.jsp</result>
</action>

<action name="InquiryAction" class="com.internousdev.webproj4.action.InquiryAction" method="execute">
<result name="success">inquiry.jsp</result>
</action>

</package>
</struts>
```

---

---

`InquiryAction`  
(`InquiryAction.java`)

---

③ Java リソース `¥src¥com.internousdev.webproj4.action` パッケージに `InquiryAction.java` を作成し、以下をプログラミングしましょう。

---

```
package com.internousdev.webproj4.action;

import com.opensymphony.xwork2.ActionSupport;

public class InquiryAction extends ActionSupport{
    public String execute() {
        return SUCCESS;
    }
}
```

---

`inquiry.jsp`

---

④ `WebContent` に `inquiry.jsp` を作成し、以下をプログラミングしましょう。

---

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="s" uri="/struts-tags" %>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>お問い合わせフォーム</title>
</head>
<body>
    <s:form method="post" action="InquiryCompleteAction">
        名前:<input type="text" name="name"/><br>
        お問い合わせの種類:<br>
        <select name="qtype" >
            <option value="company">会社について</option>
            <option value="product">製品について</option>
        </select>
    </s:form>
</body>
</html>
```

```
        <option value="support">アフターサポートについて</option>
    </select>
    <br>
    お問い合わせ内容:
    <s:textarea name="body"/>
    <br> <s:submit value="登録"/>
</s:form>
</body>
</html>
```

---

---



---

## Inquiry (お問い合わせ) 機能(2)

---

struts.xml

---

①Java リソース¥src の struts.xml に以下を追加しましょう。

```
<action name="InquiryCompleteAction" class="com.internousdev.webproj4.action.InquiryCompleteAction"
method="execute">
<result name="success">inquiryComplete.jsp</result>
</action>
```

---

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
"-//Apache Software Foundation//DTD Struts Configuration 2.3//EN"
"http://struts.apache.org/dtds/struts-2.3.dtd">
<struts>
<constant name="struts.devMode" value="true" />

<package name="com.internousdev.webproj4.action" extends="struts-default">

<action name="HelloStrutsAction" class="com.internousdev.webproj4.action.HelloStrutsAction"
method="execute">
<result name="success">hello.jsp</result>
</action>

<action name="WelcomeAction" class="com.internousdev.webproj4.action.WelcomeAction" method="execute">
<result name="success">welcome.jsp</result>
</action>

<action name="LoginAction" class="com.internousdev.webproj4.action.LoginAction" method="execute">
<result name="success">login.jsp</result>
</action>

<action name="InquiryAction" class="com.internousdev.webproj4.action.InquiryAction" method="execute">
<result name="success">inquiry.jsp</result>
</action>
```

```
<action name="InquiryCompleteAction" class="com.internousdev.webproj4.action.InquiryCompleteAction"
method="execute">
<result name="success">inquiryComplete.jsp</result>
</action>
</package>
</struts>
```

---

InquiryDTO

(InquiryDTO.java)

---

②Java リソース¥src¥com.internousdev.webproj4.DTO パッケージ  
に InquiryDTO.java を作成し、以下をプログラミングしましょう。

---

```
package com.internousdev.webproj4.dto;

public class InquiryDTO {

    private String name;
    private String qtype;
    private String body;

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getQtype() {
        return qtype;
    }
    public void setQtype(String qtype) {
        this.qtype = qtype;
    }
    public String getBody() {
        return body;
    }
    public void setBody(String body) {
        this.body = body;
    }
}
```

---

InquiryCompleteDAO  
(InquiryCompleteDAO.java)

---

③Java リソース¥src¥com.internousdev.webproj4.DAO パッケージ  
に InquiryCompleteDAO.java を作成し、以下をプログラミングしましょう。

---

```
package com.internousdev.webproj4.dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import com.internousdev.webproj4.dto.InquiryDTO;
import com.internousdev.webproj4.util.DBConnector;

public class InquiryCompleteDAO {

    List<InquiryDTO> inquiryDTOList = new ArrayList<InquiryDTO>();

    public List<InquiryDTO> select() {
        DBConnector db = new DBConnector();
        Connection con = db.getConnection();

        String sql = "select * from inquiry";
        try {
            PreparedStatement ps = con.prepareStatement(sql);
            ResultSet rs = ps.executeQuery();

            while(rs.next()) {
                InquiryDTO dto=new InquiryDTO();
                dto.setName(rs.getString("name"));
                dto.setQtype(rs.getString("qtype"));
                dto.setBody(rs.getString("body"));
                inquiryDTOList.add(dto);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

---

```

        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    try {
        con.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return inquiryDTOList;
}

public int insert(String name, String qtype, String body) {
    int ret = 0;
    DBConnector db = new DBConnector();
    Connection con = db.getConnection();

    String sql = "insert into inquiry values(?, ?, ?)";
    try {
        PreparedStatement ps = con.prepareStatement(sql);
        ps.setString(1, name);
        ps.setString(2, qtype);
        ps.setString(3, body);
        int i = ps.executeUpdate();
        if (i > 0) {
            System.out.println(i + "件登録されました");
            ret = i;
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    try {
        con.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return ret;
}
}

```

---

---

`InquiryCompleteAction`  
(`InquiryCompleteAction.java`)

---

④Java リソース¥src¥com.internousdev.webproj4.action パッケージ  
に `InquiryCompleteAction.java` を作成し、以下をプログラミングしましょう。

---

```
package com.internousdev.webproj4.action;

import java.util.ArrayList;
import java.util.List;

import com.internousdev.webproj4.dao.InquiryCompleteDAO;
import com.internousdev.webproj4.dto.InquiryDTO;
import com.opensymphony.xwork2.ActionSupport;

public class InquiryCompleteAction extends ActionSupport{

    private String name;
    private String qtype;
    private String body;

    private List<InquiryDTO> inquiryDTOList = new ArrayList<InquiryDTO>();

    public String execute() {
        String ret=ERROR;
        InquiryCompleteDAO dao=new InquiryCompleteDAO();
        int count= dao.insert(name, qtype, body);
        if(count > 0){
            inquiryDTOList=dao.select();

            ret=SUCCESS;
        }
        return ret;
    }

    public String getName() {
        return name;
    }
}
```

```
public void setName(String name) {  
    this.name = name;  
}  
  
public String getQtype() {  
    return qtype;  
}  
  
public void setQtype(String qtype) {  
    this.qtype = qtype;  
}  
  
public String getBody() {  
    return body;  
}  
  
public void setBody(String body) {  
    this.body = body;  
}  
  
public List<InquiryDTO> getInquiryDTOList() {  
    return inquiryDTOList;  
}  
  
public void setInquiryDTOList(List<InquiryDTO> inquiryDTOList) {  
    this.inquiryDTOList = inquiryDTOList;  
}  
}
```

---

[inquiryComplete.jsp](#)

---

⑤WebContent に inquiryComplete.jsp を作成し、以下をプログラミングしましょう。

---

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  
<%@ taglib prefix="s" uri="/struts-tags" %>  
<html>  
<head>  
<meta charset="UTF-8">  
<link rel="stylesheet" type="text/css" href="/css/style.css">
```

```
<title>お問い合わせ 受付完了</title>
</head>
<body>

<table>
<tbody>
  <tr>
    <th>名前</th>
    <th>お問い合わせの種類</th>
    <th>お問い合わせ内容</th>
  </tr>

  <s:iterator value="inquiryDTOList">
    <tr>

      <td><s:property value="name" /></td>

      <s:if test='qtype=="company"'>
        <td>会社について</td>
      </s:if>

      <s:if test='qtype=="product"'>
        <td>製品について</td>
      </s:if>

      <s:if test='qtype=="support"'>
        <td>アフターサポートについて</td>
      </s:if>

      <td><s:property value="body" /></td>
    </tr>
  </s:iterator>
</tbody>
</table>
</body>
</html>
```

---

⑥WebProj4 プロジェクトを実行してみましょう。

---

← → 🚫 💰 http://localhost:8080/WebProj4/

HelloStruts

Welcome

問い合わせ

← → 🚫 💰 http://localhost:8080/WebProj4/InquiryAction.action

名前: 山田太郎

お問い合わせの種類:

会社について ▼

お問い合わせ内容:

こんにちは。 ▲ ▼

登録

← → 🚫 💰 http://localhost:8080/WebProj4/InquiryCompleteAction.action

名前	お問い合わせの種類	お問い合わせ内容
山田太郎	会社について	こんにちは。



## Test 機能

---

index.jsp

---

①WebContent の index.jsp に以下を追加しましょう。

```
<h3>新規ユーザー登録</h3>
GET 通信
<s:form method="get" action="TestAction">
  <s:textfield name="username"/>
  <s:password name="password"/>
  <s:submit value="送信"/>
</s:form>
POST 通信
<s:form method="post" action="TestAction">
  <s:textfield name="username"/>
  <s:password name="password"/>
  <s:submit value="送信"/>
</s:form>
```

---

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="s" uri="/struts-tags" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>INDEX</title>
</head>
<body>
  <s:form action="HelloStrutsAction">
    <s:submit value="HelloStruts"/>
  </s:form>

  <s:form action="WelcomeAction">
    <s:submit value="Welcome"/>
  </s:form>

  <s:form action="InquiryAction">
    <s:submit value="問い合わせ"/>
```

```
</s:form>
<br>
<h3>新規ユーザー登録</h3>
GET 通信
<s:form method="get" action="TestAction">
<s:textfield name="username"/>
<s:password name="password"/>
<s:submit value="送信"/>
</s:form>
POST 通信
<s:form method="post" action="TestAction">
<s:textfield name="username"/>
<s:password name="password"/>
<s:submit value="送信"/>
</s:form>
</body>
</html>
```

---

struts.xml

---

②Java リソース¥src の struts.xml に以下を追加しましょう。

```
<action name="TestAction" class="com.internousdev.webproj4.action.TestAction" method="execute">
<result name="success">test.jsp</result>
</action>
```

---

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
"-//Apache Software Foundation//DTD Struts Configuration 2.3//EN"
"http://struts.apache.org/dtds/struts-2.3.dtd">
<struts>
<constant name="struts.devMode" value="true" />

<package name="com.internousdev.webproj4.action" extends="struts-default">

<action name="HelloStrutsAction" class="com.internousdev.webproj4.action.HelloStrutsAction"
method="execute">
<result name="success">hello.jsp</result>
<result name="error">helloStrutsError.jsp</result>
</action>
```

```
<action name="WelcomeAction" class="com.internousdev.webproj4.action.WelcomeAction" method="execute">
<result name="success">welcome.jsp</result>
</action>

<action name="LoginAction" class="com.internousdev.webproj4.action.LoginAction" method="execute">
<result name="success">login.jsp</result>
<result name="error">loginError.jsp</result>
</action>

<action name="InquiryAction" class="com.internousdev.webproj4.action.InquiryAction" method="execute">
<result name="success">inquiry.jsp</result>
</action>

<action name="InquiryCompleteAction" class="com.internousdev.webproj4.action.InquiryCompleteAction"
method="execute">
<result name="success">inquiryComplete.jsp</result>
</action>

<action name="TestAction" class="com.internousdev.webproj4.action.TestAction" method="execute">
<result name="success">test.jsp</result>
</action>
</package>
</struts>
```

---

TestDAO.java  
(TestA0.java)

---

③Java リソース¥src¥com.internousdev.webproj4.dao パッケージ  
に TestDAO.java を作成し、以下をプログラミングしましょう。

---

```
package com.internousdev.webproj4.dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
```

---

```
import com.internousdev.webproj4.dto.LoginDTO;
import com.internousdev.webproj4.util.DBConnector;

public class TestDAO {

    public List<LoginDTO> loginDTOList = new ArrayList<LoginDTO>();

    public int insert(String username, String password) {
        int ret = 0;
        DBConnector db = new DBConnector();
        Connection con = db.getConnection();

        String sql = "insert into users(user_name, password) values(?,?)";
        try {
            PreparedStatement ps = con.prepareStatement(sql);
            ps.setString(1, username);
            ps.setString(2, password);
            int i = ps.executeUpdate();
            if (i > 0) {
                System.out.println(i + "件登録されました");
                ret = i;
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        try {
            con.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return ret;
    }

    public List<LoginDTO> select(String username, String password) {

        DBConnector db = new DBConnector();
        Connection con = db.getConnection();

        String sql = "select * from users where user_name=? and password=?";
        try {
            PreparedStatement ps = con.prepareStatement(sql);
```

---

```

        ps.setString(1, username);
        ps.setString(2, password);
        ResultSet rs = ps.executeQuery();

        while (rs.next()) {
            LoginDTO dto=new LoginDTO();
            dto.setUsername(rs.getString("user_name"));
            dto.setPassword(rs.getString("password"));
            loginDTOList.add(dto);
        }

        if(loginDTOList.size()<=0){
            LoginDTO dto=new LoginDTO();
            dto.setUsername("該当なし");
            dto.setPassword("該当なし");
            loginDTOList.add(dto);
        }

    } catch (SQLException e) {
        e.printStackTrace();
    }
    try {
        con.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return loginDTOList;
}
}

```

---

TestAction  
(TestAction.java)

---

④Java リソース¥src¥com.internousdev.webproj4.action パッケージに TestAction.java を作成し、以下をプログラミングしましょう。

---

```
package com.internousdev.webproj4.action;
```

```

import java.util.ArrayList;
import java.util.List;

import com.internousdev.webproj4.dao.TestDAO;
import com.internousdev.webproj4.dto.LoginDTO;
import com.opensymphony.xwork2.ActionSupport;

public class TestAction extends ActionSupport {

    private String username;

    private String password;

    private List<LoginDTO> loginDTOList = new ArrayList<LoginDTO>();

    public String execute() {
        String ret = ERROR;

        TestDAO dao = new TestDAO();
        int count = dao.insert(username, password);

        if (count > 0) {
            ret = SUCCESS;
        } else {
            ret = ERROR;
        }

        loginDTOList = dao.select(username, password);

        return ret;
    }

    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;
    }

    public List<LoginDTO> getLoginDTOList() {
        return loginDTOList;
    }

    public void setLoginDTOList(List<LoginDTO> loginDTOList) {
        this.loginDTOList = loginDTOList;
    }
}
```

---

test.jsp

---

⑤WebContent に test.jsp を作成し、以下をプログラミングしましょう。

---

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="s" uri="/struts-tags" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<link rel="stylesheet" type="text/css" href="/css/style.css">
<title>TEST</title>
</head>
<body>
<br>
<h1>以下の新規ユーザーが登録されました。</h1>

<br>
<table>
<tbody>
<tr>
    <th>USERNAME</th>
    <th>PASSWORD</th>
</tr>
```

```

<s:iterator value="loginDTOList">
  <tr>
    <td><s:property value="username"/></td>
    <td><s:property value="password"/></td>
  </tr>
</s:iterator>

</tbody>
</table>

</body>
</html>

```

⑥WebProj4 プロジェクトを実行してみましょう。




以下の新規ユーザーが登録されました。

USERNAME	PASSWORD
saburo	234



以下の新規ユーザーが登録されました。

USERNAME	PASSWORD
shiro	345



エラーが発生しました