

<Struts2 を使ったプログラム>

JSP～Action クラス～DAO クラス～DTO クラス編

<動的 Web プロジェクトを作成する>

まず、動的プロジェクト「WebProj3」を作成しましょう。

WebProj3

<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>webproj3</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 (WebProj3) 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	

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.webproj3.action
com.internousdev.webproj3.dao
com.internousdev.webproj3.dto
com.internousdev.webproj3.util
```

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

③Java リソース¥src¥com.internousdev.webproj3.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.webproj3.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.webproj3.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.webproj3.dto;

public class HelloStrutsDTO {
    private String result;

    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.webproj3.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.webproj3.dao;

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

import com.internousdev.webproj3.dto.HelloStrutsDTO;
```

```

import com.internousdev.webproj3.util.DBConnector;

public class HelloStrutsDAO {

    public HelloStrutsDTO select() {
        DBConnector db = new DBConnector();
        Connection con = db.getConnection();
        HelloStrutsDTO dto=new HelloStrutsDTO();

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

            if (rs.next()) {
                dto.setResult("MySQL と接続できます。");
            }else{
                dto.setResult("MySQL と接続できません。");
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        try {
            con.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return dto;
    }
}

```

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.webproj3.action" extends="struts-default">

<action name="HelloStrutsAction" class="com.internousdev.webproj3.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.webproj3.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.webproj3.action;

import com.internousdev.webproj3.dao.HelloStrutsDAO;
import com.internousdev.webproj3.dto.HelloStrutsDTO;
import com.opensymphony.xwork2.ActionSupport;

public class HelloStrutsAction extends ActionSupport {

    private String result;

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

        dto=dao.select();

        System.out.println(dto.getResult());
    }
}
```

```
        result=dto.getResult();

        if(result.equals("MySQL と接続できます。")){
            ret=SUCCESS;
        }else{
            ret=ERROR;
        }
        return ret;
    }

    public String getResult() {
        return result;
    }

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

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。

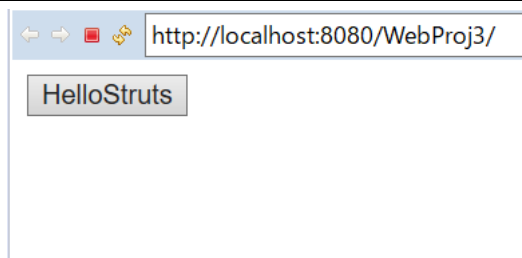
```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="s" uri="/struts-tags"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>HelloStruts</title>
</head>
<body>
<h1>HelloStruts2!</h1>
<br>
<h3><s:property value="result" /></h3>
</body>
</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>
<h3><s:property value="result" /></h3>
</body>
</html>
```

⑩WebProj3 プロジェクトを実行してみましょう。



HelloStruts2!

MySQLと接続できます。

http://localhost:8080/WebProj3/HelloStrutsAction.action

エラーが発生しました

MySQLと接続できません。

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.webproj3.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.webproj3.action.HelloStrutsAction"
method="execute">
<result name="success">hello.jsp</result>
</action>

<action name="WelcomeAction" class="com.internousdev.webproj3.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.webproj3.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.webproj3.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 method="post" 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.webproj3.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.webproj3.action" extends="struts-default">

<action name="HelloStrutsAction" class="com.internousdev.webproj3.action.HelloStrutsAction"
method="execute">
<result name="success">hello.jsp</result>
<result name="error">helloStrutsError.jsp</result>
</action>

<action name="WelcomeAction" class="com.internousdev.webproj3.action.WelcomeAction" method="execute">
<result name="success">welcome.jsp</result>
</action>

<action name="LoginAction" class="com.internousdev.webproj3.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.webproj3.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.webproj3.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.webproj3.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.webproj3.dao;
```

```

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

import com.internousdev.webproj3.dto.LoginDTO;
import com.internousdev.webproj3.util.DBConnector;

public class LoginDAO {

    public String username;
    public String password;

    public LoginDTO select(String username,String password) {
        DBConnector db = new DBConnector();
        Connection con = db.getConnection();
        LoginDTO dto=new LoginDTO();

        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();
            if (rs.next()) {
                dto.setUsername(rs.getString("user_name"));
                dto.setPassword(rs.getString("password"));
            }else{
                dto.setUsername("該当なし");
                dto.setPassword("該当なし");
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        try {
            con.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return dto;
    }
}

```

LoginAction
(LoginAction.java)

④Java リソース¥src¥com.internousdev.webproj3.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.webproj3.action;

import com.internousdev.webproj3.dao.LoginDAO;
import com.internousdev.webproj3.dto.LoginDTO;
import com.opensymphony.xwork2.ActionSupport;

public class LoginAction extends ActionSupport {
    private String username;
    private String password;

    public String execute() {
        String ret=ERROR;
        System.out.println(username);
        System.out.println(password);
        LoginDAO dao=new LoginDAO();
        LoginDTO dto=new LoginDTO();
        dto=dao.select(username, password);
        if(this.username.equals(dto.getUsername()) && this.password.equals(dto.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;  
    }  
}
```

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">  
<title>LOGIN</title>  
</head>  
<body>  
<s:property value="username"/>さん、ようこそ！  
</body>  
</html>
```

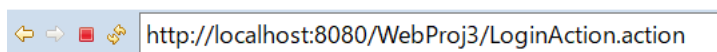
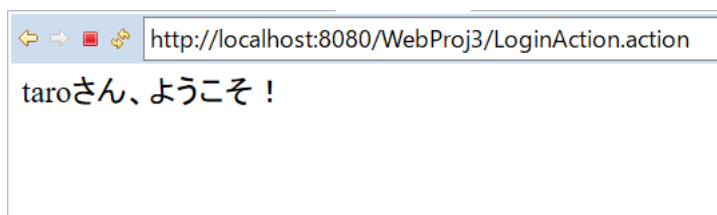
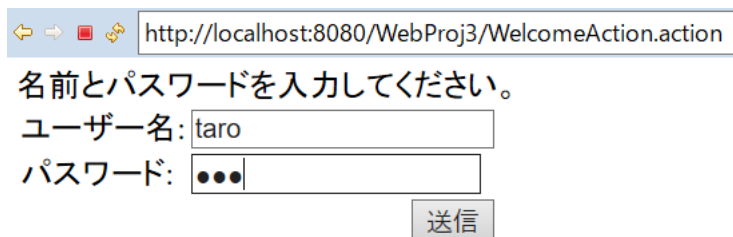
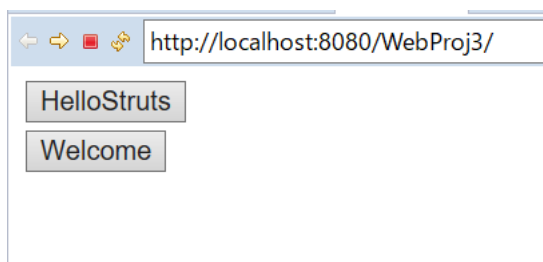
loginError.jsp

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"%>  
<!DOCTYPE>  
<html>  
<head>  
<meta charset="UTF-8">  
<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>
</body>
</html>
```

⑦WebProj3 プロジェクトを実行してみましょう。



ログインできません。

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.webproj3.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.webproj3.action" extends="struts-default">

<action name="HelloStrutsAction" class="com.internousdev.webproj3.action.HelloStrutsAction"
method="execute">
<result name="success">hello.jsp</result>
</action>

<action name="WelcomeAction" class="com.internousdev.webproj3.action.WelcomeAction" method="execute">
<result name="success">welcome.jsp</result>
</action>

<action name="LoginAction" class="com.internousdev.webproj3.action.LoginAction" method="execute">
<result name="success">login.jsp</result>
</action>

<action name="InquiryAction" class="com.internousdev.webproj3.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.webproj3.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.webproj3.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>
            <option value="support">アフターサポートについて</option>
        </select>
    </s:form>
</body>
</html>
```

```
</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.webproj3.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.webproj3.action" extends="struts-default">

<action name="HelloStrutsAction" class="com.internousdev.webproj3.action.HelloStrutsAction"
method="execute">
<result name="success">hello.jsp</result>
</action>

<action name="WelcomeAction" class="com.internousdev.webproj3.action.WelcomeAction" method="execute">
<result name="success">welcome.jsp</result>
</action>

<action name="LoginAction" class="com.internousdev.webproj3.action.LoginAction" method="execute">
<result name="success">login.jsp</result>
</action>

<action name="InquiryAction" class="com.internousdev.webproj3.action.InquiryAction" method="execute">
<result name="success">inquiry.jsp</result>
</action>
```

```
<action name="InquiryCompleteAction" class="com.internousdev.webproj3.action.InquiryCompleteAction"
method="execute">
<result name="success">inquiryComplete.jsp</result>
<result name="error">inquiry.jsp</result>
</action>
</package>
</struts>
```

InquiryDTO

(InquiryDTO.java)

②Java リソース¥src¥com.internousdev.webproj3.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.webproj3.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.webproj3.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.webproj3.dao;  
  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.SQLException;  
  
import com.internousdev.webproj3.util.DBConnector;  
  
public class InquiryCompleteDAO {  
    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.webproj3.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.webproj3.action;

import com.internousdev.webproj3.dao.InquiryCompleteDAO;
import com.opensymphony.xwork2.ActionSupport;

public class InquiryCompleteAction extends ActionSupport{

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

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

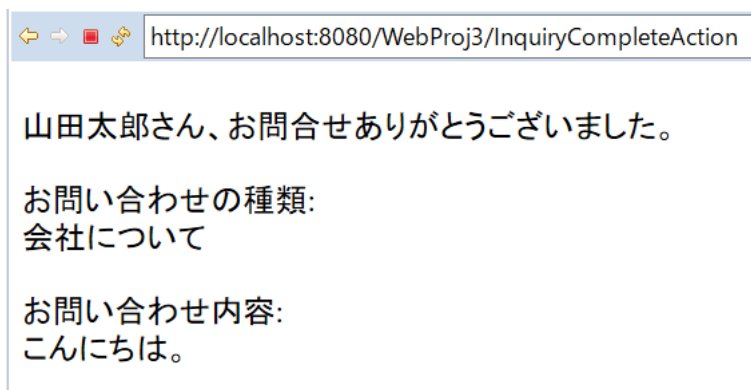
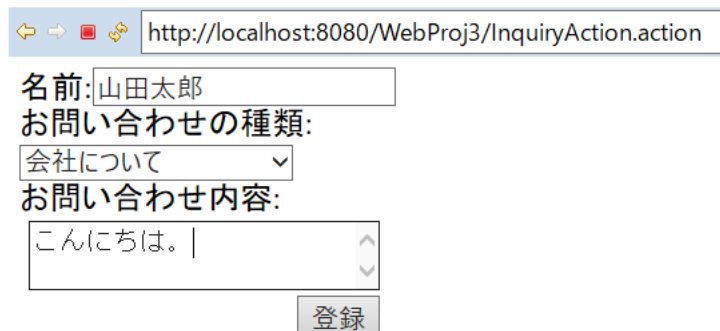
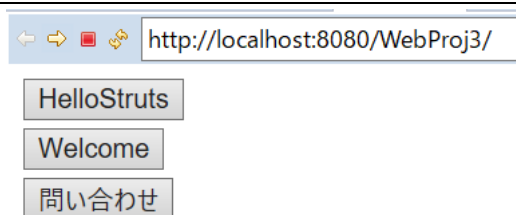
`inquiryComplete.jsp`

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">  
<title>お問い合わせ 受付完了</title>  
</head>  
<body>  
  
<br>  
<s:property value="name" />さん、お問合せありがとうございました。  
<br>  
<br>お問い合わせの種類:<br>  
<s:if test="qtype=='company'">
```

```
会社について
</s:if>
<s:if test=' qtype=="product"' >
製品について
</s:if>
<s:if test=' qtype=="support"' >
アフターサポートについて
</s:if>
<br>
<br>お問い合わせ内容:<br>
<s:property value="body" />
</body>
</html>
```

⑥WebProj3 プロジェクトを実行してみましょう。



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.webproj3.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.webproj3.action" extends="struts-default">

<action name="HelloStrutsAction" class="com.internousdev.webproj3.action.HelloStrutsAction"
method="execute">
<result name="success">hello.jsp</result>
<result name="error">helloStrutsError.jsp</result>
</action>
```

```
<action name="WelcomeAction" class="com.internousdev.webproj3.action.WelcomeAction" method="execute">
<result name="success">welcome.jsp</result>
</action>

<action name="LoginAction" class="com.internousdev.webproj3.action.LoginAction" method="execute">
<result name="success">login.jsp</result>
<result name="error">loginError.jsp</result>
</action>

<action name="InquiryAction" class="com.internousdev.webproj3.action.InquiryAction" method="execute">
<result name="success">inquiry.jsp</result>
</action>

<action name="InquiryCompleteAction" class="com.internousdev.webproj3.action.InquiryCompleteAction"
method="execute">
<result name="success">inquiryComplete.jsp</result>
</action>

<action name="TestAction" class="com.internousdev.webproj3.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.webproj3.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.webproj3.dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

import com.internousdev.webproj3.util.DBConnector;

public class TestDAO {
    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;
    }
}

```

TestAction
(TestAction.java)

④Java リソース¥src¥com.internousdev.webproj3.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.webproj3.action;

import com.internousdev.webproj3.dao.TestDAO;
import com.opensymphony.xwork2.ActionSupport;

public class TestAction extends ActionSupport{

```

```
private String username;

private String password;

public String execute() {
    String ret=ERROR;
    TestDAO dao=new TestDAO();
    int count=dao.insert(username, password);
    if(count>0) {
        ret=SUCCESS;
    }
    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;
}
}
```

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">
```

```
<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>
<s:property value="username"/>
<br>
<s:property value="password"/>
</body>
</html>
```

⑥WebProj3 プロジェクトを実行してみましょう。

← → 🏠 🔍 http://localhost:8080/WebProj3/

HelloStruts

Welcome

問い合わせ

新規ユーザー登録

GET通信

送信

POST通信

送信

← → 🏠 🔍 http://localhost:8080/WebProj3/TestAction.action?username=taro&password=123

以下の新規ユーザーが登録されました。

taro
123

← → 🏠 🔍 http://localhost:8080/WebProj3/TestAction.action

以下の新規ユーザーが登録されました。

taro
123