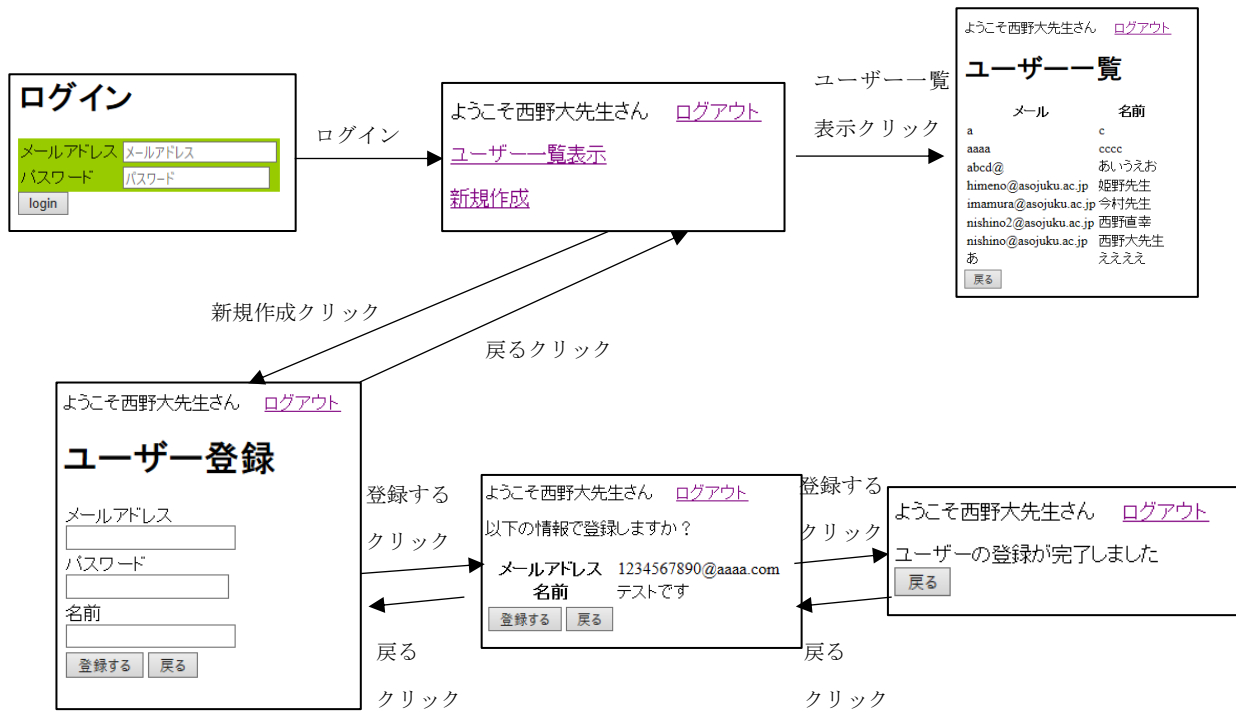


bmi_list2 プロジェクトの解説

サンプルの画面遷移です



bmi_list2 プロジェクトは Dao のサンプルプロジェクトです。以下、解説です。

bmi_list_v2

src

beans

controller

dao

filter

model

LoginInfoBeans.java

UserBeans.java

AuthServlet.java

CompleteUserRegistServlet.java

ConfirmUserRegistServlet.java

DispUserListServlet.java

InputUserRegistServlet.java

InsertUserRegistServlet.java

LoginServlet.java

LogoutServlet.java

MenuServlet.java

DaoBase.java

UserDao.java

EncodingFilter.java

UserModel.java

ファイル名		概要
LoginInfoBeans		ユーザー情報を保持するクラス
UserBeans		ユーザーの情報を格納するビーンズ
AuthServlet	C	ログイン認証を行うサーブレット
CompleteUserRegistServlet	C	ユーザー登録の画面量画面を表示する
ConfirmUserRegistServlet	C	ユーザー登録の確認画面のサーブレット
DispUserListServlet	C	ユーザーの一覧を表示するサーブレット
InputUserRegistServlet	C	ユーザー登録の入力画面サーブレット
LoginController	C	ログイン画面を表示するサーブレット
LogoutController	C	ログアウトを行うサーブレット
MenuServlet	C	メニューサーブレット
DaoBase	M	DAO のベースクラス
UserDao	M	ユーザーテーブルにアクセスする DAO
EncodingFilter	F	UTF-8 にエンコードするフィルター
UserModel	M	ユーザー機能に関するモデル

WebContent

css

main.css

META-INF

WEB-INF

jsp

header

lib

header.jsp

completeUser.jsp

confirmUser.jsp

dispUserList.jsp

inputUser.jsp

login.jsp

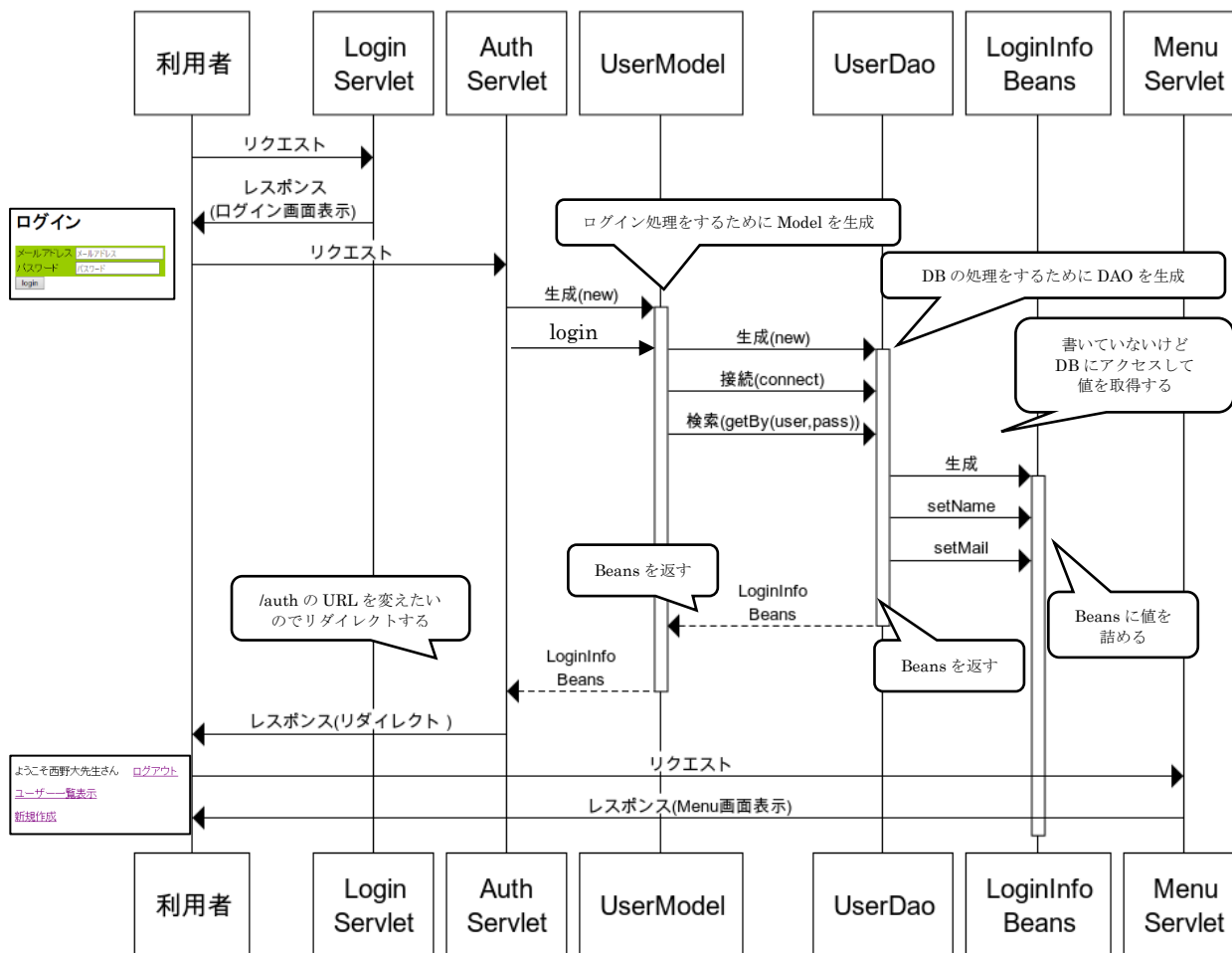
menu.jsp

mysql-connector-java-8.0.11.jar

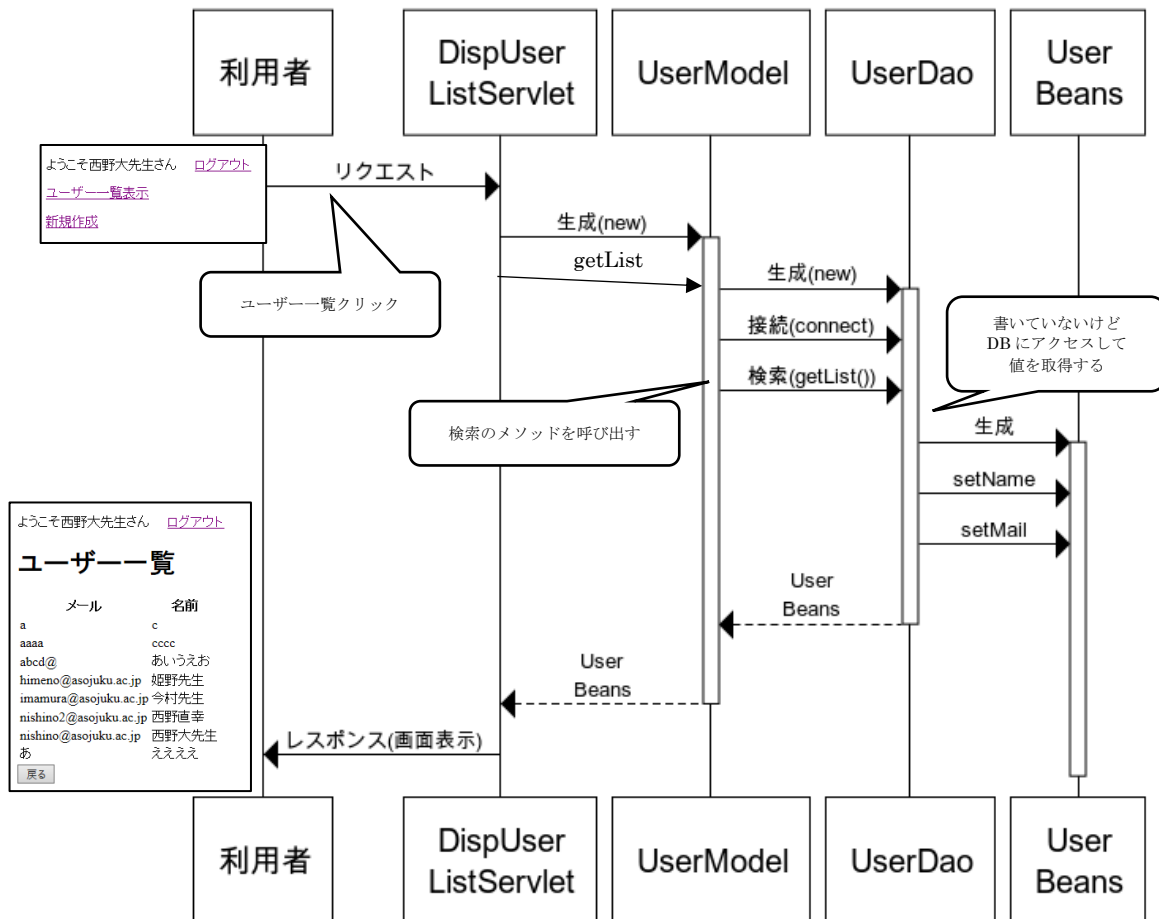
ファイル名		概要
header.jsp	V	各画面のヘッダー（上部）
completeUser.jsp	V	登録完了の画面
confirmUser.jsp	V	登録確認の画面
desipUserList.jsp	V	一覧表示の画面
inputUser.jsp	V	登録入力画面
login.jsp	V	ログイン画面
menu.jsp	V	ログイン後表示されるメニュー画面
main.css		CSS ファイル

シーケンス図

【ログイン】



【一覧表示】



【ログイン処理+UserModel、UserDAO】

① LoginServlet (ログイン画面を表示する)

```
@WebServlet("/login")
public class LoginServlet extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException
    {
        RequestDispatcher dispatcher =
            request.getRequestDispatcher("WEB-INF/jsp/login.jsp");
        dispatcher.forward(request, response);
    }
}
```

画面遷移 (転送)
login.jsp に処理を転送する

② login.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<% pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<link rel="stylesheet" href="css/main.css" type="text/css">
</head>
<body>
<% String msg = (String)request.getAttribute("errMsg"); %>
<% if(msg != null) %>
<p class="error"><%=msg %></p>
<% %>
<h1>ログイン</h1>
<form action="auth" method="post">
<table class="sample">
<tr>
<td>メールアドレス</td>
<td><input type="text" name="mail" placeholder="メールアドレス"></td>
</tr>
<tr>
<td>パスワード</td>
<td><input type="password" name="pass" placeholder="パスワード"></td>
</tr>
</table>
<input type="submit" value="login">
</form>
</body>
</html>
```

リクエストスコープからエラーメッセージを取得している
AuthServlet でエラーを検知したときに setAttribute される

if 文で分岐してエラーメッセージがあるときだけエラーメッセージを画面に表示するようにしている
「class="error"」は CSS を反映するため

action は次の画面を指定する auth と書くと
@WebServlet("auth")がついているサーブレットが
呼ばれる → AuthServlet へ

メールアドレスの入力ボックス

パスワードの入力ボックス

サブミットボタン
これをクリックすると以下の2つのことが起こる
・ form タグの action で指定したサーブレットが呼ばれる
・ form タグにかかっている input タグの情報がサーブレットに送信される

③ AuthServlet

```
/*  
 * ログイン認証 (Authentication)  
 *  
 * @author nishino  
 */  
@WebServlet("/auth")  
public class AuthServlet extends HttpServlet {  
  
    @Override  
    protected void doPost(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
  
        String mail = request.getParameter("mail");  
        String pass = request.getParameter("pass");  
  
        // モデル  
        UserModel loginModel = new UserModel();  
        LoginInfoBeans loginInfo = loginModel.login(mail, pass);  
  
        // 結果  
        if(loginInfo != null) {  
            // セッションに保存  
            HttpSession session = request.getSession(true);  
            session.setAttribute("loginInfo", loginInfo);  
            response.sendRedirect("menu");  
        } else {  
            // エラーのとき  
            request.setAttribute("errMsg", "メールアドレスかパスワードが違います");  
            RequestDispatcher dispatcher =  
                request.getRequestDispatcher("WEB-INF/jsp/login.jsp");  
            dispatcher.forward(request, response);  
        }  
    }  
}
```

login.jsp でサブミットボタンをクリックされたら呼ばれる

login.jsp で method=POST だったので doPost が呼ばれる

isp から値を取得 "mail" は isp の name 属性の値

isp から値を取得 "pass" は isp の name 属性の値

login のメソッドが LoginInfoBeans (ログイン結果) を返すので受け取る (LoginInfoBeans が null ならログイン失敗、null 以外ならログイン成功)

ログインをするために LoginModel のメソッドを呼び出す。インスタンス変数.メソッド名でメソッドを呼び出す

UserModel の login メソッド呼び出し
インスタンス変数.メソッド名

ログイン成功したときの処理。
セッションにログイン情報を保存する

menu へリダイレクト。
これが実行されると MenuServlet が呼ばれる

ログイン失敗なのでエラーメッセージをセットしている

ログイン画面を表示

④ UserModel

```
/**  
 * ユーザー情報のモデル  
 *  
 * @author nishino  
 */  
public class UserModel {
```

DisplayUserListServlet から呼び出される

```
/**  
 * ユーザー情報の一覧を取得する  
 *  
 * @return  
 */  
public List<UserBeans> getList() {  
    List<UserBeans> list = new ArrayList<UserBeans>();  
    UserDao userDao = new UserDao();  
    try {  
        //接続  
        userDao.connect();  
        //値を検索  
        list = userDao.getList();  
    } finally {  
        userDao.close();  
    }  
    return list;  
}
```

getList メソッドでは、DB の user_tbl から全件取得する。DB に関する処理は UserDao にあるので、実質的に UserDao のインスタンスを作って UserDao の getList を呼び出すだけ

UserDAO の getList メソッド呼び出し
インスタンス変数.メソッド名

```
/**  
 * ログイン処理  
 *  
 * @param mail  
 * @param password  
 * @return null以外: ログイン成功 | null: ログイン失敗  
 */  
public LoginInfoBeans login(String mail, String password) {  
    LoginInfoBeans userBeans = null;  
    UserDao userDao = new UserDao();  
    try {  
        //接続  
        userDao.connect();  
        //値を検索  
        userBeans = userDao.getBy(mail, password);  
    } finally {  
        userDao.close();  
    }  
    return userBeans;  
}
```

login メソッドでは、ログイン処理に必要な DB 処理を行う
DB にアクセスする部分は DAO に任せる

UserDAO の getBy メソッド呼び出し
インスタンス変数.メソッド名

AuthServlet から呼び出される

```
/**  
 * 引数のビーンズの値を登録する  
 *  
 * @param userBeans  
 */  
public void insert(UserBeans userBeans) {  
    UserDao userDao = new UserDao();  
    try {  
        //接続  
        userDao.connect();  
        //登録処理  
        userDao.insert(userBeans);  
    } finally {  
        userDao.close();  
    }  
}
```

InsertUserRegistServlet から
呼び出される

UserDAO の insert メソッド呼び出し
インスタンス変数.メソッド名

UserDao

```
/**  
 * ユーザーテーブルのDAO  
 *  
 *  
 * @author nishino  
 */  
public class UserDao extends DaoBase {
```

```
    /**  
     * ユーザーの一覧情報を取得する  
     *  
     * @return  
     */  
    public List<UserBeans> getList() {  
        > if( con == null ) {  
        >     return null;  
        > }  
        > List<UserBeans> list = new ArrayList<UserBeans>();  
  
        PreparedStatement stmt = null;  
        ResultSet rs = null;  
  
        try {  
            > > //////////////////////////////////////  
            > > //SELECT文の発行  
            > > stmt = con.prepareStatement("SELECT * FROM user_tbl");  
  
            > > rs = stmt.executeQuery();  
  
            > > //////////////////////////////////////  
            > > //DBから値を取得  
            > > while( rs.next() ){  
            > >     UserBeans userBeans = new UserBeans();  
  
            > >     > userBeans.setMail(rs.getString("mail"));  
            > >     > userBeans.setName(rs.getString("name"));  
  
            > >     > list.add(userBeans);  
            > >     }  
  
        } catch(SQLException e) {  
            > > e.printStackTrace();  
        }  
        > > return list;  
    }  
}
```

UserModel の getList メソッド
から呼び出される

```
    /**  
     * ユーザー情報を登録する  
     *  
     * @param userBeans  
     */  
    public void insert(UserBeans userBeans) {  
        > if( con == null ) {  
        >     return;  
        > }  
  
        PreparedStatement stmt = null;  
  
        try{  
            > > //////////////////////////////////////  
            > > //SELECT文の発行  
            > > stmt = con.prepareStatement("INSERT INTO user_tbl(mail,password,name) VALUES(?,?,?)");  
  
            > > stmt.setString(1, userBeans.getMail());  
            > > stmt.setString(2, userBeans.getPassword());  
            > > stmt.setString(3, userBeans.getName());  
  
            > > stmt.executeUpdate();  
  
        } catch(SQLException e) {  
            > > //エラー発生した場合にコンソールにログを出力する  
            > > e.printStackTrace();  
        }  
    }  
}
```

InsertUserRegistServlet から
呼び出される


```

/**
 * 指定した情報で検索をして
 * ユーザー情報をDBより取得する
 */
@param mail
@param password
@return
*/
public LoginInfoBeans getBy(String mail,String password){
    > if( con == null ) {
    >     > return null;
    > }
    > LoginInfoBeans loginInfo = null;

    PreparedStatement stmt = null;
    ResultSet rs = null;

    try {
        > > ///////////////////////////////////
        > > //SELECT文の発行
        > > stmt = con.prepareStatement("SELECT * FROM user_tbl WHERE mail=? and password=?");

        > > stmt.setString(1, mail);
        > > stmt.setString(2, password);
        > > rs = stmt.executeQuery();

        > > ///////////////////////////////////
        > > //DBから値を取得
        > > while( rs.next() ){
        > >     > loginInfo = new LoginInfoBeans();

        > > > loginInfo.setMail(rs.getString("mail"));
        > > > loginInfo.setUserName(rs.getString("name"));
        > > }

    } catch(SQLException e) {
        > > e.printStackTrace();
    }

    > > return loginInfo;
}

```

UserModel の login メソッド
から呼び出される

DB から SELECT した結果を取得
"mail" はテーブルの列名

DB から SELECT した結果を取得
"name" はテーブルの列名

【一覧表示処理】

○DisplayUserListServlet

一覧表示を行うサーブレット。一覧情報の取得は UserModel の getList メソッドが行うため、UserModel のインスタンスを作成して、getList メソッドを呼び出している。その値を dispUserList.jsp で表示するためにリクエストスコープに値を保存している

```
@WebServlet("/userList")
public class DispUserListServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        ///////////////////////////////////////////////////
        //一覧の取得
        UserModel userModel = new UserModel();

        List<UserBeans> userList = userModel.getList();

        //リクエストに一覧を保存する
        request.setAttribute("userList", userList);

        ///////////////////////////////////////////////////
        //画面遷移
        RequestDispatcher dispatcher =
            request.getRequestDispatcher("/WEB-INF/jsp/dis");
        dispatcher.forward(request, response);
    }
}
```

UserDAO の getList メソッド呼び出し
インスタンス変数.メソッド名

"userList"という箱に userList という
変数の値を set している

○dispUserList.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"%>
<%@ page encoding="UTF-8"%>
<%@ page import="beans.UserBeans"%>
<%@ page import="java.util.List"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<%
    List<UserBeans> userList = (List<UserBeans>)request.getAttribute("userList");
    <jsp:include page="/header/header.jsp"/>
    <h1>ユーザー一覧</h1>
    <table>
    <tr>
    <th>メール</th>
    <th>名前</th>
    </tr>
    <% for( UserBeans userBeans : userList ){ %>
    <tr>
    <td><%= userBeans.getMail() %></td>
    <td><%= userBeans.getName() %></td>
    </tr>
    <% } %>
    </table>
    <input type="button" value="戻る" onClick="javascript:location.href='menu'">
</body>
</html>
```

"userList"という箱から値を get している

header.jsp をここに展開する

拡張 for 文をつかっている

こう書くことで、クリック時に location.href (画面遷移する JavaScript) を呼び出している

【登録処理】

InputUserRegistServlet

入力画面を表示するサーブレット。画面を表示するだけ

```
@WebServlet("/inputuser")  
public class InputUserRegistServlet extends HttpServlet {  
    @Override  
    protected void doGet(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
        RequestDispatcher dispatcher =  
            request.getRequestDispatcher("/WEB-INF/jsp/inputUser.jsp");  
        dispatcher.forward(request, response);  
    }  
}
```

inputUser.jsp

```
<body>  
<jsp:include page="/header/header.jsp"/>  
<h1>ユーザー登録</h1>  
<form action="confirmuser" method="POST">  
    <table>  
        <tr>  
            <td>  
                メールアドレス<br>  
                <input type="text" name="mail">  
            </td>  
        </tr>  
        <tr>  
            <td>  
                パスワード<br>  
                <input type="password" name="password">  
            </td>  
        </tr>  
        <tr>  
            <td>  
                名前<br>  
                <input type="text" name="name">  
            </td>  
        </tr>  
        <tr>  
            <td>  
                <input type="submit" value="登録する">  
                <input type="button" value="戻る" onClick="javascript:location.href='menu'">  
            </td>  
        </tr>  
    </table>  
</form>  
</body>  
</html>
```

header.jsp をここに展開する

ConfirmUserRegistServlet

確認画面を表示するサーブレット。ここでは、値をビーンズに入れなおす。このビーンズは確認画面と InsertUserRegistServlet で使用する（つまりリクエストをまたいで使用する）のでセッションに保存する

```
@WebServlet("/confirmuser")
public class ConfirmUserRegistServlet extends HttpServlet {

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        // 値の取得
        String mail = request.getParameter("mail");
        String password = request.getParameter("password");
        String name = request.getParameter("name");

        // ビーンズに値をセットする
        UserBeans userBeans = new UserBeans();

        userBeans.setMail(mail);
        userBeans.setPassword(password);
        userBeans.setName(name);

        // セッションに保存（本来は保存の前に入力チェックをする）
        HttpSession session = request.getSession();
        session.setAttribute("userBeans", userBeans);

        // 画面遷移
        RequestDispatcher dispatcher =
            request.getRequestDispatcher("/WEB-INF/jsp/confirmUser.jsp");
        dispatcher.forward(request, response);
    }
}
```

確認画面と InsertUserRegistServlet の 2 か所で使用するのでセッションに保存する

confirmUser.jsp

```
<body>
<%
    UserBeans userBeans = (UserBeans)session.getAttribute("userBeans");
%>
<jsp:include page="./header/header.jsp"/>
<p>以下の情報で登録しますか？</p>

<form action="insertuser" method="POST">
<table>
<tr>
<th>メールアドレス</th>
<td><%=userBeans.getMail() %></td>
</tr>
<tr>
<th>名前</th>
<td><%=userBeans.getName() %></td>
</tr>
<tr>
<td>
<input type="submit" value="登録する">
<input type="button" value="戻る" onClick="javascript:location.href='inputuser'">
</td>
</tr>
</table>
</form>
</body>
</html>
```

セッションから取得する

header.jsp をここに展開する

InsertUserRegistServlet

登録をするサーブレット。登録する値は確認の時にセッションに保存した値を取得する

ConfirmUserRegistServlet の session.setAttribute の値を取得

```
////////////////////////////////////  
//セッションに保存 (本来は保存の前に入力チェックをする)  
HttpSession session = request.getSession();  
session.setAttribute("userBeans", userBeans);  
////////////////////////////////////
```

```
@WebServlet("/insertuser")  
public class InsertUserRegistServlet extends HttpServlet {  
  
    @Override  
    protected void doPost(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException  
    {  
  
        //////////////////////////////////////  
        //セッションから値を取得  
        HttpSession session = request.getSession();  
        UserBeans userBeans = (UserBeans)session.getAttribute("userBeans");  
  
        //////////////////////////////////////  
        //登録処理  
        UserModel userModel = new UserModel();  
  
        userModel.insert(userBeans);  
  
        //////////////////////////////////////  
        //セッション情報を削除  
        session.removeAttribute("userBeans");  
  
        //////////////////////////////////////  
        //完了画面へリダイレクト  
        response.sendRedirect("completeuser");  
    }  
}
```

セッションから取得する

UserDAO の insert メソッド呼び出し
インスタンス変数.メソッド名

転送 (forward) ではなくリダイレクトを行う。
リダイレクトを行うことで完了画面を表示するときの URL が /completeuser
となる。転送すると完了画面の URL が /insertuser となり
リロードをしたときにもう一度登録処理が走ってしまい、エラーとなる。

CompleteUserRegistServlet

完了画面を表示するサーブレット。画面を表示するだけ

```
@WebServlet("/completeuser")  
public class CompleteUserRegistServlet extends HttpServlet {  
    @Override  
    protected void doGet(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException  
    {  
        RequestDispatcher dispatcher =  
            request.getRequestDispatcher("/WEB-INF/jsp/completeUser.jsp");  
        dispatcher.forward(request, response);  
    }  
}
```

completeUser.jsp

```
<html>  
<head>  
    <meta charset="UTF-8">  
    <title>Insert title here</title>  
</head>  
<body>  
    <jsp:include page="/header/header.jsp"/>  
    ユーザーの登録が完了しました<br>  
    <input type="button" value="戻る" onClick="javascript:location.href='menu'">  
</body>  
</html>
```