

## DB サンプル登録編

### 【完成版の動作】

このサンプルが完成すると先日作った `user_tbl` に値を「挿入」できるようになります。

1 ページ目

メールアドレス	<input type="text"/>
パスワード	<input type="password"/>
名前	<input type="text"/>
<input type="button" value="登録"/>	

メールアドレス、パスワード、名前を入力して登録ボタンをクリック。

成功すると

登録に成功しました！

と表示されます。登録に失敗すると「エラーが発生しました」と表示されます。

### 【手順】

以下の手順で作成します。

1. 新たなプロジェクト「DBInsert」を作成する
2. 1 ページのコントローラ（サーブレット）とビュー（JSP）を作成する
3. jdbc ドライバーをプロジェクトにコピーしてビルドパスに追加する
4. DB への挿入処理を行うモデルを作成する
5. 2 ページ目のコントローラ（サーブレット）とビュー（JSP）を作成する

追って、具体的な修正箇所を示します。

## 1. 新たなプロジェクト「DBInsert」を作成する

※説明省略（ワークスペースは任意だが、どのワークスペースに作成したかは覚えておこう）

## 2. 1 ページのコントローラ（サーブレット）とビュー（JSP）を作成する

### ●コントローラの作成

以下のような実装にする（いつものやつ）

※パッケージは `dbinsert.controller` にしましょう

```
1 package dbinsert.controller;
2
3 import java.io.IOException;
4
11 @WebServlet("/page1")
12 public class Page1Servlet extends HttpServlet {
13
14
15
16     @Override
17     protected void doGet(HttpServletRequest req, HttpServletResponse res)
18                             throws ServletException, IOException {
19
20         //////////////////////////////////////
21         //画面遷移
22         RequestDispatcher dis=req.getRequestDispatcher("WEB-INF/jsp/page1.jsp");
23         dis.forward(req, res);
24     }
25
26 }
```

### ●JSP の作成

1 ページは、入力画面です。table タグを使って整形しましょう。

```

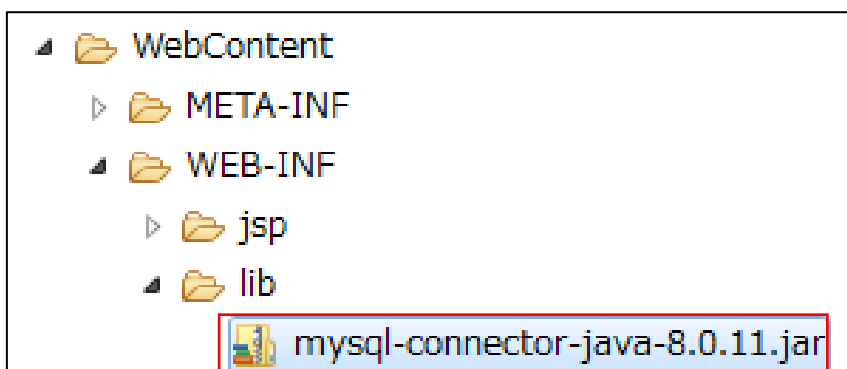
1 <%@ page language="java" contentType="text/html; charset=UTF-8"%>
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http
4 <html>
5 <head>
6 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"%>
7 <title>登録画面</title>
8 </head>
9 <body>
10 <form action="page2" method="GET">
11   <table border="1">
12     <tr>
13       <td>メアド</td>
14       <td><input type="text" name="mail"></td>
15     </tr>
16     <tr>
17       <td>パスワード</td>
18       <td><input type="password" name="password"></td>
19     </tr>
20     <tr>
21       <td>名前</td>
22       <td><input type="text" name="name"></td>
23     </tr>
24   </table>
25   <input type="submit" value="登録">
26 </form>
27 </body>
28 </html>
29

```

3. jdbc ドライバーをプロジェクトにコピーしてビルドパスに追加する

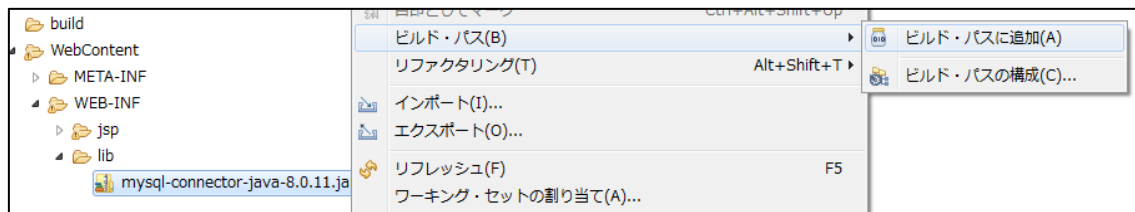
以前プロジェクト、又は GitHub にある JDBC ドライバ（mysql-connector-java-8.0.11.jar）を、プロジェクトの WebContent-WEB-INF-lib フォルダにコピーします。

↓こんな感じ。



さらに、追加した **JDBC** ドライバをビルドパスに追加します

「**JDBC** ドライバを選択」 — 「右クリック」 — 「ビルドパス」 — 「ビルドパスに追加」  
です。



※このあたりの作業も覚えよう！

#### 4. DB への挿入処理を行うモデルを作成する

DB を扱うモデルを作成します。

接続の部分は、以前の SELECT と同じですが、発行する SQL と **SQL を実行する時のメソッドが違**うことに注意です。

また、更新の場合は「更新する情報」をコントローラから受け取るため、モデルの引数が必要になる場合が多いです。

今回、**エラーが起こった場合は発生した例外をコントローラ側にスローしている**ことにも注意してください。

※パッケージは `dbinsert.controller` にしましょう

```
1 package dbinsert.model;
2
3 import java.sql.Connection;
4
5
6 public class DBModel {
7
8     /**
9      * 指定された情報をユーザーテーブルに挿入する
10     *
11     * @param mail
12     * @param password
13     * @param name
14     * @throws ClassNotFoundException
15     * @throws SQLException
16     */
17     public void insert(String mail, String password, String name)
18         throws ClassNotFoundException, SQLException {
19
20         Connection con = null;
21         PreparedStatement stmt = null;
22
23         try {
24             //DBの接続
25             Class.forName("com.mysql.cj.jdbc.Driver");
26
27             con = DriverManager.getConnection(
28                 "jdbc:mysql://localhost:3306/webtestdb?characterEncoding=UTF-8&serverTimezone=JST",
29                 "root", "自分のパスワード");
30
31             //INSERT文の発行
32             stmt = con.prepareStatement("INSERT INTO user_tbl(mail,password,name) VALUES(?,?,?)");
33
34             stmt.setString(1, mail);
35             stmt.setString(2, password);
36             stmt.setString(3, name);
37             stmt.executeUpdate();
38
39         } catch (ClassNotFoundException e) {
40             e.printStackTrace();
41             throw e;
42         } catch (SQLException e) {
43             e.printStackTrace();
44             throw e;
45         } finally {
46             if (con != null) {
47                 try {
48                     con.close();
49                 } catch (SQLException e) {
50                     e.printStackTrace();
51                 }
52             }
53         }
54     }
55 }
```

5. 2 ページ目のコントローラ（サーブレット）とビュー（JSP）を作成する

2 ページ目のコントローラで、DBModel を呼び出して、2 ページ目へ遷移します。

その流れを作っていきます。

## ●2 ページ目のコントローラ

```
1 package dbinsert.controller;
2
3 import java.io.IOException;
14
15 @WebServlet("/page2")
16 public class Page2Servlet extends HttpServlet {
17
18     @Override
19     protected void doGet(HttpServletRequest request, HttpServletResponse response)
20         throws ServletException, IOException {
21
22         //////////////////////////////////////
23         // JSPから値を取得する
24         String mail = request.getParameter("mail");
25         String password = request.getParameter("password");
26         String name = request.getParameter("name");
27
28         //////////////////////////////////////
29         // JSPで入力した情報をデータベースへセット
30         String message = "";
31         try {
32             DBModel dbModel = new DBModel();
33
34             dbModel.insert(mail, password, name);
35
36             message = "登録に成功しました！";
37         } catch (ClassNotFoundException e) {
38             message = "エラーが発生しました";
39         } catch (SQLException e) {
40             message = "エラーが発生しました";
41         }
42
43         //////////////////////////////////////
44         // リクエストスコープにメッセージをセット
45         request.setAttribute("message", message);
46
47         //////////////////////////////////////
48         // 画面遷移
49         RequestDispatcher dis = request.getRequestDispatcher("WEB-INF/jsp/page2.jsp");
50         dis.forward(request, response);
51     }
52 }
53
54
```

## ●2 ページ目の JSP

```
1  <%@ page language="java" contentType="text/html; charset=UTF-8"%>
2  <% pageEncoding="UTF-8"%>
3  <%@ page import="java.util.List"%>
4  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http
5  <html>
6  <head>
7  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"%>
8  <title>2ページ目</title>
9  </head>
10 <body>
11 <%
12     //リクエストスコープからメッセージを取得する
13     String message = (String)request.getAttribute("message");
14 %>
15 <p><%=message %></p>
16 </body>
17 </html>
```

ここまで記述したら完成です。

MySQL を起動して、実行してみましょう！



シーケンスは以下のようになる

