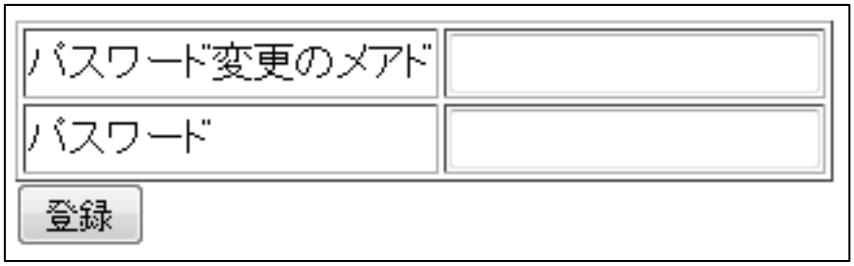
DB サンプル更新編

【完成版の動作】

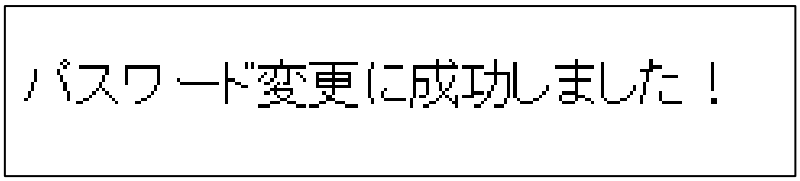
このサンプルが完成すると先日作った user_tbl のパスワードを「更新」できるようになります。

1 ページ目



メアド、パスワードを入力して登録ボタンをクリック。

成功すると、入力したメアドに設定されているパスワードが更新されます。



と表示されます。更新に失敗すると「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「DBUpdate」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「DBUpdate」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 dbupdate.controller にしましょう

```java
package dbupdate.controller;

import java.io.IOException;

@WebServlet("/page1")
public class Page1Servlet extends HttpServlet {


    @Override
    protected void doGet(HttpServletRequest req,HttpServletResponse res)
                                        throws ServletException,IOException{

        ////////////////////////////////////
        //画面遷移
        RequestDispatcher dis=req.getRequestDispatcher("WEB-INF/jsp/page1.jsp");
        dis.forward(req, res);
    }

}
```

●JSP の作成

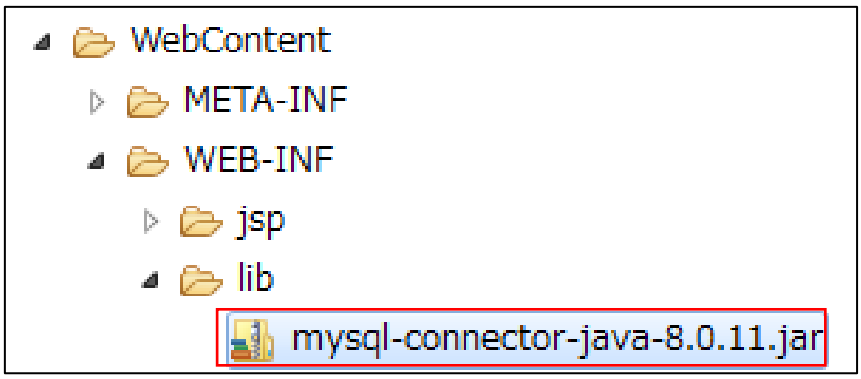
1 ページは、入力画面です。table タグを使って整形しましょう。

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http:
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>パスワード変更画面</title>
</head>
<body>
<form action="page2" method="GET">
    <table border="1">
        <tr>
            <td>パスワード変更のメアド</td>
            <td><input type="text" name="mail"></td>
        </tr>
        <tr>
            <td>パスワード</td>
            <td><input type="password" name="password"></td>
        </tr>
    </table>
    <input type="submit" value="登録">
</form>
</body>
</html>
```

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

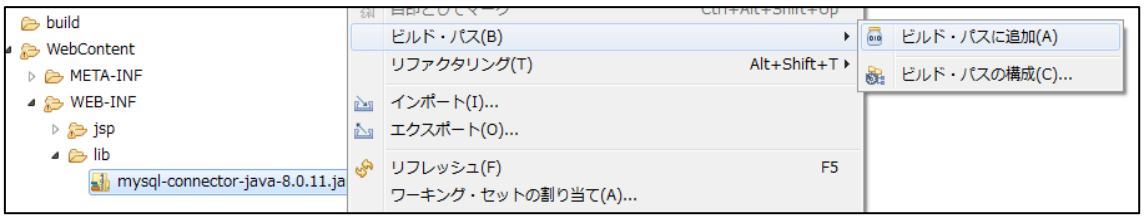
以前プロジェクト、又は GitHub にある JDBC ドライバ（mysql-connector-java-8.0.11.jar）を、プロジェクトの WebContent－WEB-INF－lib フォルダにコピーします。

↓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 dbupdate.controller にしましょう

```
package dbupdate.model;

import java.sql.Connection;

public class DBModel {


    /**
     * パスワード更新処理
     *
     * @param mail
     * @param password
     * @throws ClassNotFoundException
     * @throws SQLException
     */
    public void update(String mail,String password)
                        throws ClassNotFoundException, SQLException{

        Connection con = null;
        PreparedStatement stmt = null;

        try{
            ////////////////////////////////////
            //DBの接続
            Class.forName("com.mysql.cj.jdbc.Driver");

            con = DriverManager.getConnection(
                    "jdbc:mysql://localhost:3306/webtestdb?characterEncoding=UTF-8&serverTimezone=JST",
                    "root","password");

            ////////////////////////////////////
            //INSERT文の発行
            stmt = con.prepareStatement("UPDATE user_tbl SET password=? WHERE mail=?");

            stmt.setString(1, password);
            stmt.setString(2, mail);
            stmt.executeUpdate();

        }catch(ClassNotFoundException e) {
            e.printStackTrace();
            throw e;
        }catch(SQLException e) {
            e.printStackTrace();
            throw e;
        }finally {
            if(con!=null) {
                try {
                    con.close();
                }
                catch(SQLException e) {
                    e.printStackTrace();
                }
            }
        }

    }
}
```

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。

その流れを作って行きます。

●2 ページ目のコントローラ

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

●2 ページ目の JSP

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

※早く終わった人は、以下の問題にたいしてどうしたら良いか考えてみよう
　1 ページ目で存在しないメールアドレスを入力しても「更新に成功しました」が表示される

シーケンスは以下のようになる