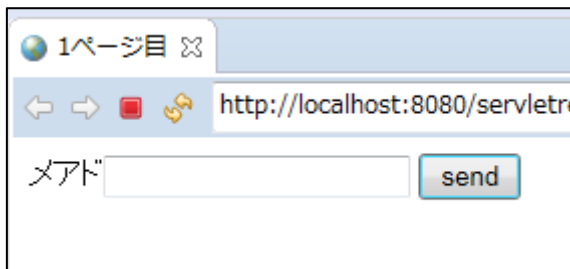


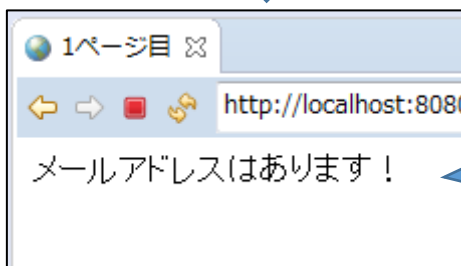
タイトル	
メールアドレスの検索！	
レベル	★★★

授業の例であった、メールアドレスの検索を作ってみよう！

【仕様】



メールアドレスを入力してボタンをクリックすると、入力したメールアドレスをDBより検索をする

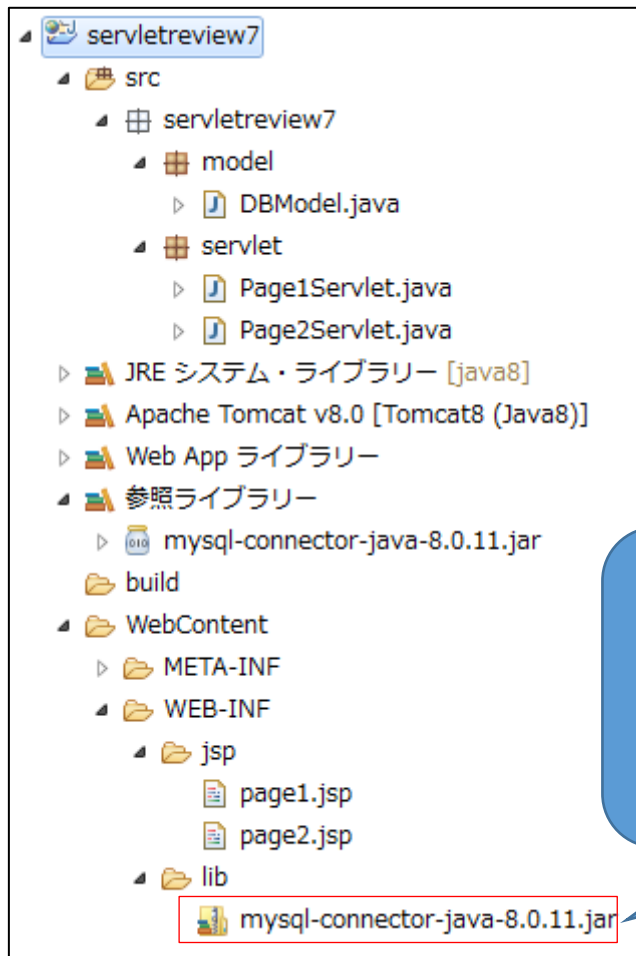
メールアドレスが存在する  
場合は「メールアドレスがあります！」  
ない場合は「メールアドレスはありません・・・」  
を表示する

検索結果を表示する

早く終わった人は↓もチャレンジ！

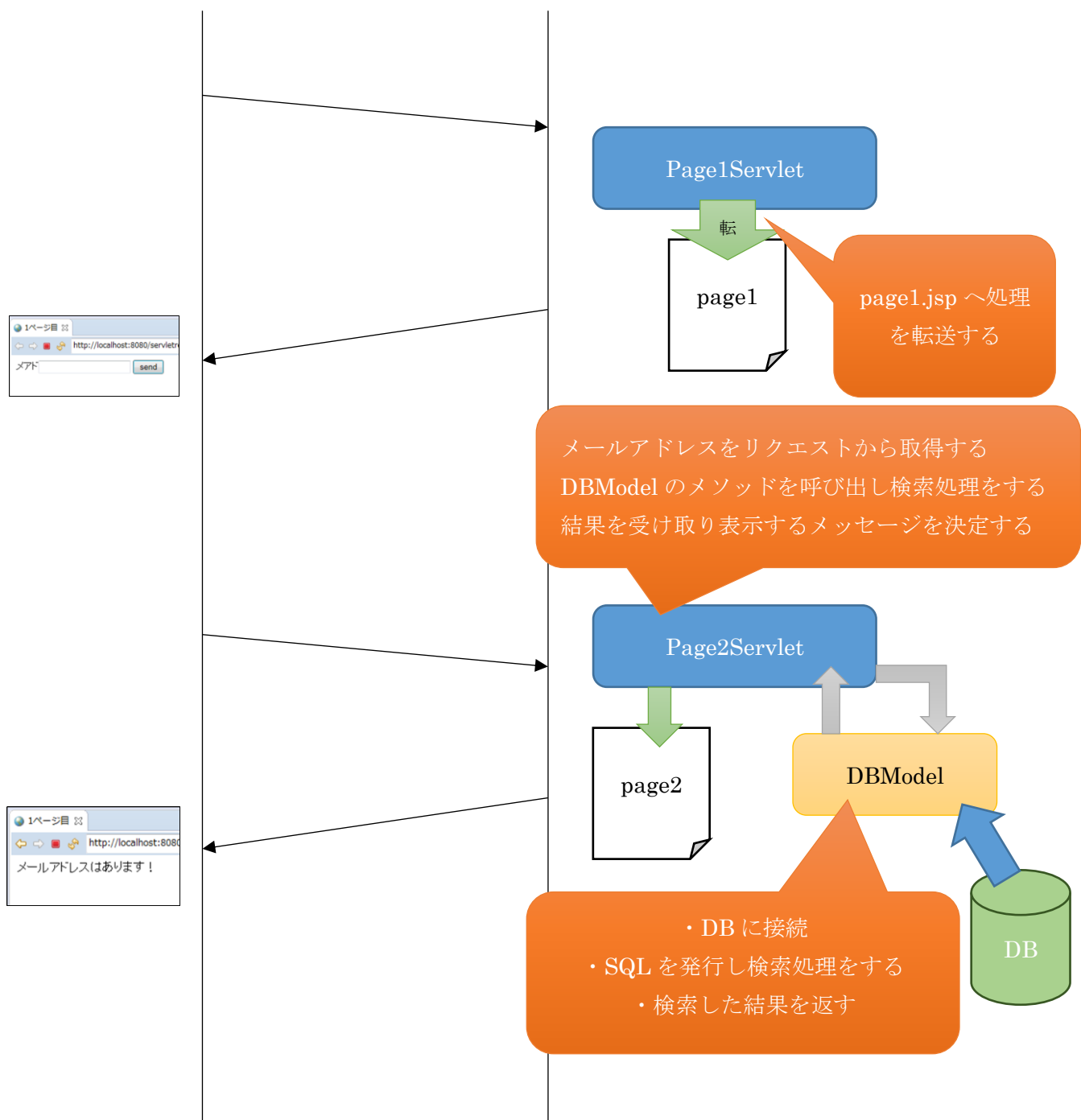
内容	難易度
入力したメールアドレスを「完全一致」ではなく「部分一致」で検索する	★★★★★

プロジェクトのクラス、JSP は以下のように配置しましょう！



JDBC ドライバのコピー&ビルドパスへの追加を忘れずにやってね！  
忘れた人は [GitHub の 2018Web / 資料 / 前期分 / DB サンプル-初級編.pdf](#) を見てみよう！

パッケージ	クラス名	内容	役割
servletreview7.servlet	Page1Servlet	・ page1.jsp へ処理を転送する	C
servletreview7.servlet	Page2Servlet	・ page1 からの情報を取得する ・ DB 取得用の Model を呼び出して検索結果を取得する ・ 検索結果から page2.jsp に送るメッセージを決定し、リクエストに保存する ・ page2.jsp へ処理を転送する	C
servletreview7.model	DBModel	・ メールアドレスを受け取り、DB に接続し検索をする。検索結果を返す。	M



※わからない人は次ページをヒントにしよう

## DBModel

```
1 package servletreview7.model;
2
3 import java.sql.Connection;
4
5 public class DBModel {
6
7     /**
8      * 指定されたメールアドレスをDBから検索する
9      *
10     * @param (引数) mail メールアドレス
11     * @return メールアドレスがあった場合はtrue、無ければfalse
12     */
13     public boolean findByMail(String mail){
14         boolean result = false;
15         Connection con = null;
16         PreparedStatement stmt = null;
17         ResultSet rs = null;
18
19         try{
20             ////////////////////////////////////
21             //DBの接続
22             Class.forName("com.mysql.cj.jdbc.Driver");
23
24             con = DriverManager.getConnection(
25                 "jdbc:mysql://localhost:3306/webtestdb?characterEncoding=UTF-8&serverTimezone=JST",
26                 "root", "password");
27
28             ////////////////////////////////////
29             //SELECT文の発行
30             stmt = con.prepareStatement("SELECT * FROM user_tbl WHERE mail=?");
31
32             stmt.setString(1, mail);
33             rs = stmt.executeQuery();
34
35             ////////////////////////////////////
36             //DBから値を取得
37             while( rs.next() ){
38                 result = true;
39             }
40
41         }catch(ClassNotFoundException e) {
42             //エラー発生した場合にコンソールにログを出力する
43             e.printStackTrace();
44         }catch(SQLException e) {
45             //エラー発生した場合にコンソールにログを出力する
46             e.printStackTrace();
47         }
48
49         finally {
50             //接続 (コネクション) を閉じる
51             if(con!=null) {
52                 try {
53                     con.close();
54                 } catch (SQLException e) {
55                     e.printStackTrace();
56                 }
57             }
58         }
59
60         return result;
61     }
62 }
63
64
65
66
67
```

## Page2Servlet

```
14 @WebServlet("/page2")
15 public class Page2Servlet extends HttpServlet {
16
17     @Override
18     protected void doGet(HttpServletRequest request, HttpServletResponse response)
19         throws ServletException, IOException {
20
21         ///////////////////////////////////
22         //JSPから値を取得
23         String mail = request.getParameter("mail");
24
25         ///////////////////////////////////
26         //DBから検索
27         DBModel dbModel = new DBModel();
28         boolean result = dbModel.search(mail);
29
30         String message = "";
31         if (result) {
32             message = "検索成功";
33         } else {
34             message = "検索失敗";
35         }
36
37         ///////////////////////////////////
38         //値をリクエストスコープに設定する
39         request.setAttribute("message", message);
40
41         ///////////////////////////////////
42         //画面遷移をする
43         RequestDispatcher dispatcher = request.getRequestDispatcher("WEB-INF/jsp/page2.jsp");
44         dispatcher.forward(request, response);
45     }
46 }
47
```