

MVC のサンプル 2

前回作った、MVC のサンプルで **JavaBeans** をつかってみよう！

前回と動作は変わりませんが、ログイン情報を保存する **JavaBeans** を使ってみます。

※前回から変更するところは、以下の場所です。

- ・ログインした人の名前とメールアドレスをセッションに保存し、2 ページ目に保存する

1. プロジェクトを開く

前回作った、「mvcsample」を開きます。

2. JavaBeans の作成

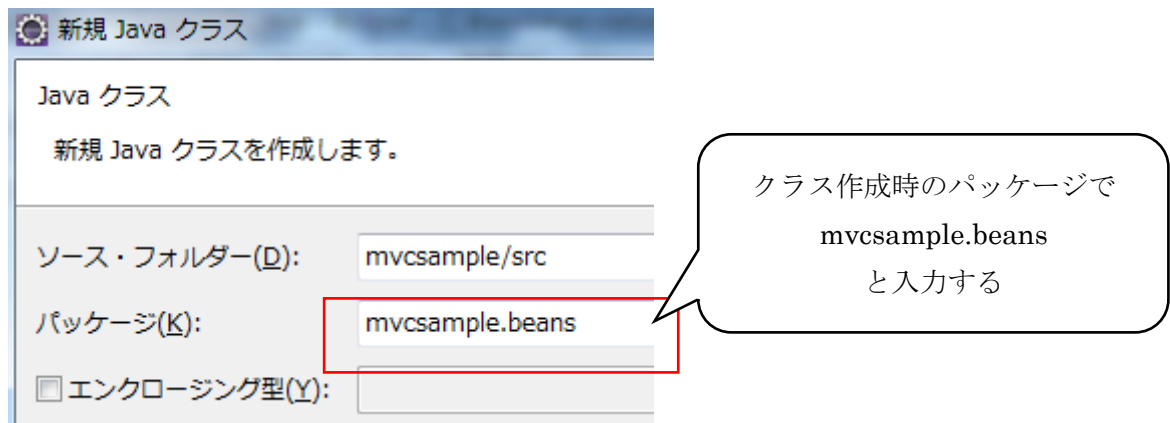
1 ページ目のントローラと JSP を記述します。

まずはサーブレットを作ります。ここではサーブレットの記述内容のみ記載します。

サーブレットの作り方を忘れた人は「サーブレットサンプル.pdf」を参考にして下さい

○LoginInfo を作り、以下のような記述をします。

【注意】 パッケージは **mvcsample.beans** にしましょう！



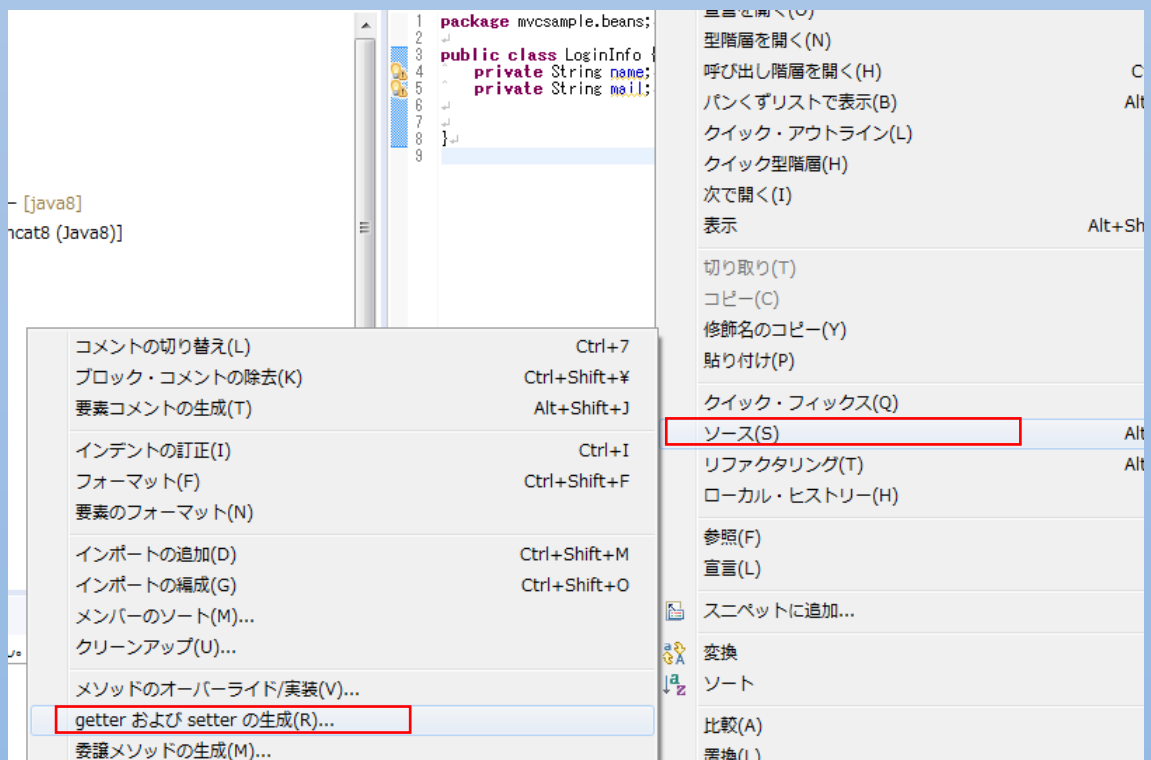
```
1 package mvcsample.beans;
2
3 public class LoginInfo {
4     private String name;
5     private String mail;
6
7     public String getName() {
8         return name;
9     }
10    public void setName(String name) {
11        this.name = name;
12    }
13    public String getMail() {
14        return mail;
15    }
16    public void setMail(String mail) {
17        this.mail = mail;
18    }
19 }
20
```

【ワンポイントアドバイス】

下のように、変数だけ書いた状態で

```
3 public class LoginInfo {  
4     private String name;  
5     private String mail;  
6 }
```

getter 及び setter の生成を選択すると、簡単に getXXX、setXXX メソッドが作成できます！



LoginStartController の修正

LoginStartController に JavaBeans をセッションに保存するコードを追加する (赤枠)

```
16 @WebServlet("/loginAuth")
17 public class LoginStartController extends HttpServlet {
18
19     @Override
20     protected void doPost(HttpServletRequest request, HttpServletResponse response)
21         throws ServletException, IOException {
22
23         //値を取得
24         String account = request.getParameter("account");
25         String password = request.getParameter("password");
26
27         //★モデルのインスタンスを生成する★
28         LoginModel loginModel = new LoginModel();
29
30         //パスワードと名前を比較して、結果によってメッセージを変更する
31         String message = "";
32         String forward = "";
33
34         //★モデルクラスのloginメソッドを呼び出して結果を受け取る★
35         boolean result = loginModel.login(account, password);
36         if (result) {
37             message = "ログイン成功しました";
38             forward = "WEB-INF/jsp/result_view.jsp";
39             //メッセージを結果画面へ送る
40             request.setAttribute("msg", message);
41
42             //◎ログイン成功時にBeanを生成する◎
43             LoginInfo loginInfo = new LoginInfo();
44
45             //◎setterを使って、値をセット◎
46             loginInfo.setMail("nishino@asojuku.ac.jp");
47             loginInfo.setName("西野直幸");
48
49             //◎セッションに値を保存する◎
50             HttpSession session = request.getSession(true);
51             session.setAttribute("loginInfo", loginInfo);
52
53         } else {
54             message = "ログイン失敗しました";
55             forward = "WEB-INF/jsp/login_view.jsp";
56             //メッセージを結果画面へ送る
57             request.setAttribute("errorMsg", "ユーザー名かパスワードが間違っています");
58         }
59     }
60 }
```

• result_view.jsp の修正

result_view.jsp で LoginInfo をセッションから取得して、値を表示するコードを追加する (赤枠)

```
1  <%@ page language="java" contentType="text/html; charset=UTF-8"%>
2      pageEncoding="UTF-8"%>
3  <%@ page import="mvc.sample.beans.LoginInfo"%>
4  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w
5  <html>
6  <head>
7  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"%>
8  <title>ログイン</title>
9  </head>
10 <body>
11 <%
12     String message = (String)request.getAttribute("msg");
13
14     LoginInfo loginInfo = (LoginInfo)session.getAttribute("loginInfo");
15 %>
16 <p>ようこそ！<%=loginInfo.getName()%>さん！</p>
17 <p><%=message%></p>
18 </body>
19 </html>
20
```