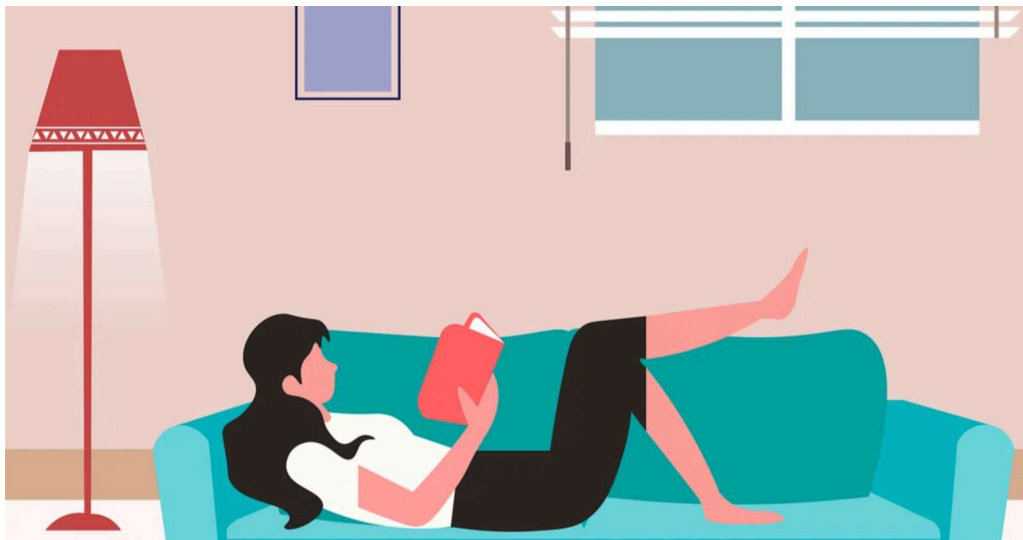




MVCモデルによるWebアプリケーション開発（No 4.カレンダー）-EE8



office・M

2024年1月11日 09:45

¥1,000



JavaEE8（JSP・Servlet）の環境でWebアプリケーションの開発方法を学ぶ講座をシリーズで提供しています。今回は部品として作成されたクラスファイル（カレンダーをhtml形式で出力するBean）を活用する方法を学びます。チーム開発での役割分担を想定しています。

2024年9月よりECLIPSEのバージョンを最新版（Version: 2024-06 (4.32.0)）に変更しました。

▼ 目次

開発概要

開発方針

設計仕様

外部設計

内部設計

実装手順

1. カレンダー作成ロジック用JavaBeansクラスを使用する準備をします

2. リクエストコントロール用Servletクラスを作成します

3. カレンダーを表示するJSPファイルを作成します

実行確認

1. InternalServerErrorとなります
 2. 実行用Beanを配置します
 3. 再度実行します
-

ソースコード例と提供Bean

1. CalendarHtmlBean.class
 2. CalendarServlet.java
 3. calendarResult.jsp
-

ソースコードファイル

連絡先

開発概要

JavaEE8アプリケーション・サーバ（Tomcat）を利用して開発を行うということを前提に実装を進めていきます。開発ツールには統合開発環境のEclipseを用いることにします。

題材としてカレンダーを出力するWebアプリケーションを作成します。入力画面に作成年と作成月を入力すると、同じ画面内に対象年月のカレンダーが表示されます。カレンダーを作成するBeanは提供します。入力画面と出力画面には同じJSPを使うため、GETリクエスト時にもBeanの実体化を行います。

JavaEE8環境ですので使用言語は当然Java（JSP・Servlet）となります。設計技法にはMVCモデルを使います。MVCモデルとはWebアプリケーションの構成をModel（業務ロジック）－View（表示）－Controller（制御）に分割して設計する技法です。3つのモデルに役割分担することで部品化が促され、ひいてはチーム開発に貢献します。

今回、業務ロジックであるBeanは作成しません。他のチームメンバーが作成したという想定で提供されたBeanを活用することにします。

実装していく過程でEclipseの利用方法が不明な場合は、別途メール等でお知らせください。この講座では、作成済みの仕様と設計書から「**JavaEE8環境でのMVCモデルの実装の仕組みを学ぶ**」ことを目的とします。

開発方針

Webアプリケーションの開発環境には図 1 のようにEclipse2024を用います。



図 1. 開発環境 Eclipse 2024

Spring等のフレームワークは使用しません。これは、MVCモデルにおけるhttpプロトコルの処理の実装を直に学んでいただきたいためです。

開発に必要な仕様と最小限の設計ドキュメントは提示します。この情報をもとに、まずは、自分で試行錯誤しながら実装してみてください。

設計仕様

【カレンダーの仕様】

以下の条件を具現化するWebアプリケーションの作成を行います。

(条件)

- ・ urlを指定してサイトにアクセスすると2つの入力欄（作成年、作成月）と作成ボタンが配置された画面を表示します
- ・ 作成月はドロップダウンリストで1月～12月を指定します
- ・ 起動時は当月のカレンダーを表示しています
- ・ 「作成ボタン」をクリックすると同じ画面内に対象年月のカレンダーを表示します

【Model（処理ロジック）】

CalendarHtmlBean.classが**提供**されます。

(仕様)

- ・ 引数で年と月を受け取る
- ・ 引数で指定された年月のカレンダーをhtml形式で作成する
- ・ 作成されたカレンダーはprivate変数に保持する
- ・ カレンダーの変数はpublicメソッドで外部に公開される

外部設計

接続urlは**http://localhost:8080/money/Calendar**とします。よってEclipseの動的Webプロジェクトの名前は**money**となり、サーブレットのurlパターンは**Calendar**となります。

Eclipseのメニューバーより

ファイル→新規→動的Webプロジェクト→「money」プロジェクトを作成する→図1.1の内容で設定する



図1.1 動的Webプロジェクトの設定

※作成済みであればこの処理は必要ありません

urlにアクセスすると図2のような、起動画面が表示されます。この時、当月のカレンダーが表示されています。作成年と作成月を入力して「作成ボタン」をクリックすると、図3のように入力された対象年月のカレンダーが表示されます。この時、新たな画面へ遷移はしません。



図2．起動画面（当月のカレンダーを表示）



図3. 入力された対象年月のカレンダーを表示

最終的には「マネーシミュレーションWebアプリ」のメニュー画面からクリックابلマップを利用して起動できるように構成します。

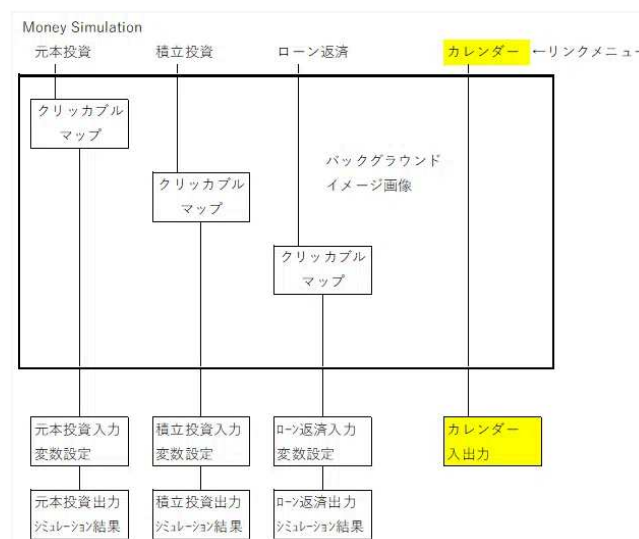


図3.1 マネーシミュレーションWebアプリ概念図

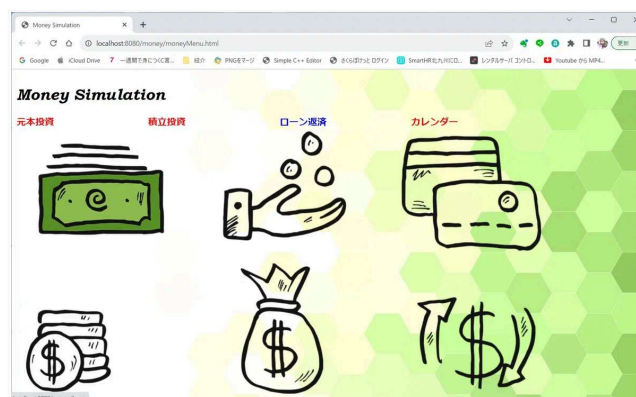


図3.2 マネーシミュレーションWebアプリの例

内部設計

クラス連携図は図 4 のようになります。

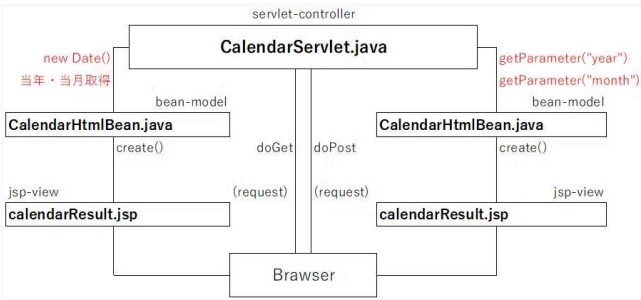


図 4. クラス連携（MVCモデル）図

提供されるカレンダー作成ロジック（CalendarHtmlBean.class）の使い方とクラス図は以下のようになります。

（使い方）

- CalendarHtmlBeanクラスを**年と月**の2つの引数をもつコンストラクタで実体化します。
- createメソッドを実行することでdayフィールドに対象年月のカレンダーがhtml文字列で保持されます。
- 利用側のクラスはgetDay()メソッドでカレンダーの内容を取得します。

（クラス図）

package:jp.ict.aso.model	
CalendarHtmlBean	
- year : int	//作成年
- month : int	//作成月
- day : String	//対象年月カレンダー（html形式）
+ CalendarHtmlBean()	//コンストラクタ
+ CalendarHtmlBean(year: int, month: int)	//コンストラクタ
+ create() : void	//カレンダー作成
+ getDay() : String	//カレンダー取得
+ getYear() : int	//作成年取得
+ getMonth() : int	//作成月取得
+ week_of_day(y: int, m:int, d:int) : int	//曜日計算
+ month_last_day(y: int, m: int) : int	//月末日計算
+ is_leap_year(y: int) : boolean	//うるう年判定
+ is_holiday(m: int, d: int) : boolean	//休日判定
-:private +:public #:protected	

図4.1 CalendarHtmlBeanクラス図

実装手順

1. カレンダー作成ロジック用JavaBeansクラスを使用する準備をします

(1)クラスファイルの準備をします

以下Windows環境を想定しています。

事前に「提供Beanフォルダ」を作成しておきます（例 c:\提供Bean）。

提供Beanフォルダ内に**CalendarHtmlBean.class**を保存しておきます。

その際**パッケージの階層**に従ってください。

（例 c:\提供Bean\jp\ict\aso\model\CalendarHtmlBean.class）

(2)CalendarHtmlBean.classをEclipseのビルド・パスに追加します

Eclipseパッケージ・エクスプローラより

moneyプロジェクトを右クリック→ビルド・パス→ビルド・パスの構成→「ライブラリ」タブ→「クラスパス」をクリック→外部クラス・フォルダーの追加→「提供Bean」を選択します→最後に「適用して閉じる」をクリック

※図5のように一度設定されていれば再度設定する必要はありません

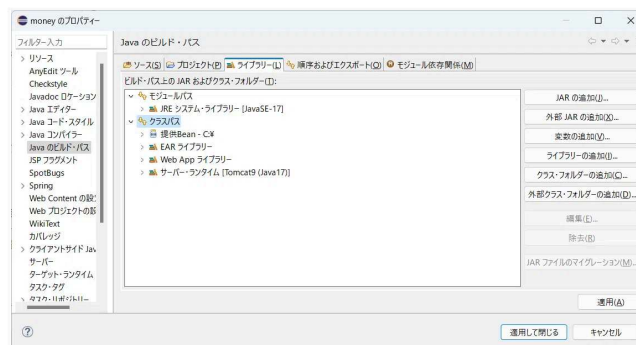


図5. Javaのビルドパス追加画面

2. リクエストコントロール用Servletクラスを作成します

Eclipseパッケージ・エクスプローラより

moneyプロジェクトを右クリック→新規→その他→Web→サーブレット
→以下の内容で作成する

- CalendarServlet.java
 - パッケージ : jp.ict.aso.controller
 - クラス名 : CalendarServlet
 - ソースコード : 考えましょう！ ※アノテーションは/Calendar

3. カレンダーを表示するJSPファイルを作成します

Eclipseパッケージ・エクスプローラより
moneyプロジェクトを右クリック→新規→その他→Web→JSPファイル
→以下の内容で作成する

- calendarResult.jsp
 - 保存場所 : money/src/main/webapp/**WEB-INF/jsp** ← 注意！
 - ファイル名 : calendarResult.jsp
 - ソースコード : 考えましょう！

実行確認

サーブレットクラス (CalendarServlet.java) を実行します。しかし、、、

1. InternalServerErrorとなります

実行用Beanのデプロイ (配置) が必要です。この設定を行わないと図6のような実行時エラーになります。

Eclipseパッケージ・エクスプローラより
CalendarServlet.javaを右クリック→実行→サーバーで実行
→図6のように**エラー**となる

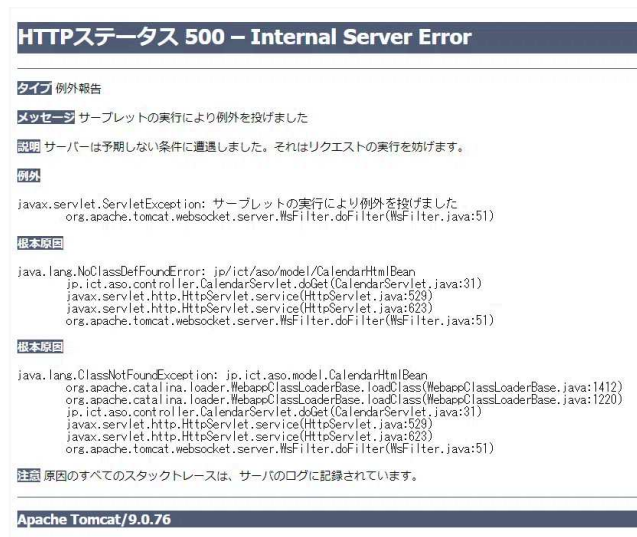


図 6. 実行時エラー画面

2. 実行用Beanを配置します

CalendarHtmlBean.classをEclipseの実行時のクラスパスに追加します。

Eclipseプロジェクト・エクスプローラより

(パッケージ・エクスプローラではありません！)

moneyプロジェクトを展開→buildを展開→classesを展開→jpを展開→ictを展開→asoを展開→modelがなければ作成



図 7. 実行時クラスパスの位置

図 7 のように model フォルダの位置へ CalendarHtmlBean.class をドラッグ & ドロップすることで実行時クラスパスに Bean が追加されます。

※Eclipseを終了させると、再度追加が必要な場合があります。

3. 再度実行します

Tomcatサーバを再起動したのち、再度サーブレットクラス（CalendarServlet.java）を実行します。

Eclipseパッケージ・エクスプローラより
CalendarServlet.javaを右クリック→実行→サーバーで実行
→図8のように実行される（背景画像は任意に設定してください）



図8. 起動画面（当月のカレンダーを表示）

ソースコード例と提供Bean

以下に各プログラムのソースコードの例（本文内では“考えましょう！”になっている部分）を示しますので、実装の参考にしてください。

また、提供BeanとしてCalendarHtmlBean.class（カレンダーの計算ロジックが含まれています）を次のセクションに置きますので、ダウンロードして使用してください。

----- このラインより上のエリアが無料で表示されます。 -----

1. CalendarHtmlBean.class

CalendarHtmlBean.class



3.28 KB

[ファイルダウンロードについて](#)

 ダウンロード

2. CalendarServlet.java

```
1 package jp.ict.aso.controller;
2
3 import java.io.IOException;
4 import java.util.Calendar;
5 import java.util.Date;
6 import javax.servlet.RequestDispatcher;
7 import javax.servlet.ServletException;
8 import javax.servlet.annotation.WebServlet;
9 import javax.servlet.http.HttpServlet;
10 import javax.servlet.http.HttpServletRequest;
11 import javax.servlet.http.HttpServletResponse;
12 import jp.ict.aso.model.CalendarHtmlBean;
13
14 @WebServlet("/Calendar")
15 public class CalendarServlet extends HttpServlet {
16
17     protected void doGet(HttpServletRequest request,
18         HttpServletResponse response)
19         throws ServletException, IOException {
20
21         // 当年当月を求める
22         Date now = new Date();
23         Calendar c = Calendar.getInstance();
24         c.setTime(now);
25         int year=c.get(Calendar.YEAR);
26         int month=c.get(Calendar.MONTH)+1;
27
28         // 入力値をプロパティに設定
29         CalendarHtmlBean cb = new CalendarHtmlBean(year, month);
30         cb.create();
31
32         // リクエストスコープに保存
33         request.setAttribute("result", cb);
```

図9. CalendarServlet.java (1)

```
34
35 // フォワード
36 RequestDispatcher dispatcher =
37     request.getRequestDispatcher
38     ("/WEB-INF/jsp/calendarResult.jsp");
39 dispatcher.forward(request, response);
40 }
41 protected void doPost(HttpServletRequest request,
42     HttpServletResponse response)
43     throws ServletException, IOException {
44
45     // リクエストパラメータの取得
46     request.setCharacterEncoding("UTF-8");
47     String syear = request.getParameter("year");
48     String smonth = request.getParameter("month");
49     int year = Integer.parseInt(syear);
50     int month = Integer.parseInt(smonth);
51
52     // 入力値をプロパティに設定
53     CalendarHtmlBean cb = new CalendarHtmlBean(year, month);
54     cb.create();
55
56     // リクエストスコープに保存
57     request.setAttribute("result", cb);
58
59     // フォワード
60     RequestDispatcher dispatcher =
61         request.getRequestDispatcher
62         ("/WEB-INF/jsp/calendarResult.jsp");
63     dispatcher.forward(request, response);
64 }
65 }
```

図10. CalendarServlet.java (2)

3. calendarResult.jsp

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8" import="jp.ict.aso.model.*" %>
3 <%
4   CalendarHtmlBean cb=(CalendarHtmlBean) request.getAttribute("result");
5 %>
6 <!DOCTYPE html>
7 <html>
8 <head>
9   <meta charset="UTF-8">
10  <title>カレンダー出力</title>
11 </head>
12 <BODY background="images/money_back1920.jpg">
13
14 <h1>カレンダーを作成します</h1>
15 <hr>
16
17 <form method="POST" action="/Calendar">
18 <p>作成年: <input type="number" name="year" step=1 value=2023</p>
19 <p>作成月: <!-- <input type="number" name="month" step=1 value=1</p> -->
20
21 <SELECT name="month">
22   <OPTION value="1" selected>1</OPTION>
23   <OPTION value="2">2</OPTION>
24   <OPTION value="3">3</OPTION>
25   <OPTION value="4">4</OPTION>
26   <OPTION value="5">5</OPTION>
27   <OPTION value="6">6</OPTION>
28   <OPTION value="7">7</OPTION>
29   <OPTION value="8">8</OPTION>
30   <OPTION value="9">9</OPTION>
31   <OPTION value="10">10</OPTION>
32   <OPTION value="11">11</OPTION>
33   <OPTION value="12">12</OPTION>
```

図11. calendarResult.jsp (1)

```
34     </SELECT>
35
36     <input type="submit" value="作成">
37
38     </form>
39
40 <p>
41   作成年 : <%=cb.getYear() %>年<br>
42   作成月 : <%=cb.getMonth() %>月<br>
43   カレンダー : <%=cb.getDay() %><br>
44 </p>
45
46 <a href="Calendar">戻る</a>
47 </body>
48 </html>
```

図12. calendarResult.jsp (2)

ソースコードファイル



CalendarServlet.java

2.03 KB

[ファイルダウンロードについて](#)

 ダウンロード



calendarResult.jsp

1.29 KB

[ファイルダウンロードについて](#)

 ダウンロード



CalendarHtmlBean.java

3.67 KB

[ファイルダウンロードについて](#)

 ダウンロード

連絡先

質問や不明な点がある場合は以下のアドレスにメールをください。可能な限り24時間以内に返信いたします。

info@slowlife.halfmoon.jp