MVCモデルによるWebアプリケーション開発（No2.おみくじ）-EE8

office・M
2024年1月9日 15:44

¥500 •••

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。２回目はリクエストパラメータを利用して画面を遷移する題材（おみくじ）をMVCモデルで実装してみます。

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。合わせて文中のソースコード（"考えましょう！"になっている部分のファイル）とおみくじの画像ファイルなどを9月限定のタイムセールで **50% off** で提供いたします。

▼ 目次

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。

題材としてサーバでおみくじをひくWebアプリケーションを作成します。画面は起動画面、おみくじスタート画面、占い結果画面の３段階に遷移します。画面遷移の判断にはリクエストパラメータを利用します。

JavaEE8環境ですので使用言語は当然Java（JSP・Servlet）となり、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）ーView（表示）ー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。

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。

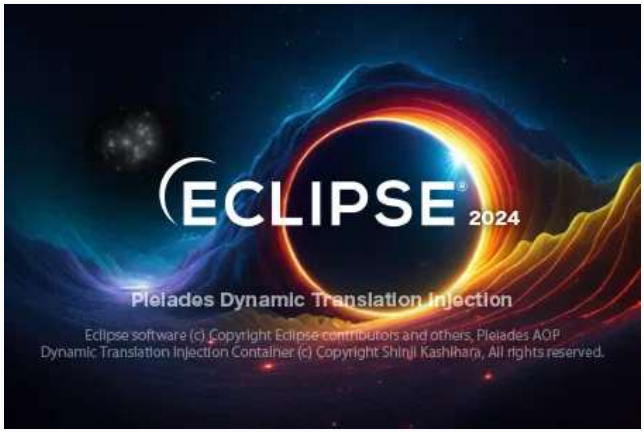

図１．開発環境 Eclipse 2024

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

【おみくじの仕様】
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おみくじ開始の画像と「スタートボタン」を表示します
・スタートボタンをクリック後おみくじを回しているアニメーションと「おみくじを引くボタン」を表示します
・おみくじを引くボタンをクリック後に占いの結果を表示します
・占いの結果は４種類で以下の確率で出現するようにします
　大吉　10%
　中吉　40%
　小吉　40%
　大凶　10%

【Model（処理ロジック）の設計方針】
以下の仕様で実装します。
（仕様）
・０から９までの乱数を発生させる
（Javaの例：int ran=new java.util.Random().nextInt(10);）
・乱数値が０ならば占い結果は「大凶」とします

・乱数値が１ならば占い結果は「大吉」とします
・乱数値が２or３or４or５ならば占い結果は「中吉」とします
・上記以外は「小吉」とします
（実装）
・JavaBeansの仕様に基づきOmikujiBean.javaを作成します
・package名は jp.ict.aso.model とします
・作成するクラスはSerializableインターフェースを実装します
・変数はprivate宣言します
・引数なしのコンストラクタを実装します
・コンストラクタ内に処理ロジックを実装します、このとき処理結果はインスタンス変数に格納します
・setterメソッドを実装します
・getterメソッドを実装します

外部設計

接続urlは**http://localhost:8080/libcon/Omikuji**とします。よってEclipseの動的Webプロジェクトの名前は**libcon**となり、サーブレットのurlパターンは**Omikuji**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→「libcon」プロジェクトを作成する
→図1.1の内容で設定する

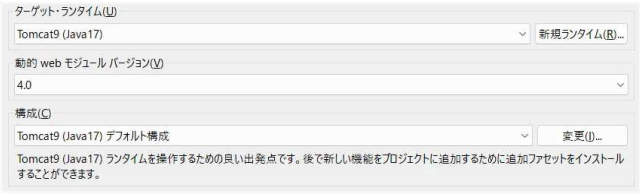

図1.1 動的Webプロジェクトの設定

※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。「スタートボタン」をクリックすると図３と図３.１のような、神社にお参りするアニメーションが表示されたおみくじスタート画面へ遷移します。ここで「おみくじを引く」ボタンをクリックすると図４のような、占いの結果画面に遷移します。

※GIFアニメーションの作り方はこのページの最後の方で説明しています。

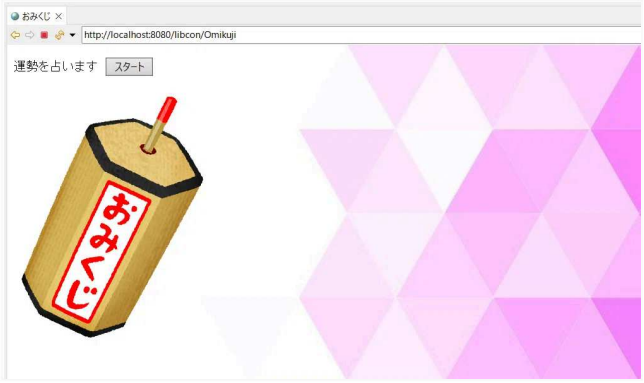

図２．起動画面

図３．おみくじスタート画面

図３.１ お参りアニメーション

図４．占い結果画面

内部設計

クラス連携図は図５のようになります。

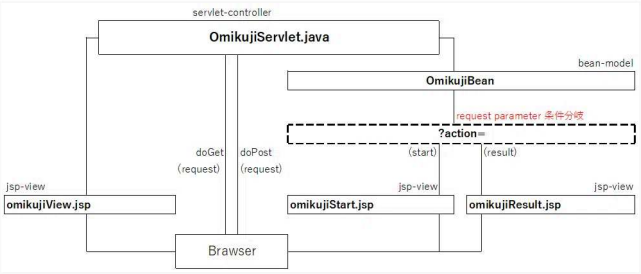

図５．MVCモデル図

おみくじロジックのクラス図は以下のようになります。

```
package: jp.ict.aso.model


-------------------------------------
              OmikujiBean
-------------------------------------
- unsei: String               // フィールド（プロパティ）
-------------------------------------
+ OmikujiBean()              // コンストラクタ
+ getUnsei(): String         // ゲッター
-------------------------------------
```

図5.1 クラス図

実装手順

１．おみくじロジック用JavaBeansクラスを作成します

Eclipseパッケージ・エクスプローラより
libconプロジェクトを右クリック→新規→クラス
→以下の内容で作成

・OmikujiBean.java
　　パッケージ：jp.ict.aso.model
　　名前：OmikujiBean
　　ソースコード：考えましょう！

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
libcon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

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

３．起動画面を表示する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
libcon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

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

４．おみくじスタート画面を表示する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
libcon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

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

５．占い結果を表示する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
libcon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

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

６．画像ファイルを配置します

Eclipseパッケージ・エクスプローラより
libconプロジェクト配下の **src/main/webapp** 右クリック→新規→フォルダ→**images**フォルダを作成
→図６のような画像を作成して各画像をimagesフォルダへコピーする

図6．おみくじ関連画像例

※画像ファイルは適切なものを用意してください。

実行確認

サーブレットクラス（OmikujiServlet.java）を実行します！！

Eclipseパッケージ・エクスプローラより
OmikujiServlet.javaを右クリック→実行→サーバーで実行
→図７のように実行される

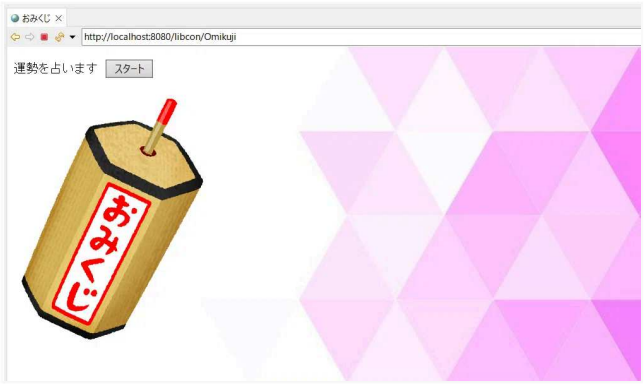

図7．起動画面

ソースコード例

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、おみくじ関連の画像ファイルをアーカイブしたZIPファイル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。

９月限定タイムセールを行います！50%オフで画像ファイルを含む全てのソースファイルがダウンロード可能になります。

１．OmikujiBean.java

```java
package jp.ict.aso.model;
import java.io.Serializable;
public class OmikujiBean implements Serializable{
  private String unsei;
  public OmikujiBean() {
   int fortune=new java.util.Random().nextInt(10);

   switch (fortune) {
    case 0:
     unsei="大吉";break;
    case 1:
     unsei="大凶";break;
    case 2:
    case 3:
    case 4:
    case 5:
     unsei="中吉";break;
    default:
     unsei="小吉";break;
   }
  }
  public String getUnsei() {
    return unsei;
  }
  public static void main(String[] args){
    OmikujiBean ob = new OmikujiBean();
    System.out.println(ob.getUnsei());
  }

}
```

図７．OmikujiBean.java

２．OmikujiServlet.java

```java
package jp.ict.aso.controller;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import jp.ict.aso.model.OmikujiBean;

@WebServlet("/Omikuji")
public class OmikujiServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        RequestDispatcher rd=request.getRequestDispatcher
                ("/WEB-INF/jsp/omikujiView.jsp");
        rd.forward(request, response);
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        OmikujiBean ob=new OmikujiBean();
        request.setAttribute("unsei", ob);
```

図８．OmikujiServlet.java（１）

```java
        //リクエストパラメータからactionの値を取得して条件判断
        String action = request.getParameter("action");
        if(action==null) {
            RequestDispatcher rd=request.getRequestDispatcher
                    ("/WEB-INF/jsp/omikujiView.jsp");
            rd.forward(request, response);

        }else if(action.equals("start")) {
            RequestDispatcher rd=request.getRequestDispatcher
                    ("/WEB-INF/jsp/omikujiStart.jsp");
            rd.forward(request, response);

        }else if(action.equals("result")) {
            RequestDispatcher rd=request.getRequestDispatcher
                    ("/WEB-INF/jsp/omikujiResult.jsp");
            rd.forward(request, response);
        }
    }
}
```

図９．OmikujiServlet.java（２）

３．omikujiView.jsp

```jsp
<%@ page language="java" contentType="text/html;
charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>おみくじ</title>
</head>
<body>
<form action="Omikuji?action=start" method="post">
<p>
運勢を占います　<input type="submit" value="スタート">
</p>
<img src=images/omikuji1.png><br>
</form>
</body>
</html>
```

図10．omikujiView.jsp

４．omikujiStart.jsp

```jsp
<%@ page language="java" contentType="text/html;
charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>おみくじ</title>
</head>
<body>
<form action="Omikuji?action=result" method="post">
<p>
運勢を占います　<input type="submit" value="おみくじを引く">
</p>
<img src=images/omikuji2.gif><br>
</form>
</body>
</html>
```

図11．omikujiStart.jsp

５．omikujiResult.jsp

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" import="jp.ict.aso.model.*"%>
<%
OmikujiBean ob=(OmikujiBean)request.getAttribute("unsei");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>運勢</title>
</head>
<body>
<p>あなたの運勢は<%= ob.getUnsei() %>です！</p>
<%
if(ob.getUnsei().equals("大吉")){
out.print("<img src=images/great.png><br>");
}
if(ob.getUnsei().equals("中吉")){
out.print("<img src=images/middle.png><br>");
}
if(ob.getUnsei().equals("小吉")){
out.print("<img src=images/small.png><br>");
}
if(ob.getUnsei().equals("大凶")){
out.print("<img src=images/bad.png><br>");
}
%>
<a href="Omikuji">戻る</a>
</body>
</html>
```

図12．omikujiResult.jsp

GIFアニメーションの作り方

GIFアニメーションの作り方.pdf

1.77 MB

ファイルダウンロードについて

ダウンロード

背景等画像のフリーサイト

https://pixabay.com/illustrations/search/background/

ソースコードファイルと画像ファイル

OmikujiBean.java

638 Bytes

ダウンロード

ファイルダウンロードについて

OmikujiServlet.java
1.56 KB
ファイルダウンロードについて
ダウンロード

omikujiView.jsp
453 Bytes
ファイルダウンロードについて
ダウンロード

omikujiStart.jsp
452 Bytes
ファイルダウンロードについて
ダウンロード

omikujiResult.jsp
744 Bytes
ファイルダウンロードについて
ダウンロード

images.zip
1.16 MB
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。

実装サポートは、原則、木曜日と金曜日を予定していますので、お気軽にお問い合わせください。

info@slowlife.halfmoon.jp