

## テンプレートフレームワーク

### 目次

1. Tiles を使用する	1
2. レイアウト定義の継承	8

## 1. Tiles を使用する

### 1 テンプレートとは

Web アプリケーションにおいて、ヘッダ、メニュー、フッタは全てのページで共通でボディ部分だけが変更されるようなサイトはよく見かけます。

このようなサイトを作成しようと思った場合、通常はヘッダ用、メニュー用、フッタ用のページを用意し、各 JSP ページでそれらのページをインクルードします。ただインクルード機能を使用しても、すべての JSP ページにおいてレイアウトを記述する必要があるため、レイアウトを変更する際の煩雑さは存在します。

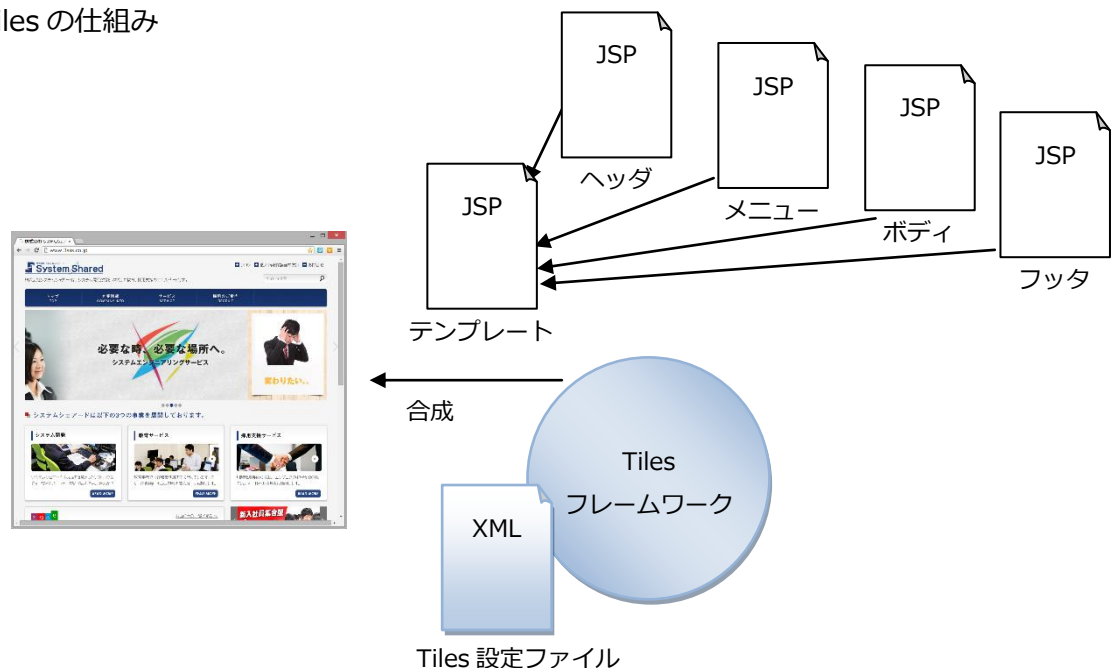
「Tiles」はこれらの煩雑さを解消するテンプレートフレームワークです。Tiles ではレイアウトが記述されたものをテンプレートと呼び、テンプレートにタイルと呼ばれるヘッダやボディなどの部品を当てはめていくことで、同じレイアウトのWebページを作成することができます。

Tiles を使用することでレイアウトとコンテンツを分離できるため、コードの冗長性やレイアウトの変更があったときの作業量も減少させることができます。

### 2 Tiles の仕組み

Tiles を利用するためには、テンプレートとなる JSP ファイル、それぞれのタイルのコンテンツ（ヘッダ、メニュー、ボディ、フッタ等）となる JSP ファイル、テンプレートとコンテンツの関係を定義する Tiles 設定ファイルが必要です。

#### ■ Tiles の仕組み



### 3 Tiles を使用する

それでは、実際に Tiles を使用してみましょう。使用する流れは、次のようになります。

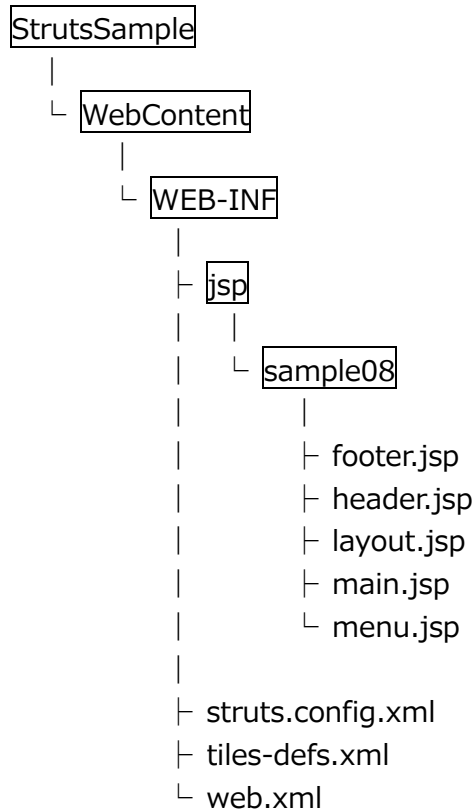
- (1) web.xml に Tiles タグライブラリを使用するための設定を行う
- (2) Struts 設定ファイルに Tiles プラグインを設定する
- (3) Tiles 設定ファイルを作成する
- (4) テンプレートを作成する
- (5) コンテンツを作成する

作成、および変更するファイルの一覧を以下に示します。

ファイル名	解説	ロケーション
struts-config.xml	Struts 設定ファイル	/WEB-INF/
layout.jsp	テンプレート JSP	/WEB-INF/jsp/sample08/
header.jsp	ヘッダ用 JSP	/WEB-INF/jsp/sample08/
menu.jsp	メニュー用 JSP	/WEB-INF/jsp/sample08/
main.jsp	ボディ用 JSP	/WEB-INF/jsp/sample08/
footer.jsp	フッタ用 JSP	/WEB-INF/jsp/sample08/
tiles-defs.xml	Tiles 設定ファイル	/WEB-INF/

Eclipse 上から見たディレクトリ構成を以下に示します。

#### 【ディレクトリ構成】



## (1) Tiles の有効化

Tiles はプラグインとして提供されていますが、デフォルトでは有効になっていないため、以下の設定を行う必要があります。

最初に Tiles タグライブラリを使用できるように設定を行います。Struts をインストールしたディレクトリ下の lib ディレクトリに、struts-tiles.tld ファイルがありますので、このファイルを任意の場所（今回は WEB-INF 直下）にコピーします。そして他のタグライブラリと同様、web.xml に以下のコードを追記します。

### 【web.xml : Tiles タグライブラリの設定】

```
<taglib>
  <taglib-uri>/tags/struts-tiles</taglib-uri>
  <taglib-location>/WEB-INF/struts-tiles.tld</taglib-location>
</taglib>
```

次に、Tiles プラグインの設定を行うため Struts 設定ファイルを編集します。

### 【struts-config.xml : Tiles プラグインの設定】

```
<struts-config>
  . . . (略) . . .
  <plug-in className="org.apache.struts.tiles.TilesPlugin">
    <set-property property="definitions-config"
      value="/WEB-INF/tiles-defs.xml"/>
  </plug-in>
</struts-config>
```

plug-in 要素は struts-config 要素にネストされる要素の最後に設定し、className 属性に org.apache.struts.tiles.TilesPlugin を指定します。次に、set-property 要素で Tiles 設定ファイルをアプリケーションルートからの相対パスで指定します。この例では tiles-defs.xml としていますが、ファイル名は任意のものでかまいません。

## (2) Tiles 設定ファイルを作成する

Tiles 設定ファイルは、テンプレートにタイルをどのように組み合わせるかを定義するものです。ここでは、最終的に以下のようなページを生成する Tiles 設定ファイルを作成します。



### 【tiles-defs.xml】

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE tiles-definitions PUBLIC
    "-//Apache Software Foundation//DTD Tiles Configuration 1.1//EN"
    "http://jakarta.apache.org/struts/dtds/tiles-config_1_1.dtd">
<tiles-definitions>
    <definition name="sampleLayout"
        path="/WEB-INF/jsp/sample08/layout.jsp">
        <put name="title" value="レイアウトサンプル"/>
        <put name="header" value="/WEB-INF/jsp/sample08/header.jsp"/>
        <put name="menu" value="/WEB-INF/jsp/sample08/menu.jsp"/>
        <put name="main" value="/WEB-INF/jsp/sample08/main.jsp"/>
        <put name="footer" value="/WEB-INF/jsp/sample08/footer.jsp"/>
    </definition>
</tiles-definitions>
```

Tiles 設定ファイルでは XML 宣言の後に、DOCTYPE 宣言を記述します。

タイルの組み合わせの定義は tiles-definitions 要素内の definition 要素に記述します。1 つの画面定義につき、1 つの definition 要素が対応します。name 属性に Web アプリケーションで一意となる論理名を指定します。ここで設定した論理名は Struts 設定ファイル内でフォワード先として指定することができます。

path 属性には使用するテンプレートの URL を、アプリケーションルートからの相対パスで指定します。

次に definition 要素内に put 要素を追加します。put 要素は、この画面定義に含まれるリソースの数だけ並べることになります。この例では最終的に<body>タグ内に出力される header、menu、body、footer と、<head>タグ内に出力される title の、合わせて 5 つのタイルを配置します。name 属性にテンプレートから直接参照できる論理名を指定し、value 属性に表示する文字列やインクルードするファイルへのパスを指定します。

### (3) テンプレートを作成する

次にテンプレートとなる JSP ファイルを作成します。レイアウト自体はテーブルを使って画面を 4 つに区切り、それぞれに挿入するタイルを設定するだけです。

#### 【layout.jsp】

```
<%@ page contentType="text/html; charset=UTF-8" %>
<%@ taglib uri="/tags/struts-html" prefix="html" %>
<%@ taglib uri="/tags/struts-tiles" prefix="tiles" %>
<html:html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <title><tiles:getAsString name="title" /></title>
</head>
<body>
    <table border="1" width="100%" cellpadding="1">
        <tr>
            <td colspan="2" height="100" align="center">
                <tiles:insert attribute="header"/>
            </td>
        </tr>
        <tr>
            <td width="400" height="100" align="center">
                <tiles:insert attribute="menu"/>
            </td>
            <td height="100" align="center">
                <tiles:insert attribute="main"/>
            </td>
        </tr>
        <tr>
            <td colspan="2" height="100" align="center">
                <tiles:insert attribute="footer"/>
            </td>
        </tr>
    </table>
</body>
</html:html>
```

まずファイルの先頭で他のタグライブラリを使用する場合と同様に、Tiles タグライブラリを使用するための宣言を記述します。

tiles:getAsString タグは、name 属性に指定した論理名のタイルから文字列を書き出すためのタグです。Tiles 設定ファイルの中で設定されている「レイアウトサンプル」の文字が"title"に書き出されます。

tiles:insert タグはタイルを挿入するためのタグで、attribute 属性で指定された論理名を持つタイルが、それぞれ挿入されます。

#### (4) コンテンツを作成する

次に先ほど作成したテンプレートのtiles:insertタグの部分に挿入するJSP ファイルを作成します。今回はサンプルのため、単純な文字列のみのファイルになっています。

##### 【header.jsp】

```
<%@ page contentType="text/html; charset=UTF-8" %>  
ヘッダ
```

##### 【menu.jsp】

```
<%@ page contentType="text/html; charset=UTF-8" %>  
メニュー
```

##### 【main.jsp】

```
<%@ page contentType="text/html; charset=UTF-8" %>  
ボディ
```

##### 【footer.jsp】

```
<%@ page contentType="text/html; charset=UTF-8" %>  
フッタ
```

#### (5) Struts 設定ファイルの forward 要素

最後に Struts 設定ファイルを編集します。通常、Action を実行した後のフォワード先は forward 要素の path 属性に JSP ファイル名、もしくは続けて実行する Action のパスを指定しますが、Tiles が生成するページへフォワードする場合は、Tiles 設定ファイルの definition 要素で指定した論理名を記述します。

以下に例を示します。

```
<forward name="success" path="sampleLayout"/>
```

これは、ForwardAction を使用するときも同様で、次のように parameter 属性に Tiles 設定ファイルで指定した論理名を指定します。

##### 【Struts-config.xml】

```
．．．(略)．．．  
  <action-mappings>  
    ．．．(略)．．．  
    <action path="/sample08/layout"  
            type="org.apache.struts.actions.ForwardAction"
```

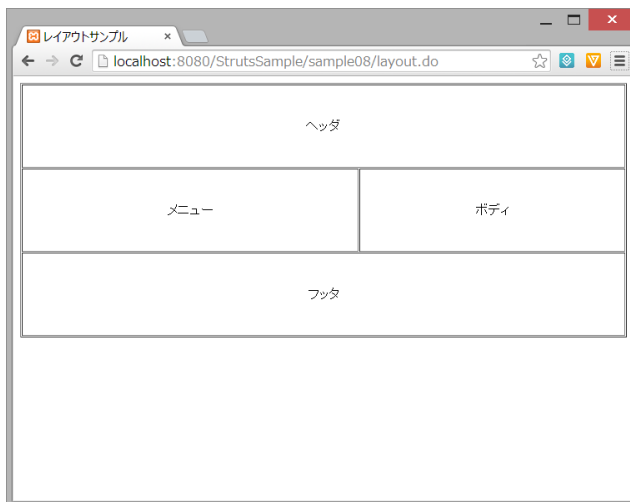


```
        parameter="sampleLayout"/>
    </action-mappings>
    . . . (略) . . .
    <plug-in className="org.apache.struts.tiles.TilesPlugin">
        <set-property property="definitions-config"
            value="/WEB-INF/tiles-defs.xml"/>
    </plug-in>
</struts-config>
```

それでは、動作を確認しましょう。Tomcat を起動し、下記 URL をブラウザで表示します。

<http://localhost:8080/StrutsSample/sample08/layout.do>

#### ■ サンプルレイアウト画面





## 2. レイアウト定義の継承

Tiles 設定ファイルの definition 要素はその 1 つ 1 つが画面定義に対応しています。つまり、たとえ使用するタイルがほぼ同一であっても、別のページの場合は似たような記述を繰り返さなければなりません。

冗長な Tiles 設定の例を以下に示します。

### 冗長な Tiles 設定ファイル

```
<definition name="layout01" path="/WEB-INF/jsp/layout.jsp">
  <put name="title" value="サンプル 1"/>
  <put name="header" value="/WEB-INF/jsp/header.jsp"/>
  <put name="menu" value="/WEB-INF/jsp/menu.jsp"/>
  <put name="body" value="/WEB-INF/jsp/sample01.jsp"/>
  <put name="footer" value="/WEB-INF/jsp/footer.jsp"/>
</definition>
<definition name="layout02" path="/WEB-INF/jsp/layout.jsp">
  <put name="title" value="サンプル 2"/>
  <put name="header" value="/WEB-INF/jsp/header.jsp"/>
  <put name="menu" value="/WEB-INF/jsp/menu.jsp"/>
  <put name="body" value="/WEB-INF/jsp/sample02.jsp"/>
  <put name="footer" value="/WEB-INF/jsp/footer.jsp"/>
</definition>
```

上記の例は、title と body 以外は全く同じものです。これでは設定が冗長であるといえるでしょう。

このような場合の対処として、Tiles には定義の継承という機能があります。定義の継承を行うことで定型的な部分を抜き出して定義することができます。

定義の継承を行うには、definition 要素で path 属性の代わりに extends 属性を使用します。上記の定義は以下のように書き直すことができます。

### 継承を使用した Tiles 設定ファイル

```
<definition name="layout01" path="/WEB-INF/jsp/layout.jsp">
  <put name="title" value="サンプル 1"/>
  <put name="header" value="/WEB-INF/jsp/header.jsp"/>
  <put name="menu" value="/WEB-INF/jsp/menu.jsp"/>
  <put name="body" value="/WEB-INF/jsp/sample01.jsp"/>
  <put name="footer" value="/WEB-INF/jsp/footer.jsp"/>
</definition>
<definition name="layout02" extends="layout01">
  <put name="title" value="サンプル 2"/>
  <put name="body" value="/WEB-INF/jsp/sample02.jsp"/>
</definition>
```

上記の例では layout01 という定義を継承して、新たに layout02 を定義しています。継承先の定義は、継承元の定義の属性やプロパティをすべて継承します。このコードにおいては、title と body の定義が上書きされています。