Anyframe Tiles Plugin

Version 1.1.0

I.Introduction

tiles plugin은 Tiles(Apache Tiles 2.2.1) 기반의 화면 레이아웃 정의를 가이드하기 위한 샘플 코드와 이 오픈소스들을 활용하는데 필요한 참조 라이브러리들로 구성되어 있다.

Installation

Command 창에서 다음과 같이 명령어를 입력하여 tiles plugin을 설치한다.

```
mvn anyframe:install -Dname=tiles
```

installed(mvn anyframe:installed) 혹은 jetty:run(mvn clean jetty:run) command를 이용하여 설치 결과를 확인해볼 수 있다.

Plugin Name	Version Range
core [http://dev.anyframejava.org/docs/anyframe/plugin/essential/core/1.6.0/reference/htmlsingle/core.html]	2.0.0 > * > 1.4.0

II.Tiles Integration

Spring MVC에서는 Tiles1, Tiles2를 각각 지원하는 viewClass를 제공한다. 본 매뉴얼에서는 Tiles2와의 연계 방안에 대해 설명할 것이며 기본적인 viewResolver에 대한 내용은 Core Plugin >> Spring MVC >> Configuration의 View Resolver [http://dev.anyframejava.org/docs/anyframe/plugin/essential/core/1.6.0/reference/htmlsingle/core.html#core_springmvc_configuration_actionservletxml_viewresolver] 정의 부분을 참고한다. Spring MVC와 Tiles2를 연계하기 위해서는 JDK1.5 이상, Tiles 2.0.X 이상 , Commons BeanUtils, Commons Digester, Commons Logging이 필요하다. Tiles2를 연계하기 위해서는 아래와 같은 절차를 따른다.

- Tiles view class 정의
- TilesConfigurer 정의
- Tiles definition 파일 작성

1.Tiles view class 정의

viewResolver 정의 부분에서 간략하게 설명했듯이 Tiles 2를 이용하기 위해서는 UrlBasedViewResolver를 정의한 후 viewClass를 아래 코드와 같이 org.springframework.web.servlet.view.tiles2.TilesView로 정의해줘야 한다.

```
<bean id="tilesViewResolver"
    class="org.springframework.web.servlet.view.UrlBasedViewResolver">
    <property name="viewClass"
        value="org.springframework.web.servlet.view.tiles2.TilesView" />
</bean>
```

2.TilesConfigurer 정의

Tiles 매핑 관련 정보가 작성되어 있는 tiles definition 파일의 위치를 정의해줘야 하는데 이 때 TilesConfigurer를 아래와 같이 정의해 준다.

```
<bean id="tilesConfigurer"
    class="org.springframework.web.servlet.view.tiles2.TilesConfigurer">
    <property name="definitions">
        <list>
            <value>/WEB-INF/tiles-def.xml</value>
        </list>
    </property>
</bean>
```

위와 같이 정의할 경우 /WEB-INF/tiles-def.xml 파일을 로드하여 각 view 이름에 맞는 tiles view를 리턴해 준다.

3.Tiles definition 파일 작성

Tiles를 사용하기 위해서는 실제 Controller에서 리턴된 view 이름을 토대로 페이지에 출력해줄 tiles attribute를 정의해주는 tiles definition을 정의해야 한다. (위의 tilesConfigurer 위치로 정의한 tiles-def.xml 파일) 다음은 tiles definition 정의 예이다.

```
<definition name="template" template="/sample/layouts/layout.jsp">
    <put-attribute name="header" value="/sample/layouts/top.jsp" />
    <put-attribute name="body" value="/sample/layouts/welcome.jsp" />
    <put-attribute name="footer" value="/sample/layouts/left.jsp" />
</definition>
<definition name="listCategory" extends="template">
    <put-attribute name="body" value="/sample/category/listCategory.jsp" />
</definition>
```

먼저 Layout을 정의한 jsp 페이지를 정의한다. 해당 layout.jsp 페이지에서 기본적으로 사용할 페이지 구성 요소(위의 예에선 header, body, footer)들을 정의한 후 다른 view들은 미리 정의된 template이라는 definition을 extends하여 body만 설정하여 사용할 수 있다. 위의 예에서 listCategory라는 이름의 view가 리턴될 경우 "/sample/layouts/layout.jsp" 페이지의 레이아웃으로 header에는 "/sample/layouts/top.jsp" body는 "/sample/category/listCategory.jsp", footer는 "/sample/layouts/left.jsp"이 될 것이다. JSP에서 tiles 구성 요소를 넣을 때는 아래와 같이 tiles taglib을 정의한 후 <tiles:insertAttribute> 태그를 이용하여 사용한다.

```
<definition name="template" template="/sample/layouts/layout.jsp">
    <put-attribute name="header" value="/sample/layouts/top.jsp" />
    <put-attribute name="body" value="/sample/layouts/welcome.jsp" />
    <put-attribute name="footer" value="/sample/layouts/left.jsp" />
</definition>
<definition name="listCategory" extends="template">
    <put-attribute name="body" value="/sample/category/listCategory.jsp" />
</definition>
```

name attribute에 들어갈 이름은 tiles definition 파일의 name attribute의 이름이 된다.

III.Apache Tiles

Apache Tiles는 Web Application을 개발할 때 화면 Layout을 간단하게 정의할 수 있는 Template Framework이다. Tiles는 개발자가 정의한 Tiles의 요소를 가지고 실행 시 화면을 완성시키게 된다. 이러한 Tiles의 요소는 Tiles Definition xml을 이용해 쉽게 정의할 수 있으며 화면 정의의 중복을 감소시키고 재사용성을 높일 수 있다. Anyframe 4.5.0 이상의 버전에서는 Apache Tiles 2.2 버전을 사용하고 있으며 이 장에서는 Tiles에 대해 알아보도록 한다.

4.Features

Tiles 2에서 새롭게 제공하는 Tiles의 주요 특징은 아래와 같다.

- Nested Definition 지원
- Freemarker, Velocity 지원
- Regular Expression을 이용한 패턴 매칭 지원
- OGNL, MVEL 지원

5.Installation

Apache Tiles는 아래의 4개의 프로젝트로 구성되어 있다.

- tiles-core
- tiles-api
- tiles-servlet
- tiles-jsp

위 프로젝트는 버전에 따라 Apache Tiles 홈페이지에서 다운로드 받을 수 있으며 Anyframe 4.5.0 이상의 버전 설치시 Tiles 2.2.1 버전이 자동 배포 된다. 또한, Tiles2.2를 사용하기 위해서는 아래와 같은 참조 라이브러리가 필요하다.

- Jakarta Commons BeanUtils(1.8.0 이상)
- Jakarta Commons Digester(2.0 이상)
- SLF4j(API와 구현체, 1.5.8 이상)

이 또한, Anyframe 설치시 자동 배포 된다.

6.구성 요소

Tiles는 기본적으로 Composite View Pattern을 사용하며 Template, Attribute, Definition으로 구성된다.

- Template : 페이지의 레이아웃이 되며 attribute를 호출하여 해당 페이지를 채운다.
- Attribute : Template으로 정의된 빈 공간을 채운다. Attribute는 아래의 3가지 타입을 가진다.
  - string : 직접 화면에 출력할 문자열
  - template : attribute를 가지거나 가지지 않는 template. attribute를 가지고 있다면 그 attribute 또한 채워져야 함
  - definition : 재사용 가능한 페이지를 조합함. 페이지에 포함되는 모든 attribute는 채워져야 함
- Definition : end-user를 위한 화면에 출력할 구성이다. 필수적으로 template들의 조합으로 이루어 지며 전체적 또는 부분적으로 attribute들을 채운다. 모든 attribute가 채워져 있으면 해당 attribute들을 포함한 화면을 출력하며 채워져 있지 않은 attribute에 대해서는 extended definition으로 정의되어 있는 definition의 attribute들을 채우게 된다.

7.화면 개발

Tiles 기반의 페이지를 개발하기 위해서는 먼저 template을 생성해야 한다. Anyframe Tiles Plugin 설치로 생성되게 되는 Anyframe Application은 다음과 같은 화면 Layout을 가진다.

위의 그림에서 보는바와 같이 화면 Layout은 top, left, body의 구성요소로 이루어져 있다. 이러한 화면 Layout을 정의하기 위해 해당 Layout을 정의한 template 파일을 생성한다. attribute의 삽입에 대해서는 <tiles:insertAttribute> tag를 사용하고 이 tag의 사용을 위해 taglib 또한 정의되어야 한다. 다음은 해당 Layout을 정의하고 있는standard.jsp 파일의 일부이다.

```
<%@ taglib prefix="tiles" uri="http://tiles.apache.org/tags-tiles" %>
<body>
<table width="100%" height="100%" border="0" cellpadding="0" cellspacing="0">
  <tr>
  <!---- Left Menu ------>
    <td width="177" height="100%" align="left" valign="top"  bgcolor="#eeeeee">
     <div id="left">
      <tiles:insertAttribute name="left"/>
     </div>
    </td>
 <!---- Body ------>
    <td width="100%" height="100%" align="left" valign="top" style="padding:0 20px 0 20px">

  <div id="body">
        <tiles:insertAttribute name="body"/>
       </div>
    </td>
  </tr>
</table>
</body>
```

위와 같이 template JSP 파일을 정의하고 해당 Definition을 정의한 xml 파일을 작성한다. 다음 코드는 위의 template을 정의한 Definition 파일의 일부이다.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE tiles-definitions PUBLIC
"-//Apache Software Foundation//DTD Tiles Configuration 2.1//EN"
"http://tiles.apache.org/dtds/tiles-config_2_1.dtd">
<tiles-definitions>
     <definition name="tilesLayout" template="/sample/tiles/standard.jsp">
        <put-attribute name="left" value="/sample/tiles/left.jsp" />
        <put-attribute name="body" value="/sample/tiles/welcome.jsp" />
    </definition>
</tiles-definitions
```

위의 Tiles Definition 파일에서 볼 수 있듯이 "/sample/tiles/standard.jsp"을 template으로 가지며 "left", "body"의 attribute를 채울 요소들을 지정해 주고 있다.

8.EL

Tiles를 사용하면 template의 요소가 되는 attribute에 대한 값을 Tiles Definition xml 파일에 정의해 줘야한다. 이 때, 각각의 view 마다 하나의 definition을 매번 정의해줘야 한다. 이에 Apache Tiles는 2.1 버전 부터 EL(Expression Language)의 사용을 지원해준다. 단, servlet spec은 2.5 이상이 되어야 한다. EL을 이용한 Tiles Definition 정의는 아래와 같다.

```
<definition name="tilesLayout" templateExpression="${layout}">
    <put-attribute name="left" value="/sample/layouts/left.jsp" />
    <put-attribute name="body" expression="${requestScope.body}" />
</definition>
```

위와 같이 정의할 경우 template의 이름은 모든 Scope 내에서 "layout"이라는 이름의 객체를 찾게 될 것이다. 또한, body는 reqeust Scope에 있는 "body"로 부터 값을 추출하게 된다.