MODULE 7

TEMPLATE

Françoise Dubisy

TABLE OF CONTENT

Why to Use a Template?

- To display several pages in the same way
- To maintain a standard look and feel in an application with a large number of pages
- To make change once and for all
- To reuse code and avoid recreating similarly constructed pages

How to Use a Template?

- Configuration
  - pom.xml
  - Configuration class
  - tiles.xml
- Create a template page
  - Defining the default page structure
  - Acting as the base for the other pages
- Create client pages
  - Pages built based on the template
  - By inserting specific content into the template

Configuration – *pom.xml*

- Add a dependency

```
<dependency>
		<groupId>org.apache.tiles</groupId>
		<artifactId>tiles-jsp</artifactId>
		<version>3.0.4</version>
</dependency>
```

Configuration – *TilesConfig class*

- Remove ***InternalResourceViewResolver*** from Configuration class

```
@Configuration
public class MainConfig extends WebMvcConfigurerAdapter {

        @Bean
        public ViewResolver viewResolver ()
        {
                InternalResourceViewResolver resolver = new InternalResourceViewResolver();
                resolver.setPrefix("/WEB-INF/jsp/");
                resolver.setSuffix(".jsp");

                return resolver;
        }
}
```

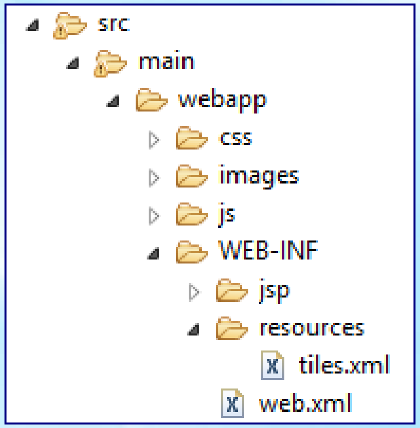
Configuration – *TilesConfig class*

- Use ***TilesViewResolver***

```
@Configuration
public class TilesConfig {
		@Bean
		public TilesConfigurer tilesConfirgurer()
		{
			final TilesConfigurer configurer = new TilesConfigurer();
			configurer.setDefinitions(new String[] {"WEB-INF/resources/tiles.xml"});
			configurer.setCheckRefresh(true);
			return configurer;
		}

		@Bean
		public ViewResolver tilesViewResolver ()
		{
			final TilesViewResolver resolver = new TilesViewResolver();
			resolver.setViewClass(TilesView.class);
			resolver.setExposeContextBeansAsAttributes(true);
			return resolver;
		}
}
```

Configuration – *tiles.xml*

- Create ***tiles.xml***
  - Definition of the template
  - Prefix of the url of pages to apply template



```
src
  main
    webapp
      css
      images
      js
      WEB-INF
        jsp
        resources
          tiles.xml
        web.xml
```

Configuration – *tiles.xml*

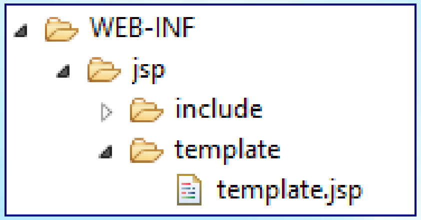
Location of the template

```
<definition name="template-main" template="/WEB-INF/jsp/template/template.jsp">
<put-attribute name="main-content" value="" />
</definition>

<!-- "ajax:" renders the page as-is, without the template -->
<definition name="ajax:**" template="/WEB-INF/jsp/ajax/{1}.jsp" />

<!-- "tiles:" renders the specified page within the template-main -->
<definition name="integrated:**" extends="template-main">
<put-attribute name="main-content" value="/WEB-INF/jsp/{1}.jsp" />
</definition>

<definition name="error" extends="template-main">
<put-attribute name="main-content" value="/WEB-INF/jsp/error.jsp" />
</definition>
```

Url of pages must be preceded by "***integrated:***" in values returned by get/post methods of controllers,
In order to apply template

Template Page

- Create template.jsp



```
◢ WEB-INF
    ◢ jsp
        ▷ include
        ◢ template
            template.jsp
```

- Import tiles tag library

```
<%@ taglib uri="http://tiles.apache.org/tags-tiles" prefix="tiles"%>
```

- Use ***<tiles:insertAttribute>*** to define editable area
  - i.e, the area to be replaced by the content of the client page

```
<div>
        <tiles:insertAttribute name="main-content" />
</div>
```

Adapt Page Url in Controller

- Adapt get/post methods in Controller in order to apply template
  - Prefix url of pages returned by the methods with prefix value specified in the tiles.xml
  - E.g,

```
@Controller
public class ColorController {

        @RequestMapping(method=RequestMethod.GET)
        public String home(Model model ...)
        {               ...
                return "integrated:colorPage";
        }

        @RequestMapping method=RequestMethod.POST)
        public String getFormData(Model model...)
        {               ...
                return "integrated:showColorPage";
        }
}
```