

我的博客

[🏠访问我的空间](#)

[返回全部博客列表](#)

永久链接: <http://comedsh.iteye.com/blog/2240037>

预览文章: **Spring 4.2.2 and apache tiles 3.0.5 整合**



2015-09-01

- 博客分类:
- [spring 模板技术](#)

[apache tiles](#)

记录 Spring 4.2.2 整合 apache tiles 3.0.5 的步骤

1. pom.xml

Java代码

- <dependency>
- <groupId>org.apache.tiles</groupId>
- <artifactId>tiles-extras</artifactId>
- <version>3.0.5</version>
- </dependency>

Java代码

- <dependency>
- <groupId>org.apache.tiles</groupId>
- <artifactId>tiles-extras</artifactId>
- <version>3.0.5</version>
- </dependency>

2. web.xml

Xml代码

- <listener>
- <listener-class>org.apache.tiles.extras.complete.CompleteAutoloadTilesListener</listener-class>
- </listener>

Xml代码

- <listener>
- <listener-class>org.apache.tiles.extras.complete.CompleteAutoloadTilesListener</listener-class>
- </listener>

官网教程有提到添加该 listener，但是，这里不建议使用，为什么，因为这里添加了该 tiles listener，它会默认覆盖掉其它的配置，也就是说如果在 Spring 中 通过 TilesConfigurer 配置的模板路径也就不会生效了；

CompleteAutoloadTilesListener 默认回到 /WEB-INF/tiels.xml 和 /WEB-INF/**/tiels.xml 的子目录中找 tiles.xml 的模板配置文件

Xml代码

- <servlet>
- <servlet-name>Tiles Dispatch Servlet</servlet-name>
- <servlet-class>org.apache.tiles.web.util.TilesDispatchServlet</servlet-class>
- </servlet>
-
- <servlet-mapping>
- <servlet-name>Tiles Dispatch Servlet</servlet-name>

```
8. <url-pattern>*.tiles</url-pattern>
9. </servlet-mapping>
```

Xml代码

```
1. <servlet>
2.   <servlet-name>Tiles Dispatch Servlet</servlet-name>
3.   <servlet-class>org.apache.tiles.web.util.TilesDispatchServlet</servlet-class>
4. </servlet>
5.
6. <servlet-mapping>
7.   <servlet-name>Tiles Dispatch Servlet</servlet-name>
8.   <url-pattern>*.tiles</url-pattern>
9. </servlet-mapping>
```

这里，意思是，模板可以通过 模板名字.tiles 进行访问，这里也不建议配置，因为 Spring MVC 的 TilesViewResolver 可以进行模板的访问。

3. Spring 配置

Xml代码

```
1. <!-- 自动将静态资源导向 -->
2. <!-- <mvc:resources mapping="/resources/**" location="/resources/" /> -->
3.
4. <!-- Order 的意义：意思是说，当有多个 view resovler 的时候，按照 order 的数字大小，由小到大依次解析；
   很关键 -->
5. <bean id="tilesViewResolver" class="org.springframework.web.servlet.view.tiles3.TilesViewResolver">
6.   <property name="viewClass" value="org.springframework.web.servlet.view.tiles3.TilesView"></property>
7.   <property name="order" value="0"></property>
8. </bean>
9.
10. <bean id="viewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver">
11.   <property name="viewClass" value="org.springframework.web.servlet.view.JstlView" />
12.   <property name="prefix" value="/"></property>
13.   <property name="suffix" value=".jsp"></property>
14.   <property name="order" value="1"></property>
15. </bean>
16.
17. <!-- 又是一个坑，Spring 4.2.2 默认启动时不会初始化 bean，而之前 3.x 会自动初始化，
18.     所以这里需要设置 lazy-init="false"，让 TilesConfigurer 在 Spring 容器启动时就初始化，否则，
   tilesConfigurer 加载 definitions 失败 -->
19. <bean id="tilesConfigurer" class="org.springframework.web.servlet.view.tiles3.TilesConfigurer" lazy-init="false">
20.
21.   <property name="definitions">
22.     <list>
23.       <value>/WEB-INF/layout/layouts.xml</value>
24.       <value>/WEB-INF/layout/views.xml</value>
25.     </list>
26.   </property>
27.
28.   <property name="preparerFactoryClass" value="org.springframework.web.servlet.view.tiles3.SpringBeanPreparerFactory" />
29.
30. </bean>
```

Xml代码

```
1. <!-- 自动将静态资源导向 -->
2. <!-- <mvc:resources mapping="/resources/**" location="/resources/" /> -->
3.
4. <!-- Order 的意义：意思是说，当有多个 view resovler 的时候，按照 order 的数字大小，由小到大依次解析；
   很关键 -->
5. <bean id="tilesViewResolver" class="org.springframework.web.servlet.view.tiles3.TilesViewResolver">
6.   <property name="viewClass" value="org.springframework.web.servlet.view.tiles3.TilesView"></property>
7.   <property name="order" value="0"></property>
8. </bean>
9.
10. <bean id="viewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver">
11.   <property name="viewClass" value="org.springframework.web.servlet.view.JstlView" />
12.   <property name="prefix" value="/"></property>
13.   <property name="suffix" value=".jsp"></property>
14.   <property name="order" value="1"></property>
```

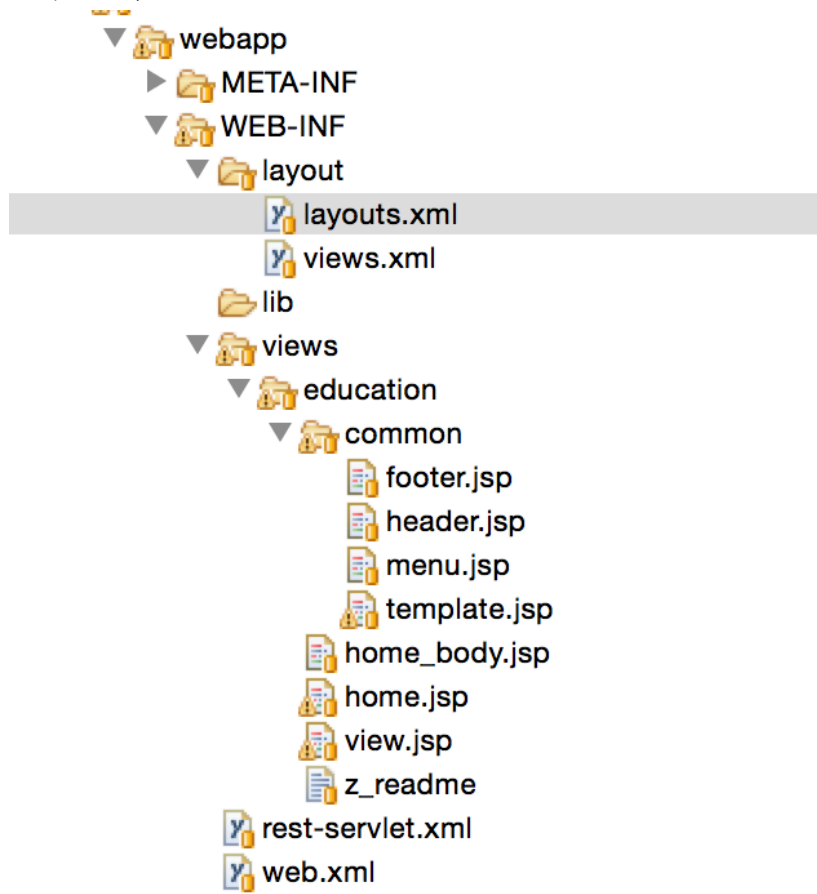
```

15. </bean>
16.
17. <!-- 又是一个坑，Spring 4.2.2 默认启动时不会初始化 bean，而之前 3.x 会自动初始化，
18.     所以这里需要设置 lazy-init="false"，让 TilesConfigurer 在 Spring 容器启动时就初始化，否则，
    titlesConfigurer 加载 definitions 失败 -->
19. <bean id="tilesConfigurer" class="org.springframework.web.servlet.view.tiles3.TilesConfigurer" lazy-init="false">
20.
21.     <property name="definitions">
22.         <list>
23.             <value>/WEB-INF/layout/layouts.xml</value>
24.             <value>/WEB-INF/layout/views.xml</value>
25.         </list>
26.     </property>
27.
28.     <property name="preparerFactoryClass" value="org.springframework.web.servlet.view.tiles3.SpringBeanPreparerFactory" />
29.
30. </bean>

```

注意，Spring 4.2.2 对配置的 bean 不会自动进行初始化，而是默认使用的懒加载，所以在配置 TilesConfigurer 的时候，记得加上 lazy-load="false"，否则配置的模板路径不会生效

4. 工程目录



5. layouts.xml

Xml代码

```

1. <tiles-definitions>
2.
3. <definition name="education.default_template" template="/WEB-INF/views/education/common/template.jsp">
4.
5.     <put-attribute name="title" value="education tutorial homepage" />
6.
7.     <put-attribute name="header" value="/WEB-INF/views/education/common/header.jsp" />
8.
9.     <put-attribute name="menu" value="/WEB-INF/views/education/common/menu.jsp" />
10.
11.     <!-- remain the body as blank, let its inheritance to fill it up -->
12.     <put-attribute name="body" value="" />
13.
14.     <put-attribute name="footer" value="/WEB-INF/views/education/common/footer.jsp" />
15.
16. </definition>
17.
18. </tiles-definitions>

```

Xml代码

```

1. <tiles-definitions>
2.
3. <definition name="education.default_template" template="/WEB-INF/views/education/common/template.jsp">
4.

```

```

5.   <put-attribute name="title" value="education tutorial homepage" />
6.
7.   <put-attribute name="header" value="/WEB-INF/views/education/common/header.jsp" />
8.
9.   <put-attribute name="menu" value="/WEB-INF/views/education/common/menu.jsp" />
10.
11.  <!-- remain the body as blank, let its inheritance to fill it up -->
12.  <put-attribute name="body" value="" />
13.
14.  <put-attribute name="footer" value="/WEB-INF/views/education/common/footer.jsp" />
15.
16. </definition>
17.
18. </tiles-definitions>

```

> template.jsp

模板文件

> title, menu, header, body, footer 全部是用来填充 template.jsp 的

> 因为 body 未填充，所以 education.default_template 是抽象模板，可以被继承模板实现

6. views.xml 子模板

Xml代码

```

1. <tiles-definitions>
2.
3. <!-- defined for education -->
4.
5.   <definition name="education_spittle_home" extends="education.default_template">
6.     <put-attribute name="body" value="/WEB-INF/views/education/home_body.jsp" />
7.   </definition>
8.
9.   <definition name="education_spittle_view" extends="education.default_template">
10.    <put-attribute name="body" value="/WEB-INF/views/education/view.jsp" />
11.  </definition>
12.
13. <!-- defined for education -->
14.
15. </tiles-definitions>

```

Xml代码

```

1. <tiles-definitions>
2.
3. <!-- defined for education -->
4.
5.   <definition name="education_spittle_home" extends="education.default_template">
6.     <put-attribute name="body" value="/WEB-INF/views/education/home_body.jsp" />
7.   </definition>
8.
9.   <definition name="education_spittle_view" extends="education.default_template">
10.    <put-attribute name="body" value="/WEB-INF/views/education/view.jsp" />
11.  </definition>
12.
13. <!-- defined for education -->
14.
15. </tiles-definitions>

```

7. Controller

Java代码

```

1. @RequestMapping("/home")
2. public String home(){
3.
4.    // by using TilesView to render this page. the full template path should be education_spittle_home.tiles.
5.    return "education_spittle_home";
6.
7. }

```

Java代码

```

1. @RequestMapping("/home")
2. public String home(){
3.

```

4. // by using TilesView to render this page. the full template path should be education_spittle_home.tiles.
5. return "education_spittle_home";
- 6.
7. }

然后，浏览器可以直接通过 http://localhost:8080/home 访问模板页面。

- [查看图片附件](#)



comedsh

博客管理

- [发表文章](#)
- [分类管理](#)
- [评论管理](#)
- [做电子书](#)
- [博客设置](#)
- [博客导入](#)

文章管理

- [全部博客 \(97\)](#)
- [Java \(42\)](#)
- [DATABASE \(6\)](#)
- [PostgreSQL \(2\)](#)
- [DB2 \(5\)](#)
- [JavaScript \(2\)](#)
- [Shell \(3\)](#)
- [Windows \(3\)](#)
- [uml \(1\)](#)
- [生活 \(6\)](#)
- [物理 \(1\)](#)
- [Daily Works \(1\)](#)
- [电脑硬件 \(5\)](#)
- [Game \(2\)](#)
- [数学 \(1\)](#)
- [Internet \(1\)](#)
- [project \(1\)](#)
- [汽车 \(3\)](#)
- [html \(1\)](#)
- [Hibernate \(1\)](#)
- [DWR \(1\)](#)
- [CSS \(1\)](#)
- [计算机 \(0\)](#)
- [tomcat \(1\)](#)
- [linux \(1\)](#)
- [git \(1\)](#)
- [macosx apache \(1\)](#)
- [git gitlab \(1\)](#)
- [apache ftp \(1\)](#)
- [spring 模板技术 \(0\)](#)
- [草稿箱 \(51\)](#)

- [回收站\(1\)](#)

声明：ITeye文章版权属于作者，受法律保护。没有作者书面许可不得转载。若作者同意转载，必须以超链接形式标明文章原始出处和作者。

© 2003-2015 ITeye.com. All rights reserved. [京ICP证110151号 京公网安备110105010620]