Mkyong.com

Home   All Tutorials   Java Core   JSF   Spring   Hibernate   Struts   Android   Others

Configure Multiple View Resolvers Priority In Spring MVC

Posted on August 12, 2010 ,   Last modified : July 27, 2011

By [mkyong](#)

Problem

In Spring MVC application, often times, you may applying few view resolver strategies to resolve the view name. For example, combine three view resolvers together : **InternalResourceViewResolver**, **ResourceBundleViewResolver** and **XmlViewResolver**.

```xml
<beans ...>
	<bean class="org.springframework.web.servlet.view.XmlViewResolver">
		<property name="location">
			<value>/WEB-INF/spring-views.xml</value>
		</property>
	</bean>

	<bean class="org.springframework.web.servlet.view.ResourceBundleViewResolver">
		<property name="basename" value="spring-views" />
	</bean>

	<bean id="viewResolver"
		class="org.springframework.web.servlet.view.InternalResourceViewResolver" >
		<property name="prefix">
			<value>/WEB-INF/pages/</value>
		</property>
		<property name="suffix">
			<value>.jsp</value>
		</property>
	</bean>
</beans>
```

But, if a view name is returned, which view resolver strategy will be used?

Solution

If multiple view resolver strategies are applied, you have to declare the priority through "**order**" property, where the **lower order value has a higher priority**, for example :

```xml
<beans ...>
	<bean class="org.springframework.web.servlet.view.XmlViewResolver">
		<property name="location">
			<value>/WEB-INF/spring-views.xml</value>
		</property>
		<property name="order" value="0" />
	</bean>

	<bean class="org.springframework.web.servlet.view.ResourceBundleViewResolver">
		<property name="basename" value="spring-views" />
		<property name="order" value="1" />
	</bean>

	<bean id="viewResolver"
		class="org.springframework.web.servlet.view.InternalResourceViewResolver" >
		<property name="prefix">
			<value>/WEB-INF/pages/</value>
		</property>
		<property name="suffix">
			<value>.jsp</value>
		</property>
		<property name="order" value="2" />
	</bean>
```

```
</beans>
```

Now, if a view name is returned, the view resolving strategy works in the following order :

```
XmlViewResolver --> ResourceBundleViewResolver --> InternalResourceViewResolver
```

> **Note**
> The **InternalResourceViewResolver** must always **assign with the lowest priority** (largest order number), because it will resolve the view no matter what view name is returned. It caused other view resolvers have no chance to resolve the view if they have lower priority.

Download Source Code

Download it – SpringMVC-ViewResolver-Priority-Example.zip (7KB)

References

1. Spring MVC InternalResourceViewResolver example
2. Spring MVC XmlViewResolver example
3. Spring MVC ResourceBundleViewResolver example

Tags : spring mvc

mkyong

Founder of Mkyong.com, love Java and open source stuff. Follow him on Twitter, or befriend him on Facebook or Google Plus. If you like my tutorials, consider make a donation to these charities.

My website mkyong.com is hosted by Liquid Web, I really like their fast and helpful support, it saved me a lot of time. Furthermore, their servers are extremely reliable, fast and stable, I don't remember when was the last time we had server downtime. Btw, they're offering $100 Coupon now, if you are sourcing for great hosting provider, go grab it.

Related Posts

▶ Ant – Spring MVC and WAR file Example
▶ Gradle – Spring MVC Web Project Example
▶ How To Get HTTP Request Header In Java
▶ Spring MVC – How to include JS or CSS files in a JSP page
▶ Spring MVC + jQuery Autocomplete example

Popular Posts

▶ Top 8 Java People You Should Know
▶ Top 20 Java Websites
▶ Top 5 Free Java eBooks
▶ Top 10 Java Regular Expression Examples
▶ Top 5 Open Source Q&A Systems

You might also like following tutorials :