

<http://baeldung.com>

A Guide to the ViewResolver in Spring MVC

Last modified: October 22, 2017

by [baeldung](http://www.baeldung.com/author/baeldung/) (<http://www.baeldung.com/author/baeldung/>)

Spring (<http://www.baeldung.com/category/spring/>) +
Spring MVC (<http://www.baeldung.com/category/spring-mvc/>)

I just announced the new *Spring 5* modules in REST With Spring:

>> CHECK OUT THE COURSE (</rest-with-spring-course#new-modules>)

1. Overview

All MVC frameworks provide a way of working with views.

Spring does that via the view resolvers, which enable you to render models in the browser without tying the implementation to a specific view technology.

The *ViewResolver* maps view names to actual views.

And the Spring framework comes with quite a few view resolvers e.g. *InternalResourceViewResolver*, *XmlViewResolver*, *ResourceBundleViewResolver* and a few others.

This is a simple tutorial showing how to set up the most common view resolvers and **how to use multiple *ViewResolver* in the same configuration**.

2. The Spring Web Configuration

Let's start with the web configuration; we'll annotate it with `@EnableWebMvc`, `@Configuration` and `@ComponentScan`:

```
1 @EnableWebMvc
2 @Configuration
3 @ComponentScan("org.baeldung.web")
4 public class WebConfig extends WebMvcConfigurerAdapter {
5     // All web configuration will go here
6 }
```

It's here that we'll set up our view resolver in the configuration.

3. Add an *InternalResourceViewResolver*

This *ViewResolver* allows us to set properties such as prefix or suffix to the view name to generate the final view page URL:

```
1  @Bean
2  public ViewResolver internalResourceViewResolver() {
3      InternalResourceViewResolver bean = new InternalResourceViewResolver();
4      bean.setViewClass(JstlView.class);
5      bean.setPrefix("/WEB-INF/view/");
6      bean.setSuffix(".jsp");
7      return bean;
8  }
```

For such simplicity of the example, we don't need a controller to process the request.

We only need a simple *jsp* page, placed in the */WEB-INF/view* folder as defined in the configuration:

```
1  <html>
2      <head></head>
3      <body>
4          <h1>This is the body of the sample view</h1>
5      </body>
6  </html>
```

4. Add a *ResourceBundleViewResolver*

As the name of this resolver suggest a *ResourceBundleViewResolver* uses bean definitions in a *ResourceBundle*.

First, we add the *ResourceBundleViewResolver* to the previous configuration:

```
1  @Bean
2  public ViewResolver resourceBundleViewResolver() {
3      ResourceBundleViewResolver bean = new ResourceBundleViewResolver();
4      bean.setBasename("views");
5      return bean;
6  }
```

The bundle is typically defined in a properties file, located in the classpath. Below is the *views.properties* file:

```
1  sample.(class)=org.springframework.web.servlet.view.JstlView
2  sample.url=/WEB-INF/view/sample.jsp
```

We can use the simple *jsp* page defined in the above example for this configuration as well.

5. Add an *XmlViewResolver*

This implementation of *ViewResolver* accepts a configuration file written in *XML* with the same *DTD* as Spring's *XML* bean factories:

```
1  @Bean
2  public ViewResolver xmlViewResolver() {
3      XmlViewResolver bean = new XmlViewResolver();
4      bean.setLocation(new ClassPathResource("views.xml"));
5      return bean;
6  }
```

Below is the configuration file, *views.xml*:

```
1  <beans xmlns="http://www.springframework.org/schema/beans (http://www.springframework.org/schema/beans)"
2      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance (http://www.w3.org/2001/XMLSchema-instance)"
3      xsi:schemaLocation="http://www.springframework.org/schema/beans (http://www.springframework.org/schema/beans)
4      http://www.springframework.org/schema/beans/spring-beans-4.2.xsd (http://www.springframework.org/schema/beans/spring
5
6      <bean id="xmlConfig" class="org.springframework.web.servlet.view.JstlView">
7          <property name="url" value="/WEB-INF/view/xmlSample.jsp" />
8      </bean>
9
10 </beans>
```

As for the previous examples we can use our simple *jsp* page defined previously.

6. Chaining *ViewResolvers* and Define an Order Priority

Spring MVC also supports **multiple view resolvers**.

This allow you to override specific views in some circumstances. We can simply chain view resolvers by adding more than one resolver to the configuration.

Once we've done that, we'll need to define an order for these resolvers. The **order property** is used to define which is the order of invocations in the chain. The higher the order property (largest order number), the later the view resolver is positioned in the chain.

To define the order we can add the follow line of code to the configuration of the our view resolvers:

```
1 | bean.setOrder(0);
```

Be careful on the order priority as the *InternalResourceViewResolver* should have a higher order – because it's intended to represent a very explicit mapping. And if other resolvers have a higher order, then the *InternalResourceViewResolver* might never be invoked.

7. Conclusion

In this tutorial we configured a chain of view resolvers using Java configuration. By playing with the order priority we can set the order of their invocation.

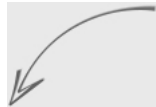
The implementation of this simple tutorial can be found in the github project (<https://github.com/eugenp/tutorials/tree/master/spring-mvc-java>) – this is an Eclipse based project, so it should be easy to import and run as it is.

I just announced the new Spring 5 modules in REST With Spring:

>> CHECK OUT THE LESSONS (</rest-with-spring-course#new-modules>)



Build your Microservice Architecture with Spring Boot and Spring Cloud



Enter your email Address

Download Now

Sort by: newest

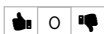


Guest

Bhargav Patel



Why we need view resolver in spring ? I am asking you because in other java frame work we don't need any view resolver or similar to it. like Servlet



1 year 5 months ago ^



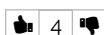
Guest

Eugen Paraschiv

(<http://www.baeldung.com/>)



Well Bhargav – different frameworks make different abstractions available – in this case, Spring does come with a lot of flexibility and one of this is the ability to control how views are resolved. Hope that clears things up – at least a bit.
Cheers,
Eugen.



1 year 5 months ago

CATEGORIES

[SPRING](http://www.baeldung.com/category/spring/) ([HTTP://WWW.BAELDUNG.COM/CATEGORY/SPRING/](http://www.baeldung.com/category/spring/))

[REST](http://www.baeldung.com/category/rest/) ([HTTP://WWW.BAELDUNG.COM/CATEGORY/REST/](http://www.baeldung.com/category/rest/))

[JAVA](http://www.baeldung.com/category/java/) ([HTTP://WWW.BAELDUNG.COM/CATEGORY/JAVA/](http://www.baeldung.com/category/java/))

[SECURITY](http://www.baeldung.com/category/security-2/) ([HTTP://WWW.BAELDUNG.COM/CATEGORY/SECURITY-2/](http://www.baeldung.com/category/security-2/))

[PERSISTENCE](http://www.baeldung.com/category/persistence/) ([HTTP://WWW.BAELDUNG.COM/CATEGORY/PERSISTENCE/](http://www.baeldung.com/category/persistence/))

[JACKSON](http://www.baeldung.com/category/jackson/) ([HTTP://WWW.BAELDUNG.COM/CATEGORY/JACKSON/](http://www.baeldung.com/category/jackson/))

[HTTPCLIENT](http://www.baeldung.com/category/http/) ([HTTP://WWW.BAELDUNG.COM/CATEGORY/HTTP/](http://www.baeldung.com/category/http/))

ABOUT

[ABOUT BAELDUNG](http://www.baeldung.com/about/) ([HTTP://WWW.BAELDUNG.COM/ABOUT/](http://www.baeldung.com/about/))

[THE COURSES](http://courses.baeldung.com) ([HTTP://COURSES.BAELDUNG.COM](http://courses.baeldung.com))

[META BAELDUNG](http://meta.baeldung.com/) ([HTTP://META.BAELDUNG.COM/](http://meta.baeldung.com/))

[THE FULL ARCHIVE](http://www.baeldung.com/full_archive) ([HTTP://WWW.BAELDUNG.COM/FULL_ARCHIVE](http://www.baeldung.com/full_archive))

[WRITE FOR BAELDUNG](http://www.baeldung.com/contribution-guidelines) ([HTTP://WWW.BAELDUNG.COM/CONTRIBUTION-GUIDELINES](http://www.baeldung.com/contribution-guidelines))

[PRIVACY POLICY](http://www.baeldung.com/privacy-policy) ([HTTP://WWW.BAELDUNG.COM/PRIVACY-POLICY](http://www.baeldung.com/privacy-policy))

[TERMS OF SERVICE](http://www.baeldung.com/terms-of-service) ([HTTP://WWW.BAELDUNG.COM/TERMS-OF-SERVICE](http://www.baeldung.com/terms-of-service))

[CONTACT](http://www.baeldung.com/contact) ([HTTP://WWW.BAELDUNG.COM/CONTACT](http://www.baeldung.com/contact))

[COMPANY INFO](http://www.baeldung.com/baeldung-company-info) ([HTTP://WWW.BAELDUNG.COM/BAELDUNG-COMPANY-INFO](http://www.baeldung.com/baeldung-company-info))

[ADVERTISE ON THE JAVA WEEKLY](http://www.baeldung.com/java-weekly-sponsorship) ([HTTP://WWW.BAELDUNG.COM/JAVA-WEEKLY-SPONSORSHIP](http://www.baeldung.com/java-weekly-sponsorship))

[CONSULTING WORK](http://www.baeldung.com/consulting) ([HTTP://WWW.BAELDUNG.COM/CONSULTING](http://www.baeldung.com/consulting))