

# Can I compose a Spring Configuration File from smaller ones?

Asked 16 years, 3 months ago    Modified 14 years, 5 months ago

Viewed 16k times



15



I have a handful of projects that all use one project for the data model. Each of these projects has its own applicationContext.xml file with a bunch of repetitive data stuff within it.

I'd like to have a modelContext.xml file and another for my ui.xml, etc.

Can I do this?

java

spring

Share

Improve this question

Follow

asked Sep 18, 2008 at 17:10



Allain Lalonde

93.2k ● 71 ● 189 ● 238

7 Answers

Sorted by:

Highest score (default)



From the [Spring Docs \(v 2.5.5 Section 3.2.2.1.\)](#):



19



It can often be useful to split up container definitions into multiple XML files. One way to then load an application context which is configured from all these XML fragments is to use the application context constructor which takes multiple Resource locations. With a bean factory, a bean definition reader can be used multiple times to read definitions from each file in turn.

Generally, the Spring team prefers the above approach, since it keeps container configuration files unaware of the fact that they are being combined with others. An alternate approach is to use one or more occurrences of the element to load bean definitions from another file (or files). Let's look at a sample:

```
<import resource="services.xml"/>
<import resource="resources/messageSource.xml"/>
<import resource="/resources/themeSource.xml"/>

<bean id="bean1" class="..." />
<bean id="bean2" class="..." />
```

In this example, external bean definitions are being loaded from 3 files, services.xml, messageSource.xml, and themeSource.xml. All location paths are considered relative to the definition file doing the importing, so services.xml in this case must be in the same directory or classpath location as the file doing the importing,

while messageSource.xml and themeSource.xml must be in a resources location below the location of the importing file. As you can see, a leading slash is actually ignored, but given that these are considered relative paths, it is probably better form not to use the slash at all. The contents of the files being imported must be valid XML bean definition files according to the Spring Schema or DTD, including the top level element.

Share Improve this answer

answered Sep 18, 2008 at 17:18

Follow



[Nicholas Trandem](#)

2,815 ● 5 ● 30 ● 32

---

Absolutely agree with the Spring docs: aggregation of config files beats explicit imports every time. If for nothing else than for unit testing. – [Boris Terzic](#) Sep 18, 2008 at 18:04

---



3



We do this in our projects at work, using the classpath\* resource loader in Spring. For a certain app, all appcontext files containing the application id will be loaded:

```
classpath*:springconfig/spring-appname-*.xml
```

Share Improve this answer

answered Sep 18, 2008 at 17:39

Follow



[Asgeir S. Nilsen](#)

1,137 ● 9 ● 13

---

I'm confused. new

ClassPathResource("classpath\*:springconfig/spring-appname-\*.xml") is what you're proposing? – [Allain Lalonde](#)

Sep 18, 2008 at 17:59

- 
- 1 have to disagree. we used to work with this approach, and it always leads to loading the wrong config file. in our case, we had the spring XML files in a special "conf" folder, but obviously they were also hiding inside the various jars. this approach loads ALL config files from anywhere in the classpath, overriding the ones you believe are loaded.  
– [ihadanny](#) Nov 30, 2011 at 13:34

---

Important context: "appname" is of course specific to the application we are building, and the various spring context files are in Maven modules where the most specific application module inherits the other parts from its dependencies. Also, the file name pattern should protect you from picking up random files. – [Asgeir S. Nilsen](#) Dec 2, 2011 at 21:48 ✎

---



Yes, you can do this via the import element.

2

```
<import resource="services.xml"/>
```



Each element's resource attribute is a valid path (e.g. classpath:foo.xml)



Share Improve this answer

Follow

answered Sep 18, 2008 at 17:18



[enricopulatz](#)

4,469 ● 2 ● 19 ● 8

---



2

Given what Nicholas pointed me to I found this in the docs. It allows me to pick at runtime the bean contexts I'm interested in.



```

GenericApplicationContext ctx = new GenericApplicationContext()
XmlBeanDefinitionReader xmlReader = new XmlBeanDefinitionReader(ctx)
xmlReader.loadBeanDefinitions(new ClassPathResource("my-context.xml"));
xmlReader.loadBeanDefinitions(new ClassPathResource("ui-context.xml"));
ctx.refresh();

```



Share Improve this answer

Follow

answered Sep 18, 2008 at 18:00



Allain Lalonde

93.2k ● 71 ● 189 ● 238



1

Here's what I've done for one of my projects. In your `web.xml` file, you can define the Spring bean files you want your application to use:



```

<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>
    /WEB-INF/applicationContext.xml
    /WEB-INF/modelContext.xml
    /WEB-INF/ui.xml
  </param-value>
</context-param>

```

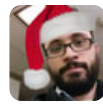


If this isn't defined in your `web.xml`, it automatically looks for `/WEB-INF/applicationContext.xml`

Share Improve this answer

Follow

answered Sep 18, 2008 at 19:28



Michael

35.3k ● 17 ● 78 ● 112

---

I'm not using Spring MVC, just the IOC – [Allain Lalonde](#)

Sep 18, 2008 at 20:04

---



0

Another thing to note is that although you can do this, if you aren't a big fan of XML you can do a lot of stuff in Spring 2.5 with annotations.



Share Improve this answer

answered Sep 18, 2008 at 20:31

Follow



bpapa

21.5k ● 25 ● 100 ● 147



0

Yes, you can using the tag inside the "Master" bean file. But what about the why? Why not listing the files in the contextConfigLocation context param of the wab.xml or als locations array of the bean factory?



I think mutliple files are much easier to handle. You may choose only some of them for a test, simply add rename or remove a part of the application and you may bundle different applications with the same config files (a webapp and a commandline version with some overlapping bean definitions).



Share Improve this answer

answered Sep 19, 2008 at 20:47

Follow



Arne Burmeister

