[all and sundry](http://www.java-allandsundry.com/)

Biju Kunjummen's Java/Spring related interests

**Saturday, August 18, 2012**

@ContextConfiguration defaults

Spring [@ContextConfiguration](http://static.springsource.org/spring/docs/3.1.x/javadoc-api/org/springframework/test/context/ContextConfiguration.html) is a way to specify the Application Context for a test.  
  
The location of a xml based test application context can be specified using the locations attribute:

[?](http://www.java-allandsundry.com/2012/08/contextconfiguration-defaults.html)

|  |  |
| --- | --- |
| 1 | @ContextConfiguration(locations={"test-context.xml"}) |

and if @Configuration is used as the context, then a classes attibute can be specified:

[?](http://www.java-allandsundry.com/2012/08/contextconfiguration-defaults.html)

|  |  |
| --- | --- |
| 1 | @ContextConfiguration(classes={TestConfiguration.class}) |

There are intelligent defaults for these attributes though and that is what I wanted to highlight in this post.  
  
If the locations or the classes attribute is not specified, the default behavior is to first look for a xml configuration with [a name as the test class name - "context.xml" file](http://static.springsource.org/spring/docs/3.1.x/javadoc-api/org/springframework/test/context/support/AbstractContextLoader.html#generateDefaultLocations(java.lang.Class))  
  
For eg. if I have a Test class this way:

[?](http://www.java-allandsundry.com/2012/08/contextconfiguration-defaults.html)

|  |  |
| --- | --- |
| 1  2  3 | @RunWith(SpringJUnit4ClassRunner.class)  @ContextConfiguration  public class TestSpringCache { |

The location of the configuration that will be tried first is "TestSpringCache-context.xml"  
  
If a context is not found at this location, then a @Configuration default is looked for by scanning all static inner classes annotated with @Configuration of the test class. So if I had the following:

[?](http://www.java-allandsundry.com/2012/08/contextconfiguration-defaults.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | @RunWith(SpringJUnit4ClassRunner.class)  @ContextConfiguration  public class TestSpringCache {     ...     @Configuration   @EnableCaching   @ComponentScan("org.bk.samples.cache")   public static class TestConfiguration{   .. |

the inner class TestConfiguration would be used as the source of the Application context.  
  
The default behaviour can be changed by supplying a loader attribute to the @ContextConfiguration, say for eg, if I want to default to @Configuration with its defaults, it can done this way:

[?](http://www.java-allandsundry.com/2012/08/contextconfiguration-defaults.html)

|  |  |
| --- | --- |
| 1  2  3  4 | @RunWith(SpringJUnit4ClassRunner.class)  @ContextConfiguration(loader=AnnotationConfigContextLoader.class)  public class TestSpringCache {  .. |

So to conclude, the defaults provided by @Configuration are a great way to make the tests a little more concise!

<http://www.java-allandsundry.com/2012/08/contextconfiguration-defaults.html>

**getResourceSuffix**

public [String](http://download.oracle.com/javase/6/docs/api/java/lang/String.html?is-external=true) **getResourceSuffix**()

Returns "-context.xml".

**Specified by:**

[getResourceSuffix](https://docs.spring.io/spring/docs/3.1.x/javadoc-api/org/springframework/test/context/support/AbstractContextLoader.html#getResourceSuffix()) in class [AbstractContextLoader](https://docs.spring.io/spring/docs/3.1.x/javadoc-api/org/springframework/test/context/support/AbstractContextLoader.html)

**Returns:**

the resource suffix; should not be null or empty

**See Also:**

#generateDefaultLocations()

<https://docs.spring.io/spring/docs/3.1.x/javadoc-api/org/springframework/test/context/support/GenericXmlContextLoader.html>