

Why am I getting a ClassCastException when generating javadocs?

Asked 16 years, 3 months ago Modified 3 years, 10 months ago Viewed 23k times



I'm using ant to generate javadocs, but get this exception over and over - why?

55



I'm using JDK version **1.6.0_06**.



```
[javadoc] java.lang.ClassCastException: com.sun.tools.javadoc.ClassDocImpl
cannot be cast to com.sun.javadoc.AnnotationTypeDoc
    [javadoc] at
com.sun.tools.javadoc.AnnotationDescImpl.annotationType(AnnotationDescImpl.java:46)
    [javadoc] at
com.sun.tools.doclets.formats.html.HtmlDocletWriter.getAnnotations(HtmlDocletWriter.java:100)
    [javadoc] at
com.sun.tools.doclets.formats.html.HtmlDocletWriter.writeAnnotationInfo(HtmlDocletWriter.java:115)
    [javadoc] at
com.sun.tools.doclets.formats.html.HtmlDocletWriter.writeAnnotationInfo(HtmlDocletWriter.java:115)
    [javadoc] at
com.sun.tools.doclets.formats.html.HtmlDocletWriter.writeAnnotationInfo(HtmlDocletWriter.java:115)
    [javadoc] at
com.sun.tools.doclets.formats.html.FieldWriterImpl.writeSignature(FieldWriterImpl.java:100)
    [javadoc] at
com.sun.tools.doclets.internal.toolkit.builders.FieldBuilder.buildSignature(FieldBuilder.java:100)
    [javadoc] at sun.reflect.GeneratedMethodAccessor8.invoke(Unknown Source)
    [javadoc] at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:46)
    [javadoc] at java.lang.reflect.Method.invoke(Method.java:597)
    [javadoc] at
com.sun.tools.doclets.internal.toolkit.builders.FieldBuilder.invokeMethod(FieldBuilder.java:100)
    [javadoc] at
com.sun.tools.doclets.internal.toolkit.builders.AbstractBuilder.build(AbstractBuilder.java:100)
    [javadoc] at
com.sun.tools.doclets.internal.toolkit.builders.AbstractMemberBuilder.build(AbstractMemberBuilder.java:100)
    [javadoc] at
com.sun.tools.doclets.internal.toolkit.builders.FieldBuilder.buildFieldDoc(FieldBuilder.java:100)
    [javadoc] at sun.reflect.GeneratedMethodAccessor51.invoke(Unknown Source)
    [javadoc] at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:46)
    [javadoc] at java.lang.reflect.Method.invoke(Method.java:597)
    [javadoc] at
com.sun.tools.doclets.internal.toolkit.builders.FieldBuilder.invokeMethod(FieldBuilder.java:100)
    [javadoc] at
com.sun.tools.doclets.internal.toolkit.builders.AbstractBuilder.build(AbstractBuilder.java:100)
    [javadoc] at
com.sun.tools.doclets.internal.toolkit.builders.AbstractMemberBuilder.build(AbstractMemberBuilder.java:100)
    [javadoc] at
com.sun.tools.doclets.internal.toolkit.builders.ClassBuilder.buildFieldDetails(ClassBuilder.java:100)
    [javadoc] at sun.reflect.GeneratedMethodAccessor43.invoke(Unknown Source)
    [javadoc] at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:46)
    [javadoc] at java.lang.reflect.Method.invoke(Method.java:597)
    [javadoc] at
com.sun.tools.doclets.internal.toolkit.builders.ClassBuilder.invokeMethod(ClassBuilder.java:100)
```

```

[javadoc]      at
com.sun.tools.doclets.internal.toolkit.builders.AbstractBuilder.build(AbstractBui
[javadoc]      at
com.sun.tools.doclets.internal.toolkit.builders.ClassBuilder.buildClassDoc(ClassBu
[javadoc]      at sun.reflect.GeneratedMethodAccessor19.invoke(Unknown Source)
[javadoc]      at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:
[javadoc]      at java.lang.reflect.Method.invoke(Method.java:597)
[javadoc]      at
com.sun.tools.doclets.internal.toolkit.builders.ClassBuilder.invokeMethod(ClassBu
[javadoc]      at
com.sun.tools.doclets.internal.toolkit.builders.AbstractBuilder.build(AbstractBui
[javadoc]      at
com.sun.tools.doclets.internal.toolkit.builders.ClassBuilder.build(ClassBuilder.ja
[javadoc]      at
com.sun.tools.doclets.formats.html.HtmlDoclet.generateClassFiles(HtmlDoclet.java:
[javadoc]      at
com.sun.tools.doclets.internal.toolkit.AbstractDoclet.generateClassFiles(Abstract
[javadoc]      at
com.sun.tools.doclets.internal.toolkit.AbstractDoclet.startGeneration(AbstractDoc
[javadoc]      at
com.sun.tools.doclets.internal.toolkit.AbstractDoclet.start(AbstractDoclet.java:64
[javadoc]      at
com.sun.tools.doclets.formats.html.HtmlDoclet.start(HtmlDoclet.java:42)
[javadoc]      at
com.sun.tools.doclets.standard.Standard.start(Standard.java:23)
[javadoc]      at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
[javadoc]      at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
[javadoc]      at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:
[javadoc]      at java.lang.reflect.Method.invoke(Method.java:597)
[javadoc]      at
com.sun.tools.javadoc.DocletInvoker.invoke(DocletInvoker.java:215)
[javadoc]      at
com.sun.tools.javadoc.DocletInvoker.start(DocletInvoker.java:91)
[javadoc]      at com.sun.tools.javadoc.Start.parseAndExecute(Start.java:340)
[javadoc]      at com.sun.tools.javadoc.Start.begin(Start.java:128)
[javadoc]      at com.sun.tools.javadoc.Main.execute(Main.java:41)
[javadoc]      at com.sun.tools.javadoc.Main.main(Main.java:31)

```

java

ant

javadoc

classcastexception

Share

Improve this question

Follow

edited Oct 20, 2010 at 23:24



[martin clayton](#)

78k ● 33 ● 218 ● 200

asked Sep 4, 2008 at 21:37



[ScArcher2](#)

87.2k ● 47 ● 122 ● 160

Mixup in what version of the JDK is used? – [Johan Lübcke](#) Sep 4, 2008 at 21:48

4 Answers

Sorted by: Highest score (default)





54

It looks like this has been reported as a [Java bug](#). It appears to be caused by using annotations from a 3rd party library (like JUnit) and not including the jar with that annotation in the javadoc invocation.



If that is the case, just use the `-classpath` option on javadoc and include the extra jar files.



Share

Improve this answer



Follow

edited Nov 10, 2023 at 19:48



Community Bot

1 • 1

answered Sep 4, 2008 at 21:51



John Meagher

24.6k • 14 • 56 • 57

2 This worked for me. `locate junit` turned up lots, so I picked one at random, put it on the classpath, and that particular exception went away. I still think javadoc is ridiculously flakey, considering how widely it's used. – [Steve Bennett](#) Feb 4, 2011 at 5:30

2 For me, I had to add the Jackson libraries to the javadoc classpath. – [Ethan Mick](#) Jan 25, 2013 at 18:37

I had to add annotations.jar to my classpath, which is in the android-sdk/tools/support directory – [crobicha](#) Jul 16, 2014 at 17:52



1

I have some idea regarding this problem but this not exact solution to get.

If you give single comment line `//` before annotation and try to run the javadoc once again. This problem will solve



Eg: `sample.java` file



```
@ChannelPipeline
```

Makes changes in

```
//@ChannelPipeline
```

Try to run javadoc command once again. Now `ClassCastException` won't occur.

Share

Improve this answer

Follow

edited Jan 27, 2021 at 6:07



M. Justin

20.6k • 9 • 125 • 160

answered Jan 8, 2010 at 19:35



KasthuriRengan

11 • 1



There is another way to get a `ClassCastException` in versions of Java from 5 through 8:

1

java.lang.ClassCastException: com.sun.tools.javadoc.MethodDocImpl cannot be cast to com.sun.tools.javadoc.AnnotationTypeElementDocImpl

It will happen when javadoc encounters a reference to an annotation method in javadoc text before processing the same annotation for the first time used in code. Take these two classes:

```
/**
 * {@link javax.annotation.Generated#value()}
 */
public class TestClass1 {}

@Generated("sometext")
public class TestClass2 {}
```

The bug is order dependent. If javadoc processes `TestClass1` first, the `ClassCastException` will be thrown. If javadoc processes `TestClass2` first, it will complete fine. The bug is reported as [JDK-8170444](#), and was resolved as "Won't Fix". The bug is no longer present in Java 9.

As a workaround, don't link to annotation methods in your documentation text.

Share

Improve this answer

Follow

edited Jan 27, 2021 at 6:13



M. Justin

20.6k ● 9 ● 125 ● 160

answered Nov 25, 2016 at 11:01



Eduard Wirch

9,902 ● 9 ● 63 ● 75

I ran into exactly this issue myself in some code that needs to be compiled against Java 8. I get this error when running Javadoc with Java 8, but not with Java 9. – M. Justin Oct 27, 2020 at 20:24 ✎

I got this problem too. I can compile properly without any errors or warnings. But when I generating javadoc , I got below error.

0

***[javadoc] java.lang.ClassCastException:
com.sun.tools.javadoc.ClassDocImpl cannot be cast to
com.sun.javadoc.AnnotationTypeDoc***

Here is my classpath loading for my third-party-libs ...

```
<path id="build.classpath">
<fileset dir=". ">
```

```
<include name="libs/*.jar" />
</fileset>
```

At my java compile target ..

```
<target name="compile" depends="clean, makedir">
<javac includeantruntime="false" srcdir="${src.dir}" destdir="${build.dir}"
classpathref="build.classpath">
  <compilerarg value="-Xlint:unchecked"/>
</javac>
```

And at my javadoc target...

```
<target name="docs" depends="compile">
<javadoc packagenames="src" sourcepath="${src.dir}" destdir="${docs.dir}"
  failonerror="no"
  author="true"
  version="true"
  windowtitle="${Name} API"
  doctitle="${Name}"
  bottom="Copyright © 2014 ColayHills.com . All Rights Reserved.">
  <fileset dir="${src.dir}">
    <include
name="main/java/com/colayhills/jpcenter/business/service/**" />
  </fileset>
</javadoc>
<echo message="java docs has been generated!"/>
</target>
```

So , I added `classpathref="build.classpath"` option to `<javadoc>` tag. Now It is fine for me.

Share Improve this answer Follow

answered May 19, 2014 at 11:49



Cataclysm

8,488 ● 21 ● 77 ● 127