Javadoc

From Wikipedia, the free encyclopedia

Javadoc is a documentation generator from Sun Microsystems for generating API documentation in HTML format from Java source code.

The "doc comments" format used by Javadoc is the de facto industry standard for documenting Java classes. Some IDEs,[1] such as Netbeans and Eclipse automatically generate Javadoc HTML. Many file editors assist the user in producing Javadoc source and use the Javadoc info as internal references for the programmer.

Javadoc also provides an API for creating doclets and taglets, which allows you to analyze the structure of a Java application. This is how JDiff can generate reports of what changed between two versions of an API.

Contents

Structure of a Javadoc comment

A Javadoc comment is set off from code by standard multi-line comment tags `/*` and `*/`. The opening tag, however, has an extra asterisk, as in `/**`.

1. The first paragraph is a description of the method documented.
2. Following the description are a varying number of descriptive tags, signifying:
   1. The parameters of the method (`@param`)
   2. What the method returns (`@return`)
   3. Any exceptions the method may throw (`@throws`)
   4. Other less-common tags such as `@see` (a "see also" tag)

Overview of Javadoc

The basic structure of writing document comments is embed them inside `/** ... */`. The Javadoc is written next to the items without any separating newline. The class declaration usually contains:

```
/**
 * @author      Firstname Lastname <address @ example.com>
 * @version     1.6                 (the version of the package this class was first
 * @since       2010-03-31          (a date or the version number of this program)
 */
public class Test {
  // class body
}
```

For methods there is (1) a short, concise, one line description to explain what the item does. This is followed by [2] a longer description that may span in multiple paragraphs. In those the details can be explained in full. This section, marked in brackets [], is optional. Last, there is (3) a tag section to list the accepted input arguments and return values of the method.

```
/**
 * Short one line description.                           (1)
 *
 * Longer description. If there were any, it would be    [2]
 * here.
 *
 * @param  variable Description text text text.           (3)
 * @return Description text text text.
 */
```

Variables are documented similarly to methods with the exception that part (3) is omitted. Here the variable contains only the short description:

```
/**
 * Description of the variable here.
 */
private int debug = 0;
```

Some of the available Javadoc tags[2] are listed in the table below:

Tag & Parameter	Usage	Applies to	Since
@author *name*	Describes an author.	Class, Interface, Enum	
@version *version*	Provides software version entry. Max one per Class or Interface.	Class, Interface, Enum	
@since *since-text*	Describes when this functionality has first existed.	Class, Interface, Enum, Field, Method	
@see *reference*	Provides a link to other element of documentation.	Class, Interface, Enum, Field, Method	
@param *name description*	Describes a method parameter.	Method	
@return *description*	Describes the return value.	Method	
@exception *classname description* **@throws** *classname description*	Describes an exception that may be thrown from this method.	Method	
@deprecated *description*	Describes an outdated method.	Method	
{**@inheritDoc**}	Copies the description from the overridden method.	Overriding Method	1.4.0
{**@link** *reference*}	Link to other symbol.	Class, Interface, Enum, Field, Method	
{**@value**}	Return the value of a static field.	Static Field	1.4.0

Example

An example of using Javadoc to document a method follows. Notice that spacing and number of characters in this example are as conventions state.

```
/**
 * Validates a chess move.
 *
 * Use {@link #doMove(int, int, int, int)} to move a piece.
 *
 * @param theFromFile file from which a piece is being moved
 * @param theFromRank rank from which a piece is being moved
 * @param theToFile   file to which a piece is being moved
 * @param theToRank   rank to which a piece is being moved
 * @return            true if the move is valid, otherwise false
 */
boolean isValidMove(int theFromFile, int theFromRank, int theToFile, int theToRank)
{
    ...
}

/**
 * Moves a chess piece.
 *
 * @see java.math.RoundingMode
 */
void doMove(int theFromFile, int theFromRank, int theToFile, int theToRank)
{
    ...
}
```

See also

- Comparison of documentation generators

Notes

1. ^ NetBeans (http://www.netbeans-blog.org/netbeans-ide/generating-javadoc-for-a-project-in-netbeans-ide.html)
2. ^ Javadoc tags in JavaSE 6 (http://java.sun.com/javase/6/docs/technotes/tools/windows/javadoc.html#javadoctags)

External links

- Javadoc tool web site (http://java.sun.com/j2se/javadoc/) HTTP response "404" Not found. Possible 301 : (Moved permanently) new link here (http://www.oracle.com/technetwork/java/javase/documentation/index-jsp-135444.html)
- JavaDoc tags and how to write comments (http://java.sun.com/j2se/javadoc/writingdoccomments/) HTTP response "404" Not found. Possible 301 : (Moved permanently) new link here (http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html)
- JSR 260 (http://www.jcp.org/en/jsr/detail?id=260) Javadoc Tag Technology Update Java Specification Request (defines new Javadoc tags)
- Improve on Javadocs with ashkelon (http://today.java.net/pub/a/today/2004/08/26/ashkelon.html)
- A collection of Javadoc doclets (http://doclet.com)
- MyJavadoc: Another search engine project, to get public Javadocs over Internet, with the possibility to submit new Javadocs (http://www.myjavadoc.net/)
- Globaldocs: A viewer to browse multiple Javadocs simultaneously. (http://globaldocs.info/)
- Various Java documentations converted to Windows Help format (http://www.allimant.org/javadoc/)
- Various Java documentations converted to Windows Help format (http://www.allimant.org/javadoc/)
- JavaDoq (http://www.javadoq.com/) is a handy open source tool to convert Java source code into browsable HTML documents with JavaDoc style and Eclipse Look-And-Feel.

Retrieved from "http://en.wikipedia.org/w/index.php?title=Javadoc&oldid=473248273"

Categories: Free documentation generators | Java development tools | Java specification requests

- This page was last modified on 26 January 2012 at 00:15.