

# What's new in JavaDoc?

New features since JDK 17

Hannes Wallnöfer  
Software Engineer  
Oracle  
February 11, 2025



# Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

# About me

**Hannes Wallnoefer**

Software Engineer, Oracle

# Agenda

- 1 ➤ JavaDoc Overview
- 2 ➤ History and Timeline
- 3 ➤ New Features in Depth
- 4 ➤ Demo

# JavaDoc Team

- Hannes Wallnoefer
- Nizar Benalla
- Jonathan Gibbons (emeritus)
- Pavel Rappo

With contributions from Joe Darcy, Jan Lahoda, Chen Liang, Christian Stein, Alex Buckley, Maurizio Cimadamore...

# JavaDoc Overview

- javadoc command line tool
- jdk.javadoc module
  - Doclet API (exported)
    - Write custom Doclets and Taglets
  - Tool and Standard Doclet implementation (internal)
    - Turn code into HTML5 documentation
  - Service interface to invoke JavaDoc tool programmatically
  - Relies on other JDK/Java SE APIs such as `javax.lang.model.*`

# JavaDoc History

- Part of Java since the beginning
- There used to be Doclets for other formats (e.g. DocBook XML)
- These days it's HTML5 (Standard Doclet)
  - But Doclet API is still there to enable other things

# Anatomy of a JavaDoc Comment

```
/**  
 * Returns a list of items added by {@link #add(Item)}.  
 *  
 * @return a list of items  
 * @since 1.5  
 */  
public List<Item> retrieve();
```

- Main description
- Block tags
- Inline tags



# Timeline

Release	Features	Issue/JEP
JDK 17	Summary Page for New APIs	<a href="#"><u>JDK-8265055</u></a>
JDK 18	Code Snippets	<a href="#"><u>JEP 413</u></a>
JDK 19	Search Enhancements (Multiple Search Terms, Search Page)	<a href="#"><u>JDK-8283640</u></a> , <a href="#"><u>JDK-8291533</u></a>
JDK 20	Summary Page for External Specifications & @spec tag CSS Themes	<a href="#"><u>JDK-8264740</u></a> <a href="#"><u>JDK-8289334</u></a>
JDK 21	JavaDoc Output Specification	<a href="#"><u>JDK-8274665</u></a>

# Timeline (continued)

Release	Features	Issue/JEP
JDK 22	Directed Inheritance with {@inheritDoc}	<a href="#"><u>JDK-8287152</u></a>
JDK 23	Markdown (JEP 467), Navigation Improvements (JDK-8320458)	<a href="#"><u>JEP 467</u></a> <a href="#"><u>JDK-8325167</u></a>
JDK 24+	UI Refinements, DocLint Improvements, Enhanced public API...	

# API Summary Pages

- Deprecated API
  - Controlled via `@java.lang.Deprecated` or `@deprecated` JavaDoc tag
- New API
  - Controlled via `@since` JavaDoc tag
- External Specifications
  - Controlled via `@spec` tag
- Preview API, Restricted Methods
  - JDK/Java SE specific features

# Summary Page for Deprecated API

OVERVIEW TREE PREVIEW NEW **DEPRECATED** INDEX SEARCH HELP

Java SE 23 & JDK 23

Search

Deprecated API

Show API deprecated in: ☐ 12 ☐ 13 ☐ 14 ☐ 15 ☐ 16 ☐ 17 ☒ 18 ☒ 19 ☒ 20 ☒ 21 ☐ 23 ☐ other ☐ all

Contents

Terminally Deprecated

Exception Classes

Fields

Methods

Constructors

Terminally Deprecated Elements

Element ↕	Deprecated in ↕	Description
<code>java.lang.Enum.finalize()</code>	18	Finalization has been deprecated for removal. See <code>Object.finalize()</code> for background information and details about migration options.



# Summary Page for New API

OVERVIEW

TREE

PREVIEW

NEW

DEPRECATED

INDEX

SEARCH

HELP

Java SE 23 & JDK 23

Search

New API since JDK 11

Show API added in: 

12

13

14

15

16

17

18

19

20

21

22

23

all

Contents

Packages

Interfaces

Classes

Enum Classes

Exception Classes

Fields

Methods

Constructors

Enum Constants

New Packages

Package	Added in	Description
java.lang.classfile <sup>PREVIEW</sup>	22	Provides classfile parsing, generation, and transformation library.
java.lang.classfile.attribute <sup>PREVIEW</sup>	22	Provides interfaces describing classfile attributes for the java.lang.classfile <sup>PREVIEW</sup> library.



# Summary Page for Preview API

OVERVIEW

TREE

PREVIEW

NEW

DEPRECATED

INDEX

SEARCH

HELP

Java SE 23 & JDK 23

Search

Preview API

Show preview API for:

☐

466: ClassFile API (Second Preview)

☐

473: Stream Gatherers (Second Preview)

☐

476: Module Import Declarations (Preview)☒☐☐☐

Contents

Classes

Methods

Classes

Class	Preview Feature	Description
java.io.IO	Implicitly Declared Classes and Instance Main Methods	A collection of static convenience methods that provide access to system console for implicitly declared classes.



# Options for API Summary Pages

- **Deprecated API**
  - Enabled by default
  - Disable with `-nodeprecated` option
- **New API**
  - Enable with `-since` and `-since-label` options:

```
javadoc -since 1.0,1.1,1.2,1.3 -since-label "New API since Release 1.0"
```

- Use `-nosince` option to ignore all `@since` tags

# Snippets

- Convenient way to add example code without using HTML

```
{@snippet :  
    public static void main(String... args) {  
        System.out.println("Hello, World!");  
    }  
}
```

```
public static void main(String... args) {  
    System.out.println("Hello, World!");  
}
```





# Snippet Features: Markup

- Highlight, Link, Replace

```
{@snippet lang=java :  
    b.flip();    // @link substring=flip target=#flip  
    b.limit(42); // @highlight substring=42  
}
```

```
b.flip();  
b.limit(42);
```



# Snippet Features: Inline, External and Hybrid Snippets

- Store snippets in `snippet-files` directory
- Refer by filename and optional region:

```
{@snippet class=ExternalSnippet }  
{@snippet class=ExternalSnippet region=print }
```

- Separate main and snippet code
- Make snippet code testable

# Snippet Features: Metadata

- Snippets know about their content type
  - You can tell them using the lang attribute
  - Default is lang=java
- Just add syntax highlighting JS and CSS

```
// Syntax highlighting FTW!  
public void sayHello(String name) {  
    System.out.println("Hello " + name);  
}
```



# Search Improvements

- Multiple search terms

“compute fun”



`Map.computeIfAbsent(K, Function)`

- CamelCase search

“HM cont”



`HashMap.containsKey(Object)`

# Search Features

- Tags to create searchable items
  - `{@index}`
  - `{@systemProperty}`
  - `{@spec}`
- HTML Headers in doc comments are added to the search index

# Search Page

- Static search page allows linking to search results
- Can be used as Bookmarklet

## Search

[Reset](#) [► Additional resources](#)

Found 173 results

**Classes and Interfaces (33)**

**Members (71)**

**Packages (3)**

**Modules (2)**

**Search Tags (64)**

### Module

### Package

java.net.http

java.net.**http**

jdk.httpserver

com.sun.net.**httpserver**

jdk.httpserver

com.sun.net.**httpserver**.spi

# New Navigation Features

- Table of Contents
- Breadcrumb navigation

OVERVIEW **CLASS** USE TREE PREVIEW NEW DEPRECATED INDEX SEARCH HELP

Java SE 24 & JDK 24  
DRAFT 24-ea+35-3640

java.base > java.util > **TreeMap**

Search

Contents Filter

**Description**

- Nested Class Summary
- Constructor Summary
- Method Summary
- Constructor Details
  - TreeMap()
  - TreeMap(Comparator)
  - TreeMap(Map)
  - TreeMap(SortedMap)
- Method Details
  - size()
  - containsKey(Object)

**Class TreeMap<K,V>**

java.lang.Object  
java.util.AbstractMap<K,V>  
java.util.TreeMap<K,V>

**Type Parameters:**

- K - the type of keys maintained by this map
- V - the type of mapped values

**All Implemented Interfaces:**

Serializable, Cloneable, Map<K,V>, NavigableMap<K,V>, SequencedMap<K,V>, SortedMap<K,V>

---

public class **TreeMap<K,V>**  
extends AbstractMap<K,V>  
implements NavigableMap<K,V>, Cloneable, Serializable

A Red-Black tree based NavigableMap implementation. The map is sorted according to the natural ordering of its keys, or by a Comparator provided at map creation time, depending on which constructor is used.

# Markdown Documentation Comments

- Markdown is a popular format to author HTML
- JavaDoc uses the Java implementation of **CommonMark**
- Use `///` end-of-line comment to create a Markdown doc comment

```
/// Prints a short welcome message.  
public void sayHello() {...}
```



# Markdown Doc Comments: JavaDoc integration

- Use [] Markdown links to link to API elements

`[MyClass#someMethod()]`

`[Link with custom label][#otherMethod()]`

- Markdown headings are added to the search index

`# Searchable Heading`

`## Subheading`

# Markdown Doc Comments: Javadoc integration

- Use Javadoc tags just like in traditional doc comments

```
/// {@inheritDoc}  
/// In addition, this methods calls [#wait()].  
///  
/// @param i the index  
public void m(int i) ...
```

# Markdown Documentation Comments

- Use `/* */` comments in code examples – no restrictions as in traditional doc comments

```
/// A code example with a traditional doc comment:  
///  
///     /**  
///      * Traditional doc comment  
///      */  
///     public void foo() {...}
```

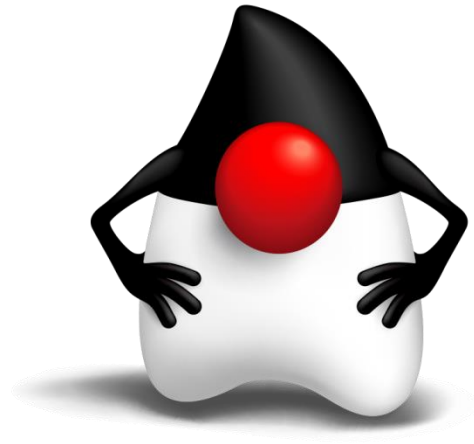
# Markdown Documentation Comments

- Supports Table Extension

```
/// | Latin | Greek |  
/// |-----|-----|  
/// | a      | alpha |  
/// | b      | beta  |  
/// | c      | gamma |
```

# Demo

# Thanks!





ORACLE®