Introducing Java

Module: When Things Fail

What could go wrong?

Figure 1. Perhaps more appropriate for the concurrency module

If something can go wrong, it will go wrong

- Unexpected input
- Configuration bugs
- System resource problems
- Network failures
- File encoding or formatting issues

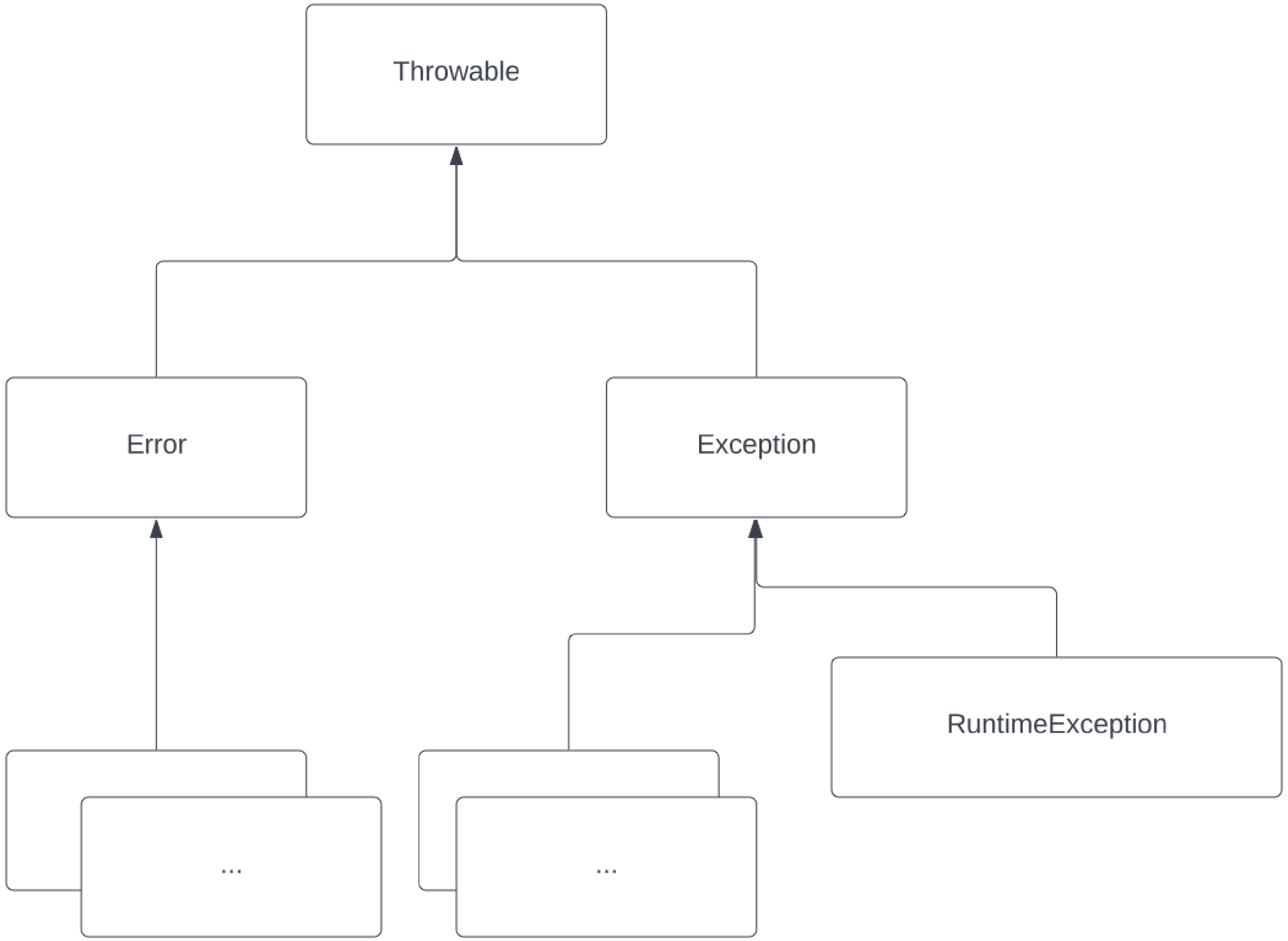
What are exceptions?

- When an *exceptional* condition arises at runtime
- Exceptions are objects that inherit from `Throwable`, and describe the problem that occurred
- They may be generated by the Java runtime or manually created and *thrown* from application code

Why do I have to work with exceptions?

- If an exception is thrown and your code doesn’t handle it correctly, it will reach the default exception handler and your program will terminate
- When abnormal situations occur, you may want to throw exceptions from your code

Throwable


Throwable
Error
Exception
RuntimeException
...
...

Throwable

- All exceptions and errors inherit from `Throwable`
- `Error` represents abnormal, usually system-related problems that we normally don’t have to deal with
- `Exception` objects indicate exceptional conditions that we should be prepared for in application code

Common Exceptions

`NullPointerException`

When we try to perform an operation on a null value, e.g. a method call

`ArrayIndexOutOfBoundsException`

When we try to access an element in an array using an index that exceeds the array’s length

`NumberFormatException`

Often seen when attempting to parse a String as a number

`IllegalArgumentException`

If the input to a method is invalid or out of accepted bounds

You will encounter many more built-in exceptions

Working with exceptions

- Five keywords: `try`, `catch`, `finally`, `throw`, and `throws`

- Exceptions are *thrown* and can be *caught*
- Statements that may throw an exception can be wrapped in a *try block*, and the exception can then be handled in a *catch block*

Try/Catch/Finally

```
try {
  // statement that may throw an exception
} catch (SomeException e) {
  // code that executes if a SomeException is thrown
} finally {
  // code that executes whether an exception was thrown or not
}
```

Try/Catch/Finally

- Try-blocks can be nested, and multiple catch blocks are allowed (be careful with the order)

```
String userInput = "fourty-two";
try {
  int x = Integer.parseInt(userInput);
  int y = Integer.parseInt("0");
  try {
    return x / y;
  } catch (ArithmeticException e) {
    log.error("Did you try to divide by zero?");
  }
} catch (NumberFormatException e) {
  log.error("Unable to parse an integer from '{}'", userInput);
} catch (Exception e) {
  log.error("Something else went wrong");
}
```

Try/Catch/Finally

- You can specify multiple exception types in one catch block

```
String userInput = "fourty-two";
try {
  int x = Integer.parseInt(userInput);
  int y = Integer.parseInt("0");
  return x / y;
} catch (NumberFormatException | ArithmeticException e) {
  log.error("Something went wrong");
}
```

Checked and unchecked exceptions

- Exceptions that inherit from `RuntimeException` are **unchecked**
- Other exceptions are **checked**
- Checked exceptions *must* be caught or declared in the signature of the method from which they are thrown

```
public InputStream openFile(String location) throws FileNotFoundException {
  return new FileInputStream(new File(location));
}
```

Throwing exceptions

- The `throw` keyword is used to throw a new exception

```
public void setPrice(int price) {
  if (price <= 0) {
    throw new IllegalArgumentException("Price must be greater than zero");
  }
  this.price = price;
}
```

Custom exceptions

- You can define your own exceptions by subclassing `Exception`, `RuntimeException`, or any existing exception

```
public class CoffeeTemperatureTooLowException extends RuntimeException {
  public CoffeeTemperatureTooLowException(String message, Throwable cause) {
    super(message, cause);
  }
}
```

Error messages

- When an exception is thrown and not handled, the default handler will print the exception message and stack trace
- The message and stack trace should help you identify where the exception was thrown

```
public class ExceptionHandling {
  public static void main(String[] args) {
    String text = null;
    System.out.println(text.length());
  }
}
```

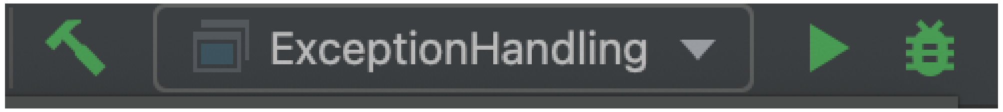
```
Exception in thread "main" java.lang.NullPointerException: Cannot invoke "String.length()" because "text" is
null
    at ExceptionHandling.main(ExceptionHandling.java:4)
```

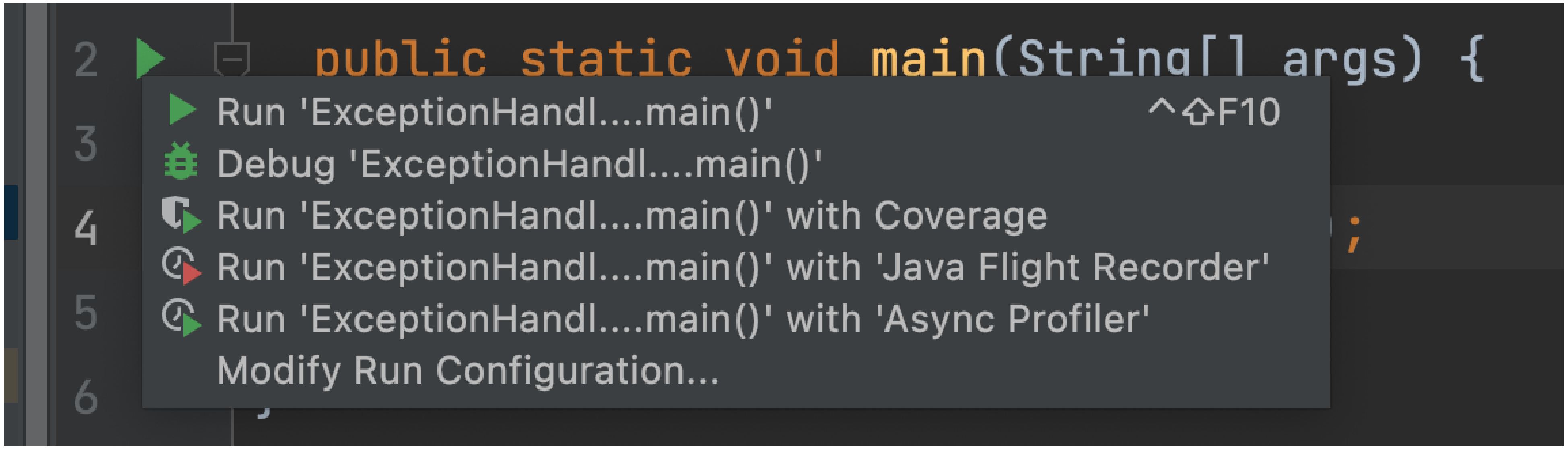
Using a debugger

- Powerful way to investigate problems and validate behaviour
- Pause execution at runtime with breakpoints, inspect object values, and step through code
- Works in the command line with `jdb`, but much easier to use your IDEs tools

Running code in debug mode

- You can run your program or tests in debug mode by clicking on the 'bug' icon in the toolbar, or selecting the option from the 'play' button beside the main method





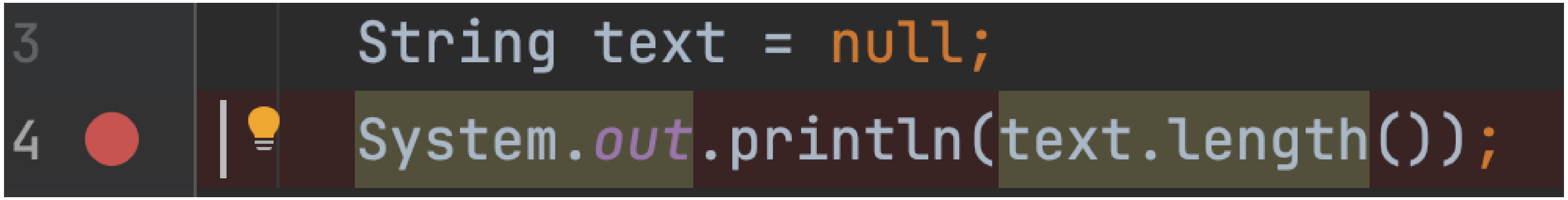
```
public static void main(String[] args) {
                                        ;
}
```

Run 'ExceptionHandl....main()' ^⇧F10
Debug 'ExceptionHandl....main()'
Run 'ExceptionHandl....main()' with Coverage
Run 'ExceptionHandl....main()' with 'Java Flight Recorder'
Run 'ExceptionHandl....main()' with 'Async Profiler'
Modify Run Configuration...

- If no breakpoints are enabled, the program will execute the same way as when you run it normally

Setting breakpoints

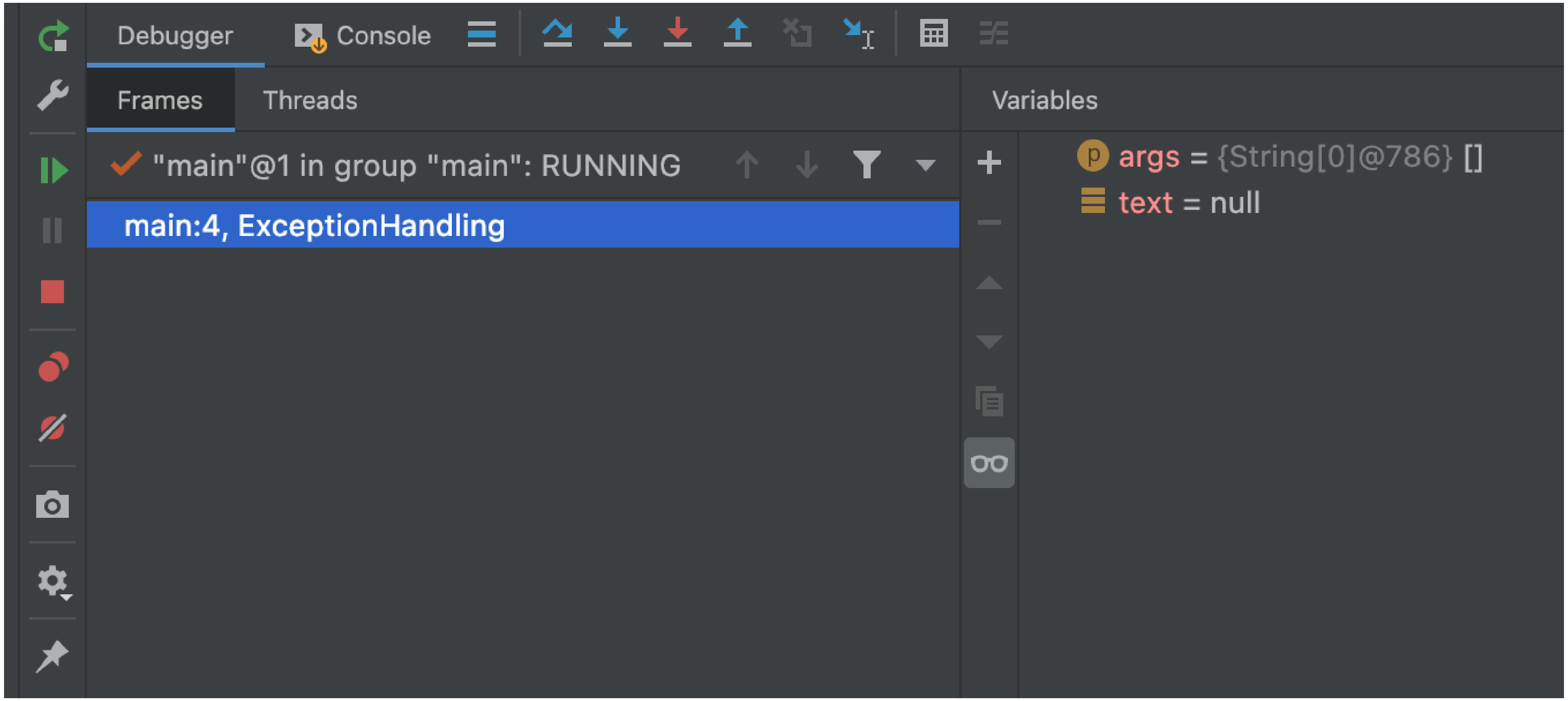
- When a breakpoint is set on a line of code, by default execution will be suspended when that line is reached
- To set a breakpoint, click on the gutter on the left side of the code editor. When it’s set you’ll see a circle



```
String text = null;
System.out.println(text.length());
```

Setting breakpoints

- When a breakpoint is hit, the debug window will open



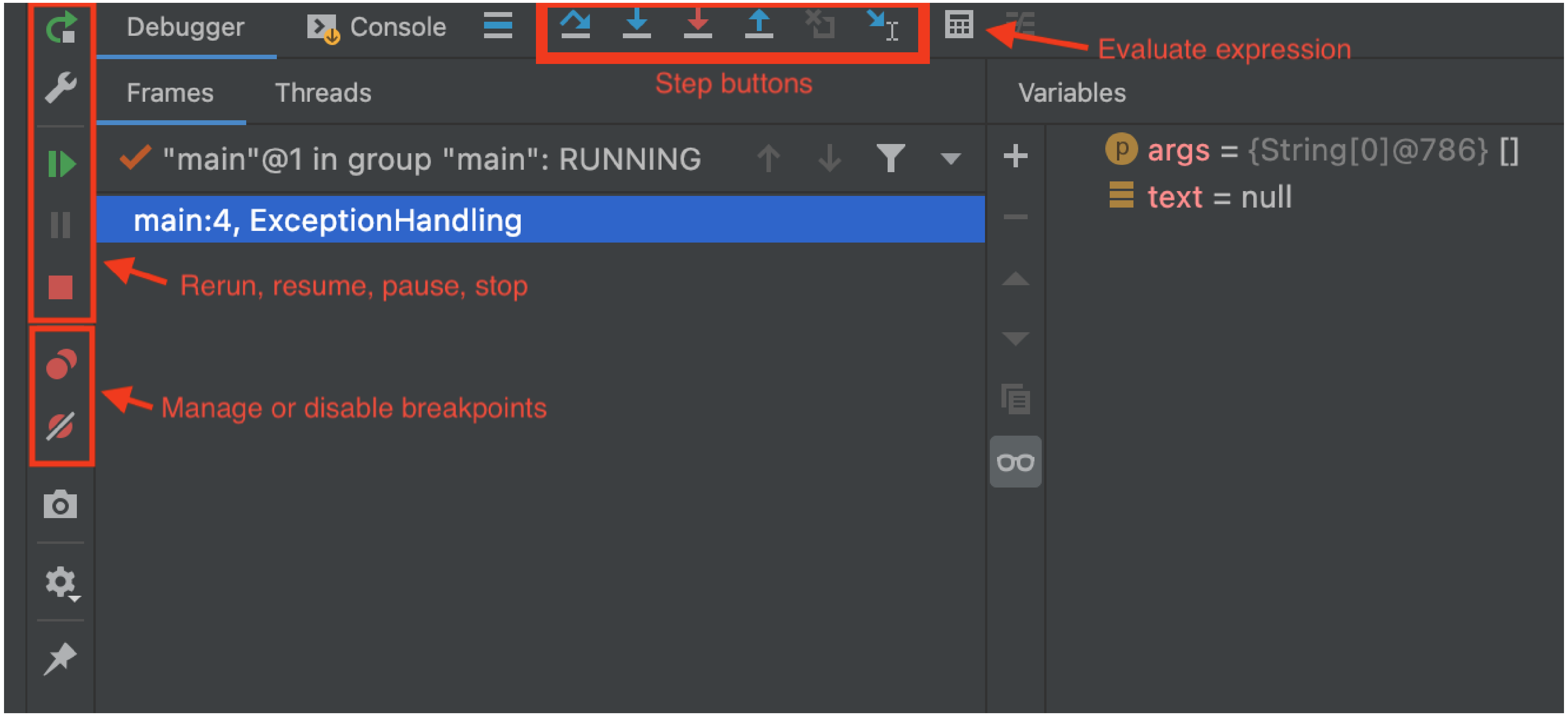
```
"main"@1 in group "main": RUNNING
main:4, ExceptionHandling
```

```
args = {String[0]@786} []
text = null
```

Inspecting the suspended program

- The variables window will show you the values of any variables in scope, such as instance variables, argument values, and local variables

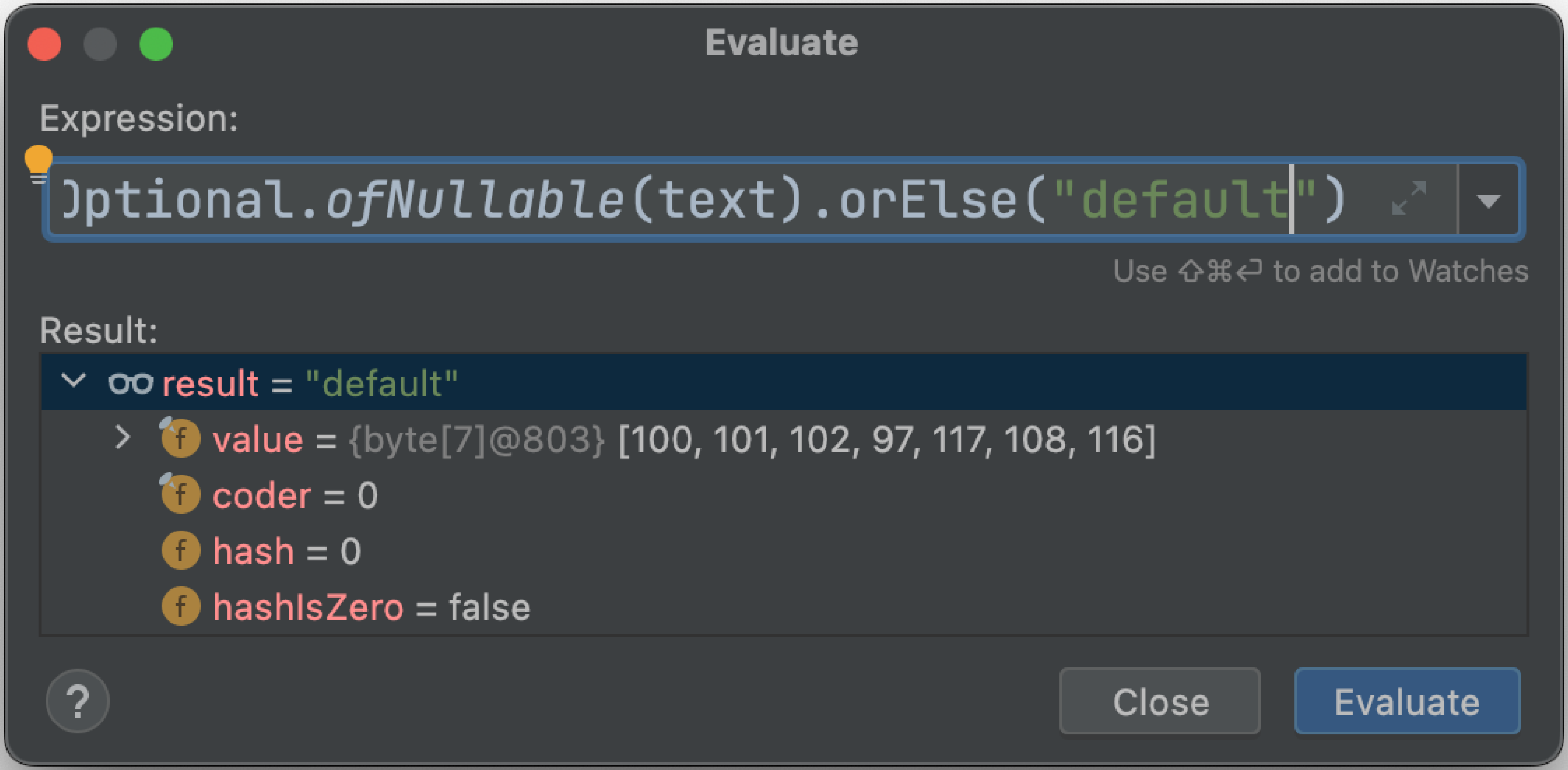
- The Frames panel allows you to look through the stack frames that led to this point, and navigate between them


Debugger
Console
Step buttons
Evaluate expression
Frames
Threads
Variables
"main"@1 in group "main": RUNNING
main:4, ExceptionHandling
Rerun, resume, pause, stop
Manage or disable breakpoints
args = {String[0]@786} []
text = null

- On the top and side of the debug window are various controls for controlling the suspended code

Evaluate expression

- A useful feature in the debug window is the 'Evaluate expression' dialog, giving you a REPL-like interface for testing expressions when a program is suspended



```
)ptional.ofNullable(text).orElse("default")
```

```
result = "default"
  value = {byte[7]@803} [100, 101, 102, 97, 117, 108, 116]
  coder = 0
  hash = 0
  hashIsZero = false
```

Exercises

- Open a text file from the resources directory, catch and handle the exceptions that can occur when opening and reading a file. If successful, print the first line from the file, otherwise print an error message
- Run your code in debug mode and set breakpoints. Experiment with the step controls (step in, step over etc.), and get familiar with the variables panel

Module: Help and Disucussions

Module: Help and Discussions

- Where to find docs
- How to use JavaDoc
- Looking things up
- Asking a question online
- Exercises

Introduction

- What to do when you're stuck?
- Ask your colleague … AGAIN!?
- Or … RTFM

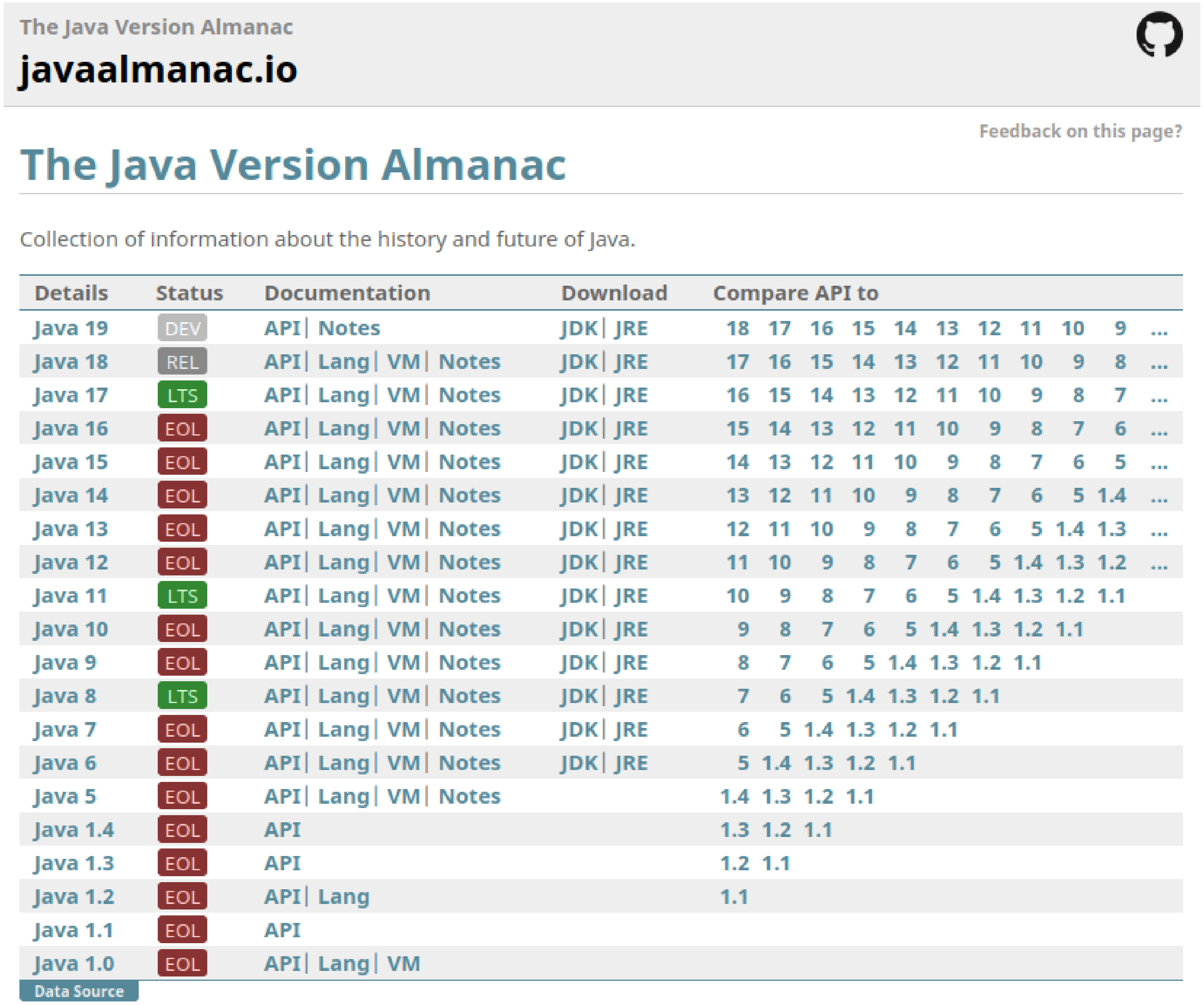
Where to find docs

- Search engine
- Java Version Almanac
- Javadoc.io

Search engine

- Will find anything
- May find too much
- May find *almost* what you need, but not quite

Java Version Almanac



The Java Version Almanac

javaalmanac.io

Feedback on this page?

The Java Version Almanac

Collection of information about the history and future of Java.

Details	Status	Documentation	Download	Compare API to
Java 19	DEV	API | Notes	JDK | JRE	18 17 16 15 14 13 12 11 10 9 ...
Java 18	REL	API | Lang | VM | Notes	JDK | JRE	17 16 15 14 13 12 11 10 9 8 ...
Java 17	LTS	API | Lang | VM | Notes	JDK | JRE	16 15 14 13 12 11 10 9 8 7 ...
Java 16	EOL	API | Lang | VM | Notes	JDK | JRE	15 14 13 12 11 10 9 8 7 6 ...
Java 15	EOL	API | Lang | VM | Notes	JDK | JRE	14 13 12 11 10 9 8 7 6 5 ...
Java 14	EOL	API | Lang | VM | Notes	JDK | JRE	13 12 11 10 9 8 7 6 5 1.4 ...
Java 13	EOL	API | Lang | VM | Notes	JDK | JRE	12 11 10 9 8 7 6 5 1.4 1.3 ...
Java 12	EOL	API | Lang | VM | Notes	JDK | JRE	11 10 9 8 7 6 5 1.4 1.3 1.2 ...
Java 11	LTS	API | Lang | VM | Notes	JDK | JRE	10 9 8 7 6 5 1.4 1.3 1.2 1.1
Java 10	EOL	API | Lang | VM | Notes	JDK | JRE	9 8 7 6 5 1.4 1.3 1.2 1.1
Java 9	EOL	API | Lang | VM | Notes	JDK | JRE	8 7 6 5 1.4 1.3 1.2 1.1
Java 8	LTS	API | Lang | VM | Notes	JDK | JRE	7 6 5 1.4 1.3 1.2 1.1
Java 7	EOL	API | Lang | VM | Notes	JDK | JRE	6 5 1.4 1.3 1.2 1.1
Java 6	EOL	API | Lang | VM | Notes	JDK | JRE	5 1.4 1.3 1.2 1.1
Java 5	EOL	API | Lang | VM | Notes		1.4 1.3 1.2 1.1
Java 1.4	EOL	API		1.3 1.2 1.1
Java 1.3	EOL	API		1.2 1.1
Java 1.2	EOL	API | Lang		1.1
Java 1.1	EOL	API		
Java 1.0	EOL	API | Lang | VM		

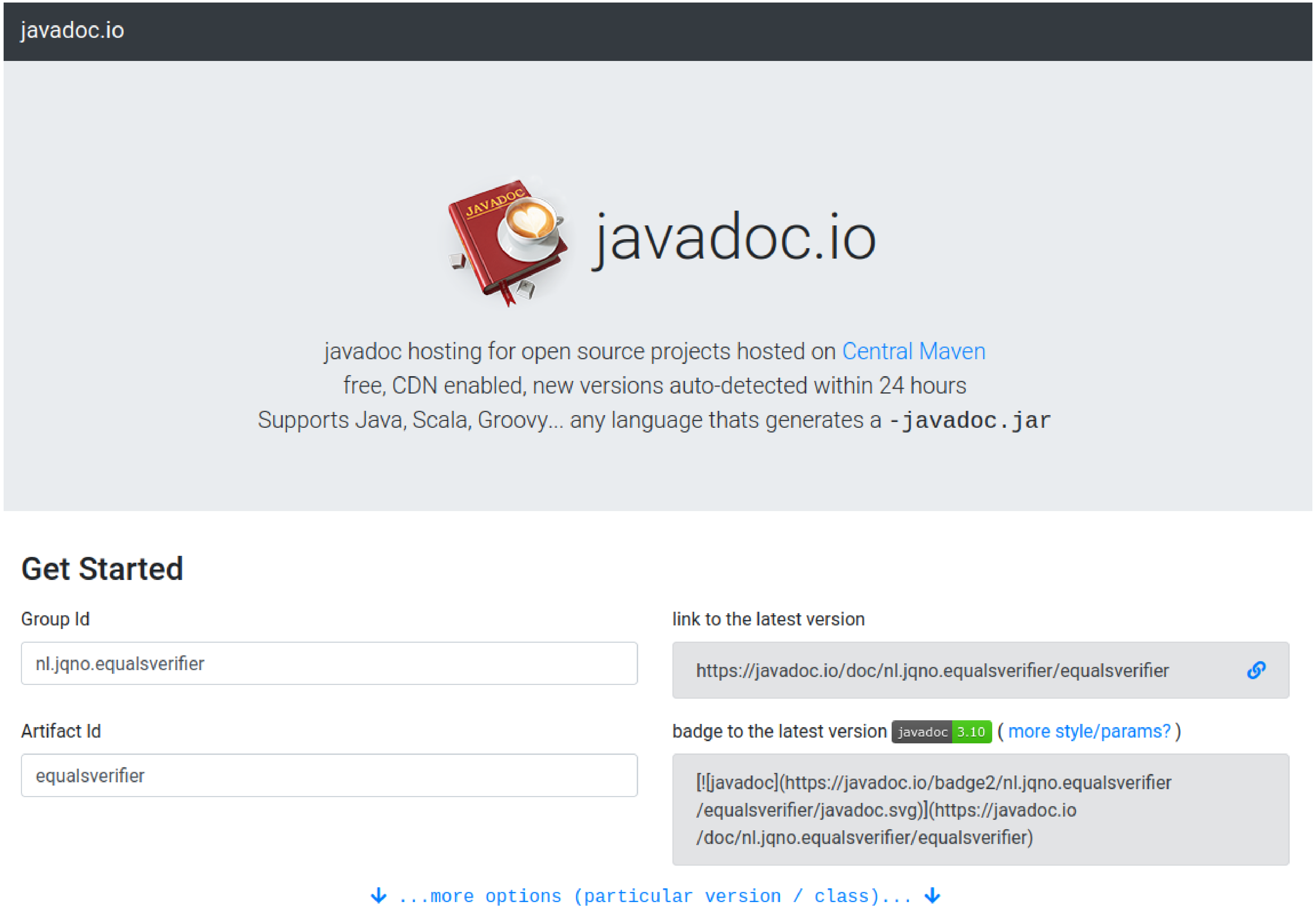
Data Source

https://javaalmanac.io

Java Version Almanac

- Explore!
- Most important: "API"

Javadoc.io



javadoc.io

javadoc.io

javadoc hosting for open source projects hosted on Central Maven
free, CDN enabled, new versions auto-detected within 24 hours
Supports Java, Scala, Groovy... any language thats generates a `-javadoc.jar`

Get Started

Group Id	link to the latest version
nl.jqno.equalsverifier	https://javadoc.io/doc/nl.jqno.equalsverifier/equalsverifier
Artifact Id	badge to the latest version javadoc 3.10 (more style/params?)
equalsverifier	[![javadoc](https://javadoc.io/badge2/nl.jqno.equalsverifier/equalsverifier/javadoc.svg)](https://javadoc.io/doc/nl.jqno.equalsverifier/equalsverifier)

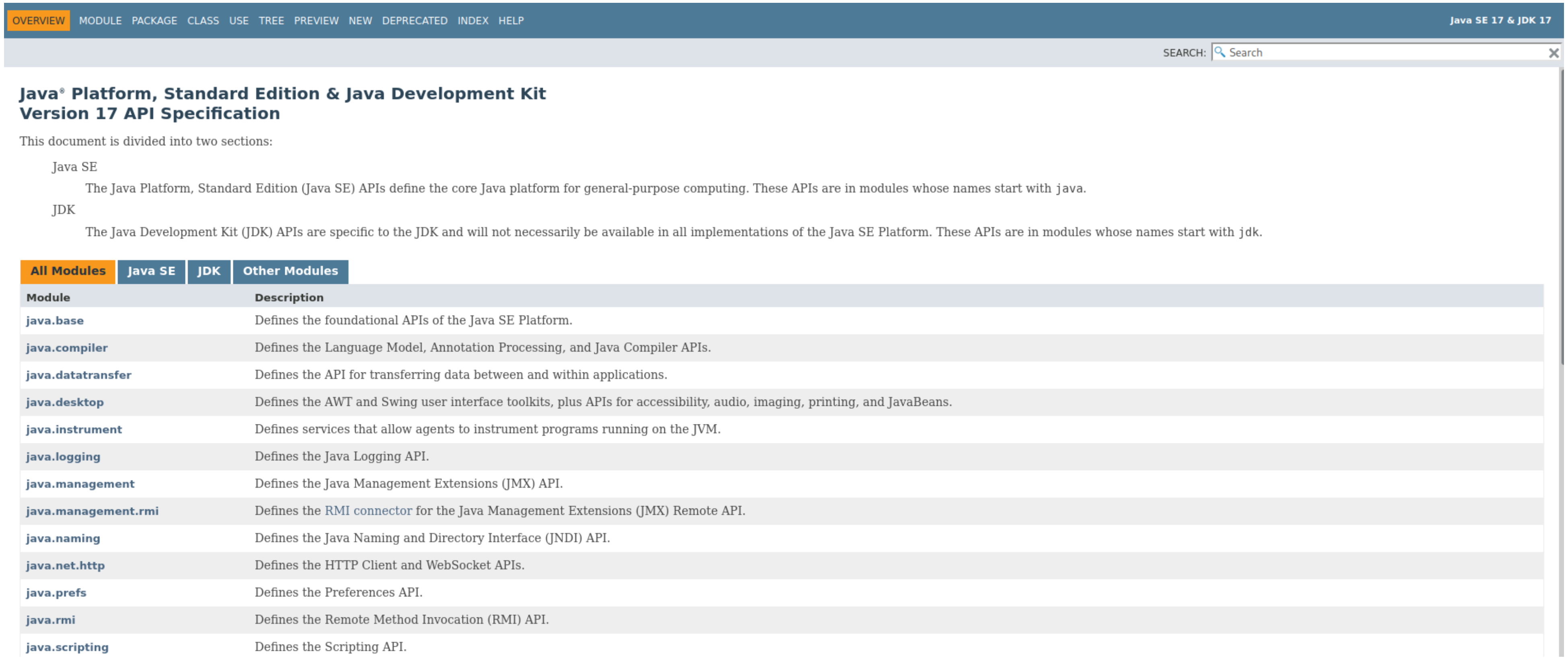
...more options (particular version / class)...

http://javadoc.io

Javadoc.io

- You have to know groupId and artifactId
- But it will autocomplete
- Click the "link" icon on the right
- Specify a version under "more options"

How to use JavaDoc


OVERVIEW MODULE PACKAGE CLASS USE TREE PREVIEW NEW DEPRECATED INDEX HELP
Java SE 17 & JDK 17
SEARCH: Search

Java® Platform, Standard Edition & Java Development Kit Version 17 API Specification

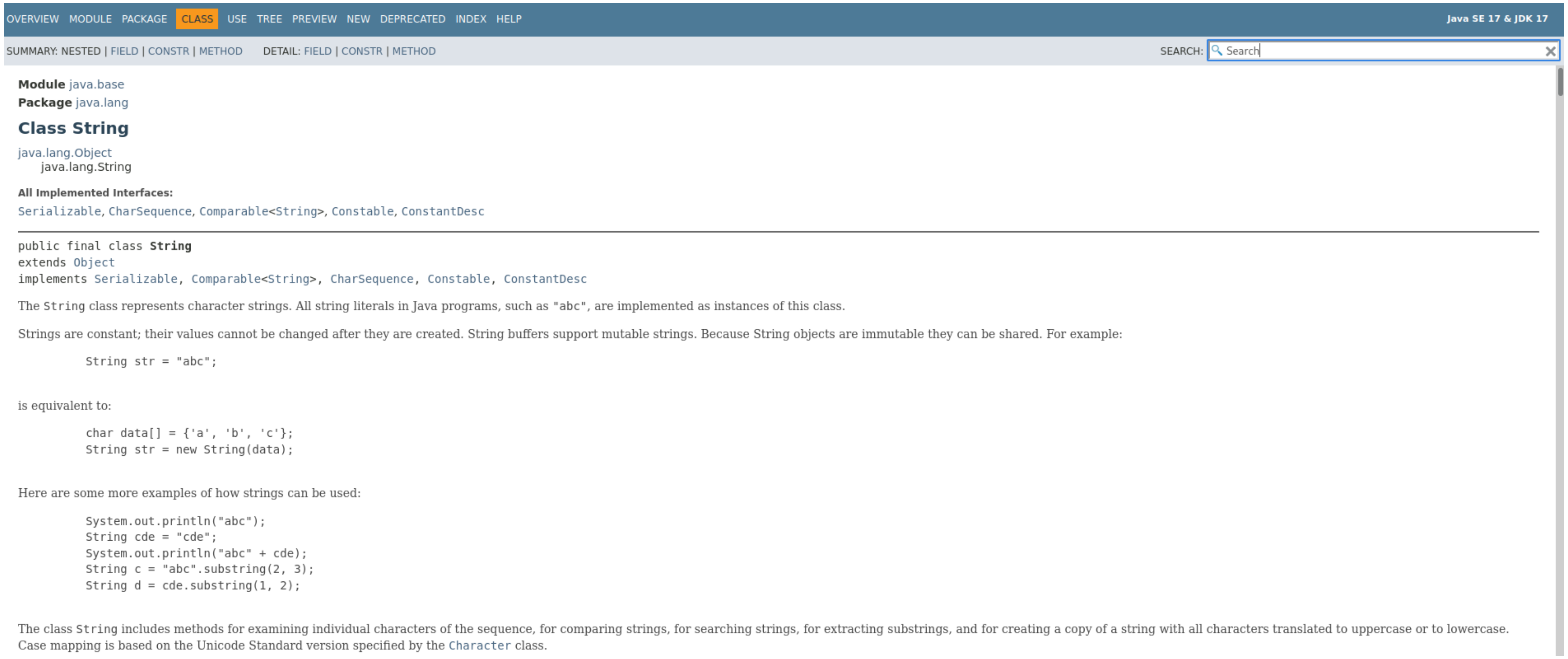
This document is divided into two sections:

Java SE
The Java Platform, Standard Edition (Java SE) APIs define the core Java platform for general-purpose computing. These APIs are in modules whose names start with `java`.

JDK
The Java Development Kit (JDK) APIs are specific to the JDK and will not necessarily be available in all implementations of the Java SE Platform. These APIs are in modules whose names start with `jdk`.

Module	Description
java.base	Defines the foundational APIs of the Java SE Platform.
java.compiler	Defines the Language Model, Annotation Processing, and Java Compiler APIs.
java.datatransfer	Defines the API for transferring data between and within applications.
java.desktop	Defines the AWT and Swing user interface toolkits, plus APIs for accessibility, audio, imaging, printing, and JavaBeans.
java.instrument	Defines services that allow agents to instrument programs running on the JVM.
java.logging	Defines the Java Logging API.
java.management	Defines the Java Management Extensions (JMX) API.
java.management.rmi	Defines the RMI connector for the Java Management Extensions (JMX) Remote API.
java.naming	Defines the Java Naming and Directory Interface (JNDI) API.
java.net.http	Defines the HTTP Client and WebSocket APIs.
java.prefs	Defines the Preferences API.
java.rmi	Defines the Remote Method Invocation (RMI) API.
java.scripting	Defines the Scripting API.

How to use JavaDoc


OVERVIEW MODULE PACKAGE CLASS USE TREE PREVIEW NEW DEPRECATED INDEX HELP
Java SE 17 & JDK 17
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD
SEARCH: Search

Module java.base
Package java.lang

Class String

java.lang.Object
    java.lang.String

All Implemented Interfaces:

`Serializable`, `CharSequence`, `Comparable<String>`, `Constable`, `ConstantDesc`

```
public final class String
extends Object
implements Serializable, Comparable<String>, CharSequence, Constable, ConstantDesc
```

The `String` class represents character strings. All string literals in Java programs, such as `"abc"`, are implemented as instances of this class.

Strings are constant; their values cannot be changed after they are created. String buffers support mutable strings. Because String objects are immutable they can be shared. For example:

```
         String str = "abc";
```

is equivalent to:

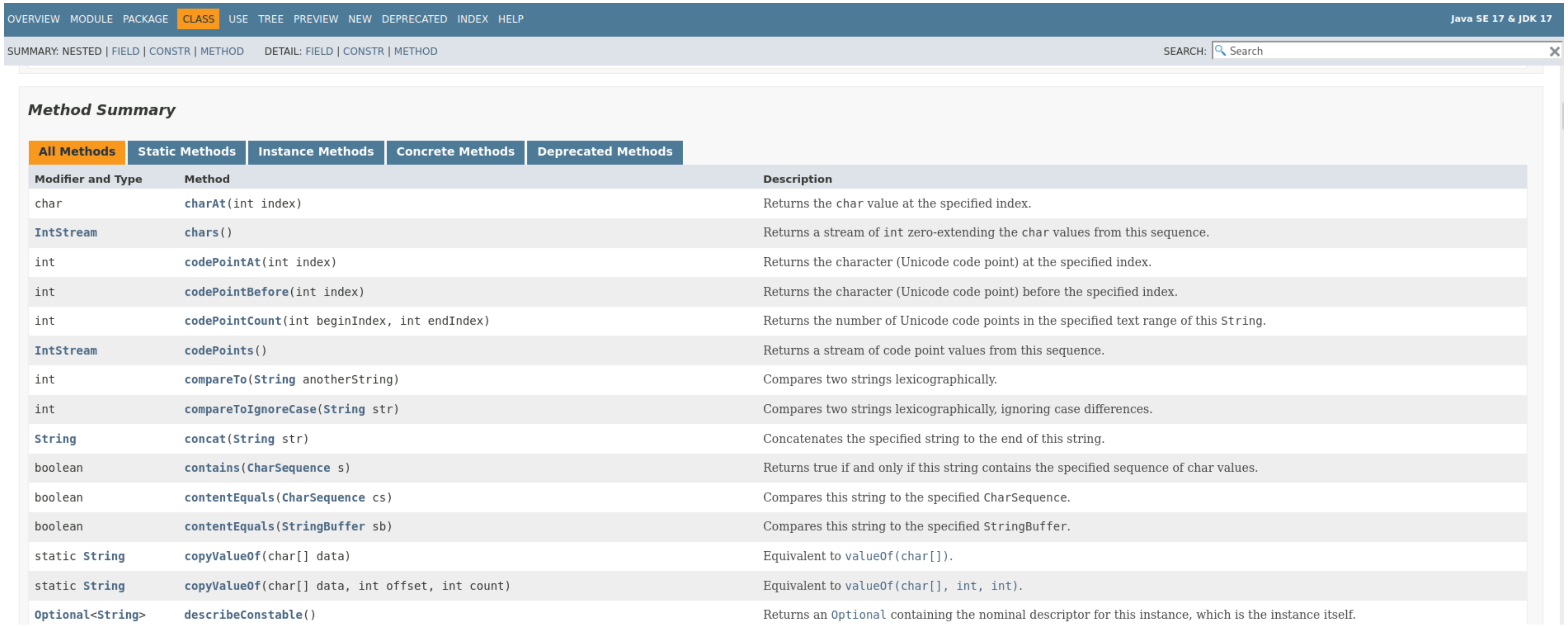
```
         char data[] = {'a', 'b', 'c'};
         String str = new String(data);
```

Here are some more examples of how strings can be used:

```
         System.out.println("abc");
         String cde = "cde";
         System.out.println("abc" + cde);
         String c = "abc".substring(2, 3);
         String d = cde.substring(1, 2);
```

The class `String` includes methods for examining individual characters of the sequence, for comparing strings, for searching strings, for extracting substrings, and for creating a copy of a string with all characters translated to uppercase or to lowercase. Case mapping is based on the Unicode Standard version specified by the `Character` class.

How to use JavaDoc


OVERVIEW MODULE PACKAGE CLASS USE TREE PREVIEW NEW DEPRECATED INDEX HELP

Java SE 17 & JDK 17

SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

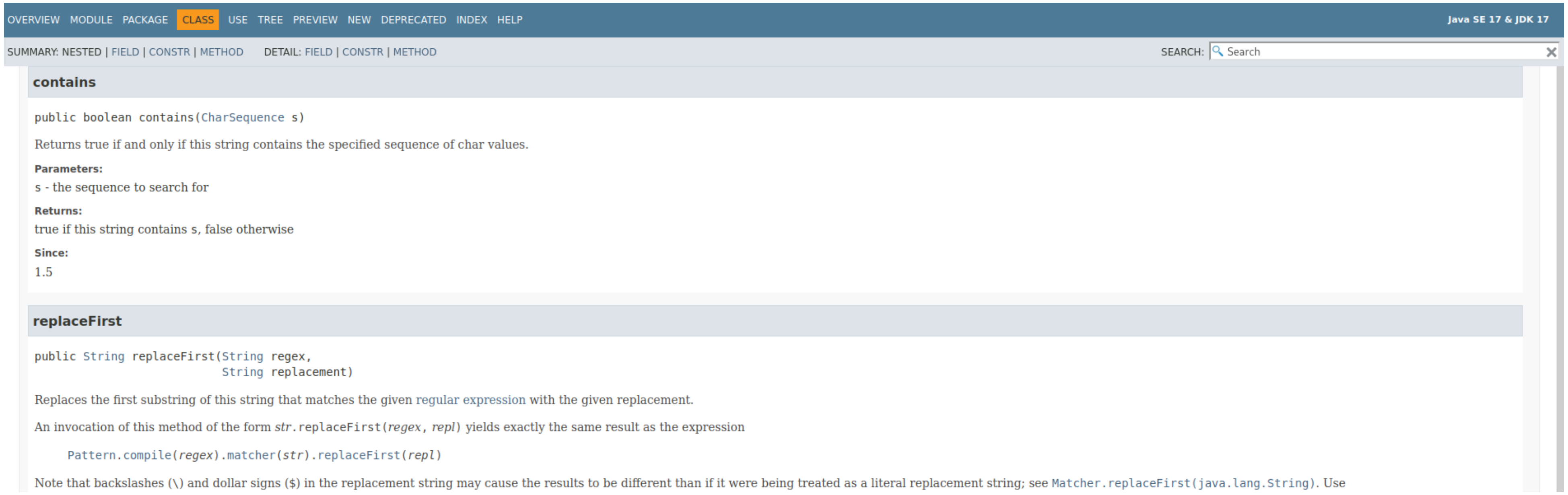
SEARCH: Search

Method Summary

All Methods | Static Methods | Instance Methods | Concrete Methods | Deprecated Methods

Modifier and Type	Method	Description
`char`	`charAt(int index)`	Returns the char value at the specified index.
`IntStream`	`chars()`	Returns a stream of int zero-extending the char values from this sequence.
`int`	`codePointAt(int index)`	Returns the character (Unicode code point) at the specified index.
`int`	`codePointBefore(int index)`	Returns the character (Unicode code point) before the specified index.
`int`	`codePointCount(int beginIndex, int endIndex)`	Returns the number of Unicode code points in the specified text range of this String.
`IntStream`	`codePoints()`	Returns a stream of code point values from this sequence.
`int`	`compareTo(String anotherString)`	Compares two strings lexicographically.
`int`	`compareToIgnoreCase(String str)`	Compares two strings lexicographically, ignoring case differences.
`String`	`concat(String str)`	Concatenates the specified string to the end of this string.
`boolean`	`contains(CharSequence s)`	Returns true if and only if this string contains the specified sequence of char values.
`boolean`	`contentEquals(CharSequence cs)`	Compares this string to the specified CharSequence.
`boolean`	`contentEquals(StringBuffer sb)`	Compares this string to the specified StringBuffer.
`static String`	`copyValueOf(char[] data)`	Equivalent to valueOf(char[]).
`static String`	`copyValueOf(char[] data, int offset, int count)`	Equivalent to valueOf(char[], int, int).
`Optional<String>`	`describeConstable()`	Returns an Optional containing the nominal descriptor for this instance, which is the instance itself.

How to use JavaDoc


OVERVIEW MODULE PACKAGE CLASS USE TREE PREVIEW NEW DEPRECATED INDEX HELP
Java SE 17 & JDK 17
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD
SEARCH: Search

contains

```
public boolean contains(CharSequence s)
```

Returns true if and only if this string contains the specified sequence of char values.

Parameters:
s - the sequence to search for

Returns:
true if this string contains s, false otherwise

Since:
1.5

replaceFirst

```
public String replaceFirst(String regex,
                           String replacement)
```

Replaces the first substring of this string that matches the given regular expression with the given replacement.

An invocation of this method of the form *str*.replaceFirst(*regex*, *repl*) yields exactly the same result as the expression

```
Pattern.compile(regex).matcher(str).replaceFirst(repl)
```

Note that backslashes (\) and dollar signs ($) in the replacement string may cause the results to be different than if it were being treated as a literal replacement string; see Matcher.replaceFirst(java.lang.String). Use

How to use JavaDoc

- Everything is cross-referenced
- Works the same way for Java APIs and thrid party projects

JavaDoc from source

```
/**
  * Returns true if and only if this string contains the specified
  * sequence of char values.
  *
  * @param s the sequence to search for
  * @return true if this string contains {@code s}, false otherwise
  * @since 1.5
```

```
   */
public boolean contains(CharSequence s) { ... }
```

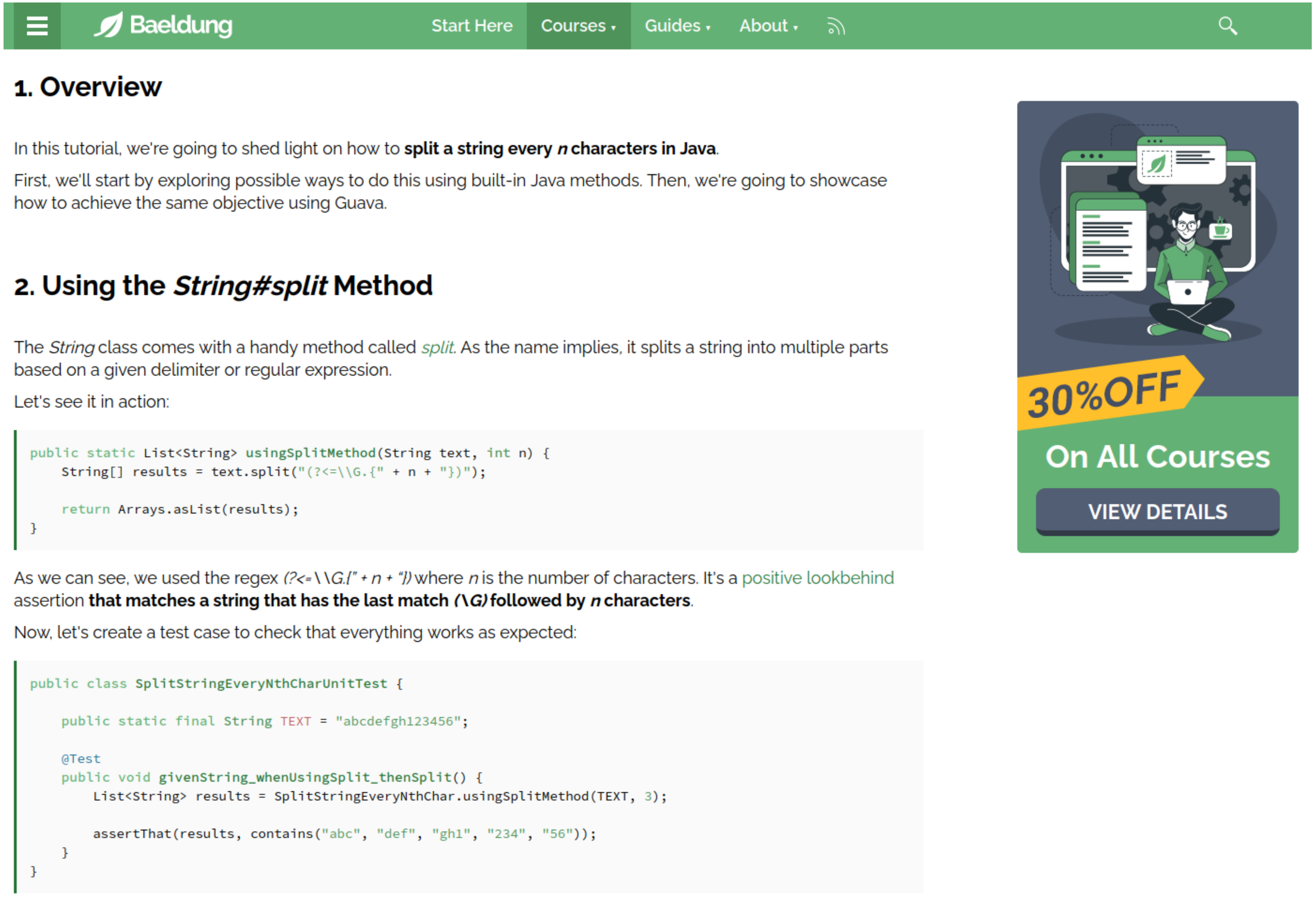
JavaDoc from source

- @param
- @throws
- @return
- @since
- @code

Looking things up

- Baeldung
- StackOverflow
- GitHub
- Search engines

Baeldung



1. Overview

In this tutorial, we're going to shed light on how to **split a string every *n* characters in Java**.

First, we'll start by exploring possible ways to do this using built-in Java methods. Then, we're going to showcase how to achieve the same objective using Guava.

2. Using the *String#split* Method

The *String* class comes with a handy method called *split*. As the name implies, it splits a string into multiple parts based on a given delimiter or regular expression.

Let's see it in action:

```
public static List<String> usingSplitMethod(String text, int n) {
    String[] results = text.split("(?<=\\G.{" + n + "})");

    return Arrays.asList(results);
}
```

As we can see, we used the regex *(?<=\\G.{" + n + "})* where *n* is the number of characters. It's a positive lookbehind assertion **that matches a string that has the last match *(\G)* followed by *n* characters**.

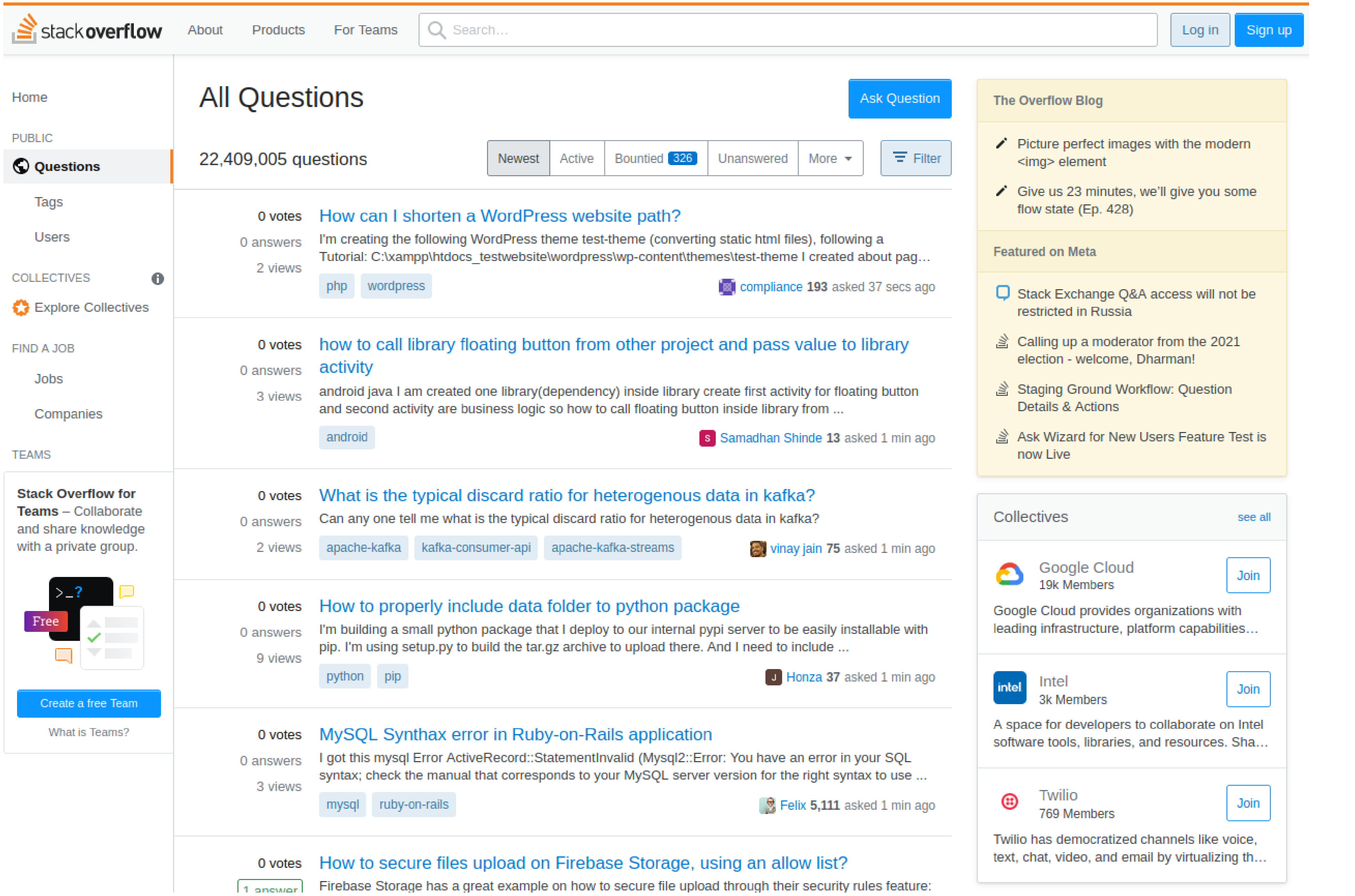
Now, let's create a test case to check that everything works as expected:

```
public class SplitStringEveryNthCharUnitTest {

    public static final String TEXT = "abcdefgh123456";

    @Test
    public void givenString_whenUsingSplit_thenSplit() {
        List<String> results = SplitStringEveryNthChar.usingSplitMethod(TEXT, 3);

        assertThat(results, contains("abc", "def", "gh1", "234", "56"));
    }
}
```

Baeldung

- Tutorials
- Specific tasks

StackOverflow


stack overflow · About · Products · For Teams · Search… · Log in · Sign up

Home

PUBLIC

Questions

Tags

Users

COLLECTIVES

Explore Collectives

FIND A JOB

Jobs

Companies

TEAMS

Stack Overflow for Teams – Collaborate and share knowledge with a private group.

Create a free Team

What is Teams?

All Questions

Ask Question

22,409,005 questions

Newest · Active · Bountied 326 · Unanswered · More · Filter

0 votes · 0 answers · 2 views

How can I shorten a WordPress website path?

I'm creating the following WordPress theme test-theme (converting static html files), following a Tutorial: C:\xampp\htdocs_testwebsite\wordpress\wp-content\themes\test-theme I created about pag…

php · wordpress — compliance 193 asked 37 secs ago

0 votes · 0 answers · 3 views

how to call library floating button from other project and pass value to library activity

android java I am created one library(dependency) inside library create first activity for floating button and second activity are business logic so how to call floating button inside library from ...

android — Samadhan Shinde 13 asked 1 min ago

0 votes · 0 answers · 2 views

What is the typical discard ratio for heterogenous data in kafka?

Can any one tell me what is the typical discard ratio for heterogenous data in kafka?

apache-kafka · kafka-consumer-api · apache-kafka-streams — vinay jain 75 asked 1 min ago

0 votes · 0 answers · 9 views

How to properly include data folder to python package

I'm building a small python package that I deploy to our internal pypi server to be easily installable with pip. I'm using setup.py to build the tar.gz archive to upload there. And I need to include ...

python · pip — Honza 37 asked 1 min ago

0 votes · 0 answers · 3 views

MySQL Synthax error in Ruby-on-Rails application

I got this mysql Error ActiveRecord::StatementInvalid (Mysql2::Error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use ...

mysql · ruby-on-rails — Felix 5,111 asked 1 min ago

0 votes · 1 answer

How to secure files upload on Firebase Storage, using an allow list?

Firebase Storage has a great example on how to secure file upload through their security rules feature:

The Overflow Blog

- Picture perfect images with the modern <img> element
- Give us 23 minutes, we'll give you some flow state (Ep. 428)

Featured on Meta

- Stack Exchange Q&A access will not be restricted in Russia
- Calling up a moderator from the 2021 election - welcome, Dharman!
- Staging Ground Workflow: Question Details & Actions
- Ask Wizard for New Users Feature Test is now Live

Collectives · see all

Google Cloud · 19k Members · Join

Google Cloud provides organizations with leading infrastructure, platform capabilities…

Intel · 3k Members · Join

A space for developers to collaborate on Intel software tools, libraries, and resources. Sha…

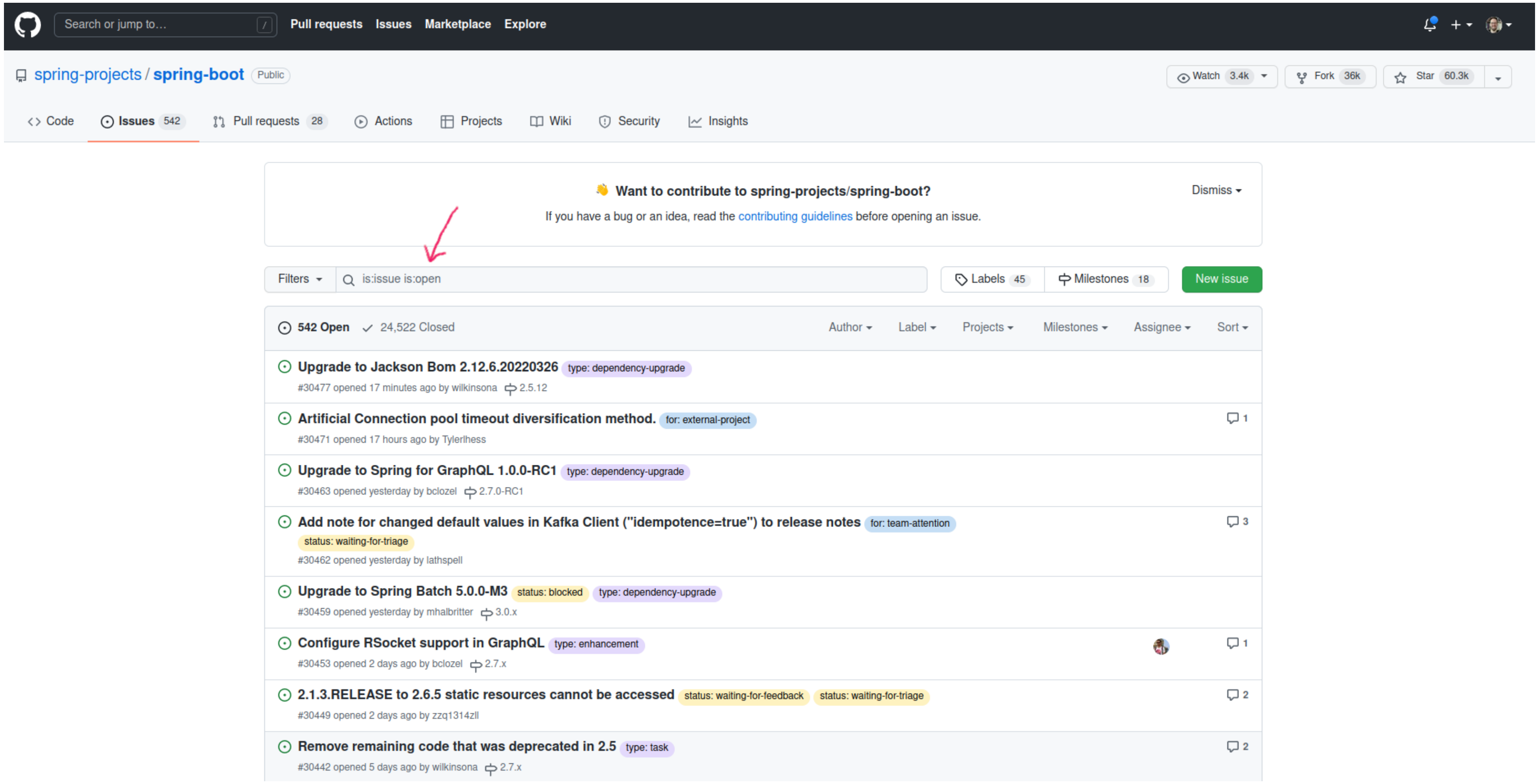
Twilio · 769 Members · Join

Twilio has democratized channels like voice, text, chat, video, and email by virtualizing th…

StackOverflow

- Q&A
- Understand before you copy-paste

GitHub Issues


Search or jump to... Pull requests Issues Marketplace Explore

spring-projects / spring-boot Public

Watch 3.4k Fork 36k Star 60.3k

Code Issues 542 Pull requests 28 Actions Projects Wiki Security Insights

Want to contribute to spring-projects/spring-boot? Dismiss

If you have a bug or an idea, read the contributing guidelines before opening an issue.

Filters is:issue is:open Labels 45 Milestones 18 New issue

542 Open ✓ 24,522 Closed Author Label Projects Milestones Assignee Sort

Upgrade to Jackson Bom 2.12.6.20220326 type: dependency-upgrade
#30477 opened 17 minutes ago by wilkinsona 2.5.12

Artificial Connection pool timeout diversification method. for: external-project 1
#30471 opened 17 hours ago by Tylerlhess

Upgrade to Spring for GraphQL 1.0.0-RC1 type: dependency-upgrade
#30463 opened yesterday by bclozel 2.7.0-RC1

Add note for changed default values in Kafka Client ("idempotence=true") to release notes for: team-attention status: waiting-for-triage 3
#30462 opened yesterday by lathspell

Upgrade to Spring Batch 5.0.0-M3 status: blocked type: dependency-upgrade
#30459 opened yesterday by mhalbritter 3.0.x

Configure RSocket support in GraphQL type: enhancement 1
#30453 opened 2 days ago by bclozel 2.7.x

2.1.3.RELEASE to 2.6.5 static resources cannot be accessed status: waiting-for-feedback status: waiting-for-triage 2
#30449 opened 2 days ago by zzq1314zll

Remove remaining code that was deprecated in 2.5 type: task 2
#30442 opened 5 days ago by wilkinsona 2.7.x

GitHub Issues

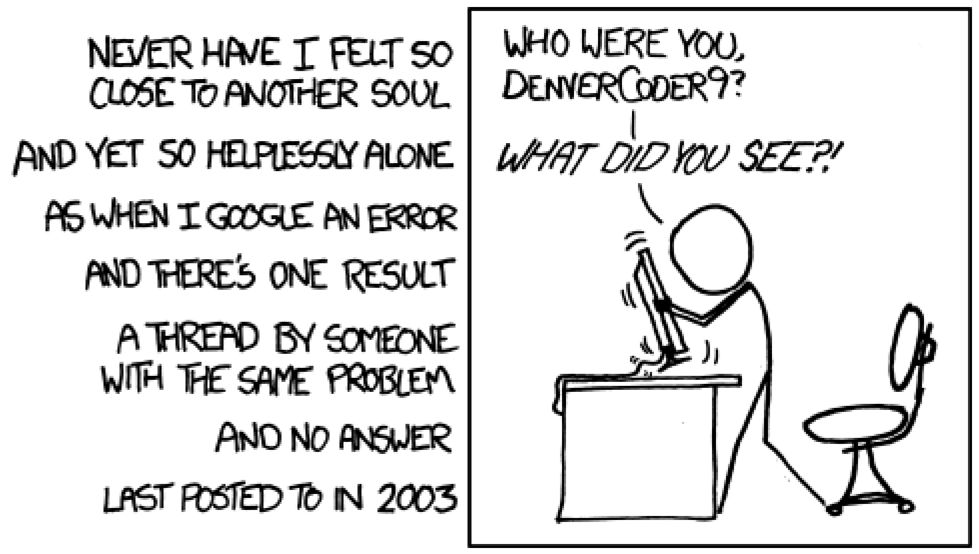
- Reporting problems
- Use the search

- Remove `is:open`, because closed issues are the ones that are solved

Search engine

- Quality of results varies
- You can find things that you can’t find elsewhere

Search Engine


NEVER HAVE I FELT SO CLOSE TO ANOTHER SOUL
AND YET SO HELPLESSLY ALONE
AS WHEN I GOOGLE AN ERROR
AND THERE'S ONE RESULT
A THREAD BY SOMEONE WITH THE SAME PROBLEM
AND NO ANSWER
LAST POSTED TO IN 2003
WHO WERE YOU, DENVERCODER9?
WHAT DID YOU SEE?!

Asking a question online

- On StackOverflow or GitHub Issues
- People answer in their spare time

- Respect their time

How to ask a question

- Research the issue
- Give as much information as possible
- Give a code example
  - As small as possible
  - Still shows the issue
- Answer your own question if possible!

The internet can be harsh

- StackOverflow questions can get closed
- People can be rude
- People may be bad at English
- People have bad days

Exercises

1. Use Baeldung to find how to sum the content of an array.
2. Use StackOverflow to find how to sum the content of an array.

3. Use GitHub to find why Semaphores don’t work in version 3.7 of the EqualsVerifier project.