Contents

[Java Stream API 2](#_Toc193797535)

[1. Convert All Strings to Uppercase 2](#_Toc193797536)

[2. Filter Strings Based on Length 2](#_Toc193797537)

[3. Find the First String that Starts with a Specific Letter 2](#_Toc193797538)

[4. Sort Strings Alphabetically 2](#_Toc193797539)

[5. Count Strings that Contain a Specific Character 3](#_Toc193797540)

[6. Join Strings Using Collectors.joining() 3](#_Toc193797541)

[7. Remove Duplicates Using distinct() 3](#_Toc193797542)

[8. Convert to List Using collect() 3](#_Toc193797543)

[9. Find Longest String Using max() 4](#_Toc193797544)

[10. Convert a String Array to Lowercase and Sort 4](#_Toc193797545)

[11. Convert String to Character Stream and Sort 4](#_Toc193797546)

[12. Check if Any String Contains a Substring Using anyMatch() 4](#_Toc193797547)

[13. Remove Empty or Blank Strings Using filter() 5](#_Toc193797548)

[14. Convert a Sentence to Word List 5](#_Toc193797549)

[15. Find All Words Starting with a Vowel 5](#_Toc193797550)

[16. Count Vowels in a String 5](#_Toc193797551)

[17. Find the Smallest String Using min() 6](#_Toc193797552)

[18. Convert Strings to ASCII Values 6](#_Toc193797553)

[19. Reverse Each Word in a Sentence 6](#_Toc193797554)

[20. Convert String Array to a Set Using collect() 6](#_Toc193797555)

Java Stream API

Java Stream API provides a powerful way to process collections and arrays using functional programming concepts.

Here are some **Java Stream API** examples for processing **Strings**:

## 1. Convert All Strings to Uppercase

String[] words = {"java", "stream", "api", "example"};

Arrays.stream(words)

.map(String::toUpperCase)

.forEach(System.out::println);

**Output:**

JAVA

STREAM

API

EXAMPLE

## 2. Filter Strings Based on Length

Arrays.stream(words)

.filter(word -> word.length() > 4)

.forEach(System.out::println);

**Output:**

stream

example

## 3. Find the First String that Starts with a Specific Letter

Optional<String> result = Arrays.stream(words)

.filter(word -> word.startsWith("e"))

.findFirst();

System.out.println("First word starting with 'e': " + result.orElse("Not Found"));

**Output:**

First word starting with 'e': example

## 4. Sort Strings Alphabetically

Arrays.stream(words)

.sorted()

.forEach(System.out::println);

**Output:**

api

example

java

stream

## 5. Count Strings that Contain a Specific Character

long count = Arrays.stream(words)

.filter(word -> word.contains("a"))

.count();

System.out.println("Count of words containing 'a': " + count);

**Output:**

Count of words containing 'a': 3

## 6. Join Strings Using Collectors.joining()

String sentence = Arrays.stream(words)

.collect(Collectors.joining(" "));

System.out.println("Joined Sentence: " + sentence);

**Output:**

Joined Sentence: java stream api example

## 7. Remove Duplicates Using distinct()

String[] data = {"apple", "banana", "apple", "orange", "banana"};

Arrays.stream(data)

.distinct()

.forEach(System.out::println);

**Output:**

apple

banana

orange

## 8. Convert to List Using collect()

List<String> wordList = Arrays.stream(words)

.collect(Collectors.toList());

System.out.println("List: " + wordList);

**Output:**

List: [java, stream, api, example]

## 9. Find Longest String Using max()

String longest = Arrays.stream(words)

.max(Comparator.comparingInt(String::length))

.orElse("No data");

System.out.println("Longest Word: " + longest);

**Output:**

Longest Word: example

## 10. Convert a String Array to Lowercase and Sort

String[] names = {"Sanjay", "Rahul", "Arjun", "Vikram"};

Arrays.stream(names)

.map(String::toLowerCase)

.sorted()

.forEach(System.out::println);

**Output:**

arjun

rahul

sanjay

vikram

## 11. Convert String to Character Stream and Sort

String str = "streamAPI";

str.chars()

.mapToObj(c -> (char) c)

.sorted()

.forEach(System.out::print);

**Output:**

AAIPemrst

## 12. Check if Any String Contains a Substring Using anyMatch()

String[] words = {"apple", "banana", "cherry", "mango"};

boolean containsBan = Arrays.stream(words)

.anyMatch(word -> word.contains("ban"));

System.out.println("Any word contains 'ban': " + containsBan);

**Output:**

Any word contains 'ban': true

## 13. Remove Empty or Blank Strings Using filter()

String[] data = {"hello", " ", "", "world", "java", " "};

Arrays.stream(data)

.filter(str -> !str.trim().isEmpty())

.forEach(System.out::println);

**Output:**

hello

world

java

## 14. Convert a Sentence to Word List

String sentence = "Learning Java Stream API is fun";

List<String> wordList = Arrays.stream(sentence.split("\\s+"))

.collect(Collectors.toList());

System.out.println(wordList);

**Output:**

[Learning, Java, Stream, API, is, fun]

## 15. Find All Words Starting with a Vowel

Arrays.stream(words)

.filter(word -> word.matches("(?i)^[aeiou].\*"))

.forEach(System.out::println);

**Output:**

apple

## 16. Count Vowels in a String

String str = "Stream API Example";

long vowelCount = str.chars()

.mapToObj(c -> (char) c)

.filter(c -> "AEIOUaeiou".indexOf(c) != -1)

.count();

System.out.println("Number of vowels: " + vowelCount);

**Output:**

Number of vowels: 7

## 17. Find the Smallest String Using min()

String smallest = Arrays.stream(words)

.min(Comparator.comparingInt(String::length))

.orElse("No data");

System.out.println("Smallest Word: " + smallest);

**Output:**

Smallest Word: mango

## 18. Convert Strings to ASCII Values

String input = "Hello";

input.chars()

.forEach(c -> System.out.println(c + " -> " + (char) c));

**Output:**

72 -> H

101 -> e

108 -> l

108 -> l

111 -> o

## 19. Reverse Each Word in a Sentence

String sentence = "Java Stream API";

String reversedSentence = Arrays.stream(sentence.split("\\s+"))

.map(word -> new StringBuilder(word).reverse().toString())

.collect(Collectors.joining(" "));

System.out.println(reversedSentence);

**Output:**

avaJ maertS IPA

## 20. Convert String Array to a Set Using collect()

String[] animals = {"dog", "cat", "dog", "elephant", "cat"};

Set<String> uniqueAnimals = Arrays.stream(animals)

.collect(Collectors.toSet());

System.out.println(uniqueAnimals);

**Output:**

[dog, cat, elephant]