

## Java Stream max()



Last Updated: March  
16, 2022



By: Lokesh  
Gupta



Java  
8



Find Max Min, Java Stream Basics, Java  
Stream Methods

The **Stream max()** method is used to select the largest element in the **Stream** according to the [Comparator](#) used for comparing the elements.

The **Comparator** imposes a total ordering on the Stream elements which may not have a natural ordering.

### Table Of Contents

#### [1. Stream max\(\) Method](#)

##### [1.1. Method Syntax](#)

##### [1.2. Description](#)

#### [2. Java Stream max\(\) Example](#)

[Example 1: Largest element in the Stream with Lambda Expression](#)

[Example 2: Largest element in the Stream with Comparator](#)

## 1. Stream max() Method

### 1.1. Method Syntax

- The method takes a **non-interfering and stateless *Comparator*** to compare elements of the stream.

- It returns an [Optional](#) describing the maximum element of the stream, or an empty `Optional` if the stream is empty.
- The `max()` method throws [NullPointerException](#) if the maximum element found is `null`.

## Syntax

```
Optional<T> max(Comparator<? super T> comparator)
```

## 1.2. Description

- The `max()` method is a **terminal operation**. So the Stream cannot be used after this method has been executed.
- It returns the maximum/largest element of this stream according to the provided `Comparator`.
- This is a special case of a **stream reduction**.
- The method argument shall be a non-interfering, stateless `Comparator`.
- The method returns an [Optional](#) describing the maximum element of this stream, or an empty `Optional` if the stream is empty.
- It may throw [NullPointerException](#) if the maximum element is `null`.

## 2. Java Stream max() Example

### Example 1: Largest element in the Stream with Lambda Expression

Java example to find the largest number from a stream of numbers using comparator as [lambda expression](#).

#### Select largest element from stream

```
List<Integer> list = Arrays.asList(2, 4, 1, 3, 7, 5, 9, 6, 8);

Optional<Integer> maxNumber = list.stream()
    .max((i, j) -> i.compareTo(j));

System.out.println(maxNumber.get());
```

Program output.

9

## Example 2: Largest element in the Stream with Comparator

Java example to find the largest number from a stream of numbers using [custom comparator](#).

```
List<Integer> list = Arrays.asList(2, 4, 1, 3, 7, 5, 9, 6, 8);

Comparator<Integer> maxComparator = new Comparator<Integer>() {

    @Override
    public int compare(Integer n1, Integer n2) {
        return n1.compareTo(n2);
    }
};

Optional<Integer> maxNumber = list.stream()
    .max(maxComparator);

System.out.println(maxNumber.get());
```

Program output.

9

Drop me your questions related to **Java 8 Stream max()** method in Java Streams to **find the largest element in the Stream**.

Happy Learning !!

[Sourcecode on Github](#)

## Was this post helpful?

Let us know if you liked the post. That's the only way we can improve.

Yes

No

## Recommended Reading:

1. [Python max\(\) and min\(\) – finding max and min in list or array](#)
2. [Java Stream – Get Object with Max Date From a List](#)
3. [Finding Max and Min from List using Streams](#)
4. [Java Stream reuse – traverse stream multiple times?](#)
5. [Java Regex to check Min/Max Length of Input Text](#)
6. [Finding Max and Min in Arrays](#)
7. [Hibernate count, min, max, sum, avg Functions](#)
8. [Java Stream min\(\)](#)
9. [Java Stream skip\(\)](#)
0. [Java Stream findFirst\(\)](#)

---

## Join 7000+ Awesome Developers

Get the latest updates from industry, awesome resources, blog updates and much more.

**Email Address**

**Subscribe**

*\* We do not spam !!*

---

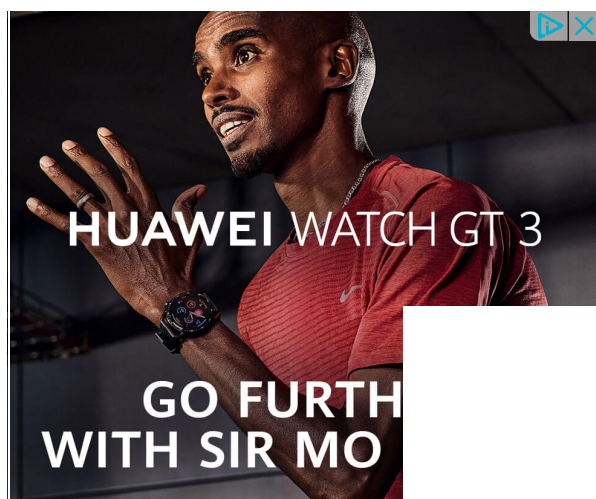
## Leave a Comment

☐ Add me to your newsletter and keep me updated whenever you publish new blog posts

## Post Comment







## HowToDoInJava

A blog about Java and related technologies, the best practices, algorithms, and interview questions.

### Meta Links

- [About Me](#)
- [Contact Us](#)
- [Privacy policy](#)
- [Advertise](#)
- [Guest Posts](#)



## Blogs

REST API Tutorial



Copyright © 2022 · Hosted on [Cloudways](#) · [Sitemap](#)