HowToDoInJava

Java Stream min()

Last Updated: March 29, 2022 | By: Lokesh Gupta | Java 8 | Java Stream Basics, Java Stream Methods

The **Stream min()** method is used to select the minimum/smallest 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 min() Method

1.1. Method Syntax

- The method takes a **non-interfering, 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 `min()` method throws NullPointerException if the minimum element found is `null`.

Method Syntax

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

1.2. Description

- This is a **terminal operation**. So stream cannot be used after this method is executed.
- Returns the minimum/smallest 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 smallest element of this stream, or an empty `Optional` if the stream is empty.
- It may throw NullPointerException if the smallest element is `null`.

2. Stream min() Examples

Example 1: Finding Smallest Element with Lambda Expression

Java example to find the minimum number from a stream of numbers using comparator as lambda expression.

Select smallest element from stream

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

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

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

Program output.

Output

```
1
```

Example 2: Finding Smallest Element with Comparator

Java example to find the minimum 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> minComparator = new Comparator<Integer>() {

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

Optional<Integer> minNumber = list.stream()
            .min(minComparator);

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

Program output.

Output

```
1
```

Drop me your questions related to *Java 8 Stream min()* API in Java Stream API to **find the smallest element in stream**.

Happy Learning !!

Recommended Reading:

1. Python max() and min() – finding max and min in list or array
2. Finding Max and Min from List using Streams
3. Java Stream reuse – traverse stream multiple times?
4. Java Regex to check Min/Max Length of Input Text
5. Finding Max and Min in Arrays
6. Hibernate count, min, max, sum, avg Functions
7. Java Stream count() Matches with filter()
8. Java Stream sorted()
9. Java Stream limit()
0. Java Stream findFirst()

1 thought on “Java Stream min()”

Ashwin Kumar

August 18, 2019 at 11:12 pm

List integerList = Arrays.asList(5,2,7,6,7,8,33,23,64);
Optional minInteger = integerList.stream().min(myCustomComparator);
Consumer action = i -> {
System.out.println("Minimum integer is :::::::: " + i);
};
minInteger.ifPresent(action);//Dont know how this action is working with proper out put.

//////////////////////////////
Hi the above snippet is working fine, but could not decipher how "i" is populated with "minInteger" in the action above.

Reply

Leave a Comment

Name *

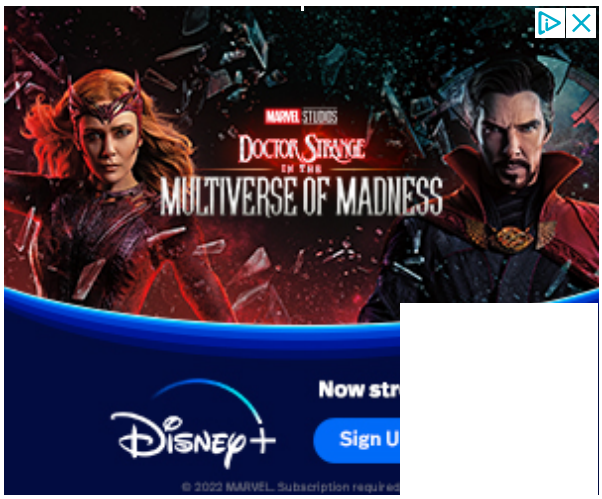
Email *

Website

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

Post Comment

Search …



HowToDoInJava

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

Meta Links

- About Me
- Contact Us
- Privacy policy
- Advertise