

## Java Stream findFirst()

📅 Last Updated: December 26, 2020    👤 By: Lokesh Gupta    📁 Java 8    💎 Java 8, Java Stream Basics, Java Stream Methods

The `findFirst()` method returns an [Optional](#) describing the **first element of the given stream** if [Stream](#) is non-empty, or an empty `Optional` if the stream is empty.

### 1. Stream findFirst() Method

Method syntax

```
Optional<T> findFirst()
```

- The `findAny()` method is a **terminal short-circuiting** operation.
- The `findFirst()` method returns an `Optional`.
- The `Optional` contains the value as first element of the given stream, if Stream is non-empty.
- The `Optional` contains the empty value, if Stream is empty.
- If the element selected is `null`, **`NullPointerException`** is thrown.
- If Stream has **defined encounter order**, the `findFirst()` returns first element in encounter order.
- If Stream has **no encounter order**, the `findFirst()` may return any element.
- The above behavior is valid for all *sequential and parallel streams*. The behavior of `findFirst()` does not change by the [parallelism](#) of the Stream.

### 2. Stream findFirst() Example

In the given example, we are getting the first element from the `Stream`. As soon as, we get the first element, the stream operation moves to `ifPresent()` method.

We print the first element in using the [method reference](#) inside `ifPresent()` method.

Stream.findFirst() API example

```
import java.util.stream.Stream;

public class Main
{
    public static void main(String[] args)
    {
        //sequential stream
```

```
Stream.of("one", "two", "three", "four")
    .findFirst()
    .ifPresent(System.out::println);

//parallel stream

Stream.of("one", "two", "three", "four")
    .parallel()
    .findFirst()
    .ifPresent(System.out::println);
}
```

Program output.

Console

```
one
one
```

### 3. Stream findFirst() vs findAny()

In non-parallel streams, `findFirst()` and `findAny()`, both may return the first element of the Stream in most cases. But `findAny()` does not offer any guarantee of this behavior.

Use `findAny()` to get any element from any parallel stream in faster time. Else we can always use `findFirst()` in most of the cases.

Happy Learning !!

#### 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. [Java Stream findFirst\(\) vs findAny\(\) API With Example](#)
2. [Java Stream reuse – traverse stream multiple times?](#)
3. [Java Stream sorted\(\)](#)
4. [Java Stream toArray\(\)](#)
5. [Java Stream findAny\(\)](#)
6. [Java Stream count\(\) Matches with filter\(\)](#)

7. [Java Stream forEach\(\)](#)
8. [Java Stream max\(\)](#)
9. [Java Stream peek\(\)](#)
10. [Java Stream limit\(\)](#)

## 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**

Name \*

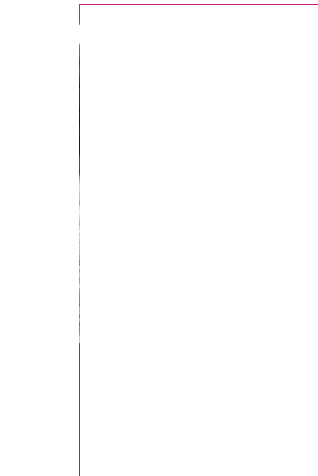
Email \*

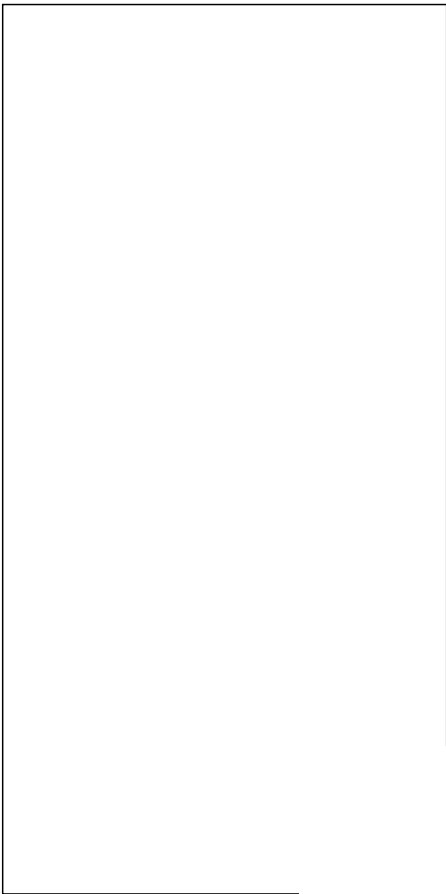
Website

☐ 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](#)
- › [Guest Posts](#)

### Blogs

[REST API Tutorial](#)



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