

Java Stream findAny()

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

The `findAny()` method returns an `Optional` describing the **any element of the given stream** if `Stream` is non-empty, or an empty `Optional` if the stream is empty.

In non-parallel streams, `findAny()` will return the first element in most of the cases but this behavior is not guaranteed.

The `Stream.findAny()` method has been introduced for performance gain in case of parallel streams, only.

1. Stream findAny() Method

Method syntax

```
Optional<T> findAny()
```

- The `findAny()` method is a **terminal short-circuiting** operation.
- The `findAny()` method returns an `Optional`.
- The `Optional` contains the value as any element of the given stream, if `Stream` is non-empty. The returned element is the first element in most of the cases.
- The `Optional` contains the empty value, if `Stream` is empty.
- If the element selected is `null`, `NullPointerException` is thrown.
- For all the *sequential and parallel streams*, it may return any element. The behavior of `findAny()` does not change by the **parallelism** of the `Stream`.
- Similarly, there is no guaranteed behavioral difference in case of a stream has defined encounter order or it has no encounter order at all.

2. Stream findAny() Example

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

We print the received element in using the **method reference** inside `ifPresent()` method.

Stream.findAny() API example

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

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

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

        //parallel stream

        Stream.of("one", "two", "three", "four")
            .parallel()
            .findAny()
            .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](https://howtodoinjava.com/java8/stream-findany/)

2. [Java Stream reuse – traverse stream multiple times?](#)
3. [Java Stream sorted\(\)](#)
4. [Java Stream toArray\(\)](#)
5. [Java Stream findFirst\(\)](#)
6. [Java Stream forEach\(\)](#)
7. [Java Stream max\(\)](#)
8. [Java Stream min\(\)](#)
9. [Java Stream limit\(\)](#)
0. [Java Stream skip\(\)](#)

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](#)