

Java Stream findFirst() vs findAny() API With Example

📅 Last Updated: August 30, 2020 👤 By: Lokesh Gupta 📁 Java 8 💡 Java Stream Basics

Java `Stream` interface has two methods i.e. **findFirst()** and **findAny()**. Both method looks very much similar but they may behave differently in certain conditions. In this post, learn the **difference between findFirst() and findAny()** methods.

1. Stream findFirst() method

1.1. Description

Method syntax

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

This method returns an [Optional](#) describing the **first element of this stream**. In case of stream has :

- **defined encounter order** – first element in encounter order in stream.
- **no encounter order** – any element may be returned.

The above theory is valid for all **sequential and parallel streams** and the behavior of `findFirst()` will not change.

1.2. Stream findFirst() example

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

2. Stream findAny() method

2.1. Description

Method syntax

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

This method returns an `Optional` describing the **any element of this stream**. In case of stream has :

- **defined encounter order** – any element may be returned.
- **no encounter order** – any element may be returned.

The above theory is valid for all **sequential and parallel streams** and the behavior of `findAny()` will not change.

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

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

2.2. Stream findAny() example

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  
three
```

3. Stream findFirst() vs findAny() – Conclusion

In this post, we learned the difference between `findFirst()` and `findAny()` methods in [Java 8 Stream API](#). In non-parallel streams, 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.

Reference :

[Java Stream Interface](#)

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\(\)](#)
2. [Java Stream findAny\(\)](#)
3. [Java Stream reuse – traverse stream multiple times?](#)
4. [Java 9 Stream API Improvements](#)
5. [Java Stream API Tutorials](#)
6. [Java WatchService API Tutorial](#)
7. [Java Stream max\(\)](#)
8. [Java Stream limit\(\)](#)
9. [Collecting Stream Items into Map in Java](#)
0. [Java 8 Stream concat\(\)](#)



career★girls

WHICH CAREER IS
RIGHT FOR YOU

TAKE THE QUIZ

Join 7000+ Awesome Developers

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

Email Address

Subscribe

** We do not spam !!*



2 thoughts on “Java Stream findFirst() vs findAny() API With Example”

Siddarth

March 16, 2019 at 12:23 pm

Can I assume that for findFirst() method, the output will not differ irrespective of whether the stream is parallel stream or sequential stream?

[Reply](#)

Lokesh Gupta

March 16, 2019 at 9:55 pm

Yes. That is the method contract.

[Reply](#)

Leave a Comment

Name *

Email *

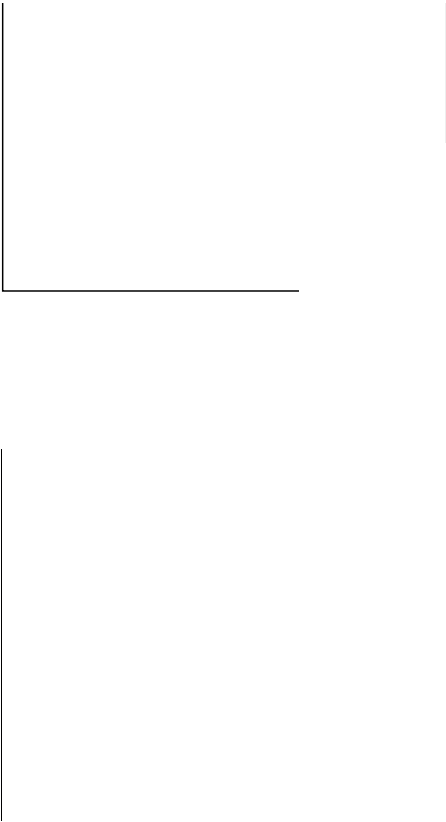
Website

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

Post Comment

Search ...





Promoted by costabrava.org

Sponsored



AY



A message from our sponsor

[Learn more](#)

Promoted by usertesting.com

Sponsored

AY

A message from our sponsor

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](#)