

Java Stream map() vs flatMap()

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

In Java, `Stream` interface has `map()` and `flatMap()` methods and both are intermediate stream operations and return another stream as method output. The primary **difference between map() vs flatMap()** is the return type of both methods.

`map()` is used for transformation only, but `flatMap()` is used for both transformation and flattening.

`flatMap()` = `map()` + Flattening

1. Difference between map() and flatMap()

The `map()` method produces one output value for each input value in the stream. So if there are `n` elements in the stream, `map()` operation will produce a stream of `n` output elements.

`map()` example

```
List<String> listOfStrings = Arrays.asList("1", "2", "3", "4", "5");

List<Integer> listOfIntegers = listOfStrings.stream()
    .map(Integer::valueOf)
    .collect(Collectors.toList());

System.out.println(listOfIntegers);    //[1, 2, 3, 4, 5]
```

`flatMap()` is two step process i.e. **map() + Flattening**. It helps in converting `Collection<Collection<T>>` to `Collection<T>`.

`flatMap()` example

```
List<Integer> list1 = Arrays.asList(1,2,3);
List<Integer> list2 = Arrays.asList(4,5,6);
List<Integer> list3 = Arrays.asList(7,8,9);

List<List<Integer>> listOfLists = Arrays.asList(list1, list2, list3);

List<Integer> listOfAllIntegers = listOfLists.stream()
    .flatMap(x -> x.stream())
    .collect(Collectors.toList());

System.out.println(listOfAllIntegers);    //[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

2. Usage of map() vs flatMap()

We can use `map()` operation when we have a stream of objects, and we need to get some unique value for each element in the stream. There is **one-to-one** mapping between input and output element. For example, we can write a program to find the **date of birth of all employees** in a stream of employees.

In case of `flatMap()`, a **one-to-many** mapping is created where for each input element/stream, we first get a multiple values and then we flatten the values from all such input streams into a single output stream. For example, we may write program to find **all district words from all lines in a text file**.

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 flatMap\(\)](#)
2. [Jackson – Convert JSON to Map and Map to JSON](#)
3. [Java Stream map\(\)](#)
4. [Collecting Stream Items into Map in Java](#)
5. [Java Stream reuse – traverse stream multiple times?](#)
6. [Sorting a Map by Keys in Java](#)
7. [Java Sort Map by Values \(ascending and descending orders\)](#)
8. [\[Solved\] org.codehaus.jackson.map.JsonMappingException: No suitable constructor found for type](#)
9. [TypeScript Map](#)
0. [Java Stream peek\(\)](#)

Promoted by **usertesting.com**

Sponsored



A message from our sponsor

Join 7000+ Awesome Developers

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

Email Address

Subscribe

** We do not spam !!*

4 thoughts on “Java Stream map() vs flatMap()”

Vish

February 16, 2020 at 3:19 pm

Hi Lokesh,

Can you tell the best solution to write program for “find all distinct words from all lines in a text file” using Java8

[Reply](#)

anil

June 2, 2020 at 2:43 am

```
String file = "file_path_here";
Stream<String> str = null;
try {
    str = Files.lines(Paths.get(file));
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

List<String> distinctwords = str.map(s-> s.split("\\s+"))
    .flatMap(Arrays::stream)
    .distinct()
    .collect(Collectors.toList());

distinctwords.forEach(s-> System.out.println(s));
}
```

this should help

[Reply](#)**Amit**

September 27, 2019 at 5:15 pm

I wonder how do you find time to share so much knowledge?

[Reply](#)

Lokesh Gupta

September 27, 2019 at 10:28 pm

Its always difficult to get time but i guess its all about priorities.

[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 ...

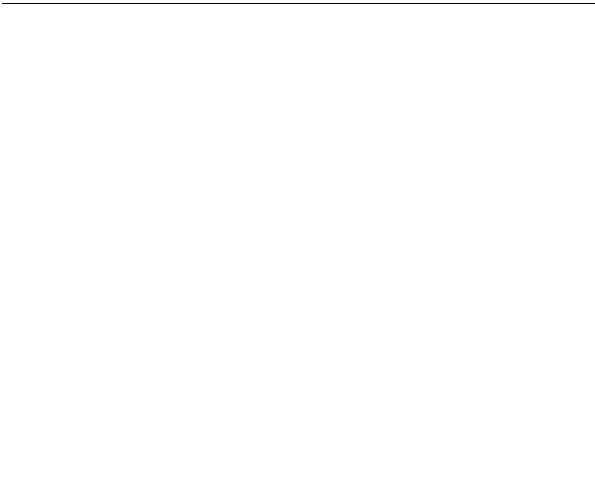




Promoted by **usertesting.com**
Sponsored



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](#)