

Convert Iterable or Iterator to Stream in Java

📅 Last Updated: March 14, 2022 👤 By: Lokesh Gupta 📁 Java 8 🔗 Java Iterator, Java Stream Basics

Learn to convert **Iterable** or **Iterator** to **Stream**. It may be desired at times when we want to utilize excellent support of **lambda expressions** and collectors in Java Stream API.

Table Of Contents

1. [Converting Iterable to Stream](#)
2. [Converting Iterator to Stream – Java 8](#)
3. [Converting Iterator to Stream – Java 9](#)

1. Converting Iterable to Stream

The *Iterables* are useful but provide limited support for lambda expressions added in Java 8. To utilize full language features, it is desired to convert the iterable to stream.

To convert, we will use `iterable.splitIterator()` method to get the **Splititerator** reference, which is then used to get the **Stream** using `StreamSupport.stream(splititerator, isParallel)` method.

```
//Iterable  
Iterable<String> iterable = Arrays.asList("a", "b", "c");
```

```
//Iterable -> Stream
```

```
//false means sequential stream
```

```
Stream<String> stream = StreamSupport.stream(iterable.splititerator(), false);
```

- The above code is only linking the `Stream` to the `Iterable` but the **actual iteration won't happen until a terminal operation is executed**.
- The second argument in `StreamSupport.stream()` determines if the resulting `Stream` should be parallel or sequential. Set it **true for a parallel stream and false for sequential stream**.

2. Converting Iterator to Stream – Java 8

The *Iterator* to *Stream* conversion follows the same path as *Iterable* to *Stream*.

The only difference is that the `Iterator` interface has no `splititerator()` method so we need to use `Spliterators.splititeratorUnknownSize()` method to get the `splititerator`. Rest everything is same.

```
// Iterator
```

```
Iterator<String> iterator = Arrays.asList("a", "b", "c")  
    .listIterator();
```

```
//Extra step to get Spliterator
```

```
Spliterator<String> splitltr = Spliterators  
    .splititeratorUnknownSize(iterator, Spliterator.ORDERED);
```

```
// Iterator -> Stream
```

```
Stream<String> stream = StreamSupport.stream(splitltr, false);
```

3. Converting Iterator to Stream – Java 9

Java 9 has made the syntax a little easier and now we don't need to use `Spliterator` explicitly. Rather it uses a `Predicate` to decide until when the elements shall be taken.

```
// Iterator  
Iterator<String> iterator = Arrays.asList("a", "b", "c")  
    .listIterator();  
  
Stream<String> stream = Stream.generate(() -> null)  
    .takeWhile(x -> iterator.hasNext())  
    .map(n -> iterator.next());
```

Drop me your questions in comments.

Happy Learning !!

[Sourcecode on Github](#)

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. [How Iterator works in java](#)
2. [Java Iterator interface example](#)
3. [Java Read XML with StAX Parser – Cursor & Iterator APIs](#)
4. [Difference between enumerator and iterator?](#)
5. [Iterator Design Pattern](#)
6. [Java Stream reuse – traverse stream multiple times?](#)

7. [Convert between Stream and Array](#)
8. [Java Stream count\(\) Matches with filter\(\)](#)
9. [Java Stream sorted\(\)](#)
10. [Java Stream peek\(\)](#)

Join 7000+ Awesome Developers

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

Email Address

Subscribe

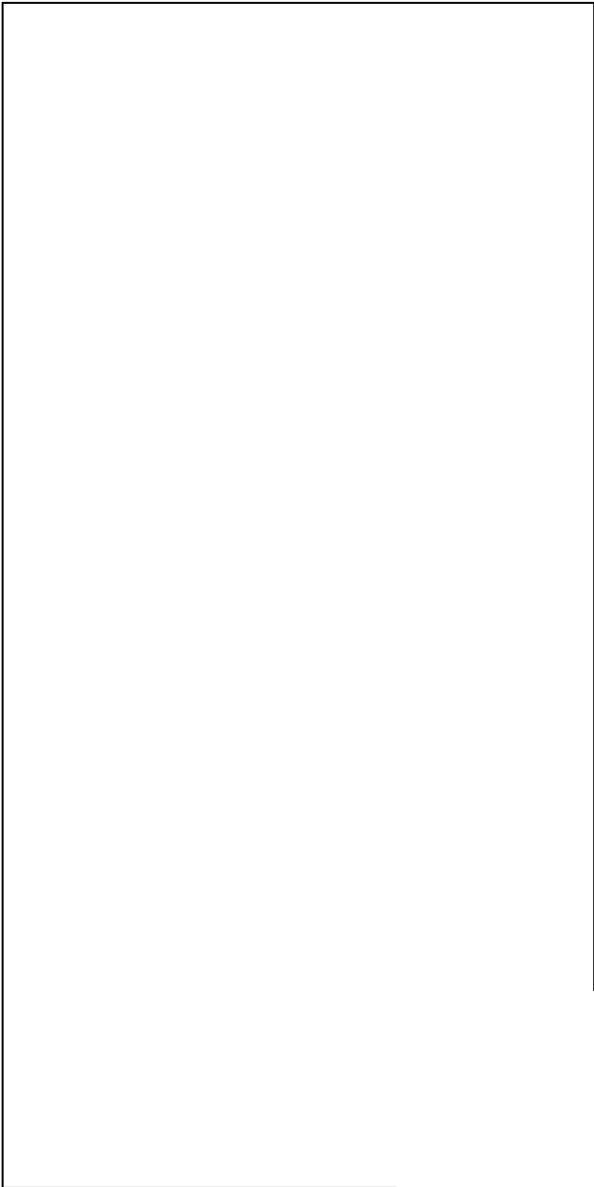
** We do not spam !!*

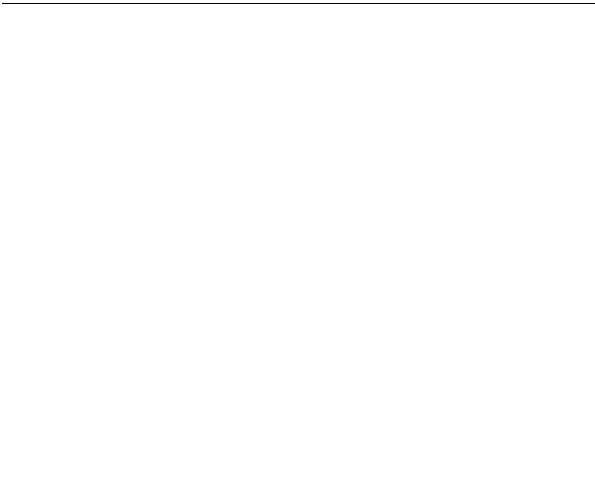
Leave a Comment

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

Post Comment







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](#)