Boxed Streams in Java

Last Updated: March 3, 2022 By: Lokesh Gupta Java 8 Java 8, Java Stream Basics

In Java, a **boxed stream is a stream of the wrapper class instances to simulate a stream of primitives**.

Table Of Contents

1. What is a Boxed Stream?

Java Stream API has been designed to work with objects, similar to Collections API. Streams do not treat the primitive types the same as objects.

In Stream API, a stream of primitives can be represented by the following 3 classes:

- `IntStream`
- `LongStream`
- `DoubleStream`

To convert from a stream of primitives to a stream of objects, these classes provide `boxed()` method that returns a *Stream* consisting of the elements of the given stream, each boxed to an object of the corresponding wrapper class.

```
Stream<Integer> stream = IntStream.of(1, 2, 3, 4, 5).boxed();

Stream<Long> stream1 = LongStream.of(1, 2, 3, 4, 5).boxed();

Stream<Double> stream2 = DoubleStream.of(1.0, 2.0, 3.0, 4.0, 5.0).box
```

2. Need of Boxed Streams

Without boxing the stream items, we cannot perform the regular stream operations on them. For example, we cannot collect the `int` values to a list, directly.

```
//Compilation issue
/*List<Integer> list = IntStream.of(1,2,3,4,5)
    .collect(Collectors.toList());*/
```

To make the above collecting process work, we must box the stream items first.

```
//Works fine
List<Integer> list = IntStream.of(1,2,3,4,5)
  .boxed()
  .collect(Collectors.toList());
```

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. Using 'if-else' Conditions with Java Streams
2. Creating Infinite Streams in Java
3. Sorting Streams in Java
4. Applying Multiple Conditions on Java Streams
5. Finding Max and Min from List using Streams
6. Java Streams API
7. Creating Streams in Java
8. Primitive Type Streams in Java
9. Java Stream sorted()
0. Java Stream findFirst()

3 thoughts on “Boxed Streams in Java”

Tim

May 3, 2019 at 1:10 am

```
int[] answer = new int[] { 3, 2 }
List<Integer> expected = Arrays.stream(answer).boxed().collect(Collectors.t
Object[] objArrayOfIntArrays = new Object[] {
                new int[] { 1, 2 }, // add 2
                new int[] { 1, 2 }, // add 2
                new int[] { 3, 2 }, // query freq 2
                new int[] { 2, 2 }, // delete 2
                new int[] { 3, 2 }, // query freq 2
            }
```

My question is, is there a way to turn `objArrayOfIntArrays` to `List<List<Integer>>` ?

Reply

Praveen

May 22, 2019 at 8:45 am

Hi Tim,

```
Object[] objArrayOfIntArrays1 = new Object[] { new int[] { 1, 2 }, // add
    new int[] { 1, 2 }, // add 2
    new int[] { 3, 2 }, // query freq 2
    new int[] { 2, 2 }, // delete 2
    new int[] { 3, 2 }, // query freq 2
};

List<List<Integer>> primitiveList = Arrays
          .stream(objArrayOfIntArrays1)
          .map(q -> {
            return Arrays.stream((int[]) q)
                    .boxed()
              .collect(Collectors.toList());
            }).collect(Collectors.toList());
```

Hope it Helps,

Praveen

Reply

Praveen

May 22, 2019 at 7:49 pm

```
Object[] objArrayOfIntArrays1 = new Object[] {
            new int[] { 1, 2 }, // add 2
            new int[] { 1, 2 }, // add 2
            new int[] { 3, 2 }, // query freq 2
            new int[] { 2, 2 }, // delete 2
            new int[] { 3, 2 }, // query freq 2
        };

    List<List<Integer>> primitiveList=Arrays.stream(objArrayOfIntArrays1)
          .map(q->{return Arrays.stream((int[]) q).boxed().collect(Collect
```

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 …

Two fifths of all older people in the UK say television is their main form of company.
Want to help them get out of the house?
DONATE HERE
SOTHEBY MEWS

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