HowToDoInJava

Stream of Random Numbers in Java

Last Updated: March 4, 2022 | By: Lokesh Gupta | Java 8 | Java Stream Basics, Random Number

Learn to get a **Stream of random numbers** in Java using the *Random* and *SecureRandom* classes.

Table Of Contents

1. The `Random` API

Java 8 release has added several methods to the `Random` class which can return **a sequential stream of random numbers** (*integers*, *longs* and *doubles*). The most widely used methods are:

- *IntStream* **ints()**
- *LongStream* **longs()**
- *DoubleStream* **doubles()**

All of the above methods have their overloaded forms. For example, `ints()` method has these overloaded forms.

- *IntStream ints()* – Returns an effectively unlimited stream of pseudorandom `int` values.
- *IntStream ints(long streamSize)* – Returns a stream producing the given number of pseudorandom `int` values.
- *IntStream ints(int origin, int bound)* – Returns an effectively unlimited stream of `int` values, each conforming to the given origin (inclusive) and bound (exclusive).
- *IntStream ints(long streamSize, int origin, int bound)* – Returns a stream producing the given number of pseudorandom `int` values, each conforming to the given origin (inclusive) and bound (exclusive).

2. Example of Stream of Random Numbers

Let's learn to use the above-discussed methods to create a stream of random numbers.

```
Random random = new Random();

//1
IntStream randStream = random.ints(5);
randStream.forEach(System.out::println);

//2
DoubleStream doubleStream = random.doubles(5, 0, 0.5);
doubleStream.forEach(System.out::println);

//collect to list
List<Long> longs = random.longs(5)
   .boxed()
   .collect(Collectors.toList());
```

3. Stream of Secure Random Numbers

To get the stream of secure random numbers (i.e. cryptographically strong random number), use the subclass `SecureRandom`. **By default, *SecureRandom* uses the SHA1PRNG algorithm.** The default constructor also uses this algorithm.

CSPRNG (cryptographically strong pseudo-random *number generator*) uses entropy, which is nothing but an unpredictable input (true random source). It might be a hardware random number generator or possibly some unpredictable system process, such as the timings events, interrupts etc.

```
SecureRandom secureRandomGenerator = SecureRandom.getInstanceStrong()

//1. Get 128 random bytes
byte[] randomBytes = new byte[128];
secureRandomGenerator.nextBytes(randomBytes);

//2. Get random integer
int r = secureRandomGenerator.nextInt();

//3. Get random integer in range
int randInRange = secureRandomGenerator.nextInt(999999);
```

Please note that all the above-discussed methods (*ints(), longs(), doubles() and their overloads*) also work with the `SecureRandom` class.

```
SecureRandom sRand = SecureRandom.getInstanceStrong();

IntStream randStream = sRand.ints(5);

randStream.forEach(System.out::println);
```

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. Java Random Numbers Generators
2. Secure Random Number Generation in Java
3. Java Stream reuse – traverse stream multiple times?
4. Java Regex for North American Phone Numbers
5. Java regex to validate international phone numbers
6. Java Regex to Validate SSN (Social Security Numbers)
7. Java program to find prime numbers from 2 to N
8. Java program to find first N prime numbers
9. Java program to swap two numbers
0. Program to calculate average of N numbers

Leave a Comment

Name *

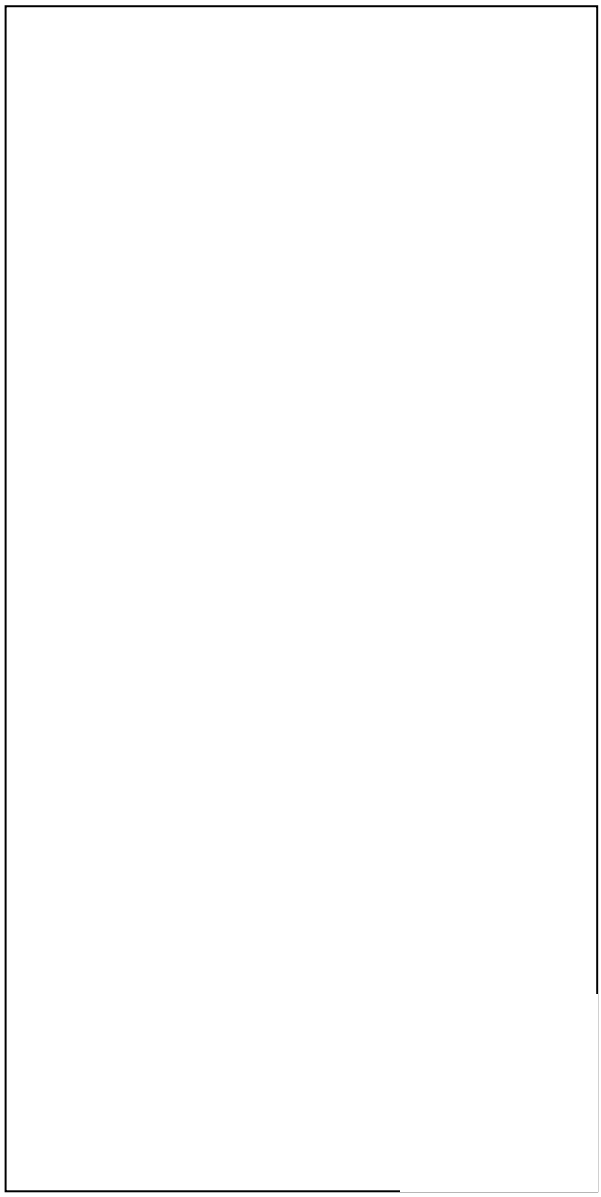
Email *

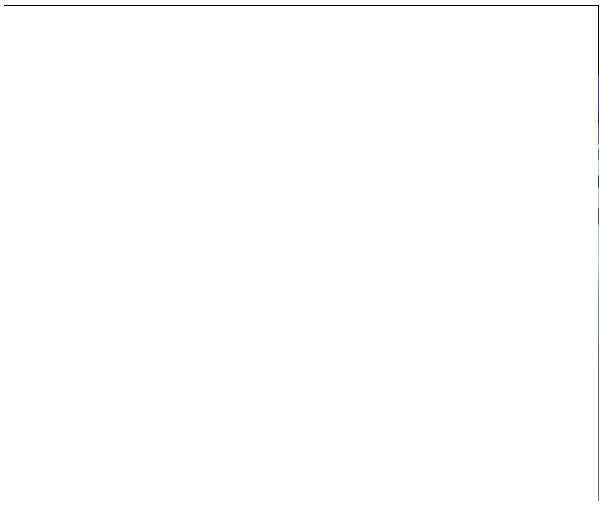
Website

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