10 Examples of Stream API in Java 8 - count + filter + map + distinct + collect :

**lambda expression and Stream API,** which takes advantage of parallel processing and functional operations like [filter](http://www.java67.com/2018/03/java-8-stream-find-first-and-filter-example.html), [map](http://www.java67.com/2016/09/map-reduce-example-java8.html), [flatMap](http://www.java67.com/2016/03/how-to-use-flatmap-in-java-8-stream.html), etc. Since then, a lot of Java developers are trying their hands to learn those significant changes like lambda expression, method reference, new Date and Time classes, and, more importantly, Stream API for bulk data operations.

Streams are one of the most important additions on JDK, it allows you to leverage other changes like [lambda expression](http://javarevisited.blogspot.sg/2014/02/10-example-of-lambda-expressions-in-java8.html), [method reference](https://javarevisited.blogspot.com/2017/08/how-to-convert-lambda-expression-to-method-reference-in-java8-example.html), functional interface, and internal iteration introduced via the [forEach()](https://javarevisited.blogspot.com/2015/09/java-8-foreach-loop-example.html" \l "axzz5Hvw8WGhy) method.  
  
Some of the most common things we do with Streams are filtering a collection, applying map and reduce function on all elements of the collection and taking advantage of lazy evaluation, built-in parallelism via parallelStream().