java.util.stream.Stream Operations

Operation	Return Type	Type Used
filter	Stream<T>	Predicate<T>
distinct	Stream<T>	
limit	Stream<T>	long
map	Stream<R>	Function<T,R>
flatMap	Stream<R>	Function<T,Stream<R>>
sorted	Stream<T>	Comparator<T>
anyMatch/noneMatch/allMatch	boolean	Predicate<T>
findAny/findFirst	Optional<T>	
forEach	void	Consumer<T>
collect	R	Collector<T,A,R>
reduce	Optional<T>	BinaryOperator<T>
count	long	
mapToInt (similarly for Long/Double)	IntStream	ToIntFunction<T>

java.util.stream.Collectors Factory Methods

Factory Method	Return Type	Used to
toList	List<T>	Gather into a list
toSet	Set<T>	Gather into a set
counting	Long	Count items
summingInt/averagingInt	Integer/Double	Summing/Averaging
summarizingInt	IntSummaryStatistics	Max,Min,Total,Average
maxBy/minBy	Optional<T>	Max/Min with Comparator
reducing	Type of reduction	Reduce to single value
mapping	Type produced by mapping	Map one type to another
groupingBy	Map<K,List<T>>	Group by K
partitioningBy	Map<Boolean,List<T>>	Group by true/false

groupingBy and **partitioningBy** can take a **Collector** as a second argument, which will change the value (second) type of the returning **Map**.

Stream Data Sources

Source	Generator	Returned Type
Values	Stream.of(T... values)	Stream<T>
Array	java.util.Arrays.stream(int[]) (similarly for long[] and double[])	IntStream
Array	java.util.Arrays.stream(T[])	Stream<T>
Numerical range (similarly for long)	IntStream.range(lo,hi_exclusive)/ IntStream.rangeClosed(lo,hi_inclusive)	IntStream
File	java.nio.file.Files.lines(Path) Path ← Paths.get(filename)	Stream<T>
Iterate	Stream.iterate(T,UnaryOperator<T>)	Stream<T>
Generate	Stream.generate(Supplier<T>)	Stream<T>