

Coding Dojo Java Streams

Mardi 05 mars 2024

Stream, quand et comment ?

> Depuis le JDK 8, en 2014

Here are all new features in Java 8 by category.

JVM

- Remove Permanent Generation (JEP 122)

Language

- Lambda Expressions (JSR 335)
- Default Methods in Interfaces (JSR 335)

Collection.java

```
742  @Contract(pure = true)
743  default Stream<E> stream() {
744      return StreamSupport.stream(spliterator(), parallel: false);
}
```


Construction fonctionnelle

- > Favorise l'immutabilité
- > Map / Filter / Reduce
- > Opérations intermédiaires et terminales
- > Plus lisible dans certains cas, parfois non... (cf. API collector dans tous ses états)



THALES GROUP LIMITED DISTRIBUTION

OBJECTIF DU DOJO : FAIRE LES TODO SANS BOUCLE FOR

Nous allons utiliser les données du World Development Indicator [WDI - Home \(worldbank.org\)](https://data.worldbank.org/)

3 fichiers CSV: *Countries*, *Series* et *Data*

Données déjà parsées dans la classe **DataStore** dans des collections d'objets : **Country**, **Series** et **DataEntry**

Java 8 Streams Cheat Sheet

For more awesome cheat sheets
visit rebellabs.org!



Definitions

- ✓ A stream **is** a pipeline of functions that can be evaluated.
- ✓ Streams **can** transform data.
- ✗ A stream **is not** a data structure.
- ✗ Streams **cannot** mutate data.

Intermediate operations

- Always return streams.
- Lazily executed.

Common examples include:

Function	Preserves count	Preserves type	Preserves order
<i>map</i>	✓	✗	✓
<i>filter</i>	✗	✓	✓
<i>distinct</i>	✗	✓	✓
<i>sorted</i>	✓	✓	✗
<i>peek</i>	✓	✓	✓

Stream examples

Get the unique surnames in uppercase of the first 15 book authors that are 50 years old or over.

```
library.stream()
    .map(book -> book.getAuthor())
    .filter(author -> author.getAge() >= 50)
    .distinct()
    .limit(15)
    .map(Author::getSurname)
    .map(String::toUpperCase)
    .collect(toList());
```

Compute the sum of ages of all female authors younger than 25.

```
library.stream()
    .map(Book::getAuthor)
    .filter(a -> a.getGender() == Gender.FEMALE)
    .map(Author::getAge)
    .filter(age -> age < 25)
    .reduce(0, Integer::sum);
```

Terminal operations

- Return concrete types or produce a side effect.
- Eagerly executed.

Common examples include:

Function	Output	When to use
reduce	concrete type	to cumulate elements
collect	list, map or set	to group elements
forEach	side effect	to perform a side effect on elements

Parallel streams

Parallel streams use the common ForkJoinPool for threading.

```
library.parallelStream()...
```

or intermediate operation:

```
IntStream.range(1, 10).parallel()...
```

Useful operations

Grouping:

```
library.stream().collect(
    groupingBy(Book::getGenre));
```

Stream ranges:

```
IntStream.range(0, 20)...
```

Infinite streams:

```
IntStream.iterate(0, e -> e + 1)...
```

Max/Min:

```
IntStream.range(1, 10).max();
```

FlatMap:

```
twitterList.stream()
    .map(member -> member.getFollowers())
    .flatMap(followers -> followers.stream())
    .collect(toList());
```

Pitfalls

- ✗ Don't update shared mutable variables i.e.

```
List<Book> myList =
    new ArrayList<>();
library.stream().forEach(
    (e -> myList.add(e));
```
- ✗ Avoid blocking operations when using parallel streams.

BROUGHT TO YOU BY
JRebel

TIPS & LIENS

On a le droit d'utiliser *forEach* pour afficher des résultats.

Pour les TODO 10 & 11 il faut utiliser la bibliothèque de José PAUMARD

[GitHub - JosePaumard/streams-utils](#)

- [L'API Collector dans tous ses états \(José Paumard\) – YouTube](#)
- [Stream \(Java SE 17 & JDK 17\) \(oracle.com\)](#)
- <https://www.jrebel.com/system/files/java-8-streams-cheat-sheet.pdf>



Contact

MORTIER Nicolas

Architecte Logiciel – DT BL CIS



+33 (0) 6 59 48 85 59



nicolas.mortier@thalesgroup.com