

Working with Streams – The Basics



Jesper de Jong
Software Architect

@jesperdj www.jesperdj.com

Overview



Streams concepts

Differences between streams and collections

Obtaining streams

Filtering, transforming, searching streams

Collecting streams

Understanding Streams

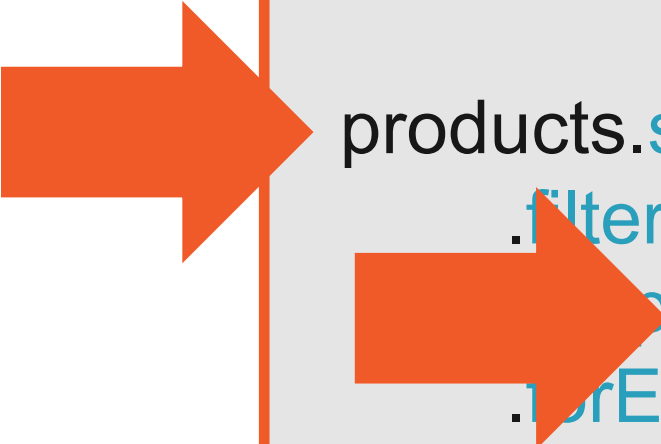
What Is a Stream?

The stream pipeline

```
products.stream()  
  .filter(product -> product.getCategory() == Category.FOOD)  
  .map(Product::getName)  
  .forEach(System.out::println);
```

What Is a Stream?

The stream pipeline



```
products.stream()  
    .filter(product -> product.getCategory() == Category.FOOD)  
    .map(Product::getName)  
    .forEach(System.out::println);
```

interface `java.util.stream.Stream`

Intermediate and Terminal Operations

```
products.stream()
```

Intermediate operations



```
.filter(product -> product.getCategory() == Category.FOOD)  
.map(Product::getName)
```

Terminal operation



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

Stream processing is lazy

Differences between Streams and Collections

Differences between Streams and Collections

Collection

Stores elements in a data structure

```
list.sort()
```

Eager evaluation

Imperative programming

Do modify the collection

Can be iterated multiple times

Never infinite

Stream

Does not store elements

Lazy evaluation

```
stream.sorted()
```

Functional programming

Does not modify its source

Iterating consumes the stream

May be infinite

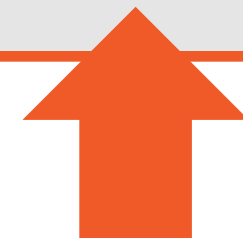
Internal vs External Iteration

External iteration

```
for (int i = 0; i < products.size(); i++) {  
    Product p = products.get(i);  
    System.out.println(product);  
}
```

Internal iteration

```
products.stream()  
    .forEach(System.out::println);
```



Obtaining Streams

Filtering and Transforming Streams

Searching in Streams

Searching in Streams

```
filter(...)  
findFirst() or findAny()
```

Find a particular element

```
anyMatch(...)  
allMatch(...)  
noneMatch(...)
```

Check if elements exist

Short-circuiting

Short-circuiting

`findFirst()`

`findAny()`

`anyMatch()`

Not short-circuiting

`allMatch()`

`noneMatch()`

Reducing and Collecting Streams

Summary



Streams concepts

- **Stream pipeline**
- **Intermediate and terminal operations**

Differences between streams and collections

- **Imperative vs functional programming**
- **External vs internal iteration**

Working with streams

- **map and flatMap**
- **Searching in streams**
- **Collecting streams**

Up Next: Working with Streams – In Depth
