

new spike

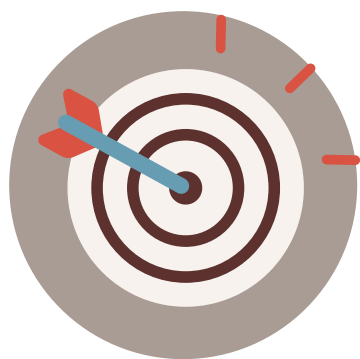
JAVA INTERVIEW QUESTIONS

STREAMS

Posted By

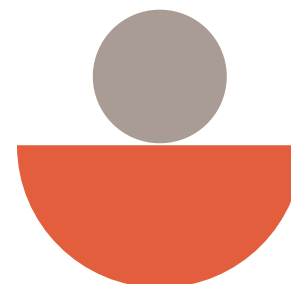
Andrei Saizu

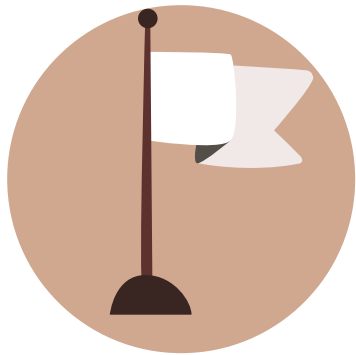




INTRO

Having spoken about **Functional Interfaces** and **Lambda Expressions**, it's only right we continue our series talking about **Streams**.



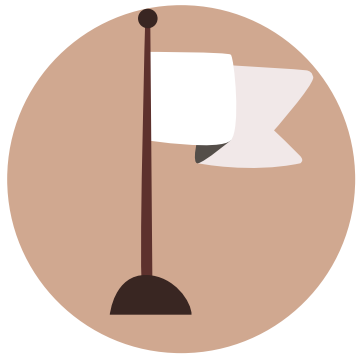


INITIALIZE STREAMS

There are multiple ways to create and initialize streams.

- ◆ `Collections.stream()`
- ◆ `Stream.builder()`
- ◆ `Stream.of()`
- ◆ `Arrays.stream()`

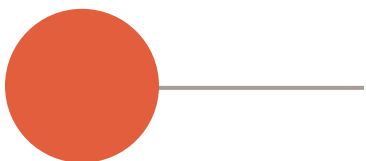


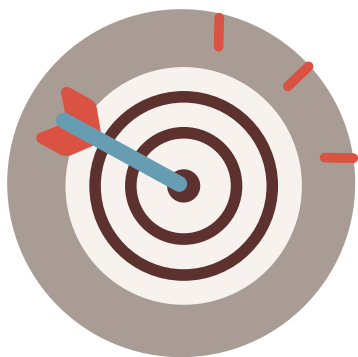
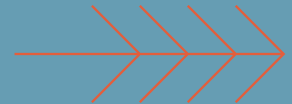


INTERMEDIATE OPERATIONS

While the elements are in the pipe you can:

- ◆ change their state and type
- ◆ filter out some of them
- ◆ skip some elements
- ◆ etc.

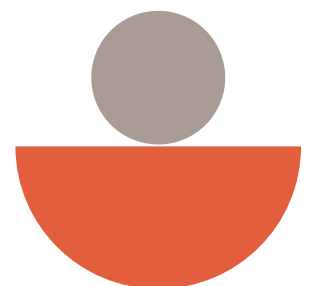


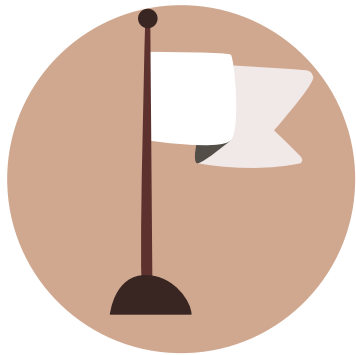


INTERMEDIATE OPERATIONS

! In a stream, the intermediate operations
are **lazy**.

Until you apply a **terminal operation**, the
intermediate ones won't be called.

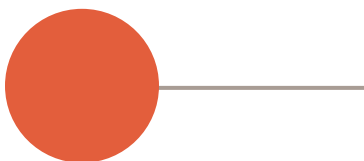


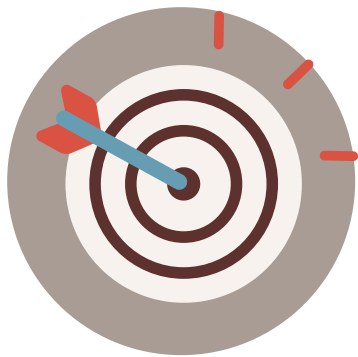


TERMINAL OPERATIONS

At the end of the pipe you can:

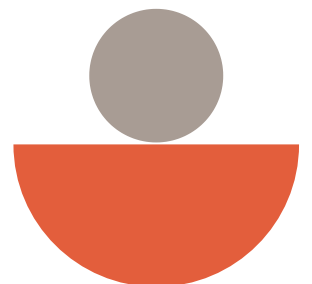
- ◆ collect the items into another dataset
- ◆ return a single result
- ◆ return void

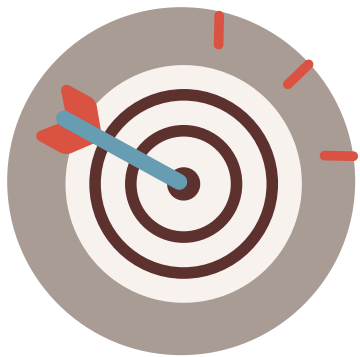




CLOSING STREAMS

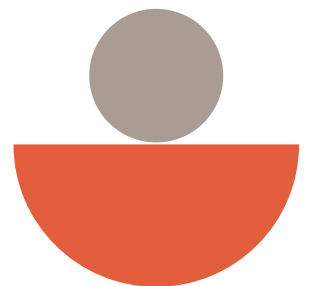
Streams implement the **AutoCloseable** interface, that means that they implement the **close()** method.





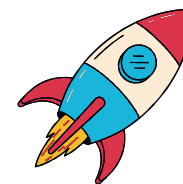
CLOSING STREAMS

! You should be careful when closing a stream, because once closed, it will throw an **IllegalStateException** if you try to operate on it again.





STAY UP TO DATE!



Check out the [link in the comments](#)
for a [free](#) in-depth breakdown on
how it works in the background!

