
AGENDA

Java Streams Basics – Practical Foundations for Beginners

1 day · Individual date · On-site or online · German / English · Customizable agenda · Certificate of completion

The course alternates between short theory segments, hands-on code examples, and interactive exercises – so you learn the fundamentals of the Stream API in a well-structured and practical way.

Functional Programming in Java:

- Lambda expressions: syntax, types, and usage
- Functional interfaces (Consumer, Supplier, Function, Predicate, etc.)
- Method references: static, instance, constructor references
- Hands-on exercises: Lambda expressions vs. anonymous classes

Stream Basics:

- What is a stream? Conceptual differences from collections
- Anatomy of a stream pipeline: Source → Intermediate Operations → Terminal Operations
- Lazy evaluation and its implications
- Creating streams: from collections, arrays, files, ranges
- Primitive streams: IntStream, LongStream, DoubleStream
- Hands-on exercises: First stream pipelines with primitive streams

Intermediate Operations Deep Dive:

- Transformations with `map()`, `flatMap()`, `filter()`
- Selection and sorting: `distinct()`, `sorted()`, `limit()`, `skip()`
- Debugging with `peek()` – used correctly
- Chaining operations and performance aspects
- Hands-on exercises: More complex data processing

Terminal Operations:

- `collect()` – the most versatile terminal operator
- Built-in collectors: `toList()`, `toSet()`, `toMap()`
- `forEach()`, `forEachOrdered()`
- `reduce()` for aggregations
- Finding operations: `findFirst()`, `findAny()`
- Matching operations: `allMatch()`, `anyMatch()`, `noneMatch()`
- Hands-on exercises: Terminal operations with standard collectors