+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Updated on 16/10/2024

Sign up

Micronaut training - Reactive programming with Java

3 days (21 hours)

Presentation

Micronaut is an open-source framework for building microservice applications that are easy to test and scale. Developed by OCI, the creators of Grails created Micronaut for rapid deployment of microservices.

One of the framework's most important features is compile-time dependency injection. Thanks to this feature, the framework is light on memory and startup. Micronaut offers several options, including the creation of applications supported by Java, Groovy or Kotlin.

Its Eureka and Consul features make it an excellent framework for developing cloud-native applications. Unlike other frameworks such as Spring or Grails, Micronaut has many features that are tailored to microservices, such as Reactive streams, message-driven microservices.

After completing our Reactive Programming with Java training course, you'll be able to quickly create applications with a microservices architecture, with conventional requirements and the latest functionalities.

Our Reactive Programming with Java course will introduce you to its latest version, Micronaut 4.7.

Objectives

- How to build a JVM-type application
- Creating microservices with different languages
- How to write functional tests
- How to write endpoints and use dependency injection (DI)
- Configure Micronaut applications to register with Consul

Target audience

Web developers

Prerequisites

Knowledge of languages, Java, Kotlin, Groovy

Program of our Reactive Programming with Java training course

Introduction ? Benefits of reactive programming + use cases Theoretical

principles

- Control reversal
- Aspect Oriented Programming
- Patterns: MVC, Observer, Publish-Subscribe, Chain-of-responsibility
- Functional paradigm
- Calculation models

Initial concepts of reactive programming in Java

- History of reactive programming
- Presentation of the main Java APIs implementing the various points covered in the "Theoretical principles" section
- Implementing observer and publish-subscribe patterns in Spring
- Le manifeste réactif - Reactive Manifesto
- The Reactive Stream API
- Benefits and limitations of RxJava
- Comparison of frameworks implementing the Reactive Stream API

Focus on the Reactor project

- Overview + comparison of versions 1.x and 2.x
- Essential Reactor? concepts Flow, mono, sequences, processor, error management, backpressure, hot vs. cold streams, composition and transformation
- Advanced Reactor? concepts Lifecycle and scheduling
- Integration with SpringBoot
- WebFlux vs Web MVC
- Asynchronous data access
- Testing and benchmarking
- Monitoring

Focus on Micronaut

- Review of theoretical principles in Micronaut? Lifecycle, Control Reversal, AoP
- Configuring a Micronaut application
- Filters
- Error management
- Controllers
- Cloud configuration
- Integration with other technologies: Reactor, Kafka, JDBC, Redis, etc.
- Testing and benchmarking

Conclusion? Wrap-up and returns

Companies concerned

This course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced computer technology, or to acquire specific business knowledge or modern methods.

Positioning on entry to training

Positioning at the start of training complies with Qualiopi quality criteria. As soon as registration is finalized, the learner receives a self-assessment questionnaire which enables us to assess his or her estimated level of proficiency in different types of technology, as well as his or her expectations and personal objectives for the training to come, within the limits imposed by the selected format. This questionnaire also enables us to anticipate any connection or security difficulties within the company (intra-company or virtual classroom) which could be problematic for the follow-up and smooth running of the training session.

Teaching methods

Practical course: 60% Practical, 40% Theory. Training material distributed in digital format to all participants.

Organization

The course alternates theoretical input from the trainer, supported by examples, with brainstorming sessions and group work.

Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

Sanction

A certificate will be issued to each trainee who completes the course.