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

Updated 10/10/2024

Sign up

Quarkus training

3 days (21 hours)

PRESENTATION

Quarkus is the leading framework in terms of startup time and memory consumption. Developed by Red Hat, it is designed for native Java applications and runs on the JVM (Java Virtual Machine).

Quarkus is able to combine in a single runtime environment advantages such as rapid development in scripting languages, the Java ecosystem and the efficiency of native compilation. Another advantage of Quarkus is that it adapts its development model to the application you're developing.

With Quarkus you can build lightweight Java code and native code from Java classes, and create container images that you can run on Kubernetes or OpenShift.

In our Quarkus training course, you'll discover how to develop Java Native applications, learn about microservices architecture with Quarkus and use GraalVM extensions.

As with all our training courses, this one will introduce you to the very latest version of Quarkus (at the time of writing: Quarkus 2.10).

OBJECTIVES

- Managing the Quarkus framework
- Developing Java Native applications
- Mastering microservices architecture
- Reduce memory consumption

TARGET AUDIENCE

- Developers
- Architects

Prerequisites

Knowledge of Java, Web architecture, HTTP protocol and REST concepts

OUR QUARKUS TRAINING PROGRAM

Introduction to Quarkus

- What is Quarkus
- The benefits of Quarkus
- Integration with Kubernetes
- Memory and first response time
- Basic workflow

Core de Quarkus

- Application configuration
- Custom configuration converter
- Application initialization and termination
- Testing your application
- Logging configuration

RESTful service development

- Creating a simple REST API endpoint
- Extract query parameters
- Using semantic HTTP response status codes
- Linking HTTP methods
- Activate cross-origin resource sharing (CORS)
- Using reactive routes
- Intercept HTTP requests
- Secure connections with SSL

Configuration

- Configure application with custom properties
- Access configuration properties programmatically
- Overwrite configuration values externally
- Configure with profiles
- Modification of recorder configuration
- Add application logs
- Advanced logging

- Configuration with customized profiles
- Creating custom sources
- Create custom converters
- Group configuration values
- Confirm configuration values

Quarkus programming model

- Packing/unpacking JSON
- Marshalling/Unmarshalling XML
- Validation of input and output values
- Create custom validations
- Programmatic object validation
- Dependency injection
- Executing object and application lifecycle events
- Using custom qualifiers
- Qualify and configure annotations
- Create interceptors
- Writing behavioral tests
- Writing unit tests
- Creating Mock Objects with Mockito
- Combine several annotations into a single meta-annotation
- Code execution before or after a test

Packaging Quarkus applications

- Command mode execution
- Creating an executable JAR file
- Packaging Über-JAR
- Building a native executable
- Building a Docker container for the JAR file
- Building a Docker container for a native file
- Building and Docking a native SSL application

Advanced Keycloak training

Pentest Web training

Resilient Infrastructure Training

Companies concerned

This training 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.