AMBIENT-IT

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

Updated on 27/06/2025

Sign up

Helidon.io training

3 days (21 hours)

Presentation

Master Helidon to create modern, high-performance, cloud-ready Java microservices. This comprehensive training course guides you through designing REST APIs, securing your services and deploying your applications.

You'll learn how to develop efficient REST services with a minimalist API, exploit virtual threads for maximum performance, and structure your projects around an externalized and typed configuration.

You'll know how to secure your services with JWT or OAuth2, manage access to your SQL databases, and expose robust, traceable endpoints. Helidon will also give you the tools to track, monitor and diagnose your services in a Kubernetes environment.

You'll be trained in native packaging with GraalVM, Docker/K8s deployment, and observability via Prometheus, Grafana and OpenTelemetry.

As with all our training courses, it will take place on my latest version of Helidon

Objectives

- Understand Helidon architecture and the differences between Helidon SE and Helidon MP.
- Install, configure and structure a Helidon application with Maven or the CLI, exploiting virtual threads.
- Develop robust REST APIs with Helidon SE or JAX-RS, handling routing, JSON formats JSON formats, error handling, and endpoint security with JWT or OAuth2
- Access SQL databases via the Helidon DB Client or JPA, manage transactions, and structure data processing in a modular, testable architecture
- Implement full observability with metrics, health checks, distributed traces, and deploy Helidon microservices in a Kubernetes environment using Docker.

Target audience

- Backend developer
- Fullstack developers
- Software architects

Prerequisites

- Basic knowledge of web development
- Mastery of the Java language

Helidon.io training program

Introduction to Helidon

- Differences between Helidon SE and Helidon MP
- Position in relation to Spring Boot, Micronaut and Quarkus
- Installing the Helidon CLI
- Using Maven archetypes
- Structure of a Helidon project
  - SE vs MP

Development with Helidon SE

- WebServer and Routing
- Imperative programming with virtual threads
- Lightness and performance
- Route configuration
- Request and response processing
- Using JSON support
- Application.yaml and application.properties files
- Injecting configuration into code
- Environment variables and profiles

Logging and debugging

- Configuring Java Util Logging or SLF4J
- Good traceability practices

Development with Helidon MP

- Introduction to Jakarta EE and Eclipse MicroProfile

- MP application architecture
- Resource creation
- Dependency injection with CDI
- Data formats
- Multiple configuration sources
- Config injection in beans
- Fault Tolerance
- Metrics
- Health checks
- OpenAPI and automatic documentation

Security and access to resources

- Helidon Security module: basic principles
- Role-based security configuration
- Integration with JWT and OAuth2
- Securing REST endpoints
- CSRF, CORS, security headers
- Secure storage of secrets
- Integration with external providers

Database access

- Connection configuration
- Fluent API queries
- Results mapping
- Explicit start/commit/rollback
- Error handling
- Integration with JPA via Hibernate
- Access to NoSQL

Observability and monitoring

- Integration with SLF4J/Logback
- Structured logs
- OpenTracing / OpenTelemetry support
- Integration with Jaeger or Zipkin
- Prometheus display
- Customized dashboards

Deployment and production

- Using jlink
- Creating an executable JAR with Maven
- Using GraalVM Native Image
- Dockerfile for Helidon
- Deployment.yaml file and readiness/liveness probes
- ConfigMap and Secret
- Integration with Prometheus, Grafana

- Logging and alerting

Performance and best practices

- Use of virtual threads (Java 21)
- Lazy loading / light threads
- Profiling with JFR
- Memory, startup, response time
- Choice according to use cases
- Breakdown into modules
- Business/infrastructure separation
- Dependency injection with Helidon Inject

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 forthcoming course, 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 training: 60% hands-on, 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 is used to check that skills have been correctly acquired.

Certification

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