

Mis à jour le 27/06/2025

S'inscrire

Formation Helidon.io

3 jours (21 heures)

Présentation

Maîtrisez Helidon pour créer des microservices Java modernes, performants et cloud-ready. Cette formation complète vous guide dans la conception d'APIs REST, la sécurisation de vos services, et le déploiement de vos applications.

Vous apprendrez à développer des services REST efficaces avec une API minimaliste, à exploiter les virtual threads pour des performances maximales, et à structurer vos projets autour d'une configuration externalisée et typée.

Vous saurez sécuriser vos services avec JWT ou OAuth2, gérer les accès à vos bases SQL, et exposer des endpoints robustes et traçables. Helidon vous offrira aussi les outils pour tracer, surveiller et diagnostiquer vos services dans un environnement Kubernetes.

Vous serez formé au packaging natif avec GraalVM, au déploiement Docker/K8s, et à l'observabilité via Prometheus, Grafana et OpenTelemetry.

Comme pour toutes nos formations, elle se déroulera sur ma toute dernière version de [Helidon](#)

Objectifs

- Comprendre l'architecture de Helidon et les différences entre Helidon SE et Helidon MP.
- Installer, configurer et structurer une application Helidon avec Maven ou le CLI, en exploitant les virtual threads.
- Développer des APIs REST robustes avec Helidon SE ou JAX-RS, en gérant le routage, les formats JSON, les erreurs, et la sécurisation des endpoints avec JWT ou OAuth2
- Accéder à des bases de données SQL via le Helidon DB Client ou JPA, gérer les transactions, et structurer les traitements de données dans une architecture modulaire et testable
- Mettre en place une observabilité complète avec metrics, health checks, traces distribuées, et déployer les microservices Helidon dans un environnement Kubernetes avec Docker.

Public visé

- Développeur Backend
- Développeur fullstack
- Architectes logiciels

Pré-requis

- Connaissances de base en développement web
- Maîtrise du langage Java

Programme de la formation Helidon.io

Introduction à Helidon

- Différences entre Helidon SE et Helidon MP
- Position face à Spring Boot, Micronaut, Quarkus
- Installation du CLI Helidon
- Utilisation des archetypes Maven
- Structure d'un projet Helidon
 - SE vs MP

Développement avec Helidon SE

- WebServer et Routing
- Programmation impérative avec virtual threads
- Légèreté et performance
- Configuration des routes
- Traitement des requêtes et réponses
- Utilisation du JSON support
- Fichiers application.yaml et application.properties
- Injection de configuration dans le code
- Variables d'environnement et profils

Logging et Debug

- Configuration du logger Java Util Logging ou SLF4J
- Bonnes pratiques de traçabilité

Développement avec Helidon MP

- Introduction à Jakarta EE et Eclipse MicroProfile

- Architecture d'une application MP
- Création de ressources
- Injection de dépendances avec CDI
- Formats de données
- Sources de configuration multiples
- Injection de config dans les beans
- Fault Tolerance
- Metrics
- Health checks
- OpenAPI et documentation automatique

Sécurité et accès aux ressources

- Helidon Security module : principes de base
- Configuration de la sécurité par rôles
- Intégration avec JWT et OAuth2
- Sécurisation des endpoints REST
- CSRF, CORS, en-têtes de sécurité
- Stockage sécurisé des secrets
- Intégration avec des fournisseurs externes

Accès aux bases de données

- Configuration des connexions
- Requêtes avec Fluent API
- Mapping des résultats
- Début/commit/rollback explicites
- Gestion des erreurs
- Intégration avec JPA via Hibernate
- Accès à NoSQL

Observabilité et monitoring

- Intégration avec SLF4J/Logback
- Logs structurés
- OpenTracing / OpenTelemetry support
- Intégration avec Jaeger ou Zipkin
- Affichage Prometheus
- Tableaux de bord personnalisés

Déploiement et production

- Utilisation de jlink
- Création d'un JAR exécutable avec Maven
- Utilisation de GraalVM Native Image
- Fichier Dockerfile pour Helidon
- Fichier deployment.yaml et readiness/liveness probes
- ConfigMap et Secret
- Intégration avec Prometheus, Grafana

- Logging et alerting

Performances et bonnes pratiques

- Utilisation des virtual threads (Java 21)
- Chargement lazy / threads légers
- Profiling avec JFR
- Mémoire, démarrage, temps de réponse
- Choix en fonction des cas d'usage
- Découpage en modules
- Séparation métier / infrastructure
- Injection de dépendances avec Helidon Inject

Sociétés concernées

Cette formation s'adresse à la fois aux particuliers ainsi qu'aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son inscription définitive, l'apprenant reçoit un questionnaire d'auto-évaluation nous permettant d'apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d'anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l'acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.

