

Mis à jour le 29/01/2026

S'inscrire

Formation Red Hat Fuse

4 jours (28 heures)

Présentation

Notre formation Red Hat Fuse vous permettra de concevoir, déployer et superviser des flux d'intégration modernes, depuis des traitements simples (fichier ? base de données) jusqu'à la publication d'APIs REST conteneurisées, prêtes à être exécutées dans une plateforme cloud-native comme OpenShift.

Vous apprendrez à utiliser Apache Camel, les composants d'intégration, les Enterprise Integration Patterns (EIP), et à créer des routes robustes et sécurisées.

Vous saurez créer des microservices Camel en Spring Boot, les conteneuriser avec Docker, les intégrer dans un pipeline CI/CD, et les déployer dans OpenShift avec supervision intégrée. Un point d'attention particulier sera mis sur la modularité, la maintenabilité et l'intégration continue des flux d'entreprise.

À l'issue de la formation, vous serez en mesure de développer et exploiter une architecture d'intégration agile, interopérable, évolutive et DevOps-compatible.

Comme toutes nos formations, celle-ci vous présentera **la dernière version stable** de la technologie et ses nouveautés.

Objectifs

- Comprendre l'architecture de Red Hat Fuse et son cœur Apache Camel
- Créer et déployer des routes d'intégration : REST, fichier, base, API
- Utiliser les composants Camel (HTTP, JMS, Kafka, JDBC...)
- Conteneuriser et exécuter vos intégrations avec Spring Boot et Docker
- Déployer vos flux dans OpenShift et les intégrer dans un pipeline CI/CD
- Superviser vos intégrations via Hawtio, Prometheus et Grafana

Public visé

- DevOps
- Développeurs backend
- Architectes techniques
- Administrateurs applicatifs
- Toute équipe technique cherchant à bâtir un ESB distribué, léger et cloud-ready

Pré-requis

- Bonnes bases en Java et en ligne de commande
- Notions de base en HTTP, API, fichiers, bases de données
- Connaissances en conteneurisation (Docker) appréciées
- Connaissances de Kubernetes/OpenShift utiles mais non obligatoires

Programme de notre formation Red Hat Fuse

Introduction à Red Hat Fuse et à l'intégration moderne

- Les enjeux de l'intégration applicative dans un SI hybride
- Fuse vs ESB traditionnels vs iPaaS modernes
- Architecture de Red Hat Fuse et composants open source intégrés
- Apache Camel, la colonne vertébrale de Fuse
- Cas d'usage DevOps, cloud, API, microservices
- Panorama des outils complémentaires : AMQ, 3scale, OpenShift

Premiers pas avec Apache Camel

- Structure d'une route Camel : endpoint ? processor ? output
- Formats supportés (Java DSL, XML DSL, YAML)
- Introduction aux Enterprise Integration Patterns (EIP)
- Mise en place d'un projet Maven Camel simple
- Outils : IntelliJ, VS Code, Fuse Tooling, hawtio
- Atelier : Créer une route Camel locale pour lire un fichier CSV, le transformer, et écrire une sortie JSON

Connecteurs et transformation de données

- Les composants Camel (HTTP, JMS, Kafka, JDBC, File...)
- Ajout d'un connecteur à une base de données
- Utilisation de Processors (Java) et beans
- Transformations simples : JSON ? XML, CSV ? objet Java
- Appels REST et formats structurés (Jackson, JAXB, etc.)

Routage, logique et gestion des erreurs

- Content-Based Router, Choice, Splitter, Aggregator
- Traitement conditionnel, filtrage, transformation
- Gestion d'erreur : Try-Catch, Dead Letter Channel
- Redelivery policy et circuit breaker
- Logique métier distribuée dans les flux

Créer des APIs REST avec Camel

- Définir un service REST exposé avec Camel (REST DSL)
- Mapping de routes sur des endpoints HTTP
- Swagger/OpenAPI : génération de doc automatique
- Gestion des paramètres (path, query, headers, body)
- Atelier : Créer une API REST CRUD exposée par Fuse avec transformation JSON ? SQL

Intégration avec les bases de données

- Accès en lecture/écriture via JDBC ou JPA
- Mapping objets/colonnes, batchs de requêtes
- Déclenchement par polling ou événement
- Logs d'intégration, traitement d'erreurs SQL
- Synchronisation entre deux bases

Conteneurisation avec Spring Boot et Docker

- Exécuter Camel dans un projet Spring Boot autonome
- Configuration via application.properties et profil Spring
- Création d'une image Docker du microservice Camel
- Bonnes pratiques : healthcheck, secrets, ports
- Atelier : Conteneuriser une route Camel dans Spring Boot + Docker build + test REST local

Déploiement cloud-native avec OpenShift

- Concepts OpenShift : pods, routes, S2I, ConfigMaps
- Déploiement de microservices Camel sur OpenShift
- Intégration avec pipelines CI/CD (Git + build auto)
- Création de route HTTP publique + monitoring
- Atelier : Déployer un microservice Camel sur OpenShift avec configuration dynamique

Sécurité, authentification et filtrage

- Sécuriser une API REST avec Basic Auth ou OAuth2
- Gestion des headers, tokens, et validation d'entrée
- Exposition via HTTPS sur OpenShift
- Accès contrôlé à des bases ou services internes
- Logging et masquage de données sensibles

Monitoring et supervision des routes

- Supervision avec hawtio : statut, metrics, logs
- Intégration Prometheus + Grafana
- Détection d'erreurs, timeout, latence
- Journaux d'audit et messages rejetés
- Alerting via OpenShift ou Prometheus

Scénarios d'intégration distribués

- Orchestration de microservices via Fuse
- Appels asynchrones, files JMS ou Kafka
- Patterns de saga ou compensation
- Traitement d'événements (EDA)
- Synchronisation inter-systèmes (API ? queue ? DB)

Bonnes pratiques DevOps et industrialisation

- Structuration des projets Camel : modules, profiles
- Versioning des flux, test unitaire (Camel Test, JUnit)
- Intégration dans GitLab CI / Jenkins / ArgoCD
- Packaging en chart Helm / templates K8s
- Stratégies de rollback et de mise à l'échelle

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.