

# Apache Camel, mise en œuvre

Cours Pratique de 3 jours - 21h

Réf : APC - Prix 2024 : 2 070CHF HT

Dans l'écosystème JEE, Apache Camel est le framework de référence implémentant les patterns d'architecture d'entreprise. Vous aborderez les problématiques d'intégration d'applications d'entreprise. Vous mettrez en œuvre une solution opérationnelle. Vous maîtriserez le suivi d'activités et la gestion des erreurs.

## OBJECTIFS PÉDAGOGIQUES

À l'issue de la formation l'apprenant sera en mesure de :

Identifier les situations justifiant l'usage de pattern EIP.

Maîtriser l'architecture du framework Apache Camel.

Exploiter les principaux composants endpoint d'Apache Camel.

Concevoir des routes efficaces et robustes.

Effectuer des transformations sur les messages.

Réaliser le routage des messages de manière statique et dynamique. Assurer le suivi d'activité des routes.

## TRAVAUX PRATIQUES

Implémentation d'une étude de cas « fil rouge » servant de cadre à l'ensemble des travaux pratiques réalisés.

## LE PROGRAMME

dernière mise à jour : 06/2021

### 1) Les défis de l'intégration d'entreprise

- Les styles d'intégration d'entreprise.
- Architecture orientée services versus orientée événements.
- Le style Messaging.
- Introduction aux EIP.

*Travaux pratiques : Appropriation de l'étude de cas.*

### 2) Le framework Apache Camel

- Camel et les EIP.
- Anatomie des messages.
- Constituants d'une route (exchange, endpoints, processor, router, transformer), modèle Pipe and Filter.
- Implémentation d'une route en java DSL et Spring DSL (XML).
- Travailler le contenu des messages via les langages de script (simple, XPath, SpEL...).
- Composants Endpoint : file, mail, ActiveMQ, JMS, direct.

*Travaux pratiques : Conception de routes : transfert de fichiers vers des queues JMS, diffusion d'informations par mail.*

### 3) Transformer le contenu des messages

- EIP Message Translator : les différentes techniques (Processor, Beans, DSL, Expressions).
- EIP Content Enricher : enrich versus pollEnrich.
- Transformation XML : XSLT, marshaling / unmarshaling.

## PARTICIPANTS

Développeurs JEE, architectes logiciel, chefs de projet SOA.

## PRÉREQUIS

Bonne connaissance des architectures de SI et des protocoles standards de communication. Bonne connaissance du langage Java. La connaissance du framework Spring est un plus.

## COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

## MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

## MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les stages pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
- À l'issue de chaque stage ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
- Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le stagiaire a bien assisté à la totalité de la session.

## MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

## ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Vous avez un besoin spécifique d'accessibilité ? Contactez Mme FOSSE, référente handicap, à l'adresse suivante psh-accueil@orsys.fr pour étudier au mieux votre demande et sa faisabilité.

- Composants prédéfinis de transformation de format (csv, xml, json, zip...).
- Transformation à l'aide de templates (Velocity, Freemarker).
- Concevoir ses propres convertisseurs.

*Travaux pratiques* : Transformer le contenu de messages véhiculés par les routes précédemment créées.

#### 4) Composants endpoint et gateway

- Les composants Camel.
- Communication en mémoire (Direct, SEDA, VM), via des services web (CXF, CXFRS, Restlet), asynchrones (JMS).
- Accès bases de données : JDBC, JPA.
- Activation périodique de routes : Timer, Quartz.
- Utilisation de gateway : CamelProxy et annotations @Consume et @Produce.

*Travaux pratiques* : Intégration d'applications existantes, proxification de services web (SOAP et REST), utilisation de gateway.

#### 5) Routage des messages

- Filtrage et diffusion de messages.
- Routage statique : content-based router, recipientList.
- Routage dynamique : routingSlip, dynamicRouter.
- Load balancing et failover.

*Travaux pratiques* : Implémentation de stratégies de routage, mise en œuvre du load balancing.

#### 6) Suivi d'activités et gestion des erreurs

- Suivi d'activité des routes : Log et Wiretap, la JConsole.
- Erreurs recouvrables ou non, stratégies associées.
- Traitements d'erreurs prédéfinis : DefaultErrorHandler, DeadLetterChannel, LoggingErrorHandler.
- Relivraison après erreur.
- Gestion des exceptions : onException().

*Travaux pratiques* : Ajout de métriques, historisation de messages, gestion d'erreurs.

## LES DATES

---

CLASSE À DISTANCE

2024 : 05 juin, 18 sept., 02 déc.