

Architecture Microservices avec Spring Boot

INFORMATIONS GÉNÉRALES

Type de formation : Formation continue

Éligible au CPF : Non

Domaine : Développement

Action collective : Non

Filière : Java JEE

Rubrique : Serveurs d'applications Java

Code de formation : LEDN203

€ Tarifs

Prix public : 3475 €

Tarif & financement :

Nous vous accompagnons pour trouver la meilleure solution de financement parmi les suivantes :

Le plan de développement des compétences de votre entreprise : rapprochez-vous de votre service RH.

Le dispositif FNE-Formation.

L'OPCO (opérateurs de compétences) de votre entreprise.

France Travail: sous réserve de l'acceptation de votre dossier par votre conseiller Pôle Emploi.

CPF -MonCompteFormation

Contactez nous pour plus d'information : contact@aston-institut.com

PRÉSENTATION

Objectifs & compétences

A l'issue de la formation, le stagiaire sera capable :
Maîtriser l'utilisation de Spring Boot pour la construction de web services

Public visé

Développeurs Java EE

Pré-requis

Maîtriser la programmation orientée objet en Java – Maîtrise Spring Core

Lieux & Horaires

Durée : 35 heures

Délai d'accès : Jusqu'à 8 jours avant le début de la formation, sous condition d'un dossier d'inscription complet

PROGRAMME

Comprendre l'architecture de microservices

Architecture en couches : du monolythe au microservices
Contraintes d'architecture des microservices
Gestion de l'authentification centralisée dans une architecture microservices
Intérêt d'une passerelle d'API
Gestion centralisée des traces

Développement de microservices avec Spring Boot

Galaxie Spring : présentation, apports
Spring Framework : inversion de contrôle et injection de dépendances
Spring Boot : principe, fonctionnalités, pré-requis
Création d'un projet : starters, gestion des dépendances, packaging, exécution, debug
Gestion du rechargement à chaud
Configuration du projet (.properties ou .yaml) et utilisation de profils ou d'une configuration externe
Configuration de Logback pour la gestion des logs (logback.xml)
Organisation des couches du projet : controllers, services, repositories
Intérêt d'une couche de DTOs, utilisation d'un mapper d'objets
Implémentation de contrôleurs REST : mapping global ou spécifique, méthodes, types de retours,
annotations jackson
Gestion des paramètres de méthodes et du mapping
Gestion du download
Gestion de l'upload, configuration
Gestion des services et des transactions associées
Gestion du cross origin et restriction des domaines appelants
Test de l'api REST avec Postman
Ecriture de tâches asynchrones, planification
Cache web

Atelier : Écriture de micro-services avec Spring web - Test des méthodes de services avec Postman ou autre

Documenter une API REST

Open API Specification (Swagger) : présentation, outil
Utilisation de Spring Doc Open API UI
Visualisation avec Swagger Editor

Prochaines sessions

Consultez-nous pour les prochaines sessions.

Documentation du code Java, génération de javadoc

Atelier : Documentation de l'api

Intercepter des requêtes et gérer les erreurs

ControllerAdvice et gestion globale des exceptions

Capture d'exceptions personnalisées (@ExceptionHandler)

Intercepteurs de requêtes/réponses

Atelier : Gestion des exceptions et implémentation d'intercepteurs

Appeler d'autres API REST (écriture de clients)

RestTemplate : présentation, méthodes

Ecriture de requêtes GET, POST, PUT, DELETE - utilisation de la méthode exchange()

Gestion des paramètres et du corps de la requête

Gestion des headers

Gestion des réponses et utilisation d'object mappers

Atelier : Implémentation de clients Java pour un service REST

Gérer efficacement la couche de persistance

Spring Data JPA : apports, mise en place, configurations multiples

Mapping des entités, relations

Gestion de la concurrence : @Version, locking

Ecriture de repositories : requêtes avancées JP-QL, SQL

Repositories personnalisés

Gestion du chargement des collections : lazy vs eager

Configuration du cache : @Cacheable

Mise en place d'une solution d'audit de tables (historique de modifications)

Atelier : Implémentation d'une couche complète de persistance - mise en place d'un cache

Sécuriser un service web

Gestion des données d'entête

Gestion de la sécurité avec Spring Security

Gestion des utilisateurs et des rôles

Atelier : Intégration de Spring Security

Tester une application Spring Boo

Stratégies de tests, types supportés

Configuration de l'application

Mocking des couches de l'application

Tests auto-configurés

Exécution et reporting

Atelier : implémentation et exécution de tests

MODALITÉS

Modalités

Modalités : en présentiel, distanciel ou mixte . Toutes les formations sont en présentiel par défaut mais les salles sont équipées pour faire de l'hybride. – Horaires de 9H à 12H30 et de 14H à 17H30 soit 7H – Intra et Inter entreprise.

Pédagogie : essentiellement participative et ludique, centrée sur l'expérience, l'immersion et la mise en pratique. Alternance d'apports théoriques et d'outils pratiques.

Ressources techniques et pédagogiques : Support de formation au format PDF ou PPT Ordinateur, vidéoprojecteur, Tableau blanc, Visioconférence : Cisco Webex / Teams / Zoom.

Pendant la formation : mises en situation, autodiagnosics, travail individuel ou en sous-groupe sur des cas réels.

Méthode

Fin de formation : entretien individuel.

Satisfaction des participants : questionnaire de satisfaction réalisé en fin de formation.

Assiduité : certificat de réalisation.

Validations des acquis : grille d'évaluation des acquis établie par le formateur en fin de formation.