Formation Spring – Développement d’applications Java JEE

Informations

Durée : 3 jours (21h.)

Tarif* : 1890 € HT
Intra: Nous consulter

Réf : SPR5

Niveau : Moyen

inter à distance / intra

Mise à jour le 03/10/25

*tarif valable jusqu'au 31/12/2025

Prochaines sessions

30 mars - 01 avril
(à distance)

24 août - 26 août
(à distance)

16 novembre - 18 novembre
(à distance)

Pré-requis

- Bonne maîtrise des concepts fondamentaux du langage Java (classes, objets, héritage, interfaces, exceptions)
- Connaissance des concepts de base de la Programmation Orientée Objet
- Compréhension des concepts de base du développement web (requêtes HTTP, réponses, modèle client-serveur)
- Compréhension des concepts de base de l’architecture logicielle et des principes de design (comme le design pattern MVC)
- Expérience avec des frameworks ou des bibliothèques Java comme Hibernate ou d'autres frameworks de persistance (utile mais

Objectifs

Objectifs pédagogiques :

- Créer des applications Web avec Spring MVC
- Créer des Web Api REST FULL
- Mettre en oeuvre l’injection de dépendances avec Spring 5
- Optimiser le paramétrage du conteneur léger de Spring 5
- Accéder à vos données avec Spring JDBC et Spring ORM
- Mettre en oeuvre la programmation par Aspect
- Tester des applications Spring 5
- Sécuriser vos applications avec Spring Security

Objectifs opérationnels :

- Mettre en œuvre les bonnes pratiques de développement d'applications d'entreprise avec Spring

Programme

Jour 1 – Fondamentaux Spring Framework

Introduction

Évolution de Spring : de J2EE à Spring 6.x (Java 21, Jakarta EE)
Concepts clés : Inversion de Contrôle (IoC) et Injection de Dépendances (DI)
Programmation Orientée Aspects (AOP)

Spring Core

Beans et cycle de vie (BeanFactory, ApplicationContext)
Configuration par XML et Java Config (@Configuration, @Bean)
Injection de dépendances (constructeur, setter, champs)
Scopes : singleton, prototype, request, session
SpEL (Spring Expression Language)
Annotations Core : @Component, @Autowired, @Qualifier, @Resource, @Value

Jour 2 – Accès aux données & Web MVC

Accès aux données

Abstraction DAO et gestion des ressources
JDBC via JdbcTemplate
Spring ORM : intégration avec Hibernate
Gestion des transactions (PlatformTransactionManager, @Transactional)
Configuration SessionFactory avec PostgreSQL
Implémentation DAO avec Hibernate et Spring

Spring MVC

Architecture MVC de Spring
Configuration de Spring MVC sans Boot (dispatcher servlet, config Java/XML)
Vues JSP / JSTL
Binding et validation (@Valid, Bean Validation)
Internationalisation (i18n)
Gestion des erreurs et exceptions

non obligatoire)

Annotations Web : @Controller, @RestController, @RequestMapping, @GetMapping, etc.
Création d’une API REST classique avec Spring MVC

Jour 3 – Programmation Réactive, Sécurité & Communication temps réel

Programmation Réactive

Introduction à Spring WebFlux (même sans Boot)
Types réactifs : Mono, Flux
Scheduler, threads et backpressure
Gestion des erreurs et debugging

Spring Security

Mise en place de Spring Security (sans Boot)
Authentification (formulaire, Basic Auth)
Autorisation et sécurisation des routes
Gestion avancée avec ACL

Communication temps réel

WebSocket avec Spring : principe pub/sub
STOMP, SockJS et fallback
Configuration côté serveur et côté client

Tests

JUnit 5 et Spring Test
Tests unitaires et d’intégration (@ExtendWith(SpringExtension.class), @ContextConfiguration)
Profils de test
Tests concurrents et gestion de contexte (@DirtiesContext)