

Développement d'un Backend Java pour Angular

Programme (Mis à jour le 30/01/2026)

Cadrage, architecture cible, projet fil rouge (2h)

- Clarifier contraintes, périmètre, critères de qualité
- Initialiser le repo, modules Maven/Gradle, conventions
- Définir les couches : API, service, persistance, UI
- (ATELIER) Démarrer le squelette du projet fil rouge

Conception en couches et design patterns Java (3h)

- Séparer responsabilités : contrôleur, service, DAO/repository
- Appliquer DTO, mapper, exceptions métier
- Choisir patterns : Factory, Strategy, Template Method
- Travaux pratiques : Refactorer un mini-projet en couches avec DTO

Spring Core : IoC, DI, composants (4h)

- Déclarer beans : @Component, @Configuration, @Bean
- Gérer scopes, cycle de vie, profils Spring
- Injecter dépendances : constructor injection, qualifieurs
- (ATELIER) Câbler services et composants via conteneur Spring

Spring Boot : configuration et packaging (3h30)

- Configurer application.yml, propriétés, @ConfigurationProperties
- Gérer environnements : profils, variables, secrets
- Exposer actuators et logs structurés
- Travaux pratiques : Packager un service Boot et profiler la config

Spring Web MVC : endpoints et validation (3h30)

- Créer controllers REST : @RestController, mappings
- Valider entrées : Bean Validation, messages d'erreur
- Gérer erreurs : @ControllerAdvice, codes HTTP
- (ATELIER) Implémenter CRUD REST avec validation et erreurs normalisées

JPA/Hibernate : mapping et transactions (4h)

- Mapper entités : relations, héritage, contraintes
- Gérer fetch, cascade, lazy loading, N+1
- Piloter transactions : @Transactional, propagation
- Travaux pratiques : Modéliser domaine et persister avec JPA/Hibernate

Spring Data : requêtes et performance (3h30)

- Utiliser repositories, méthodes dérivées, pagination
- Écrire requêtes JPQL, Criteria, projections
- Optimiser : indexes, batch, fetch join
- (ATELIER) Diagnostiquer et corriger un problème N+1

API REST : design et industrialisation (3h30)

- Concevoir ressources : verbes, statuts, idempotence
- Gérer versioning, tri, filtres, pagination
- Documenter : OpenAPI/Swagger, exemples

Référence

THIL3645

Durée

5 jours / 35 heures

Prix HT / stagiaire

2750€

Objectifs pédagogiques

- Structurer une application Java en conception en couches maintenable
- Mettre en œuvre l'injection de dépendances avec Spring Framework
- Implémenter la persistance avec JPA/Hibernate et transactions
- Concevoir et exposer une API REST Spring conforme aux bonnes pratiques
- Consommer une API REST depuis une application cliente Angular
- Appliquer des design patterns courants sur un cas applicatif
- Diagnostiquer des problèmes de performance et de configuration

Niveau requis

- Maîtriser les technologies Web (HTTP, HTML, JavaScript, JSON)
- Avoir une expérience en développement Web Java

Public concerné

- Développeurs Java Web souhaitant structurer des applications d'entreprise
- Chefs de projets techniques pilotant des développements Java/Spring

Formateur

Les formateurs intervenants pour Themanis sont qualifiés par notre Responsable Technique Olivier Astre pour les formations informatiques et bureautiques et par Didier Payen pour les formations management.

Conditions d'accès à la formation

Délai : 3 mois à 1 semaine avant le démarrage de la formation dans la limite des effectifs indiqués

Moyens pédagogiques et techniques

Salles de formation (les personnes en situation de handicap peuvent avoir des besoins spécifiques pour suivre la formation. N'hésitez pas à nous contacter pour en discuter) équipée d'un ordinateur de dernière génération par stagiaire, réseau haut débit et vidéo-projection UHD

Documents supports de formation projetés
Apports théoriques, étude de cas concrets et exercices

Mise à disposition en ligne de documents supports à la suite de la formation

- Travaux pratiques : Spécifier et documenter l'API du projet fil rouge

Qualité : tests, sécurité, erreurs fréquentes (3h)

- Tester services : JUnit, Mockito, tests de slice
- Tester API : MockMvc, tests d'intégration
- Sécuriser : Spring Security, JWT, CORS
- (ATELIER) Ajouter authentification et tests sur endpoints critiques

Angular : consommation API et écrans clés (4h)

- Configurer HttpClient, interceptors, gestion erreurs
- Implémenter services, models, RxJS, async pipe
- Construire UI : routing, guards, formulaires réactifs
- Travaux pratiques : Créer un écran CRUD Angular connecté à l'API

Synthèse, restitution et plan d'action (1h)

- Consolider architecture, décisions, conventions de code
- Restituer le projet : démo, points forts, axes de progrès
- Identifier prochains chantiers : tests, sécurité, performance
- (ATELIER) Checklist de mise en production et rétro-plan technique