+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Mis à jour le 09/10/2025

S'inscrire

Formation Spring Reactor

3 jours (21 heures)

Présentation

Spring Reactor est la bibliothèque réactive de l'écosystème Spring. Basée sur la spécification Reactive Streams, elle fournit des API non bloquantes (Mono et Flux) pour construire des applications asynchrones, scalables et résilientes.

Notre formation Spring Reactor vous permettra de maîtriser la modélisation de flux, l'orchestration et la gestion d'erreurs, tout en intégrant WebFlux, les clients réactifs et l'accès aux données via R2DBC ou MongoDB Reactive.

Vous apprendrez à concevoir, tester et observer des pipelines réactifs performants : Schedulers, backpressure, StepVerifier, traçabilité et intégration CI/CD.

À l'issue de la formation, vous saurez développer des APIs WebFlux robustes, optimiser la latence et le débit, sécuriser vos traitements et migrer progressivement vos services depuis des modèles bloquants.

Comme toutes nos formations, celle-ci couvre la dernière version stable de l'écosystème Reactor et privilégie une approche résolument pratique et opérationnelle.

Objectifs

- Comprendre les principes de la programmation réactive
- Maîtriser Mono, Flux et les Schedulers
- Construire des API Spring WebFlux performantes
- Accéder aux données via R2DBC / MongoDB Reactive
- Tester avec StepVerifier et instrumenter l'observabilité
- Industrialiser : BOM, CI/CD, sécurité et performance

Public visé

- Développeurs Java / Kotlin
- Leads techniques et architectes applicatifs

Pré-requis

- Bonne maîtrise de Java (ou Kotlin)
- Connaissances de Spring Boot et des API REST
- Notions de tests automatisés

Programme de notre formation Spring Reactor

[Jour 1 – Matin]

Fondamentaux de la programmation réactive

- Pourquoi la programmation réactive : I/O non bloquants, élasticité et résilience
- Spécification Reactive Streams : Publisher, Subscriber, Subscription
- Présentation de Project Reactor et de son écosystème
- Différences avec Future/CompletableFuture, RxJava, code bloquant
- Choisir Reactor : cas d'usage et limites
- Atelier pratique : démarrer un projet Maven/Gradle et exécuter un premier flux

[Jour 1 – Après-midi]

Manipuler les flux : Mono et Flux

- Création de Mono et Flux (factory methods, fromIterable, interval)
- Opérateurs clés : map, flatMap, filter, reduce, collectList
- Backpressure et stratégie de demande (request, limitRate)
- Gestion d'erreurs : onErrorResume, onErrorReturn, retryWhen
- Introduction à StepVerifier pour tester les flux
- Atelier pratique : pipelines de transformation et tests unitaires avec StepVerifier

Concurrence et planification

- Schedulers : boundedElastic, parallel, single, immediate
- publishOn vs subscribeOn : propagation des threads
- Cold/Hot publishers, ConnectableFlux, autoConnect
- Combiner des flux : merge, concat, zip, combineLatest
- Debug, checkpoint(), logs réactifs

- Atelier pratique : optimisation d’un pipeline CPU/I/O avec Schedulers

[Jour 2 – Matin]

Spring WebFlux : le framework réactif

- WebFlux vs Spring MVC : servlets vs Netty, non-blocant de bout en bout
- Contrôleurs annotés et fonctionnels (RouterFunction, HandlerFunction)
- Client réactif WebClient : timeouts, retry, backoff
- Sérialisation Jackson réactive et validation
- Gestion des SSE et WebSocket
- Atelier pratique : créer une API REST réactive + client WebClient

[Jour 2 – Après-midi]

Accès aux données réactif

- R2DBC : drivers réactifs SQL, transactions et limites
- MongoDB Reactive : flux réactifs, tailable cursors
- Patterns de repository réactifs, pagination et backpressure
- Cache réactif et Circuit Breaker (ex. Resilience4j)
- Stratégies de propagation de contexte (tracing, sécurité)
- Atelier pratique : brancher WebFlux à R2DBC ou Mongo Reactive

Patterns d’architecture réactive

- Fan-out/Fan-in, timeout, fallback, bulkhead
- Idempotence et réessais en flux
- Traitement temps réel : SSE, WebSocket, Kafka (reactor-kafka)
- Backpressure bout-en-bout : du client à la base
- Observabilité : Micrometer, Tracing, zipkin/otel
- Atelier pratique : service temps réel (WebSocket + backpressure)

[Jour 3 – Matin]

Qualité, test et industrialisation

- Tests unitaires et test scheduler, virtual time
- Tests slice WebFlux, WebTestClient
- Contrats, Testcontainers (Mongo/R2DBC)
- BlockHound : détection d’appels bloquants
- Intégration CI/CD, BOM Reactor, versioning
- Atelier pratique : pipeline CI avec tests réactifs et BlockHound

[Jour 3 – Après-midi]

Performance, tuning et sécurité

- Mémoire, GC, tailles de batch, prefetch
- Tuning Netty (pools, timeouts), reactor.netty basics
- Rate limiting, circuit breaker, hedging
- Sécurité réactive avec Spring Security (authN/authZ non-bloquantes)
- Profiling & replays (JFR et événements réactifs)
- Atelier pratique : profiling d’une API WebFlux et remédiations

Migration, bonnes pratiques et runbook

- Migrer de Spring MVC/RxJava vers WebFlux/Reactor
- Anti-patterns : block(), subscribe() sauvage, parallel() abusif
- Guidelines de revue de code réactif, checklist de PR
- Runbook : SLO/SLI, timeouts, budgets d’erreur
- Stratégies de feature flags et déploiement canari
- Atelier pratique : plan de migration d’une API MVC vers WebFlux

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.