

# Formation Architecture logicielle Avancée pour développeurs Java

<b>Durée :</b>	5 jours
<b>Public :</b>	Développeurs Java expérimentés, Architectes
<b>Pré-requis :</b>	Expertise en développement Java Spring - Maîtrise de la programmation orientée objet et des concepts SOLID
<b>Objectifs :</b>	Mettre en œuvre des architectures microservices, orientée événements - Apprendre les différents anti-patterns d'architecture
<b>Sanction :</b>	Attestation de fin de stage mentionnant le résultat des acquis
<b>Taux de retour à l'emploi:</b>	Aucune donnée disponible
<b>Référence:</b>	ARC101593-F
<b>Note de satisfaction des participants:</b>	4,92 / 5

## Se rappeler des fondamentaux

Architecture logicielle : définition, documentation, vues architecturales  
Démarche de production d'un dossier d'architecture technique  
Panorama de composants logiciels et contraintes d'architecture  
Styles d'architecture et organisation des couches d'une application

## Concevoir une application basée sur les microservices

Décomposition fonctionnelle et Domain Driven Design  
Exigences de conception des microservices : réactivité, compatibilité rétroactive, communication flexible, idempotency, fonctionnement efficace  
La méthodologie à 12 facteurs  
Structure d'un microservice, organisation des couches  
Utilisation des événements au sein d'un microservice

### Atelier : Implémenter un micro-service avec Spring Boot, Web et Data JPA

## Utiliser un message broker

Middleware Orienté Messages : intérêt, modes, protocoles  
Architecture microservices orientée événements  
Panorama de messages brokers et comparaison

### Atelier : Implémentation d'une architecture microservices orientée messages (producer/consumer) avec Spring AMQP ou JMS

## Implémenter des WebSockets

WebSocket : le standard, intérêt et contextes d'utilisation  
Schéma de communication des WebSockets  
APIs disponibles

**Atelier : Implémentation de web sockets avec Spring WebSocket et utilisation du protocole STOMP**

## Gérer efficacement des logs

Démarche globale de gestion des traces : support, type, quoi stocker  
Interfaçage d'un microservice avec une solution de logs : ELK ou Graylog  
Réflexion d'architecture : log direct, log via message broker, log direct vers fichier puis agrégation via un outil

**Atelier : Configuration d'un microservice pour tracer en direct vers Elasticsearch, ou vers un message broker puis consommation via Logstash**

## Anti-patterns d'architecture logicielle

Anti-patterns : définition, détection

Panorama d'anti-patterns :

- Autogenerated Stovepipe
- Stovepipe Enterprise
- Jumble
- Stovepipe System
- Cover Your Assets
- Vendor Lock-In
- Wolf Ticket
- Architecture by Implication
- Warm Bodies
- Design by Committee
- Swiss Army Knife
- Reinvent the Wheel
- The Grand Old Duke of York