

Plan intervention

Pré-requis d'installation :

- JDK 11
- IDE
- Git
- Docker et docker-compose

Tous les supports (Présentations, Énoncés d'ateliers , et solutions) seront fournis sous forme de dépôt github.

Table des matières

Jour 1 (présentiel) : Etat des lieux.....	2
Environnement de développement.....	2
Gestion des sources.....	2
Pilotage projet.....	2
Qualité :.....	2
Releasing.....	2
Tests d'intégration.....	2
Déploiement.....	2
Surveillance et observabilité.....	2
Synthèse.....	3
Jour 2,3,4 (présentiel) : Architecture micro-services.....	3
Exemple micro-service SpringBoot3.....	3
Design patterns pour les micro-services.....	3
Introduction au architecture micro-services.....	3
Stratégies de décomposition.....	3
Configuration externalisée et centralisée.....	3
Interactions entre services.....	3
Cohérence des données et transactions.....	4
Logique métier.....	4
Requêtage.....	4
API Externe.....	4
Sécurité.....	4
Observabilité.....	4
Jour 5 (présentiel).....	4
Identification du POC BCEAO.....	4
Définition de l'architecture et APIs.....	4
Répartition des groupes projet.....	4
Jour 6 et 7 et 8 à 11: POC.....	4

Jour 1 (présentiel) : Etat des lieux

Environnement de développement

- IDE, plugins
- Outil de build
- Stack technologique, Frameworks
- Pratique des tests unitaires
- Cycle Code/ Deploy / Test

Gestion des sources

- Dépôt utilisés, branches,
- workflow de collaboration

Pilotage projet

- Planification et Suivi des travaux
- Implication du métier

Qualité :

- Revue de code
- Analyse statique qualité
- Analyse de vulnérabilité

Releasing

- Processus de release
- Packaging
- Dépôts d'artefacts

Tests d'intégration

- Environnement d'intégration
- Types de test d'intégration

Déploiement

- Relation avec les gestionnaire d'infra
- Procédure de déploiement
- Gestion des migrations de schéma

Surveillance et observabilité

- Mécanismes de monitoring
- Statistiques d'utilisation
- Feed-back utilisateurs finaux
- Gestion des issues

Synthèse

- Document de synthèse
- Identification d'une application BCEAO typique et objectifs d'évolution : Migration socle 3, micro-service ?

Jour 2,3,4 (présentiel) : Architecture micro-services

Exemple micro-service SpringBoot3

Objectifs : Présenter les nouveaux starters et des pratiques de développement dans l'environnement SpringBoot 3. Insister sur les problématiques de l'environnement de développement d'un microservice avec dépendances

L'exemple est une architecture micro-service inspiré de l'existant BCEAO supposé

3 micro-services font partie de l'architecture :

- Reactive Mongo
- Impératif PostgreSQL
- Gateway

Pour chaque projet, le code source, les tests, le packaging sont commentés. Les stagiaires exécutent le projet sur leur poste

Design patterns pour les micro-services

Support classique avec slides et ateliers pédagogiques

Introduction au architecture micro-services

Objectifs, Bénéfices attendus, Contraintes

Problématiques et Design patterns

Services transverses : Framework ou Infra

Stratégies de décomposition

Business Capacity Pattern

Subdomain Pattern

Définition de l'API

Configuration externalisée et centralisée

Solutions

Interactions entre services

Introduction

Interactions RPC
Messaging

Cohérence des données et transactions

Saga Pattern

Logique métier

Transactional Script Pattern
Patterns orienté objet
Event Sourcing Pattern

Requêtage

API Composition Pattern
CQRS Pattern

API Externe

Gateway Pattern

Sécurité

- Stratégies
- Sécurisation via oAuth2
- Propagation de Jeton
- Client-credentials

Observabilité

- Health
- Métriques
- Tracing

Jour 5 (présentiel)

Identification du POC BCEAO

Définition de l'architecture et APIs

Répartition des groupes projet

Jour 6 et 7 et 8 à 11: POC

Démarrage de projet, organisation des tâches

Choix des starters