



**Ingénieur développement full-stack**  
Confirmé Java / Certifié Kafka / Certifié Liferay DXP  
**8 ans d'expériences**

**INFORMATIONS**

<b>Nom Prénom</b>	BENMOHAMED Charfeddine
<b>Email</b>	charfeddinebenmohamed24@gmail.com
<b>Numéro</b>	(+33) 06 16 089 279

**COMPETENCES TECHNIQUES**

<b>Langages</b>	JAVA SE 8/17/21 , HTML, Typescript, Python (basic)
<b>Architecture</b>	MICROSERVICE, Monolithic
<b>Frameworks</b>	HIBERNATE, JPA, SPRING, SPRING BOOT, ANGULAR (JS/2+), OpenAPI, Kafka, Apache camel

<b>SGBDR</b>	MYSQL, PostgreSQL, MONGODB, CoucheBase, GreenPlum
<b>Modélisation</b>	UML, DESIGN PATTERNS
<b>IDE</b>	ECLIPSE 4.3, IntelliJ IDEA, Visual Studio code , DBeaver
<b>Serveurs d'applications</b>	TOMCAT, JBoss
<b>Intégration continue</b>	MAVEN 3, JENKINS, SONAR , CloudBees, GitLab
<b>Gestion de versions</b>	SVN, GIT
<b>Tests unitaires</b>	JUNIT, MOCKITO
<b>Systèmes</b>	WINDOWS, LINUX
<b>Méthodologies</b>	AGILE SCRUM
<b>Outil de gestion de projet</b>	JIRA
<b>Web services</b>	SOAP, REST Outils
<b>DevOps</b>	DOCKER, K8S, IBM Cloud, GCP

**DIPLOMES**

**2018** Diplôme Ingénieur Logiciel

**2015** Diplômé En Électronique Electrique, Automatique

**2012** Baccalauréat en sciences technique

**LANGUES**

**Anglais : B2 (Certificate iCIMS )**

**FORMATIONS**

Formation en l'intelligence artificielle de programmation avec IBM Watson.

**CERTIFICATION**

Nom prénom : BENMOHAMED Charfeddine Clé de certificat : 22312403  
Nom de certificate : Liferay DXP 7.2 - Certified Professional Back-End Developer Final Version  
Date de certificat : 30 juin 2021  
Lien de vérification : <https://www.liferay.com/fr/services/certification/verify-a-certification>

Nom prénom : BENMOHAMED Charfeddine  
Nom de certificate : Confluent Certified Developer for Apache Kafka (CCDAK)  
Date de certificat : 12 août 2025

## EXPERIENCES PROFESSIONNELLES

MGEN

De JAN 2025

Ingénieur développement JAVA – KAFKA – APACHE CAMEL

---

### Contexte :

Intégré à l'équipe Data de la MGEN, j'ai contribué à un projet structuré autour de trois volets majeurs :

- Développement d'APIs REST en environnement microservices pour exposer les données aux courtiers et adhérents.
- Mise en place d'un traitement batch avec Apache Camel (routes, processors, EIP, data transformation) pour intégrer, transformer et charger de larges volumes de données issus du data Warehouse vers une base NoSQL (Couchbase), afin d'alimenter directement les APIs.
- Implémentation d'Apache Kafka pour la messagerie et le streaming de données : en mode producteur pour diffuser des événements vers d'autres systèmes, et en mode consommateur pour traiter des flux entrants (ex. mises à jour clients en temps réel) garantissant une synchronisation fiable, une intégrité renforcée et une cohérence optimale des données clients dans tout l'écosystème d'information.

### Réalisation :

- Développement de microservices et d'APIs REST avec Spring Boot et Java 21.
- Conception et implémentation de traitements batch Apache Camel avec gestion de routes, processors, filtrage, transformation et intégration de greeplum vers Couchbase.
- Mise en œuvre de Kafka en mode messaging (producer et consumer) pour échanger des événements métier critiques avec d'autres systèmes, assurant la diffusion et la réception fiables d'informations.
- Utilisation de Kafka en mode streaming pour traiter en continu les flux de mises à jour clients (ex. changement d'adresse ou modification de contrat) et enrichir automatiquement la base NoSQL en temps quasi-réel.
- Application d'un profilage avancé sur le batch à l'aide de JProfiler afin d'identifier et corriger les goulots d'étranglement, optimisant ainsi les temps de traitement.
- Génération automatique de contrats de service avec OpenAPI.
- Orchestration et déploiement des applications via Kubernetes et Docker.

- Utilisation des services GCP pour le déploiement, le suivi des logs applicatifs, la supervision et le dépannage en environnement cloud.
- Gestion des configurations et déploiements avec CloudBees.
- Automatisation des pipelines CI/CD avec GitLab et CloudBees.
- Rédaction de documentation technique.

#### **Environnement Technique :**

- **Langages** : Java SE 21, SQL
- **Framework Back-End** : Spring (Boot, Data, Boot Admin, Cloud), Apache Camel  
Messagerie & Streaming : Apache Kafka
- **Base de Données** : NoSQL (Couchbase), PostgreSQL
- **Architecture** : Microservices
- **DevOps** : Docker, Kubernetes, GCP
- **CI/CD** : GitLab CI, CloudBees, Jenkins, Sonar, Maven 3
- **Outils de Développement** : Git, JIRA, IntelliJ, DBeaver
- **Méthodologie Agile (SCRUM)** : Planification, réunions quotidiennes, grooming, revues, rétrospectives, scoring, sprints

**BNP Paribas Cardif**

**De Oct 2021 à Déc 2024**

**Ingénieur développement Full stack JAVA EE - ANGULAR 14**

---

#### **Contexte :**

Participation au développement d'applications en marque blanche pour des assurances, servant d'outil d'aide à la vente pour divers types de contrats d'assurance prévoyance destinés aux clients et conseillers.

#### **Réalisation :**

- Développement de la partie Back-end et mise en place de microservices avec Spring Boot.
- Configuration centralisée de la partie Back-end via Spring Cloud Config.
- Développement et implémentation de tests unitaires pour l'ensemble de l'application de la partie Back-end.
- Génération automatique de contrats de service avec OpenAPI (Back-end/Front-end).
- Développement d'APIs RESTful dans la partie Back-end.
- Implémentation de Kafka comme sender/receiver pour les messaging queues afin d'assurer une communication asynchrone et scalable entre les microservices.
- Développement d'un module de collecte de données, incluant un algorithme en Python pour optimiser le temps de traitement. Utilisation de Jython pour assurer la communication et l'interopérabilité entre Java et Python.
- Utilisation de l'outil de profilage Java JProfiler pour analyser et optimiser les performances de la partie Back-end.
- Mise en place d'un tableau de bord de monitoring incluant des comparateurs de versions de toute la partie Back-end.
- Gestion des appels externes via des web services (DXC, Magnum, etc.).
- Développement de la partie Front-end en Angular 14.
- Intégration de services Front-end avec GraphQL.

- Déploiement des applications sur un cloud privé (IBM).
- Gestion des configurations et des déploiements avec Kubernetes.
- Automatisation des processus CI/CD avec Jenkins.
- Rédaction de la documentation technique.

#### **Environnement Technique :**

- **Langages:** JAVA SE 1.8 / 11 / 17, TypeScript, Python
- **Framework Back-End:** Spring (Boot, Data, Boot Admin, Cloud, Web Services)
- **Framework Front-End:** Angular 14
- **Web Services:** RESTful API, SOAP
- **Architecture:** Microservices
- **DevOps:** Docker, Kubernetes
- **SGBDR:** MySQL
- **Intégration Continue :** Sonar, Maven 3, Jenkins
- **Outils de Développement :** Git, JIRA, IntelliJ, Visual Studio Code
- **Méthodologie Agile (SCRUM) :** Planification, réunions quotidiennes, grooming, revues, rétrospectives, scoring, sprints

**Tessi**

**De Fév 2017 à Août 2021**

**Ingénieur développement Full stack JAVA EE- ANGULAR 8/10 - Liferay**

---

#### **Contexte :**

1. Développement d'une application de feuille de temps pour enregistrer et suivre les heures de travail des développeurs sur différentes tâches, avec intégration des données vers une application de gestion de projet pour un suivi optimisé.
2. Le développement des applications marque blanche pour le compte de tiers (assurances, mutuelles, courtiers...) :
  - Développement pour les contrats collectifs d'assurances d'un processus d'affiliation en ligne d'un salarié en temps réel avec reprise.
  - Développement d'une application web avec 3 accès (Assuré, Entreprise, Agent ou Courtiers) permettant de faire les réclamations, les demandes de devis, les remboursements, etc. les en envoyer vers une application GED existante.
  - Développement d'une application web comme un outil aide à la vente pour tous ce qu'est contrat d'assurance (SANTE, PREVOYANCE).
  - La création d'une application SPRING boot admin permet de regrouper tous les retours métriques d'un ensemble des applications en une interface unique afin de pouvoir moniturer le tout en temps réel.
3. La migration de l'environnement de développement de LIFERAY 6.3 vers LIFERAY 7.3 et la documentation de cette réalisation.

#### **Réalisation :**

- La mise en place d'une partie Back-End avec l'approche MAVEN Modules.
- La mise en place d'une partie Front-End avec l'approche modulaire.
- La mise en place d'un projet web API.
- La mise en place d'un projet Batch.
- La mise en place d'un projet SPRING boot admin.
- La mise en place de la couche de sécurité.
- La mise en place d'une connexion SSO via un JWT.
- La mise en place de la communication client avec le projet SPRING boot admin.
- Développement des RESTFUL APIs de la partie Back-end.
- Développement des portlets MVC avec JAVA, JSP, JAVASCRIPT et JQuery.
- Développement des Jobs batch (Mise à jour les références des contrats, Validations des contrats, transfères des contrats signés, extractions des fichiers csv, synchronisation des statuts des contrats avant et après la signature d'assuré, etc.).
- La mise en place d'une communication SOAP avec GED.
- La mise en place d'une communication SOAP avec un générateur des documents en utilisant JAXB et Marshaller.
- La mise en place d'une communication avec un système de payement (Sherlock's, Crédit Agricole).
- La mise en place d'une communication avec un système de signature en ligne (TessiSign)
- La mise en place d'un API Java pour se connecter avec un PDF interactif formulaires que j'ai créé avec Adobe DC Pro.
- La mise en place l'outil de profilage Java JProfiler afin de fixer des fuites mémoire sous l'environnement PROD.
- La mise en place d'une couche de transfère d'objets par DOZER.
- La mise en place d'une couche de transfère d'objets par MAPSTRUCT.
- La mise en place des tests unitaires avec JUNIT.
- Documentation Technique.
- Projet mené en méthodologie Agile (Daily meeting, sprint, user stories, rétrospectives).
- La mise en production.

### **Environnement Technique :**

- Langages : LIFERAY 6.3 / 7.2, JAVA 8, TypeScript
- Framework Back-End: Spring (Boot, Data, Security, Batch, Boot Admin, Cloud)
- Framework Front-End: JSP, JAVASCRIPT, JQuery, ANGULAR 8
- Protocole de sécurité : Ouath2.
- Web Services: RESTFUL API, SOAP.
- Architecture : Monolithique.
- SGBDR: MYSQL, No SQL (Mongo DB).
- Intégration continue : SONAR, MAVEN 3
- Outils de développement : GIT, JIRA, IntelliJ, Visual Studio Code, WINSCP, MobaXterm.
- Agile (SCRUM): planning, daily meeting, grooming, review, retrospective, scoring, sprint