

# DEVELOPPEUR FULL STACK JAVA SPRING / ANGULAR

## Compétences, technologies & secteurs d'intervention

<b>Compétences Techniques Transverses</b>	IHM, Développement Logiciel temps réel et embarqué, Test logiciel, REST API, Intégration frontend-backend, CI/CD, Utilisation de Linux, Sécurité applicative
<b>Compétences Méthodes &amp; Support</b>	Documentation technique, Méthodes de développement Agile (SCRUM), CI/CD avec Jenkins, Gestion de projet technique
<b>Compétences d'Activité Transverses SI &amp; Réseaux</b>	Architecture, Conception, Intégration, Déploiement sur serveurs distants Linux, Administration, Exploitation systèmes et réseaux, Sécurité des API avec JWT

<b>Retail et Services</b>	E-commerce / plateforme de vente de produits
---------------------------	--

<b>Logiciels</b>	Apache, Git, GitLab, Hibernate, Jenkins 5.2.11, JQUERY, Maven, MySQL, Web Services, Docker, PostgreSQL 16, IntelliJ, VS Code, Postman, FTP, Swagger, GitHub, pgAdmin 4
<b>Langages</b>	Dart 3.2.3, TypeScript 5.4.5, CSS, SCSS, Java 17, Java 11, JavaScript, SQL 2016, JSON, HTML5, Bash 5.2, PHP, C 8.3, Python 3.12
<b>Technologies</b>	Swagger, Postman, Azure, AJAX, Bootstrap, Jpa, Rest, Spring, pgAdmin 4, Spring Boot 3.3.5, Laravel, Flutter, Junit, Swing, Angular 18, PrimeNG
<b>Méthodes</b>	Agile SCRUM, CI-CD
<b>Systèmes</b>	Windows, Linux
<b>Normes et protocoles</b>	API, HTTP, JWT, HTTPS, SSH, FTP

# Formation, habilitations & langues

Formation	École supérieure polytechnique de Dakar (ESP) – Diplôme d'ingénieur de conception en informatique	
Langues	Français - Maîtrise Anglais - Autonome	

### Test technique client final

17/05/2025 - 21/05/2025

(5 jours)

**Contexte général:** Ce projet a été réalisé dans le cadre d'un processus de recrutement pour le client final **Alten**, spécialisé dans le conseil en ingénierie et en informatique.

L'objectif de l'entreprise dans ce contexte était d'évaluer la capacité des candidats à concevoir, développer et sécuriser une application complète en autonomie, en simulant un environnement de production Agile.

Le projet visait la mise en œuvre d'une plateforme e-commerce de démonstration, permettant de couvrir l'ensemble des aspects techniques d'un développement full stack moderne, notamment la sécurisation des API avec JWT, l'intégration frontend-backend, et le respect des bonnes pratiques de développement logiciel.

Le principal enjeu était de livrer une application fonctionnelle, documentée et sécurisée, tout en respectant un cahier des charges précis intégrant une restriction d'accès aux fonctionnalités d'administration.

**Sujet du projet :** Le projet consistait à concevoir et développer une mini-application e-commerce full stack dans le cadre d'un test technique pour le client final **Alten**.

L'objectif était de créer une API REST sécurisée par authentification permettant de gérer des produits, un panier et une liste d'envies (wishlist). Ces fonctionnalités devaient être restreintes à un utilisateur administrateur identifié par l'email admin@admin.com.

Côté frontend, l'application Angular devait permettre l'affichage des produits, l'ajout au panier, la suppression, le filtrage, l'ajustement des quantités ainsi que l'envoi d'un formulaire de contact avec validation des champs.

Le projet a été développé en autonomie en suivant les principes de l'architecture RESTful et en intégrant des bonnes pratiques de sécurité et de documentation (Swagger).

Il s'agissait d'un développement from scratch, avec un périmètre fonctionnel clairement défini et des livrables attendus pour l'évaluation technique d'un poste de développeur full stack Java/Angular.

**Responsabilités occupées :** Développeur Full stack Java Spring Boot / Angular

#### Travail réalisé :

##### ❖ Stratégie de tests.

Pour garantir la conformité fonctionnelle de l'API et la robustesse de l'application, une stratégie de tests manuels a été mise en œuvre principalement via Swagger.

Des tests fonctionnels ont été réalisés sur l'ensemble des routes de l'API pour valider les opérations CRUD sur les produits, ainsi que la gestion du panier, de la wishlist et du système d'authentification JWT.

Des tests d'intégration ont également été simulés pour vérifier la cohérence des interactions entre les différents modules backend (authentification, sécurité, persistance des données), les tests ont permis de vérifier:

- La validité des tokens et la sécurisation des routes sensibles (admin uniquement)
- La gestion des statuts HTTP en cas d'erreurs (400, 401, 403, 404)
- Le bon enchaînement des actions (connexion > ajout au panier > récupération du panier, etc.)
- L'automatisation des tests n'a pas été mise en place dans ce contexte, le projet étant réalisé sur une durée courte. Toutefois, les endpoints sont documentés via Swagger, facilitant la mise en œuvre de futures campagnes de tests automatisés.

## ❖ Développement

Le projet a été développé from scratch, avec une architecture full stack Angular/Spring Boot.

Méthodologie : Développement en mode Agile avec itérations courtes et validation incrémentale des fonctionnalités.

Bonnes pratiques mises en œuvre :

- Respect des principes SOLID dans la structuration du code backend
- Mise en œuvre du design pattern Builder pour la création d'entités
- Utilisation de TDD partiel (vérification manuelle via Postman pour chaque module développé)
- Séparation claire des couches Controller / Service / Repository
- Gestion centralisée des erreurs avec @ControllerAdvice et handlers personnalisés

## ❖ Versions et frameworks utilisés :

- Java 17 avec Spring Boot 3.3.5, Spring Security, Spring Data JPA
- Angular 18 avec TypeScript 5.4.5
- ORM : Hibernate / JPA
- Base de données : PostgreSQL 16
- Frontend : PrimeNG, Bootstrap

## ❖ Fonctionnalités développées :

- Backend : API REST sécurisée avec JWT, gestion des produits, panier et wishlist
- Authentification : création de compte et login via tokens JWT
- Rôle admin (basé sur l'email) pour restreindre certaines actions sensibles
- Frontend : interfaces pour boutique, panier, contact, routage, pagination, filtrage
- Validation des formulaires, gestion des quantités dynamiques, UI réactive

## ❖ Outils utilisés :

- Tests manuels avec Postman
- Documentation dynamique avec Swagger 2.2.0
- Gestion de version : Git / GitHub
- Environnements de développement : IntelliJ, VS Code

## ❖ Rédaction de la documentation des API

- Utilisation de l'outil SWAGGER 2.2.0

## Résultats obtenus

- ❖ Application fonctionnelle, sécurisée, documentée et conforme aux spécifications.
- ❖ Couverture fonctionnelle à 100% des besoins exprimés

<b>Environnement technique</b>	<b>Logiciels</b>	Git, IntelliJ, VS Code, Postman, Swagger, GitHub
	<b>Langages</b>	SCSS, Java 17, JavaScript, SQL2016, TypeScript 5.4.5, HTML5
	<b>Technologies</b>	Swagger 2.2.0, Postman, Angular 18, Jpa, Rest, Spring boot 3.3.5, PrimeNG
	<b>Systèmes</b>	Windows 11