

GoMind

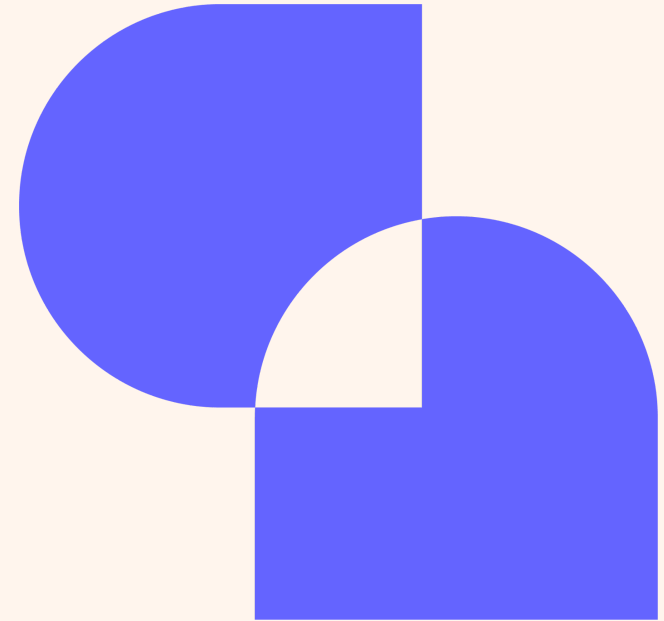
“Agir avec l'esprit éveillé”

DOSSIER DE COMPÉTENCES

Développeur Sénior Angular

Autres expertises : NestJS, Java, Kafka,
Kubernetes

8 ans d'expérience



PRINCIPALES EXPÉRIENCES

MONDIAL RELAY : Lead dev Angular/NestJS/Java/ DevOps (2023-2025)

COINSHARES : Développeur Sénior
Angular/Java/NestJS/DevOps (2021-2023)

TRANSFONUM : Développeur Angular/NestJS (2018-2021)

CONNAISSANCES TECHNIQUES

Langages : Java (8 à 21), Typescript, Javascript, HTML/CSS

Librairies/framework :

Front : Angular (V 11, 16, 17, 19), Material, Tailwind, RxJS, Ionic, ElectronJS, Signals, NgRx, NGXS, Webpack, Gulp, Highcharts, MomentJS, Full Calendar, Swiper, Slick, Storybook, JQuery, Bootstrap, LeafletJS, OpenStreetMap,

Back : ExpressJS, NestJS, Websocket, TypeScript, Kafka, Spring (Boot, MVC, DATA, Security, Batch), JWT, JPA, Hibernate, RabbitMQ, Apache Airflow, Fastify,

Architecture/Design Pattern : BFF, MVC, Microservices, Microfrontend, Architecture en layer couches, Architecture hexagonale

Outils/Devops : Kubernetes, Elasticsearch, Logstash, Kibana, AWS (Lambda, S3, EC2, Cloudwatch), GCP, Firebase,

Git, Gitflow, Gitlab, Github, Jenkins, Docker, Helm, Terraform, Ansible, Swagger, Redis, Tensorflow js, Mljs

Bonnes pratiques : TDD, BDD, DDD, KISS, SOLID, Pair Programming, Code Review, DRY, YAGNI, Optimisation SEO, ACID compliance, Singleton, Injection de dépendance (DI), Observer, Strategy, Decorator, Composite, Facade, Factory, Command,

Tests : Junit, Cypress, Cucumber, Mockito, Jest, Sonar, Gherkins

Base de données : Redis, MongoDB, Elasticsearch PostgreSQL, MySQL, MariaDB, Hibernate, TypeORM SQL, NoSQL, CouchDB / PouchDB

Gestion de projet : Agile, Jira, Kanban, Scrum

EXPÉRIENCES TECHNIQUES

MONDIAL RELAY : Lead dev Angular/NestJS/Java
(2023-2025)

Refactoring de la plateforme de la facturation en Angular 16
et Java 21 et NetsJS

Workflow :

- Analyse fonctionnelle
- Conception fonctionnelle et technique
 - Création des diagrammes de classes
 - Réunion fonctionnelles avec le métier
 - Pilotage des réunions d'affinage de la solution
- Suivi des spécifications fonctionnelles
 - Intégration des feedbacks
- Gestion de projet en Kanban
- Équipe de 6 personnes
 - Cérémonies : daily, démo, rétro
 - Participation aux Chiffrage
 - Rédaction des US (defintion of done / defintion of ready)
- Encadrement de développeurs juniors :
 - Montée en compétences sur Angular et sur le back
 - Pair programming sur certaines features
 - Code review
 - Participation aux recrutements de l'équipe (entretiens techniques)
- Mise en place d'une architecture Microservice
 - Mise en place de SOLID
 - Utilisation de SOLID, KISS, YAGNI
 - Utilisation de la méthode BDD avec Gherkins
 - Suivi de la clean architecture
 - Migration de l'architecture 3tiers vers le DDD
 - Découpage service monolithe en microservices
 - Mise en place pattern : Singleton, Injection de dépendance (DI), Observer, Factory
- Développement Angular :
 - Opération de CRUD
 - Création de Reuse Component pour encapsuler les fonctionnalités spécifiques et l'UI des éléments
 - Mise en place des Ife cycle (ngOnInit, ngOnChanges, ngOnDestroy)
 - Utilisation de TrackBy avec NgFor Directive
 - Utilisation des NG-Modules
 - Écriture de composants dynamiques avec les Dialog
 - Utilisation du SDK d'Angular Material
 - Utilisation des Reactive Forms
 - Utilisation de projection de contenu (Ng-content)
 - Utilisation d'HTTP d'Angular (Get, Post, Put)
 - Développement de tableaux de bord Angular pour les partenaires logistiques afin d'améliorer l'efficacité de la résolution des litiges.
 - Gestion en temps réel avec Socket.io

- Mise en œuvre de la gestion des états NGXS pour les flux de facturation complexes.
- Optimisation des performances d'Angular grâce à l'implémentation du lazy loading et avec Onpush Change Detection
- Utilisation des Async Pipe pour les Observer
- Utilisation des pipes dans les templates
- Utilisation des directives d'attributs et structurelles dans les templates
- Mise en place du routing
- Utilisation de Guards dans la gestion du routing
- Utilisation de RxJS pour gérer la réactivité
 - Utilisation de mapping operator comme mergeMap, concatMap, or exhaustMap
- Utilisation du Subscription/Unsubscription Pattern
- Mise en place des "types"
- Conversion de l'application en PWA
 - Gestion du caching avec Angular/PWA
- **Développement Backend Java & NestJS :**
 - Utilisation de TypeScript
 - POO
 - Typages des classes
 - Développement et maintenance des microservices Spring Boot (Java 21)
 - Migration des services Java 8 vers Java 21, en résolvant des problèmes de compatibilité.
 - Utilisation de toute la suite Spring CORE (batch, security, data...)
 - Construction d'API REST robustes avec Spring Boot, capables de gérer des pics de trafic.
- Développement microservice en NestJS pour un système de facturation à haut volume et la gestion en temps réel
- Gestion de l'authentification
- Développement d'algorithmes de réconciliation pour détecter les divergences de facturation.
- Optimisation des requêtes PostgreSQL par le biais d'une indexation pour optimiser les performances de la génération de factures.
- Implémentation de flux d'événements Kafka pour permettre des mises à jour de factures en temps réel.
 - Design pattern utilisé Topic et Consumer
 - Gestion de 20 000 messages/minutes
 - Gestion des erreurs sur les queues
 - Ajouts de nouveaux topics
 - Gestion de la communication entre les microservices

- Mise en place de la mise en cache Redis pour réduire la charge de la base de données.
 - Mise en place de collections MongoDB
 - Documentation API via Swagger
- Versioning avec Git
 - Suivi du Gitflow
- Mise en place de tests unitaires avec Jest
 - Développement de tests d'intégrations
- Implémentation de suites de tests automatisés avec Cypress pour la gestion de flux de données
- Création des variables d'environnement via Ansible
- Automatisation du déploiement via Gitlab CI
- Mise en place de Docker
- Mise en place de Kubernetes
 - Migration de Bare Metal vers AWS
- Gestion AWS :
 - Utilisation d'AWS secret pour la gestion des données sensibles
 - Utilisation de RDS pour les bases de données
 - Utilisation des Lambda
 - Utilisation de cloudWatch
 - Utilisation de S2 & S3
- Documentation avec Confluence

Environnement technique :

Microservice, SOLID, Angular, NGXS, NestJS, Websocket, Java 21, Spring Boot, Spring CORE, PostgreSQL, Kafka, Rest, GitlabCI, Gitflow, Docker, Kubernetes, Redis, Ansible, BDD (Gherkins), TDD, AWS,

COINSHARES : Développeur Senior Angular/NestJS/DevOps (2021-2023)

Développement from scratch d'une plateforme de trading automatisé en Angular 11, puis migration vers 15

Workflow :

- Analyse fonctionnelle
- Suivi des spécifications technique et fonctionnelles
- Gestion de projet en Scrum
 - Participations aux cérémonies Safe (Daily, démo, rétro , Pi Planning....)
 - Sprint de deux semaines
- Découpage et planification des US techniques
 - Ticketing avec JIRA
- Équipe de 5 personnes
- Encadrement de développeurs juniors
- Mise en place d'une architecture Microservice :
 - Mise en place de SOLID
 - Utilisation de la méthode BDD avec Gherkins
 - Suivi de la clean architecture
 - Mise en place pattern : Singleton, Injection de dépendance (DI), Observer, Strategy, Decorator, Composite, Facade, Factory, Command, Orientée aspect (AOP)
 - Documentation sur Confluence
 - Création et implémentation du design system avec Story book
- Développement Angular :

- Mise en place Clean architecture sur le développement des services/modules
- Suivi du SRP
- Opération de CRUD
- Création de Reuse Component
- Création des prosp
- Mise en place des life cycle (ngOnInit, ngOnChanges, ngOnDestroy)
- Utilisation de TrackBy avec NgFor Directive
- Utilisation des NG-Modules
- Écriture de composants dynamiques avec les Dialog
- Utilisation du d'Angular Material et Ionic
- Gestion des styles/CSS
- Utilisation des Reactive Forms
- Utilisation de projection de contenu (Ng-content)
- Utilisation d'HTTP d'Angular (Get, Post, Put)
- Développement de tableaux de bord Angular pour les partenaires logistiques afin d'améliorer l'efficacité de la résolution des litiges.
 - Gestion en temps réel avec Socket.io
- Utilisation des Async Pipe pour les Observer
- Utilisation des pipes dans les templates
- Gestion des props
- Utilisation de RxJS
 - Utilisation de mapping operator comme mergeMap, concatMap, or exhaustMap
- Optimisation des performances d'Angular grâce à l'implémentation du lazy loading.
- Conversion de l'application en PWA
 - Gestion du caching avec Angular/PWA
- Création de tableaux de bord en temps réel pour la gestion et la surveillance des robots quantitatifs
- Gestion des tableaux avec NGXs data table
- Mise en œuvre de la gestion des états NGXs pour les flux de facturation complexes.
- Gestion de l'authentification et des rôles avec Keycloak et Guards
- Envoie des données vers le back avec Socket.IO
- **Développement NestJS & Java :**
 - Utilisation de TypeScript
 - POO
 - Typages des classes
 - Création de tableaux de bord en temps réel pour la gestion et la surveillance des robots quantitatifs
 - Conception de plus de 15 microservices pour l'exécution des transactions et la conformité
 - Mise en œuvre d'un flux d'événements basé sur Kafka pour le traitement des données de marché
 - Conception de l'infrastructure avec Docker/Kubernetes (EKS) pour le déploiement des robots

- Mise en place d'une authentification sécurisée par Keycloak pour tous les services
- Gestion du pricing en temps réel via des appels API (2000 appels/h) et contourner les limites avec un système intelligent
- Personnalisation d'Apache Airflow pour les opérations de trading programmées
- Mise en œuvre d'une solution de mise en cache basée sur Redis pour les calculs de risque de portefeuille
- Gestion des secrets à l'aide de secrets AWS avec des contrôles d'accès granulaires
- Utilisation de Compodoc pour la génération de la documentation
 - Intégration de Swagger pour la doc API
- Versioning avec Git
 - Suivi du Gitflow
- Application des principes GitOps pour la gestion de l'infrastructure
 - Utilisation de Just pour lancer les commandes
- Mise en place de tests unitaires avec Jest
- Création d'un cadre de test E2E à l'aide de Cypress et Cucumber
- Création des variables d'environnement via Ancible
- Mise en place de pipelines CI/CD avec GitLab, Helm et Docker
- Optimisation des clusters EKS pour les charges de trading à haute fréquence
- Mise en place de Kubernetes :
 - Configuration Ingress

- Mise en place de config secret
- Déploiement et services
- Gestion AWS :
 - Mise en place d'AWS CloudWatch pour une surveillance complète de la plateforme
 - Configuration de la partie réseau (adresse ip, DNS)

Environnement technique :

Microservices, SOLID, Angular 16, NGXS, EKS, Kafka, Docker, Cypress, Jest, Just, Cucumber, Apache, Redis, AWS, GitOps, Gitflow, Scrum,

TRANSFONUM : Développeur Angular/NestJS (2018-2021)

Développement from scratch de la plateforme en Angular 11 et NestJS

- Analyse fonctionnelle et technique
- Participation aux choix d'architecture et technique
- Équipe de 4 personnes développeurs fullstack
 - Validation des Code reviews
 - Pair Programming
- Gestion de projet en Scrum :
 - Participations aux cérémonies Safe (Daily, démo, rétro , Pi Planning....)
 - Sprint de deux semaines
- Participation à la conception des US
- Ticketing avec JIRA

- Mise en place de l'architecture microservices
 - Création/ Maintenance de microfrontends
 - Utilisation de la méthode DDD
 - Mise en place de SOLID, Suivi de la clean architecture
 - Mise en place pattern : Singleton, Injection de dépendance (DI), Observer, Strategy, Decorator, Composite, Facade, Factory, Command, orientée aspect (AOP)
 - Documentation sur Confluence
- Développement Angular :
 - Customisation Angular Material
 - Migration d'Angular 10 vers Angular14
 - NGXS pour envoyer les informations fournies par les headers
 - Création de graphes génériques avec ngx-charts
 - Optimisation du référencement à l'aide d'Angular Universal server-side rendering.
 - Utilisation de RxJS pour gérer la réactivité
 - Utilisation de mapping operator comme mergeMap, concatMap, or exhaustMap
 - Développement app mobil avec Ionic et CapacitorJS
 - Gestion de deeplinks
 - Gestion du responsives
 - Création des forms dynamics avec Reactive Forms
 - Utilisation de leafletJS et Openstreetmap
- Configuration de OAuth2 avec Auth0
- Mise en place du PWA
- Gestion des rôles, menus, dashboard,
- Développement NestJS :
 - Utilisation de TypeScript
 - POO
 - Typages des classes
 - Conception d'un moteur de recherche alimenté par Elasticsearch avec des filtres géospatiaux et une logique de requête complexe.
 - Implémentation de la mise en cache Redis pour optimiser les temps de réponse de l'API et les performances de l'application mobile.
 - Intégration du système de paiement Stripe pour la gestion des abonnements et la facturation.
 - Création de notifications de réservation en temps réel en utilisant WebSocket (NestJS) et RabbitMQ.
 - Sécurisation de la plateforme avec l'enregistrement des audits, le RBAC et l'authentification JWT.
 - Conception de schémas de base de données PostgreSQL pour l'évolutivité et la conformité ACID.
 - Mise en place de la pile ELK (Elasticsearch, Logstash, Kibana) pour la journalisation et le débogage centralisés.

- Mise en place d'un système d'alerte automatisé pour réduire le temps de réponses en cas d'incident
- Utilisation de Git
 - Gestion du workflow Git
- Mise en œuvre des cadres de tests unitaires avec Jest
- Mise en place de tests E2E avec Cypress
- Analyse de la qualité de code via SonarQube
- Conteneurisation des applications avec Docker
- Orchestration sur GCP App Engine/Kubernetes.
- Mise en place de pipelines GitLab CI/CD avec tests et déploiements automatisés.

Environnement technique :

Microservices, DDD, SOLID, Angular 16, Ionic, Capacitor, Elasticsearch, JWT, RBAC, ACID, Postgre, Redis, Rest, Websocket, RabbitMQ, LeafletJS, OpenStreetMap, Elasticsearch, Logstash, Kibana, GCP, Kubernetes, Cypress, GitlabCI,

FORMATION

(2017) Master, Mathématique et informatique

LANGUES

Anglais

Français