



## PROGRAMME ET METHODE PEDAGOGIQUE

*Le TDD, Clean Architecture et DDD dans le monde  
Node.js et Typescript*

### **OBJECTIFS DU PARCOURS PEDAGOGIQUE**

- Sensibilisation à la discipline TDD et à la Clean Architecture dans le monde Java afin de prévenir la complexité accidentelle.  
Y compris aux notions principales du DDD et de CQRS.
- Montée en compétences au TDD et à la Clean Architecture dans le monde Node.js / Typescript si choisi à travers un live coding très professionnel et approfondi à vocation de démonstration et de mises en situation pour les exercices.  
Il consistera en la réalisation from scratch d'une application backend digne de ce qu'on attend de nous en entreprise.
- Compréhension et démonstration des différents types de tests (unitaire / acceptation / end-to-end / intégration)
- Clarification de chacun des concepts maîtres et annexes à travers des séances de questions-réponses et autres démonstrations concrètes.

**Durée :** 14 heures sur 2 jours – 45 minutes de repas le midi.

**Dates :** Mercredi 28 et jeudi 29 janvier 2026 – 09h15 / 17h00 chaque jour.

**Lieu :** En Zoom (Visio)

**Public visé :**

Particuliers et professionnels :

- Technical Leader
- Développeur Frontend
- Développeur Full Stack
- Architecte technique

**Pré requis :**

- Bonne maîtrise de Javascript et/ou Typescript, et des Promises.
- Bonnes connaissances d'un framework Node type Express.js / Nest
- Bonnes connaissances des concepts d'injection de dépendances.
- Bonnes connaissances en OOP et/ou Functional Programming
- Capacité à écrire un simple test unitaire avec Jest ou Mocha

## DÉROULEMENT DU PARCOURS PEDAGOGIQUE

### Jour 1

- Rapide tour de table, présentation de chacun et exposition des attentes
- Introduction et cours théorique sur le TDD cassant les énormes quiproquos à son sujet
- Si le groupe est vraiment novice en TDD, Kata éventuel et judicieusement choisi de mises en pratique avec Node.js, Typescript et Jest
- Introduction et cours théorique sur la Clean Architecture
- Démarrage d'écriture d'une application "from scratch" digne d'un cas réel d'entreprise en TDD (User Story avec plusieurs règles de gestion) tout en respectant la Clean Architecture dans le monde Node.js / Typescript.  
Approche agile avec le mindset NoEstimates initiée par un mini atelier BDD judicieusement mené.  
Notions DDD principales expliquées et prises en compte (Aggregates / Value Objects / Repositories / Factories (au sens DDD), Bounded Contexts / Domain Events).
- Séances de questions / réponses tout au long

### Jour 2

- Suite du live coding de l'application "from scratch" digne d'un cas réel d'entreprise
- Séances de refactoring au fil de l'eau du code ET des tests de l'application exemple, rendues plaisantes et sans crainte grâce au TDD
- Clarification de concepts subtils relatifs au TDD et à la Clean Architecture
- Liaison à une base de données PostgreSQL avec TestContainers – démonstration de tests d'intégration.
- Compréhension du concept de "test d'intégration"
- Séance de questions / réponses tout au long

## CAPACITES DÉVELOPPÉES

- Maîtrise de l'approche des problèmes algorithmiques avec le mindset TDD
- Maîtrise de l'écriture des tests unitaires orientés comportements, de tests d'acceptation, de tests d'intégration et de tests end-to-end
- Maîtrise du concept d'inversion de dépendances (DIP)
- Maîtrise de la Clean Architecture (Hexagonale Architecture) et de l'arborescence de fichiers optimale
- Capacité à relier une base de données avec Knex.js (interchangeable avec TypeORM), dans un environnement respectant les principes de la Clean Architecture.
- Capacité à refactorer des tests astucieusement (techniques de chapitrage / nommage / D.R.Y), essentielle pour évoluer sereinement en TDD
- Bonnes notions des concepts DDD principaux et de CQRS

## DEMARCHE ET METHODES PROPOSEES

La formation se présentera sous la forme d'une alternance entre découverte de concepts théoriques, annihilation de fausses croyances, démonstrations complètes de l'approche par du live coding, exercices et échanges constants avec les participants.

L'outil de communication requis pour cette formation est l'alternance entre l'oral, l'écrit, le partage d'écran si visio (audio/video) et la manipulation d'un éditeur de code.

## LES MOYENS PEDAGOGIQUES

- Des apports théoriques sur le processus
- Des exemples concrets
- Des démonstrations complètes par le formateur en live coding
- Exercices réalisés en live par les participants afin de s'exercer sur cette application d'entreprise.
- Challenges proposés quant au TDD, Clean Archi et au refactoring de code (modification de structure du code)

## EVALUATION FORMATIVE

Une feuille d'évaluation de compétences est remise à chaque stagiaire leur permettant d'évaluer leurs acquis.

Une deuxième évaluation permet d'évaluer les modalités pédagogiques et le contenu de la formation, de mesurer le degré de satisfaction à chaud du stagiaire pour adapter les formations suivantes aux besoins spécifiques des participants.

## DOCUMENTS ET SUPPORTS POUR LES STAGIAIRES

À l'issue de cette session, il sera remis à chaque stagiaire le support de formation, le projet sur Git réalisé ensemble, ainsi qu'une certification de réalisation de l'action de formation.

## ENCADREMENT PEDAGOGIQUE

La formation sera assurée par [Michaël AZERHAD](#)

Profil et qualité : Président de S.A.S.U WealCome et expert technique de plus de 15 ans d'expérience

## MODALITES D'ACCES

Contactez-nous au [06 09 88 51 02](tel:0609885102) ou par mail : [contact@wealcomecompany.com](mailto:contact@wealcomecompany.com)

## DELAIS D'ACCES

Nous pouvons programmer les formations en fonction de vos contraintes et de nos disponibilités.

Nos formations sont accessibles aux personnes en situation de handicap.  
Les aspects, l'accessibilité et le type de handicap au regard des modalités pédagogiques sont à évoquer impérativement au cours de l'entretien préalable à toute contractualisation afin de pouvoir orienter ou accompagner au mieux les personnes en situation de handicap.