

# **Diaporama de point d'étape : Développeur Full Stack - Java et Angular**

# Première partie de Développeur Full Stack

Mes réalisations clés :

- P2 : Développez le front-end en utilisant Angular
- P3 : Développez le back-end en utilisant Java et Spring
- P4 : Définissez les technologies et les paradigmes d'une application
- P5 : Testez une application full-stack
- P6 : Développez une application full-stack complète

## P2 – Développer le front-end en utilisant Angular

Étapes que j'ai trouvé plus faciles :

- Appréhender la construction d'une application avec Angular
- Utiliser Angular pour créer une application de type SPA
- Utiliser un service pour récupérer les données d'un back-end

Étapes que j'ai trouvé plus difficiles :

- Utiliser la librairie de composants dans le but de proposer un affichage adaptés aux besoins techniques et en lien avec les maquettes

Points d'intérêt :

- Découvrir de nouvelles librairies de composants et d'éléments graphiques dans le but de créer des charts

# P3 – Développer le back-end en utilisant Java et Spring

Étapes que j'ai trouvé plus facile :

- Créer une architecture en couches séparées de sorte à répartir la logique métier en fonction des fonctionnalités de chaque classe

Étapes que j'ai trouvé plus difficile :

- Configurer la chaîne de sécurité dans le but de permettre une interaction sécurisée avec l'API
- Apprendre à administrer les CORS et le CSRF

Points d'intérêt :

- Maîtriser l'utilisation de Spring Security dans le but de créer un back-end résilient et performant

# P4 – Définir les technologies et le paradigme d'une application

Étapes que j'ai trouvé plus facile :

- Réaliser une veille technologique pour obtenir des informations récentes sur les technologies que j'allais employer par la suite
- Comprendre les différences entre les framework front-end et back-end

Étapes que j'ai trouvé plus difficile :

- Utiliser les outils de veilles fournis (certains n'avaient pas de capacité de partage à un groupe sans devoir dépenser pour l'offre payante)

Points d'intérêt :

- Comprendre et utiliser des outils modernes dans le but d'automatiser la récolte d'informations dans le but de se tenir à la page et savoir quelles technologies seront les plus adaptées à la création d'un applicatif full-stack

# P5 – Tester une application Full Stack

Étapes que j'ai trouvé plus facile :

- Réaliser les tests unitaires de la partie front-end avec Jest
- Utiliser Junit et Mockito pour réaliser des tests unitaires dans l'applicatif back-end
- Utiliser le fichier pom.xml (Maven) pour configurer l'exclusion de certains éléments de l'applicatif back-end dans le but de ne pas les comptabiliser pour la couverture de test

Étapes que j'ai trouvé plus difficile :

- Apprendre à utiliser Cypress correctement dans le but de fournir un fichier de coverage permettant de savoir globalement le score de couverture de code de l'applicatif front-end

Points d'intérêt :

- Découvrir l'utilisation de Cypress et comprendre son fonctionnement pour réaliser des tests E2E d'un applicatif full-stack.
- Voir l'applicatif fonctionner « automatiquement » via l'utilisation de Cypress et voir l'ensemble des tests se dérouler sous mes yeux, graphiquement

# P6 – Développer une application Full Stack complète

Étapes que j'ai trouvé plus facile :

- Relier l'application front-end avec l'application back-end via l'utilisation d'appels REST et de DTOs
- Découper la logique métier en composants ou en couches dans le cadre de l'applicatif front / back.
- Créer l'applicatif et la logique back-end et la relier à une base de données conteneurisée

Étapes que j'ai trouvé plus difficile :

- Utiliser et comprendre le fonctionnement de la librairie de composants UI, Angular Material
- Construire le design de l'application via l'utilisation de SCSS et d'Angular Material de sorte à la rendre responsive

Points d'intérêt :

- Découvrir la librairie de composants UI Angular Material dont on ne cessait de me parler depuis quelques années
- Créer un applicatif complet et capable de fonctionner correctement en lien avec une demande client

# Retour sur la progression

Points sur lesquels j'ai durement travaillé :

- Découverte de nouvelles technologies de composants UI
- Découverte de Cypress et de son utilisation

Points sur lesquels j'ai progressé :

- Création d'un applicatif back-end avec Spring Security
- Ecriture de tests d'applicatifs front-end avec Jest
- Ecriture de mocks pour le backend avec Mockito
- Réalisation de code coverage, pour le front-end et le back-end, avec les librairies dédiées