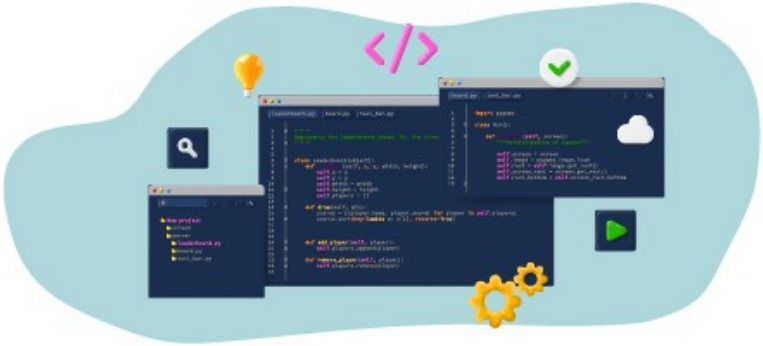
ORION



Justification des choix techniques

Projet MDD

Auteur : [PELISSIER Julien]

Version 0.0.1

SOMMAIRE :

Synthèse :

Dans cette partie nous allons détailler le choix que nous avons mené pour répondre aux besoins de l'application.
Certains choix ont été déterminé dans le document de contraintes techniques. C'est le cas pour le back-end avec Java et Spring et côté front-end avec Angular et Typescript. Ces choix ne représentent pas à eux seuls les composants du projet.
Nous traiterons par la suite des autres éléments nécessaires à savoir des librairies, frameworks, outils ou encore design patterns que nous utiliserons pour réaliser à bien cette application.

Compilation :

Afin de compiler notre front-end et notre back-end, nous allons respectivement utiliser Npm et node ainsi que maven.
Npm et node fonctionnent de pair pour pouvoir télécharger les différents composants et compiler une API front quant à maven, côté back, qui gère ces deux fonctionnalitées.

Framework :

En ce qui concerne les frameworks nous en avons choisis 3.
Pour le back-end, Junit nous permettra de tester notre application facilement.
Pour le front-end, Jest nous permettra de rédiger des tests unitaires ainsi que des tests d'intégrations. Nous aurons ensuite besoin de Cypress pour effectuer des tests end to end sur notre application.

Librairie :

Ensuite, nous allons traiter des librairies que nous avons choisis pour faciliter le développement de notre API back-end.
Dans un premier temps nous allons utiliser Data Jpa pour gérer les données mysql.
Ensuite, Mockito nous permettra de simuler le comportement d'une classe de manière à pouvoir isoler le comportement des classes à tester. Nous testerons à travers des tests d'intégrations qui utiliserons une base de données en mémoire, grâce à la librairie H2.
Puis nous utiliserons Lombok pour gérer les setters & getters facilement.
Pour finir nous utilserons JaCoCo pour déterminer le pourcentage de couverture que nos tests effectuent.

Choix techniques

Mysql

choix technique	lien vers le site / la documentation / une ressource	but du choix
mysql	*https://dev.mysql.com/doc/refman/8.4/en/*	*Gérer la base de données*

Justification du choix technique : Cet outil va nous permettre de gérer complètement la base de données de notre application.

Data JPA

choix technique	lien vers le site / la documentation / une ressource	but du choix
Spring Data JPA	*https://spring.io/projects/spring-data-jpa#learn*	*Facilite l'accès aux données*

Justification du choix technique : Cet outil va permettre de rendre plus facile l'accès aux données notamment avec un CrudRepository.

Junit

choix technique	lien vers le site / la documentation / une ressource	but du choix
Junit	*https://junit.org/junit5/docs/current/user-guide/*	*Tester le back-end*

Justification du choix technique : Cet outil va nous permettre de tester l'ensemble de l'API back-end. Cet aspect de test n'avait pas été pris en compte lors de la rédaction des contraintes techniques.

H2

choix technique	lien vers le site / la documentation / une ressource	but du choix
H2 database	*https://www.h2database.com/html/quickstart.html*	*Simuler une base de données*

Justification du choix technique : Cet librairie va nous permettre de simuler une base de données en mémoire. Il sera très utile pour réaliser des tests sans toucher à la vraie base de données. Cet aspect de test n'avait pas été pris en compte lors de la rédaction des contraintes techniques.

Mockito

choix technique	lien vers le site / la documentation / une ressource	but du choix
mockito	*https://javadoc.io/doc/org.mockito/mockito-core/latest/org/mockito/Mockito.html*	*Simuler le fonctionnement d'une classe*

Justification du choix technique : Cette librairie nous permet de simuler le comportement d'une classe et ainsi d'isoler une classe par rapport à une autre. Il nous sera utile lors de tests unitaires notamment. Cet aspect de test n'avait pas été pris en compte lors de la rédaction des contraintes techniques.

Lombok

choix technique	lien vers le site / la documentation / une ressource	but du choix
Lombok	*https://projectlombok.org/features/*	*Gérer les getters & setters via une annotation spring*

Justification du choix technique : Cette librairie va nous permettre de gérer l'appel de getters et setters même s'il ne sont pas rédigés via une simple annotation Spring.

Maven

choix technique	lien vers le site / la documentation / une ressource	but du choix
Maven	*https://maven.apache.org/guides/index.html*	*Compiler l'application back-end*

Justification du choix technique : Cet outil open source va nous permettre de compiler notre application Back-end et de gérer l'installation des modules nécessaires au bon fonctionnement.

JaCoCo

choix technique	lien vers le site / la documentation / une ressource	but du choix
Jacoco	*https://www.jacoco.org/jacoco/trunk/doc/*	*Contrôler le pourcentage de couverture des tests back-end*

Justification du choix technique : JaCoCo nous permettra d'obtenir et de contrôler le pourcentage de couverture de nos tests sur le code global de l'API back-end. Cet aspect de test n'avait pas été pris en compte lors de la rédaction des contraintes techniques.

Jest

choix technique	lien vers le site / la documentation / une ressource	but du choix
Jest	*https://archive.jestjs.io/docs/en/22.x/getting-started.html*	*Tester le front-end*

Justification du choix technique : Cet outil va nous permettre de tester l'ensemble de l'API front-end. Cet aspect de test n'avait pas été pris en compte lors de la rédaction des contraintes techniques.

Npm & node

choix technique	lien vers le site / la documentation / une ressource	but du choix
Npm & node	*https://docs.npmjs.com/getting-started*	*Gestion de la configuration front-end, installation de module et compilation*

Justification du choix technique : Ces deux outils qui fonctionnent ensemble vont nous permettre de gérer la configuration de l'application front-end au point de vue des modules.

Cypress

choix technique	lien vers le site / la documentation / une ressource	but du choix
cypress	*https://docs.cypress.io/guides/overview/why-cypress*	*Réaliser des tests end to end côté front-end*

Justification du choix technique : Cet outil nous permettra de réaliser des tests end to end pour notre front-end de manière à valider l'intéraction entre le front et le back. Cet aspect de test n'avait pas été pris en compte lors de la rédaction des contraintes techniques.