HOANG DOAN SON TUNG

DOSSIER DE SPÉCIFICATION ET TEST

Candidature E2VR - Application Web

Remarque :

Version des modules utilisés (On les trouve dans *package.json*) :

- Node : 15.6.1
- Express : 4.17.1
- MongoDB : 3.6.16
- Cors : 2.8.5
- Body Parser : 1.19.0

1. Choix de mise en œuvre

Pour réaliser l'application, j'ai choisi d'utiliser Express avec Node JS et MongoDB.

a) Express

Express est un Framework pour créer des applications Web au-dessus de Node.js. Cela simplifie le processus de création de serveur déjà disponible dans Node.

Avantage : Gagne du temps par rapport aux opérations Parse des requests entrantes.

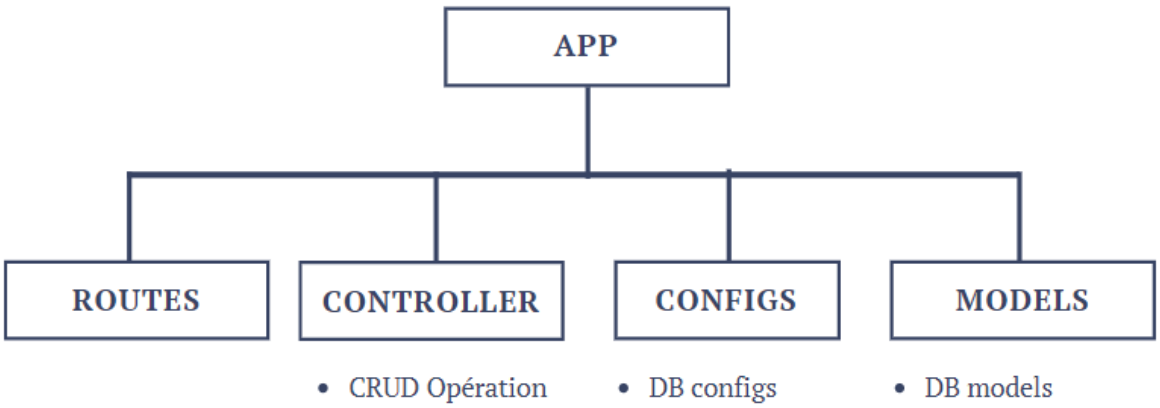
b) MongoDB

- MongoDB est très flexible et adaptable aux situations et exigences réelles du monde des affaires.
- MongoDB utilise JavaScript comme langage de requête. (D'où on travaille avec Node JS)
- MongoDB enregistre les documents dans un format de type JSON, ce qui signifie qu'il est très simple de convertir les requêtes et les résultats dans un format que notre code frontend comprend.
- MongoDB peut fonctionner sur plusieurs serveurs, équilibrant la charge et / ou dupliquant les données pour maintenir le système opérationnel en cas de panne matérielle.

c) Node JS

- Open source, populaire.
- Utilisé Google V8 Engine
- En plus d'être efficace dans ce qu'il fait, Node.js est populaire car il possède un énorme écosystème actif, open-source et basé sur JavaScript. En outre, cela n'a pas tendance à rompre la compatibilité entre les versions de manière majeure.
-

2. Structure des fichiers codes



Je définis la structure de mes dossiers de manière qu'il soit facile et évident de trouver des morceaux de code plus tard. Ainsi, on trouve :

- Configs : Variable d'environnement et configuration
- Controllers : Intéragir avec les modèles pour récupérer des informations et finalement répondre aux demandes des utilisateurs.
- Models : Models de la base de données.
- Routes : Contient contrôleurs de routes pour api

3. Test

a) Connexion vers DB

b) **Nom de l'application**	**Nature du test**	**Référence**
Appli E2VR	Nominal	
Auteur	**Test de connexion au Base de données**	**Version**
HDST		1.0

Objectifs du test

Référence :

Connection aux bases de données

Description du test :

Il s'agit de vérifier que l'application peut connecter au base de données

Contexte

Configuration matériel et logiciel :

CLI de Windows 10 (cmd)

Résultat :

```
F:\CONFIDENTIAL\LEARN\Web\appli-e2vr>node server.js
Hey! I am listening on port 8080 :D
Connection Successful
```

c) Créer un utilisateur

Nom de l'application	**Nature du test**	**Référence**
Appli E2VR	Nominal	
Auteur	**Test de création d'un utilisateur**	**Version**
HDST		1.0

Objectifs du test

Référence :

Création d'un utilisateur

Description du test :

Il s'agit de vérifier que l'application peut créer un utilisateur dans la base de données

Contexte

Configuration matériel et logiciel :

Postman

Résultat :

Dans le cas on ne met rien :

```
{
    "message": "Error: Empty field! "
}
```

d) Modifier un utilisateur
   Pas réussir.
e) Supprimer un utilisateur
   Pas réussir.
f) Lire un utilisateur
   Pas réussir.
g) Lire tous les utilisateurs
   Pas réussir.