

Atelier 06

Objectifs d'apprentissages :

- Utilisation de la bibliothèque Mongoose

La bibliothèque Dotenv

1- Définition

Le dotenv est un module qui charge les variables d'environnement d'un fichier .env dans process.env.

2- Installation

```
npm install dotenv
```

3- Exemple de contenu de fichier .env :

```
.env
HOST = localhost
DATABASE = ydb
PORT = 5432
```

4- Utilisation

```
simple.js
require('dotenv').config()

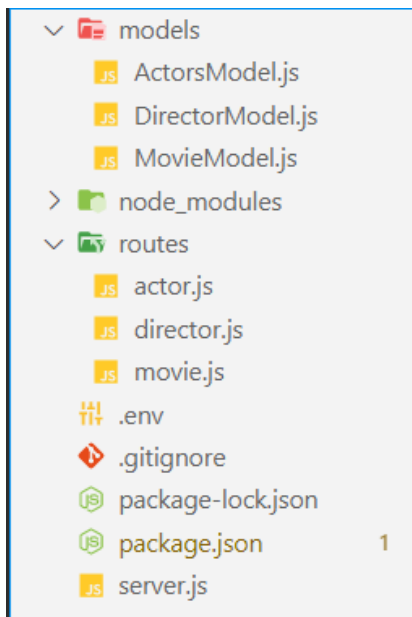
const hostname = process.env.HOST;
const database = process.env.DATABASE;
const port = process.env.PORT;

console.log(hostname);
console.log(database);
console.log(port);
```

Travail a réalisé

1. Créer un dossier sous le nom "Atelier06-express".
2. Lancer la commande de création de fichier package.json
3. Installer les packages suivants :
 - Express
 - Mongoose
 - Nodemon
 - Cors
 - Dotenv

4- Structurer le dossier de votre projet de la façon suivante :



5- Dans le fichier .env, déclarer les variables d'environnement suivante :

Nom de variable	Description
PORT	3000
URL_MONGOOSE	Chaine de connexion vers MongoDB
DBNAME	DBMOVIES

6- Dans le fichier server.js :

- Récupérer les variables d'environnement
- Permettre l'analyse des requêtes JSON.
- Connexion à la base de données MongoDB.
- Les requêtes qui commencent par /actors seront redirigées vers le fichier actor dans le dossier routes.
- Les requêtes qui commencent par /director seront redirigées vers le fichier director dans le dossier routes.
- Les requêtes qui commencent par /movie seront redirigées vers le fichier movie dans le dossier routes.
- Démarrer le serveur dans le port 3000 (valeur déjà défini dans le fichier .env).

7- Dans le fichier ActorsModel :

- Définir le schéma qui représente un document de la collection Actors.
- Exporter par défaut le modèle de ce schéma.

8- Dans le fichier DirectorModel :

- Définir le schéma qui représente un document de la collection Director.
- Exporter par défaut le modèle de ce schéma.

9- Dans le fichier MovieModel :

- Définir le schéma qui représente un document de la collection Movie.

- Exporter par défaut le modèle de ce schéma.

10. Dans le fichier actor.js, créer les routes suivantes :

Chemin	Méthode	Description
/all	GET	Retourne la liste de tous les acteurs.
/names	GET	Retourne les noms des acteurs
/movies	GET	retourne le nombre de film par acteur (retourne le nom de l'acteur et le nombre de film)
/add	POST	Ajout d'un acteur (passe dans le corps de la requête)
/update/:name	PUT	Modifie les informations d'un acteur en se basant sur son nom.
/delete/:name	DELETE	Supprime un acteur

11. Dans le fichier director.js, créer les routes suivantes :

Chemin	Méthode	Description
/all	GET	Retourne la liste de tous les directeurs.
/names	GET	Retourne les noms des directeurs
/movies	GET	retourne le nombre de film par acteur (retourne le nom de l'acteur et le nombre de film)
/add	POST	Ajout d'un directeur (passe dans le corps de la requête)
/update/:name	PUT	Modifie les informations d'un directeur en se basant sur son nom.
/delete/:name	DELETE	Supprime un directeur

12. Dans le fichier movie.js, créer les routes suivantes :

Chemin	Méthode	Description
/all	GET	Retourne la liste de tous les films.
/actors/:filmname	GET	Retourne toutes les informations sur les acteurs d'un film passe en paramètre.
/directors/:filmname	GET	Retourne toutes les informations sur les directeurs d'un film passe en paramètre.
/listCategorie/:category	GET	Retourne la liste de tous les films d'une catégorie passe en paramètre.
/list	GET	
/years/:year1/:year2	GET	Retourne les films sortis entre les deux paramètres year1 et year2
/add	POST	Ajout d'un directeur (passe dans le corps de la requête)
/update/:name	PUT	Modifie les informations d'un directeur en se basant sur son nom.
/delete/:name	DELETE	Supprime un directeur

13. Tester votre API