

self-service-machine-api - STEP 5

Dans cette étape nous allons mettre en place la base de données !

MySQL

Nous allons utiliser une base de données MySQL.

Comme vous en avez déjà l'habitude nous allons utiliser les 2 conteneurs docker.

- 1 conteneur pour MySQL
- 1 conteneur pour PhpMyAdmin

Utilisation d'un ORM

Un **ORM** (Object Relational Mapping).

Grâce à l'ORM on va "mapper" (associer) 1 classe js, appelé un modèle, avec une table MySQL

Un des objectifs de l'ORM est de permettre au développeur d'interroger la base de données sans utiliser la syntaxe SQL mais en faisant de la programmation orientée objet (POO).

Installer l'ORM Sequelize

Nous allons installer l'ORM Sequelize :

```
npm install sequelize --save
```

Nous allons également installer un driver MySQL :

```
npm install mysql2 --save
```

Connection à la base de données

Dans un 1er temps, nous devons créer une base de données dans MySQL.

Par exemple, créons la db `db_products` .

Ensuite on crée un fichier `sequelize.mjs` dans le répertoire `db` .

```
import { Sequelize } from "sequelize";

const sequelize = new Sequelize(
  "db_products", // Nom de la DB qui doit exister
  "root", // Nom de l'utilisateur
  "root", // Mot de passe de l'utilisateur
  {
    host: "localhost",
    //port: "6033", pour Les conteneurs docker MySQL
    dialect: "mysql",
    logging: false,
  }
);
```

```
export { sequelize };
```

Maintenant dans le fichier `src/app.js` :

```
...  
const port = 3000;  
  
import { sequelize } from "../db/sequelize.mjs";  
  
sequelize  
  .authenticate()  
  .then((_) =>  
    console.log("La connexion à la base de données a bien été établie")  
  )  
  .catch((error) => console.error("Impossible de se connecter à la DB"));  
  
...
```

Maintenant si vous lancez l'application `npm start` vous devriez voir dans la console 'La connexion à la base de données a bien été établie'

Passons à l'étape n°6