

self-service-machine-api - STEP 6

Maintenant que nous avons pu nous connecter à notre base de données via l'ORM, nous allons pouvoir commencer à utiliser l'ORM sequelize.

Créer le model Product

Nous allons créer un fichier `products.mjs` dans le dossier `/src/models`

// <https://sequelize.org/docs/v7/models/data-types/>

```
const ProductModel = (sequelize, DataTypes) => {
  return sequelize.define(
    "Product",
    {
      id: {
        type: DataTypes.INTEGER,
        primaryKey: true,
        autoIncrement: true,
      },
      name: {
        type: DataTypes.STRING,
        allowNull: false,
      },
      price: {
        type: DataTypes.FLOAT,
        allowNull: false,
      },
    },
    {
      timestamps: true,
      createdAt: "created",
      updatedAt: false,
    }
  );
};

export { ProductModel };
```

Synchroniser les données de MOCK avec la DB

Nous pouvons maintenant utiliser le modèle `ProductModel` pour créer une relation 1 <--> 1 avec la table `products` de la base de données.

De plus, pour avoir quelques produits dans cette table, nous allons utiliser les produits présents dans le fichier `mock-product.mjs` et les importer dans la DB.

Le code ci-dessous doit être présent dans le fichier `src/db/sequelize.mjs`.

```
import { Sequelize, DataTypes } from "sequelize";
import { ProductModel } from "../models/products.mjs";

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",
    dialect: "mysql",
  }
);
```

```

    logging: false,
  }
});

import { products } from "../mock-product.mjs";

// Le modèle product
const Product = ProductModel(sequelize, DataTypes);

let initDb = () => {
  return sequelize
    .sync({ force: true }) // Force la synchro => donc supprime Les données également
    .then((_) => {
      importProducts();
      console.log("La base de données db_products a bien été synchronisée");
    });
};

const importProducts = () => {
  // import tous les produits présents dans le fichier db/mock-product
  products.map((product) => {
    Product.create({
      name: product.name,
      price: product.price,
    }).then((product) => console.log(product.toJSON()));
  });
};

export { sequelize, initDb, Product };

```

Il nous reste plus qu'à appeler la méthode `initDb()` dans le fichier `src/app.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"));

initDb();
...

```

Voilà ! Les produits ont tous été importés dans la table `products` de la DB.

D'ailleurs vous devriez les voir dans la console du serveur.

Passons à l'étape n°7