

Site e-commerce de la M2L – Documentation technique



## Sommaire :

- I) Langage utilisés.....
- II) Mise en place du projet.....
  - A) Installation du front/back.....
  - B) Liaison avec l'API.....

## I) Langages utilisés :

### JavaScript :

- Créer les différents composants du site pour définir sa structure
- Communiquer avec l'API REST afin de récupérer et d'ajouter des informations dans la base de données du site



### MySQL :

- Créer la base de données du site



### NodeJs :

- Créer l'API REST et permettre la liaison entre le front et le back



## II) Mise en place du projet :

### A) Installation du front/back :

Pour l'installation du front end, nous importons dans chaque composant du projet les modules externes nécessaires.

```
1 import { useParams, useNavigate } from "react-router-dom";
2 import { useEffect, useState } from 'react';
3 import { useForm } from "react-hook-form";
4 import { Link } from 'react-router-dom'
5 import axios from 'axios';
```

- useParams et useNavigate permettent respectivement d'extraire les paramètres de l'URL dans un composant de l'application et de naviguer vers une autre URL
- useEffect et useState permettent d'exécuter du code en réponse à un changement d'état ou de propriété d'un composant et de déclarer et mettre à jour des variables d'état
- useForm facilite la gestion des formulaires en React
- Link permet de créer des liens de navigation au sein de l'application
- Axios va permettre d'effectuer les requêtes http directement à l'API

Il faut bien penser à installer tout ces modules au préalable via un terminal.

Pour l'installation du back end il faut commencer par la création de l'API qui communiquera entre l'application et la base de données :

```

1  const express = require('express') // la récupération d'express
2  const app = express() // variable utilisant la librairie express
3  let cors = require('cors')
4
5  require('dotenv').config()
6
7  // connexion à la bdd
8  const mariadb = require('mariadb');
9
10 const pool = mariadb.createPool({
11   host: process.env.DB_HOST,
12   database: process.env.DB_DTB,
13   user: process.env.DB_USER,
14   password: process.env.DB_PWD,
15 });
16
17 app.use(express.json())
18 app.use(cors())
19
20 //Début de l'api côté produit
21
22 app.get('/produit', async(req,res) => { // affichage des produits
23   let conn;
24   try{
25     console.log("lancement de la connexion")
26     conn = await pool.getConnection();
27     console.log("lancement de la requete")
28     const rows= await conn.query('SELECT * FROM produit');
29     console.log(rows);
30     res.status(200).json(rows)
31   }
32   catch(err){
33     console.log(err);
34   }
35 })

```

Dans l'exemple ci-dessus, on retrouve la constante « pool » qui permet à l'aide de variable de se connecter à la base de données. Les variables étant définies dans le fichier .env dans lequel se trouvent les informations de connexion de la base.

On y retrouve aussi un exemple de requête (ici un get) qui sera plus tard utilisée dans les composants de l'application.

## B) Liaison avec l'API :

Une fois le front end et le back end installés, il faut maintenant les lier afin que l'application fonctionne comme prévu.

Pour cela, nous allons utiliser le module axios, installé plus tôt dans la création du front end.

```
const recup = async () => {  
  await axios.get(`http://localhost:8000/produit`)  
    .then(res => {  
      console.log(res)  
      setProduits(res.data)  
      setAffichage(true)  
    })  
}
```

Ce module va permettre d'envoyer une requête sur le serveur où l'on retrouve le back end.