- **Installer docker (logiciel)**
  docker --version
  docker-compose –version
  wsl --list –verbose
  wsl --install
  - Ouvre **Docker Desktop** sur Windows.
  - Clique sur **Settings**.
  - Va dans **Resources → WSL Integration**.
  - Active **Enable integration with my default WSL distro** (ou coche la distro que tu utilises, ex : Ubuntu).
  - Clique sur **Apply & Restart**.

  wsl

  docker --version
  docker-compose –version
  docker run hello-world
- **npm init -y**

  Initialisation du projet

- **npm install express mongoose cors dotenv body-parser**

  express : framework web pour Node.js

  mongoose : ODM pour MongoDB

  cors : pour gérer les requêtes cross-origin

  dotenv : pour les variables d'environnement

  body-parser : pour parser le JSON des requêtes

- **Structure du backend**

commercialiseo/

├docker-compose.yml
├backend/
│
├– server.js # point d'entrée du serveur
├–. env # variables d'environnement
├– package.json
├– config/
│ └– db.js # connexion MongoDB
├– models/ # schémas Mongoose (Admin, Boutique, Acheteur, Produits, etc.)
├– routes/ # routes Express
├ - testRoutes.js
└– controllers/ # logique des routes
└– Dockerfile

- **Voici les fichiers :**
  - Server.js

```
const express = require('express');
const dotenv = require('dotenv');
const cors = require('cors');
const connectDB = require('./config/db');

dotenv.config();
connectDB();

const app = express();

// Middleware
app.use(cors());
app.use(express.json());

// Routes
app.use('/api/test', require('./routes/testRoutes'));

const PORT = process.env.PORT || 5000;
app.listen(PORT, () => console.log(`Server running on port ${PORT}`));

```

  - .env

```
PORT=5000
MONGO_URI=mongodb://mongodb:27017/commercialiseo
```

  - db.js

```
const mongoose = require('mongoose');

const connectDB = async () => {
    try {
        await mongoose.connect(process.env.MONGO_URI);
        console.log('MongoDB connected');
    } catch (err) {
        console.error(err.message);
        process.exit(1);
    }
};

module.exports = connectDB;
```

- testRoutes.js

```
const express = require('express');
const router = express.Router();

router.get('/', (_req, res) => {
    res.json({ message: 'API fonctionne parfaitement !' });
});

module.exports = router;
```

- Dockerfile

```
# Utiliser Node 22 LTS
FROM node:22

# Créer un dossier de travail dans le container
WORKDIR /usr/src/app

# Copier package.json et package-lock.json
COPY package*.json ./

# Installer les dépendances
RUN npm install

# Copier tout le code source
COPY . .

# Exposer le port que le serveur va utiliser
EXPOSE 5000

# Lancer le serveur
CMD ["node", "server.js"]

```

- docker-compose.yml

```yaml
version: '3.8'

services:
  mongodb:
    image: mongo:6
    container_name: mongodb
    restart: always
    ports:
      - "27017:27017"
    environment:
      MONGO_INITDB_DATABASE: commercialiseo
    volumes:
      - mongo-data:/data/db

  backend:
    build: ./backend        # chemin vers le Dockerfile
    container_name: backend
    restart: always
    ports:
      - "5000:5000"
    environment:
      MONGO_URI: mongodb://mongodb:27017/commercialiseo
    depends_on:
      - mongodb

volumes:
  mongo-data:
```

- **docker-compose up –build**

Pour faire marcher le Docker du backend (Express) +mongoDB

- **docker-compose down**

Pour arrêter les containers

- **Tester l'API avec** : http://localhost:5000/api/test

Reponse :

{

"message": "API fonctionne parfaitement !"

}