



Documentation générale

BOWLBOX 3000

Ce document comporte les informations de développement du produit et du site web.



I. Le produit

Pour réaliser la conception 3D de l'enceinte bluetooth, le logiciel utilisé est SolidWorks. Afin de mieux comprendre comment il a été assemblé, consultez la notice d'assemblage disponible dans le même répertoire que ce fichier.

Pour tous autres détails et spécifications technique, consultez le site de vente en ligne de la Bowl Box 3000 : <https://bowlbox3000.netlify.app/>

II. Le site web

1. Back-end

Le backend du projet est développé en Node.js avec TypeScript, utilisant Express.js et MongoDB.

Fonctionnalités

- Authentification JWT
 - Gestion des commandes
 - Documentation API Swagger
 - Middleware d'authentification
 - Modèles Mongoose pour MongoDB
 - Architecture modulaire (Routes, Controllers, Services, Models)
-

Installation

1. Cloner le repo

git clone <https://github.com/usine-du-futur-enceinte/Website-Backend.git>

cd Website-Backend

2. Installer les dépendances

npm install

3. Configurer les variables d'environnement

Créer un fichier .env à la racine du projet et ajouter les variables d'environnement suivantes :

PORT=3000

MONGO_URI=mongodb+srv://[DB_USERNAME]:[DB_PASSWORD]@usinedufutur.ltjag.mongodb.net/

JWT_SECRET=your_secret_key

JWT_REFRESH_SECRET=your_refresh_secret_key

4. Démarrer le serveur

- En mode développement : npm run dev
 - En mode production : npm start
-

Tests

1. Configurer les variables d'environnement de test

Créer un fichier .env.test à la racine du projet et ajouter les variables d'environnement suivantes :

NODE_ENV=test

PORT=3030

MONGO_URI=mongodb+srv://[DB_USERNAME]:[DB_PASSWORD]@usinedufutur.ltjag.mongodb.net/

2. Lancer les tests

npm test



3. Générer un rapport de couverture

npm run test:coverage

Structure du projet

Website-Backend/

```
|— 📁 src
|— 📁 config/      # Configuration (Swagger, DB)
|— 📁 controllers/ # Logique métier des routes
|— 📁 middlewares/ # Middleware d'authentification JWT
|— 📁 models/      # Schémas Mongoose (Users, Orders)
|— 📁 routes/      # Définition des routes API
|— 📁 services/    # Logique métier centralisée
|— app.ts         # Point d'entrée Express
|— server.ts      # Lancement du serveur
```

—  tests/	# Tests unitaires et d'intégration
—  coverage/	# Rapport de couverture des tests
— package.json	# Dépendances et scripts npm
— tsconfig.json	# Configuration TypeScript
— .env.example	# Exemple de configuration
— README.md	# Documentation du projet

Documentation API

Une documentation swagger existe sur la route **/api-docs**

Technologies Utilisées

- Node.js + Express.js
 - TypeScript
 - MongoDB + Mongoose
 - JWT pour l'authentification
 - Swagger pour la documentation API
 - Cors pour gérer les requêtes cross-origin
 - Dotenv pour la gestion des variables d'environnement
-

Contribuer

1. Forker le projet
2. Créer une branche (git checkout -b feature-nouvelle-feature)
3. Commit (git commit -m "Ajout d'une nouvelle feature")
4. Pousser (git push origin feature-nouvelle-feature)
5. Ouvrir une Pull Request

2. Front-end

Pour accéder au code github du frontend, suivez le lien

<https://github.com/usine-du-futur-enceinte/Website-Frontend>

Technologies utilisées : Pour la partie visible du site web, les technologies utilisées sont : Réact Vite, Typescript, CSS.

Fonctionnalités

- Page d'accueil
- Page d'authentification
- Page panier
- Design basé sur Figma

<https://www.figma.com/design/HYwmqC5VdPo9iANIFkNuUf/Enceinte-3D-WireFrames?node-id=1-2&t=rPqupnee6U6COu6p-1>

- Chargement rapide grâce à Vite

Pour les détails spécifiques du développement, veuillez consulter les **“ReadMe”** présents dans le github.

Plus d'informations, contactez

Frontend : [Axel DEFO](#)

Backend : [Damien Riandiere](#)