🚢 Port Russell – Rapport technique du projet API

1. Présentation générale

Le projet Port Russell est une API Express.js dédiée à la gestion d’un port de plaisance. Il permet de :

* gérer les utilisateurs
* consulter les emplacements de catways
* administrer les réservations
* sécuriser l'accès avec un système JWT
* interagir avec une interface HTML locale pour tester les routes

2. Stack technique utilisée

| **Technologie** | **Rôle** |
| --- | --- |
| Express.js | Serveur backend Node.js |
| MongoDB + Mongoose | Base de données pour les utilisateurs, catways, réservations |
| JWT (jsonwebtoken) | Authentification sécurisée via token |
| bcryptjs | Hachage sécurisé des mots de passe |
| HTML/CSS/JS | Interface de test locale dans le navigateur |
| Thunder Client | Tests de requêtes backend (optionnel) |

3. Fonctionnement de l’authentification

* Un utilisateur se connecte via /login
* Le serveur vérifie l’identité grâce à bcrypt.compare()
* Si les identifiants sont valides :
  + un **token JWT** est généré via jwt.sign()
  + ce token est envoyé au frontend
  + le token est stocké en localStorage dans le navigateur
* Les routes protégées (/users, /reservations, etc.) exigent un en-tête :

Authorization: Bearer

* Le backend utilise un middleware verifyToken qui vérifie le token avec jwt.verify()
* Si le token est valide → accès autorisé
* Sinon → réponse 401 ou 403

4. Fichier test-api.html – Dashboard navigateur

Une page HTML permet de :

* se connecter via un formulaire
* enregistrer le token JWT
* afficher les **utilisateurs** et **réservations** dans des tableaux dynamiques
* tester l’API sans outil externe

Le script JS utilise fetch() pour envoyer les requêtes avec le token stocké.

5. Gestion des données

Les modèles Mongoose utilisés sont :

* User → contient email, mot de passe hashé, rôle, nom
* Catway → emplacement dans le port
* Reservation → client, date, catway réservé

Des scripts de peuplement (seed) ont été utilisés pour injecter des données réalistes dans la base MongoDB :

npm run seed:users npm run seed:catways npm run seed:reservations

6. Lancement du projet

Pour démarrer localement :

npm start

Le serveur tourne sur http://localhost:3000

7. Tests et validation

* Le fonctionnement du token a été vérifié via la console du navigateur (localStorage.getItem("token"))
* Les routes /users et /reservations retournent bien les données attendues après authentification
* Le terminal serveur affiche le token généré en clair pour validation

8. Déploiement possible

Le projet est prêt pour être hébergé sur des plateformes comme :

* [Render.com](https://render.com/)
* [Railway](https://railway.app/)
* [Vercel](https://vercel.com/) pour le frontend HTML

🧭 Conclusion

Ce projet démontre une parfaite maîtrise d’une API Node.js avec :

* sécurité par token
* gestion des rôles utilisateurs
* interface navigateur interactive
* structure de code claire : routes, contrôleurs, modèles, seeders
* accès maîtrisé aux données protégées

Ce port numérique est prêt à accueillir ses plaisanciers 🤝⚓

Tu veux que je t’aide à exporter ça en Markdown .md pour GitHub, ou à styliser le rapport avec des titres Word formatés ? Je peux aussi le transformer en PDF ou y ajouter ton nom, logo, date, et version 😄📄🛥️