

Documentation du Projet React / Express / MySQL

Généré le : 05/06/2025 22:27:32

Table des matières

- 1. Structure du projet**
- 2. Dépendances installées**
- 3. Authentification**
- 4. Fonctionnalités implémentées**
- 5. API REST (Express)**
- 6. Historique des modifications**

Ø>Ýp Documentation du projet (générée le 18/05/2025 14:50)

Ø=ÜÁ Structure du projet

1. Frontend – hassan-test-react/

- /src
- App.js – Point d'entrée principal, contient le routeur et la logique de navigation.
- index.js – Rend l'application dans le DOM.
- /components – Tous les composants React réutilisables :
- Header.jsx – Barre de navigation supérieure.
- Sidebar.jsx – Barre latérale avec menu.
- Connexion.jsx – Page de connexion.
- Inscription-users.jsx – Formulaire d'inscription utilisateur.
- ListeUtilisateurs.jsx – Liste avec recherche, pagination, actions.
- ModalEditUser.jsx – Modale de modification des utilisateurs.
- MessageBannerGlobal.jsx – Affichage des messages globaux (succès/erreur).
- ConfirmationSupModal.jsx – Modale de confirmation personnalisée.
- PrivateRoute.jsx – Protéger les routes avec authentification.
- Accueil.jsx – Page d'accueil après connexion.
- PageInstruction.jsx – Page d'aide ou en construction.
- /styles – Fichiers CSS :
- App.css, index.css, modal.css, utilisateurs.css, etc.

2. Backend – /backend/

- index.js – Fichier principal Express.
- /routes/ – Routes API (ex: utilisateurs.js).
- /middleware/ – Middleware pour sécuriser les routes via JWT.
- /config/ – Connexion base MySQL.

Ø=Üæ Dépendances installées

Frontend (React) :

axios, react-router-dom, framer-motion

Backend (Express) :

express, mysql2, cors, dotenv, bcrypt, jsonwebtoken

Ø=Ý Authentification

- Token JWT stocké dans localStorage côté client
- Middleware `verifierToken` pour sécuriser les routes backend
- Connexion = génération d'un token / Déconnexion = suppression du token

Ø=Ý' Fonctionnalités implémentées

- ' Inscription utilisateur avec validation mot de passe
- ' Connexion avec JWT
- ' Affichage de la liste des utilisateurs
- ' Modification utilisateur via une modale
- ' Suppression avec modale de confirmation personnalisée
- ' Affichage de messages globaux (succès / erreur)
- ' Protection des routes (PrivateRoute)
- ' Recherche temps réel (nom, prénom, email)
- ' Pagination (5 utilisateurs par page)

Ø=Üá API REST (Express)

GET /api/utilisateurs

- Description : Récupérer la liste de tous les utilisateurs.
- Auth requise : ' Oui (token JWT)

- Headers :

Authorization: Bearer <token>

- Réponse :

```
[  
  {{ "id": 1, "nom": "Dupont", "prenom": "Jean", "login": "jean.dupont@example.com", "date":  
    "2025-05-17T10:00:00.000Z" }}  
]
```

PUT /api/utilisateurs/:id

- Modifier un utilisateur existant

- Auth requise : ' Oui

- Body :

```
{{ "nom": "Durand", "prenom": "Alice", "login": "alice@example.com", "password":  
  "nouveauMotDePasse" }}
```

DELETE /api/utilisateurs/:id

- Supprimer un utilisateur

- Auth requise : ' Oui

POST /api/login

- Authentifier un utilisateur

- Auth requise : 'L Non

- Body :

```
{{ "login": "email@example.com", "password": "motdepasse" }}
```

Ø=ÝS Historique des modifications

- [18/05/2025 14:50] Ajout de la recherche et pagination dans ListeUtilisateurs.jsx

- [2025-05-17] Confirmation stylisée de suppression (modale)

- [2025-05-13] Inscription avec validation frontend

- [2025-05-04] Authentification JWT + Bcrypt