

TD/TP React

JS/TS en Front-End

M. Mohammed BELATAR

m.belatar@emsi.ma

avec Mme. FZ MOUTAI et M. YF EBOBISSE

3ème Année Ingénierie en Informatique et Réseaux

23 décembre 2024

Exercice 1 : Page de Connexion (/login)

- **Remarque** : On va utiliser les APIs déjà créées dans la série d'exercices NodeJS/Express.
- Créer une page **/login** avec un formulaire contenant :
 - Un champ email.
 - Un champ password.
 - Un bouton Se connecter.
- Appeler l'API POST `/api/login` pour valider l'utilisateur.
- Si l'authentification réussit :
 - Sauvegarder le token retourné (par exemple dans `localStorage`).
 - Rediriger vers la page `/dashboard`.
- En cas d'erreur, afficher un message sous le formulaire.

Exercice 2 : Dashboard avec Navigation

- Créer la page **/dashboard**, accessible uniquement avec un token valide.
- Ajouter un menu latéral avec les liens suivants :
 - Lister les utilisateurs.
 - Créer un utilisateur.
- Configurer les sous-routes pour les liens :
 - `/dashboard/users` : Affiche le composant `ListUsers`.
 - `/dashboard/users/:id` : Affiche le composant `EditUser`.
 - `/dashboard/users/create` : Affiche le composant `EditUser` pour créer un nouvel utilisateur.

Exercice 3 : Composant ListUsers

- Créer un composant ListUsers pour afficher les utilisateurs sous forme de tableau.
- Appeler l'API GET `/api/users` pour récupérer les données.
- Chaque ligne du tableau doit contenir :
 - Les colonnes ID, Nom, Email.
 - Un bouton Éditer qui redirige vers `/dashboard/users/:id`.
- Ajouter un bouton Créer User en haut du tableau, qui redirige vers `/dashboard/users/create`.

Exercice 4 : Composant EditUser

- Créer un composant `EditUser` qui affiche les informations d'un utilisateur sous forme de formulaire.
- Pour un utilisateur existant :
 - Appeler l'API `GET /api/users/:id`.
 - Pré-remplir les champs du formulaire (Nom, Email, etc.).
 - Permettre la modification avec l'API `PUT /api/users/:id`.
- Pour créer un nouvel utilisateur :
 - Laisser les champs vides.
 - Utiliser l'API `POST /api/users`.
- Ajouter une validation pour vérifier que tous les champs requis sont remplis.

Exercice 5 : Utilisation de socket.io

- Intégrer socket.io dans votre projet.
- Ajouter une fonctionnalité pour suivre le nombre d'utilisateurs connectés en temps réel.
- Afficher cette information dans le coin supérieur droit du Dashboard.
- Étapes :
 - Configurer socket.io côté serveur.
 - Utiliser socket.io-client pour se connecter côté client.
 - Mettre à jour dynamiquement l'affichage avec le nombre d'utilisateurs connectés.