

Gestion des Tâches

Objectif :

- Mettre en place des composants TaskList et Task.
- Gérer les tâches avec les opérations CRUD directement depuis le front-end, en utilisant le useState.

Étape 1 : Créer le projet

1. Ouvrez un terminal et tapez la commande suivante pour créer un projet React :

```
npx create-react-app task-manager
```

2. Rendez-vous dans le dossier du projet et démarrez le serveur de développement :

```
cd task-manager  
npm start
```

Votre navigateur devrait s'ouvrir avec l'application React par défaut.

Étape 2 : Nettoyer le template par défaut

1. Ouvrez le fichier `src/App.js` et supprimez tout le contenu à l'intérieur de la balise `

`.
2. Supprimez les fichiers suivants dans le dossier `src` pour simplifier le projet : `App.css`, `App.test.js`, `logo.svg`, `reportWebVitals.js`, et `setupTests.js`.
3. Modifiez le fichier `index.js` pour ressembler à :

```
import React from 'react';  
import ReactDOM from 'react-dom';  
import App from './App';  
  
ReactDOM.render(  
  <React.StrictMode>  
    <App />  
  </React.StrictMode>,  
  document.getElementById('root')  
)
```

Étape 3 : Créer le composant TaskList

1. Créez un dossier `components` dans `src` et à l'intérieur, créez un fichier `TaskList.js`.
2. Ajoutez le code suivant dans `TaskList.js` :

```
import React, { useState } from 'react';
import Task from './Task';

function TaskList() {
  const [tasks, setTasks] = useState([]);
  const [newTask, setNewTask] = useState('');

  const handleAddTask = () => {
    if (newTask) {
      setTasks([...tasks, { id: Date.now(), text: newTask }]);
      setNewTask('');
    }
  };

  const handleDeleteTask = (id) => {
    setTasks(tasks.filter(task => task.id !== id));
  };

  return (
    <div>
      <input
        type="text"
        value={newTask}
        onChange={(e) => setNewTask(e.target.value)}
        placeholder="Nouvelle tâche"
      />
      <button onClick={handleAddTask}>Ajouter</button>
      <ul>
        {tasks.map(task => (
          <Task key={task.id} task={task} onDelete={handleDeleteTask} />
        ))}
      </ul>
    </div>
  );
}

export default TaskList;
```

Étape 4 : Créer le composant Task

1. Créez un fichier `Task.js` dans le dossier `components`.
2. Ajoutez le code suivant dans `Task.js` :

```
import React from 'react';

function Task({ task, onDelete }) {
  return (
    <li>
      {task.text}
      <button onClick={() => onDelete(task.id)}>Supprimer</button>
    </li>
  );
}

export default Task;
```

Étape 5 : Mettre à jour App.js

1. Ouvrez le fichier `App.js` et modifiez-le pour utiliser le composant `TaskList` :

```
import React from 'react';

import TaskList from './components/TaskList';

function App() {
  return (
    <div>
      <h1>Gestionnaire de Tâches</h1>
      <TaskList />
    </div>
  );
}

export default App;
```

Étape 6 : Tester l'application

1. Assurez-vous que le serveur de développement est en cours d'exécution (`npm start`).
2. Ouvrez l'application dans le navigateur. Vous devriez voir un champ de texte pour ajouter une nouvelle tâche et un bouton 'Ajouter'.
3. Ajoutez des tâches, supprimez-les et vérifiez que tout fonctionne correctement.