

Fetch, Axios, en React y Express

Julian F. Latorre

Julio 31 de 2024

Índice

1. Introducción	1
2. Configuración del Proyecto	1
2.1. Backend (Express)	1
2.2. Frontend (React)	2
3. Tareas para Completar	4
4. Pistas	4

1. Introducción

En este ejercicio práctico, crearemos una aplicación de lista de tareas (todo list) utilizando React para el frontend y Express para el backend. El ejercicio está diseñado para ser completado paso a paso, con algunas partes que deberás completar por tu cuenta.

2. Configuración del Proyecto

2.1. Backend (Express)

Primero, configuraremos nuestro servidor Express.

1. Crea un nuevo directorio para el proyecto y navega a él:

```
mkdir todo-app
cd todo-app
mkdir backend
cd backend
npm init -y
```

2. Instala las dependencias necesarias:

```
npm install express cors body-parser
```

3. Crea un archivo `server.js` en el directorio `backend` y añade el siguiente código:

```
const express = require('express');
const cors = require('cors');
const bodyParser = require('body-parser');

const app = express();
const port = 5000;

app.use(cors());
app.use(bodyParser.json());

let todos = [];

app.get('/todos', (req, res) => {
  res.json(todos);
});

app.post('/todos', (req, res) => {
  const newTodo = {
    id: Date.now(),
    text: req.body.text,
    completed: false
  };
  todos.push(newTodo);
  res.status(201).json(newTodo);
});

// TODO: Implementa la ruta para actualizar un todo (PUT /
// todos/:id)

// TODO: Implementa la ruta para eliminar un todo (DELETE
// /todos/:id)

app.listen(port, () => {
  console.log(`Server running on http://localhost:${port}`);
});
```

2.2. Frontend (React)

Ahora, configuraremos nuestro frontend React.

1. En el directorio raíz del proyecto, crea una nueva aplicación React:

```
npx create-react-app frontend
cd frontend
```

2. Instala axios para hacer peticiones HTTP:

```
npm install axios
```

3. Reemplaza el contenido de `src/App.js` con el siguiente código:

```
import React, { useState, useEffect } from 'react';
import axios from 'axios';

function App() {
  const [todos, setTodos] = useState([]);
  const [newTodo, setNewTodo] = useState('');

  useEffect(() => {
    fetchTodos();
  }, []);

  const fetchTodos = async () => {
    const response = await axios.get('http://localhost:5000/todos');
    setTodos(response.data);
  };

  const addTodo = async (e) => {
    e.preventDefault();
    await axios.post('http://localhost:5000/todos', { text
: newTodo });
    setNewTodo('');
    fetchTodos();
  };

  // TODO: Implementa la función para actualizar un todo
  // TODO: Implementa la función para eliminar un todo

  return (
    <div>
      <h1>Todo List</h1>
      <form onSubmit={addTodo}>
        <input
          type="text"
          value={newTodo}
          onChange={(e) => setNewTodo(e.target.value)}
          placeholder="Add a new todo"
        />
        <button type="submit">Add</button>
      </form>
      <ul>
        {todos.map((todo) => (
          <li key={todo.id}>
            {todo.text}
            { /* TODO: Añade botones para completar y
eliminar todos */ }
          </li>
        ))}
      </ul>
    </div>
  );
};
```

```
}  
  
export default App;
```

3. Tareas para Completar

Ahora que tenemos la estructura básica de nuestra aplicación, es tu turno de completar algunas funcionalidades:

1. En el backend (`server.js`):
 - Implementa la ruta para actualizar un todo (PUT `/todos/:id`)
 - Implementa la ruta para eliminar un todo (DELETE `/todos/:id`)
2. En el frontend (`App.js`):
 - Implementa la función para actualizar un todo
 - Implementa la función para eliminar un todo
 - Añade botones para completar y eliminar todos en la lista renderizada

4. Pistas

Aquí tienes algunas pistas para ayudarte a completar las tareas:

- Para la ruta PUT en el backend, usa `app.put('/todos/:id', (req, res) =>{ ... })`
- Para la ruta DELETE en el backend, usa `app.delete('/todos/:id', (req, res) =>{ ... })`
- En el frontend, para actualizar un todo, puedes usar `axios.put(http://localhost:5000/todos/$todoId, updatedTodo)`
- Para eliminar un todo en el frontend, puedes usar `axios.delete(http://localhost:5000/todos/$todoId)`