



Challenge

Coding Challenge: User API with NestJS

Kontext

Du arbeitest an einem Projekt, das auf Node.js mit TypeScript und dem NestJS-Framework basiert. Deine Aufgabe ist es, eine einfache API zu erstellen, die Benutzerinformationen aus einer vorhandenen JSON-Datei (`users.json`) abrufen und über einen Endpunkt (`/v1/users`) verfügbar macht. Zusätzlich sollst du Unit Tests schreiben, um die Funktionalität deiner API sicherzustellen.

Anforderungen

1. Projektstruktur:

- Erstelle ein neues NestJS-Projekt oder nutze ein bestehendes Projekt.
- Lege eine Datei `users.json` im `src` -Verzeichnis an, die eine Liste von Benutzern enthält.
- Beispielinhalt von `users.json` :
-

```
1  [  
2    {  
3      "id": 1,  
4      "name": "John Doe",  
5      "email": "john.doe@example.com"  
6    },  
7    {  
8      "id": 2,  
9      "name": "Jane Smith",  
10     "email": "jane.smith@example.com"  
11   }  
12 ]  
13
```

2. Controller:

- Implementiere einen `UserController` , der den Endpunkt `/v1/users` bereitstellt.
- Der Controller soll die Daten aus der `users.json` -Datei lesen und als JSON-Antwort zurückgeben.

3. **Service:**

- Implementiere einen `UserService` , der für das Laden und Verarbeiten der Benutzerdaten aus der JSON-Datei verantwortlich ist.
- Der Service sollte eine Methode bereitstellen, um die Benutzerliste zu erhalten.

4. **Unit Tests:**

- Schreibe Unit Tests für den `UserController` und den `UserService` .
- Die Tests sollten die Funktionalität des Controllers und Services überprüfen, einschließlich:
 - Laden und Rückgabe der Benutzerliste.
 - Fehlerbehandlung, falls die `users.json` -Datei nicht gefunden wird oder ungültige Daten enthält.

5. **Bonus (optional):**

- Implementiere eine zusätzliche Methode, um einen einzelnen Benutzer basierend auf seiner ID abzufragen (`/v1/users/:id`).
- Schreibe auch für diesen Fall Unit Tests.

Abgabe

- Erstelle ein Git-Repository für deine Lösung.
- Stelle sicher, dass alle relevanten Dateien, einschließlich `users.json` , `UserController` , `UserService` und die Unit Tests, im Repository enthalten sind.
- Füge eine `README.md` hinzu, die erklärt, wie man das Projekt startet und die Tests ausführt.