

Getting Started Guide für Node.js, Mocha und Chai

Einleitung

Dieser Getting Started Guide soll dir einen einfachen Einstieg in die Verwendung von Node.js, Mocha und Chai für das Schreiben von Unit Tests geben. Node.js ist eine serverseitige JavaScript-Laufzeitumgebung, die es ermöglicht, JavaScript außerhalb des Browsers auszuführen. Mocha ist ein Testframework für JavaScript-Anwendungen, und Chai ist eine Assertions-Bibliothek, die nahtlos mit Mocha verwendet werden kann, um aussagekräftige Testaussagen zu erstellen.

Schritt 1: Node.js Installation

Bevor du mit Mocha und Chai beginnen kannst, musst du Node.js auf deinem System installieren. Gehe auf die offizielle Node.js-Website (<https://nodejs.org>) und lade die neueste LTS-Version (Long-Term Support) herunter. Folge den Installationsanweisungen für dein Betriebssystem.

Schritt 2: Projektverzeichnis erstellen

Erstelle ein neues Projektverzeichnis, in dem du deine Node.js-Anwendung und deine Tests speichern wirst. Du kannst das Projektverzeichnis manuell erstellen oder die Befehlszeile verwenden:

```
mkdir rechner  
cd rechner
```

Schritt 3: Initialisierung des npm-Projekts

Navigiere in deinem Projektverzeichnis und initialisiere ein neues npm-Projekt mit dem folgenden Befehl:

```
npm init -y
```

Dadurch wird eine `package.json`-Datei erstellt, in der die Abhängigkeiten und Konfigurationen deines Projekts gespeichert werden.

Schritt 4: Installation von Mocha und Chai

Jetzt installiere Mocha und Chai als Entwicklerabhängigkeiten in deinem Projekt:

```
npm install mocha chai --save-dev
```

Durch `--save-dev` werden Mocha und Chai als Entwicklerabhängigkeiten hinzugefügt, da sie nur für die Entwicklung und das Testen benötigt werden.

Schritt 5: Erstellung des Projekts

In diesem Projekt implementieren wir einen Taschenrechner um das Konzept von Unit Tests zu veranschaulichen. Erstelle also eine neue Datei `rechner.js` mit folgendem Inhalt:

```
function add(a, b) {  
  return a + b;  
}
```

```
}

function multiply(a, b) {
  return a * b;
}

function subtract(a, b) {
  return a - b;
}

function divide(a, b) {
  return a / b;
}

module.exports = {
  add,
  multiply,
  subtract,
  divide
};
```

Schritt 6: Erstellung von Testdateien

Erstelle einen Ordner namens "test" in deinem Projektverzeichnis, in dem du deine Testdateien speichern wirst:

```
mkdir test
```

In diesem Ordner kannst du separate Testdateien für verschiedene Teile deiner Anwendung erstellen.

Schritt 7: Schreiben von Unit Tests mit Mocha und Chai

Öffne eine Testdatei in `test/rechner_test.js` in deinem bevorzugten Code-Editor und importiere die benötigten Module:

```
const { expect } = require('chai');
const { add } = require('../rechner.js');
```

Schreibe dann deine Testfälle unter Verwendung der Mocha- und Chai-Syntax:

```
const { expect } = require('chai');
const { add, multiply, subtract, divide } = require( '../rechner');

describe('Teste alle Rechner Funktionen', () => {

  it('Test für Add Funktion', () => {
    // Wenn 1 + 1 gerechnet wird, dann soll das Ergebnis 2 sein
    expect(add(1, 1)).to.equal(2);
  });

});
```

Schritt 8: Ausführen der Tests

Führe die Tests in der Befehlszeile aus, indem du den Befehl `mocha` eingibst:

```
npx mocha
```

Mocha wird die Testdateien im "test"-Ordner erkennen und die Tests ausführen. Du solltest sehen, dass die Tests entweder erfolgreich sind oder Fehlermeldungen anzeigen, wenn etwas nicht wie erwartet funktioniert.

Schritt 9: Ändern der `package.json` Datei

Um die Tests in Zukunft noch einfacher ausführen zu können, macht es Sinn die `package.json` Datei abzuändern und ein Script für das Testen einzufügen. Ändere also daher das Script folgendermaßen ab:

```
"scripts": {  
  "test": "npx mocha"  
}
```

Jetzt kannst du jederzeit mit folgendem Befehl deine Tests ausführen:

```
npm run test
```