

GABRIELE ROMANATO

Introduzione al TDD in Node.js

Il Test-Driven Development (TDD) è una metodologia di sviluppo software che prevede la scrittura dei test prima del codice applicativo. In ambiente Node.js, questa pratica è molto diffusa grazie alla disponibilità di framework di testing come Mocha, Chai e Jest.

Perché usare il TDD

- Favorisce una progettazione del software più modulare e manutenibile
- Aiuta a prevenire bug sin dalle fasi iniziali di sviluppo
- Rende il refactoring del codice più sicuro

Strumenti comuni

Per implementare il TDD in Node.js, i tool più comuni sono:

- **Mocha**: framework di test
- **Chai**: libreria di asserzioni
- **Jest**: framework completo con supporto integrato per asserzioni e mocking

Installazione di Mocha e Chai

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

Aggiungi uno script di test al tuo `package.json`:

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

```
}  
}
```

Esempio pratico: funzione somma

Scriviamo un test per una funzione che somma due numeri:

```
const { expect } = require('chai');  
const sum = require('../sum');  
  
describe('sum', () => {  
  it('dovrebbe restituire la somma di due numeri', () => {  
    expect(sum(2, 3)).to.equal(5);  
  });  
});
```

Ora implementiamo la funzione sum:

```
function sum(a, b) {  
  return a + b;  
}  
  
module.exports = sum;
```

Eseguire i test

Lancia i test con:

```
npm test
```

Conclusione

Il TDD è una pratica potente per migliorare la qualità e l'affidabilità del codice. In Node.js, grazie alla semplicità degli strumenti disponibili, è facile

iniziare e ottenere subito benefici concreti.