

Testes em JavaScript

Framework Jest

Testes de Software

- Conjunto de tarefas que são executadas para verificar se um método está fazendo o que ele deveria fazer.
- Principais benefícios:
 - Identificar e corrigir erros;
 - Melhorar a qualidade do Software;
 - Reduzir custos;
 - Aumentar a confiança do usuário;

Jest

- É um framework de teste de JavaScript moderno e fácil de usar, criado pelo Facebook.
O Jest é conhecido por sua velocidade de execução e sua capacidade de rodar testes paralelamente.
- Funciona com projetos usando: Babel, TypeScript, Node, React, Angular, Vue e etc.
- Ele suporta testes assíncronos e síncronos, e oferece recursos como geração de relatórios, rastreamento de código de origem e cobertura de código.



Aplicações

- **Teste de unidades de código:** Testar funções e módulos individuais em um projeto JavaScript, garantindo que cada parte do código esteja funcionando corretamente antes de integrá-las ao projeto como um todo.
- **Teste de API:** Testar APIs de back-end, verificando se a resposta da API está correta para diferentes aprovações.
- **Teste de interface do usuário:** Testar a interface do usuário de aplicativos da web, simulando cliques do usuário em botões e verificando se a interface do usuário está respondendo corretamente.

Aplicações

- **Teste de integração:** Testar a integração de diferentes partes do código em um projeto. Por exemplo, podemos testar como o front-end e o back-end interagem entre si, ou como diferentes módulos em um projeto se comunicam.
- **Teste de cobertura de código:** Gerar relatório de cobertura de código, que mostra quanto do código do projeto foi testado.

Caso de Uso

- Usando o Jest para testar uma função simples que adiciona dois números:

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

- Primeiro criamos um arquivo chamado "add.test.js":

```
javascript Copy code  
  
const add = require('./add');  
  
test('adds 1 + 2 to equal 3', () => {  
  expect(add(1, 2)).toBe(3);  
});
```

Caso de Uso

- Neste teste, estamos importando a função "add" do arquivo "add.js" e usando o método "test" do Jest para executar o teste. O método "test" recebe dois argumentos: uma descrição do teste e uma função que contém as expectativas.
- Em seguida, estamos usando o método "expect" para verificar se a função "add" retorna o resultado esperado. O método "toBe" verifica se o valor retornado é exatamente igual a 3.
- Ao executar o comando "npm test" na linha de comando, o Jest executará todos os testes no diretório "test" e fornecerá informações sobre os testes passados e falhos.