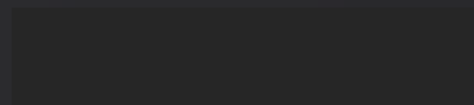


Testes Unitários com Jasmine e Karma

Nesta apresentação, exploraremos os fundamentos dos testes unitários e como utilizá-los efetivamente em projetos de software, usando as ferramentas Jasmine e Karma.



Introdução aos Testes Unitários

1 Definição

Testes unitários focam em validar partes isoladas do código, como funções e métodos.

2 Objetivo

Verificar se cada unidade de código funciona corretamente, de forma independente.

3 Exemplo

Testar se uma função de soma retorna o valor esperado.

Introdução ao Jasmine

O que é Jasmine?

Framework Behavior-Driven Development (BDD) para testes de JavaScript.

Permite escrever testes unitários com sintaxe clara e legível.

Principais Características

- describe, it, expect: Estrutura de teste simples e organizada.
- Testes isolados: Cada teste foca em uma unidade do código.
- Suporte a testes assíncronos.



javascript

```
describe('Exemplo de Teste Jasmine', () => {  
  it('deve retornar a soma correta', () => {  
    const resultado = soma(2, 3);  
    expect(resultado).toBe(5);  
  });  
});
```

Exemplo de Código com Jasmine

Estrutura do Teste

O describe agrupa testes relacionados e o it define um teste específico.

Verificação de Resultados

O expect verifica se o comportamento da função é o esperado.

Introdução ao Karma

O Karma atua como um executor de testes de JavaScript, responsável por executar testes escritos em frameworks como Jasmine. Ele automatiza a execução dos testes, garantindo um ambiente de teste consistente e confiável. Essa funcionalidade é crucial para verificar se o código, além de funcionar isoladamente, também é compatível com diferentes navegadores e plataformas, proporcionando uma análise abrangente da qualidade do código.

O Karma configura um servidor web que executa os testes e fornece feedback instantâneo em tempo real. Ele oferece suporte a uma variedade de navegadores, frameworks de testes e ferramentas de cobertura de código, garantindo que seus testes sejam executados em diferentes ambientes, proporcionando uma visão completa da qualidade do código.

Karma



Suporte a Múltiplos Navegadores

Testa o código em diferentes plataformas e navegadores.



Execução Automática

Executa os testes sempre que há alterações no código.



Integração CI/CD

Ideal para pipelines de entrega contínua.

Integração Jasmine e Karma

Jasmine

Define o comportamento esperado do código através de testes unitários.

Karma

Executa esses testes em diferentes ambientes, automatizando o processo.

Angular

O Angular utiliza Jasmine e Karma para testes unitários, fornecendo uma estrutura completa para você começar a testar. Quando você cria um novo projeto Angular, através da CLI do Angular, já é configurado o executor de testes e testes básicos para garantir que o Angular está funcionando como esperado. Oferecendo uma base sólida para iniciar a criação de testes para suas próprias aplicações!



Métodos do Jasmine

describe

Agrupa os testes relacionados, fornecendo contexto e organização aos testes.

it

Define um caso de teste individual, descrevendo o comportamento esperado do código.

expect

Verifica se o resultado obtido é o esperado, usando matchers para comparar valores e objetos.

spyOn

Monitora a chamada de métodos e verifica seu comportamento, permitindo verificar se um método foi chamado, quantas vezes foi chamado e com quais argumentos.

beforeEach

Executa código antes de cada teste, ideal para inicializar variáveis, objetos ou mockups para cada teste.

afterEach

Executa código depois de cada teste, usado para limpar dados, restaurar o estado inicial ou finalizar recursos.