

Material de apoio

Site: [Geração Tech](#)
Curso: Formação em Desenvolvedor Web - Online
Livro: Material de apoio

Impresso por: JOÃO VITOR DE MELO FREITAS
Data: sábado, 17 ago. 2024, 14:58

Índice

1. Configurando o Cypress

1.1. Vídeo Aula

2. Criando um Teste Simples

2.1. Vídeo Aula

1. Configurando o Cypress

Empresas que utilizam Cypress: Muitas empresas utilizam Cypress para garantir a qualidade de suas aplicações, e você também pode usar essa ferramenta para melhorar o seu fluxo de trabalho.

Configurando o Cypress

Vamos começar configurando o Cypress em nosso projeto:

1. **Criando o `package.json`:** No terminal, execute:

```
bash
```


 Copiar código

```
npm init -y
```

Isso criará um arquivo `package.json` com as configurações básicas.

2. **Instalando o Cypress:** Instale o Cypress como uma dependência de desenvolvimento:

```
bash
```

 Copiar código

```
npm install cypress --save-dev
```

3. **Configurando o Script de Teste:** No arquivo `package.json`, adicione um script para rodar o Cypress:

```
json
```

 Copiar código

```
"scripts": { "cypress:open": "cypress open" }
```

4. **Inicializando o Cypress:** Execute o comando abaixo para abrir o Cypress:

```
bash
```

 Copiar código

```
npm run cypress:open
```

O Cypress criará automaticamente os arquivos de configuração necessários e uma pasta chamada `cypress`, onde ficarão os arquivos de testes.

Escolhendo o Navegador

O Cypress permite que você escolha o navegador no qual os testes serão executados.

Por exemplo, se você costuma trabalhar com o Edge, pode configurar o Cypress para executar os testes em outro navegador, como o Firefox, para diferenciá-los.

Explorando a Interface do Cypress

Quando o Cypress abre, ele mostra um painel com as opções de navegadores disponíveis para executar os testes. Selecione um navegador (por exemplo, o Firefox) e inicie o Cypress. Ele carregará uma tela de administração do Cypress e executará um teste de exemplo.

Se você fechar o navegador de testes, o Cypress detectará isso e permitirá que você selecione outro navegador para continuar os testes.

Testes de Exemplo do Cypress

O Cypress vem com uma série de testes de exemplo que você pode explorar para entender melhor como ele funciona.

Esses exemplos mostram como testar diferentes elementos da interface e como o Cypress pode interagir com eles.

Criando um Teste Simples

Vamos criar um teste simples para acessar o Google e verificar se a página carrega corretamente:

1. **Criando o Teste:** Crie um arquivo dentro da pasta `cypress/e2e` com o seguinte conteúdo:

javascript



```
describe('Acessar Google', () => { it('Deve acessar a página do Google', () => { cy.visit('https://www.google.com'); cy.contains('Google'); }); });
```

2. **Executando o Teste:** Salve o arquivo e volte ao painel do Cypress. Clique no arquivo de teste e observe o Cypress executar o teste automaticamente. Se tudo estiver correto, o teste passará e o Cypress marcará o teste como verde.

Conclusão

Nessa aula, vimos como configurar o Cypress do zero, criar um primeiro teste simples e explorar as funcionalidades básicas da ferramenta.

Na próxima aula, continuaremos explorando o Cypress e aprenderemos a criar testes mais complexos.

Até mais, pessoal! Continuem praticando e configurando seus projetos. Valeu!

1.1. Vídeo Aula

dia 47 cypress 01 converted



2. Criando um Teste Simples

E aí, pessoal! Tudo bem? Vamos continuar nossa aula de Cypress.

Na aula passada, vimos como instalar e configurar o Cypress. Agora, vamos criar um teste simples para testar a página do Google.

Criando um Teste Simples

Vamos criar um teste para acessar o site do Google e realizar uma busca. Aqui está o que faremos:

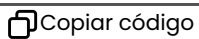
1. Criação do Teste:

- Vamos criar um arquivo de teste no Cypress que acessa o Google e faz uma busca.

2. Estrutura do Teste:

- Primeiro, descrevemos o teste com a função `describe`, que recebe dois parâmetros: uma descrição do teste e uma função `arrow` que contém os passos do teste.
- Dentro do `describe`, usamos a função `it` para definir o que o teste deve fazer. No nosso caso, vamos acessar o Google e fazer uma busca.

javascript



```
describe('Acessando o Google', () => { it('Deve acessar o site e realizar uma busca', () => {  
  cy.visit('https://www.google.com'); cy.get('input[name="q"]').type('Geração Tech{enter}'); }); });
```

Explicando o Código

- `cy.visit()`: Esse comando acessa a URL especificada, neste caso, o site do Google.
- `cy.get()`: Localiza um elemento na página. No nosso exemplo, estamos procurando o campo de busca usando o atributo `name="q"`.
- `type('Geração Tech{enter}')`: Digita "Geração Tech" no campo de busca e simula a tecla "Enter" para iniciar a pesquisa.

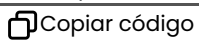
Executando o Teste

Ao salvar o arquivo, o Cypress detecta automaticamente a mudança e executa o teste. O Cypress abrirá o navegador, acessará o Google, buscará por "Geração Tech" e exibirá os resultados da pesquisa.

Interagindo com Elementos na Página

Podemos aprimorar nosso teste clicando em elementos na página:

javascript



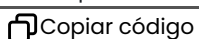
```
cy.contains('Geração Tech').click();
```

- `cy.contains()`: Busca um elemento que contém o texto especificado. Neste exemplo, procuramos um elemento que contém "Geração Tech" e clicamos nele.

Exemplo de Clique em Botão

Vamos supor que queremos clicar em um botão com o ID "inscreva-se". Podemos fazer isso da seguinte maneira:

javascript



```
cy.get('#inscreva-se').click();
```

Se o botão tiver um ID duplicado na página, podemos selecionar o botão específico que queremos clicando em um container ou outro critério mais específico.

Lidando com Erros e Debugging

Se houver um erro durante a execução do teste, o Cypress mostrará uma mensagem indicando o problema. Por exemplo, se dois elementos na página tiverem o mesmo ID, o Cypress exibirá uma mensagem informando que não conseguiu determinar qual elemento deveria ser clicado.

Testando Outros Comandos

O Cypress oferece uma série de comandos para diferentes interações na página. Por exemplo:

- `cy.check()`: Para marcar uma checkbox.
- `cy.clear()`: Para limpar um campo de texto.
- `cy.doubleClick()`: Para realizar um clique duplo.

Esses comandos podem ser usados para testar várias funcionalidades na interface do usuário.

Conclusão

Nessa aula, criamos um teste simples para acessar o Google, realizar uma busca e interagir com a página.

O Cypress nos permite automatizar esses testes de maneira eficiente, garantindo que nossas aplicações funcionem conforme o esperado.

Lembre-se de explorar a documentação do Cypress para aprender mais sobre os comandos disponíveis e como usá-los em seus testes.

Até a próxima, pessoal! Continuem praticando e configurando seus testes com Cypress. Valeu!

2.1. Vídeo Aula

dia 47 cypress 02 converted

