

07 Testando acesso ao Adopet



Transcrição

Nosso teste de visita a uma URL já está rodando e passando. No entanto, a URL que queremos testar a visita não é a de exemplo do Cypress, mas sim a do **Adopet**.

Abrindo o arquivo de testes num editor de código

É importante saber que a interface gráfica do Cypress é um ambiente de **visualização**, para rodarmos os testes, analisá-los passo a passo e identificar eventuais problemas. Para **manipular** esses testes na forma escrita, precisaremos de um **editor de código**.

No mercado de trabalho, sempre escrevemos nossos testes em um editor de código!

O Cypress facilita esse processo para nós. Quando passamos o cursor por cima do arquivo `spec.cy.js` na interface gráfica do Cypress, é mostrada uma pequena mensagem de aviso "*Open in IDE*". Ou seja, o arquivo será aberto em um editor de código.

E, de fato, clicando no arquivo, ele é aberto no nosso editor de código local, como o VS Code.

Ao fazer isso, o VS Code exibe numa janela modal algumas informações sobre o projeto aberto. No caso da instrutora, a mensagem informa que foi encontrado um repositório *Git* no diretório raiz do arquivo em questão. Podemos fechar essa mensagem e focar no código do Cypress.

Somente o código do teste foi aberto, mas queremos abrir todo o projeto. Para fazer isso, podemos utilizar o atalho "**Ctrl + Shift + E**". Com isso, uma aba é expandida à esquerda da tela, exibindo o explorador de arquivos ("*Explorer*").

Essa aba terá uma mensagem dizendo que ainda não abrimos nenhuma pasta. Podemos selecionar a opção "Open Folder" (ou "Abrir diretório") para abrir a pasta do projeto na nossa máquina.

O nosso arquivo de testes está no caminho "**Área de trabalho > curso_cypress**", conforme verificamos no explorador de documentos que foi aberto.

Vamos selecionar essa pasta e, com isso, o *VS Code* vai trazer para a aba Explorador toda a **estrutura de pastas** do nosso projeto, contendo os diretórios que criamos no início do nosso curso, com as configurações do ambiente:

CURSO_CYPRESS

- cypress
 - e2e
 - spec.cy.js
 - fixtures
 - support
- node_modules
 - cypress.config.js
 - package-lock.json
 - package.json

Vamos acessar o arquivo `package.json`, que faz parte do nosso projeto criado via NPM.

`package.json`

```
{  
  "name": "curso-cypress",  
  "version": "1.0.0",  
  "description": "",  
  "main": "index.js",  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1",  
  },  
  "author": "",  
  "license": "ISC",  
  "devDependencies": {  
    "cypress": "13.6.4"  
  }  
}
```

COPIAR CÓDIGO

Nesse código, podemos identificar que o Cypress foi instalado na versão 13.6.4 e foi configurado como uma *Dev Dependency*, ou seja, uma **dependência de desenvolvimento**. Isso significa que o Cypress é utilizado neste projeto não como uma ferramenta obrigatória para fazer o Adopet funcionar, por exemplo.

Uma dependência de desenvolvimento é útil e nos ajuda a trabalhar de diferentes formas, mas **não é essencial** para que o projeto funcione. Neste caso, por exemplo, poderíamos utilizar outro *framework* para realizar os testes E2E, mas escolhemos o Cypress.

Primeiro teste da Adopet: editando o arquivo de testes

Vamos voltar para o arquivo `spec.cy.js`. O que precisamos fazer, então, para que o Cypress faça a visita no nosso Adopet?

Na linha 3 do código, temos o método `cy.visit()` com a URL de exemplo passada como parâmetro. Vamos substituir essa

URL de exemplo pela URL do Adopet:

```
spec.cy.js
```

```
describe('template spec', () => {  
  it('passes', () => {  
    cy.visit('https://adopet-frontend-cypress.ver  
  })  
})
```

COPIAR CÓDIGO

Vamos salvar o arquivo.

Antes de rodar o novo teste, precisamos **interromper o Cypress**. Para isso, vamos abrir o nosso terminal e pressionar o atalho "Ctrl + C" para fechá-lo.

Vamos rodar de novo o comando seguinte para reabrir o Cypress:

```
npx cypress open
```

[COPIAR CÓDIGO](#)

De volta à interface gráfica do Cypress, vamos verificar o que vai acontecer agora com o nosso teste.

Clicamos em "**E2E Testing > Chrome > Start E2E Testing in Chrome**" para inicializar o teste E2E no Chrome.

Na nova tela, temos uma lista com os nossos arquivos na aba esquerda. Vamos clicar em `spec.cy.js`, o arquivo em que está o nosso teste. Ao fazer isso, visualizamos a página da Adopet na aba direita da página! Isso significa que nosso teste passou e conseguimos visitar a página desejada.

No próximo vídeo, vamos testar mais cenários e casos de testes. Esperamos você lá!