

Autenticação

Conceito

Quando precisamos executar vários testes que dependem de um usuário logado, refazer o login em todos eles pode deixar a suíte mais lenta e redundante.

O **Playwright** oferece o recurso de **storage state**, que salva cookies e `localStorage` em um arquivo. Assim, é possível reutilizar a sessão logada entre execuções, evitando ter que refazer o login.

Explicação

O método:

```
context.storage_state(path="state.json")
```

→ Salva o estado atual de cookies e localStorage no arquivo `state.json`.

E:

```
browser.new_context(storage_state="state.json")
```

→ Cria um contexto já logado, reutilizando o arquivo salvo.

Essa técnica é muito útil para:

- Testes que exigem autenticação.
- Evitar múltiplos logins desnecessários.
- Aumentar a velocidade da execução.

Implementação no `conftest.py`

```
import os

STORAGE_FILE = "playwright/auth/state.json"

@pytest.fixture(scope='session')
def contexto(browser):
```

```

if os.path.isfile(STORAGE_FILE):
    contexto = browser.new_context(
        base_url='https://automationexercise.com/',
        record_video_dir='videos',
        storage_state=STORAGE_FILE
    )
else:
    contexto = browser.new_context(
        base_url='https://automationexercise.com/',
        record_video_dir='videos'
    )

yield contexto

os.makedirs(os.path.dirname(STORAGE_FILE), exist_ok=True)

if not os.path.isfile(STORAGE_FILE):
    contexto.storage_state(path=STORAGE_FILE)

contexto.close()

```

Exemplo de teste de login que gera e usa o storage_state

```

from playwright.sync_api import expect
from pages.cadastro_login import CadastroLogin
from pages.produtos import Produtos
from pages.carrinho import Carrinho

def teste_login(page):
    login = CadastroLogin(page)
    login.acessar_cadastro_login()
    login.fazer_login(email='teste@testeabcde.com', senha='123456789')
    expect(page.get_by_role("link", name="Logout")).to_be_visible()

def test_adicionar_produtos_ao_carrinho(page):
    produtos = Produtos(page)
    produtos.acessar_produtos()

```

```

produtos.adicionar_produto_ao_carrinho(indice_produto='0')
produtos.botao_continuar_comprando.click()
produtos.adicionar_produto_ao_carrinho(indice_produto='1')
produtos.botao_continuar_comprando.click()
page.pause()

def test_validar_carrinho(page):
    carrinho = Carrinho(page)
    carrinho.acessar_carrinho()
    carrinho.validar_carrinho(indice_produto='0',
        cabecalho_descricao_produto='Blue Top',
        descricao_produto='Women > Tops',
        preco_produto='Rs. 500',
        preco_total_produto='Rs. 500')
    carrinho.validar_carrinho(indice_produto='1',
        cabecalho_descricao_produto='Men Tshirt',
        descricao_produto='Men > Tshirts',
        preco_produto='Rs. 400',
        preco_total_produto='Rs. 400')
    page.pause()

```

Fluxo de uso

1. Na primeira execução:

- O arquivo `state.json` não existe.
- O teste de login é executado normalmente.
- Ao final da sessão, o estado é salvo.

2. Nas execuções seguintes:

- O arquivo `state.json` é encontrado.
- O contexto já é criado com sessão logada.
- Você pode acessar páginas autenticadas sem refazer o login.

Referência oficial:

🔗 <https://playwright.dev/python/docs/auth>