Playwright

Assertions: Um guia completo

Preparado e traduzido por:

João Vitor de Souza
Tech Lead

Playwright: Dominando o Essencial

Dominando as Assertions no Playwright: Um Guia Completo

O Playwright oferece capacidades de assertions poderosas através da função expect, facilitando a validação dos seus testes. Para fazer uma assertion, chame expect(value) e selecione um matcher que corresponda à sua expectativa. Matchers genéricos como toEqual, toContain e toBeTruthy permitem afirmar várias condições de forma eficaz.

Exemplo:

```
await expect(page).toHaveTitle(/Playwright/);
```

Matchers Assíncronos Específicos para Web

O Playwright também fornece matchers assíncronos específicos para web, que esperam que as condições esperadas sejam atendidas. Considere o exemplo abaixo:

```
await expect(page.locator('data-testid=status')).toHaveText('Submitted');
```

Nesse caso, o Playwright irá continuamente buscar o elemento com o test ID de status e verificar seu texto até que ele corresponda a "Submitted" ou até que o tempo limite seja atingido.

Por padrão, os tempos limites de assertions são definidos para 5 segundos.

Assertions com Auto-Retry

As seguintes assertions irão tentar novamente até que passem ou o tempo limite seja atingido. Como essas assertions são assíncronas, sempre use await.

Assertion	**Descrição**
await expect(locator).toBeAttached()	O elemento está anexado
await expect(locator).toBeChecked()	A caixa de seleção está marcada
await expect(locator).toBeDisabled()	O elemento está desativado
await expect(locator).toBeEditable()	O elemento é editável
await expect(locator).toBeEmpty()	O contêiner está vazio
await expect(locator).toBeEnabled()	O elemento está ativado

Assertion	Descrição
await expect(locator).toBeFocused()	O elemento está focado
await expect(locator).toBeHidden()	O elemento não está visível
await expect(locator).toBeInViewport()	O elemento cruza a viewport
await expect(locator).toBeVisible()	O elemento está visível
await expect(locator).toContainText()	O elemento contém texto
await expect(locator).toHaveAccessibleDescription()	O elemento tem uma descrição acessível correspondente
await expect(locator).toHaveAccessibleName()	O elemento tem um nome acessível correspondente
await expect(locator).toHaveAttribute()	O elemento tem um atributo DOM
await expect(locator).toHaveClass()	O elemento tem uma propriedade de classe
await expect(locator).toHaveCount()	A lista tem um número exato de filhos
await expect(locator).toHaveCSS()	O elemento tem uma propriedade CSS
await expect(locator).toHaveId()	O elemento tem um test ID
await expect(locator).toHaveJSProperty()	O elemento tem uma propriedade JavaScript
await expect(locator).toHaveRole()	O elemento tem uma função ARIA específica
await expect(locator).toHaveScreenshot()	O elemento tem uma captura de tela
await expect(locator).toHaveText()	Elemento corresponde ao texto
await expect(locator).toHaveValue()	A entrada tem um valor
await expect(locator).toHaveValues()	Selecionar tem opções selecionadas
await expect(page).toHaveScreenshot()	Page tem uma captura de tela
await expect(page).toHaveTitle()	Page tem um título
await expect(page).toHaveURL()	Page tem um URL
await expect(response).toBeOK()	A resposta tem um status OK

Assertions Sem Auto-Retry

Essas asserções testam as condições sem retrying. Tenha cuidado ao usá-los em testes da web, pois eles podem causar resultados flaky se a página carregar de forma assíncrona.

Prefira asserções de auto-retrying, mas para casos complexos que precisam de retrying, use expect.poll ou expect.toPass.

Assertion	**Descrição**
expect(value).toBe()	O valor é o mesmo
expect(value).toBeCloseTo()	O número é aproximadamente igual
expect(value).toBeDefined()	O valor não é indefinido
expect(value).toBeFalsy()	O valor é falso (por exemplo, falso, 0, nulo)
expect(value).toBeGreaterThan()	O número é maior que
expect(value).toBeGreaterThanOrEqual()	O número é maior ou igual
expect(value).toBeInstanceOf()	O número é maior ou igual
expect(value).toBeLessThan()	O número é menor que
expect(value).toBeLessThanOrEqual()	O número é menor ou igual
expect(value).toBeNaN()	O valor é NaN
expect(value).toBeNull()	O valor é nulo
expect(value).toBeTruthy()	O valor é verdadeiro (por exemplo, não falso, 0, nulo)
expect(value).toBeUndefined()	O valor é indefinido
expect(value).toContain()	String contém uma substring
expect(value).toContainEqual()	Array ou conjunto contém um elemento semelhante
expect(value).toEqual()	O valor é semelhante – igualdade profunda e correspondência de padrões
expect(value).toHaveLength()	Array ou string tem comprimento
expect(value).toHaveProperty()	Objeto tem uma propriedade
expect(value).toMatch()	String corresponde a uma expressão regular
expect(value).toMatchObject()	O objeto contém propriedades especificadas
expect(value).toStrictEqual()	O valor é semelhante, incluindo tipos de propriedade
expect(value).toThrow()	Função gera um erro
expect(value).any()	Corresponde a qualquer instância de uma classe/primitiva
expect(value).anything()	Combina com qualquer coisa

Assertion	Descrição
expect(value).arrayContaining()	Array contém elementos específicos
expect(value).closeTo()	O número é aproximadamente igual
expect(value).objectContaining()	Objeto contém propriedades específicas
expect(value).stringContaining()	String contém uma substring
expect(value).stringMatching()	String corresponde a uma expressão regular

Negando Matchers

Inverta expectativas usando .not para verificar o oposto

```
await expect(locator).not.toBeVisible();
await expect(locator).not.toHaveText('Incorrect Text');
```

Soft Assertions

As soft assertions não terminam a execução do teste em caso de falha, mas marcam o teste como falho. Isso permite que você continue testando mesmo que algumas verificações falhem.

Exemplo:

```
await expect.soft(page.locator('input')).toBeVisible();
```

Você pode verificar falhas em soft assertions a qualquer momento:

```
expect(test.info().errors).toHaveLength(0);
```

Nota: Soft assertions só funcionam com o **Playwright test runner**.

Mensagens de Expectativas Customizadas

Você pode adicionar mensagens customizadas às suas assertions para melhor contexto nos relatórios:

```
await expect(page.getByText('Name'), 'should be logged in').toBeVisible();
```

Mensagens customizadas são visíveis tanto em assertions que passam quanto nas que falham, tornando-as inestimáveis para depuração.

Soft assertions também oferecem suporte a mensagens personalizadas:

Exemplo:

```
expect.soft(value, 'my soft assertion').toBe(56);
```
