

Testing con Cypress

Funciones y propiedades

Julian F. Latorre

August 6, 2024

`cy.get()`

`cy.get('selector')`

- Obtiene uno o más elementos DOM que coinciden con el selector.
- Ejemplos de selectores:
 - 'button ': Selecciona por etiqueta
 - '.clase ': Selecciona por clase CSS
 - '#id ': Selecciona por ID
 - '[atributo= " valor "] ': Selecciona por atributo
- Se puede encadenar con otras funciones.

cy.should

```
cy.should( assertion , [valor] )
```

- Crea una aserción sobre el elemento o elementos seleccionados.
- Aserciones comunes:
 - 'exist': Verifica que el elemento existe
 - 'be.visible': Verifica que el elemento es visible
 - 'have.text': Verifica el texto del elemento
 - 'have.attr': Verifica un atributo del elemento
- Se puede encadenar múltiples aserciones con .and()

`cy.click()`

`cy.click()`

- Simula un clic en el elemento seleccionado.
- Se puede usar con opciones: `.click(force: true)`
- Útil para botones, enlaces y elementos interactivos.

`cy.type()`

`cy.type('texto')`

- Simula la entrada de texto en un elemento.
- Comúnmente usado con campos de entrada y áreas de texto.
- Opciones adicionales: `.type('texto', delay: 100)`

cy.visit()

```
cy.visit( 'https://ejemplo.com' )
```

- Navega a una URL específica.
- Generalmente usado al inicio de una prueba.
- Puede incluir opciones como timeout.

cy.intercept()

```
cy.intercept( 'METHOD', 'path', response )
```

- Intercepta peticiones de red.
- Útil para espiar, stubear o modificar respuestas de API.
- Se puede usar con `.as()` para crear un alias.

`cy.wait()`

`cy.wait('@alias')`

`cy.wait(1000)`

- Espera a que se complete una petición interceptada.
- También puede esperar un tiempo específico en milisegundos.
- Útil para sincronizar pruebas con operaciones asíncronas.

`cy.contains()`

`cy.contains('texto')`

- Obtiene el primer elemento DOM que contiene el texto especificado.
- Puede combinarse con un selector: `cy.contains('button', 'Enviar')`
- Útil para encontrar elementos por su contenido visible.

`cy.url()`

`cy.url()`

- Obtiene la URL actual de la página.
- A menudo se usa con `.should()` para verificar navegación.
- Ejemplo: `cy.url().should('include', '/pagina'`)

cy.check() y cy.uncheck()

```
cy.check()  
.uncheck()
```

- `check()`: Marca una casilla de verificación o botón de radio.
- `uncheck()`: Desmarca una casilla de verificación.
- Se puede usar con múltiples elementos a la vez.

`cy.select()`

`cy.select('opción')`

- Selecciona una opción en un elemento `<select>`.
- Puede seleccionar por valor, texto o índice.
- Ejemplo: `cy.get('select').select('opción2')`

cy.submit()

cy.submit()

- Envía un formulario.
- Se aplica al elemento `<form>` o a cualquier elemento dentro del formulario.
- Ejemplo: `cy.get('form').submit()`

cy.viewport()

cy.viewport(width, height)

- Establece el tamaño y la orientación del viewport.
- Útil para probar diseños responsivos.
- Ejemplo: `cy.viewport('iphone-6', 'landscape')`

cy.scrollTo()

cy.scrollTo('bottom')

- Desplaza la ventana o un elemento a una posición específica.
- Opciones incluyen 'top', 'center', 'bottom' o coordenadas.
- Útil para interactuar con elementos fuera de la vista.

cy.request()

```
cy.request( 'POST', '/api/users',  name: 'Usuario' )
```

- Realiza una petición HTTP.
- Útil para configurar el estado de la prueba o verificar APIs.
- Soporta todos los métodos HTTP y opciones personalizadas.

Funciones de Debugging

`cy.debug()`

`cy.pause()`

- `debug()`: Pausa la ejecución y permite inspeccionar el estado.
- `pause()`: Pausa la ejecución y permite reanudar paso a paso.
- Útiles para depurar pruebas complejas.

Aserciones: exist y not.exist

```
.should( 'exist')  
.should( 'not.exist')
```

- 'exist': Verifica que el elemento está en el DOM.
- 'not.exist': Verifica que el elemento no está en el DOM.
- Ejemplo: `cy.get('h1').should('exist')`

Aserciones: `be.visible` y `be.hidden`

```
.should( 'be.visible')  
.should( 'be.hidden')
```

- `'be.visible'`: Verifica que el elemento es visible.
- `'be.hidden'`: Verifica que el elemento está oculto.
- Ejemplo: `cy.get('div').should('be.visible')`

Aserciones: `be.enabled` y `be.disabled`

```
.should( 'be.enabled')  
.should( 'be.disabled')
```

- `'be.enabled'`: Verifica que el elemento está habilitado.
- `'be.disabled'`: Verifica que el elemento está deshabilitado.
- Ejemplo: `cy.get('input').should('be.enabled')`

Aserciones: have.text y contain

```
.should( 'have.text', 'Texto exacto')
```

```
.should( 'contain', 'Texto parcial')
```

- 'have.text': Verifica el texto exacto del elemento.
- 'contain': Verifica si el elemento contiene el texto especificado.
- Ejemplo: `cy.get('p').should('contain', 'Bienvenido')`

Aserciones: `have.class`

```
.should( 'have.class', 'clase-css' )
```

- Verifica si el elemento tiene la clase CSS especificada.
- Ejemplo: `cy.get('div').should('have.class', 'activo')`

Aserciones: `have.attr`

```
.should( 'have.attr', 'atributo', 'valor')
```

- Verifica si el elemento tiene el atributo con el valor especificado.
- Ejemplo: `cy.get('a').should('have.attr', 'href', 'https://ejemplo.com')`

Aserciones: `have.value`

```
.should( 'have.value', 'valor' )
```

- Verifica el valor de un elemento de formulario.
- Comúnmente usado con `inputs` y `textareas`.
- Ejemplo: `cy.get('input').should('have.value', 'texto ingresado')`

Aserciones: `be.checked`

```
.should( 'be.checked')  
.should( 'not.be.checked')
```

- Verifica si un checkbox o radio button está marcado o no.
- Ejemplo: `cy.get('input[type="checkbox"]').should('be.checked')`

Aserciones: `have.length`

```
.should( 'have.length', numero )
```

- Verifica el número de elementos que coinciden con el selector.
- Ejemplo: `cy.get('li').should('have.length', 5)`

Aserciones: `be.empty` y `not.be.empty`

```
.should( 'be.empty')  
.should( 'not.be.empty')
```

- Verifica si un elemento está vacío o no.
- Útil para elementos de formulario o contenedores.
- Ejemplo: `cy.get('textarea').should('be.empty')`

Aserciones: `be.focused`

```
.should( 'be.focused' )  
.should( 'not.be.focused' )
```

- Verifica si un elemento tiene el foco.
- Útil para probar la navegación por teclado.
- Ejemplo: `cy.get('input').should('be.focused')`

Aserciones: have.css

```
.should( 'have.css', 'propiedad', 'valor')
```

- Verifica el valor de una propiedad CSS del elemento.
- Ejemplo: `cy.get('button').should('have.css', 'background-color', 'rgb(0, 123, 255)')`

Aserciones Numéricas

```
.should( 'be.gt', numero )  
.should( 'be.gte', numero )  
.should( 'be.lt', numero )  
.should( 'be.lte', numero )
```

- 'be.gt': Mayor que
- 'be.gte': Mayor o igual que
- 'be.lt': Menor que
- 'be.lte': Menor o igual que
- Ejemplo: `cy.get('.items').its('length').should('be.gte', 3)`

Combinando Aserciones

```
.should( 'be.visible')  
.and( 'contain', 'Texto')  
.and( 'have.class', 'activo')
```

- Se pueden encadenar múltiples aserciones con .and()
- Útil para verificaciones complejas en un solo elemento