**Handling Dropdowns**

select() in Cypress takes 3 kinds of argument - text, value attribute and Index.

select() is overloaded. -

**Method overloading** - Developing multiple methods with the same name but variations in the argument list is called Method overloading.

Cypress has overloaded select() with three kinds of arguments - text, value and index.

1. Select by Value

cy.get('#dropdown-class-example').select('option2').should('have.value','option2')

1. Select by Text

cy.get('#dropdown-class-example').select('Option3').should('have.text','Option3');

1. Select by index

cy.get('#dropdown-class-example').select(2).should('have.value','option2');

### **4.3. Handling Dropdowns**

Dropdowns can be either static (predefined options) or dynamic (options generated based on user input or other actions).

#### **4.3a Static Dropdowns**

Static dropdowns are usually implemented using the <select> tag in HTML.

<select id="staticDropdown">

<option value="option1">Option 1</option>

<option value="option2">Option 2</option>

<option value="option3">Option 3</option>

</select>

Cypress :

// Select an option by value

cy.get('#staticDropdown').select('option2')

// Select an option by text

cy.get('#staticDropdown').select('Option 2')

// Assert the selected option

cy.get('#staticDropdown').should('have.value', 'option2')

#### **4.3bDynamic Dropdowns**

Dynamic dropdowns populate their options based on user input or other actions, often implemented using a combination of <input> tags and JavaScript.

// Type into the input to trigger dynamic dropdown

cy.get('#dynamicDropdown').type('Option')

// Wait for options to be populated and select an option

cy.get('#dropdownOptions').contains('Option 2').click()

### **4.4. Handling Checkboxes**

Checkboxes are used to select or deselect options.

**Example:**

// Check the checkbox

cy.get('#checkbox1').check()

// Uncheck the checkbox

cy.get('#checkbox1').uncheck()

// Assert the checkbox is checked

cy.get('#checkbox1').should('be.checked')

// Assert the checkbox is unchecked

cy.get('#checkbox1').should('not.be.checked')

### **3. Handling Select Elements**

Select elements are a type of dropdown typically used for long lists of options.

// Select an option by value

cy.get('#selectElement').select('value2')

// Select an option by text

cy.get('#selectElement').select('Value 2')

// Assert the selected option

cy.get('#selectElement').should('have.value', 'value2')

**Test case combining all**

describe('Form Handling', () => {

it('Handles dropdowns, checkboxes, selects, and dynamic dropdowns', () => {

cy.visit('https://example.com/form')

// Handle static dropdown

cy.get('#staticDropdown').select('option2')

cy.get('#staticDropdown').should('have.value', 'option2')

// Handle checkbox

cy.get('#checkbox1').check().should('be.checked')

cy.get('#checkbox1').uncheck().should('not.be.checked')

// Handle select element

cy.get('#selectElement').select('value2')

cy.get('#selectElement').should('have.value', 'value2')

// Handle dynamic dropdown

cy.get('#dynamicInput').type('Option')

cy.get('#dynamicDropdown').contains('Option 2').click()

})

})

describe('dropdown tests', () => {

it('static dropdown test- by value attribute', () => {

cy.visit('https://rahulshettyacademy.com/AutomationPractice/');

// handle the dropdown using value attribute

cy.get('#dropdown-class-example').select('option3').should('have.value','option3');

cy.wait(1000)

cy.get('#dropdown-class-example').select('option2').should('have.value','option2');

cy.wait(1000)

cy.get('#dropdown-class-example').select('option1').should('have.value','option1');

cy.wait(1000)

cy.get('#dropdown-class-example').select('').should('have.value','');

});

it('Select - by text', () => {

cy.visit('https://rahulshettyacademy.com/AutomationPractice/');

// handle the dropdown using value attribute

cy.get('#dropdown-class-example').select('Option1').should('have.value','option1');

cy.get('#dropdown-class-example').select('Option2').should('have.value','option2');

cy.get('#dropdown-class-example').select('Option3').should('have.value','option3');

cy.get('#dropdown-class-example').select('Select').should('have.value','');

});

it('using index', () => {

cy.visit('https://rahulshettyacademy.com/AutomationPractice/');

// handle the dropdown using value attribute

cy.get('#dropdown-class-example').select(3).should('have.value','option3');

cy.get('#dropdown-class-example').select(2).should('have.value','option2');

cy.get('#dropdown-class-example').select(1).should('have.value','option3');

cy.get('#dropdown-class-example').select(0).should('have.value','');

});

it.skip('Dynamic dropdown test', () => {

cy.visit('https://rahulshettyacademy.com/AutomationPractice/');

// Step 1 - locator for suggestion class and then type the keyword

cy.get('[placeholder="Type to Select Countries"]').type('ind');

// using find(), we are finding the locator which results in options of dropdown(children)

// iterating those options using each()

// if the element is matching the text, we are clicking on the option

cy.get('#ui-id-1').find('.ui-menu-item').each(($ele)=>{

if ($ele.text()=='Indonesia') {

cy.wrap($ele).click();

}

})

});

it.only('dynamic -2', () => {

cy.visit('https://rahulshettyacademy.com/AutomationPractice/');

// Step 1 - locator for suggestion class and then type the keyword

cy.get('[placeholder="Type to Select Countries"]').type('Aus');

// using find(), we are finding the locator which results in options of dropdown(children)

// iterating those options using each()

// if the element is matching the text, we are clicking on the option

cy.get('.ui-menu-item').each(($ele)=>{

if ($ele.text()=='Australia') {

cy.wrap($ele).click();

}

})

cy.get('[placeholder="Type to Select Countries"]').clear().type('ch'); // clear the textbox

// Selecting a dynamic dropdown using contains()

cy.get('.ui-menu-item').contains('China').click()

});

});