Cypress Training

By OSCAR

Lesson 2

LESSON 2: Elements, Locators

- DOM Elements
- Cypress Locators
- Types of Cypress Assertions

CYDEO

DOM Elements

- In simple terms, any web page that you encounter is a document. Viewing the document as an HTML source or in a browser window is just an approach of presenting one thing in different ways. T
- he document object model (DOM) represents a webpage as objects to enable its manipulation. With the document-oriented model, a scripting language such as JavaScript may modify it by targeting the objects.
- When reading HTML code, it is important to differentiate elements, such as tags, attribute names, attribute values etc.

CYDEO

What are Cypress Locators?

- A Selector or Locator is an object that finds and returns web page items/elements on a page based on the query.
- In Automation, to perform any operation on the web elements first, you need to locate an element and perform an action on that element.
- The Locator or Selector helps to locate an element in the webpage. There are different types of locators, such as id, CSS, XPath, tag-based selectors, etc.

CYDEO

What are Cypress Locators?

- Cypress supports various locators such as tags, id, class, attributes, text, etc. Cypress also supports XPath Selectors; however, it needs installation of the third-party plugin cypress-xpath. Xpath has not been supported out of the box in the Cypress framework.
- To fetch the HTML element in Cypress using different types of locators cy.get() method is used.

CYDEO

Why do we need Locators?

- We need Locators in order to manipulate elements on the page, to interact with them as user would manually do.
- For example: in order to click on certain element, or type a text, you have to first locate exactly which element that is on the page.
- Giving what we know so far, let's put this into action.
  - Create another test file under /e2e/intro folder and call it for example locators.cy.js
  - Locators: Get elements by different locator strategies
  - Navigate to test page: /login

CYDEO

Types of Cypress Assertions

- Cypress integrates multiple assertions from various JS assertion libraries such as Chai, jQuery, etc.
- We can broadly classify all of these assertions into two segments based on the subject on which we can invoke them:
  - Implicit Assertions: When the assertion applies to the object provided by the parent chained command, it's called an Implicit assertion.
  - Explicit Assertions: When there is a need to pass an explicit subject for the assertion, it falls under the category of Explicit assertion.

CYDEO

Implicit Assertions

- This category of assertions generally includes commands such as ".should()" and ".and()". As these commands don't stand independently and always depends on the previously chained parent command, they automatically inherit and acts on the object yielded by the previous command.
- Generally, we use Implicit assertions when we want to:
  - Assert multiple validations about the same subject.
  - Alter the subject before making the assertions on the subject.

CYDEO

Explicit Assertions

- This category of assertions contains the commands such as "expect()" and "assert()", which allow you to pass an explicit subject/object.
- Generally, you will be using "Explicit assertions" when you want to:
  - Perform some custom logic before making the assertions on the given subject.
  - Perform multiple assertions against the same subject after applying custom logic.

CYDEO