https://vatgia.com/hoidap/4608/281876/hoi-ve-tester-va-developer.html#google\_vignette

Trang này sẽ cho biết một số trends về npm <https://npmtrends.com>

Topics coverd

1. cơ bản về javascripts

2. cách test tự động vs cypress

3. API vs cypress

4. xây dựng cypress frameworks vs mocha và cucumber

5. intercepting network responses và browses với cypress

6. tự động test database với cypress

7. single sign on và accessibility automation với cypress

Dùng cypress có thể unit testing, intergration testing, end – to end testing,

đầu tiên cần phải install visual studio code đã

install nodejs trước đã : -> vs MAC

sau đó phải add PATH vào file của nó

nó nằm ở chỗ này

nếu gõ vim ~/.bash\_profile

mà vẫn không ra gì thì phải add path vào nó export PATH=$PATH:/usr/local/bin

install nodejs trước đã : -> vs WIN

cài nodejs

Sau đó ở trong Environment Variables -> phần System Variables -> tạo New – Trên Variable điền nodejs Dưới Variable value điền C:\Program Files\nodejs

Để check vào cmd gõ lệnh node -v

để ghi lại thì phải làm như sau

ấn esc -> ấn :wq -> ấn ENTER

Tạo folder CypressAutomation trong visualstudio

Dùng lệnh npm -i init để intall -> nó sẽ có một số bước để xác nhận -> ok hết

Dùng tiếp lệnh npm install cypress –save-dev để install cypress

Trong đó đã lên đến "cypress": "^13.7.1"

Bài 10

muốn chạy cypress chúng ta chạy lệnh node\_modules/.bin/cypress open

trên windows chúng ta dùng 2 lệnh như sau để chạy cypress

npx cypress open

node\_modules\.bin\cypress.cmd open

Nó có 2 cái E2E testing vs cả Component Testing

Bây giờ chọn cái E2E testing để test

Tạo một folder trong folder cypress tên là integration/example/Test1.js

Các file như trên

File test1.js nó ở đây

<https://github.com/cypress-io/cypress-example-kitchensink/blob/master/cypress/e2e/2-advanced-examples/actions.cy.js>

Sau đó phải sửa file cypress.config.js phía dưới dấu }, dưới // gõ cái này

specPattern: ‘cypress/integration/examples/\*.js’

chạy cypress chúng ta chạy lệnh node\_modules/.bin/cypress open

Đợi nó cài thêm một số cái react gì đó

Trang dùng để test đầu tiên test thử sẽ là trang này

<https://example.cypress.io/commands/actions>

cho nên phải sửa cái dòng code https:510232 gì gì đó ở trong cái Test1.js thành cái dòng kia để nó test thử

btw sửa nó thành ra như thế này

beforeEach(() => {

cy.visit('https://example.cypress.io/commands/actions')

})

muốn chạy cypress chúng ta chạy lệnh node\_modules/.bin/cypress open

Chọn cái E2E testing

Đối với windows:

Để có thể chạy được các file test cần phải sửa file cypress.config.js thành như sau:

const { defineConfig } = require("cypress");

module.exports = defineConfig({

e2e: {

setupNodeEvents(on, config) {

// implement node event listeners here

},

specPattern:

[ "cypress/e2e/3-integration/example/\*.js",

"cypress/e2e/1-getting-started/\*.js",

"cypress/e2e/2-advanced-examples/\*.js"

]

},

});

--

Bài 10

Build một file Test2.js để test

Chúng ta sử dụng trang này để test

<https://rahulshettyacademy.com/seleniumPractise/#/>

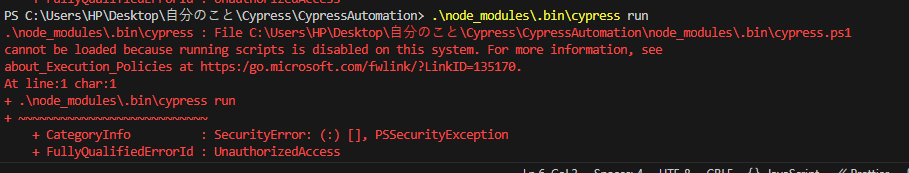
Muốn chạy file Cypress dùng lệnh npx cypress open đối với windows

--

Bài 11

Chú ý khi test web sẽ phải test trên đủ mọi thể loại browser v.v và nó cho phép test trên các thể loại v.v

Khi gặp lỗi bị như sau



Thì vào cmd sau đó gõ lệnh sau

Set-ExecutionPolicy RemoteSigned

Sau đó nhấn Y để đồng ý

--

Bài 12

Exploring the Cypress Project Framework Structure

Trong file cypress.config.js chúng ta sẽ thêm dòng như sau ở bên dưới dòng spec

screenshotsFolder: ‘cypress/screenshots’,

--

Section 4 Getting Started With Cypress Test Automation

Bài 13

Từ bài tiếp theo chúng ta sẽ dùng trang này để test

<https://rahulshettyacademy.com/seleniumPractise/#/>

Các trang ấy chính là trang này

https://rahulshettyacademy.com/seleniumPractise/#/. - (

--

Bài 14

Vào trang này <https://rahulshettyacademy.com/seleniumPractise/#/>

Ở windows bật phím Ctrl + Shift + I -> nó hơi khác hướng dẫn một chút

Cypress supports Css selector only

Ở phần này trang này chỉ dùng firefox với một extension là chropath plugin để nhìn được các thông tin

Bài 15

Khi gõ từ khóa ca vào ô tìm kiếm Flight Booking chúng ta thấy ra 4 kết quả

Thử code test ở file test như sau

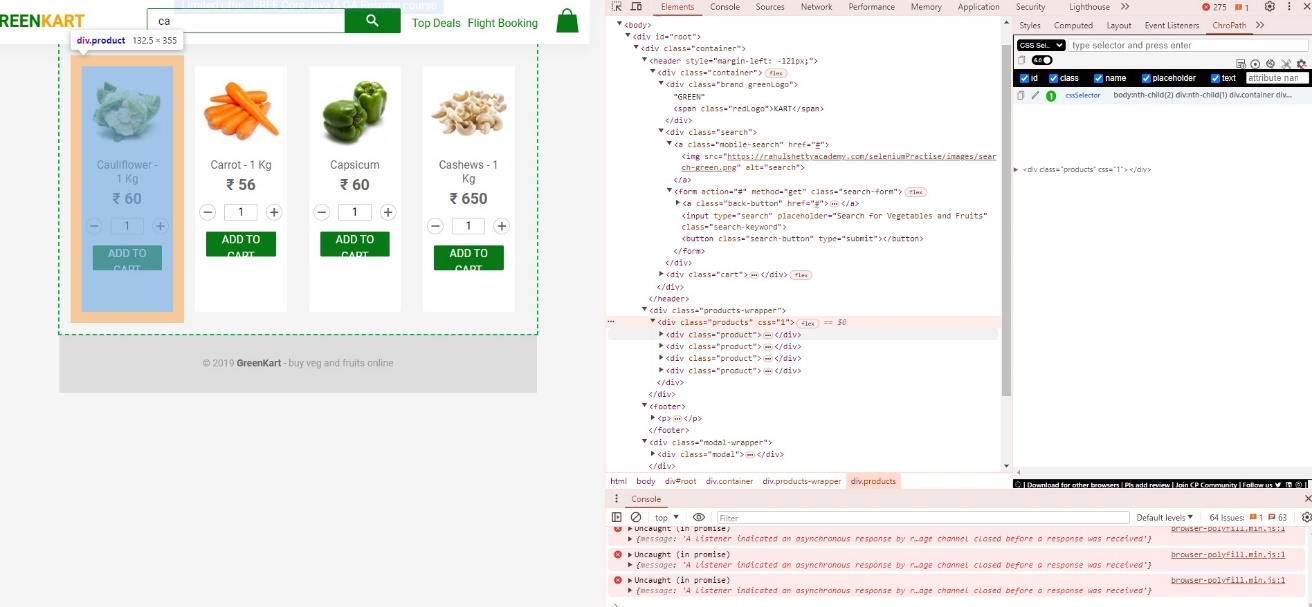
Chúng ta muốn test ô thấy nó viết là “Search for Vegetables and Fruit” -> tìm trong Elements thấy class của nó là search-keyword nên sẽ viết code để test như sau:

cy.get('.search-keyword').type('ca')

-> vì sao type lại là ca ? bởi vì khi gõ vào từ khóa tìm kiếm là ca cho nên cái type là ca , cái này có thể thay đổi được

Bài 16

Với từ khóa ca có thể cho thấy 4 kết quả



Như thấy ở trên chúng ta thấy có 4 product

Bây giờ viết test cho các product theo code như sau

Chúng ta cần phải tham khảo code về length theo link sau:

<https://docs.cypress.io/guides/references/assertions#Common-Assertions>

Ở trong đó có phần Length và phải chờ một time, do đó chúng ta code như sau

cy.visit("https://rahulshettyacademy.com/seleniumPractise/#/");

cy.get('.search-keyword').type('ca')

cy.wait(2000)

cy.get('.product').should('have.length',4)

Để chạy cypress ở windows chúng ta phải chạy như sau:

Mở terminal : nhấn CTRL + `

Bài 17

Vì sao có bug length of 4 but got 5

Tại vì nó tự động ra 4 kết quả nhưng có 5 kết quả ra như vậy

Vì có một cái bị ẩn cho nên mới như vậy

Phải sửa cái product thành product:visible nó mới hiện ra cái 5

--

Section 5

Bài 18

Trang web để test là trang này

<https://rahulshettyacademy.com/seleniumPractise/#/>

Bài 19

ở phần products-wrapper bên trong đó có products chứa 4 product

code của nó như sau

dùng cái DOM snapshot để tìm ra cái để có thể add vào code

-> cần một công cụ khác để thay thế cho công cụ DOM snapshot

-> get the DOM element contain the text ? How to get?

Bài 20

Có thể hỏi đáp tại các link sau

**FB QA Group** : <https://www.facebook.com/groups/RahulShettyAcademy/>  
  
**LinkedIn**: <https://www.linkedin.com/in/rahul-shetty-trainer/>

**Telegram Group**:  https://t.me/rahulshettyacademy

Bài 21

Trong Cypress có một lệnh tên là Each và lệnh đó dùng để làm gì?

cy.get('.products').find('.product').each($el, index, $list) => {

        const textVeg=$el.find('h4.product-name').text()

        if(textVeg.includes('Cashews')

        {

            cy.wrap($el).find('button').click()

        })

    }

should('have.length', 5)

    cy.get('.product:visible').should('have.length', 4)

    cy.get('.products').find('.product').should('have.length',4)

    cy.get(':nth-child(3) > .product-action > button').click()

    cy.get('.products').find('.product').eq(2).contains('ADD TO CART').click()

    cy.get('.products').find('.product').each($el, index, $list) => {

        const textVeg=$el.find('h4.product-name').text()

        if(textVeg.includes('Cashews')

        {

            cy$el.find('button').click()

        })

    }

Bài 22 Cypress Asynchronous nature and its promise handling

Bây giờ ở dưới lệnh sau

cy.get('.products').find('.product').eq(2).contains('ADD TO CART').click()

chúng ta thêm lệnh

cy.log(‘hello’)

như vậy lệnh này có ý nghĩa gì?

Thay lệnh này bằng lệnh

Console.log(‘sf’)

Nên nhớ rằng lệnh

It(‘ xxxx’, function() {

Cy.visit(‘/my/resource/path’) -> không xảy ra gì cả

Cy.get(‘awesome-selector’) -> không có gì

.click() -> không có gì

Cy.url()

.should(‘include’, ‘/my/resource/path#awesomeness’) -> chính cái lệnh này mới làm thay đổi cái gì đó

})

Bài 23 Understanding the difference between Jquery methods and cypress commands

Dùng một tiện ích khác tên là Cypress Smart để tìm các tên của slogan v.v

Nếu thêm một đoạn code bên dưới như sau

const logo=cy.get('.brand')

cy.log(logo.text())

thì sẽ bị lỗi nên phải sửa lại

Khi dùng tool CssViewer sẽ thấy code của bên GreenKart là .brand nên viết code của nó là như trên .brand

Bài 24

ở bên trên viết thêm một đoạn code

cy.get('.products').as('.productLocator')

Ở bên dưới sẽ sửa .productLocator trỏ vào @productLocator

-> productLocator là gì?

Tức là khi di chuột ra giữa trang sẽ thấy hàm tên là products, dòng ở trên đặt tên hàm products tên là productLocator

Dòng này

cy.get('@productLocator').find('.product').should('have.length',4)

có ý nghĩa là cái đó nó có 4 cái ô trong đó

Công cụ chúng ta sử dụng tên là CssViewer

Bài 26

Họ bỏ dòng này đi

Vì cái cart-icon có cái img cho nên chỉ viết là cy.get('.cart-icon > img').click()

Sau khi add một item nào đó thì có thể viết thêm một code như sau

cy.contains('PROCEED TO CHECKOUT').click()

bởi vì tên cái nút nó là PROCEED TO CHECKOUT nên viết như thế

dùng fn + f12 để tìm ra cái tên button đó

-> khi trỏ vào tên của button sẽ thấy nó hiện lên là place order nên viết code là place order

--

Section 6 Handling Web Controls UI using Cypress

Bài 28 -> tự động verify vs checkbox vs Cypress

<https://rahulshettyacademy.com/AutomationPractice/>

Câu hỏi được đặt ra là làm thế nào để tự động checkbox và tự động điền các từ khóa để kiểm thử

Đầu tiên bắt đầu vs checkbox

cy.get('#checkBoxOption1').check().should('be.checked').and('have.value','option1')

Ở ô Suggession Nhập India

Ở ô Dropdown Example nhập Option 1

Bài 29

Ở đoạn code bên dưới, giả sử chúng ta thấy có thể chọn select option 2 và nó phải có giá trị option 2 chúng ta sẽ viết đoạn code như sau:

cy.get('select').select('option2').should('have.value','option2')

chạy code thấy nó chạy được như vậy ok

Bài 30

Khi trỏ vào ô Type to Select Countries sẽ thấy nó có id là autocomplete, gõ ind xem trong Elements thấy có 3 cái ui-menu-item nên sẽ lấy cái đó để viết code auto test

Muốn làm đẹp code, nhấn Ctrl + Shift + P -> gõ code : Prettier: Format Document

Ở bên dưới có thể thêm dòng code này, nó có nghĩa là khi thêm autocomplete, nó sẽ cho vào ô tự động dòng India

cy.get("#autocomplete").should('have.value', 'India');

Bài 31

Tức là nếu thêm đoạn code này

cy.get('#hide-textbox').click()

cy.get('#displayed-text').should('not.be.visible')

cy.get('#show-textbox').click()

cy.get('#displayed-text').should('be.visible')

Thì nó click vào nút hide và show và nó sẽ ẩn và hiện cái textbox

--

Section 7

Bài 33

Khi click vào nút Alert nó sẽ hiện ra một pop up OK và bây giờ sẽ auto test nó như thế nào

Ở windows bật phím Ctrl + Shift + I -> nó hơi khác hướng dẫn một chút

Ở đây chúng ta sẽ test 2 thứ

Khi thấy nút Alert nó có id là alertbtn và nút Confirm có value=”Confirm”

Chúng ta sẽ viết code như sau

cy.get('#alertbtn').click()

cy.get('[value="Confirm"]').click()

có thể thử lại code như sau

cy.get('[value="Alert"]').click()

cy.get('#confirmbtn').click()

-> vẫn ok

Như vậy lý thuyết của mình là chuẩn

Vấn đề ở chỗ là nó hiện ra cái Alert và Confirm và ở trong cái ô test nó cũng sẽ hiện ra cái Alert và nội dung ở cái bên cạnh chính là nội dung của cái textbox đó

Về cypress events, nó có một cái được gọi là windows alert

Với đoạn code như sau, nó sẽ so sánh đoạn text mình input vào với đoạn message hiện ra để xem xem nó có giống nhau hay không

Nếu giống nhau, nó sẽ pass và hiện ra thông báo pass

describe("My Second Test Suite", function() {

it("My FirstTest case", function() {

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

cy.get('#alertbtn').click()

cy.get('[value="Confirm"]').click()

cy.on('window:alert', (str) => {

expect(str).to.equal('Hello , share this practice page and share your knowledge')

})

});

});

Như vậy, mình đã pass được thông báo

Với confirm mình cũng làm tương tự

Bài 34

Chúng ta sẽ sử dụng trang này để test qaclickacademy.com

Khi click vào home nó sẽ chuyển về trang qaclickacademy.com

Chú ý khi vào nút open tab chúng ta sẽ thấy code của nó như sau:

<a id="opentab" class="btn-style class1 class2" href="https://www.qaclickacademy.com" target="\_blank">Open Tab</a>

Như vậy lệnh này

cy.get("#opentab").invoke('removeAttr', 'target').click();

-> cái cy.get("#opentab") là vì nó có id của cái nút opentab là opentab

Cái invoke('removeAttr', 'target').click(); là vì cả cái đoạn code sau được dùng để mở code của cái nút ấy ra

Ở cái trang qaclickacademy.com chúng ta xem element của nó

Ở phần HOME COURSES BLOG ABOUT US v.v click vào cái ABOUT US

Sẽ thấy

ở đoạn sau sẽ thấy cần phải viết đoạn code này trước cái đã

cy.origin("https://www.qaclickacademy.com", () => {

cy.get("#navbarSupportedContent a[href\*='about']").click();

})

Tức là cho nó vào trang cần phải link tới rồi mới đến click vào nút cần click vào

Ở đoạn bên dưới có một cái dòng tên là Welcome to QAClick Academy

Click vào nó sẽ thấy nó có cái div class là section-title mt-50 cho nên viết code như sau:

cy.get(".mt-50 h2").should('contain', 'QAClick Academy')

--

Bài 36

Cái đoạn code để check của nó là đoạn code này

it('My FirstTest case', function() {

cy.visit("https://rahulshettyacademy.com/AutomationPractice/")

cy.get('tr td:nth-child(2)').each(($e1, index, $list) => {

const text = $e1.text()

if (text.includes("Python")) {

cy.get("tr td:nth-child(2)").eq(index).next().then(function(price) {

const priceText = price.text()

expect(priceText).to.equal('25')

})

}

})

})

Vấn đề là làm thế nào để ra được cái đoạn td:nth-child(2)

-> sử dụng tool selectorsHub để tra ra code đó -> trong phần jQuery (tuy nhiên phải sửa lại một chút)

Bài 38

Bây giờ chúng ta quan tâm đến nút Mouse Hover

Nếu viết code như thế này sẽ bị lỗi

cy.get('#mousehover').invoke('show')

cy.contains('Top').click()

cy.url().should('include', 'top')

Vì sao

Vì nó còn một cái con ở bên dưới nó là mouse-hover-content

Do đó cần phải sửa lại code

Bây giờ lại sửa lại code nữa

Nếu bỏ cái dòng này

cy.get('div.mouse-hover-content').invoke('show')

và

bỏ force true ở phần contains đi tức là click(force true)

thì nó sẽ bị lỗi

--

Section 8 Understand How to Automate Frames & Child Windows & Calendars In Cypress

Bài 39

Bây giờ chúng ta cần học cách handling frames với cypress tức là xử lý các khung với cypress

ở trang này

<https://rahulshettyacademy.com/AutomationPractice/>

ở bên dưới có iFrame

chúng ta gõ lệnh npm install -D cypress-iframe để install frame cho cypress

ở bên trên chúng ta thêm đoạn code sau để thêm iframe vào cypress

import 'cypress-iframe'