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/#/>

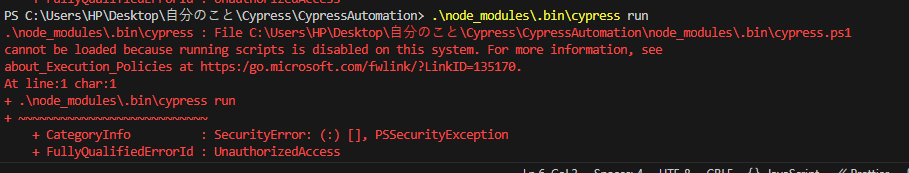
Muốn chạy file Cypress dùng lệnh npx cypress open đối với windows

--

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

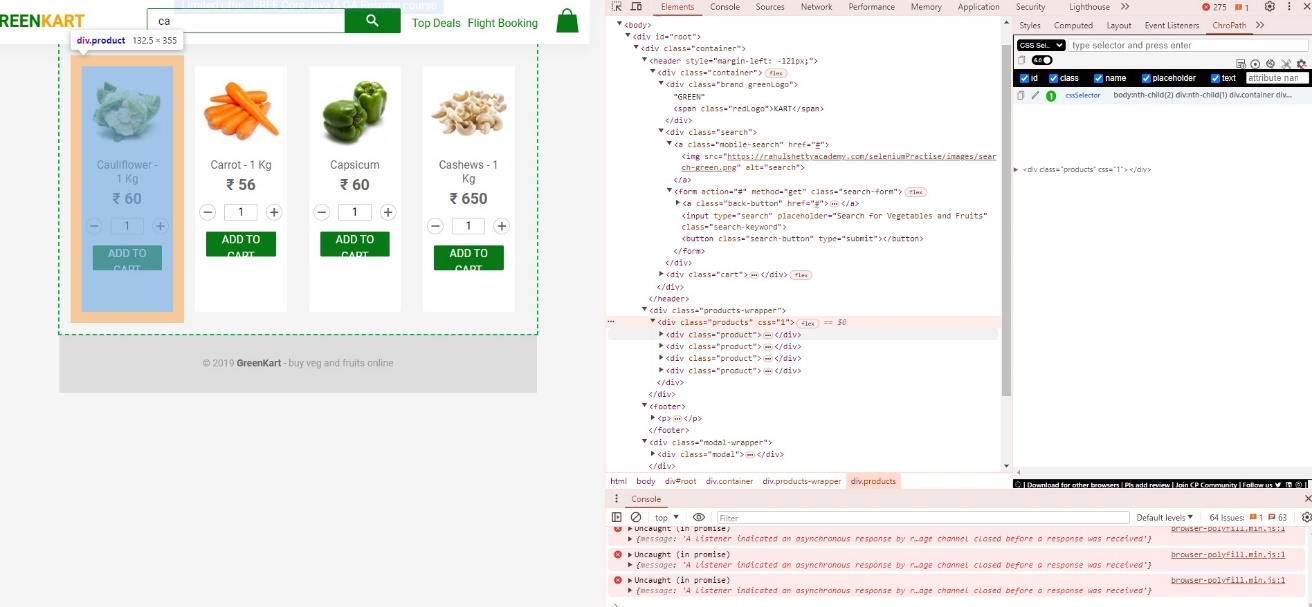
Chúng ta muốn test ô thấy nó viết là “Search for Vegetables and Fruit” -> tìm trong Elements thấy class của nó là search-keyword nên sẽ viết code để test như sau:

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()

        })

    }