In previous call I got to know about the Cypress tool. So curious about it I tried some scripting and tried to execute some observations I had for Hardware support page using Cypress.

Test case 01 - Check suggestions returned for search term from ajax call.

//Need to check on some javascript exceptions thrown while loading pages

Cypress.on('uncaught:exception', (err, runnable) => {

    // returning false here prevents Cypress from

    // failing the test

    return false

})

describe ('Hardware support page test suite',function()

{

  it('Test case 01 - Check suggestions returned for search term from ajax call',function()

  {

    const searchTerm='camera'

    cy.log('Test case 01')

    cy.visit("https://mathworks.com")

    cy.contains('a','United States').click()

    //cy.contains('a','India').click()  //India   //cy.contains('a','Denmark').click() //Denmark    //cy.contains('a','France').click() //France

    //cy.get('a').contains('India').click() // India

    //generic country selection as displayed by browser- cy.get('#recommended\_domain\_button').click()

    cy.get('.navbar-header > .navbar-collapse > .topNavigation > #topnav > .topnav\_support > a').click()

    cy.get('#fatfooter\_products > .list-unstyled > :nth-child(4) > a').click()

    //check suggestions response json

    cy.intercept('\*\*&term='+searchTerm).as('suggestionsCheck')

    cy.get('#search').type(searchTerm)

    cy.wait('@suggestionsCheck')

    cy.get('@suggestionsCheck').then( outputRespose =>{

      console.log(outputRespose)

      expect(outputRespose.response.statusCode).to.equal(200)

      expect(outputRespose.response.body).to.include(searchTerm)

      }

    )

  }

  )

}

)

Test case 02 - Check search results response body contains search term.

//Need to check on some javascript exceptions thrown while loading pages

Cypress.on('uncaught:exception', (err, runnable) => {

    // returning false here prevents Cypress from

    // failing the test

    return false

})

describe ('Hardware support page test suite',function()

{

  it('Test case 02 - Check search results response body contains search term',function()

  {

    const searchTerm='camera'

    const searchCount=52

    cy.log('Test case 02 - Check search results response body contains search term')

    cy.visit("https://mathworks.com")

    cy.contains('a','United States').click()

    //cy.contains('a','India').click()  //India   //cy.contains('a','Denmark').click() //Denmark    //cy.contains('a','France').click() //France

    //cy.get('a').contains('India').click() // India

    //generic country selection as displayed by browser- cy.get('#recommended\_domain\_button').click()

    cy.get('.navbar-header > .navbar-collapse > .topNavigation > #topnav > .topnav\_support > a').click()

    cy.get('#fatfooter\_products > .list-unstyled > :nth-child(4) > a').click()

    //check search results response body contains search term

    cy.intercept('GET','\*\*&q='+searchTerm).as('searchResultsCheck')

    cy.get('#search').type(searchTerm)

    cy.get('#searchbutton\_customized').click()

    cy.wait('@searchResultsCheck')

    cy.get('@searchResultsCheck').then( outputSearchResults =>{

      console.log(outputSearchResults)

      console.log(outputSearchResults.request.url)

      expect(outputSearchResults.response.statusCode).to.equal(200)

      expect(outputSearchResults.response.body.response.numFound).to.equal(searchCount)

      expect(outputSearchResults.response.body.response.docs[0].body\_en).to.include(searchTerm)

      expect(outputSearchResults.response.body.response.docs[0].body\_en).to.include('Suraj') //check to see if it fails

      //Need to check how to do same test for search containing multiple word

      })

  })

})

Test case 03 - check vendor filter

//Need to check on some javascript exceptions thrown while loading pages

Cypress.on('uncaught:exception', (err, runnable) => {

    // returning false here prevents Cypress from failing the test

    return false

})

describe ('Hardware support page test suite',function()

{

  it('Test case 03 - check vendor filter',function()

  {

    const searchTerm='camera'

    const searchvendorCount=1

    cy.log('Test case 03 - check vendor filter')

    //cy.visit("https://in.mathworks.com/hardware-support.html?q=camera&page=1")

    cy.visit("https://mathworks.com")

    cy.contains('a','United States').click()

    cy.get('.navbar-header > .navbar-collapse > .topNavigation > #topnav > .topnav\_support > a').click()

    cy.get('#fatfooter\_products > .list-unstyled > :nth-child(4) > a').click()

    cy.get('#search').type(searchTerm)

    cy.get('#searchbutton\_customized').click()

    const vendorCSSselector ='#pagefacets > .myfacets > [data-facet-id="hardware-support-vendor:"] > .search\_refine\_scroll > .list-unstyled > :nth-child(2) > .facet-link'

    let vendorID = cy.get(vendorCSSselector).invoke('attr', ' data-id').then(function(fieldValue){

      //const vendorName=fieldValue

      /\*Able to fetch the the value data-id="hardware-support-vendor:adlink" and store in vendorName. But need to check how to parse and remove some values to get only 'adlink'

      so code will be generic and can be used for any vendor selected to intercept the url as per it and check it's value in response\*/

      const vendorName='adlink'

      cy.log(vendorName)

      cy.intercept('GET','\*'+vendorName+'\*').as('vendorFiltered')

      cy.get(vendorCSSselector).click()

      cy.wait('@vendorFiltered')

      cy.get('@vendorFiltered').then( vendorFilteredResults =>{

        console.log(vendorFilteredResults)

        expect(vendorFilteredResults.response.statusCode).to.equal(200)

        expect(vendorFilteredResults.response.body.response.docs[0]['hardware-support-vendor']).to.include('adlink')

        expect(vendorFilteredResults.response.body.response.numFound).to.equal(searchvendorCount)

         }) })

  }) })

Test case 04 - Check search results response correctly mapped in multiple UI elements.

//Need to check on some javascript exceptions thrown while loading pages

Cypress.on('uncaught:exception', (err, runnable) => {

    // returning false here prevents Cypress from

    // failing the test

    return false

})

describe ('Hardware support page test suite',function()

{

  it('Test case 04 - Check search results response correctly mapped in multiple UI elements',function()

  {

    const searchTerm='camera'

    cy.log('Test case 04 - Check search results response coorectly mapped in UI elements')

    cy.visit("https://mathworks.com")

    cy.contains('a','United States').click()

    //cy.contains('a','India').click()  //India   //cy.contains('a','Denmark').click() //Denmark    //cy.contains('a','France').click() //France

    //cy.get('a').contains('India').click() // India

    //generic country selection as displayed by browser- cy.get('#recommended\_domain\_button').click()

    cy.get('.navbar-header > .navbar-collapse > .topNavigation > #topnav > .topnav\_support > a').click()

    cy.get('#fatfooter\_products > .list-unstyled > :nth-child(4) > a').click()

    //check search results response body contains search term

    cy.intercept('GET','\*\*&q='+searchTerm).as('searchResultsCheck')

    cy.get('#search').type(searchTerm)

    cy.get('#searchbutton\_customized').click()

    cy.wait('@searchResultsCheck')

    cy.get('@searchResultsCheck').then( outputSearchResults =>{

      console.log(outputSearchResults)

      console.log(outputSearchResults.request.url)

      //Title validation

      const firsttitleSelector =':nth-child(1) > .search\_result\_content > .search\_result\_title > a'

      const firsttitleinResponse=outputSearchResults.response.body.response.docs[0].title\_en;

      cy.get(firsttitleSelector).then(function(fieldValue){

        const firsttitleDisplayed=fieldValue.text()

        cy.log(firsttitleDisplayed)

        expect(firsttitleDisplayed).to.include(firsttitleinResponse)

      })

      //Total no of results displayed validation [numcount]

      const numcountSelector='#result\_summary\_top > .add\_font\_color\_mediumgray > .add\_margin\_0'

      const numcountinResponse=outputSearchResults.response.body.response.numFound

      cy.get(numcountSelector).then(function(numcountObject){

        const numcountDisplayed=numcountObject.text()

        cy.log(numcountDisplayed)

        expect(numcountDisplayed).to.include(numcountinResponse)

      })

       //Metadescription validation

      const metadescriptionSelector=':nth-child(1) > .search\_result\_content > .search\_result\_desc'

      const metadescinResponse=outputSearchResults.response.body.response.docs[0].meta\_description\_en

      cy.get(metadescriptionSelector).then(function(metadescObject){

        const metadescDisplayed=metadescObject.text()

        cy.log(metadescDisplayed)

        expect(metadescDisplayed).to.include(metadescinResponse)

        //expect(metadescDisplayed).to.include('Suraj')  //test to see if fails for invalid content

      })

      })

  })

})