1. CheckBox-

- .check()

For assertion

- .check().should(‘be.checked’)

.uncheck().should('not.be.checked')

1. Static Dropdown

Cy.get(‘select’).select('option1').should('have.value','option1')

1. Dynamic Dropdown

cy.get('#autocomplete').type('Ind')

     cy.get('.ui-menu-item div').each(($el, index, $list) =>

     {

        if($el.text()=='India')

        {

           cy.wrap($el).click()

        }

     })

1. Handeling visibe and invisible element

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

     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')

1. Radio button

.check()

 cy.get('input[value="radio2"]').check().should('be.checked')

1. Alert/ conform

- Cypress automatically handle **Alert** and **Popups. window:alert** event get fired when alert open.so we have to fire event through cypress to handle text on alert

#Alert

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

    })

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

    {

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

#conform (also string compare code)

//Conform

    cy.on('window:confirm', (str)=>

    {

      expect(str).to.equal('Hello , Are you sure you want to confirm?')

    })

1. Child Browser

To handle child browser using cypress need to remove **Target** attribute from html.

This target attribute we can remove by using method **.invoke(‘removeAttr’,’target)**

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

1. Browser Navigation

- backword:- Cy.go(‘back’)

- forword:- cy.go(‘forword’)

1. Get url

- cy.url()

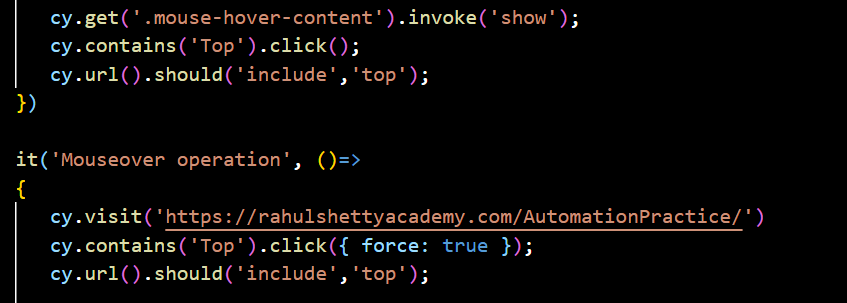
Assertion also included in below code

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

1. Mouse over handeling

- .invoke(‘show’)

-.click({force : true})



1. Iframe handelling

- install iframe

- cy.iframeloaded(‘#iframe’)

-cy.iframe.find(‘…..’)

/// <reference types= "Cypress" />

/// <reference types="cypress-iframe" />

import 'cypress-iframe'

describe('handeling of child browser', function(){

    it('verify the element of chuld browser',()=>{

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

        cy.frameLoaded('#courses-iframe');

        cy.iframe().find("a[href\*='mentorship']").eq(0).click();

        cy.wait(2000)

        cy.iframe().find("div[class\*='pricing-header']").should('have.length',2)

    })

})

Create short commond in package.json for CI - CD intigrestion in Jenkin

  "scripts": {

    "test": "node\_modules\\.bin\\cypress run",

    "headedTest": "npm run test -- --headed",

    "ChromeTest": "npm run headedTest -- --browser chrome",

    "RecordTest": "npm run ChromeTest -- --record --key c26ec069-7899-4aa8-b266-a02a8d1d42a1",

    "ReportTest": "npm run ChromeTest -- --record --key c26ec069-7899-4aa8-b266-a02a8d1d42a1 --reporter mochawesome"

For jenkin

Download war file from jenkin official website

**# for cucumber**

-Search for badeball cypress cucumber preprocesser github file

“npm install @badeball/cypress-cucumber-preprocessor”

- need to register cucumber plugin to “steupnodevent” in config.js file

async function setupNodeEvents(on, config) {

  // This is required for the preprocessor to be able to generate JSON reports after each run, and more,

  await preprocessor.addCucumberPreprocessorPlugin(on, config);

  on("file:preprocessor", browserify.default(config));

  // Make sure to return the config object as it might have been modified by the plugin.

  return config;

}

1. Import depedancy

const preprocessor = require("@badeball/cypress-cucumber-preprocessor");

const browserify = require("@badeball/cypress-cucumber-preprocessor/browserify");

Ii. In packge.json add below dependancy

"@cypress/browserify-preprocessor": "latest"

- Install “ VS code Cucumber BDD extention”

- create **feature file** and **steps folder name** same

- In Steps.js file import below for suggestion

const { Given, When, Then,  } = require("@badeball/cypress-cucumber-preprocessor");

**#Reports**

1. add this to package.json

"cypress-cucumber-preprocessor":

  {

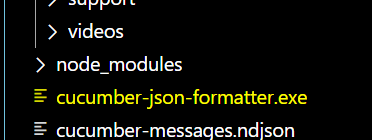
    "json": {

      "enabled": true,

      "output": "cypress/Reports/JsonResult.json"

    },

Add jason formatter in project path



Download above from : [GitHub - cucumber/json-formatter: Provides a language-agnostic command-line tool to convert cucumber messages into a JSON document.](https://github.com/cucumber/json-formatter)

\*Install the html reporter using - (search with multipal cucumber html reporter)

npm install multiple-cucumber-html-reporter

\* cretae **“cucumber-html-report.js”** file in projct level and pest below data

const report = require("multiple-cucumber-html-reporter");

report.generate({

jsonDir: "./path-to-your-json-output/",

reportPath: "./path-where-the-report-needs-to-be/",

metadata: {

browser: {

name: "chrome",

version: "60",

},

device: "Local test machine",

platform: {

name: "ubuntu",

version: "16.04",

},

},

customData: {

title: "Run info",

data: [

{ label: "Project", value: "Custom project" },

{ label: "Release", value: "1.2.3" },

{ label: "Cycle", value: "B11221.34321" },

{ label: "Execution Start Time", value: "Nov 19th 2017, 02:31 PM EST" },

{ label: "Execution End Time", value: "Nov 19th 2017, 02:56 PM EST" },

],

},});

Run the above in VS code using - “node **cucumber-html-report.js”**