Unzip the folder and open the folder in the vs code terminal execute the following command

1. Create fresh package.json by using => Npm -i init keep pressing enter till you get outside.
2. Install cypress freshly => npm install [cypress@13.6.6](mailto:cypress@13.6.6)
3. Open cypress => npx cypress open

You are all set to run test case

To record video adjust setting in cypress.config file

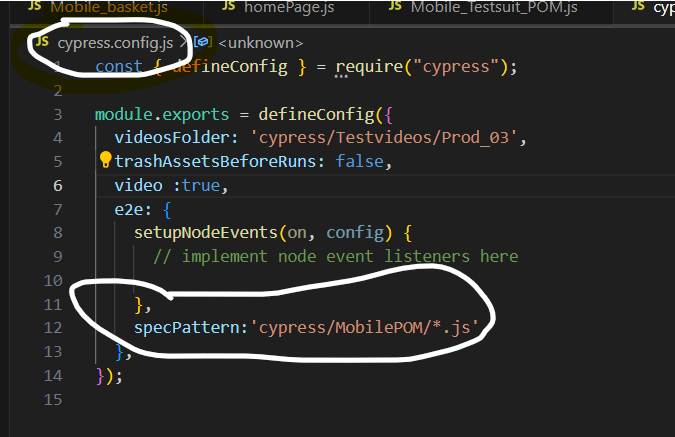
  videosFolder: 'cypress/Testvideos/Prod\_03', (setting the path of recorded video)

  trashAssetsBeforeRuns: false, (this will not delete older recorded video)

  video :true, (to record video)

Set path for specs Test case

specPattern : "cypress/Mysuit"



Hovering using real event

1. install cypress-real-events -> npm i cypress-real-events or npm i cypress-real-events --f.
2. import this statement -> import "cypress-real-events"; into your commands.js or index.js whichever u have in under your support folder.
3. Now use it like this -> cy.get(".cssmenu > :nth-child(1)").realHover();

Disabling logging XHR/ Logs

<https://docs.cypress.io/guides/cloud/account-management/data-storage-and-masking#:~:text=Disable%20logging%20of%20XHR%2FFetch,false%20%7D%20in%20the%20second%20parameter>.

What works now - in cypress/support/e2e.js add this code:

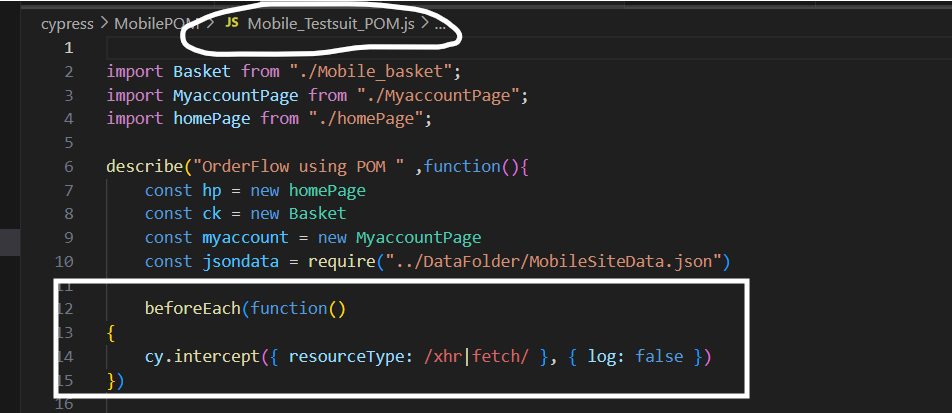
before(() => {

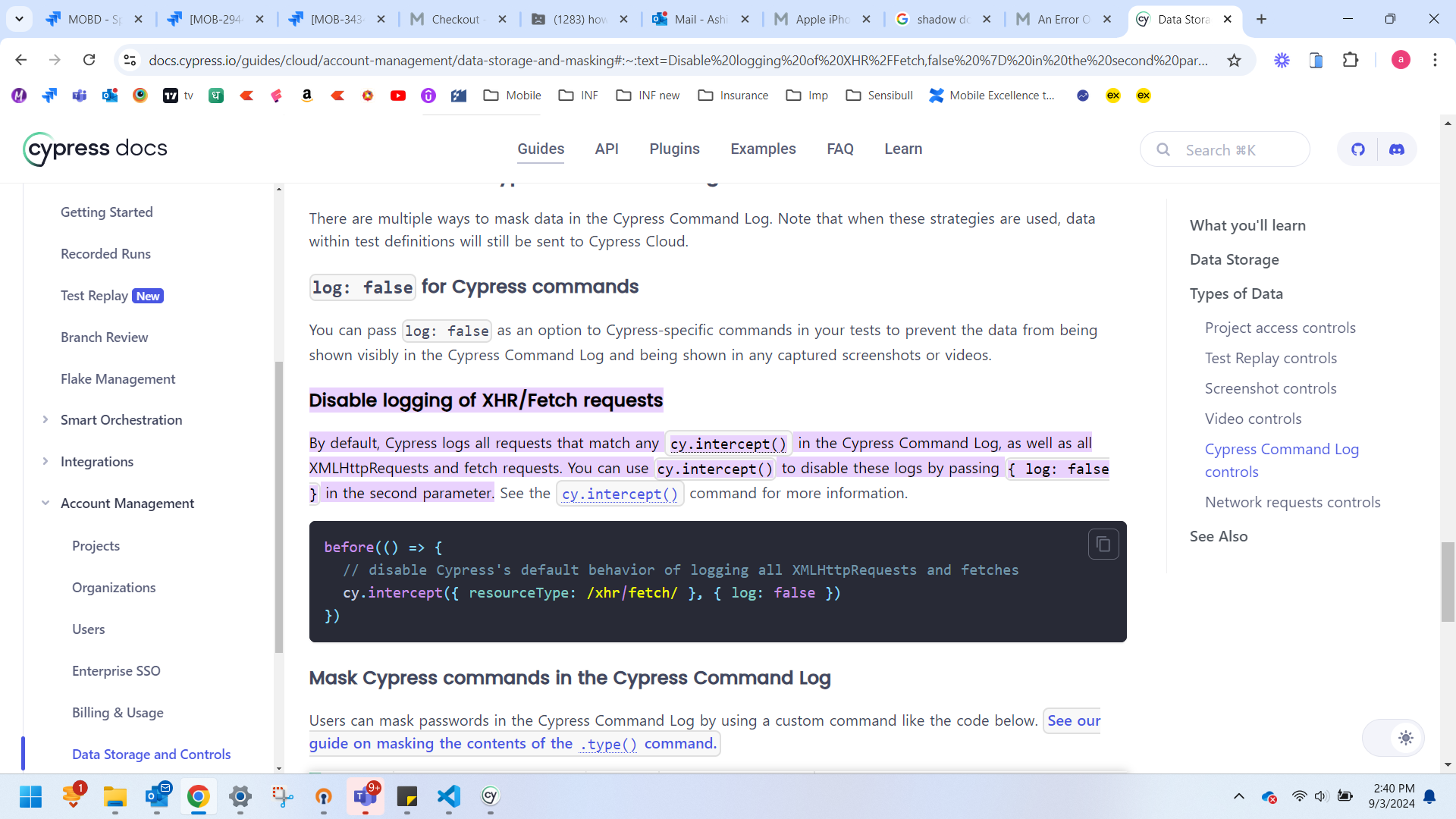
// disable Cypress's default behavior of logging all XMLHttpRequests and fetches

cy.intercept({ resourceType: /xhr|fetch/ }, { log: false })

})

Best practice is to add this in beforeeach block at the beginning of the describe block so it will clear all logs at before running test.

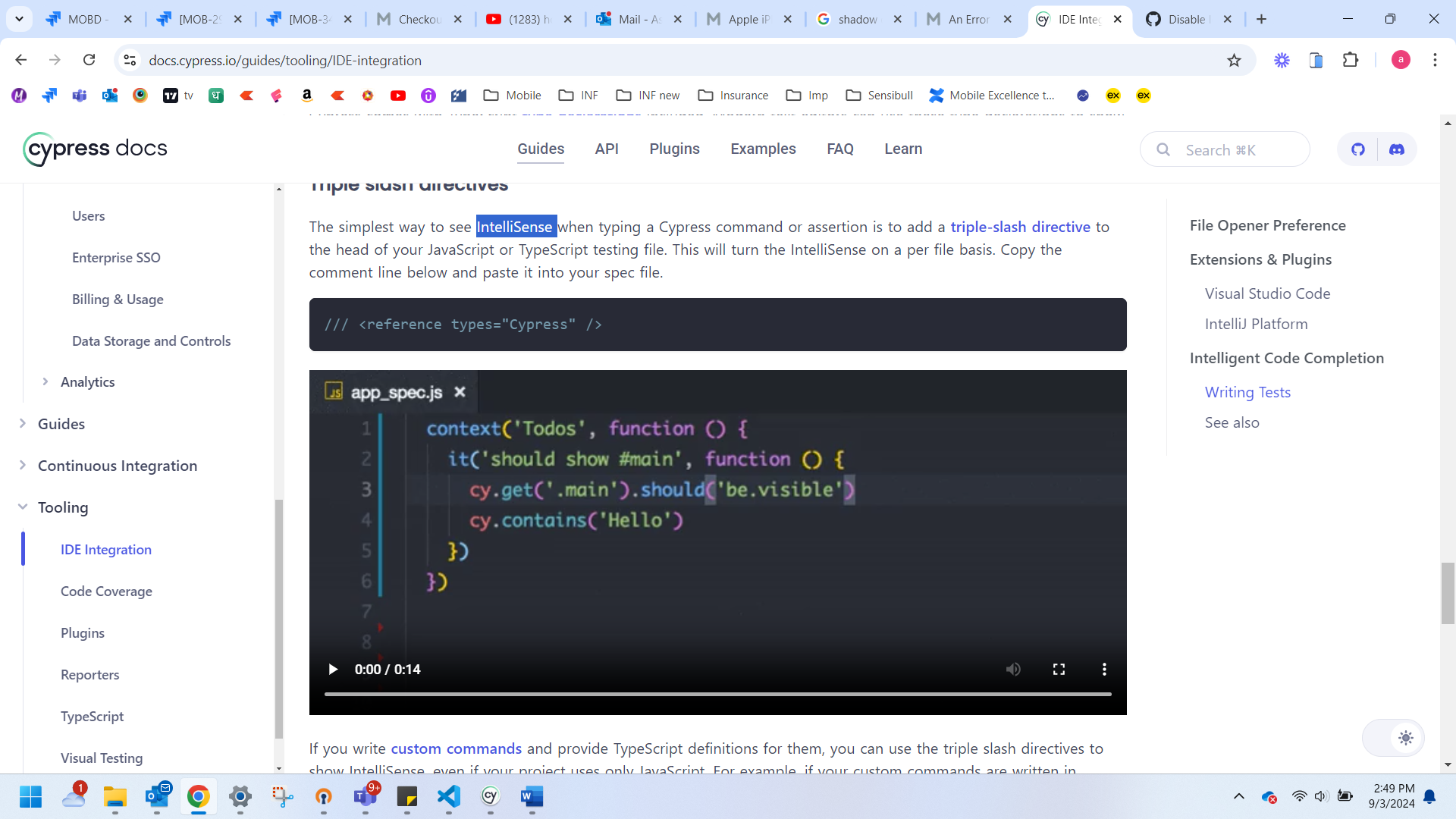




For IntelliSense

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

The simplest way to see IntelliSense when typing a Cypress command or assertion is to add a [triple-slash directive](http://www.typescriptlang.org/docs/handbook/triple-slash-directives.html) to the head of your JavaScript or TypeScript testing file.



For IntelliSense  Global level define

Instead of adding triple slash directives to each JavaScript spec file, some IDEs (like VS Code) understand a common jsconfig.json file in the root of the project. In that file, you can include the Cypress module and your test folders.

What works now - in ROOT level folder/jsconfig.json add this code:

{

"include": ["./node\_modules/cypress", "cypress/\*\*/\*.js"]

}

Uncaught exception

What works now - in cypress/support/e2e.js add this code:

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

    // returning false here prevents Cypress from

    // failing the test

    return false

})

To hover on element

npm install cypress-real-events

Register new commands by adding this to your cypress/support/e2s.js

import "cypress-real-events";

Video recording in Cypress

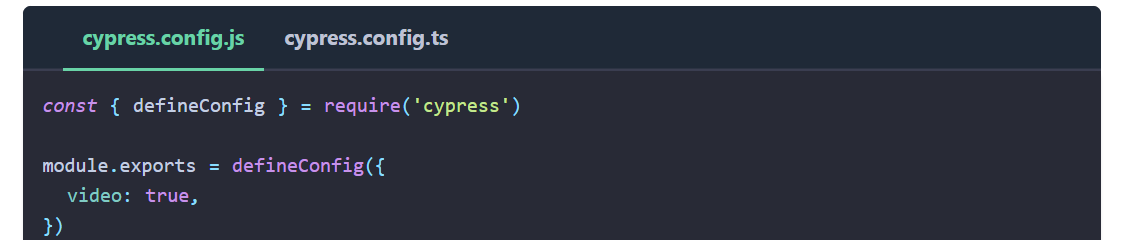
Video recording is disabled by default, but can be turned on by setting [video](https://docs.cypress.io/guides/references/configuration#Videos) to true from within your configuration.

If enabled, Cypress records a video for each spec file when running tests during cypress run. Videos are *not* recorded during cypress open.

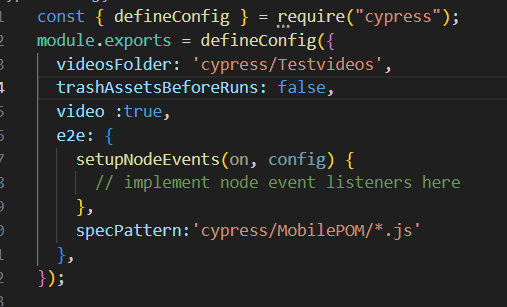
To run cypress from terminal

npx cypress run --browser electron --headed --spec "cypress\MobilePOM\e2save\_Testsuit\_POM.js"

video: true,

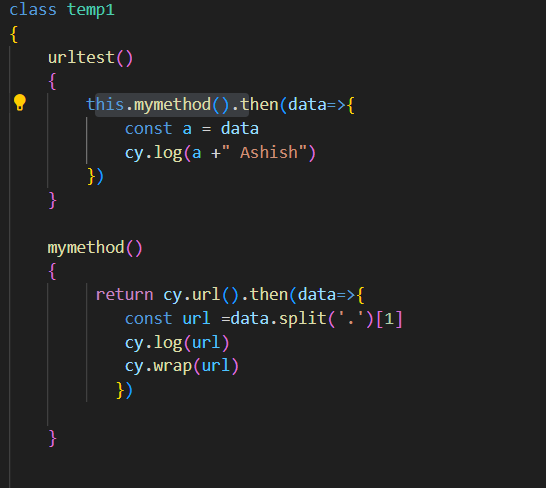


Cypress clears any existing videos before a cypress run. If you do not want to clear your videos folder before a run, you can set [trashAssetsBeforeRuns](https://docs.cypress.io/guides/references/configuration" \l "Videos) to false.



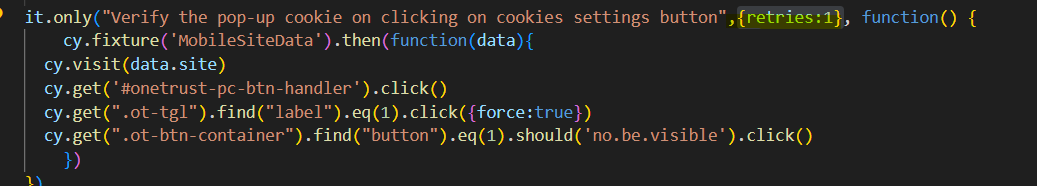
Returning value from methods

As cypress is asynchronous we can not directly return the value ve have to use wrap and alis to return the value



Rerun Failed test

For that we have to add {retries:1} in test block it after test name ,



To ignore case sensitivity

Cypress contains methods already have such feature. You can just pass the matchCase as options argument and cypress will handle case sensistive/insensitive conditions



To run test in incognito mode

Add code in cypress.config file <https://stackoverflow.com/questions/68896471/how-to-run-cypress-headed-tests-using-chrome-incognito>

 on('before:browser:launch',(browser ={}, launchoptions )  =>{

        if (browser.family ==='chromium' && browser.name !=='electron') {

          launchoptions.args.push("--incognito");

        }

        if(browser.name ==='electron')

        {

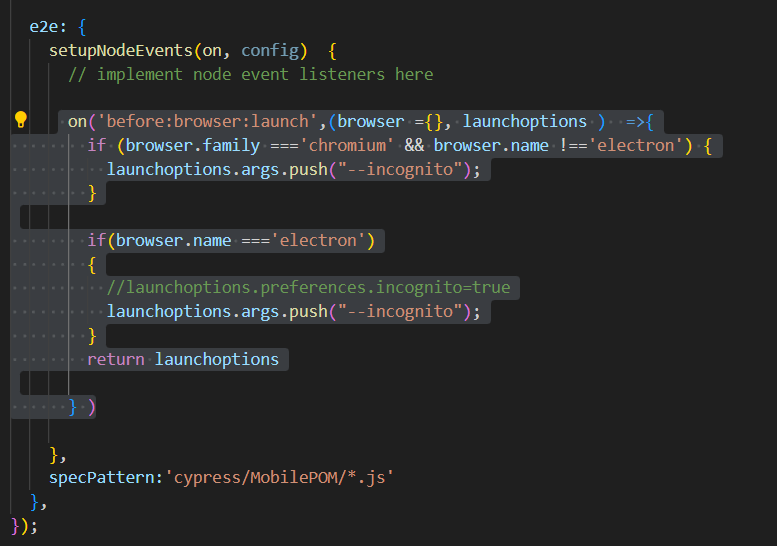
          //launchoptions.preferences.incognito=true

          launchoptions.args.push("--incognito");

        }

        return launchoptions

      } )



Core concepts :-

Alert

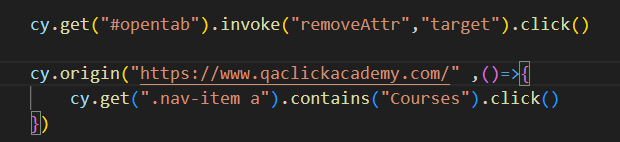
1. Cypress auto accepts alerts
2. Cypress have capability of browser events “window.alert” this Is event which is get fired from alert event. So we are fireng event by cypress and capturing alert text.

Child window

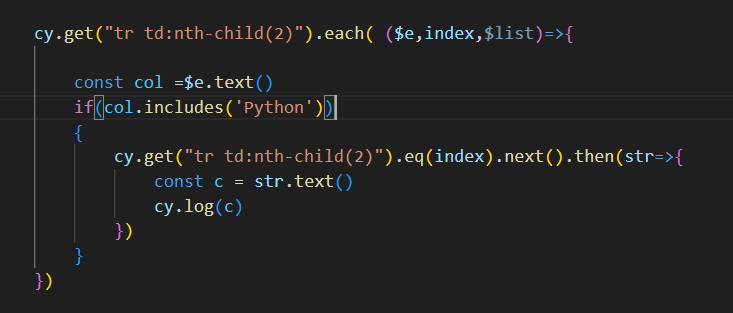
1. Cypress cannot handle multiple window multiple tab , so approach is , remove target attribute from element and click on element.

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

when you moved to new domain cypress cannot handle it you have to tell cypress by calling origin and inside that you have to give operation you want to perform on that domain



To iterate through table we have to use Each

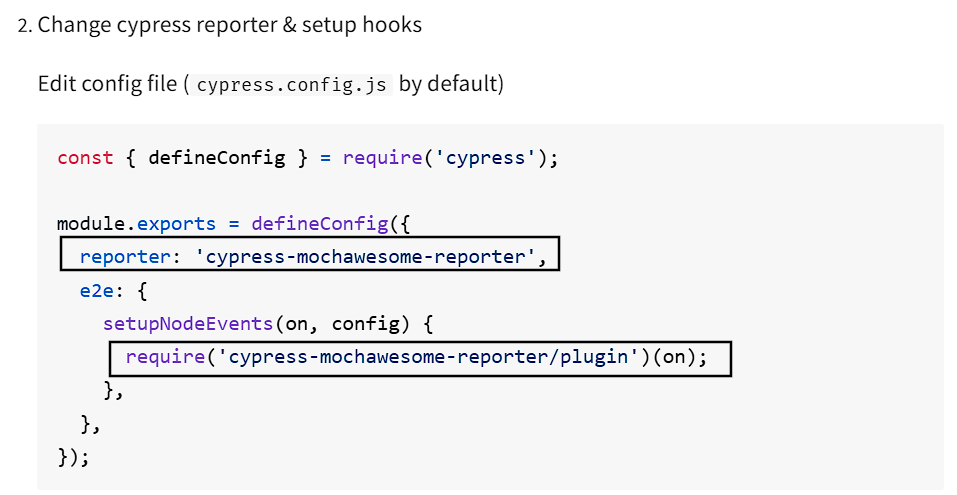


Cypress -Mochawesome-reporter

1. Install mochawesome- reporter

npm install cypress-mochawesome-reporter

1. reporter: 'cypress-mochawesome-reporter',





1. run cypress