**Method 1:**

const{Builder, By, Key}=require("selenium-webdriver");

const assert=require("assert");

var should=require("chai").should();

var expect=require("chai").expect();

let driver;

async function example(){

driver=await new Builder().forBrowser('chrome').build();

await driver.get("https://lambdatest.github.io/sample-todo-app/");

await driver.findElement(By.id("sampletodotext")).sendKeys("Hello World",Key.RETURN);

//assert

let actualText=await driver.findElement(By.xpath("//li[last()]")).getText().then(function(value){

return value;

}); // then function will return the value from getText() and store it in actualText variable

//assert using node assertion

assert.strictEqual(actualText,"Hello World");

//assert using chai should

actualText.should.equal("Hello World");

await driver.quit();

}

example() // it will call the function while running the test

**Method 2:**

const{Builder,By,Key}=require('selenium-webdriver');

let driver;

(async function functionName(){

driver=await new Builder().forBrowser('chrome').build();

await driver.get('url');

await driver.sleep(2000);

await driver.quit();

})();

**Method 3: Mocha Test Framework**

const{Bulder, By, until, Key}=require('selenium-webdriver');

const assert = require('assert');

describe('Test feature',function()

{

let driver;

beforeEach(async function(){

driver=await new Builder().forBrowser('chrome').build();

await driver.get('web url');

await driver.manage().setTimeout({implicit:3000});

});

afterEach(async function(){

await driver.quit();

});

it('test login',async function(){

await driver.findElement(By.name("")).sendKeys('');

await driver.findElement(By.xpath("//input[@type='password']")).sendKeys('');

let clickSubmit= await driver.findElement(By.Id(''));

clickSubmit.click();

});

it('emailTyping', async function() {

//await driver.get('https://library-app.firebaseapp.com');

await driver.findElement(By.xpath("//input[@id='ember26']")).sendKeys('abc@gmail.com');

await driver.sleep(2000);

await driver.findElement(By.xpath("//button[contains(text(),'Request invitation')]")).click();

var enableElement=await driver.findElement(By.xpath("//div[contains(text(),'Thank you! We saved a fake email address with the ')]"));

await driver.wait(until.elementIsEnabled(enableElement), 4000);

}).timeout(5000);

// once u specify timeout(), then u can use driver.sleep() . Since, mocha has default timeout of 2 seconds.

// or you can comment .timeout(); , when you specify "test": "mocha --no-timeouts" in package.json then you don’t need to write .timeout();

it('test home',async function(){

await driver.findElement(By.name("")).sendKeys('');

await driver.findElement(By.xpath("//input[@type='password']")).sendKeys('');

let clickSubmit= await driver.findElement(By.Id(''));

clickSubmit.click();

//explicit wait

await driver.wait(until.elementLocated(ele),5000);

});

});

==================================================================================

**//Some sample commands for WebDriver JS:-**

**//Assert**

assert.StrictEqual(await foo.getText(),'asdsd csc');

**//fluent wait**

await driver.wait(until.elementLocated(ele),30000,'After every 30 sec',5000);

**//finding all elementLocated with tag p**

let elements= await driver.findElements(By.tagName('p'));

for(let element of elements)

{

console.log(await element.getText());

}

**// Enter text "webdriver" and perform keyboard action "Enter"**

await driver.findElement(By.css('')).sendKeys('dsdfsf',Key.ENTER);

**//Clear a text field**

await driver.findElement(By.name('email')).clear();

**//Select functionality**

var selElement= await driver.findElement(By.xpath("//input[@name='csc']"));

var select=await new Select(selElement);

await select.SelectByValue('sdsds');

await select.SelectByVisibleText('cdcd');

await select.SelectByIndex(2);

var allOptions=await select.getOptions();

var allSelectedOption=await select.getSelectedOptions();

await select.deselectByValue('efsd');

**//Actions API**

**Pause:**

Pointer movements and Wheel scrolling allow the user to set a duration for the action, but sometimes you just need to wait a beat between actions for things to work correctly.

var clickableAct=await driver.findElement(By.id('clickable'));

await driver.actions().move({origin:clickable}).pause(1000).press().pause(1000).sendKeys('cdss').perform();

**//Release all actions**

await driver.actions().clear();

**//Key down**

await driver.actions().keyDown(Key.SHIFT).sendKeys('a').perform();

**//Key up**

await driver.action.keyDown(Key.SHIFT).sendKeys('a').keyUp(Key.SHIFT).sendKeys('b').perform();

**//Designated Element**

var element= await driver.findElement(By.id('cdcs'));

await driver.actions().sendKeys(element,'hello world').perform();

**//Click and hold**

var element= await driver.findElement(By.id('click'));

await driver.actions({async:true}).move({origin:element}).press().perform();

**//Click and release**

var element= await driver.findElement(By.id('click'));

await driver.actions({async:true}).move({origin:element}).click().perform();

**//Context click**

var element= await driver.findElement(By.id('click'));

await driver.actions(async:true).contextClick(element).perform();

**// Double click**

const clickable = driver.findElement(By.id("clickable"));

const actions = driver.actions({async: true});

await actions.doubleClick(clickable).perform();

**// Move to element**

var element= await driver.findElement(By.id('click'));

await driver.actions({asyc:true}).move({origin:element}) .perform();

**// Drag and Drop on Element**

const draggable = driver.findElement(By.id("draggable"));

const droppable = await driver.findElement(By.id("droppable"));

await driver.actions({async:turue}).dragAndDrop(draggable,droppable).perform();

**// Scroll to element**

const iframe = await driver.findElement(By.css("iframe"));

await driver.actions().scroll(10,10,0,200,iframe).perform();

**//Navigate**

await driver.navigate().back();

await driver.navigate().forward();

await driver.navigate().refresh();

**//Alert**

await driver.wait(until.alertIsPresent());

var al=await driver.switchTo().alert();

var txt=al.getText();

await al.accept();

await al.sendKeys('dssd')

await al.dismiss();

**//Cookies**

// set a cookie on the current domain

await driver.manage().addCookie({ name: 'key', value: 'value' });

**//frames**

// Store the web element

const iframe = driver.findElement(By.css('#modal > iframe'));

**// Switch to the frame**

await driver.switchTo().frame(iframe);

**// Now we can click the button**

await driver.findElement(By.css('button')).click();

**// Using the ID**

await driver.switchTo().frame('buttonframe');

**// Or using the name instead**

await driver.switchTo().frame('myframe');

**// Now we can click the button**

await driver.findElement(By.css('button')).click();

**// Switches to the second frame**

await driver.switchTo().frame(1);

**// Return to the top level**

await driver.switchTo().defaultContent();