**WebDriver I/O v.5 from scratch**

**1. Prerequisites**

**1.1. Node.js:**

Install the latest Node.js **Recommended For Most Users** from <https://nodejs.org/en/>

**1.2. Install Git:**

<https://git-scm.com/>

**1.3. Install Java SE Development Kit:**

<https://www.oracle.com/technetwork/java/javase/downloads/jdk12-downloads-5295953.html>

===

**Additional MacOS prerequisites:**

**1.4. Install Xcode:**

<https://developer.apple.com/xcode/>

===

**Additional Windows prerequisites:**

**1.5. Install all the required tools and configurations using Microsoft's windows-build-tools:**

Open Command Prompt **as administrator** and run the following script:

npm install --global --production windows-build-tools

**1.6. Install Python 2.7.**

<https://www.python.org/downloads/release/python-2716/>

**1.7. Use Git Bash instead of using Windows Command Prompt (for WebStorm).**

1. Open WebStorm
2. Go to File => Settings
3. Go to Tools => Terminal
4. Specify Shell Path to Git Bash e.g.C:\Program Files\Git\bin\sh.exe

**2. Creating project**

**2.1. Create project folder:**

Open terminal and navigate to desired folder where you're going to store all of your projects. For example projectsfolder:

cd projects

Create a folder for automation project. For example, automation-webdriverio:

mkdir test-automation-webdriverio

And navigate to the created folder:

cd test-automation-webdriverio

**2.2. Initialize your project:**

npm init -y

This action creates package.json file.

**3. Modules installation and configuration**

**3.1. Install webDriver I/O:**

npm i webdriverio

**3.2. Install CLI:**

npm i @wdio/cli

**3.3. Create WebDriver I/O configuration by running:**

./node\_modules/.bin/wdio config

This script runs WDIO Configuration Helper which will help you to create configuration file.

Where should your tests be launched > local \*\*click Enter\*\*

Shall I install the runner plugin for you? > type Y \*\*click Enter\*\*

Where is your automation backend located? > On my local machine \*\*click Enter\*\*

Which framework do you want to use? > mocha \*\*click Enter\*\*

Shall I install the framework adapter for you? > type Y \*\*click Enter\*\*

Do you want to run WebdriverIO commands synchronous or asynchronous? > sync \*\*click Enter\*\*

Where are your test specs located? > type ./test/\*.js \*\*click Enter\*\*

Which reporter do you want to use? > dot (select by pressing Space) > spec (select by pressing Space) > allure (select by pressing Space) \*\*click Enter\*\*

Shall I install the reporter library for you? > type Y \*\*click Enter\*\*

Do you want to add a service to your test setup? > selenium-standalone (select by pressing Space) \*\*click Enter\*\*

Shall I install the services for you? > type Y \*\*click Enter\*\*

Level of logging verbosity > silent \*\*click Enter\*\*

What is the base url? > type https://reactbugtracker.com \*\*click Enter\*\*

Wait till the end of the installation process.

**4. Update WebDriver I/O configuration:**

Open configuration file:

open wdio.conf.js

**4.1. Configure browser:**

maxInstances: 1, browserName: 'chrome'

**4.2. Configure reporters:**

Add the following code under reporters: ['dot','spec','allure'],:

reporterOptions: { allure: { outputDir: 'allure-results' } },

**5. Modify test script:**

Open package.json:

open package.json

and modify test script to get the following:

"test": "wdio wdio.conf.js"

**6. Creating the first test**

**6.1. Create test folder and open it:**

mkdir test cd test

**6.2. Create test.js file and open it:**

touch test.js open test.js

**6.3. Add the first test:**

import assert from 'assert'; describe('Client', function () { //define suite title by passing a string describe('Page Level', function () { //define sub-suite title by passing a string it('Get title', function () { //define test title by passing a string browser.url('/'); //open baseUrl + string passed in .url() function let title = browser.getTitle(); //get page title and assign it to the "title" variable browser.pause(2000); //just pause to visually see that something is happening on the page assert.equal(title, 'Bug Tracker'); //compare {title} (actual) and "Bug Tracker" (expected) }) }); });

**8. Instal Babel to use JavaScript ES6 syntax**

**8.1. Go back to the root folder and install necessary modules:**

cd .. npm install @babel/core @babel/cli @babel/preset-env @babel/register

**8.2. Create and open Babel Configuration file:**

touch babel.config.js open babel.config.js

**8.3. Paste the following code:**

module.exports = { presets: [ ['@babel/preset-env', { targets: { node: 8 } }] ] }

**8.5 Update WebDriver I/O configuration adding Babel configuration:**

Open configuration file:

open wdio.conf.js

And add the following line to the mochaOpts objest:

compilers: ['js:@babel/register']

It should look like this at the end:

mochaOpts: { ui: 'bdd', timeout: 60000, compilers: ['js:@babel/register'] },

**9. Running the first test**

**9.1. Run the first test:**

npm test

Wait until test is finished. You should see the message with one suite, one sub-suite, and one test passed:

[chrome #0-0] Client [chrome #0-0] Page Level [chrome #0-0] ✓ Get title [chrome #0-0] [chrome #0-0] 1 passing (3.4s)

**10. Adding more tests:**

Replace the old code by the following one in test.js:

import assert from 'assert'; describe('Client', function () { describe('Page Level', function () { it('Get title', function () { browser.url('/'); let title = browser.getTitle(); assert.equal(title, 'Bug Tracker'); }) it('Favicon', function () { browser.url('/favicon.ico'); let icon = $('img'); let width = icon.getCSSProperty('width').parsed.value; let height = icon.getCSSProperty('height').parsed.value; let size = `${width}x${height}`; assert.equal(size, '256x256'); }) }); describe('Elements exist', function () { it('Header', function () { browser.url('/'); let header = $('.custom-header').isDisplayed(); assert.equal(header, true); }) it('App', function () { let header = $('.site-content').isDisplayed(); assert.equal(header, true); }) it('Footer', function () { let header = $('.custom-footer').isDisplayed(); assert.equal(header, true); }) }); });

Now your code testing Client suite, Page Level and Elements Exist sub-suites, and five tests. More info regarding Webdriver I/O commands you can find here: <https://webdriver.io/docs/api.html>

**11. Working with Git:**

**TODO:**

GIT .gitignore