**Mocha Framework**

WebDriverIO is one of the open source testing framework which is designed to automate modern web applications using NodeJS.

In my project we have used mocha framework. Mocha is an open source JavaScript testing framework that runs on Node.js and in the browser. It’s designed for testing both synchronous and asynchronous code.

The default interface for mocha is BDD approach. Apart from that it supports TDD, Exports and QUnit. While mocha supports different assertion, we have used ‘Chai’ assertions in our project.

Mocha provides us describe block and it block for writing and executing test scripts. It also provides us few hooks for giving pre and post conditions.

Hooks of Mocha:

* before(): Runs once before first it block.
* after(): Runs once after last it block.
* beforeEach(): Runs before each it block.
* afterEach(): Runs after each it block.

As we are implementing WebDriverIO, it gives us a set of predefined libraries which includes:

1. **node\_modules:** It contains all the dependencies executable files and related libraries and service configuration. Whenever we will add some dependency, the corresponding packages and executable files will be loaded here.
2. **test:** This folder contains test related sub-folders.
   * + 1. pageobjects: It contains all the POM classes where we are storing web elements of its respective pages.
       2. specs: This folder contains all the test scripts which we want to execute for our project.
       3. utilities: It contains all the generic utilities files which is useful for the project. Different utilities like excel utility, jsutility(Random number, Date)
3. **package**.**json**:
4. **wdio.conf.js:** The configuration file contains all necessary information to run your test suite. It’s having different configuration for different purpose.

**Runner:** To run the code in local or GCP.

**Specs:** It defines the file which we want to run. We have to mention the relative path to run. \* for executing all the test scripts in specs folder.

**Exclude:** Excludes the particular which we don’t want to run.

**Suite:** It is used to grouping the test scripts. So that is can be segregate into smoke and regression.

**Capabilities:** One of the configuration which allows us to run our test scripts in multiple browsers. Cross browser testing can achieved.

**Bail:** Whenever we want to stop execution after some test script failures then we will be going for bail. Default is 0

**Base Url:** Without writing the URL in test scripts we can mention here to launch directly.

**connectionRetryTimeout:** Timeout for any WebDriver request to a driver or grid.

**connectionRetryCount:** For how many times it should retry.

**Services:** We have to mention the driver executable files here.

**Reporters:** It is used to get reports like spec and allure. Allure reports contains passed test scripts and passed test scripts with its percentage. It will also store the screenshots of fail test scripts. To take a screenshot we have to use afterTest function.

**Execution:**

**For Batch:**

suite :{

        Smoke: ['. /test/specs/practice.js]

    },

We have to mention suite function after exclude. To execute w.r.to suite we have to the below code:

***npx wdio run ./wdio.conf.js –suite Smoke***

**For parallel:** For parallel test execution we have to increase max instances inside the capabilities

**For cross-browser:** We have to add different browser name to perform cross browser testing.

**Through package.json file:** Here we have to store the relative path of test scripts as value and we have to give a name for it as the key. To run it we need to use ***npm run “Key”.***

**Mochaopts Keyword:** We can run the scripts based on keywords present inside the describe block and it block.

***npx wdio run wdio.conf.js --mochaOpts.grep "keyword with spaces"***

***npx wdio run wdio.conf.js --mochaOpts.grep keyword***

**Dependencies:**

 "@wdio/allure-reporter": "^8.15.10",

    "@wdio/cli": "^8.15.6",

    "@wdio/local-runner": "^8.15.6",

    "@wdio/mocha-framework": "^8.15.6",

    "@wdio/spec-reporter": "^8.15.6",

    "allure-commandline": "^2.24.0"