Testing web applications using Mocha and WebdriverIO



Overview

WebdriverIO is the next-gen browser and mobile automation test framework for Node.js.

Prerequisites

For this example we will use the WebdriverIO framework with the default assertion library provided by WebdriverIO and Junit as the reporter.

In the official documentation we can find the following description:

WebdriverIO is a progressive automation framework built to automate modern web and mobile applications. It simplifies the interaction with your app and provides a set of plugins that help you create a scalable, robust and flakiness test suite.

If we want, we can use other runners or reporters.

We will need:

- · Access to heroku demo site that we aim to test
- Node.js environment with WebDriverIO

To start using WebDriverIO please follow the Get Started documentation.

WebDriverIO provides a client that after being installed will guide you through bootstraping a *Hello World* test suite into your project. For this tutorial we will use the code generated by this tool for simplicity (with page objects). The test consists in validating the login feature (with valid and invalid credentials) of the de mo site, for that we have created a base page object that will contain all methods and functionality that is shared across all page objects, a login page, that will extend the base page, that will have all the methods for interacting with the login page and a result page that will have the methods to interact in the page that is loaded after the login operation.

./pageobjects/Page.js

```
/**
* main page object containing all methods, selectors and functionality
* that is shared across all page objects
*/
module.exports = class Page {
    /**
    * Opens a sub page of the page
    * @param path path of the sub page (e.g. /path/to/page.html)
    */
    open (path) {
        return browser.url(`https://the-internet.herokuapp.com/${path}`)
    }
}
```

./pageobjects/login.page.js

```
const Page = require('./page');
/**
\ast sub page containing specific selectors and methods for a specific page
 */
class LoginPage extends Page {
   /**
    * define selectors using getter methods
    */
   get inputUsername () { return $('#username') }
    get inputPassword () { return $('#password') }
   get btnSubmit () { return $('button[type="submit"]') }
    /**
    * a method to encapsule automation code to interact with the page
    \star e.g. to login using username and password
    */
    async login (username, password) {
       await (await this.inputUsername).setValue(username);
       await (await this.inputPassword).setValue(password);
       await (await this.btnSubmit).click();
    }
    /**
    * overwrite specifc options to adapt it to page object
    */
    open () {
       return super.open('login');
    }
}
module.exports = new LoginPage();
```

./pageobjects/secure.page.js

```
const Page = require('./page');
/**
 * sub page containing specific selectors and methods for a specific page
 */
class SecurePage extends Page {
    /**
    * define selectors using getter methods
    */
    get flashAlert () { return $('#flash') }
}
module.exports = new SecurePage();
```

Define the test that will assert if the operation is successful or not

```
./example.e2e.js
const LoginPage = require('../pageobjects/login.page');
const SecurePage = require('../pageobjects/secure.page');
describe('My Login application', () => {
    it('should login with valid credentials', async () => {
        await LoginPage.open();
        await LoginPage.login('tomsmith', 'SuperSecretPassword!');
        await expect(SecurePage.flashAlert).toBeExisting();
        await expect(SecurePage.flashAlert).toHaveTextContaining(
            'You logged into a secure area!');
    });
});
describe('My Login application', () => {
    it('should not login with invalid credentials', async () => {
        await LoginPage.open();
        await LoginPage.login('tom', 'SuperPassword!');
        await expect(SecurePage.flashAlert).toBeExisting();
        await expect(SecurePage.flashAlert).toHaveTextContaining(
            'Your username is invalid!');
    });
});
```

All of the above were created by the tool provided by WebDriverIO, to create those we followed the docu mentation and executed first the command to install the WebDriverIO test runner:

npm install @wdio/cli

Then we answer a series of questions that will define the code to be generated using:

npx wdio config

The output of the questionnaire will look like this:



The last two steps to have everything configured is to define that we will use the Junit framework, for that we execute the following command:

```
npm install @wdio/junit-reporter --save-dev
```

In wdio.conf.js , we have added, in the reporters area, the following Junit definition:

```
....
reporters: ['spec',
    ['junit', {
        outputDir: './',
        outputFileFormat: function(options) { // optional
            return `results.xml`
        }
      }]
    ],
....
```

Once the code is implemented (and we will make it fail on purpose on one test, to show the failure reports), it can be executed with the following command:



The results are immediately available in the terminal.



In this example, one test has failed and the other one has succeed, the output generated in the terminal is the above one and the correspondent Junit report is as below:

```
Junit Report

<pre
```

```
<property name="specId" value="0"/>
      <property name="suiteName" value="My Login application"/>
      <property name="capabilities" value="chrome.91_0_4472_101.macosx"/>
      <property name="file" value="./test/specs/example.e2e.js"/>
    </properties>
    <testcase classname="chrome.91_0_4472_101.macosx.My Login application"
name="should login with valid credentials" time="2.369">
      <system-out><![CDATA[
COMMAND: POST /session/80357f832dc7f646258291140deaecbe/url - {"url":"
https://the-internet.herokuapp.com/login"}
RESULT: {"url":"https://the-internet.herokuapp.com/login"}
COMMAND: POST /session/80357f832dc7f646258291140deaecbe/element -
{"using":"css selector","value":"#username"}
RESULT: {"using":"css selector","value":"#username"}
COMMAND: POST /session/80357f832dc7f646258291140deaecbe/element/bdbe5639-
b66b-4f3e-9551-d701b1040909/clear - {}
RESULT: {}
COMMAND: POST /session/80357f832dc7f646258291140deaecbe/element/bdbe5639-
b66b-4f3e-9551-d701b1040909/value - {"text":"tomsmith"}
RESULT: { "text": "tomsmith" }
COMMAND: POST /session/80357f832dc7f646258291140deaecbe/element -
{"using":"css selector","value":"#password"}
RESULT: {"using":"css selector","value":"#password"}
COMMAND: POST /session/80357f832dc7f646258291140deaecbe/element/f9b56cb7-
f274-4370-889d-044db3b07ecb/clear - {}
RESULT: {}
COMMAND: POST /session/80357f832dc7f646258291140deaecbe/element/f9b56cb7-
f274-4370-889d-044db3b07ecb/value - {"text":"SuperSecretPassword!"}
RESULT: {"text":"SuperSecretPassword!"}
COMMAND: POST /session/80357f832dc7f646258291140deaecbe/element ·
{"using":"css selector","value":"button[type=\"submit\"]"}
RESULT: {"using":"css selector","value":"button[type=\"submit\"]"}
COMMAND: POST /session/80357f832dc7f646258291140deaecbe/element/3f43e518-
e698-4bff-b647-6255d6fbeab5/click - {}
RESULT: {}
COMMAND: POST /session/80357f832dc7f646258291140deaecbe/element -
{"using":"css selector","value":"#flash"}
RESULT: {"using":"css selector","value":"#flash"}
COMMAND: POST /session/80357f832dc7f646258291140deaecbe/elements -
{"using":"css selector","value":"#flash"}
RESULT: {"using":"css selector","value":"#flash"}
COMMAND: POST /session/80357f832dc7f646258291140deaecbe/element -
{"using":"css selector","value":"#flash"}
RESULT: {"using":"css selector","value":"#flash"}
COMMAND: GET /session/80357f832dc7f646258291140deaecbe/element/7e87ed7b-
be88-4023-bb59-b3439d06d53f/text - {}
RESULT: {}
]]></system-out>
    </testcase>
  </testsuite>
  <testsuite name="My Login application" timestamp="2021-06-12T08:42:23"
time="10.709" tests="1" failures="1" errors="1" skipped="0">
    <properties>
      <property name="specId" value="0"/>
      <property name="suiteName" value="My Login application"/>
      <property name="capabilities" value="chrome.91_0_4472_101.macosx"/>
      <property name="file" value="./test/specs/example.e2e.js"/>
    </properties>
    <testcase classname="chrome.91 0 4472 101.macosx.My Login application"
name="should not login with invalid credentials" time="10.707">
      <failure/>
      <error message="Expect $(`#flash`) to have text containing</pre>
[32m- Expected - 1[39m
[31m+ Received + 2[39m
[32m- Your username is invalid[7m.[27m[39m
[31m+ Your username is invalid[7m![27m[39m
[31m+ ×[39m"/>
      <system-out><![CDATA[
COMMAND: POST /session/80357f832dc7f646258291140deaecbe/url - {"url":"
```

```
https://the-internet.herokuapp.com/login" }
RESULT: {"url":"https://the-internet.herokuapp.com/login"}
COMMAND: POST /session/80357f832dc7f646258291140deaecbe/element -
{"using":"css selector","value":"#username"}
RESULT: {"using":"css selector","value":"#username"}
COMMAND: POST /session/80357f832dc7f646258291140deaecbe/element/af5062d3-
a7f3-41f2-9fb1-9d7db3641681/clear - {}
RESULT: {}
COMMAND: POST /session/80357f832dc7f646258291140deaecbe/element/af5062d3-
a7f3-41f2-9fb1-9d7db3641681/value - {"text":"tom"}
RESULT: { "text": "tom" }
COMMAND: POST /session/80357f832dc7f646258291140deaecbe/element -
{"using":"css selector","value":"#password"}
RESULT: {"using":"css selector","value":"#password"}
COMMAND: POST /session/80357f832dc7f646258291140deaecbe/element/b4df7026-
0164-4c38-93ee-419a9b963d67/clear - {}
RESULT: {}
COMMAND: POST /session/80357f832dc7f646258291140deaecbe/element/b4df7026-
0164-4c38-93ee-419a9b963d67/value - {"text":"SuperPassword!"}
RESULT: { "text": "SuperPassword!" }
COMMAND: POST /session/80357f832dc7f646258291140deaecbe/element -
{"using":"css selector","value":"button[type=\"submit\"]"}
RESULT: {"using":"css selector","value":"button[type=\"submit\"]"}
COMMAND: POST /session/80357f832dc7f646258291140deaecbe/element/457ae344-
2625-495d-a17c-fal189ceb0c8/click - {}
RESULT: {}
COMMAND: POST /session/80357f832dc7f646258291140deaecbe/element -
{"using":"css selector","value":"#flash"}
RESULT: {"using":"css selector","value":"#flash"}
COMMAND: POST /session/80357f832dc7f646258291140deaecbe/elements -
{"using":"css selector","value":"#flash"}
RESULT: {"using":"css selector","value":"#flash"}
COMMAND: POST /session/80357f832dc7f646258291140deaecbe/element -
{"using":"css selector","value":"#flash"}
RESULT: {"using":"css selector","value":"#flash"}
COMMAND: GET /session/80357f832dc7f646258291140deaecbe/element/f73dfed1-
9b8b-4dd0-86c2-ca9e2dea499f/text - {}
RESULT: {}
```

```
COMMAND: GET /session/80357f832dc7f646258291140deaecbe/element/f73dfed1-
9b8b-4dd0-86c2-ca9e2dea499f/text - {}
RESULT: { }
COMMAND: GET /session/80357f832dc7f646258291140deaecbe/element/f73dfed1-
9b8b-4dd0-86c2-ca9e2dea499f/text - { }
RESULT: {}
COMMAND: GET /session/80357f832dc7f646258291140deaecbe/element/f73dfed1-
9b8b-4dd0-86c2-ca9e2dea499f/text - {}
RESULT: {}
COMMAND: GET /session/80357f832dc7f646258291140deaecbe/element/f73dfed1-
9b8b-4dd0-86c2-ca9e2dea499f/text - {}
RESULT: {}
COMMAND: GET /session/80357f832dc7f646258291140deaecbe/element/f73dfed1-
9b8b-4dd0-86c2-ca9e2dea499f/text - {}
RESULT: { }
COMMAND: GET /session/80357f832dc7f646258291140deaecbe/element/f73dfed1-
9b8b-4dd0-86c2-ca9e2dea499f/text - { }
RESULT: {}
COMMAND: GET /session/80357f832dc7f646258291140deaecbe/element/f73dfed1-
9b8b-4dd0-86c2-ca9e2dea499f/text - {}
RESULT: {}
COMMAND: GET /session/80357f832dc7f646258291140deaecbe/element/f73dfed1-
9b8b-4dd0-86c2-ca9e2dea499f/text - {}
RESULT: {}
COMMAND: GET /session/80357f832dc7f646258291140deaecbe/element/f73dfed1-
9b8b-4dd0-86c2-ca9e2dea499f/text - {}
COMMAND: DELETE /session/80357f832dc7f646258291140deaecbe - {}
RESULT: {}
RESULT: { }
]]></system-out>
      <system-err><![CDATA[
Error: Expect $(`#flash`) to have text containing
[32m- Expected - 1[39m
[31m+ Received + 2[39m
[32m- Your username is invalid[7m.[27m[39m
[31m+ Your username is invalid[7m![27m[39m
[31m+ ×[39m
    at Context.<anonymous> (/Users/cristianocunha/Documents/Projects
/webdriverio/test/specs/example.e2e.js:21:45)
    at processTicksAndRejections (internal/process/task_queues.js:93:5)
    at async Context.executeAsync (/Users/cristianocunha/Documents/Projects
/webdriverio/node_modules/@wdio/utils/build/shim.js:136:16)
    at async Context.testFrameworkFnWrapper (/Users/cristianocunha
/Documents/Projects/webdriverio/node_modules/@wdio/utils/build/test-
framework/testFnWrapper.js:52:18)
]]></system-err>
    </testcase>
  </testsuite>
</testsuites>
```

Notes:

 There are a lot of other options on how to use WebDriverIO, please check their documentation for more information.

Integrating with Xray

As we saw in the example above where we produced Junit reports with the result of the tests, it is now a matter of importing those results to your Jira instance. This can be done by simply submitting automation results to Xray through the REST API, by using one of the available CI/CD plugins (e.g. for Jenkins) or using the Jira interface to do so.

API

API

Once you have the report file available you can upload it to Xray through a request to the REST API endpoint for JUnit. The first step is to follow the instructions in v1 or v2 (depending on your usage) to obtain the token we will be using in the subsequent requests.

JUnit XML results

We will use the API to perform the below request with the definition of some common fields on the Test Execution, such as the target project, Test Plan, etc.

```
curl -H "Content-Type: multipart/form-data" -u
$JIRA_USERNAME:$JIRA_PASSWORD -F "file=@results.xml" $JIRA_BASEURL/rest
/raven/1.0/import/execution/junit?projectKey=XT&testPlanKey=XT-136
```

With this command we are creating a new Test Execution in the referred Test Plan with a generic summary and two tests with a summary based on the test name.

*	Test E	Add Test	Executions					
	₽~					Show 10 v	entries	Columns -
		¢ Key	§ Summary	0 WTests	¢ Issue Assignee	Test Environments	Status	
	0	XT-218	Execution results - results.xml - [1624456772482]	2	Xpand IT Admin	None		

In Xray we can see that the tests are associated to a Test Plan and we can identify what tests are failing or passing, below you can see two tests (for valid and invalid credentials):



Jenkins

Jenkins

As you can see below we are adding a post-build action using the "*Xray: Results Import Task*" (from the X ray plugin available), where we have some options, we will focus on the one called "*Junit XML*."

Junit XML

• the Jira instance (where you have your Xray instance installed)

- the format as "JUnit XML"
- the test results file we want to import
- the Project key corresponding of the project, in Jira, where the results will be imported

	mport Task		
Jra Instance	https://www-cc-demo.atlassian.net/		
Format	JUNITIONE		
Parameters			
	Import to Same Test Execution		
		When this option is check, if you are importing multiple execution report files using a glob expression, the results will be imported to the same Test Execution	
	Execution Report File (file path with file name)	results.vml	
	Project Key	хт	
	Test Execution Key		
	Test Plan Key	XT-126	
	Test Environments		
	Besision		
	Fix Version		

Tests implemented will have a corresponding Test entity in Xray. Once results are uploaded, Test issues are auto-provisioned, unless they already exist.

of the test of tes	ay Tutorials / hould not	xt-213 t login	with invali	d crede	entials					
🖋 Edit	Q Commer	t Assi	ign More v	To Do	In Progress	Done	Admin ~			
✓ Details										
Type:		🖸 Test				Sta	us:	TO DO	/iew Workflow)	
Priority:		O Trivia	al			Res	olution:	Unresolv	ed	
Labels:		None								
 Descript Click to a 	tion add descriptic	in								
✓ Test Det	ails									
Type:		Generic								
Definitio	n:	chrome.	91_0_4472_106	macosx.My	Login applica	ation.shou	ıld not login wi	th invalid crede	ntials	
Part Extension of the second s	ay Tutorials / xecution	xT-218 results	s - results.:	xml - [1	624455	7724	82]			
🖋 Edit	Q Commen	t Assi	gn More 🗸	To Do	In Progress	Done	Admin ~			
 Details 										
Type:		🖸 Test 8	Execution			Stat	us:	TO DO (V	iew Workflow)	
Priority:		O Trivia				Res	olution:	Unresolve	d	
Labels:		None								
Test Plan	1:	XT-136								
Test Envi	ironments:	None								
v Descript	ion									
Execution	n results impo	rted from	external source							
 Tests 										
Overall Ex	ecution Status									+ Add ~
2 PAS	s									
Total Test	ts: 2									
7	F Filter(s)									
∎~	Filter(s) Apply Rank								Show 100 v entries	Columns -
	🔻 Rank	() Key	Summary		👌 Test Typ	pe #Rec	#Def	Assignee	tatus status stat	
	2	XT-213	should not login credentials	with invalid	Generic	0	0	Xpand IT Admin	PASS	•
	1	XT-212	should login wit credentials	h valid	Generic	0	0	Xpand IT Admin	PASS	• •
Showing	1 to 2 of 2 entr								Eirest Demoisture	A Next Last

Xray uses a concatenation of the suite name and the test name as the the unique identifier for the test (In this case it was prefixed with the browser name and version).

In Xray, results are stored in a Test Execution, usually a new one. The Test Execution contains a Test Run per each test that was executed.

*	Tests									+ Add ×	Votes: Watchers:
	Overall Ex	ecution State	n								Dates Created: Updated:
	Total Test	n: 2 Filter(s)									 Agile View on Board
	•	Apply Ra	nk † Key	¢ Summary	0 Test Type	#Req	#Def	Assignee	Show 100 v entries	Columns -	
		2	XT-213	should not login with invalid credentials	Generic	0	0	Xpand IT Admin	PASS	•	
		1	XT-212	should login with valid credentials	Generic	0	0	Xpand IT Admin	PASS	Execution I	Details th Exploratory App
*	Showing 1to 2 of 2 entries First Previous 👔									TODO	6
		FAIL									
> ~	Structure	•								ABORTED BLOCKED PENDING	

Detailed results, including logs and exceptions reported during execution of the test, can be seen on the execution screen details of each Test Run, accessible through the *Execution details* as we can see here:

Execution	Details					
Test Description						~
None						
Custom Fields						~
There are no Test Run (Custom Fields defined.					
Test Details						^
Test Type:	Generic					
Definition:	chrome.91_0_4472_106.macoax.M	y Login application should not login with it	valid credentials			
Results						^
Contest		Output		Duration	Status	
TestSuite M	Login application	-		645.000 ms	PASS	
Activity						~

Jira UI

Jira UI

💌 Test Runs					
					Execute In 👻
Verified Verified	sion (project		P-4		New Test Execut
All Projects v S	endent) Status	V DD-MM	ENG DD	MM-YYYY HH:MM C	Exploratory App.
Key Ey D	vision Executed Started	Finished De	forts Test	Show 10 v entrie	es Columns -
♦ ♦ Version/s ♦	⇔ _{By} ≎	¢	Environment	а ^ф	
XT- 218	Xpand IT 11 minutes Admin ago	ago	None	PASS	•
Create new te	est execution to	run XT-212	2		
Project	Xray Tutorials		~		
Summary	Ad-boc execution	4			
ounnury	Au not execution	for should logi	n with valid c	recentiais	1
Assignee	O Xpand IT Admi	n	n with valid c	recentiais	-
Assignee	Xpand IT Admi Choose a user to assign	n the Test Execution	n with valid c	redentials	~
Assignee	Xpand IT Admi Choose a user to assign Blocker	n the Test Execution	n with valid c	recentials	•
Assignee Priority	Xpand IT Admi Choose a user to assign Blocker Start typing to get a lis	n the Test Execution	n with valid o	vn to select.	•
Assignee Priority Fix Version/s	Xpand IT Admi Choose a user to assig Blocker Start typing to get a lis	n the Test Execution	n with valid c	vn to select.	•
Assignee Priority Fix Version/s	Xpand IT Admi Xpand IT Admi Choose a user to assign Blocker Start typing to get a lis	n the Test Execution of possible match	n with valid o	vn to select.	· · ·
Assignee Priority Fix Version/s Sprint	Xpand IT Admi Xpand IT Admi Choose a user to assign Blocker Start typing to get a lis	n the Test Execution t of possible match	n with valid o	vn to select.	•
Assignee Priority Fix Version/s Sprint	Xpand IT Admi Xpand IT Admi Choose a user to assign Blocker Start typing to get a lis Start typing to get a lis Start typing to get a lis	n the Test Execution to possible match to f po	n with valid c	vn to select. vn to select.	•
Assignee Priority Fix Version/s Sprint Test Environments	Xpand IT Admi Xpand IT Admi Chose a user to assig Blocker Start typing to get a lis Start typing to get a lis	n the Test Execution of the Test Execution of the Test Execution of the test of test o	n with valid c	vn to select. vn to select.	•
Assignee Priority Fix Version/s Sprint Test Environments	Xpand IT Admi Xpand IT Admi Chose a user to assig Blocker Start typing to get a lis	n the Test Execution t of possible match t of possible match t of possible match t of possible match	In with valid c	vn to select. vn to select. vn to select. vn to select.	•
Assignee Priority Fix Version/s Sprint Test Environments	Xpand IT Admi Xpand IT Admi Chose a user to assig Blocker Start typing to get a lis Each environment when	n the Test Execution to possible match t of possible match t of possible match t of possible match t of possible match	In with valid c	vn to select. vn to select. vn to select.	•
Assignee Priority Fix Version/s Sprint Test Environments Revision	Xpand IT Admi Xpand IT Admi Choses a user to assig Blocker Start typing to get a lis Start typing to get a lis	n n tor should regi n tor possible match tor possib	In with valid c	vn to select. vn to select. vn to select.	
Assignee Priority Fix Version/s Sprint Test Environments Revision	Xpand IT Admi Xpand IT Admi Chose a user to assig Blocker Start typing to get a lis Start typing to get a lis Start typing to get a lis Exh environment when The system revision for	n the Test Execution n the Test Execution t of possible match t of possible match	n with valid c on hes or press dow hes or press dow hes or press dow hes or press dow	vn to select. vn to select. vn to select. vn to select.	•

3

Open the Test Execution and import the JUnit report.

Ac	y Tutorials /	xT-219 (ecutio	on fe	or should login	with valie	d crede	ntials			
🖋 Edit	Q Commer	nt Ass	ign	More 👻 To Do	In Progress	Done A	dmin 🛩			
 Details 				Log work						
Type: Priority: Labels: Test Plan:		Test Block None None	Exec	Agile Board Rank to Top Rank to Bottom		Status: Resoluti	on:	Unresolve	lew Workflow) rd	
Test Envir	Test Environments: None			Archive						
Description Click to address to add	Description Click to add description			Attach files Attach Screenshot						
🗹 Tests				Voters Stop watching Watchers						+ Add ~
Overall Exe	ecution Statue			Create sub-task Convert to sub-task						
1 торо				Move Link						
Total Tests	:: 1 Filter(s)			Clone Labels						
 · · ·	Annly Ran	ć		Delete					Phone 100 M option	Columna -
_y ·	• Rank	¢ Key	¢ s	Reset Defect Count	Test Type	#Req	#Def	Assignee		corumns •
	1	XT-212	she	Export to Cucumber	ana Jeneric	0	0	Xpand IT	TODO	•
			cre	import execution Resi	JILS			Admin		
Showing 1	to 1 of 1 entr	es		Export lest Runs to C	57				First Previous	1 Next Last

4

Choose the results file and press "Import."

Import Execution Results	
Choose file No file chosen The file with the execution results for the Test Execution.	
Import Cancel	

5

The Test Execution is now updated with the test results imported.

D	Execution E	Details			
Test D	escription				
None					
Custo	m Fields				
There a	ne no Test Run Ci	istom Fields defined.			
Test	Details				
Test	Type:	Generic			
Def	nition:	chrome.91_0_4472_396.macosx.My Login application.	should not login with invalid credentials		
Resu	Ita				
	Context		Output	Deration	Stotas
	TestSuite My	Login application		646.000 ms	PASS
Activ	ity				

Tips

- after results are imported in Jira, Tests can be linked to existing requirements/user stories, so you can track the impact of their coverage. results from multiple builds can be linked to an existing Test Plan, to facilitate the analysis of
- ٠ test result trends across builds.
- results can be associated with a Test Environment, in case you want to analyze coverage and test results by that environment later on. A Test Environment can be a testing stage (e.g. dev, staging, preprod, prod) or an identifier of the device/application used to interact with the system (e.g. browser, mobile OS).

References

- https://webdriver.io/https://webdriver.io/docs/gettingstarted