

# Mobile Web Automation using Protractor

AUTOMATION TESTING, TESTING

14 / SEP / 2016

BY [KIMI AGARWAL](#)

[0 COMMENTS](#)

Share this blog

Email

Twitter

Facebook

LinkedIn

Google+

---

Protractor is an end-to-end **automation testing** framework tool. To test Angular web page on a mobile device, we have to use the Appium with Protractor to perform our E2E testing. Both Appium and Protractor are open-source automation testing tool. Appium provides a bridge between a system and a mobile device. It supports three types of mobile application:

- Native
- Mobile Web
- Hybrid application

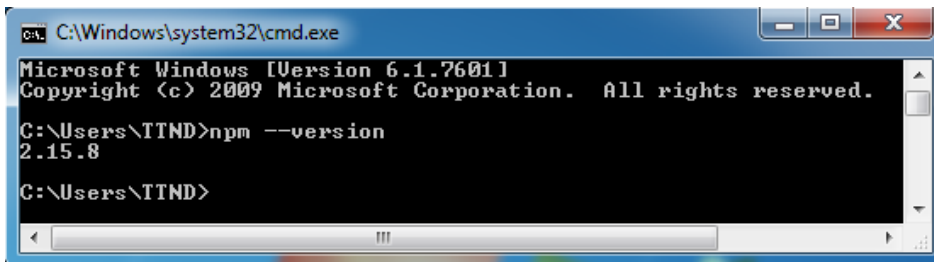
## Step-by-Step Installation

**Step 1:-** Download & install the Chrome browser in the system where we run our test.

**Step 2:-** Download the 'Recommended For Most Users' one Node.js and then install it.

**Step 3:-** Now we need to check whether Node.js is installed correctly or not on our machine

Open console window and type **npm -version** [This will display the npm version]

A screenshot of a Windows Command Prompt window. The title bar reads 'C:\Windows\system32\cmd.exe'. The window content shows the following text: 'Microsoft Windows [Version 6.1.7601] Copyright (c) 2009 Microsoft Corporation. All rights reserved. C:\Users\ITND>npm --version 2.15.8 C:\Users\ITND>'. The window has a blue title bar and a black background for the command prompt.

**Step 4:-** To install Protractor, type **npm install -g protractor** [This command globally installs the Protractor using node package manager]

Type **protractor -version** [This will display the Protractor version]

Type **npm -g list** [This will give all the nodes which you have installed]

**Step 5:-** Install and set up Java

Type **java -version** [You will get the java version]

**Step 6:-** Setting up a Selenium Server

Type **webdriver-manager update** [This is for updating the Chrome driver]

Type **webdriver-manager start** [This command will start up a Selenium Server and will output a cluster of data logs]

**Step 7:-** To install Appium through NPM type **npm install -g appium**

**Step 8:-** Download Appium on your system

Install it by clicking on appium-installer.exe than launch Appium.

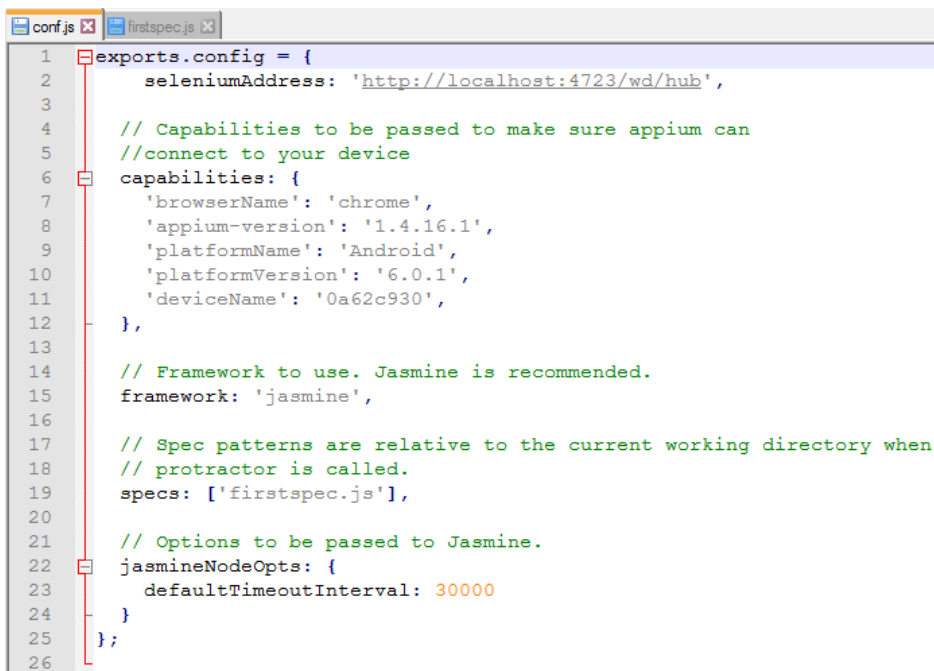
**Step 9:-** Attach mobile device with your system and check that whether the device is attached or not with the help of command **adb devices** on the command prompt window.

**Setup conf.js File**

Normally, Protractor runs test on localhost:4444. However, we need Protractor to connect with Appium which runs test on localhost:4723

Some capabilities must be set in the conf.js to make sure Appium can connect to your device which are given below:

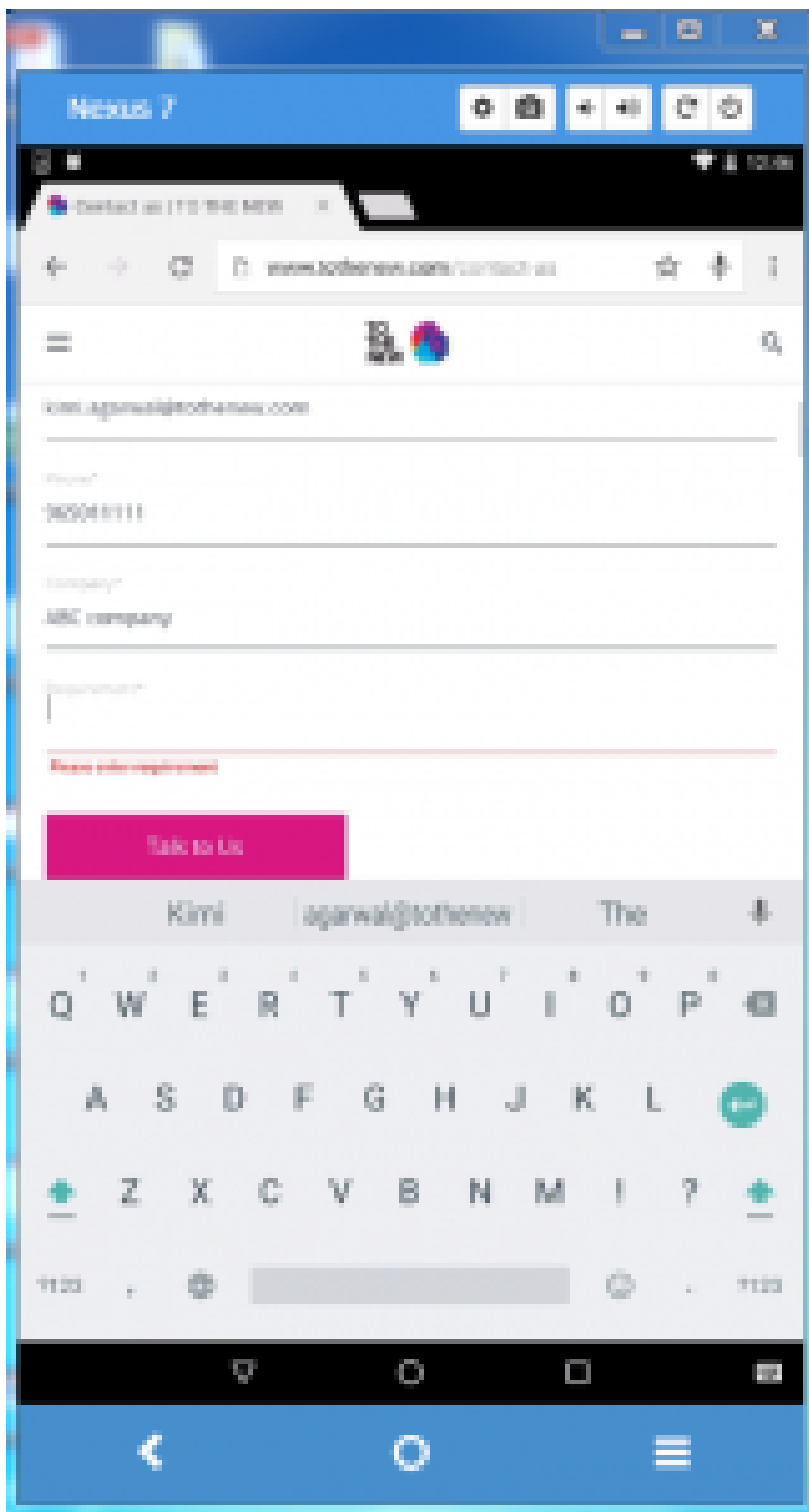
- **appium-version:** Version of Appium tool.
- **platformName:** Name of the device OS platform wants to utilize
- **platformVersion:** Name of the device OS version wants to use
- **deviceName:** Actual device number or name which is connected to our system.



```
1 exports.config = {
2   seleniumAddress: 'http://localhost:4723/wd/hub',
3
4   // Capabilities to be passed to make sure appium can
5   // connect to your device
6   capabilities: {
7     'browserName': 'chrome',
8     'appium-version': '1.4.16.1',
9     'platformName': 'Android',
10    'platformVersion': '6.0.1',
11    'deviceName': '0a62c930',
12  },
13
14  // Framework to use. Jasmine is recommended.
15  framework: 'jasmine',
16
17  // Spec patterns are relative to the current working directory when
18  // protractor is called.
19  specs: ['firstspec.js'],
20
21  // Options to be passed to Jasmine.
22  jasmineNodeOpts: {
23    defaultTimeoutInterval: 30000
24  }
25 };
26
```

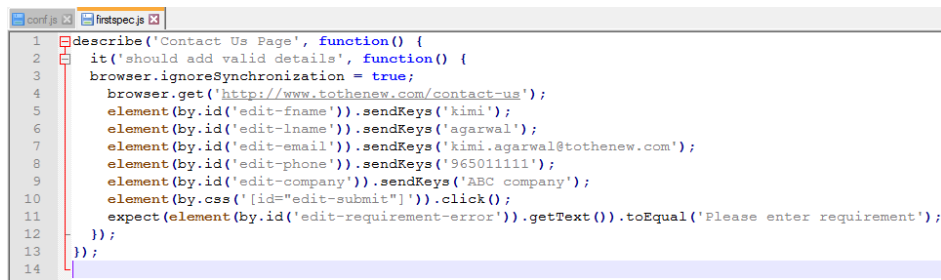
## A Use Case

1. Go to 'Contact Us' page
2. Enter all the valid credentials in the Request for services form leaving behind the 'Requirement' field
3. Click on the 'Talk to Us' Button
4. Verify the error message.



## Our First Test Script

Following is the code piece for our use case.



```

1 describe('Contact Us Page', function() {
2   it('should add valid details', function() {
3     browser.ignoreSynchronization = true;
4     browser.get('http://www.tothenew.com/contact-us');
5     element(by.id('edit-fname')).sendKeys('kimi');
6     element(by.id('edit-lname')).sendKeys('agarwal');
7     element(by.id('edit-email')).sendKeys('kimi.agarwal@tothenew.com');
8     element(by.id('edit-phone')).sendKeys('9650111111');
9     element(by.id('edit-company')).sendKeys('ABC company');
10    element(by.css('[id="edit-submit"]')).click();
11    expect(element(by.id('edit-requirement-error')).getText()).toEqual('Please enter requirement');
12  });
13 });
14

```

As my application is non-angularJS for that I'm using the following command in my script `browser.ignoreSynchronization = true;`

In this test script, in order to click the button we are using the script: `element(by.css('[id="edit-submit"]'))`;

In our mobile device web application, the HTML Code for the button is: `<input data-drupal-selector="edit-submit" type="submit" id="edit-submit" name="op" value="Talk to Us" class="button js-form-submit form-submit">`

## Types of Locators

A locator directly helps Protractor to identify elements. Following is the list of locators we can use:

1. Find an element utilizing a CSS selector `css('myclass')`
2. Find an element with a specified id `id('myid')`
3. Find an element with a specific ng-model `model('name')`
4. Find an element bound to a given variable `binding('bindingname')`

## Run the Test Script

- Once all the above steps are completed. Open a new console window and give the path of the test script  
'AppData\Roaming\npm\node\_modules\protractor\example'.
- Run the script with **protractor conf.js**

As the script runs, the Chrome browser window will maximize on the mobile device and will be navigated to the **TO THE NEW contact us**

page, automatically details in the 'Request for Service' form will be filled and 'Talk to Us' button will be clicked and with that the browser closes itself.

The protractor automation tool is specially designed for **AngularJS application**, but we can also use this tool to test the non-AngularJS application.

**HAPPY TESTING!**