

# 1. Protractor Introduction

1. What is protractor ?
2. Why we use protractor
3. Similarities and differences between Protractor and Selenium with Java/C#
4. Protractor and webdriver architecture

## 1. What is protractor ?

Protractor is an end-to-end test framework for Angular and AngularJS applications. Protractor runs tests against your application running in a real browser, interacting with it as a user would.

## 2. Why we use protractor

### Test Like a User:

Protractor is built on top of WebDriverJS, which uses native events and browser-specific drivers to interact with your application as a user would.

### For Angular Apps:

Protractor supports Angular-specific locator strategies, which allows you to test Angular-specific elements without any setup effort on your part.

(Protractor additional bindings like By.repeater, ByModel and By.binding, By.buttontext, partialbutton text ... provides more options to work with angular directives)

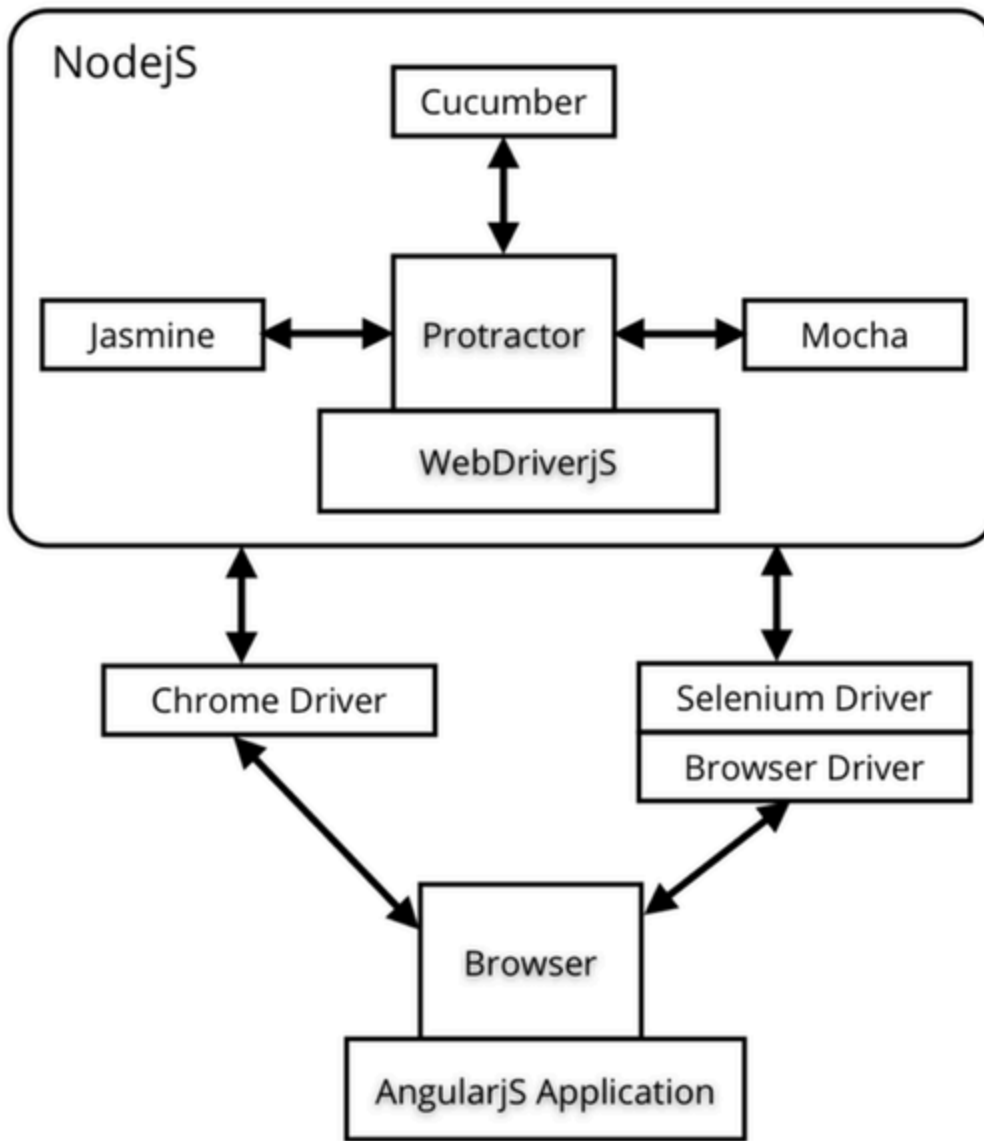
### Automatic Waiting:

You no longer need to add waits and sleeps to your test. Protractor can automatically execute the next step in your test the moment the webpage finishes pending tasks, so you don't have to worry about waiting for your test and webpage to sync.

## 3. Similarities and differences between Protractor and Selenium with Java/C#

	<b>Protractor</b>	<b>Selenium with Java/C#</b>
Locators	Apart from the usual selenium locators, additional bindings like By.repeater, ByModel and By.binding, By.buttontext, partialbutton text Provides More option to work with angular directives	ByName, ID, CSS, xpath et at
IDE	Netbeans (Free)	Eclipse for java (free) and Visual studio for c#(Paid)
Programming language	JavaScript only	C#, Java, Python, Ruby
Browser Support	Chrome, Mozilla, Safari, MSEdge (No different from Selenium)	Chrome, Mozilla, Safari, MSEdge
Support	Online Open source Community along with Selenium Support	Selenium Support
Packages	Whole range of NPM Packages available	Limited to Jar's and assemblies in Java and c#
Reporting	NPM modules available. (protractor-beautiful-reporter, Protractor-jasmine2-screenshot-reporter)	TestNG for Java, Nunit and Mstest for C#
Test frameworks	Jasmine, Mocha, Chai	TestNG for Java, Nunit and Mstest for C#
BDD support	Yes, using Cucumber	Available using SpecFlow
Programming	Asynchronous Programming uses promises and callback. Little tricky compared to OO Programming	Object oriented : Tried and tested.
Debugging	Tricky	Fairly Easy
Synchronization	Automatic Waiting.	Explicit waiting using synchronization
CI Integration	Fairly easy with Jenkins	Easy with Jenkins and other CI tools
Recording	Not possible	Possible with Selenium IDE but rarely works.
Skill Required	JS, Asynchronous Programming	OOP (java and c#)
Browsers	Runs on both real and headless Browsers	Only on real browsers
Speed	about 20-30% faster than Selenium	Normal

#### 4. Protractor and webdriver architecture



Protractor is a framework for automation of functional tests, so its intention isn't to be the only way to test an AngularJS application, but to cover the acceptance criteria required by the user.

It runs on top of the Selenium, and thus provides all the benefits and advantages from Selenium.

In addition, it provides customizable features to test AngularJS applications.

It is also possible to use some drivers which implement WebDriver's wire protocol like [ChromeDriver](#) and [GhostDriver](#), as Protractor runs on top of the Selenium. In the case of ChromeDriver it is possible to run tests without the Selenium Server. However to run tests using GhostDriver you need to use PhantomJS which uses GhostDriver to run tests in Headless mode.

The framework integrated with Jasmine can be used to create and organize tests and user expectations. Jasmine is compatible with Protractor due to which all resources that are extracted from browsers can be used to make tests as promises. Those promises are resolved internally by using the "expect" command from Jasmine. That way the promises work smoothly while creating tests.