

Get a start with



Protractor

end to end testing for AngularJS

for AngularJS Web Automation

TPC November 2016 Meetup

By Abhishek Yadav (@abhishekyd)



Agenda



- Overview of Web Automation
- About Protractor
- About AngularJS
- Difference about Protractor over Selenium WebDriver
- Protractor Installation
- Scripting using Protractor for AngularJS Web Automation
- Working with Jasmine and Protractor
- Framework Development using Protractor

Overview of Web Automation

Now a days every organization prefers automated testing once a product reaches to the stable phase to reduce the testing effort. Since testing cost is an important factor for any project, organizations have started preferring open source test automation tools (which have reached a stage where they now rival the commercial ones) instead of investing in costly commercial testing tools.

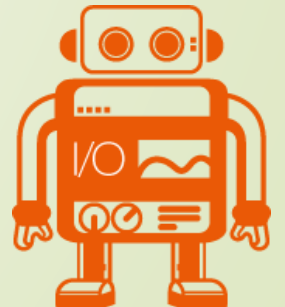
A variety of open source automation testing tools is available for almost all types of testing such as functional, Web, UAT, regression, performance etc. Some open source tools are Selenium, Cucumber, FitNesse, Sahi, Watir etc).

*“Testing is about gaining confidence that your code does what
you think it should do”*



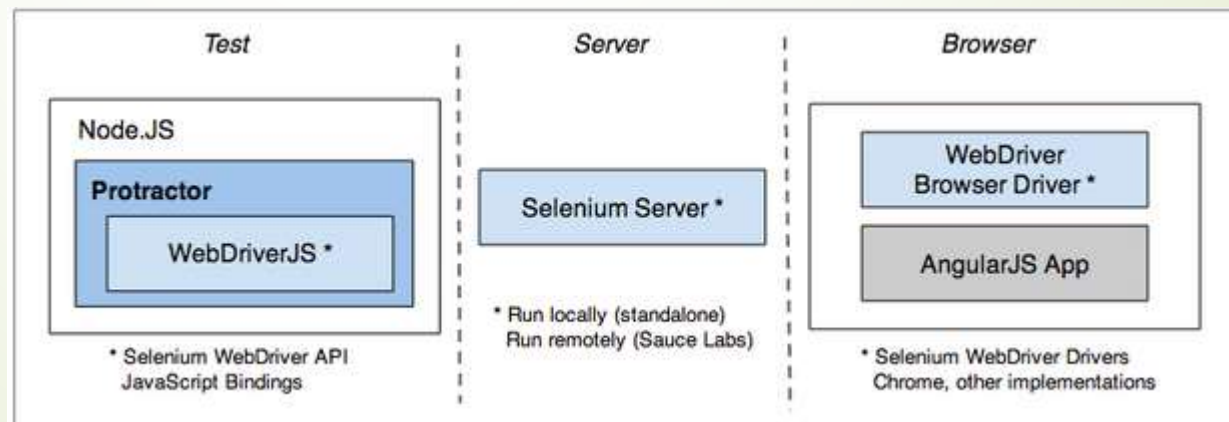
About Protractor

- Protractor is an end-to-end test framework for AngularJS applications. Protractor runs tests against your application running in a real browser, interacting with it as a user would.
- Protractor is built on top of WebDriverJS, which uses native events and browser-specific drivers to interact with your application as a user would.
- It also supports non-angular APP.
- With protractor, you can write e2e tests with JavaScript, the language you write with Angular app.



About Protractor

- Protractor works in conjunction with Selenium to provide an automated test infrastructure that can simulate a user's interaction with an Angular application running in a browser or mobile device.
- it has Angular-specific features
- It's element finders wait for Angular's \$digest loop and \$http to finish. So you'll have less chance to struggle with sleep and timing issues.



About AngularJS

AngularJS is what HTML would have been, had it been designed for building web-apps. Declarative templates with data-binding, MVW, MVVM, MVC, dependency injection and great testability story all implemented with pure client-side JavaScript!



Difference about Protractor over Selenium WebDriver

className
css
id
linkText
js
name
partialLinkText
tagName
xpath

addLocator
binding
exactBinding
model
buttonText
partialButtonText
repeater
exactRepeater
cssContainingText
options
deepCss

<http://www.protractortest.org/#/api?view=ProtractorBy>



Protractor Installation

- Download Node.JS <http://nodejs.org/download/>

```
node --version
```

```
npm install protractor -g
```

```
webdriver-manager update
```

```
webdriver-manager update -ie
```

```
webdriver-manager start
```



Scripting using Protractor for AngularJS Web Automation

- Configuration, copy following into "conf.js":

```
exports.config = {  
  seleniumAddress: 'http://localhost:4444/wd/hub',  
  specs: ['protractor-test.js']  
};
```

- Test, copy following into "protractor-test.js":

```
describe('angularjs homepage', function() {  
  it('should greet the named user', function() {  
    browser.get('http://www.angularjs.org');  
    element(by.model('yourName')).sendKeys('Julie');  
    var greeting = element(by.binding('yourName'));  
    expect(greeting.getText()).toEqual('Hello Julie!');  
  });  
});
```



Scripting using Protractor for AngularJS Web Automation

Capabilities

```
exports.config = {  
  seleniumAddress: 'http://localhost:4444/wd/hub',  
  specs: ['protractor-test.js'],  
  capabilities: {  
    'browserName': 'chrome'  
  },  
  jasmineNodeOpts: {  
    showColors: true  
  }  
};
```

Multicapabilities

```
exports.config = {  
  seleniumAddress: 'http://localhost:4444/wd/hub',  
  specs: ['protractor-test.js'],  
  multiCapabilities: [  
    {  
      'browserName': 'firefox'  
    },  
    {  
      'browserName': 'chrome'  
    }  
  ],  
  jasmineNodeOpts: {  
    showColors: true  
  }  
};
```

<https://github.com/angular/protractor/blob/master/lib/config.ts>

Framework Development using Protractor

- **Page Objects** - These are the js files where you map the elements and write the functions to perform actions;
- **Exports and Require** - This is how you connect your Page Objects to your Test Specs;
- **Test specs** - These are the js files where you write your tests using jasmine syntax.





Page Objects

```
var AngularHomepage = function() {  
    this.nameInput = element(by.model('yourName')); this.greeting =  
        element(by.binding('yourName'));  
  
    this.get = function() { browser.get('http://www.angularjs.org');  
        };  
  
    this.setName = function(name) {  
        this.nameInput.sendKeys(name);  
    };  
};  
  
var AngularHomepage = require('homepage.js');  
  
describe('HomePage Tests', function() {  
    var angularHomepage = new AngularHomepage();  
    angularHomepage.get();  
    angularHomepage.nameInput.sendKeys('Rafael');  
});
```



Separate your tests in various test suites

```
exports.config = {  
  seleniumAddress: 'http://localhost:4444/wd/hub', capabilities: {  
    'browserName': 'chrome'  
  },  
  suites: {  
    homepage: 'homepage/*',  
    search: ['SearchSpec.js']  
  },  
  jasmineNodeOpts: {  
    showColors: true  
  }  
};
```

```
protractor protractor.conf.js --suite homepage
```



Using OnPrepare

```
exports.config = {  
  seleniumAddress: 'http://localhost:4444/wd/hub',  
  capabilities: {  
    'browserName': 'chrome'  
  },  
  onPrepare: function() {  
    browser.driver.manage().window().maximize();  
  },  
  jasmineNodeOpts: {  
    showColors: true  
  }  
};
```




Using OnPrepare

```
exports.config = {  
  seleniumAddress: 'http://localhost:4444/wd/hub',  
  capabilities: {  
    'browserName': 'chrome'  
  },  
  onPrepare: function() {  
    global.mydriver = browser.driver;  
    mydriver.manage().window().maximize();  
  },  
  jasmineNodeOpts: {  
    showColors: true  
  }  
};
```



Using params

```
exports.config = {
  seleniumAddress: 'http://localhost:4444/wd/hub',
  capabilities: {
    'browserName': 'chrome'
  },
  // This can be changed via the command line as:
  // --params.login.user 'ngrocks'
  params: {
    login: {
      user: 'protractor-br',
      password: '#ng123#'
    }
  },
  jasmineNodeOpts: {
    showColors: true
  }
};
```



Using params

```
describe('login page', function() {  
  
    var params = browser.params;  
  
    it('should login successfully', function() {  
        element( by.model('username') ).sendKeys( params.login.user );  
        element( by.model('password') ).sendKeys( params.login.password );  
        element( by.css('[ng-click="login()"]') ).click();  
        expect( element(by.binding('username') ).getText() ).toEqual(  
params.login.user );  
    });  
  
});
```



References



- <http://www.protractortest.org/#/>
- <http://www.protractortest.org/#/api>
- <https://github.com/angular/protractor>
- <https://angularjs.org/>
- <https://jasmine.github.io/2.0/introduction.html>
- <https://github.com/angular/protractor/blob/master/docs/toc.md>
- <http://www.ng-newsletter.com/posts/practical-protractor.html>
- <https://www.youtube.com/watch?v=idb6hOxlyb8>





Thank
you!