

# Angular 2 End-to-End Testing

Sang Shin  
JPassion.com  
“Code with Passion!”



# Topics

- What is and Why End-to-End testing?
- End-to-End testing in Angular
- Steps for writing and running Protractor test
- Interacting with browser
- CSS selectors
- Interacting with elements
- Debugging End-to-End testing code

# **What is and Why End-to-End Testing?**

# What is and Why End-to-End Testing?

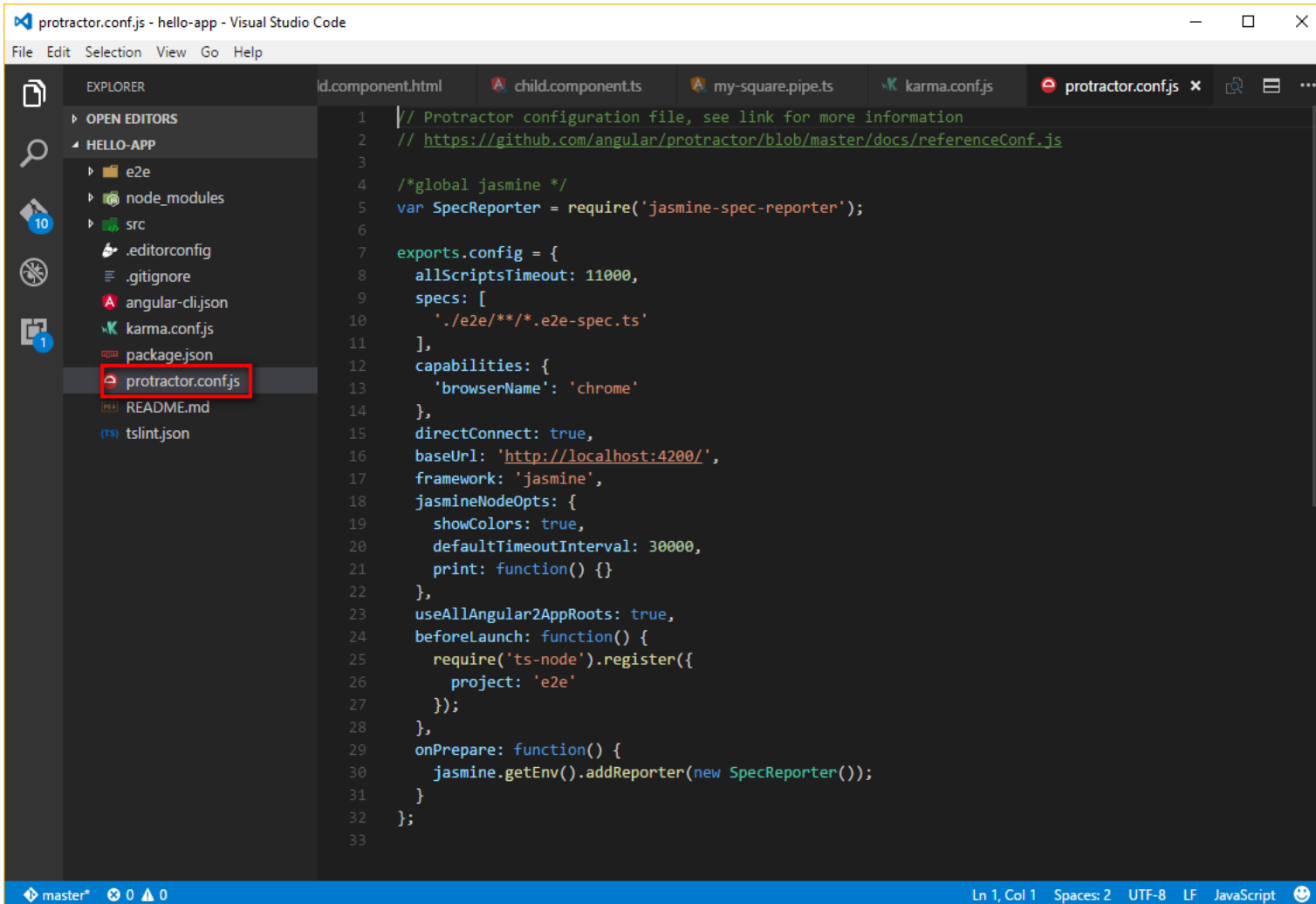
- Unit tests can't easily test templates, DOM manipulation or interoperability of components and services
- You can use Protractor, an end to end test runner which simulates user interactions for testing Angular apps

# **End-to-End Testing with Protractor**

# End-to-End Testing Tool: Protractor

- Used to control browsers and simulate user interactions
  - > Uses Selenium WebDriver under the covers
  - > You need JDK to run Selenium server
- You write Protractor test specs in Jasmine (like in Unit testing)
  - > describe(), it(), expect()
- Protractor installation and configuration
  - > It is a Node package
  - > For “angular-cli” generated Angular app, it is automatically installed and configured (with default settings)

# Protractor Configuration



The screenshot displays the Visual Studio Code interface with the 'protractor.conf.js' file open in the editor. The Explorer sidebar on the left shows the project structure, with 'protractor.conf.js' highlighted. The editor window shows the following code:

```
1 // Protractor configuration file, see link for more information
2 // https://github.com/angular/protractor/blob/master/docs/referenceConf.js
3
4 /*global jasmine */
5 var SpecReporter = require('jasmine-spec-reporter');
6
7 exports.config = {
8   allScriptsTimeout: 11000,
9   specs: [
10    './e2e/**/*.e2e-spec.ts'
11  ],
12  capabilities: {
13    'browserName': 'chrome'
14  },
15  directConnect: true,
16  baseUrl: 'http://localhost:4200/',
17  framework: 'jasmine',
18  jasmineNodeOpts: {
19    showColors: true,
20    defaultTimeoutInterval: 30000,
21    print: function() {}
22  },
23  useAllAngular2AppRoots: true,
24  beforeLaunch: function() {
25    require('ts-node').register({
26      project: 'e2e'
27    });
28  },
29  onPrepare: function() {
30    jasmine.getEnv().addReporter(new SpecReporter());
31  }
32 };
33
```

The status bar at the bottom indicates the file is on the 'master' branch, with 0 errors and 0 warnings. The editor settings show 'Ln 1, Col 1', 'Spaces: 2', 'UTF-8', 'LF', and 'JavaScript'.

## How to run it?

- Make sure the application is up and running
  - > Unlike Unit testing
- For “angular-cli” generated application
  - > `ng e2e`



# Writing E2E Testing

# Protractor Provided Global Variables

- Protractor defines various globals to help you interact with the browsers and its contents
  - > “browser” - represents the browser window
  - > “element” - allows you to find an element in the page
  - > “by” - lets you specify search criteria

# Interacting with Browser

# Using Browser

- Accessing a page
  - > `browser.get('http://www.angularjs.org');`
  - > `browser.get('index.html');`
- Getting absolute URL
  - > `browser.getLocationAbsUrl()`

# **Brief Tutorial on CSS Selectors**

# CSS Selectors

- CSS selectors are patterns used to select the element(s) you want to style
- CSS1 and CSS2 selector examples

.myclass Select all elements with class="myclass"

p.myclass Select all <p> elements with class="myclass"

#myid Select the element with id="myid"

p Select all <p> elements

div,p Select all <div> elements and all <p>

div p Select all <p> elements inside <div> elements

div>p Selects all <p> elements where the parent is a <div> element

div+p Selects all <p> elements immediately after <div> elements

# Selectors Added in CSS3

- CSS version 3 added many new selectors

|                              |   |
|------------------------------|---|
| <code>p~ul</code>            | Selects every <code>&lt;ul&gt;</code> element that are preceded by a <code>&lt;p&gt;</code> element           |
| <code>div.note~img</code>    | Select every <code>&lt;img&gt;</code> element that are preceded by <code>&lt;div&gt;</code> with “note” class |
| <code>a[src^="https"]</code> | Selects every <code>&lt;a&gt;</code> element whose src attribute value begins with "https"                    |
| <code>a[src\$=".pdf"]</code> | Selects every <code>&lt;a&gt;</code> element whose src attribute value ends with ".pdf"                       |
| <code>p:nth-child(2)</code>  | Selects every <code>&lt;p&gt;</code> element that is the second child of its parent                           |
| <code>p:last-child</code>    | Selects every <code>&lt;p&gt;</code> element that is the last element   |
| <code>input:enabled</code>   | Selects every enabled <code>&lt;input&gt;</code> element  |
| <code>input:disabled</code>  | Selects every disabled <code>&lt;input&gt;</code> element   |
| <code>input:checked</code>   | Selects every checked <code>&lt;input&gt;</code> element  |

# Interacting with Elements



# Locating Elements

- By id
  - > `var anElement = element(by.id("myId"));`
- By css
  - > `var anElement = element(by.css('#search'));`

# Simulating User Actions

- `element()` returns an `ElementFinder` object, which has various methods to simulate user interaction
- Send keys to an element, and clean an element
  - > `anElement.sendKeys("Hello world");`
  - > `anElement.clear();`
- Click on an element
  - > `anElement.click();`
- Get the text value for an element, and get an attribute value
  - > `anElement.getText();`
  - > `var value = anElement.getAttribute("href");`

# Example

```
describe('routing-order app', function () {  
  
  beforeEach(function () {  
    browser.get("index.html");  
  });  
  
  it('should display all orders', function () {  
    expect(element.all(by.css('item')).count()).toBe(4);  
  });  
  
  it('should display first order when first link is clicked', function () {  
    element.all(by.css('li a')).first().click();  
    expect(browser.getLocationAbsUrl()).toBe('/order/0');  
  });  
  
  it('should have Home link from an individual order page', function () {  
    element.all(by.css('li a')).first().click();  
    element(by.css('#homelink')).click();  
    expect(browser.getLocationAbsUrl()).toBe('/');  
  });  
});
```

# Example

```
it('should have Help link in the home page', function () {  
    expect(element(by.css('a[href$="help"]')).getText()).toBe("Help");  
});
```

```
it('should have Enter order number field in Help page', function () {  
    element(by.css('a[href$="help"]')).click();  
    expect(element(by.css('input[ng-model$="orderNumber"]'))).not.toBe(undefined);  
});
```

```
it('should display order 2 when 2 is entered in the help page', function () {  
    element(by.css('a[href$="help"]')).click();  
    element(by.css('input')).sendKeys('2');  
    element(by.css('button')).click();  
    expect(browser.getLocationAbsUrl()).toBe('/order/2');  
});
```

```
});
```

**Code with Passion!**  
**[JPassion.com](http://JPassion.com)**

