

2. Build up and create a simple test

- 1. Installation
- 2. Object Locators
- 3. A Simple Test
- 5. Source Codes

1. Installation

To set up protractor, we need to install below three things

- Node.js
- Protractor
- Selenium Webdriver

Use npm to install Protractor globally with:

```
npm install -g protractor
```

This will install two command line tools, `protractor` and `webdriver-manager`. Try running `protractor --version` to make sure it's working.

The `webdriver-manager` is a helper tool to easily get an instance of a Selenium Server running. Use it to download the necessary binaries with:

```
webdriver-manager update
```

Now start up a server with:

```
webdriver-manager start
```

This will start up a Selenium Server and will output a bunch of info logs. Your Protractor test will send requests to this server to control a local browser. You can see information about the status of the server at <http://localhost:4444/wd/hub>.

2. Object Locators

Apart from the usual element by xpath, by id, by name, by css etc, Protractor offers additional bindings like by model, by repeater, by buttontext, by bindings, by partialbuttontext.

☒ `element()` vs `element.all()`

```
element( by.binding('appName') );  
// clicks the 3rd element  
element.all( by.css('[ng-click="openPage()"]') ).get(2).click();
```

☒ `by.binding`

```
<span ng-bind="myModel"></span>  
<!-- or  
<span>{{myModel}}</span>element( by.binding('myModel') );
```

☒ `by.model`

```
<input ng-model="myModel" />
element( by.model('myModel') );
```

✓ by.repeater

```
<ul>
  <li ng-repeat="user in users">
    <span>{{user.name}}</span>
  </li>
</ul>
element( by.repeater('user in users').row(0).column('name') );
```

✓ by.css

```
<button ng-click="sendMail()">Send mail!</button>
element( by.css('[ng-click="sendMail()"]') );
```

✓ Executing events

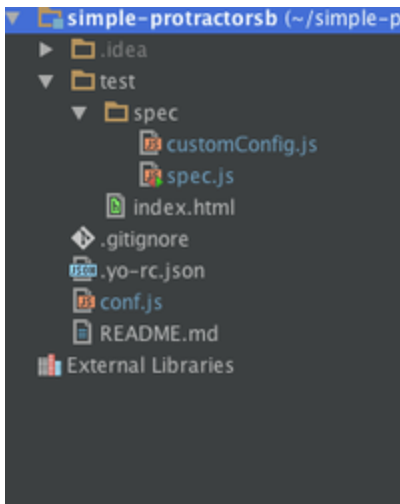
```
<textarea ng-model="commentText"><textarea>
element( by.model('commentText') ).sendKeys("Hi!",
protractor.Key.ENTER);
```

```
// Example of getText() promise
element( by.model('zipcode') ).getText()
  .then(function(val) {
    var num = val.substring(0, 4);
    var isNum = !isNaN(num);
    expect( isNum ).toBeTruthy();
  });
```

Please see the full list of available methods in protractor [Full list of available methods](#)

3. A Simple Test

project structure



Create conf.js : is the configuration file which controls the execution, where reports can be configured, environment details and browsers can be configured for a run. It looks something like this

```
exports.config = {  
  framework: 'jasmine',  
  seleniumAddress: 'http://localhost:4444/wd/hub',  
  specs: ['test/spec/customConfig.js', 'test/spec/spec.js']  
  ,allScriptsTimeout: 15000  
}
```

Create spec.js

```

// spec.js
describe('Protractor Demo App', function() {
    //register button in home page
    var registerBtn = element.all(by.linkText('Register')).get(0);
    var firstname = element(by.model('vm.user.firstName'));
    var lastname = element(by.model('vm.user.lastName'));
    var username = element(by.model('vm.user.username'));
    var password = element(by.model('vm.user.password'));
    //register button in register page
    var registerDetailBtn =
element.all(by.buttonText('Register')).get(0);
    //message is showed after navigating to the home page
    var message = element(by.binding('flash.message'));

    //data
    var i_firstName = 'Thuy';
    var i_lastName = 'Nguyen';
    var i_username = 'tixiuthu';
    var i_password = '123456';
    var i_successful_message = 'Registration successful';

    beforeEach(function() {
        browser.get('http://localhost/demo');
    });

    //register a user
    it('register a user', function() {
        //expectation: the system navigated to the Dashboard screen
        expect(browser.getTitle()).toEqual('Dashboard');
        //click register button
        registerBtn.click();
        //expectation: the system navigated to the Register screen
        expect( firstname.isDisplayed()).toEqual(true);
        firstname.sendKeys(i_firstName);
        lastname.sendKeys(i_lastName);
        username.sendKeys(i_username);
        password.sendKeys(i_password)
        registerDetailBtn.click().then(function () {
            expect(message.getText()).toBe(i_successful_message);
        });
    });
});
});

```

Reduce speed of protractor tests (Customise)

create customConfig.js

```
var origFn = browser.driver.controlFlow().execute;

browser.driver.controlFlow().execute = function () {
  var args = arguments;

  origFn.call(browser.driver.controlFlow(), function () {
    //increase or reduce time value, its in millisecond
    return protractor.promise.delayed(400);
  });

  return origFn.apply(browser.driver.controlFlow(), args);
};
```

Simple way to run

- Start up a server with: ***sudo webdriver-manager start*** (Sometimes I must run ***sudo webdriver-manager update*** firstly, after using the start server command)
- cd to the project folder
- run protractor conf.js

5. Source Codes



fusion-protractorsb.zip



simple-protractor.zip