

Providing Protractor to Java / .Net

Serguei Kouzmine
kouzmine_serguei@yahoo.com



Angular Protractor in Javascript

Many of technologies considered part of Protractor Angular application "end to end" integration testing framework have same or better counterparts in Java and .Net stack outside of Selenium:

- mocha
- jasmine
- webdriverJs
- cucumber
- Junit / testNg
- Mockito
- Selenium / selenium Grid
- cucumber

From Selenium perspective, two outstanding features of Protractor are Angular semantics integration with locator and delegating the wait during element detection to Angular runtime.



Bare-bones Angular page

```
<html ng-app="myApp">
<head>
<script src="angular.js"></script>
<script type="text/javascript">
var app = angular.module('myApp', []);
app.controller('myCtrl',
  function($scope){
    $scope.myChoice = 1;
    $scope.options = [
      {name: 'one', value: 1},
      {name: 'two', value: 2},
    ]
  });
</script>
</head>
```

```
<body ng-controller="myCtrl">
  <select ng-model="myChoice">
    <option ng-repeat="option in
options" value="{{option.value}}"
ng-selected=
  "option.value == myChoice"
>{{option.name}}</option>
  </select>
  <div>Value = {{myChoice}}</div>
</body>
</html>
```

Protractor 'client side' Javascript snippets injected in the browser enable test operate same language as page developer by offering element finder methods "by" binding, model, repeater etc. It also provides Angular agnostic methods like buttonText, cssContainingText

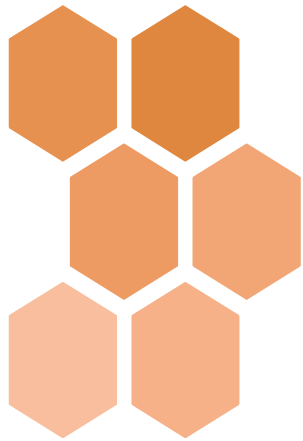


Integration with QA / CI

- Framework (module) layers offered for integrating Selenium Testing with CI by Java, .Net and Node.js offer similar functionality but are not easy to combine (examples exist)
- Angular MVC Clientside API were only available through Protractor



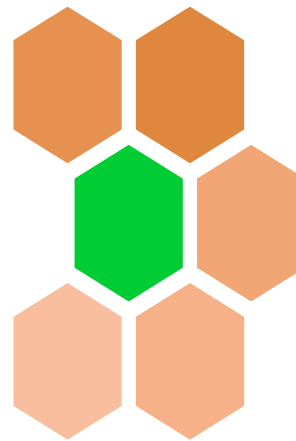
Adding Key Protractor Methods



Java



Node.js



jProractor,
Protractor-
net



WebDriver Wire Protocol

Command	Method	URI Template
Find Element	POST	/session/{ <i>session id</i> }/element
Find Element from Another	POST	/session/{ <i>session id</i> }/element/{ <i>element id</i> }/element
Get Element Attribute	GET	/session/{ <i>session id</i> }/element/{ <i>element id</i> }/attribute/{ <i>name</i> }
Get Element Property	GET	/session/{ <i>session id</i> }/element/{ <i>element id</i> }/property/{ <i>name</i> }
Get Element CSS Value	GET	/session/{ <i>session id</i> }/element/{ <i>element id</i> }/css/{ <i>property name</i> }
Get Element Text	GET	/session/{ <i>session id</i> }/element/{ <i>element id</i> }/text
Get Element Tag Name	GET	/session/{ <i>session id</i> }/element/{ <i>element id</i> }/name
Get Element Rectangle position	GET	/session/{ <i>session id</i> }/element/{ <i>element id</i> }/rect
Execute Script	POST	/session/{ <i>session id</i> }/execute/sync
Execute Async script	POST	/session/{ <i>session id</i> }/execute/async

<https://w3c.github.io/webdriver/webdriver-spec.html#dfn-command>

Protract methods

Method
model
binding
options
selectedOption, selectedRepeaterOption
repeater
RepeaterRow, repeaterColumn, repeaterElement
buttonText, partialButtonText
cssContainingText
testForAngular, waitForAngular
evaluate



WebDriver with Protractor API

- ByClassName, ByCssSelector, ById, ByLinkText, ByName, ByPartialLinkText, ByTagName, ByXPath (nested classes of By)
- NgBy (JavaScriptBy): binding, buttonText, cssContainingText, model, options, repeater, repeaterRows, repeaterElement, selectedOption, selectedRepeaterOption
- ExpectedConditions
- WebElement.findElements
- WebElement.evaluate
- Wait
- WebDriver.waitForAngular
- SendKeys
- Actions



Anotating the code

Core Selenim 'By' classes support *a generic* DOM node therefore need annotated :

```
// finding selected customer
WebElement element = driver.findElement(By.xpath(
    "/div[13]/ul/li[4]/select"));

WebElement element = driver.findElement(By.cssSelector(
    "div#outer div.inner"));
```

Javascript frequently used for visual cue:

```
// highlight
(JavascriptExecutor) driver.executeScript(
    "arguments[0].style.border='3px solid yellow'", element);
```

And can be equally used as well to return element information:

```
Object result = (JavascriptExecutor)driver.executeScript("return
    angular.identity(angular.element(arguments[0])).html();",
    element);
```

<https://docs.angularjs.org/api/ng/function/angular.identity>



Convenience of Protractor Locators

Tests become more readable once locators reflect the context

```
NgWebElement cell = ngDriver.findElement(  
    NgBy.repeaterElement("row in rows", rowNum, "cellName"));
```

```
ngDriver.findElement(ByAngular.repeater("car in Cars")  
    .row(2)  
    .column("item.isSelected")).click();
```

```
List<WebElement> customers = ngDriver.findElements(NgBy.repeater("cust in  
Customers"));
```

```
NgWebElement selectedCustomer =  
ngDriver.findElement(NgBy.model("selectedCustomer"));
```

```
WebElement customerName =  
selectedCustomer.findElement(NgBy.binding("Name"));
```

```
Object customerCountry = selectedCustomer.evaluate("customer.Country");
```

```
ArrayList<Long> accounts = (ArrayList<Long>)  
    new NgWebElement(ngDriver, selectedCustomer).evaluate(  
"customer.accountNo");
```



Protractor and Cucumber

@test **Scenario Outline:** Existing Customer Login

Given I am at the Home Page

When I continue as "Customer Login"

And I am existing customer "<FirstName>", "<LastName>"

Then I am greeted by "<FirstName>", "<LastName>"

And I see balance on one of my accounts
"<AccountNumbers>"

And I can switch to any of accounts "<AccountNumbers>"
I have

And I can not see any other accounts...

Examples:

AccountNumbers	FirstName	LastName
1004,1005,1006	Harry	Potter



Step Definitions

```
@Test
public void testCustomerLogin() throws
Exception {
    // When I proceed to "Customer Login"
    ngDriver.findElement(
    NgBy.buttonText("Customer Login")).click();

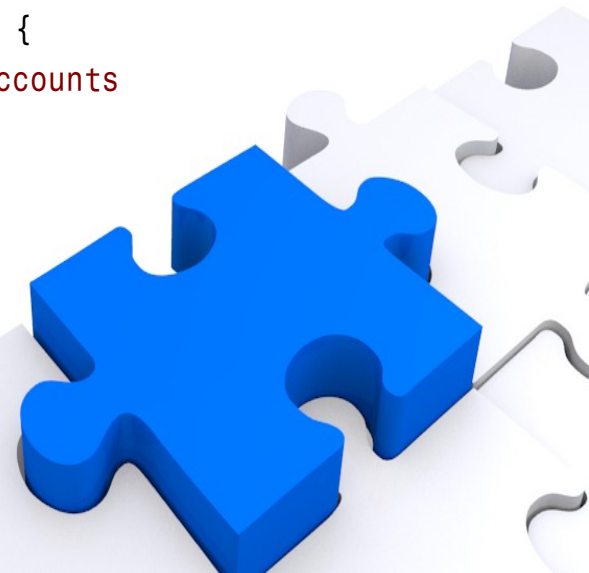
    // And I choose from existing customers
    for (WebElement currentCustomer :
    ngDriver.findElements(
    NgBy.repeater("cust in Customers")))) {
        if (currentCustomer.getText().indexOf(
        "Harry Potter") >= 0) {
            currentCustomer.click();}
    }

    // And I log in
    ngDriver.findElement(NgBy.buttonText("Login")
    ).click();

    // Then I am greeted by my first and last
    name
    assertThat(ngDriver.findElement(NgBy.binding(
    "user")).getText(), containsString("Harry"));
```

```
// And I see one of my accounts
Matcher matcher = Pattern.compile(
    "(1004|1005|1006)")
    .matcher(
    ngDriver.findElement(NgBy.binding("account
    No")).getText());
    assertTrue(matcher.find());

    Enumeration<WebElement> accounts =
    Collections.enumeration(ngDriver
    .findElements(NgBy.options(
    "account for account in Accounts")));
    while (accounts.hasMoreElements()) {
        // And I can not see any other accounts
        then my own
        matcher = Pattern.compile(
        "^(?!1004|1005|1006).*$")
        .matcher(
        accounts.nextElement().getText());
        assertFalse(matcher.find());
    }
```



Protractor Annotations

```
NgWebElement firstNumber =  
ngDriver.findElement(NgBy.model("first"));
```

```
NgWebElement operator=  
ngDriver.findElement(NgBy.options(  
"value for (key, value) in operators"));
```

```
@FindBy(how = How.MODEL, using = "first")  
private WebElement firstNumber;  
@FindBy(how = How.OPTIONS, using =  
"value for (key, value) in operators")  
private WebElement operatorSelect;  
new  
Select(operatorSelect).selectByValue("MULTIPLICATION");
```



Clientsidescripts.js

18 methods

3 to > 100 lines of JavaScript / method

```
var element = arguments[0];  
var expression = arguments[1];  
return angular.element(element).scope().$eval(expression);
```

<https://github.com/angular/protractor/blob/master/lib/clientsidescripts.js#L623>



Clientsidescripts.js

```
var findBindings = function(binding, exactMatch, using, rootSelector) {
  var root = document.querySelector(rootSelector || 'body');
  using = using || document;
  var bindings = using.getElementsByClassName('ng-binding');
  var matches = [];
  for (var i = 0; i < bindings.length; ++i) {
    var dataBinding = angular.element(bindings[i]).data('$binding');
    var bindingName = dataBinding.exp || dataBinding[0].exp || dataBinding;
    var matcher = new RegExp('{{|\\s|^|\\|}}' +
      binding.replace(/[\-\\[\]\/\{\}\(\)\*\+\?\.\\\^\$\\]/g,
        '\\$&') +
      '({|\\s|\\$|\\|})');
    if (matcher.test(bindingName)) {
      matches.push(bindings[i]);
    }
  }
  return matches; /* Return the array for webdriver.findElements. */
};

var using = arguments[0] || document;
var binding = arguments[1];
var rootSelector = arguments[2];
return findBindings(binding, using, rootSelector);
```

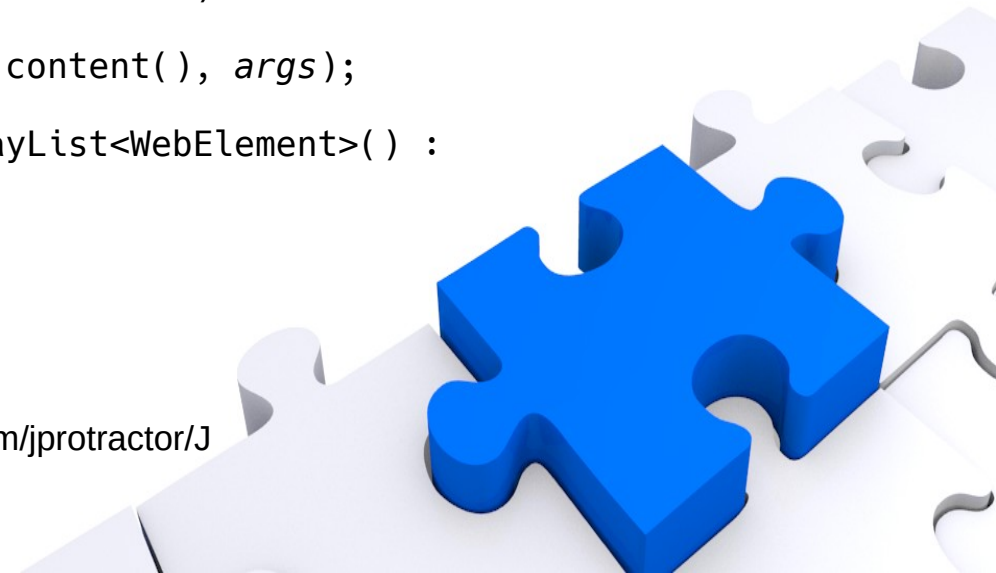


Java Plumbing

```
public final class JavaScriptBy extends By {
    public List<WebElement> findElements(final SearchContext
context) {
        final Object[] scriptargs = new Object[this.args.length + 1]
        scriptargs[0] = this.root;
        System.arraycopy(this.args, 0, scriptargs, 1,
this.args.length);
        return this.elements(context, scriptargs);
    }

    private List<WebElement> elements( final SearchContext context,
final Object[] args) {
        List<WebElement> elements = (List<WebElement>)
this.executor(context)
        .executeScript(this.script.content(), args);

        return (elements == null) ? new ArrayList<WebElement>() :
elements;
    }
}
```



Java Plumbing , contd.

```
public static By model(final String model) {  
    return new JavaScriptBy(new FindModel(), model)  
}  
  
public final class FindModel implements Script {  
    public String content() {  
        return new Loader("model").content();  
    }  
}  
  
final class Loader {  
    Loader(final String file) {  
        this.filename = file + ".js";  
    }  
  
    String content() {  
        final InputStream stream = Loader.class.getClassLoader()  
            .getResourceAsStream(this.filename);  
        final byte[] bytes = new byte[stream.available()];  
        Stream.read(bytes);  
        return new String(bytes, "UTF-8");  
    }  
}
```



Questions, Demos



Resources

- <https://github.com/angular/protractor>
- <https://github.com/caarlos0/jProtractor>
- <https://github.com/F1tZ81/Protractor-jvm>
- <https://github.com/henrrich/jpagefactory>
- <https://github.com/paul-hammant/ngWebDriver>
- <https://github.com/bbaia/protractor-net>
- <https://github.com/sergueik/protractor-net>
- <https://github.com/greengerong/maven-ng-protractor>



Resources

- <http://www.java2s.com/Tutorials/AngularJS/>
- <http://juliemr.github.io/protractor-demo/>
- <http://qualityshepherd.com/angular/friends/>
- <http://www.way2automation.com/angularjs-protractor/banking>
- <https://htmlpreview.github.io/?https://github.com/angular-ui/>
- <http://amitava82.github.io/angular-multiselect/>
- <http://www.codeproject.com/Articles/1066968/Developing-Protractor-tests-in-Csharp-or-Java>

