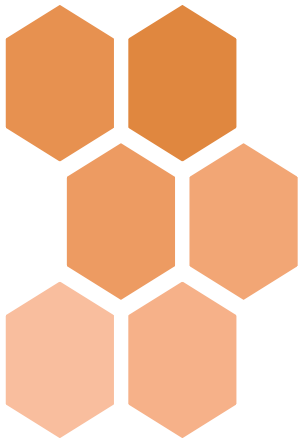
Providing Protractor to Java / .Net

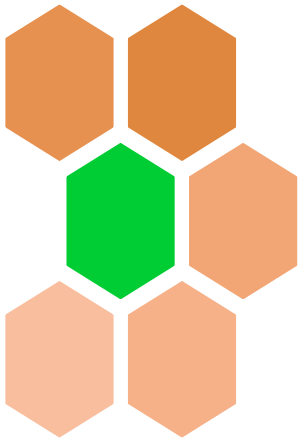
Serguei Kouzmine
kouzmine_serguei@yahoo.com
http://www.linkedin.com/in/skouzmine

Adding Key Client-side Protractor Methods to Java or .Net



Java

Node.js



jProractor,
Protractor-
net

Framework-generated code

```
<div id="mega-topbar"
  class=
  "Pos(r) H(22px) mini-header_Mt(-19px) Reader-open_Mt(-19px)
Bxsh($topbarShadow) Z(7) Bg($topbarBgc)"
  data-ylk="rspns:nav;t1:a1;t2:hd;t3:tb;sec:hd;itc:0;elm:itm;elmt:pty;">
  <ul class="Pos(r) Miw(1000px) Pstart(9px) Lh(1.7) Reader-open_Op(0)
mini-header_Op(0)" role="navigation">
    <li class="D(ib) Mstart(22px) Mend(8px)">
      <a class="C(#fff)" href="#" data-rapid_p="16">
          Make Yahoo Your Homepage<span> »</span>
      </a>
    </li>
  </ul>
</div>
```

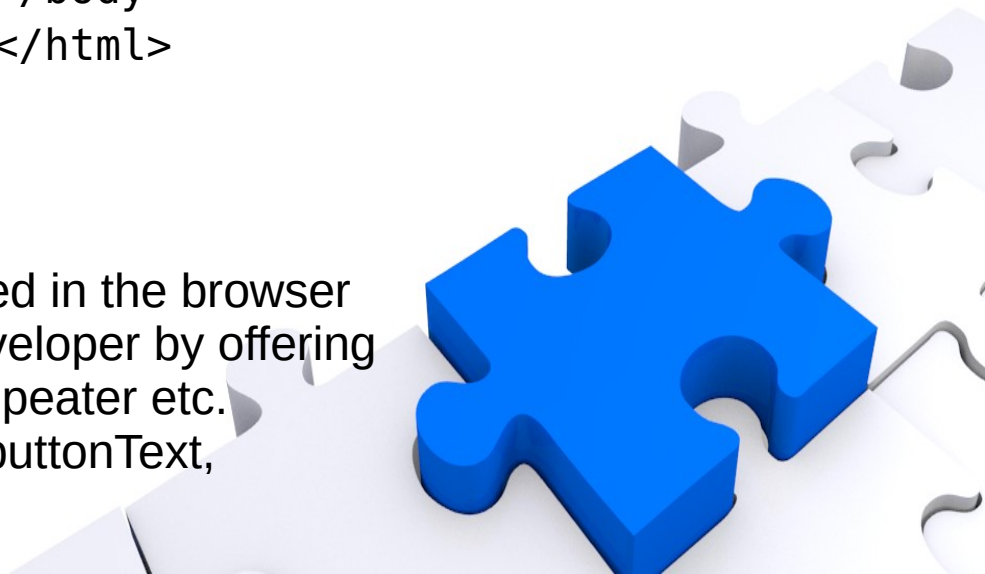
```
 <ul ng-repeat="country in region.countries" class="ng-scope">
   <li> <h3 class="country ng-binding">United States</h3>
       <ul class="country-list">
         <li ng-repeat="airport in country.airports track by $index"
            class="popupLinks city  jetblue">
           <a href="javascript:void(0)" ng-keydown="keydown($event, airport)"
              ng-click="select(airport)"
              class="ng-binding">Fort Lauderdale, FL (FLL)
           </a>
```

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



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.

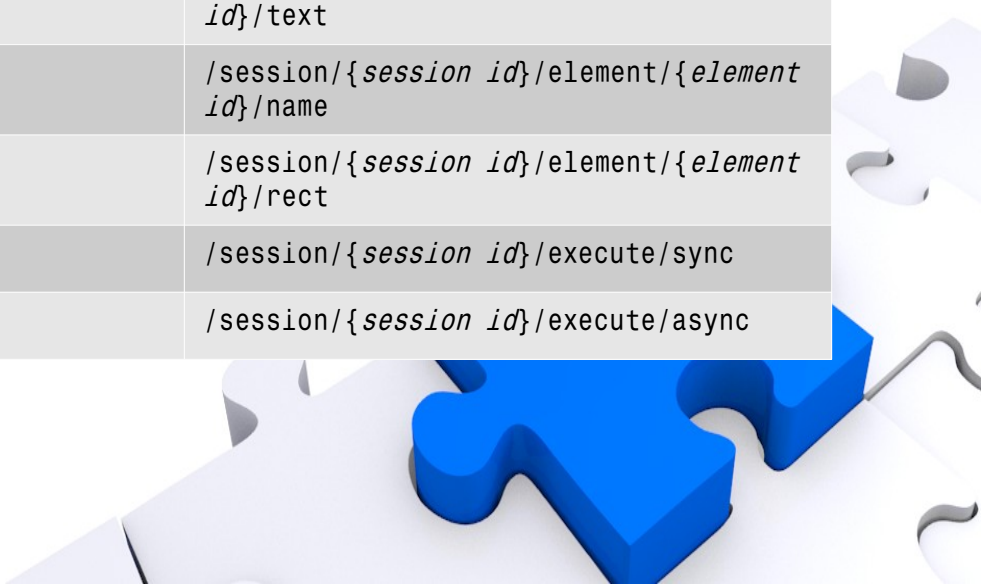
Projects Timeline

- angular/protractor Jan 2013, 247 contibutors
- bbaia/protractor-net Oct 2013, 10 contributors
- paul-hammant/ngWebDriver Sep 2013, 6 contributors
- greengerong/maven-ng-protractor Mar 2014, 1 contributor
- caarlos0/jProtractor Jul 2014, 2 contributors
- kpodl/pytractor Aug 2014, 2 contributors
- GRITAG/Protractor-jvm Jun 2015, 2 contributors
- henrrich/jpagefactory May 2016, 2 contributors

WebDriver Wire Protocol

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

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



WebDriver Implementation

```
public class FirefoxDriver extends RemoteWebDriver {



}
```

https://github.com/SeleniumHQ/selenium/blob/master/java/client/src/org/openqa/selenium/firefox/FirefoxDriver.java

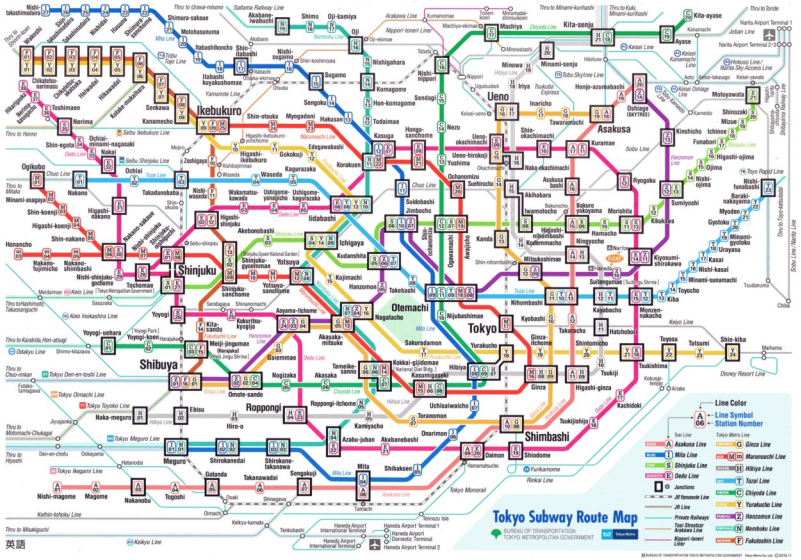
```
public class RemoteWebDriver implements
    WebDriver, JavascriptExecutor,
    FindsById, FindsByClassName,
    FindsByLinkText, FindsByName,
    FindsByCssSelector, FindsByTagName,
    FindsByXPath, HasInputDevices,
    HasCapabilities, Interactive,
    TakesScreenshot {

}
```

https://github.com/SeleniumHQ/selenium/blob/master/java/client/src/org/openqa/selenium/remote/RemoteWebDriver.java

Russian Nesting Doll Architecture

Java program launches Selenium to be the carrier of Javascript snippet from Protractor library that commands Anguar to find and process Page Elements

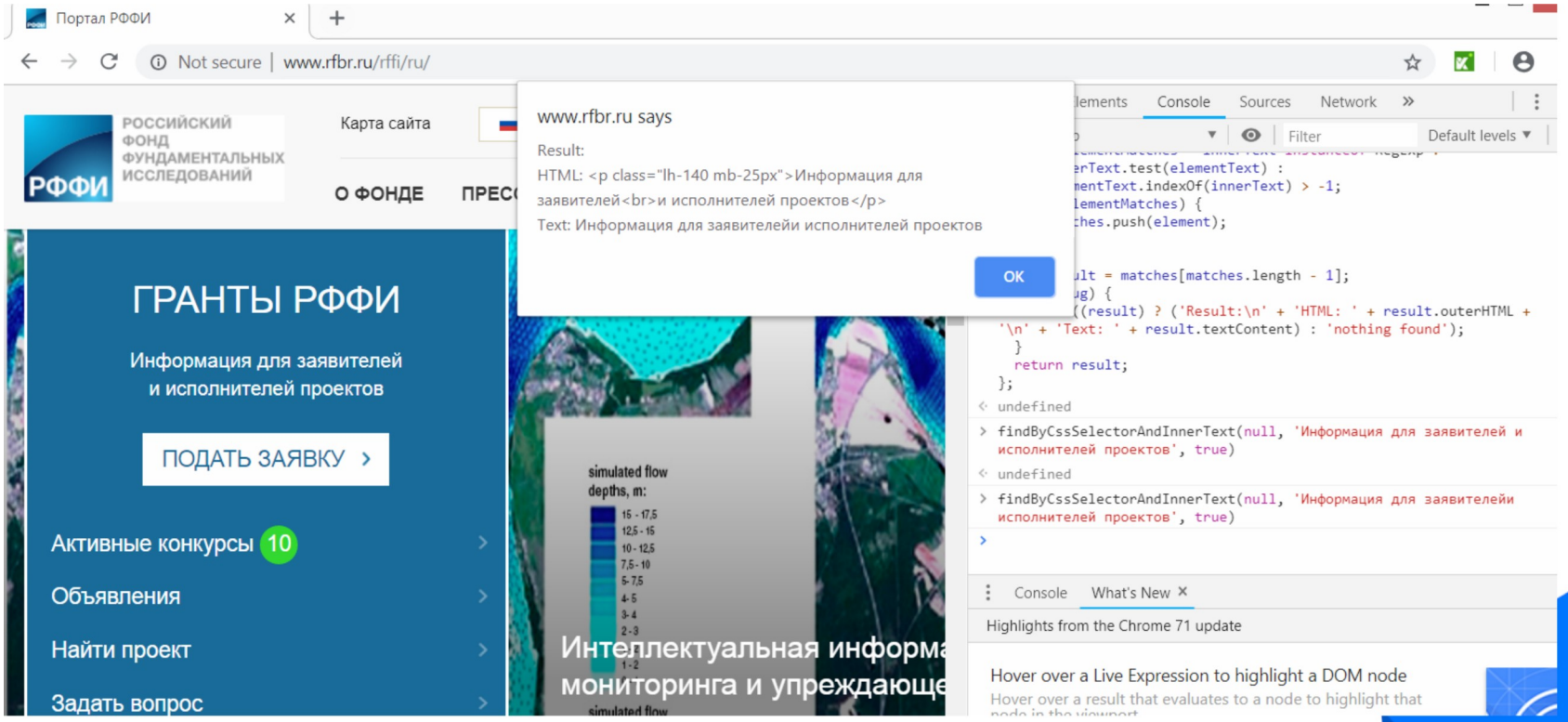
Internal Design

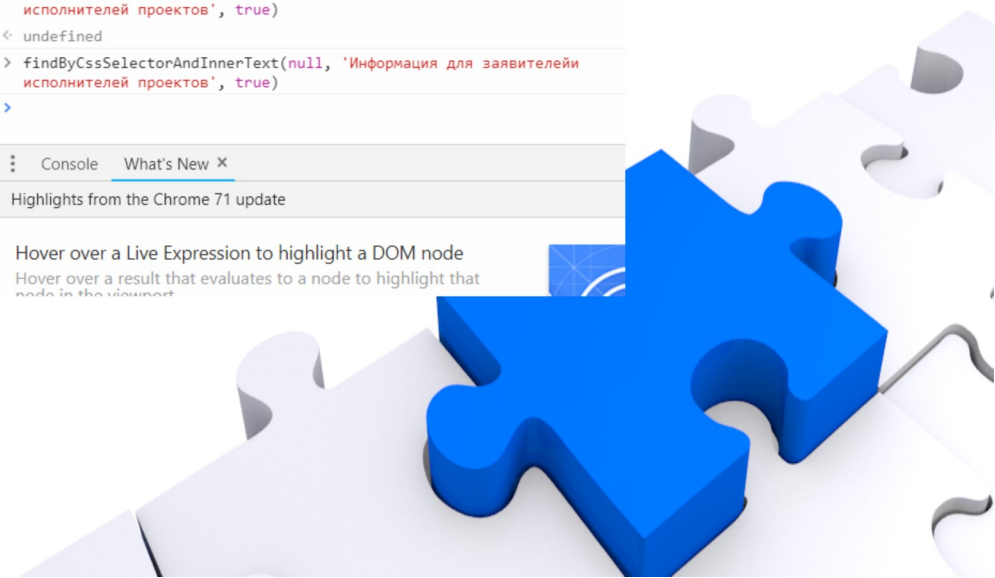


Swisswatch complexity vs. Tokyo Subway map complexity

Non-Protractor example

Locate element by the text content with JavaScript, in the Browser developer console.





Non-Protractor example, contd.

JavaScript snippet

```
var findByCssSelectorAndInnerText = function(cssSelector, textToSearch) {
  var elements = document.querySelectorAll(cssSelector); var matches = [];
  for (var i = 0; i < elements.length; ++i) {
    var element = elements[i];
    var elementText = element.textContent || element.innerText ||
        element.getAttribute('placeholder') || '';
    if (elementText.replace(/\n/, ' ').indexOf(textToSearch) > -1) {
      matches.push(element);
    }
  }
  return matches[matches.length - 1];
}
```

Selenium `JavascriptExecutor` class responsible for passing argument `Object[]` array into the browser and casting the response into `String`, `RemoteWebElement` as appropriate

```
WebElement element = JavascriptExecutor.class.cast(driver)
                    .executeScript(script,
                                   arguments);
```

Non-Protractor example, contd.

Legacy way, through XPath condition of DOM node selector

```
String xpath = String.format(
  ”//*[contains(normalize-space(translate(text(), '\\t\\n\\r','')),'%s')]”,
  textToSearch);
WebElement element = driver.findElement(By.xpath(xpath));
```

Protractor methods

Method
model
binding
options
selectedOption, selectedRepeaterOption
repeater
repeaterRow, repeaterColumn, repeaterElement
buttonText, partialButtonText
cssContiningText
testForAngular, waitForAngular
evaluate

http://www.protractortest.org/#/api

WebDriver extended with Protractor

- 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

`

Selenium JavascriptExecutor usage

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"));
```

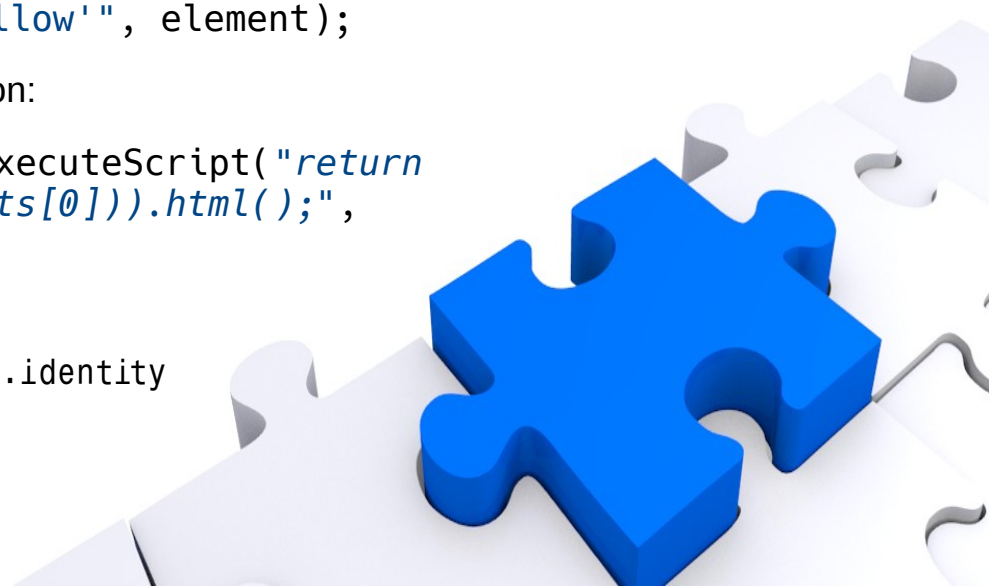
Javascript frequently used for visual cue:

```
// 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



Selenium Wait ExpectedConditions

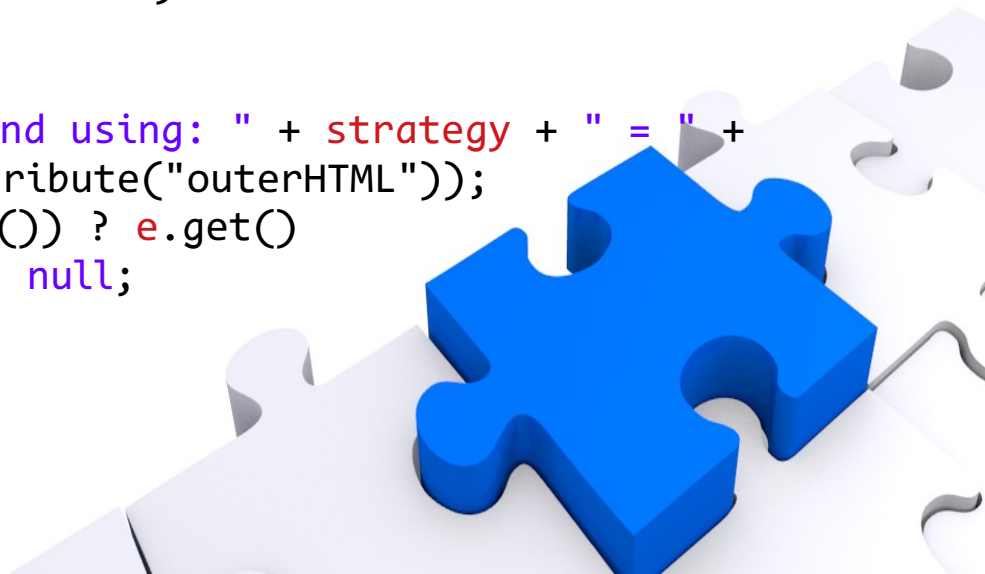
- alertIsPresent :
- attributeContains
- attributeToBe :
- attributeToBeNotEmpty :
- elementSelectionStateToBe :
- elementToBeClickable :
- elementToBeSelected :
- frameToBeAvailableAndSwitchToIt :
- invisibilityOf :
- invisibilityOfAllElements :
- javaScriptThrowsNoExceptions :
- jsReturnsValue :
- and :
- or :
- not :
- numberOfElementsToBe :
- numberOfElementsToBeLessThan :
- numberOfElementsToBeMoreThan :
- numberOfWindowsToBe :

https://seleniumhq.github.io/selenium/docs/api/java/

- presenceOfAllElementsLocatedBy :
- presenceOfElementLocated :
- presenceOfNestedElementLocatedBy :
- presenceOfNestedElementsLocatedBy :
- refreshed :
- stalenessOf :
- textMatches :
- textToBe :
- textToBePresentInElement :
- textToBePresentInElementLocated :
- textToBePresentInElementValue :
- titleContains :
- titleIs :
- urlContains :
- urlMatches :
- urlToBe :
- visibilityOf :
- visibilityOfAllElements :
- visibilityOfAllElementsLocatedBy :
- visibilityOfElementLocated :
- visibilityOfNestedElementsLocatedBy :

Finding elements with explicit wait

```
 switch (strategy) {
   case "css":
     locator = By.cssSelector(selectorValue);
     break;
 };

element = (new WebDriverWait(driver, timeout)).until(new
ExpectedCondition<WebElement>() {
     @Override
     public WebElement apply(WebDriver d) {
       Optional<WebElement> e = d.findElements(locator)
                                 .stream()
                                 .findFirst();
    if (e.isPresent()) { System.err.println("find using: " + strategy + " = " +
       selectorValue + "  => " + e.get().getAttribute("outerHTML"));
   return (e.isPresent() && e.get().isDisplayed()) ? e.get()
                                 : (WebElement) null;
  });
```



Protractor Wait

```
var el = document.querySelector(arguments[0]);
var callback = arguments[1];
angular.element(el).injector().get('$browser').notifyWhenNoOutstandingRequests(call
back);
```

** in current version of Protractor – approx 100 lines*

```
public bool Displayed {
  get {
    this.ngDriver.WaitForAngular();
    return this.element.Displayed;
  }
}
```

https://github.com/angular/protractor/blob/master/lib/clientsidescripts.js

Convenience of Protractor Locators

Choosing the same language the page and the Test makes test more readable: 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");
```

Sample Scenario

```
// Given I am at the Home Page And I proceed to customer login
ngDriver.findElement( NgBy.buttonText("Customer Login"))
.click();

// I choose myself among existing customers
WebElement custId = ngDriver.findElement(NgBy.input("custId"));
custId.click();

WebElement customer = custId.findElements(
NgBy.repeater("cust in Customers")).stream().filter(cust ->
cust.getText()
.containsString("Hermoine Granger"))
.findFirst().get();

customer.sendKeys(Keys.SPACE); customer.click();
ngDriver.waitForAngular();
```

Sample Scenario, contd.

```
// And I log in
login = ngDriver.findElement( NgBy.buttonText("Login"));
assertTrue(login.isEnabled());
highlight(login);
login.click();

// Then I am greeted by my name
assertThat("greeting", ngDriver.findElement( NgBy.binding("user")).getText(),
containsString("Hermoine Granger"));

// And I see balance on one of my accounts

NgWebElement accountNo = ngDriver.findElement(NgBy.binding("accountNo"));

assertThat("account number", accountNo.getText(),
anyOf(is("1001"), is("1002"), is("1003")));
highlight(accountNo);
```

Sample Scenario, contd.

```
// And I open account

ngDriver.findElements(NgBy.options("account for account in Accounts"))
.stream().filter(account -> account.getText().matches("1001"))
.collect(Collectors.toList()).get(0).click();

// And I click “Transactions” button

NgWebElement transactionsButton = ngDriver
.findElement(NgBy.partialButtonText("Transactions"));
highlight(transactionsButton);
transactionsButton.click();


// And I inspect transactions
wait.until(ExpectedConditions.visibilityOf(
ngDriver.findElement(
NgBy.repeater("tx in transactions")).getWrappedElement()));
```

Sample Scenario, contd.

```
// Then for every transaction

for (WebElement tx : ngDriver.findElements(
  NgBy.repeater("tx in transactions"))) {

  NgWebElement ngTx = new NgWebElement(ngDriver, tx);

  // it would be either Debit or Credit type
  assertThat("transaction type", ngTx.evaluate("tx.type").toString(),
  anyOf(containsString("Debit"), containsString("Credit")));

  // it will have non zero amount
  assertThat("it will have non zero amount",
  Integer.parseInt(ngTx.evaluate("tx.amount").toString()),
  greaterThan(0));

   // it date should be the past
   assertThat("transaction date",
   dateFormat.parse(ngTx.evaluate("tx.date").toString()),
   sameOrBefore(new Date()));

}
```

Sample Scenario, contd.

```
// perform some Credit transaction-specific
checksngDriver.findElements(NgBy.repeaterColumn("tx in transactions",
"tx.type"))
.stream()
.filter(txType -> txType.getText().equalsIgnoreCase("Credit"))
.forEach(txType -> highlight(txType));

// perform some set check over all acounts of certain customer
List<Long> accounts = (List<Long>) new NgWebElement(ngDriver,
  ngDriver.findElement(
    NgBy.repeaterColumn("cust in Customers", "cust.fName"))
  ).evaluate("cust.accountNo");

System.err.println(accounts == null ? "No accounts"
  : "Accounts: " + Arrays.toString(accounts.toArray()));
```

Sample Scenario, contd.

```
<tr id="anchor1" ng-repeat="tx in transactions |
  orderBy:sortType:sortReverse | sDate:startDate:end" class="ng-scope"/>
<tr id="anchor2" ng-repeat="tx in transactions |
      orderBy:sortType:sortReverse | sDate:startDate:end"
    class="ng-scope">
  <td class="ng-binding">Jan 1, 2015 12:00:00 AM</td>
  <td class="ng-binding">4</td>
  <td class="ng-binding">Debit</td>
</tr>
```

```
<table>
  <tbody>
  <tr id="anchor{{$index}}" ng-repeat="tx in transactions |
      orderBy:sortType:sortReverse | sDate:startDate:end">
    <td>{{ tx.date | date:'medium'}}</td>
    <td>{{ tx.amount }}</td>
    <td>{{ tx.type }}</td>
  </tr>
  </tbody>

</table>
```

view-source:http://www.way2automation.com/angularjs-protractor/banking/listTx.html

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 choose my "<FirstName>", "<LastName>" among
existing customers
  And I log in
  Then I am greeted by "<FirstName>", "<LastName>"
  And I can switch to any of my accounts "<AccountNumbers>"
  And I see balance
  And I can not see any other accounts...
  Examples:
  | AccountNumbers | FirstName | LastName |
  | 1001,1002,1003 | Harry | Potter |
```

https://github.com/sergueik/jProtractor/blob/master/src/test/java/com/jprotractor/integration/NgWay2AutomationIntegrationTest.java

Protractor and Cucumber cont.

```
// for scenario: Existing Customer Login
@When("^I am choose my \"([^\"]*)\, \"([^\"]*)\"$ among existing
customers$")
public void loginWithFirstAndLastName(String fName, String lName) {
  WebElement customer = custId.findElements(
  NgBy.repeater("cust in Customers"))
      .stream()
      .filter(cust -> cust.getText()
      .containsString(String.format("%s %s", fName, lName)))
      .findFirst().get();

  customer.sendKeys(Keys.SPACE);
  customer.click();
  ngDriver.waitForAngular();
}
```

Protractor Annotations

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

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


@FindBy(how = How.MODEL, using = "first")

private WebElement element;
@FindBy(how = How.OPTIONS, using = "value for (key, value) in
operators")
```

https://github.com/henrrich/jpagefactory

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);
```

https://github.com/angular/protractor/blob/master/lib/clientsidescripts.js#L115

Questions, Demos

Resources

- https://github.com/angular/protractor
- https://giithub.com/sergueik/jProtractor
- https://github.com/henrrich/jpagefactory
- https://github.com/paul-hammant/ngWebDriver
- https://github.com/bbaia/protractor-net

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