

# Providing Protractor to Java / .Net

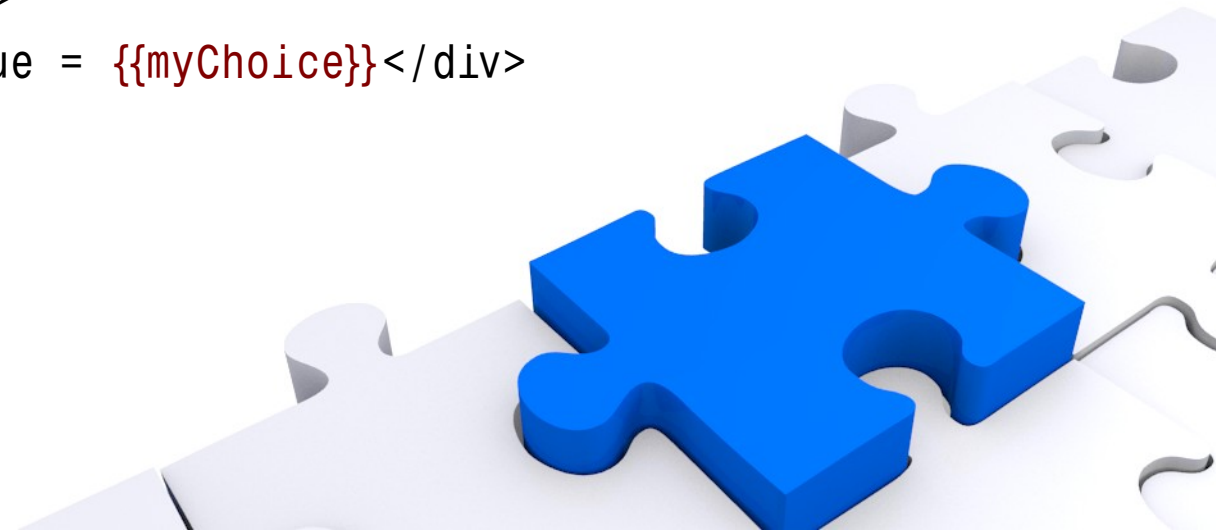
Serguei Kouzmine  
kouzmine\_serguei@yahoo.com



# Bare-bones Angular page

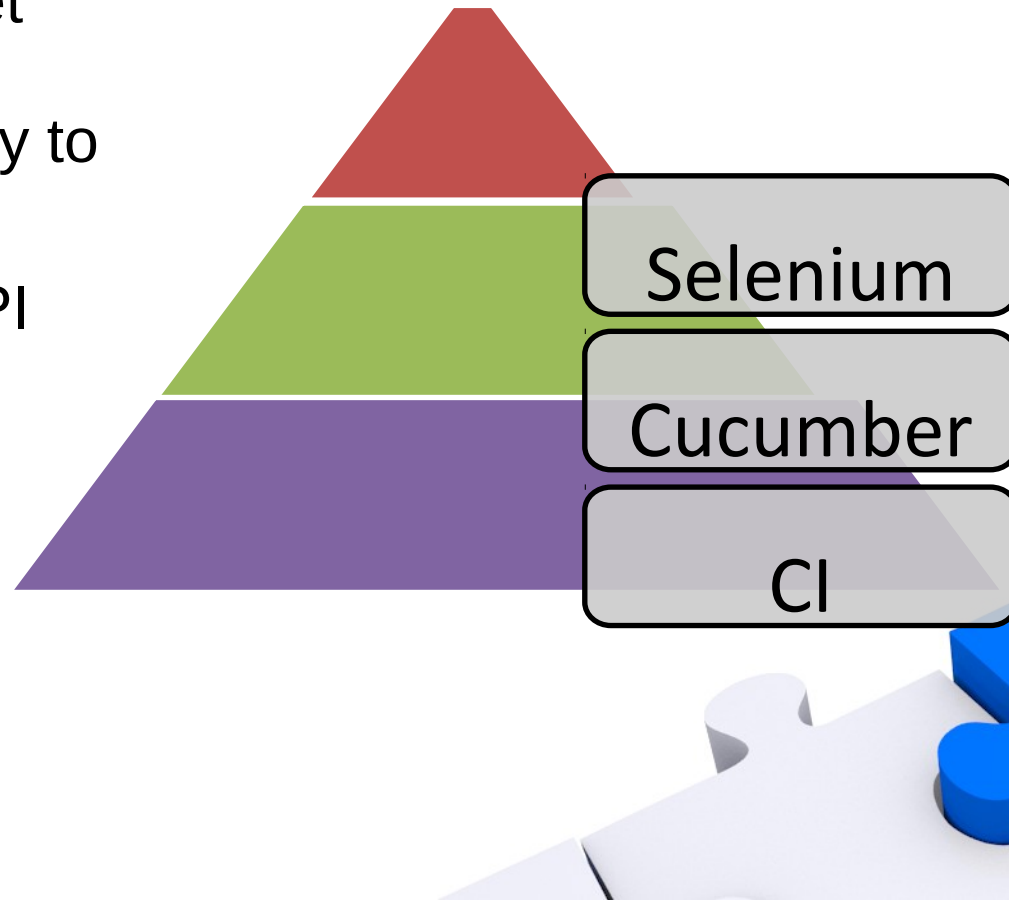
```
<html ng-app="myApp">
<head>
<script src="angular.js"></script>
<script type="text/javascript">
var app = angular.module('myApp', []);
app.controller('myCtrl',
function($scope){
$scope.myChoice = 1;
  $scope.options = [
{ name: 'one', value: 1},
{ name: 'two', value: 2},
  ]});
</script>
</head>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

```
<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>
```

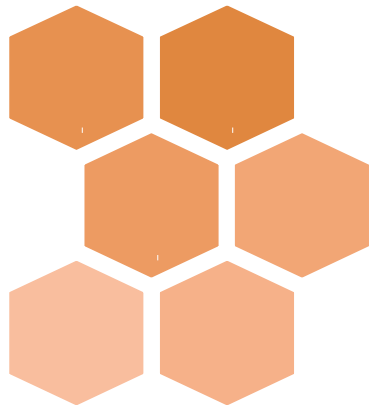


# Integration with QA / CI

- 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



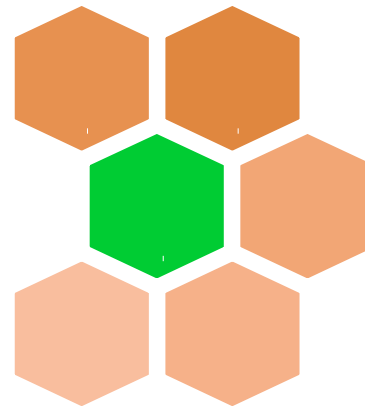
# Adding Key Protractor Methods



Java



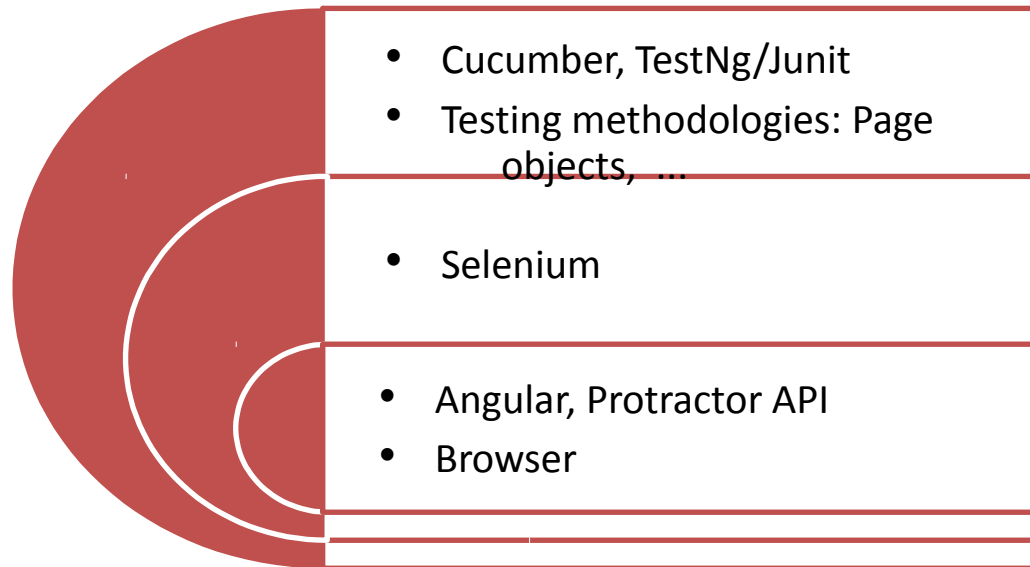
Node.js



jProtractor,  
Protractor-net



# Tool Chain



# 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>



# Frequently Used Selenium API

- ByClassName, ByCssSelector, ById, ByLinkText, ByName, ByPartialLinkText, ByTagName, ByXPath (nested classes of By)
- ExpectedConditions
- WebElement.findElements
- Wait
- SendKeys
- Actions



# WebDriver with Protractor API

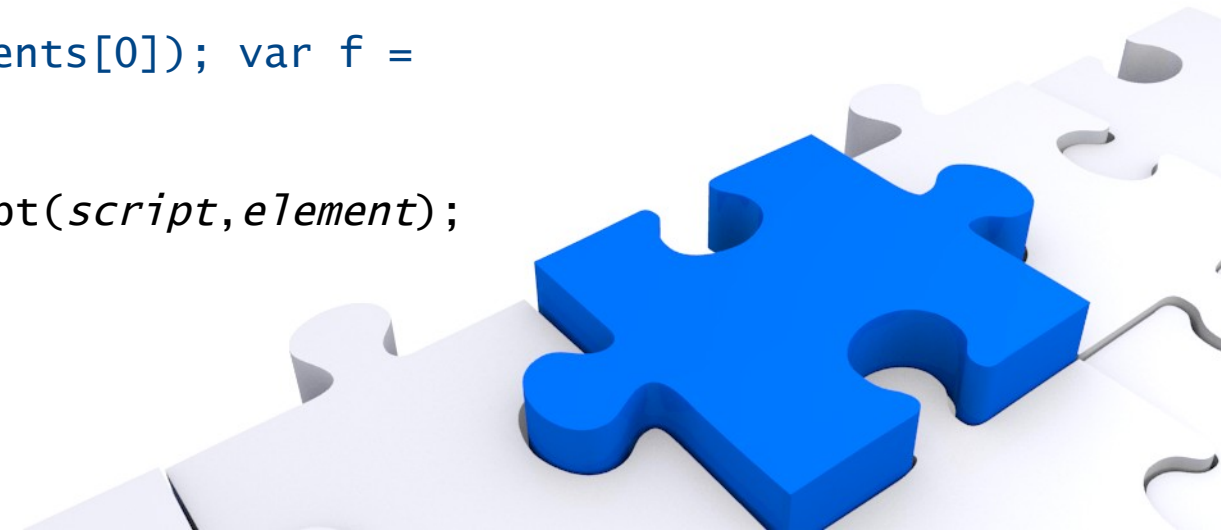
- ByClassName, ByCssSelector, ById, ByLinkText, ByName, ByPartialLinkText, ByTagName, ByXPath (nested classes of By)
- NgBy (JavaScriptBy): binding, buttonText, cssContainingText, model, options, repeater, repeaterRows, repeaterElement, selectedOption, selectedRepeaterOption
- ExpectedConditions
- WebElement.findElements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





# Convenience

```
WebElement element =  
driver.findElement(By.xpath("html/body/div[13]/ul/li[4]/select"));  
  
WebElement element = driver.findElement(By.cssSelector("div#ud-root  
div#outer div.panel div.slogan div.kid"));  
  
(JavascriptExecutor)seleniumDriver.executeScript("arguments[0].style.borde  
r='3px solid yellow'", element);  
  
String script = "var e = angular.element(arguments[0]); var f =  
e.scope().myFile; return f.some_property";  
  
Object result =  
(JavascriptExecutor)seleniumDriver.executeScript(script, element);
```



# Convenience

```
NgWebElement selectedCar =  
ngDriver.findElement(NgBy.model("selectedCar"));  
WebElement customerName = customer.findElement(NgBy.binding("Name"));  
List<WebElement> customers = ngDriver.findElements(NgBy.repeater("cust  
in Customers"));  
Object customerCountry = customer.evaluate("customer.Country");  
NgWebElement cell = ngDriver.findElement(NgBy.repeaterElement("row in  
rows", rowNum, "row.cellName"));
```



# Clientsidescripts.js

```
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;  
        matches.push(bindings[i]);  
        ...  
    }  
    return matches; /* Return the whole array for webdriver.findElements. */  
};
```

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



# 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;  
    }  
}
```

<https://github.com/sergueik/jProtractor/blob/master/src/main/java/com/jprotractor/JavaScriptBy.java>



# 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");  
    }  
}
```

<https://github.com/sergueik/jProtractor/tree/master/src/main/java/com/jprotractor>



# 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/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>

