Providing Protractor to Java / .Net

- Serguei Kouzmine kouzmine_serguei@yahoo.com

Sample bare-bones Angular 1.3 page

```html
<html ng-app="myApp">
...
<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>
```

http://www.java2s.com/Tutorials/Javascript/AngularJS_Example/Form/Bind_
select_option_data_from_array.htm

Typical Angular page Script

```html
<!DOCTYPE html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.2.13/
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>
```
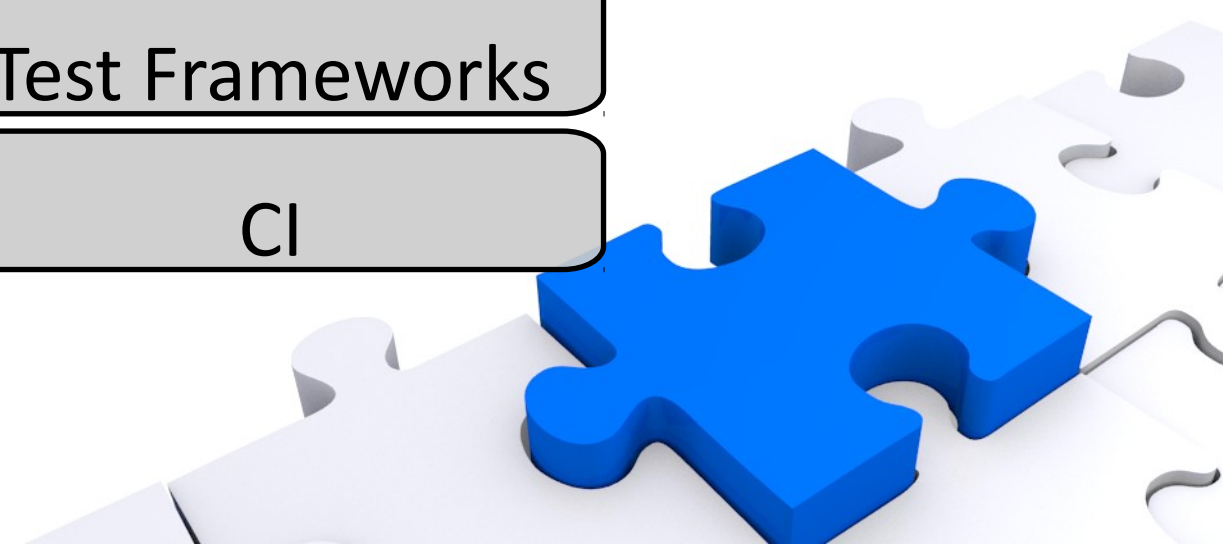
Integration with QA/CI

- 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)

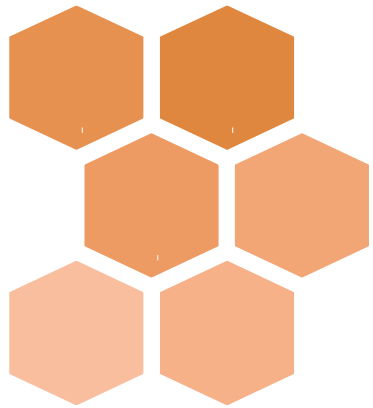- Angular MVC Clientside API were only available through Protractor
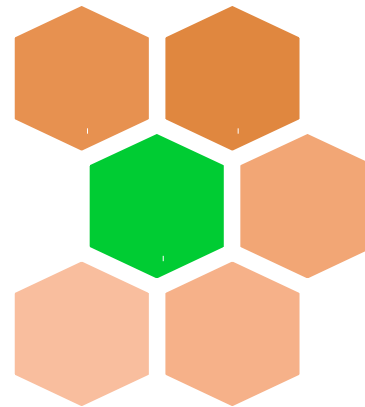
Selenium

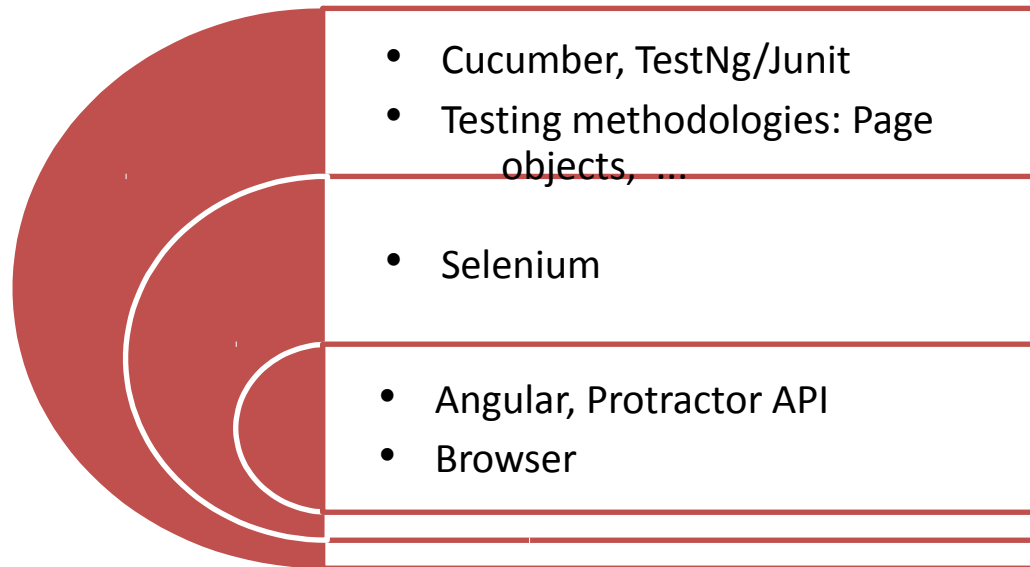Test Frameworks

CI

Adding Protractor Methods

Java

Node.js

jProractor,
Protractor-net

Tool Chain

- Cucumber, TestNg/Junit
- Testing methodologies: Page objects, ...

- Selenium

- Angular, Protractor API
- Browser

WebDriver Wire Protocol

Command	Method	URI Template
	POST	/session/{*session id*}/element
	POST	/session/{*session id*}/element/{element id}/element
	GET	/session/{*session id*}/element/{*element id*}/attribute/{*name*}
	GET	/session/{*session id*}/element/{*element id*}/property/{*name*}
	GET	/session/{*session id*}/element/{*element id*}/css/{*property name*}
	GET	/session/{*session id*}/element/{*element id*}/text
	GET	/session/{*session id*}/element/{*element id*}/name
	GET	/session/{*session id*}/element/{*element id*}/rect
	POST	/session/{*session id*}/execute/sync
	POST	/session/{*session id*}/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

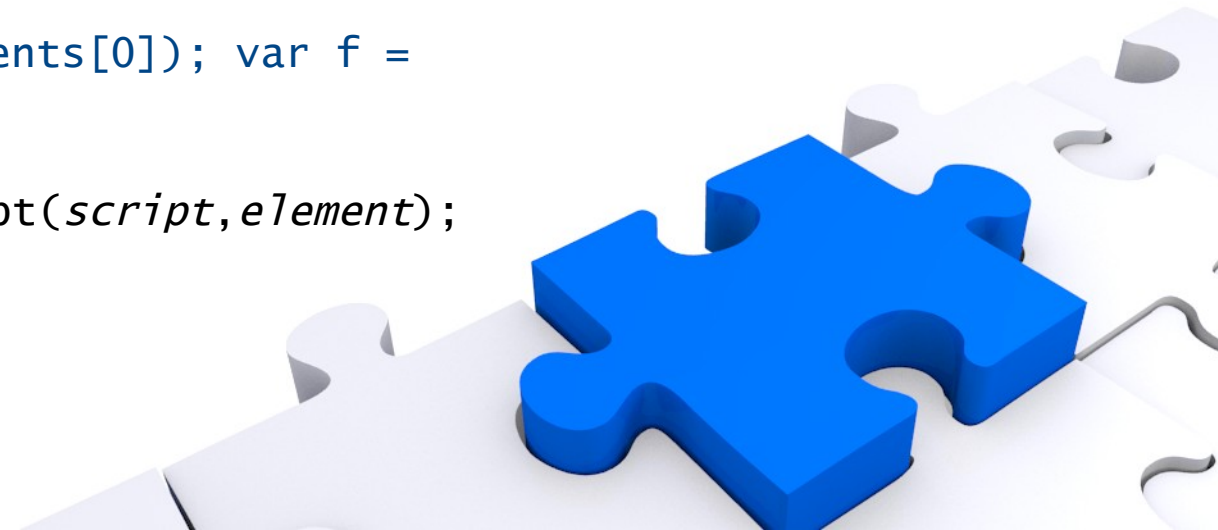- Wait

- SendKeys

- Actions

Selenium Extended 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
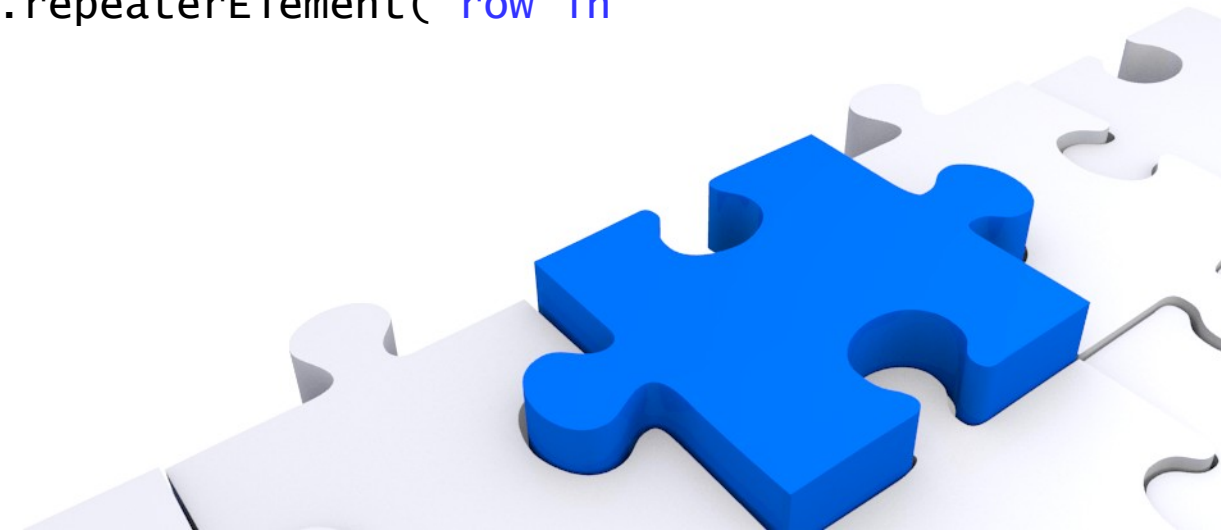
- SendKeys

- 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

```javascript
/**
 * Evaluate an Angular expression in the context of a given element.
 * arguments[0] {Element} The element in whose scope to evaluate.
 * arguments[1] {string} The expression to evaluate.
 * @return {?Object} The result of the evaluation.
 */
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

```javascript
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

```java
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.

```java
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/bbaia/protractor-net
- https://github.com/sergueik/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