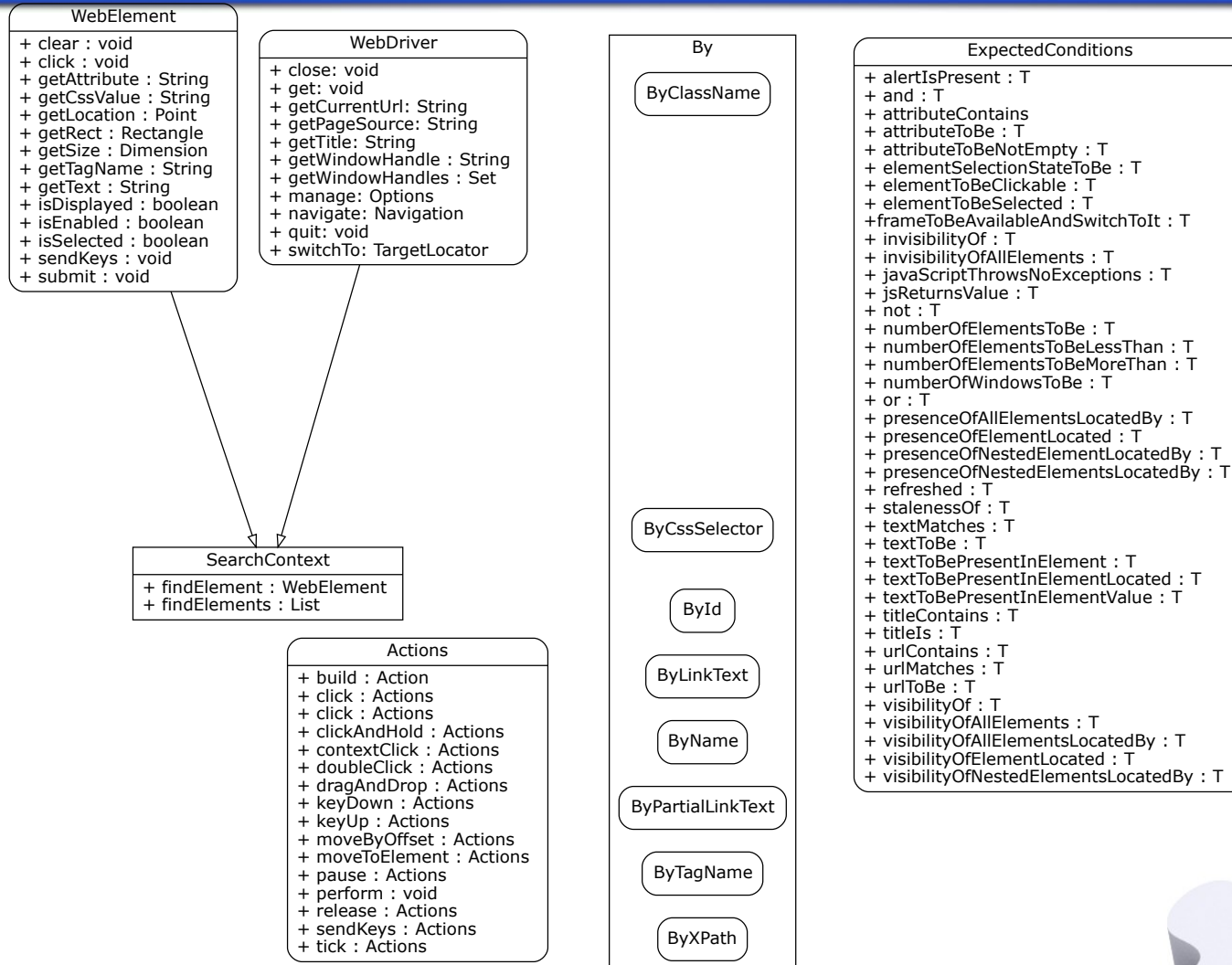


Keyword-Driven Framework for Selenium and Protractor

Serguei Kouzmine
kouzmine_serguei@yahoo.com



Core Selenium Class Hierarchy



Java Protractor Wrapper Methods

The screenshot displays the Java Class Viewer interface, showing the package structure and the methods of the `ByAngular.class` file.

Left Panel (Package Structure):

- Expand All Collapse All
- C:\developer\sergueik\keyword_driven_framework\ngwebdriver\target\lib\ngwebdriver-1.0.jar
 - META-INF
 - com
 - paulhammant
 - ngwebdriver
 - ByAngular\$BaseBy.class
 - ByAngular\$Factory.class
 - ByAngular.class**
 - ByAngularBinding.class
 - ByAngularButtonText.class
 - ByAngularCssContainingText.class
 - ByAngularExactBinding.class
 - ByAngularModel.class
 - ByAngularOptions.class
 - ByAngularPartialButtonText.class
 - ByAngularRepeater.class
 - ByAngularRepeaterCell.class
 - ByAngularRepeaterColumn.class
 - ByAngularRepeaterRow.class
 - NgWebDriver.class
 - VariableNotInScopeException.class

Right Panel (Class File Details):

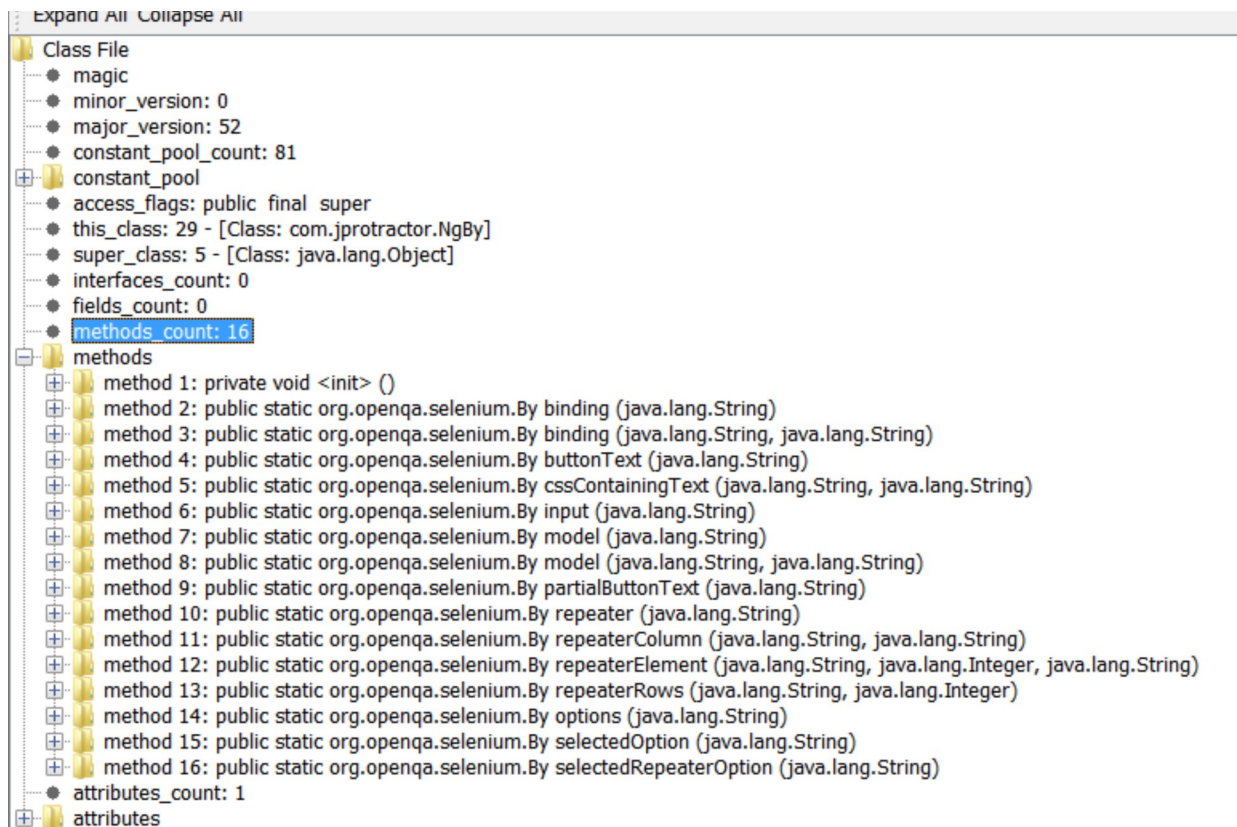
C:\developer\sergueik\keyword_driven_framework\ngwebdriver\target\lib\ngwebdriver-1.0.jar - com

Expand All Collapse All

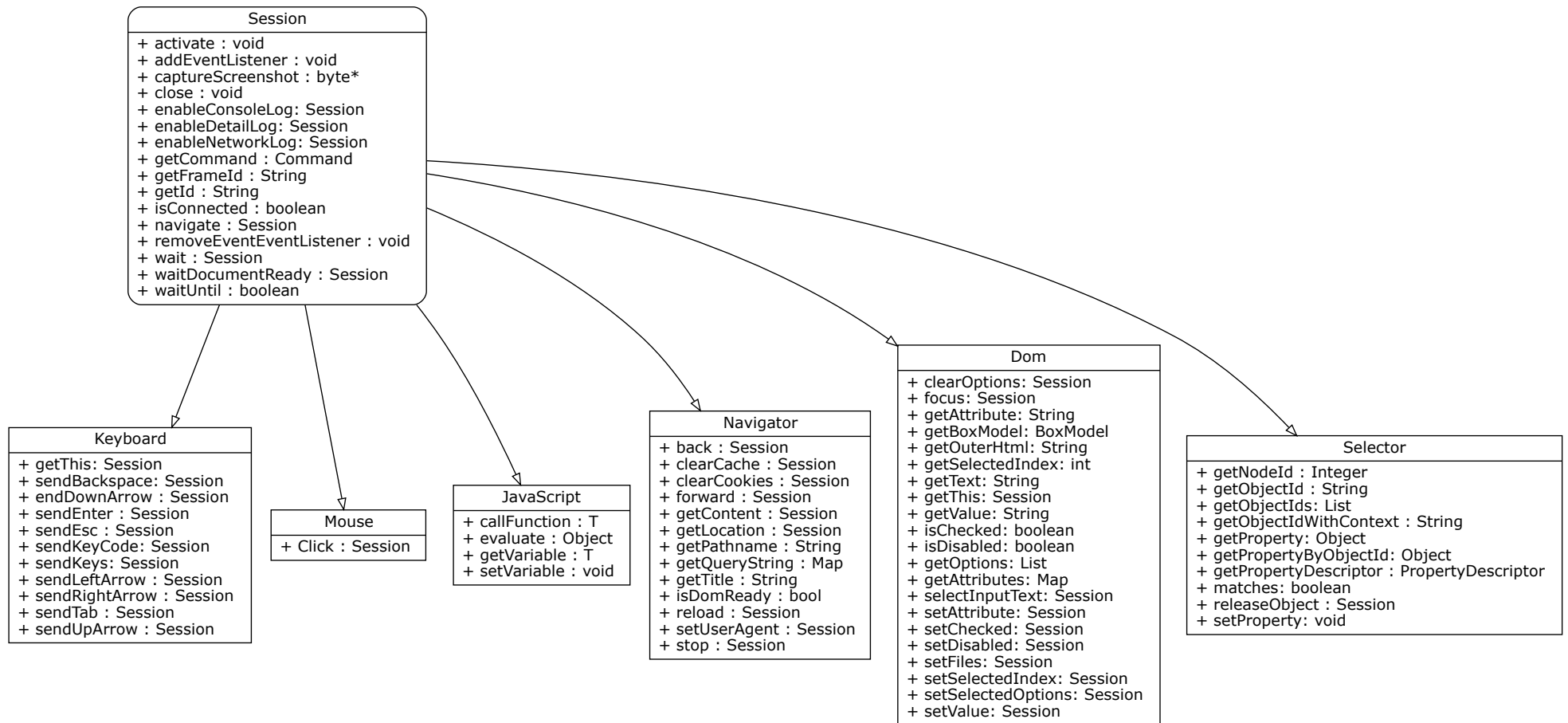
Class File

- magic
- minor_version: 0
- major_version: 49
- constant_pool_count: 256
- constant_pool
 - access_flags: public super
 - this_class: 52 - [Class: com.paulhammant.ngwebdriver.ByAngular]
 - super_class: 72 - [Class: java.lang.Object]
 - interfaces_count: 0
 - fields_count: 1
- fields
- methods_count: 15
- methods
 - method 1: private static void iterateOverJsFunctionsInSource (java.lang.String)
 - method 2: private static void inlineUtilityFunctionsIfNeeded ()
 - method 3: private static void storeJavaScriptFunction (java.lang.String)
 - method 4: private void <init> ()
 - method 5: public static com.paulhammant.ngwebdriver.ByAngular\$Factory withRootSelector (java.lang.String)
 - method 6: public static com.paulhammant.ngwebdriver.ByAngularRepeater repeater (java.lang.String)
 - method 7: public static com.paulhammant.ngwebdriver.ByAngularRepeater exactRepeater (java.lang.String)
 - method 8: public static com.paulhammant.ngwebdriver.ByAngularBinding binding (java.lang.String)
 - method 9: public static com.paulhammant.ngwebdriver.ByAngularExactBinding exactBinding (java.lang.String)
 - method 10: public static com.paulhammant.ngwebdriver.ByAngularModel model (java.lang.String)
 - method 11: public static com.paulhammant.ngwebdriver.ByAngularOptions options (java.lang.String)
 - method 12: public static com.paulhammant.ngwebdriver.ByAngularButtonText buttonText (java.lang.String)
 - method 13: public static com.paulhammant.ngwebdriver.ByAngularPartialButtonText partialButtonText (java.lang.String)
 - method 14: public static com.paulhammant.ngwebdriver.ByAngularCssContainingText cssContainingText (java.lang.String, java.lang.String)
 - method 15: static void <clinit> ()

Alternative Java Protractor Wrapper Methods



CDP Class Hierarchy



<https://webfolder.io/cdp4j/javadoc/index.html>

TestCase, core Selenium

	A	B	C	D	E	F	G	H	I
1	Keyword	param1	param2	param3	param4	param5	param6	param7	status
2	CREATE_BROWSER								Passed
3	GOTO_URL	http://www.seleniumeasy.com/test/						1	Passed
4	clickLink	css	#navbar-brand-centered > ul:nth-child(1) > li:nth-child(1) > a						Passed
5	CLICK_LINK	text	Simple Form Demo						Passed
6	SET_TEXT	css	#user-message			hello			Passed
7	GET_TEXT	css	#user-message						Passed
8	GET_ATTR	css	#user-message			outerHTML			Passed
9	getElementAttribute	css	#user-message			value			Passed
10	CLOSE_BROWSER								Passed



Test Case, Protractor

A	B	C	D	E	F	G	H	I
Keyword	param1	param2	param3	param4	param5	param6	param7	status
CREATE_BROWSER								Passed
GOTO_URL	http://juliemr.github.io/protractor-demo/							Passed
SET_TEXT	model	first			40			Passed
SET_TEXT	model	second			2			Passed
GET_ATTR	options	value for (key, value) in operators			innerHTML			Passed
CLICK_BUTTON	buttontext	Go!						Passed
GET_TEXT	binding	latest						Passed
CLOSE_BROWSER								



Test Case, Protractor

	A	B	C	D	E	F	G	H	I
1	Keyword	param1	param2	param3	param4		param6	param7	status
2	CREATE_BROWSER								
3	GOTO_URL	http://amitava82.github.io/angular-multiselect/							
4	VERIFY_TAG	model	selectedCar			am-multiselect			
5	CLICK_BUTTON	buttontext	Select Some Cars						
6	WAIT							1000	
7	VERIFY_TEXT	repeaterElement	i in items	2	i.label	Honda			
8	CLICK_LINK	repeaterElement	i in items	1	i.label				
9	CLICK_LINK	repeaterElement	i in items	0	i.label				
10	WAIT							1000	
11	VERIFY_TEXT	css	am-multiselect > div > button			There are 2 car(s) selected			
12	CLOSE_BROWSER								

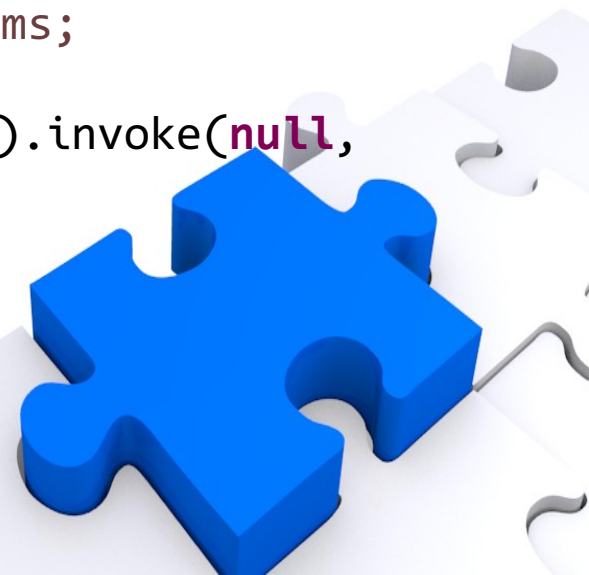


Keyword processing through Java Reflection

```
private static Map<String, String> methodTable = new HashMap<>();
    static {
        methodTable.put("CLICK_BUTTON", "clickButton");
        ...
    }
private static final Class<?> _class = Class.forName("Keyword Library
class");

public static void callMethod(String keyword,
                               Map<String, String> _params;

    // for static methods
    _class.getMethod(methodTable.get(keyword), Map.class).invoke(null,
_params);
    // for instance methods
    _class.getMethod(methodTable.get(keyword),
Map.class).invoke( _class.newInstance(), _params);
```



Building Steps from Excel

```
List<Object[]> steps = utils.createDataFromExcel2003(testCase);  
    // utils.createDataFromExcel2007(testCase);  
    // utils.createDataFromOpenOffice(testCase);  
  
for (int step = 0; step < steps.size(); step++) {  
    Object[] row = steps.get(step);  
    Map<String, String> data = new HashMap<>();  
    String keyword = (String) row[0];  
    for (int col = 1; col < row.length; col++) {  
        data.put(String.format("param%d", col), row[col].toString());  
    }  
    KeywordLibrary.callMethod(keyword, data);  
}
```



Keyword method implementanion

```
provate static WebDriver driver;  
private static NgWebDriver ngDriver;  
private static WebElement element;  
private static String status;  
private static String result;
```

```
public static void clickButton(Map<String, String> params) {  
    element = _findElement(params);  
    if (element != null) {  
        highlight(element);  
        element.click();  
        status = "Passed";  
    } else {  
        status = "Failed";  
    }  
}
```



Extracting arguments from params map

```
public static void verifyText(Map<String, String> params) {  
    boolean flag = false;  
    expectedText = params.get("param5");  
    element = _findElement(params);  
    if (element != null) {  
        highlight(element);  
        flag = element.getText().equals(expectedText);  
    }  
    if (flag)  
        status = "Passed";  
    else  
        status = "Failed";  
}
```



Find element using Selenium and Protractor

```
private static WebElement _findElement(Map<String, String> params) {  
    strategy = params.get("param1");  
    if (!strategies.containsKey(strategy)) {  
        throw new RuntimeException("Unknown strategy: " + strategy);  
    }  
    value = params.get("param2");  
    if (params.containsKey("param3")) {  
        row = params.get("param3");  
    }  
    if (params.containsKey("param4")) {  
        column = params.get("param4");  
    }  
  
    if (params.containsKey("param5")) {  
        containedText = params.get("param5");  
    }  
    if (params.containsKey("param6")) {  
        tagName = params.get("param6");  
    }  
}
```



Supported Locator Strategies

```
private static Map<String, Method> strategies = new HashMap<>();
strategies.put("css",
    By.class.getMethod("cssSelector", String.class));
strategies.put("id",
    By.class.getMethod("id", String.class));

strategies.put("binding",
    NgBy.class.getMethod("binding", String.class));
strategies.put("repeaterColumn",
    NgBy.class.getMethod("repeaterColumn", String.class,
        String.class));
strategies.put("repeaterElement",
    NgBy.class.getMethod("repeaterElement", String.class,
        Integer.class, String.class));

strategies.put("repeaterRow",
    methodMissing);
```



Prohy method

```
// phony method
Method methodMissing = null;
try {
    @SuppressWarnings("rawtypes")
    Constructor<Method> methodConstructor = Method.class
        .getDeclaredConstructor(new Class[] { String.class,
            String.class });
    methodConstructor.setAccessible(true);
    methodMissing = (Method) methodConstructor.newInstance();
} catch (NoSuchMethodException |
        IllegalAccessException |
        InstantiationException |
        IllegalArgumentException |
        InvocationTargetException e) {
    System.out.println(e.toString()); // slurp
}
```



Find element using Selenium and Protractor, contd.

```
WebElement _element = null;
```

```
switch (strategy) {  
    case "binding":  
        _element = ngDriver.findElement(NgBy.binding(value));  
        break;  
  
    case "css":  
        _element = driver.findElement(By.cssSelector(value));  
        break;  
  
    case "repeaterRow":  
    case "repeaterRows":  
        _element = ngDriver.findElement(NgBy.repeaterRows(  
            value, row));  
        break;  
}  
return _element;  
}
```



Find element extending Selenium and Protractor, contd.

case "text":

// Option 1: construct XPath selector

```
Map<String, String> newParams = new HashMap<>();
newParams.put("param1", "xpath");
newParams.put("param2", String.format(
    "//%s[contains(normalize-space(text()), '%s')]",
    (tagName != null) ? tagName : "*", value));
```

```
_element = _findElement(newParams);
```

// Option 2: use Java streams/filter

```
if (tagName != null) {
    _element = driver.findElements(
        By.tagName(tagName)).stream().filter(
            o -> o.getText().contains(value))
        .findFirst().get();
}
break;
```



Finding elements with explicit wait

```
switch (strategy) {
    case "css":
        locator = By.cssSelector(selectorValue);
        break;
}
timeout = (long) (Float.parseFloat(params.get("param7")));
_wait = new WebDriverWait(driver, timeout);
_element = _wait.until(new ExpectedCondition<WebElement>() {
    @Override
    public WebElement apply(WebDriver d) {
        Optional<WebElement> e = d.findElements(locator)
            .stream()
            .findFirst();

        if (e.isPresent()) // log some debugging info
            logger.debug("find using strategy: " + strategy +
                " => " + e.get().getAttribute("outerHTML"));
        return (e.isPresent()) ? e.get() :
            (WebElement) null;
    }
});
```

