

Selenide



Jumpstart your test automation

Paweł Wasiulewski and

Aleksander Zaleski

Kim jesteśmy?

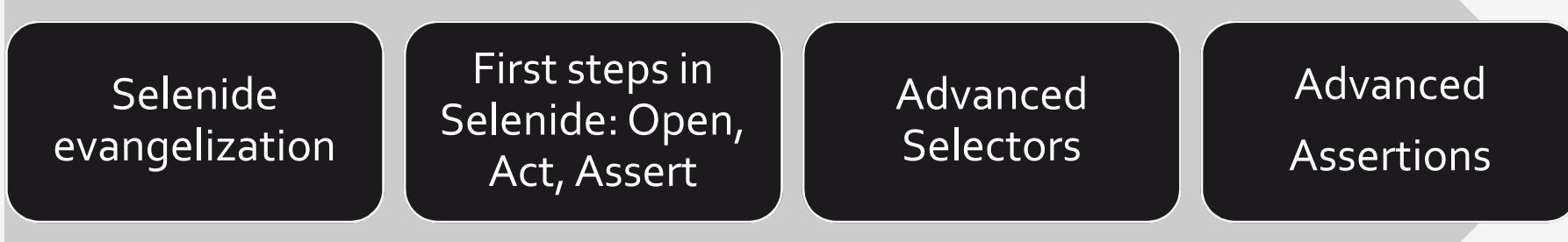


Paweł Wasiulewski
Filmy, Konsole, rzutki.
Testowanie: gier, protokołów, konfiguratorów



Aleksander Zaleski
Konferencje, Quizy, Hulajnoga,
Testowanie: eksploracyjne i każde inne





Agenda



*What is
Selenium
Webdriver?*



Selenide



Selenide is a concise and expressive wrapper API specifically for **Selenium WebDriver**, which is used to control browser actions.

Selenide is not a framework it is a library.

Be prepared!

Tweets by @jselenide

 **selenide** @jselenide
@iNikem @testcontainers @asolntsev Sure!
github.com/selenide-examp...

 **selenide-examples/testcontainers**
Selenide + TestContainers (Docker) sample project.
Contribute to testcontainers development by creating an account on GitHub.
github.com

13h

 **selenide** @jselenide
We released Selenide 4.8 with support for #selenium 3.6.0!

Don't worry, be up-to-date.

Oct 8, 2017

 **selenide** @jselenide
Army of Selenide lovers is growing!

Advantages

- Readable
- Less Code
- Selectors: extra + parent
- Filters
- Screen shots
- Automatic Waits
- Library, !Framework
- Open-source [MIT license](#)



<https://prezi.com/d18jggopjyaj/selenide-shorten-your-code/>

Webdriver vs Selenide

```
WebElement customer = driver.findElement(By.id("customerContainer"));
```

VS

```
WebElement customer = $(By.id("customerContainer"));
```

```
assertEquals("Customer profile", driver.findElement(By.id("customerContainer")).getText());
```

VS

```
$("#customerContainer").shouldHave(text("Customer profile"));
```

Webdriver vs Selenide

```
FluentWait<By> fluentWait = new FluentWait<By>(By.tagName("TEXTAREA"));
fluentWait.pollingEvery(100, TimeUnit.MILLISECONDS);
fluentWait.withTimeout(1000, TimeUnit.MILLISECONDS);
fluentWait.until(new Predicate<By>() {
    public boolean apply(By by) {
        try {
            return browser.findElement(by).isDisplayed();
        } catch (NoSuchElementException ex) {
            return false;
        }
    }
});
assertEquals("John", browser.findElement(By.tagName("TEXTAREA")).getAttribute("value"));
```

VS

```
$( "TEXTAREA" ).shouldHave(value("John"));
```

Browsers

- Chrome
- IE
- Edge
- Firefox (geckodriver)
- PhantomJS
- Safari



Three simple steps

open(page)

\$(element) doAction

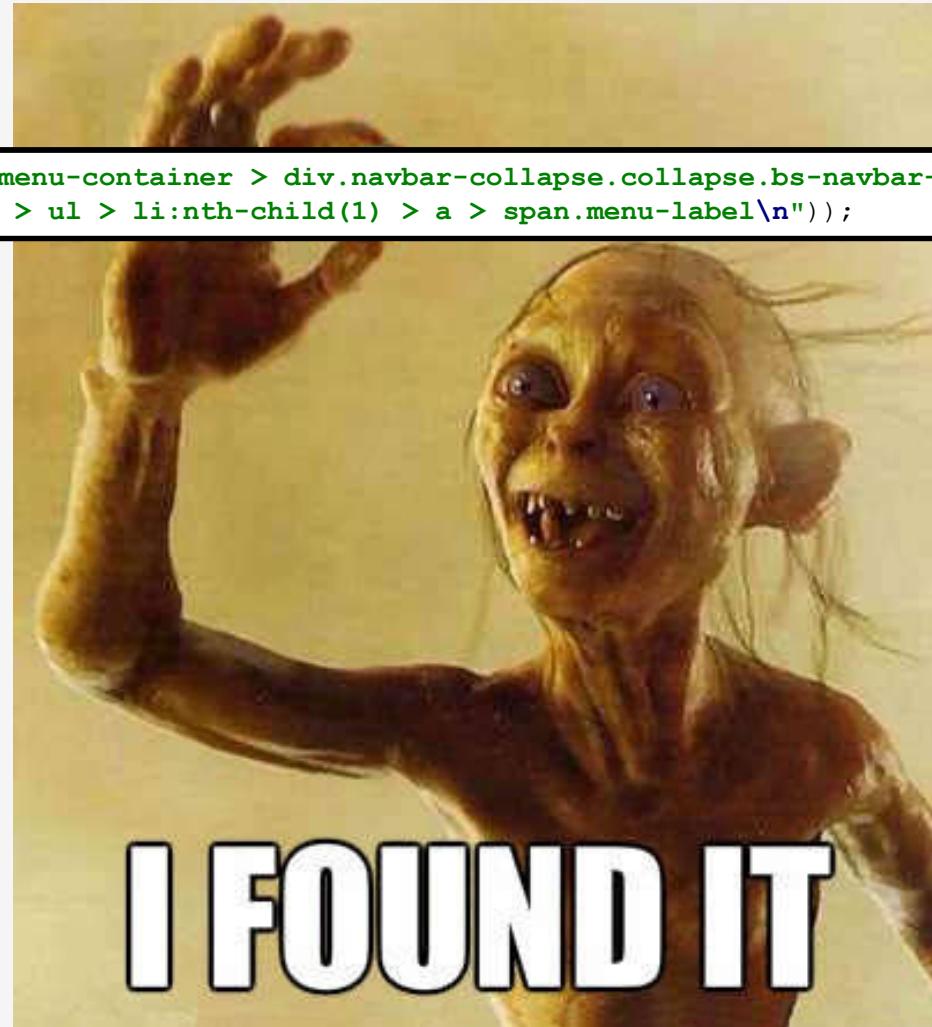
\$(element) check(condition)

```
open("http://automationpractice.com/index.php");
$("#contact-link").click();
$(".navigation_page").should(appear);
```

Selectors

\$(element).doAction;

```
driver.findElement(By.cssSelector("#menu-container > div.navbar-collapse.collapse.bs-navbar-collapse > ul:nth-child(1) > li.dropdown.open > ul > li:nth-child(1) > a > span.menu-label\n"));
```



Selectors:

\$=find

```
$("#elementid");// By id (same as $(byId("elementid"));)  
$(".classname");//By class (same as $(byClassName("classname"));)  
$(byText("cheezburger"));// Searches exact text  
$(withText("ezburger"));//Search based on partial text  
$(by("md-svg-src", "img/icons/android.svg"));//Search based on attribute  
$("element", 1); // index starts with 0, the second element will be selected  
$(byTitle("title"));// By title (same as $("[title=tytul]"));  
$(byValue("value"));// By value (same as $("[value=wartosc]"));
```

Chainable Selectors:

Multiple selectors can be used in one statement:

```
$("div.container").$("button.buycheesburger");  
$(".container").$(".subarea").$(withText("ezburger"));
```

Parent and closest(tag) helps navigation in DOM:

```
$(byText("Cheezburger")).parent().closest("ul").shouldBe(visible);
```

Using and/or with selectors:

Only one of the requirements has to be met:

```
$( "h1" ).shouldBe( or( ".class", visible, hidden ) );
```

All of the requirements have to be met:

```
$( "h1" ).shouldBe( and( "hamburger", visible, not( disabled ) ) );
```

Mouse and keyboard actions

```
$(element).doAction();
```



Mouse actions:

```
$(".class").click();  
$("#id").contextClick(); //right click on element  
$("h1").doubleClick(); //double click on element  
$(".class").hover(); //mouse over element  
$("#id").scrollTo(); //scrolling to element
```

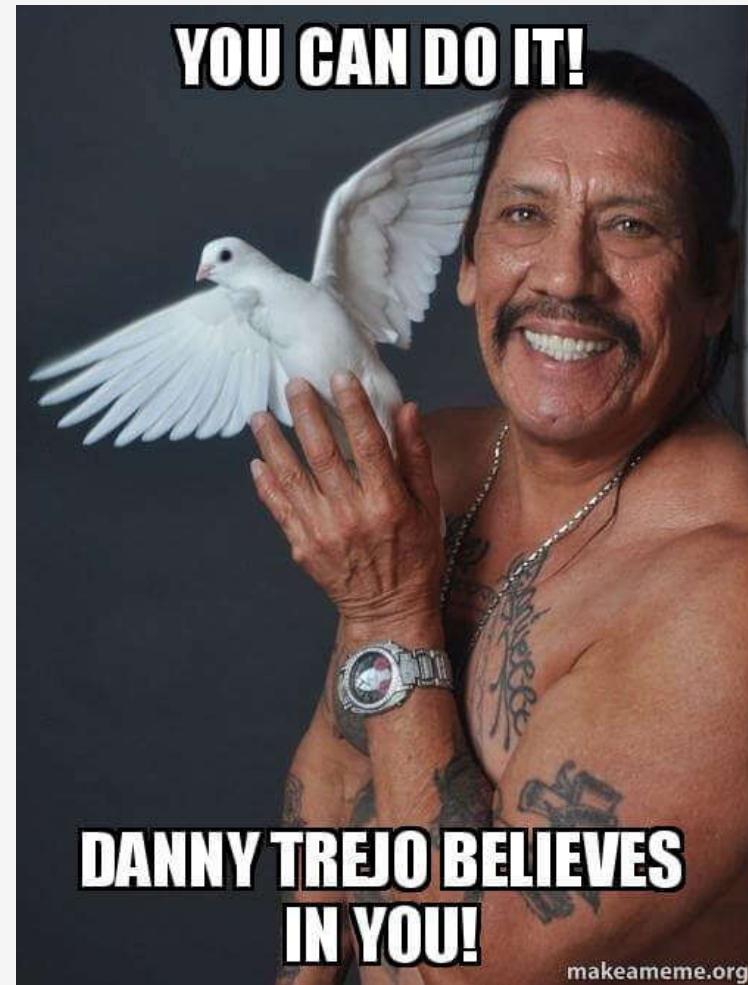
Keyboard actions:

```
actions().keyDown(Keys.ALT)  
    .sendKeys("a")  
    .keyUp(Keys.ALT);
```

Excercise 1

Clicking on an element

1. Open www.automationpractice.com page
2. Find „Sign in” button by class/title/withText and click
3. Find „Create an Account” button by text/class/index and click
4. Find „Women” button by title/class/text and click
5. Scroll to the last dress on the page and click



Assertions

```
$(element).check(condition);
```



IT IS A BUG

I CAN TELL FROM SOME OF THE PIXELS AND
FROM SEEING QUITE A FEW BUGS IN MY LIFE

Visibility and existence assertions:

Positive assertions:

```
$("selector").should(appear); // Same as $("selector").shouldBe(visible);  
$("selector").should(exist); //element should exist in DOM
```

Negative assertions:

```
$("selector").shouldBe(hidden); // Same as $("selector").should(disappear);  
$("selector").shouldNot(exist); //element should not exist in DOM
```

Text assertions:

```
$("selector").shouldHave(text("CHEEZBURGER")) ;  
//case insensitive, partial match  
  
$("selector").shouldHave(textCaseSensitive("Cheezburger")) ;  
//case sensitive, partial match  
  
$("selector").shouldHave(exactText("DOUbLe CHeezBurGer")) ;  
//case insensitive, exact match  
  
$("selector").shouldHave(exactTextCaseSensitive("Double Cheezburger")) ;  
//case sensitive, exact match  
  
$("selector").should(matchText("[A..Z]eezburger")) ;  
//regexp
```

Other assertions:

```
$("selector").shouldHave(attribute("layout", "row"));  
//with value  
  
$("selector").shouldNotHave(attribute("href"));  
//any value  
  
$("selector").shouldHave(cssClass(".class"));  
  
$("[type=button]").shouldHave(type("button"));  
  
$("[type=button]").shouldNotHave(id("idselector"));
```

Excercise 2

Assert that element is visible/appears

1. Find „Women” menu element by title and hover over it
2. Check that **Evening Dresses** element is visible/appears
3. Click on **Evening Dresses**
4. Check that product category image has „Evening Dresses” text and does not have text „Casual dresses”



Pause



Collections

\$(selector)



Collections basics:

Collections are iterable SelenideElements:

```
ElementsCollection collection = $$("div");  
$$("selector"); // doesn't search anything!
```

They use the same Selectors as in \$:

```
$$ (withText("eezburger"));
```

Operations on Collections:

Choosing an element:

```
$$("div").first();
$$("div").last();
$$("div").get(2); // get third element, index from 0
$$("div").findBy(text("123")); // finds the first element
```

Filtering:

```
$$("div").filterBy(text("123")); //only with text 123
$$("div").excludeWith(text("123")); // all except with text 123
```

Collection assertions:

Size assertion:

```
$$("div").shouldBe(CollectionCondition.empty);  
$$("div").shouldHave(size(5)); / or $$("div").shouldHaveSize(5);  
$$("div").shouldHave(sizeGreaterThan(1));  
$$("div").shouldHave(sizeGreaterThanOrEqual(1));  
$$("div").shouldHave(sizeLessThan(100));  
$$("div").shouldHave(sizeLessThanOrEqual(100));  
$$("div").shouldHave(sizeNotEqual(1984));
```

Text assertions (case sensitive):

```
$$("div").shouldHave(texts("kjlj", "TTHT")); //partial matches of every  
single text  
$$("div").shouldHave(exactTexts("Here is your", "Hamburger"));
```

Operations returns:

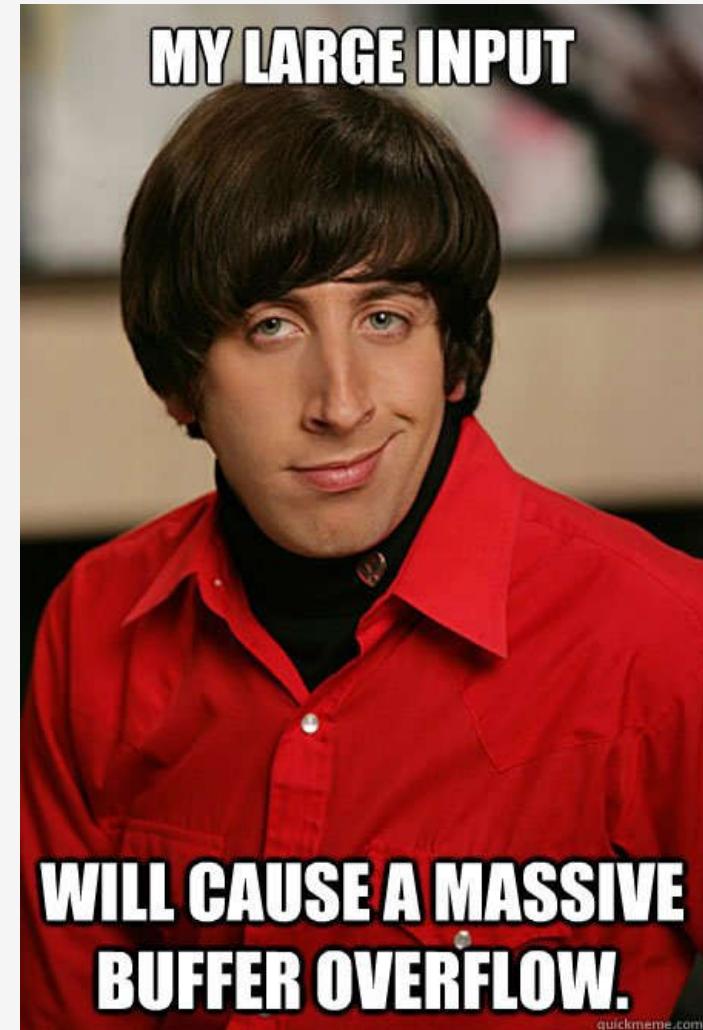
```
ElementsCollection collection = $$("div").filterBy(text("123"));
// even when filters to a single element!
ElementsCollection collection2 = $$("div").filterBy(text("xyz"));
// empty collection!
SelenideElement element = $$("div").get(2);
SelenideElement element2 = $$("div").find(text("xyz")); // null?
```

Excercise 3

1. Click on „Dresses” button
2. Find collection of all dresses and filter them by text to find „Printed Summer Dress”.
Then use get(index) and click first element
3. Assert that there is the correct numer of colors available to choose for this dress



Inputs, dropdowns and checkboxes



Inputs:

```
$("#name").setValue("Aleks");  
$("#name").shouldHave(value("Aleks"));
```

Input assertions:

```
$("input[ng-model='user.company']").shouldBe(disabled);  
$("input[ng-model='user.firstName']").shouldBe(enabled);  
$("input[ng-model='user.firstName']").shouldBe(empty);  
$("input[ng-model='user.company']").shouldHave(value("Goo"));  
//partial match  
$("input[ng-model='user.company']").shouldHave(exactValue("Google"));
```

Checkboxes:

```
$("input[type=checkbox]").click();  
$("input[type=checkbox]").shouldBe(checked);  
  
//or  
$("input[type=checkbox]").setSelected(true);  
$("input[type=checkbox]").shouldBe(checked);  
  
$("input[type=checkbox"], 1).setSelected(false);  
$("input[type=checkbox"], 1).shouldNotBe(checked);
```

ARE YOU DRUNK?

YES

NO

X

Dropdown menu:

Only standard HTML selects

```
$("#dropdown").selectOption("Option 2"); //exact text match  
$("#dropdown").selectOptionContainingText("ption 1"); //partial match  
$("#dropdown").selectOption(3); //starting from 0, fourth option  
$("#dropdown").selectOptionByValue("1");  
  
$("#dropdown").getSelectedOption().shouldBe(disabled);  
$("#dropdown").getSelectedOption().shouldHave(text("Option 1"), value("1"));
```

Excercise 4

1. Go to „Sign in”
2. Enter an email and click „Create account”
3. Fill in all of the Personal Information fields, checkboxes, dropdowns, etc.

GOOD LUCK! (You might need it ☺)

I am a tiny potato

And I believe in you



YOU CAN DO THE THING



makeameme.org

Page Object Pattern (POP)

"Page objects are a classic example of encapsulation - they hide the details of the UI structure and widgetry from other components (the tests)."

(c) Martin Fowler, PageObject.

What is NOT Page Object

```
public class GoogleTest {  
    @BeforeClass  
    public static void setup() {  
        Configuration.baseUrl = "http://google.com/ncr"  
    }  
    @Test  
    public void userCanSearch() {  
        open("/");  
        $(By.name("q")).val("selenide").pressEnter();  
        $$(".srg .g").shouldHave(size(10));  
        $$(".srg .g").get(0).shouldHave(text("Selenide: concise UI tests  
in Java"));  
    }  
}
```

POP Mid-level encapsulation

```
public class GoogleTest {  
    @BeforeClass  
    public static void setup() {  
        Configuration.baseUrl = "http://google.com/ncr"  
    }  
    @Test  
    public void userCanSearch() {  
        GooglePage google = new GooglePage();  
        google.open().searchFor("selenide");  
        google.results().shouldHave(size(10));  
        google.results().get(0).shouldHave(text("Selenide:  
concise UI tests in Java"));  
    }  
}
```

POP Mid-level encapsulation

```
public class GoogleTest {  
    @BeforeClass  
    public static void setup() {  
        Configuration.baseUrl = "http://google.com/ncr"  
    }  
    @Test  
    public void userCanSearch() {  
        GooglePage google = new GooglePage();  
        google.open().searchFor("selenide");  
        google.results().shouldHave(size(10));  
        google.results().get(0).shouldHave(text("Selenide:  
concise UI tests in Java"));  
    }  
}
```

POP higher-level encapsulation

```
public class GoogleTest {  
  
    GooglePage google = new GooglePage();  
  
    SearchResults results = new SearchResults();  
  
    @BeforeClass  
  
    public static void setup() {  
  
        Configuration.baseUrl = "http://google.com/ncr"  
    }  
  
    @Test  
  
    public void userCanSearch() {  
  
        GooglePage google = new GooglePage();  
  
        google.open().searchFor("selenide");  
  
        results.shouldHaveSize(10)  
  
        .shouldHaveResult(o, "Selenide: concise UI tests in Java");  
    }  
}
```

POP higher-level encapsulation

```
public class GooglePage {  
    public GooglePage open() {  
        Selenide.open("/");  
        return this;  
    }  
  
    public SearchResults searchFor(String text) {  
        $(By.name("q")).val(text).pressEnter();  
        return new SearchResults();  
    }  
}
```

POP higher-level encapsulation

```
public class SearchResults {  
    private final ElementsCollection elements = $$(".srg .g");  
  
    public SearchResults shouldHaveSize(int size) {  
        elements.shouldHaveSize(size);  
        return this;  
    }  
  
    public SearchResults shouldHaveResult(int index, String text) {  
        elements.shouldHave(text(text));  
        return this;  
    }  
}
```

References

- <http://selenide.org/documentation.html>
- <https://selenide.gitbooks.io/user-guide/content/en/three-simple-steps.html>
- Page Objects <https://selenide.gitbooks.io/user-guide/content/en/pageobjects.html>
- Feature request:
<https://github.com/codeborne/selenide/issues>

Video:

- Selenide101 – Getting up and running with real web tests (Alexei Vinogradov, Germany) – Parts 1 and 2
<https://www.youtube.com/watch?v=OQUk6xYYSkw>
- Selenide & KISS-Driven-Development :
<https://www.youtube.com/watch?v=Q7fqTLOr2YE>

*Thank you
for attention!*



Les frères, rue du Docteur Lecène Paris
1934 Doisneau