SELENIUM

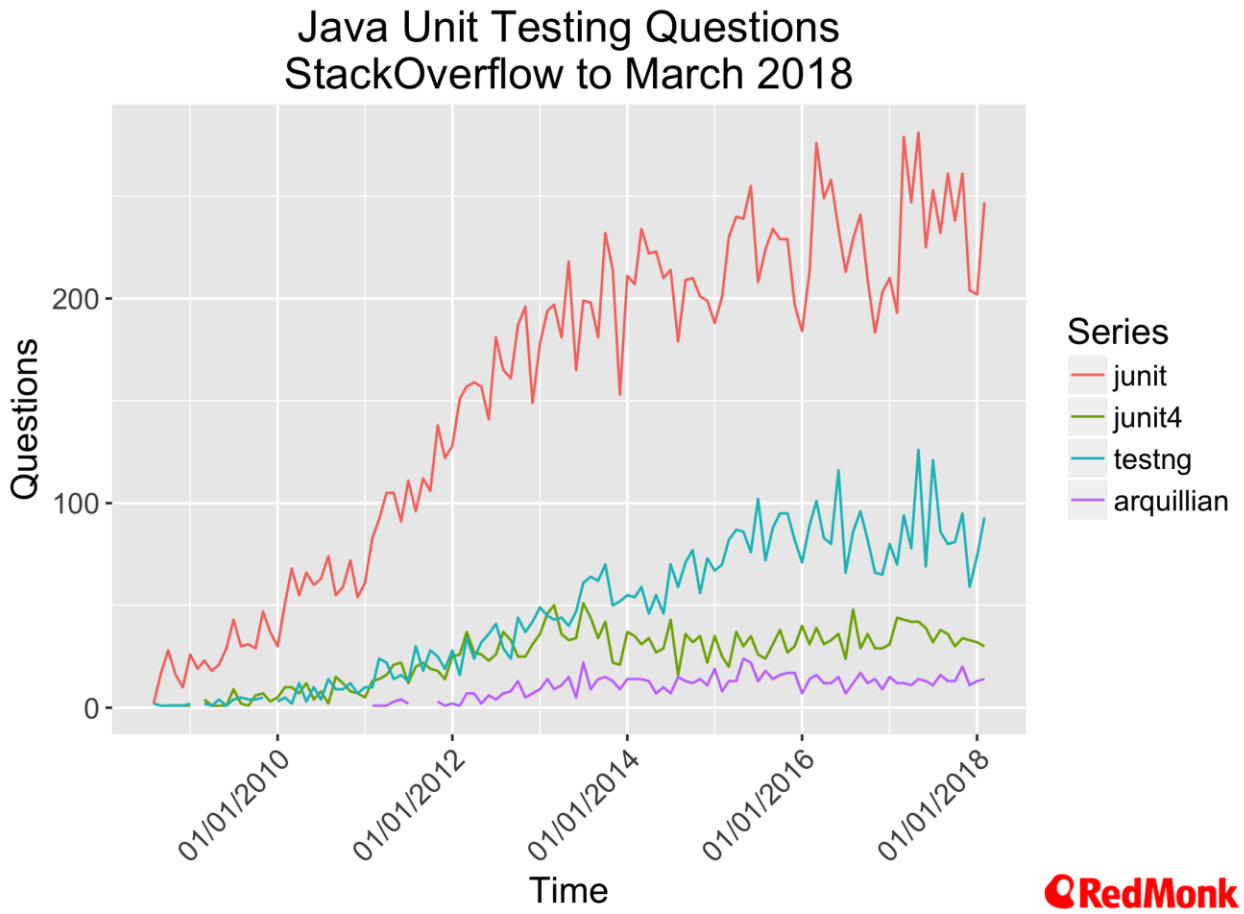
ELŻBIETA SĄDEL

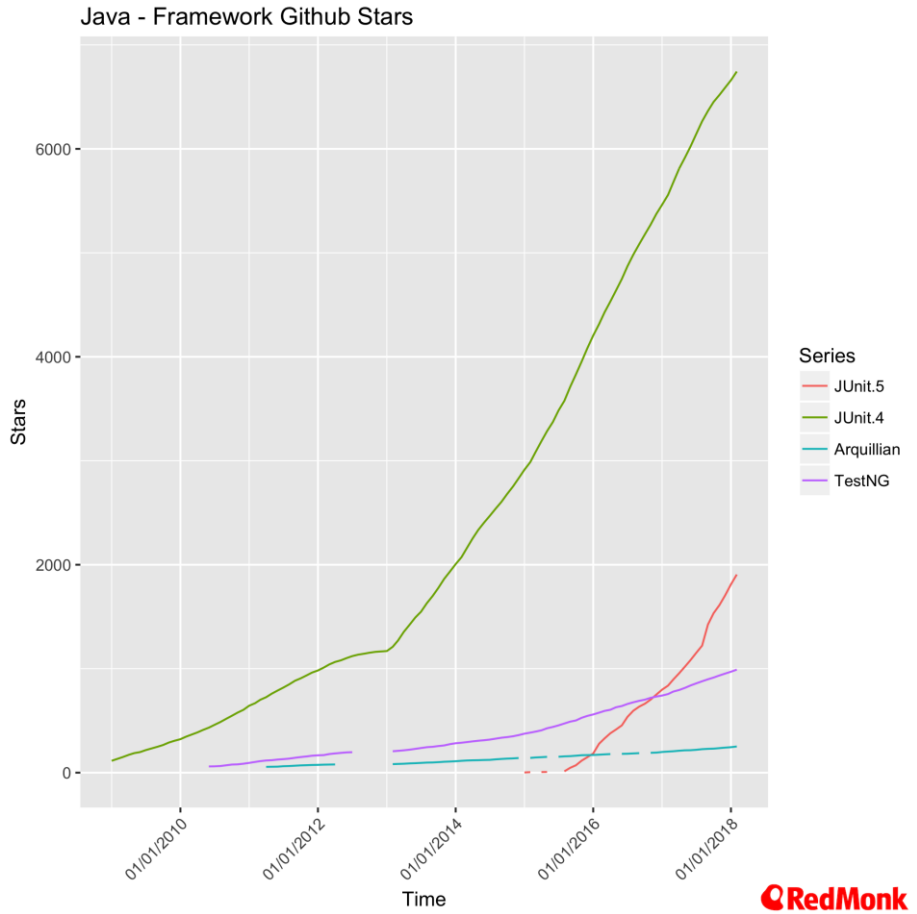
GRZEGORZ WITEK

DLACZEGO JAVA I JUNIT?

Aug 2018	Aug 2017	Change	Programming Language	Ratings	Change
1	1		Java	16.881%	+3.92%
2	2		C	14.966%	+8.49%
3	3		C++	7.471%	+1.92%
4	5	^	Python	6.992%	+3.30%
5	6	^	Visual Basic .NET	4.762%	+2.19%
6	4	v	C#	3.541%	-0.65%
7	7		PHP	2.925%	+0.63%
8	8		JavaScript	2.411%	+0.31%
9	-	^^	SQL	2.316%	+2.32%
10	14	^^	Assembly language	1.409%	-0.40%

DLACZEGO JAVA I JUNIT?





ASSERT I SOFT ASSERT

- **Asercja** (ang. assertion) – predykat (forma zdaniowa w danym języku, która zwraca prawdę lub fałsz), umieszczony w pewnym miejscu w kodzie. W przypadku gdy predykat jest fałszywy (czyli niespełnione są warunki postawione przez programistę) asercja powoduje przerwanie wykonania programu.

- **Miękka asercja** (ang. soft assertion) – W przypadku gdy predykat jest fałszywy (czyli niespełnione są warunki postawione przez programistę) miękka asercja nie powoduje przerwanie wykonania programu po każdym warunku lecz dopiero po wywołania metody sprawdzającej wszystkie warunki.

ARRANGE-ACT-ASSERT

- **Arrange** - przygotowywanie testu – ustawienie wszystkich danych wejściowych i konfiguracja środowiska testowego
- **Act** - wykonanie testu – wszystkie kroki, które są niezbędne do sprawdzenia danej funkcjonalności
- **Assert** - sprawdzenie poprawności wykonania testu – wywoływanie asercji sprawdzających czy zaistniałą sytuacja jest zgodna z oczekiwaną

ARRANGE-ACT-ASSERT

- **Arrange** - przygotowywanie testu – ustawienie wszystkich danych wejściowych i konfiguracja środowiska testowego
  - **Given** – w jakim stanie znajduje się system
- **Act** - wykonanie testu – wszystkie kroki, które są niezbędne do sprawdzenia danej funkcjonalności
  - **When** – kroki wykonywane przez użytkownika
- **Assert** - sprawdzenie poprawności wykonania testu – wywoływanie asercji sprawdzających czy zaistniałą sytuacja jest zgodna z oczekiwaną
  - **Then** – oczekiwany stan po wykonaniu wszystkich kroków

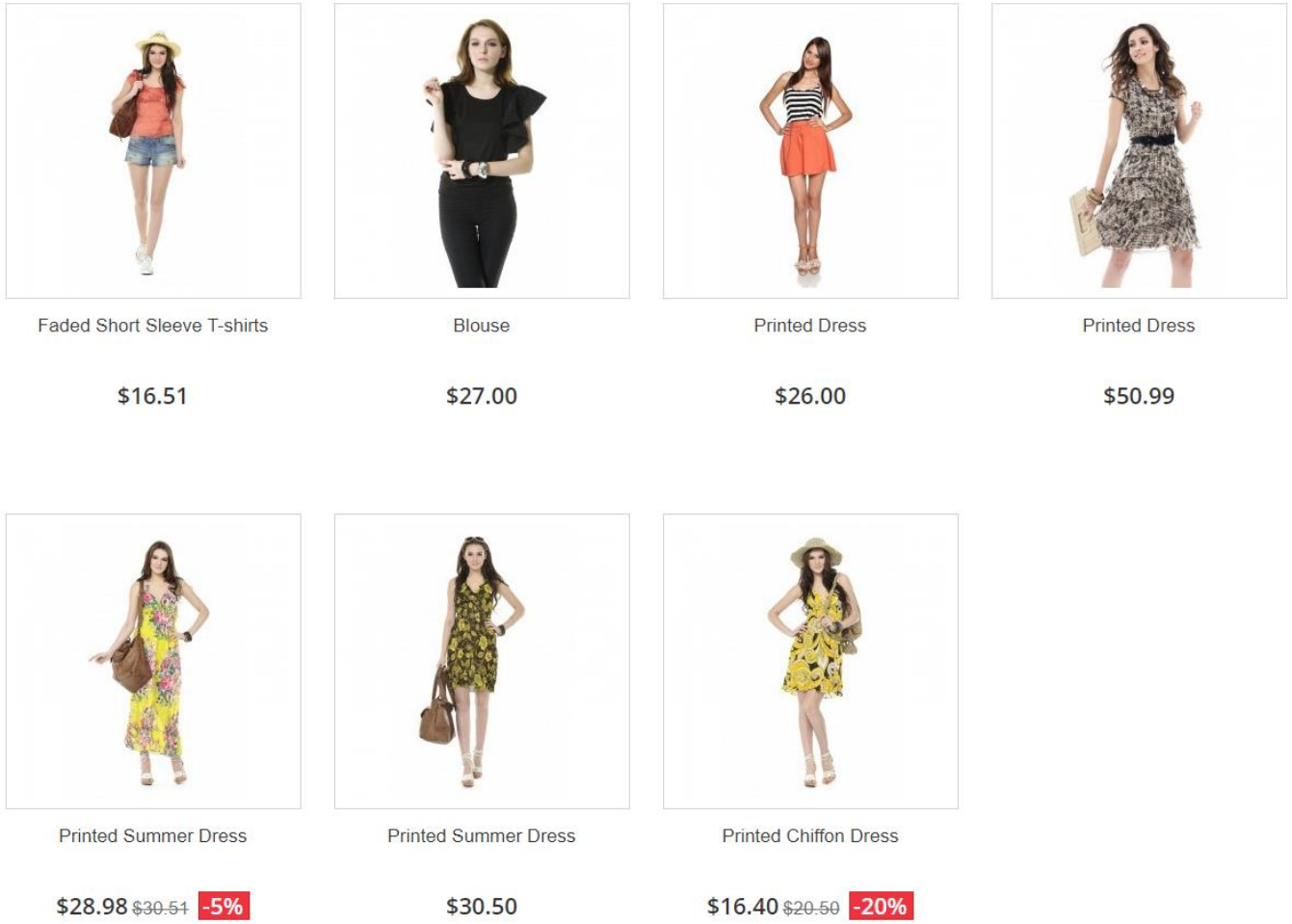
```
Scenario: User logs in
    Given User is one the homepage
    When User logs in with "temp@pp.pp" email and "temp123" password
    Then User should see login notification
```

ADNOTACJE W JUNIT

- **@Test** – (dla metody) oznacza ze metoda public void jest testem i moze byc wykonywana jako test.
- **@Before / @After** – (dla metody) oznacza że dana metoda będzie wykonywana przed/po każdym teście (każdej metodzie z adnotacją @Test).
- **@BeforeClass / @AfterClass** – (dla metody) oznacza że dana metoda będzie wykonywana raz przed/po wykonaniu wszystkich testów z danej klasy gdzie znajdują sie testy (metody z adnotacją @Test), musi byc static.
- **@Ignore** – (dla metody/klasy testowej) oznacza że dany test lub testy nie będą wykonywane.

OPERACJE NA LIŚCIE

- Sprawdzanie wartości elementów menu, listy, paragrafów
- Klikanie wszystkich elementów z menu/linków w celu walidacji poprawności działania linków (broken links)
- Weryfikacja czy pojawił się element na stronie pośród kilku podobnych (pozycja losowa)



WAITY - DLACZEGO PRĘDZEJ CZY PÓŹNIEJ BĘDĄ POTRZEBNE

IMPLICIT WAIT

- prosta i szybka implementacja
- wywoływane raz dla WebDrivera
- aplikuje się tylko do metody findElement
- ustawienie globalne - jedna wartość timeoutu dla wszystkich szukanych elementów
- ograniczone zastosowanie
- rzuca NoSuchElementException

```
driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
```

EXPLICIT WAIT

- szerokie zastosowanie i dużo możliwości
- warunek, na który czekamy, może być właściwie dowolny
- nie definiujemy go dla całego WebDrivera na czas wykonywania testu - robimy to tam, gdzie potrzebujemy
- rzuca TimeoutException
- pomocnicza klasa ExpectedConditions

```
WebDriverWait wait = new WebDriverWait(driver, 5);
wait.until(warunek);
```

EXPECTED CONDITIONS

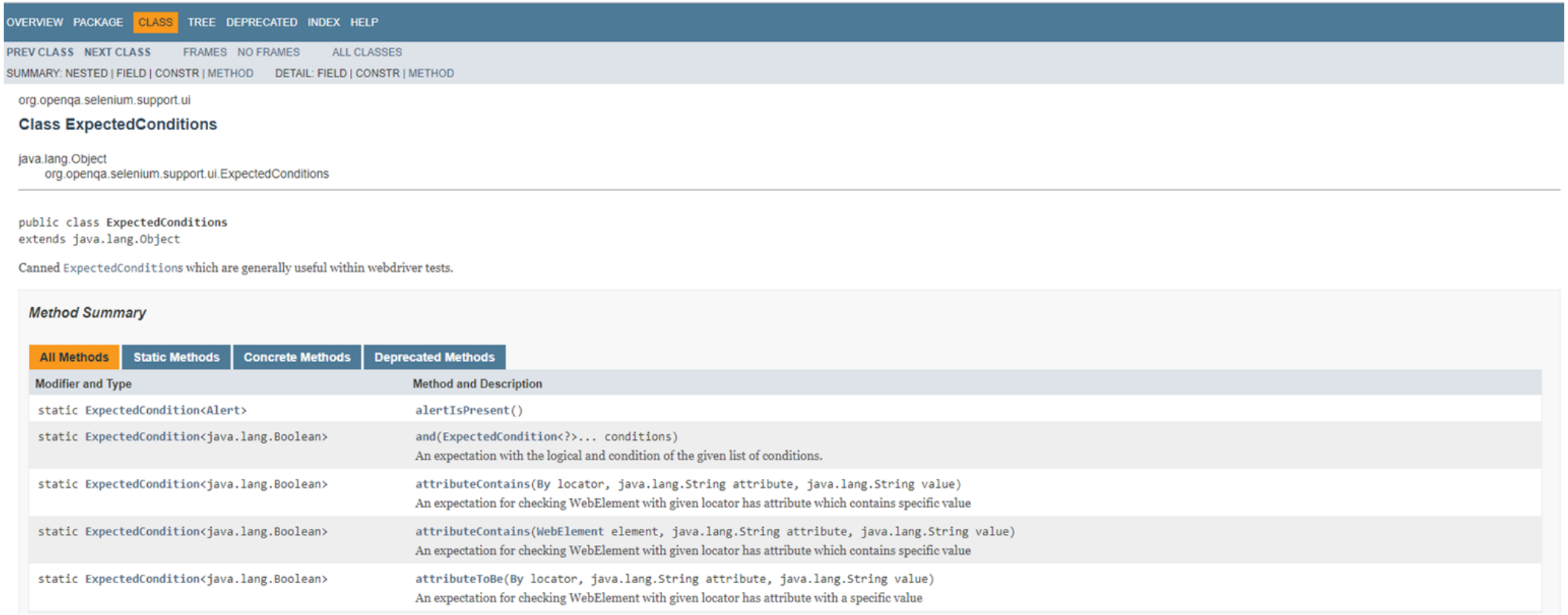
```
wait.until(ExpectedConditions.presenceOfElementLocated(
    By.cssSelector("#identifierId")));
```

```
wait.until(ExpectedConditions.attributeContains(By.cssSelector("#identifierId"),
    "class", "RRP0oc");
```

```
wait.until(ExpectedConditions.invisibilityOfElementLocated(
    By.cssSelector("#profileIdentifier")));
```

EXPECTED CONDITIONS

- https://seleniumhq.github.io/selenium/docs/api/java/org/openqa/selenium/support/ui/ExpectedConditions.html



OVERVIEW PACKAGE CLASS TREE DEPRECATED INDEX HELP

PREV CLASS NEXT CLASS FRAMES NO FRAMES ALL CLASSES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

org.openqa.selenium.support.ui

Class ExpectedConditions

java.lang.Object
  org.openqa.selenium.support.ui.ExpectedConditions

```
public class ExpectedConditions
extends java.lang.Object
```

Canned `ExpectedConditions` which are generally useful within webdriver tests.

Method Summary

All Methods | Static Methods | Concrete Methods | Deprecated Methods

Modifier and Type	Method and Description
`static ExpectedCondition<Alert>`	`alertIsPresent()`
`static ExpectedCondition<java.lang.Boolean>`	`and(ExpectedCondition<?>... conditions)` An expectation with the logical and condition of the given list of conditions.
`static ExpectedCondition<java.lang.Boolean>`	`attributeContains(By locator, java.lang.String attribute, java.lang.String value)` An expectation for checking WebElement with given locator has attribute which contains specific value
`static ExpectedCondition<java.lang.Boolean>`	`attributeContains(WebElement element, java.lang.String attribute, java.lang.String value)` An expectation for checking WebElement with given locator has attribute which contains specific value
`static ExpectedCondition<java.lang.Boolean>`	`attributeToBe(By locator, java.lang.String attribute, java.lang.String value)` An expectation for checking WebElement with given locator has attribute with a specific value

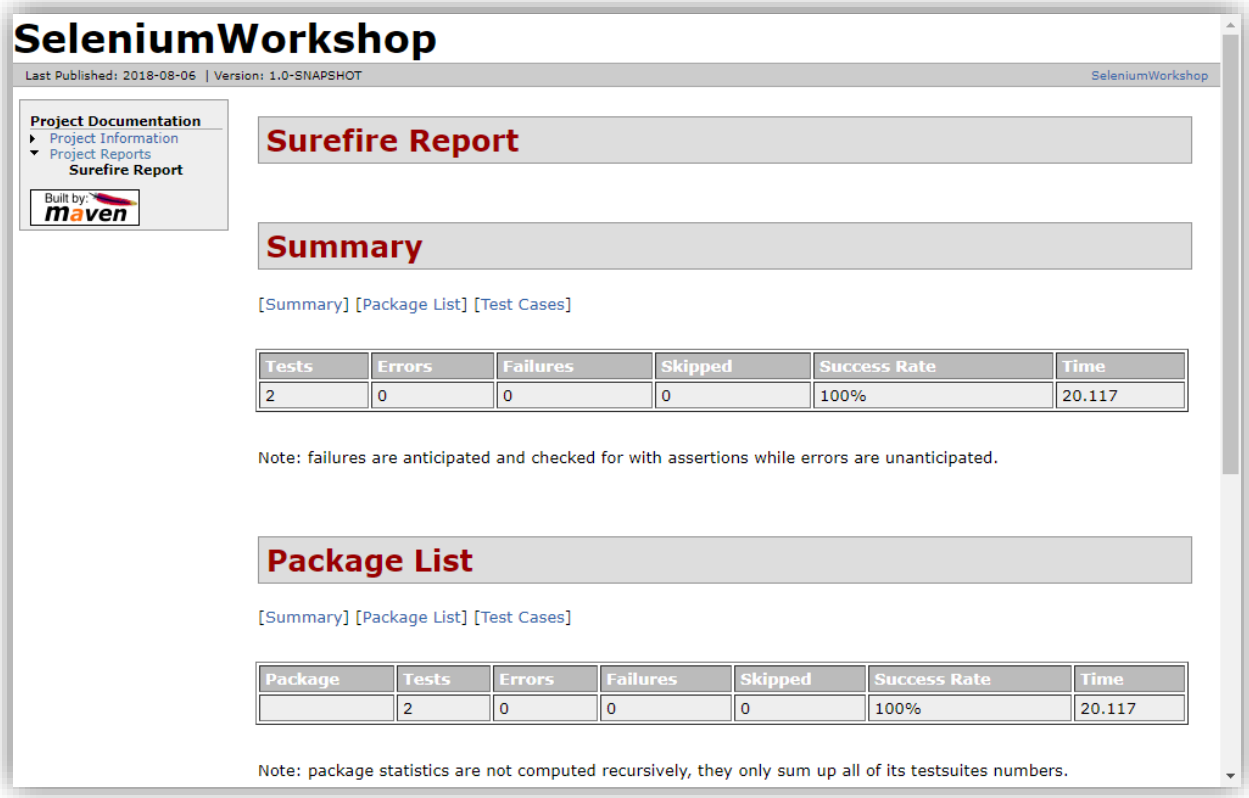
CO WARTO ZAIMPLEMENTOWAĆ SAMEMU?

```
public WebElement WaitForElementAndReturn(By locator, int timeoutInSeconds){
    WebDriverWait wait = new WebDriverWait(driver, timeoutInSeconds);
    wait.until(ExpectedConditions.presenceOfElementLocated(locator));
    return driver.findElement(locator);
}
```

KONFIGURACJA RAPORTU

Surefire report:

- Łatwy w implementacji – wymage wtyczki org.apache.maven.plugins w zaleznosciach
- Konwertuje pliki TEST-*.xml generowane z frameworków testowych do raportu html
- Generuje dodatkowe informacje o projekcie
- Nie zachwyca wizualnie



DZIĘKUJEMY