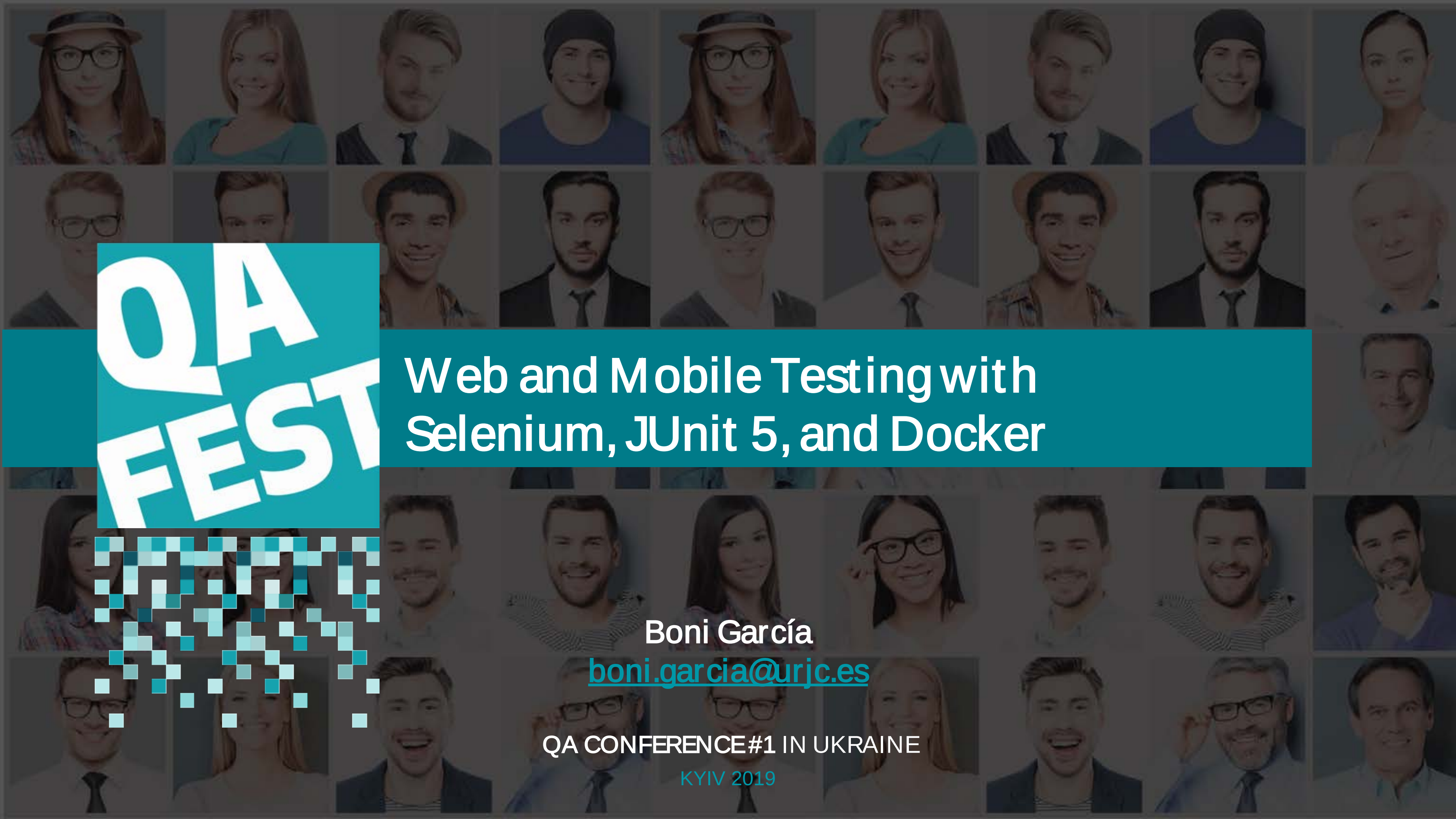


Web and Mobile Testing with Selenium, JUnit 5, and Docker

Boni García
boni.garcia@urjc.es

QA CONFERENCE #1 IN UKRAINE

KYIV 2019

Boni García

- Assistant Professor at King Juan Carlos University (URJC) in Spain
- Author of 35+ research papers in different journals, magazines, international conferences, and the book Mastering Software Testing with JUnit 5
- Maintainer of different open source projects, such as WebDriverManager, Selenium-Jupiter, or DualSub

http://bonigarcia.github.io/

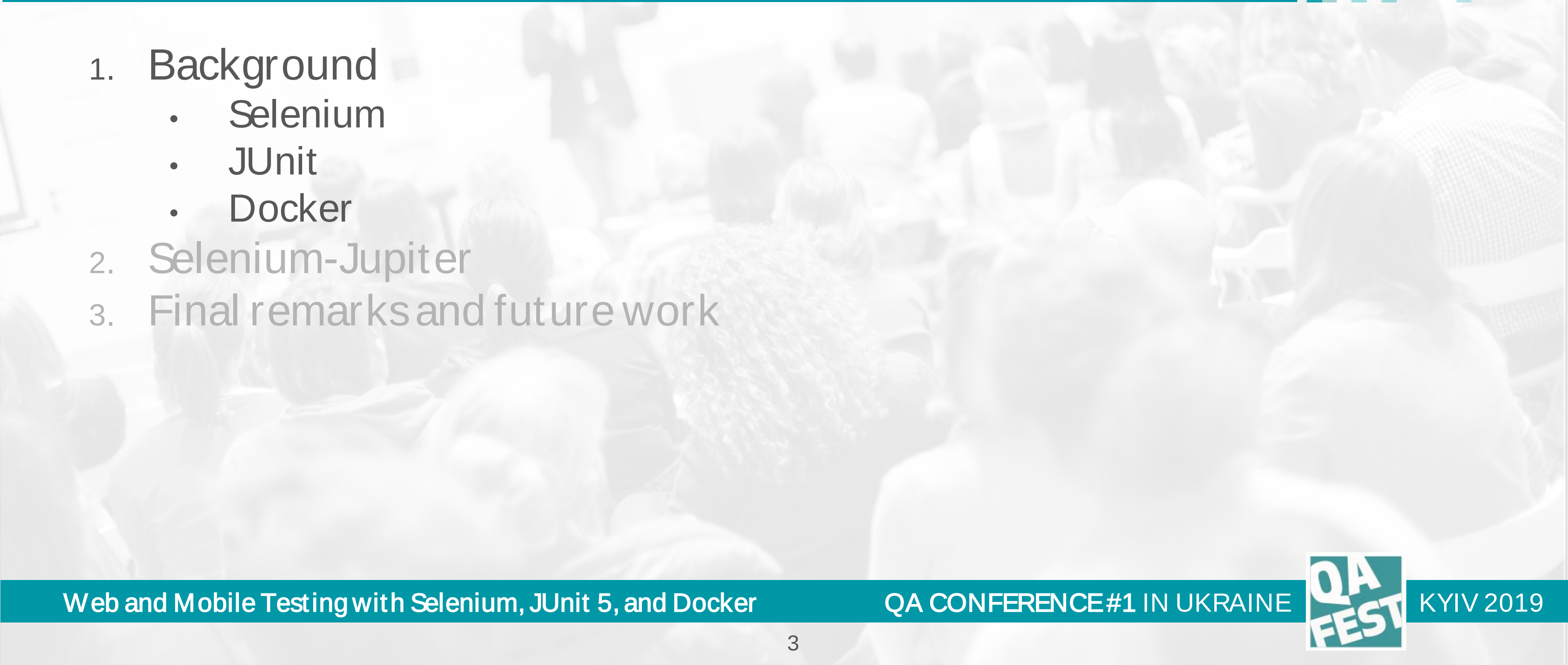
Boni García

Mastering Software Testing with JUnit 5

Comprehensive guide to develop high quality Java applications

Packt>

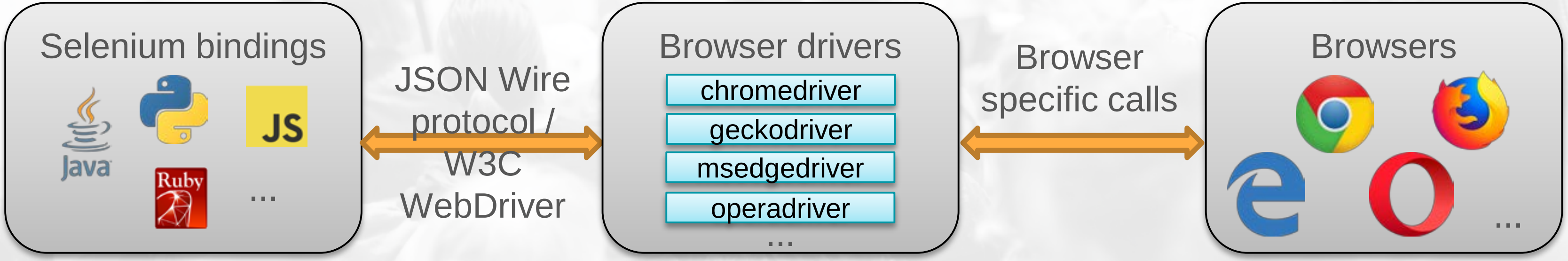
Table of contents



1. Background
   - Selenium
   - JUnit
   - Docker
2. Selenium-Jupiter
3. Final remarks and future work

1. Background - Selenium

- **Selenium** is a family of projects for automated testing with browsers
  - **WebDriver** allows to control web browsers programmatically



Selenium bindings

Java

Ruby

JS

...

JSON Wire protocol / W3C WebDriver

Browser drivers

chromedriver

geckodriver

msedgedriver

operadriver

...

Browser specific calls

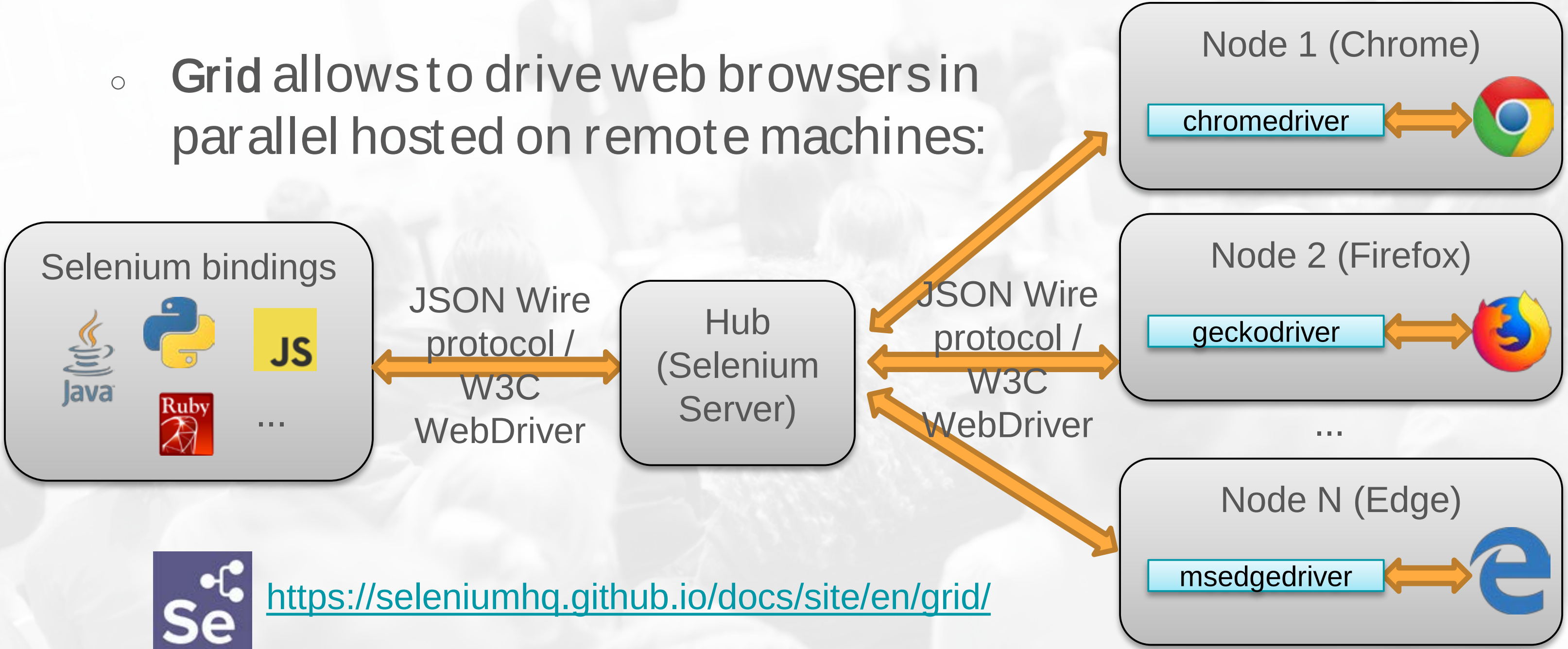
Browsers

...

https://seleniumhq.github.io/docs/site/en/webdriver/

1. Background - Selenium

- **Grid** allows to drive web browsers in parallel hosted on remote machines:


Selenium bindings

Java

JS

Ruby

...

JSON Wire protocol / W3C WebDriver

Hub (Selenium Server)

JSON Wire protocol / W3C WebDriver

Node 1 (Chrome)

chromedriver

Node 2 (Firefox)

geckodriver

...

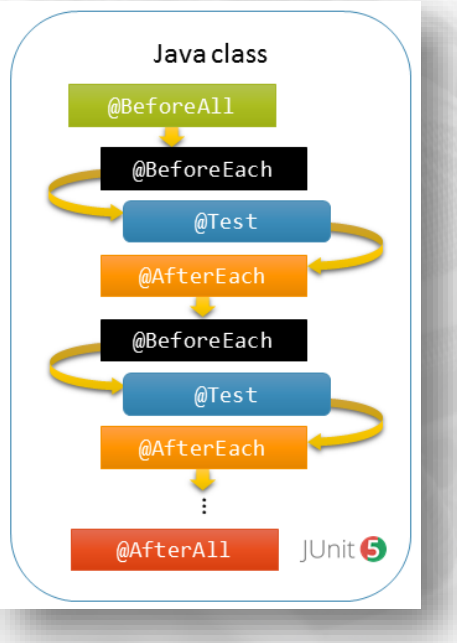
Node N (Edge)

msedgedriver

https://seleniumhq.github.io/docs/site/en/grid/

1. Background - JUnit

- **JUnit** is the most popular testing framework for Java and can be used to implement different types of tests (unit, integration, end-to-end, …)
- **JUnit 5** (first GA released on September 2017) provides a brand-new programming an extension model called **Jupiter**

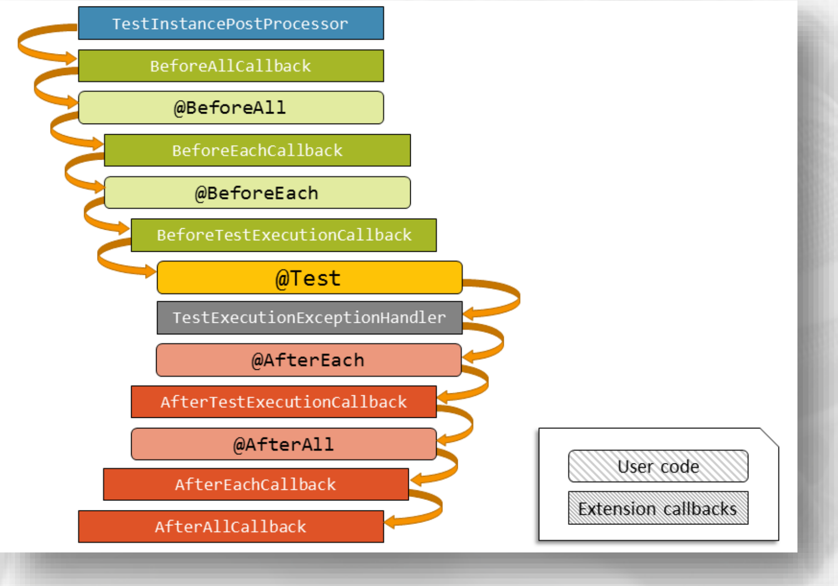


https://junit.org/junit5/docs/current/user-guide/

1. Background - JUnit

- The **extension model** of Jupiter allows to add custom features to the programming model:
  - Dependency injection in test methods and constructors
  - Custom logic in the test lifecycle
  - Test templates

Very convenient for Selenium!



TestInstancePostProcessor

BeforeAllCallback

@BeforeAll

BeforeEachCallback

@BeforeEach

BeforeTestExecutionCallback

@Test

TestExecutionExceptionHandler

@AfterEach

AfterTestExecutionCallback

@AfterAll

AfterEachCallback

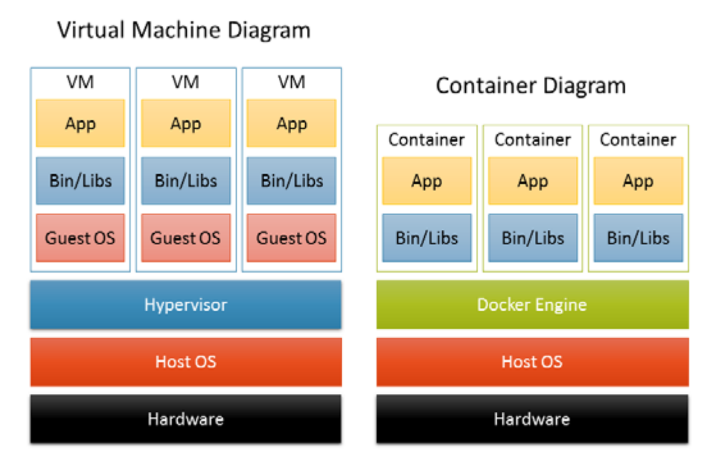
AfterAllCallback

User code

Extension callbacks

1. Background - Docker

- **Docker** is a software technology which allows to pack and run any application as a lightweight and portable **container**
- The Docker platform has two main components: the Docker Engine, to create and execute containers; and the Docker Hub (https://hub.docker.com/), a cloud service for distributing containers



https://www.docker.com/

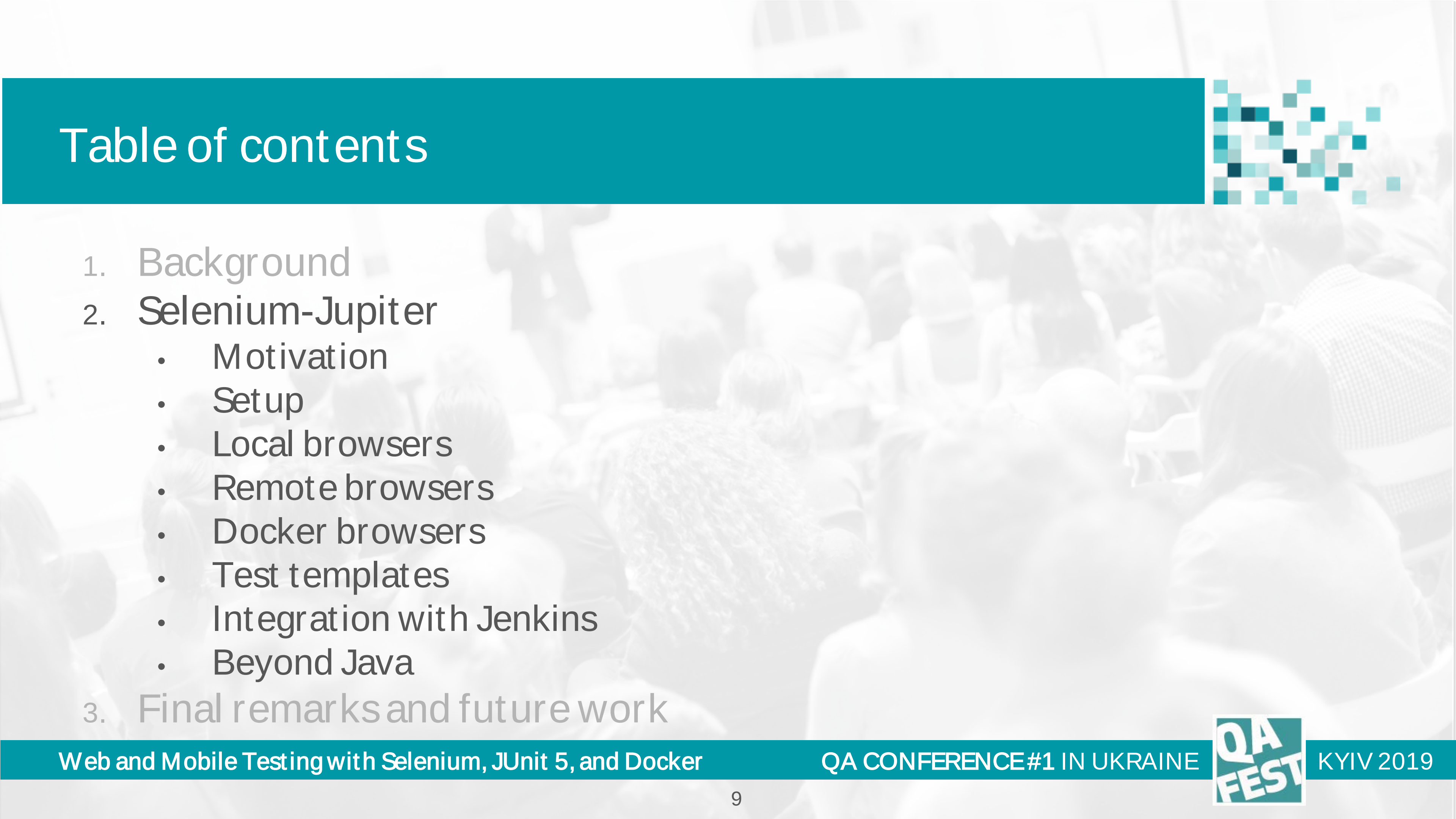


Table of contents

1. Background
2. Selenium-Jupiter
   - Motivation
   - Setup
   - Local browsers
   - Remote browsers
   - Docker browsers
   - Test templates
   - Integration with Jenkins
   - Beyond Java
3. Final remarks and future work

2. Selenium-Jupiter - Motivation

- **Selenium-Jupiter** is a JUnit 5 extension aimed to ease the use of Selenium and Appium from Java tests

✓ **Clean** test code (reduced boilerplate)

✓ Effortless **Docker** integration (web browsers and Android devices)

✓ **Advanced** features for tests

https://bonigarcia.github.io/selenium-jupiter/

2. Selenium-Jupiter - Setup

- **Selenium-Jupiter** can be included in a Java project as follows:

```
<dependency>
   <groupId>io.github.bonigarcia</groupId>
   <artifactId>selenium-jupiter</artifactId>
   <version>3.3.1</version>
   <scope>test</scope>
</dependency>
```

Using the latest version is always recommended!

```
dependencies {
   testCompile("io.github.bonigarcia:selenium-jupiter:3.3.1")
}
```

2. Selenium-Jupiter - Setup

Fork me on GitHub

- Source code: https://github.com/bonigarcia/selenium-jupiter
- Documentation: https://bonigarcia.github.io/selenium-jupiter/
- Examples: https://github.com/bonigarcia/selenium-jupiter-examples

Requirements to run these examples:

- Java
- Maven/Gradle (alternatively some IDE)
- Docker Engine
- Linux (only required when running Android in Docker)

2. Selenium-Jupiter - Local browsers

- JUnit 4 and Selenium VS JUnit 5 and Selenium-Jupiter:

2. Selenium-Jupiter - Local browsers

- Selenium-Jupiter uses JUnit 5's **dependency injection**

Valid types: `ChromeDriver, FirefoxDriver, OperaDriver, SafariDriver, EdgeDriver, InternetExplorerDriver, HtmlUnitDriver, PhantomJSDriver, AppiumDriver, SelenideDriver`

```java
@ExtendWith(SeleniumExtension.class)
class SeleniumJupiterTest {

    @Test
    void test(ChromeDriver chromeDriver) {
        // Use Chrome in this test
    }

}
```

2. Selenium-Jupiter - Local browsers

- Seamless integration with **Selenide** (fluent API for Selenium in Java)

Selenide
CONCISE UI TESTS IN JAVA

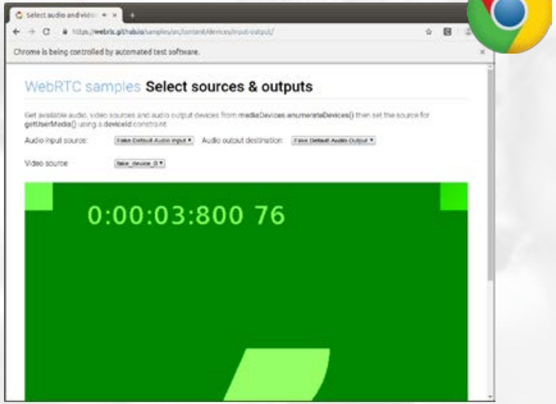
https://selenide.org/

```
@ExtendWith(SeleniumExtension.class)
class SelenideDefaultTest {

    @Test
    void testWithSelenideAndChrome(SelenideDriver driver) {
        driver.open(
            "https://bonigarcia.github.io/selenium-jupiter/");
        SelenideElement about = driver.$(linkText("About"));
        about.shouldBe(visible);
        about.click();
    }

}
```

2. Selenium-Jupiter - Local browsers

- Use case: **WebRTC** applications (real-time communications using web browsers)
    - We need to specify **options** for browsers:

WebRTC



Select audio and vide...

https://webrtc.github.io/samples/src/content/devices/input-output/

Chrome is being controlled by automated test software.

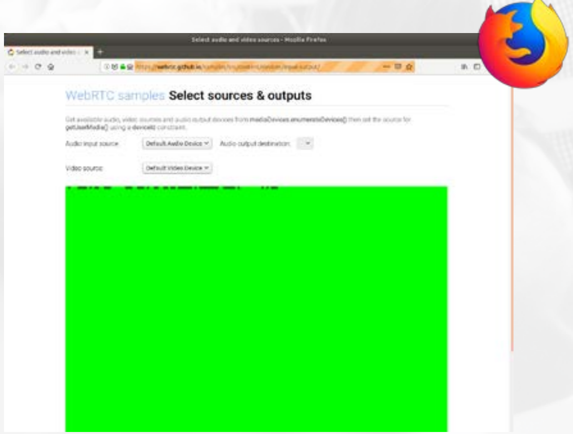
WebRTC samples **Select sources & outputs**

Get available audio, video sources and audio output devices from **mediaDevices.enumerateDevices()** then set the source for **getUserMedia()** using a **deviceId** constraint.

Audio input source: Fake Default Audio Input ▾ Audio output destination: Fake Default Audio Output ▾

Video source: fake_device_0 ▾

0:00:03:800 76



Select audio and video sources - Mozilla Firefox

Select audio and video ...

https://webrtc.github.io/samples/src/content/devices/input-output/

WebRTC samples **Select sources & outputs**

Get available audio, video sources and audio output devices from **mediaDevices.enumerateDevices()** then set the source for **getUserMedia()** using a **deviceId** constraint.

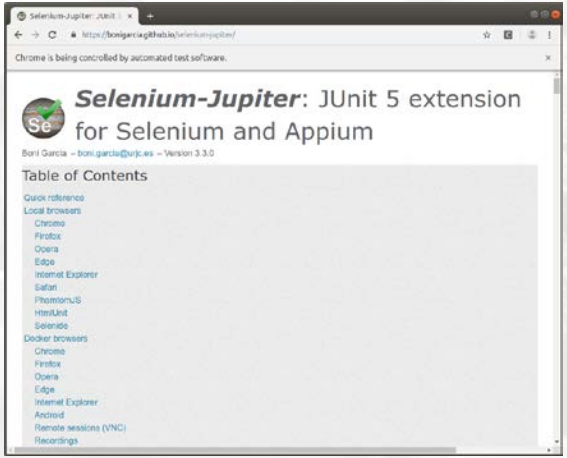
Audio input source: Default Audio Device ▾ Audio output destination: ▾

Video source: Default Video Device ▾

2. Selenium-Jupiter - Local browsers

- Use case: reuse same browser by different tests
  - ○ Convenient for ordered tests (JUnit 5 new feature)

1



Chrome is being controlled by automated test software.

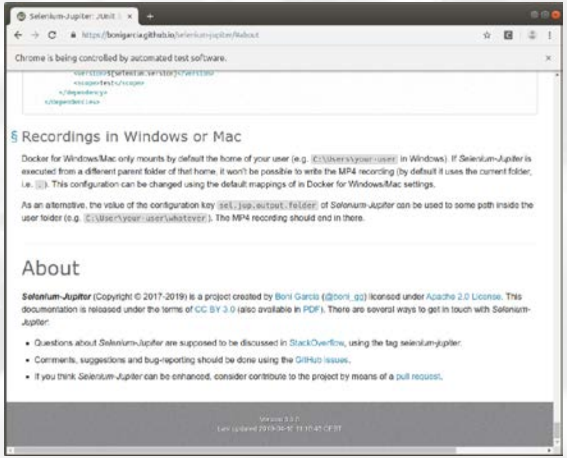
Selenium-Jupiter: JUnit 5 extension for Selenium and Appium

Boni García – boni.garcia@urjc.es – Version 3.3.0

Table of Contents

- Quick reference
- Local browsers
  - Chrome
  - Firefox
  - Opera
  - Edge
  - Internet Explorer
  - Safari
  - PhantomJS
  - HtmlUnit
  - Selenide
- Docker browsers
  - Chrome
  - Firefox
  - Opera
  - Edge
  - Internet Explorer
  - Android
  - Remote sessions (VNC)
  - Recordings

2



Chrome is being controlled by automated test software.

```
            <version>${selenium.version}</version>
            <scope>test</scope>
        </dependency>
    </dependencies>
```

§ Recordings in Windows or Mac

Docker for Windows/Mac only mounts by default the home of your user (e.g. `C:\Users\your-user` in Windows). If *Selenium-Jupiter* is executed from a different parent folder of that home, it won't be possible to write the MP4 recording (by default it uses the current folder, i.e. `.`). This configuration can be changed using the default mappings of in Docker for Windows/Mac settings.

As an alternative, the value of the configuration key `sel.jup.output.folder` of Selenium-Jupiter can be used to some path inside the user folder (e.g. `C:\User\your-user\whatever`). The MP4 recording should end in there.

About

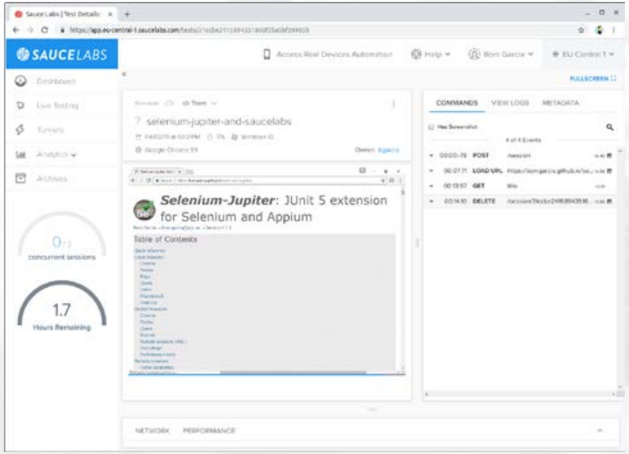
Selenium-Jupiter (Copyright © 2017-2019) is a project created by Boni Garcia (@boni_gg) licensed under Apache 2.0 License. This documentation is released under the terms of CC BY 3.0 (also available in PDF). There are several ways to get in touch with *Selenium-Jupiter*:

- Questions about *Selenium-Jupiter* are supposed to be discussed in StackOverflow, using the tag selenium-jupiter.
- Comments, suggestions and bug-reporting should be done using the GitHub issues.
- If you think *Selenium-Jupiter* can be enhanced, consider contribute to the project by means of a pull request.

Version 3.3.0
Last updated 2019-04-10 11:10:41 CEST

2. Selenium-Jupiter - Remote browsers

- Selenium-Jupiter provides the annotations `@DriverUrl` and `@DriverCapabilities` to control remote browsers and mobiles, e.g.:



? selenium-jupiter-and-saucelabs

Google Chrome 59

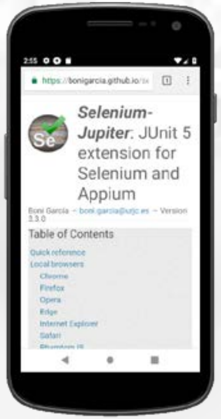
Selenium-Jupiter: JUnit 5 extension for Selenium and Appium

Table of Contents

```
POST      /session
LOAD URL  https://bonigarcia.github.io/se...
GET       /title
DELETE    /session/...
```

SAUCELABS

https://saucelabs.com/



https://bonigarcia.github.io/se

Selenium-Jupiter: JUnit 5 extension for Selenium and Appium

Boni Garcia – boni.garcia@urjc.es – Version 3.3.0

Table of Contents

Quick reference
Local browsers
Chrome
Firefox
Opera
Edge
Internet Explorer
Safari

appium

http://appium.io/

2. Selenium-Jupiter - Docker browsers

- Selenium-Jupiter provides seamless integration with Docker using the annotation `@DockerBrowser`:
  - Chrome, Firefox, and Opera:
    - Docker images for stable versions are maintained by Aerokube
    - Beta and unstable (Chrome and Firefox) are maintained by ElasTest
  - Edge and Internet Explorer:
    - Due to license, these Docker images are not hosted in Docker Hub
    - It can be built following a tutorial provided by Aerokube
  - Android devices:
    - Docker images for Android (docker-android project) by Budi Utomo

2. Selenium-Jupiter - Docker browsers

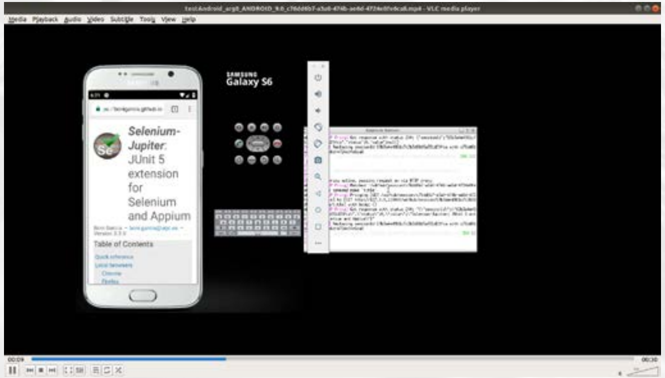
```
@ExtendWith(SeleniumExtension.class)
class DockerBasicTest {

    @Test
    void testFirefoxBeta(
            @DockerBrowser(type = FIREFOX, version = "beta") RemoteWebDriver driver) {
        driver.get("https://bonigarcia.github.io/selenium-jupiter/");
        assertThat(driver.getTitle(),
                containsString("JUnit 5 extensio      Selenium"));
    }

}
```

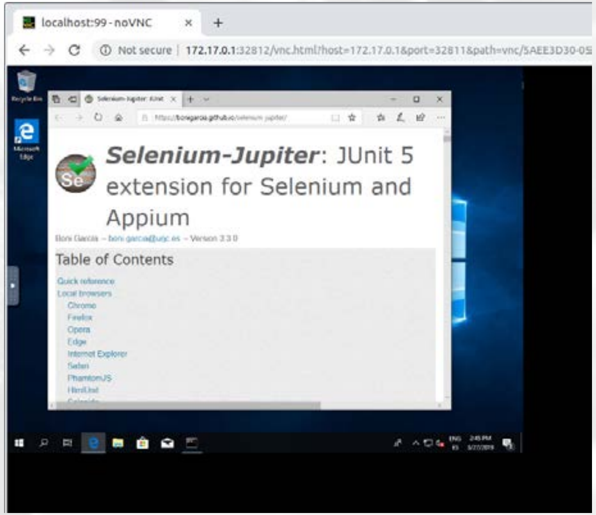
Supported browser types are: *CHROME*, *FIREFOX*, *OPERA*, *EDGE* , *IEXPLORER* and *ANDROID*

If *version* is not specified, the latest container version in Docker Hub is pulled. This parameter allows fixed versions and also the special values: *latest*, *latest-**, *beta*, and *unstable*

2. Selenium-Jupiter - Docker browsers

- The use of Docker enables a rich number of features:
    - ○ Remote session access with **VNC**
    - ○ Session **recordings**
    - ○ **Performance** tests




localhost:99 - noVNC
Not secure | 172.17.0.1:32812/vnc.html?host=172.17.0.1&port=32811&path=vnc/5AEE3D30-05

Selenium-Jupiter: JUnit 5 extension for Selenium and Appium

Boni García – boni.garcia@urjc.es – Version 3.3.0

Table of Contents

- Quick reference
- Local browsers
    - Chrome
    - Firefox
    - Opera
    - Edge
    - Internet Explorer
    - Safari
    - PhantomJS
    - HtmlUnit

2. Selenium-Jupiter - Docker browsers

- The possible **Android** setup options are the following:

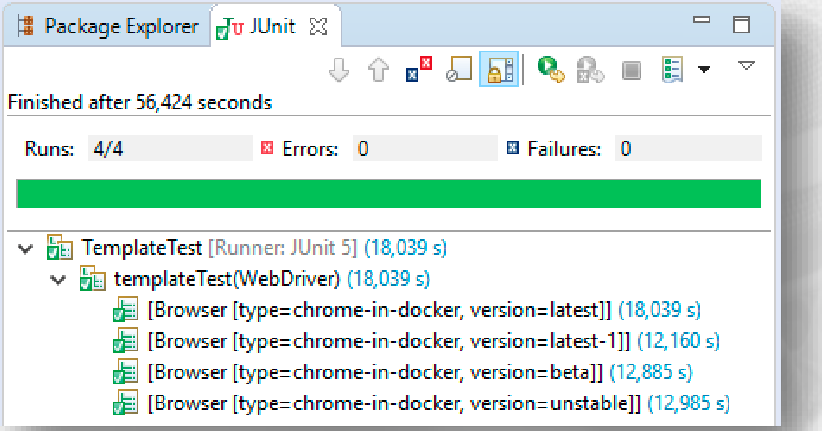
Type	Device name
Phone	Samsung Galaxy S6
Phone	Nexus 4
Phone	Nexus 5
Phone	Nexus One
Phone	Nexus S
Tablet	Nexus 7

Android version	API level	Browser name
5.0.1	21	browser
5.1.1	22	browser
6.0	23	chrome
7.0	24	chrome
7.1.1	25	chrome
8.0	26	chrome
8.1	27	chrome
9.0	28	chrome

2. Selenium-Jupiter - Test templates

- Selenium-Jupiter use the JUnit 5's support for **test templates**

```java
@ExtendWith(SeleniumExtension.class)
public class TemplateTest {

    @TestTemplate
    void templateTest(WebDriver driver) {
        // test
    }

}
```

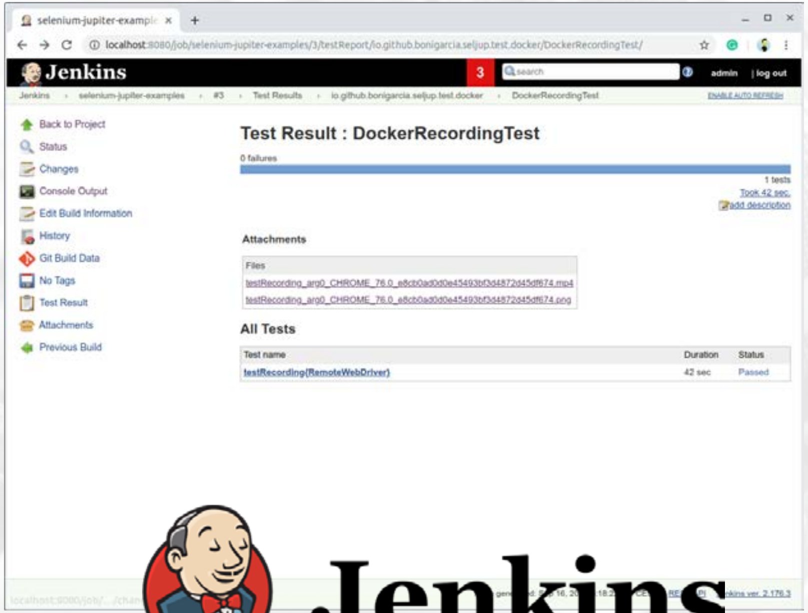


```
Finished after 56,424 seconds

Runs:  4/4        Errors:  0        Failures:  0

TemplateTest [Runner: JUnit 5] (18,039 s)
  templateTest(WebDriver) (18,039 s)
    [Browser [type=chrome-in-docker, version=latest]] (18,039 s)
    [Browser [type=chrome-in-docker, version=latest-1]] (12,160 s)
    [Browser [type=chrome-in-docker, version=beta]] (12,885 s)
    [Browser [type=chrome-in-docker, version=unstable]] (12,985 s)
```

2. Selenium-Jupiter - Integration with Jenkins

- Seamless integration with Jenkins through the Jenkins attachment plugin
- It allows to attach output files in tests (e.g. PNG screenshots and MP4 recordings) in the Jenkins GUI
- For example:

```
 $ mvn clean test -Dtest=DockerRecordingTest \
    -Dsel.jup.recording=true \
    -Dsel.jup.screenshot.at.the.end.of.tests=true \
    -Dsel.jup.screenshot.format=png \
    -Dsel.jup.output.folder=surefire-reports
```


selenium-jupiter-example
localhost:8080/job/selenium-jupiter-examples/3/testReport/io.github.bonigarcia.seljup.test.docker/DockerRecordingTest/
Jenkins
3
search
admin
| log out
Jenkins › selenium-jupiter-examples › #3 › Test Results › io.github.bonigarcia.seljup.test.docker › DockerRecordingTest
ENABLE AUTO REFRESH
Back to Project
Status
Changes
Console Output
Edit Build Information
History
Git Build Data
No Tags
Test Result
Attachments
Previous Build

Test Result : DockerRecordingTest

0 failures

1 tests

Took 42 sec.

add description

Attachments

Files
testRecording_arg0_CHROME_76.0_e8cb0ad0d0e45493bf3d4872d45df674.mp4
testRecording_arg0_CHROME_76.0_e8cb0ad0d0e45493bf3d4872d45df674.png

All Tests

Test name	Duration	Status
testRecording(RemoteWebDriver)	42 sec	Passed

2. Selenium-Jupiter - Beyond Java

- Selenium-Jupiter can be also used:

1. As **CLI** (Command Line Interface) tool:

```
$ java -jar selenium-jupiter-3.3.1-fat.jar chrome unstable
[INFO] Using Selenium-Jupiter to execute chrome unstable in Docker
...
```

Selenium-Jupiter allows to control Docker browsers through VNC (manual testing)

2. As a **server** (using a REST-like API):

```
$ java -jar webdrivermanager-3.3.1-fat.jar server
[INFO] Selenium-Jupiter server listening on http://localhost:4042/wd/hub
```

Selenium-Jupiter becomes into a Selenium Server (Hub)

Table of contents

1. Background
2. Selenium-Jupiter
3. Final remarks and future work

3. Final remarks and future work

- Selenium-Jupiter has another features such as:
  - Configurable screenshots at the end of test (as PNG image or Base64)
  - Integration with Genymotion (cloud provider for Android devices)
  - Generic driver (configurable type of browser)
  - Mapping volumes in Docker containers
  - Access to Docker client to manage custom containers
- Selenium-Jupiter is in constant development. Its roadmap includes:
  - Implement a browser console (JavaScript log) gathering mechanism
  - Improve test template support (e.g. specifying options)
  - Improve scalability for performance tests (e.g. using Kubernetes)

Web and Mobile Testing with Selenium, JUnit 5, and Docker

Thank you very much!

Boni García
boni.garcia@urjc.es

QA CONFERENCE #1 IN UKRAINE

KYIV 2019