D

B

5.3.1, 04-11-2022

Table of Contents

Chapter 1. Motivation

D . - A (. ., C , , ,) (. ., , , , C#, D

[redacted]

D *driver* . (C), ( ), (, A 3C D *N Wire Prot* .

[redacted]

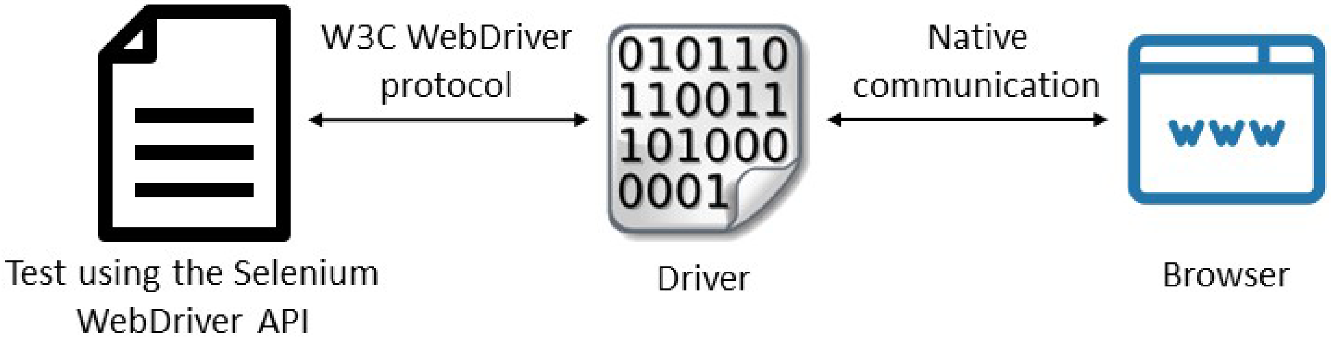


Figure 1. Se river Architecture

[redacted]

, *management process* :

D - . , . ., C), , , ,), (, 32 64), , . C , ,).

2. . , , , PATH . , . , :

[redacted]

```
System.setProperty("webdriver.chrome.driver", "/path/to/chromedriver");
System.setProperty("webdriver.gecko.driver", "/path/to/geckodriver");
System.setProperty("webdriver.edge.driver", "/path/to/msedgedriver");
System.setProperty("webdriver.opera.driver", "/path/to/operadriver");
System.setProperty("webdriver.ie.driver", "C:/path/to/IEDriverServer.exe");
```

3. . ,

.

(. ., *evergreen*), , -

, D C

, : "*this version of chromedriver only supports chrome version* [redacted] ,

).

What is WebDriverManager?

D - (. .,

, ,) D (. .,

, , , .) . ,

5, D ,

, D (

`ChromeDriver`, `FirefoxDriver`, `EdgeDriver`, .), D

, .

Chapter 2. Setup

D (
). *build tool* ( ) D
. , (`test` ,
):

```
<dependency>
    <groupId>io.github.bonigarcia</groupId>
    <artifactId>webdrivermanager</artifactId>
    <version>5.3.1</version>
    <scope>test</scope>
</dependency>
```

, D (,):

```
dependencies {
    testImplementation("io.github.bonigarcia:webdrivermanager:5.3.1")
}
```

Chapter 3. Features

D A WebDriverManager (
io.github.bonigarcia.wdm). *managers*, . .,
.

3.1. Driver Management

D .
, D A (. ., chromedriver()
C) setup(). 5,
D , D , A ( ). ,
D (@BeforeAll). ,
( ) D C .

```java
import static org.assertj.core.api.Assertions.assertThat;

import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

import io.github.bonigarcia.wdm.WebDriverManager;

class ChromeTest {

    WebDriver driver;

    @BeforeAll
    static void setupClass() {
        WebDriverManager.chromedriver().setup();
    }

    @BeforeEach
    void setupTest() {
        driver = new ChromeDriver();
    }

    @AfterEach
    void teardown() {
        driver.quit();
    }

    @Test
    void test() {
        // Exercise
        driver.get("https://bonigarcia.dev/selenium-webdriver-java/");
        String title = driver.getTitle();

        // Verify
        assertThat(title).contains("Selenium WebDriver");
    }

}
```

D *managers* C , , , C ,
.

```
WebDriverManager.chromedriver().setup();
WebDriverManager.firefoxdriver().setup();
WebDriverManager.edgedriver().setup();
WebDriverManager.operadriver().setup();
WebDriverManager.chromiumdriver().setup()
WebDriverManager.iedriver().setup();
```

A 5, D (`safaridriver()`).

D (-).
, D D
(D).

A
D ,
.
4 . A ,
D (
).

3.1.1. Resolution Algorithm

D *resolution algorithm* `setup()` .
A
D , 2021.

1. D . , D
   (. ., `google-chrome --version`
2. , .
   C ,
   ( )
   ,
   89, ,

([redacted])

C [redacted] [redacted] , D

3. , D

(~/.cache/selenium). .

[redacted] , (. .,

webdriver.chrome.driver C).

, . .,

. (. .,), D

(resolution.properties

. *time-to-live* ( ).

1 1 . ,

1 , 1 .

( )

(. .,

).

3.1.2. Generic Manager

D *generic manager*, . .,

(C [redacted] , .). ,

getInstance()

:

- getInstance(Class<? extends WebDriver> webDriverClass): webDriverClass D ChromeDriver.class, FirefoxDriver.class, .
- getInstance(DriverManagerType driverManagerType): driverManagerType D .
- getInstance(String browserName): browserName String (. ., "Chrome", "Firefox", "Edge", "Opera", "Chromium", "Safari", "IExplorer").
- getInstance(): , wdm.defaultBrowser (C). D .

5 (ChromeDriver.class FirefoxDriver.class). A , D WebDriver (D B).

```
import static org.assertj.core.api.Assertions.assertThat;

import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.params.ParameterizedTest;
```

```java
import org.junit.jupiter.params.provider.ValueSource;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

import io.github.bonigarcia.wdm.WebDriverManager;

class GenericTest {

    WebDriver driver;

    @AfterEach
    void teardown() {
        driver.quit();
    }

    @ParameterizedTest
    @ValueSource(classes = { ChromeDriver.class, FirefoxDriver.class })
    void test(Class<? extends WebDriver> webDriverClass) {
        // Driver management and WebDriver instantiation
        driver = WebDriverManager.getInstance(webDriverClass).create();

        // Exercise
        driver.get("https://bonigarcia.dev/selenium-webdriver-java/");
        String title = driver.getTitle();

        // Verify
        assertThat(title).contains("Selenium WebDriver");
    }

}
```

3.1.3. Advanced Settings

D
, D A ( )
, , , ,
( ), , ,
. .
D A , :

[redacted]

- `getDownloadedDriverPath()`:
  D .
- `getDownloadedDriverVersion()`:
  D .
- `getDriverVersions()`: .
- `getDriverManagerType()`: (`enum`) .

D (. ., .), 5, . , getDownloadedDriverPath() getDownloadedDriverVersion() D 5 (. ., D).

3 [redacted]wser Finder

A 5, D . , getBrowserPath(). Optional<Path>, ( ) .

. , (. .,) A (- 5). ( ), .

```
import static org.assertj.core.api.Assertions.assertThat;
import static org.assertj.core.api.Assumptions.assumeThat;

import java.nio.file.Path;
import java.util.Optional;

import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.safari.SafariDriver;

import io.github.bonigarcia.wdm.WebDriverManager;

class SafariTest {

    WebDriver driver;

    @BeforeAll
    static void setupClass() {
        Optional<Path> browserPath = WebDriverManager.safaridriver()
                .getBrowserPath();
        assumeThat(browserPath).isPresent();
    }

    @BeforeEach
    void setupTest() {
        driver = new SafariDriver();
    }
```

```java
    @AfterEach
    void teardown() {
        driver.quit();
    }

    @Test
    void test() {
        // Exercise
        driver.get("https://bonigarcia.dev/selenium-webdriver-java/");
        String title = driver.getTitle();

        // Verify
        assertThat(title).contains("Selenium WebDriver");
    }
}
```

, D
.

3.3. WebDriver Builder

A 5, D WebDriver (. . ChromeDriver,
FirefoxDriver, .) D A .
, create(). .
D setup()
D . D
quit() WebDriver .

```java
import static org.assertj.core.api.Assertions.assertThat;

import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.openqa.selenium.WebDriver;

import io.github.bonigarcia.wdm.WebDriverManager;

class ChromeCreateTest {

    WebDriver driver;

    @BeforeEach
    void setupTest() {
        driver = WebDriverManager.chromedriver().create();
    }

    @AfterEach
    void teardown() {
        driver.quit();
    }

    @Test
    void test() {
        driver.get("https://bonigarcia.dev/selenium-webdriver-java/");
        assertThat(driver.getTitle()).contains("Selenium WebDriver");
    }

}
```

, D
WebDriver . ,
.
teardown .

D A WebDriver
, :

create(). WebDriver
. .

capabilities(): D Capabilities ( ).

remoteAddress(): RemoteWebDriver ,
(, , .).
wdm.remoteAddress ( ).
.

. , D (
):

```
import static org.assertj.core.api.Assertions.assertThat;

import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.grid.Main;

import io.github.bonigarcia.wdm.WebDriverManager;

class ChromeRemoteTest {

    WebDriver driver;

    @BeforeAll
    static void setupClass() {
        // Resolve driver for Selenium Grid
        WebDriverManager.chromedriver().setup();

        // Start Selenium Grid in standalone mode
        Main.main(new String[] { "standalone", "--port", "4444" });
    }

    @BeforeEach
    void setupTest() {
        driver = WebDriverManager.chromedriver()
                .remoteAddress("http://localhost:4444/wd/hub").create();
    }

    @AfterEach
    void teardown() {
        driver.quit();
    }

    @Test
    void test() {
        driver.get("https://bonigarcia.dev/selenium-webdriver-java/");
        assertThat(driver.getTitle()).contains("Selenium WebDriver");
    }

}
```

3.4. Browsers in Docker

A D 5
D . D
. , `browserInDocker()`
`create()` . , D
D , , `WebDriver` .
C D :

```
import static org.assertj.core.api.Assertions.assertThat;

import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.openqa.selenium.WebDriver;

import io.github.bonigarcia.wdm.WebDriverManager;

class DockerChromeTest {

    WebDriver driver;

    WebDriverManager wdm = WebDriverManager.chromedriver().browserInDocker();
    @BeforeEach
    void setupTest() {
        driver = wdm.create();
    }

    @AfterEach
    void teardown() {
        wdm.quit();
    }

    @Test
    void test() {
        driver.get("https://bonigarcia.dev/selenium-webdriver-java/");
        assertThat(driver.getTitle()).contains("Selenium WebDriver");
    }

}
```

D , `quit()`
D A ( ).

D D A
( ). ,
D D C (
, , , .

. , *Safari* D
. , ,
, (D)
.

, : C (. ., C
A). , `browserInDockerAndroid()`
C , .

( )
C .

3.4.1. Browser Versions

A D
D
( ). , *dockerized*
D - ,
.

, D `browserVersion()`

D A . String .
(. ., 91.0), :

- "latest" : ( ).
- "latest-N" : N latest-1 (. ., *latest version minus one*), , ( ).
- "beta": . C , D (A C).
- "dev":)).

```
import static io.github.bonigarcia.wdm.WebDriverManager.isDockerAvailable;

import static org.assertj.core.api.Assertions.assertThat;
import static org.assertj.core.api.Assumptions.assumeThat;

import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.openqa.selenium.WebDriver;

import io.github.bonigarcia.wdm.WebDriverManager;

class DockerChromeBetaTest {

    WebDriver driver;

    WebDriverManager wdm = WebDriverManager.chromedriver().browserInDocker()
            .browserVersion("beta");

    @BeforeEach
    void setupTest() {
        assumeThat(isDockerAvailable()).isTrue();
        driver = wdm.create();
    }

    @AfterEach
    void teardown() {
        wdm.quit();
    }

    @Test
    void test() {
        driver.get("https://bonigarcia.dev/selenium-webdriver-java/");
        assertThat(driver.getTitle()).contains("Selenium WebDriver");
    }

}
```

3.4.2. Remote Desktop

A D
. , D
enableVnc() .
, :

C (C), . D ,
C .

C, - - C . D , C D
C.

. D C
INFO . , ,
getDockerNoVncUrl().
( ).

```
import static org.assertj.core.api.Assertions.assertThat;

import java.net.URL;
import java.time.Duration;

import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.openqa.selenium.WebDriver;

import io.github.bonigarcia.wdm.WebDriverManager;

class DockerChromeVncTest {

    WebDriver driver;

    WebDriverManager wdm = WebDriverManager.chromedriver().browserInDocker()
            .enableVnc();

    @BeforeEach
    void setupTest() {
        driver = wdm.create();
    }

    @AfterEach
    void teardown() {
        wdm.quit();
    }
    @Test
    void test() throws Exception {
        driver.get("https://bonigarcia.dev/selenium-webdriver-java/");
        assertThat(driver.getTitle()).contains("Selenium WebDriver");

        // Verify URL for remote session
        URL noVncUrl = wdm.getDockerNoVncUrl();
        assertThat(noVncUrl).isNotNull();

        // Pause for manual inspection
        Thread.sleep(Duration.ofSeconds(60).toMillis());
    }

}
```

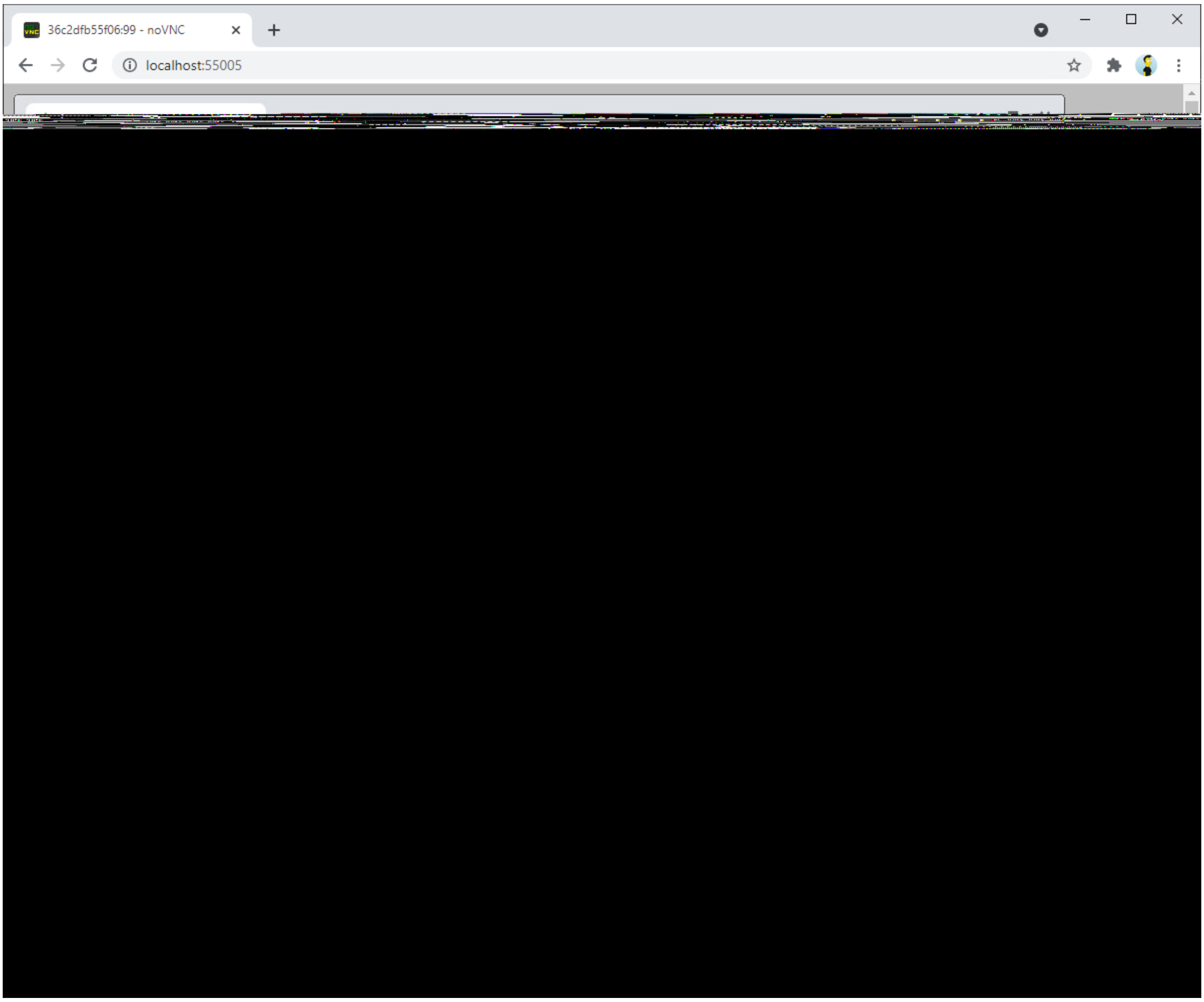


Figure 2. Example of noVNC session using Chrome in Docker

3.4.3. Recordings

. ,
`enableRecording()` D . ,

D D .
A , 4 (,
and the system timestamp, plus another
) (.
.

```
import static org.assertj.core.api.Assertions.assertThat;

import java.nio.file.Path;
import java.time.Duration;

import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;

import io.github.bonigarcia.wdm.WebDriverManager;

class DockerChromeRecordingTest {

    WebDriver driver;

    WebDriverManager wdm = WebDriverManager.chromedriver().browserInDocker()
            .enableRecording();

    @BeforeEach
    void setupTest() {
        driver = wdm.create();
    }

    @AfterEach
    void teardown() {
        wdm.quit();
    }

    @Test
    void test() throws Exception {
        driver.get("https://bonigarcia.dev/selenium-webdriver-java/");
        assertThat(driver.getTitle()).contains("Selenium WebDriver");

        // Pause to see the navigation in the recording
        Thread.sleep(Duration.ofSeconds(2).toMillis());

        driver.findElement(By.partialLinkText("form")).click();

        // Pause to generate a longer recording
        Thread.sleep(Duration.ofSeconds(2).toMillis());

        // Verify recoding file
```

```
            Path recordingPath = wdm.getDockerRecordingPath();
            assertThat(recordingPath).exists();
        }

    }
```

3.4.4. Advanced Settings

D , D A , , , D - (*tmpfs*), , ( ). .

, D A , :

- `getDockerRecordingPath()`: .
- `getDockerNoVncUrl()`: C .
- `getDockerSeleniumServerUrl()`: ( ) ( ) .
- `getDockerService()`: D (-) D (. ., ,).
- `getDockerBrowserContainerId()`: (D)
- `getWebDriver()`: `WebDriver` (`create()`.
- `getWebDriverList`: `WebDriver` ( ).

A `WebDriver` , `getDockerRecordingPath()`, `getDockerNoVncUrl()`, `getDockerSeleniumServerUrl()` `getDockerBrowserContainerId()`, `quit()` , `WebDriver` . , D `WebDriver` (, . ., `WebDriver` .

3.5. Browsers Monitoring

A 5.2.0, D B .
B C , . B

3.5.1. Console Log Gathering

.
, (
, [redacted]). B -
. D
D
D create() watch(). , B
. A
, D getLogs()
. .

```
import static java.lang.invoke.MethodHandles.lookup;
import static org.assertj.core.api.Assertions.assertThat;
import static org.slf4j.LoggerFactory.getLogger;

import java.util.List;
import java.util.Map;

import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.openqa.selenium.WebDriver;
import org.slf4j.Logger;

import io.github.bonigarcia.wdm.WebDriverManager;

class GatherLogsFirefoxTest {

    static final Logger log = getLogger(lookup().lookupClass());

    WebDriverManager wdm = WebDriverManager.firefoxdriver().watch();
    WebDriver driver;

    @BeforeEach
    void setup() {
        driver = wdm.create();
    }

    @AfterEach
    void teardown() {
        driver.quit();
    }

    @Test
    void test() {
        driver.get(
                "https://bonigarcia.dev/selenium-webdriver-java/console-logs.html");

        List<Map<String, Object>> logMessages = wdm.getLogs();
        assertThat(logMessages).hasSize(5);

        logMessages.forEach(map -> log.debug("[{}] [{}] {}",
                map.get("datetime"),
                String.format("%1$-14s",
                        map.get("source").toString().toUpperCase() + "."
                                + map.get("type").toString().toUpperCase()),
                map.get("message")));
    }

}
```

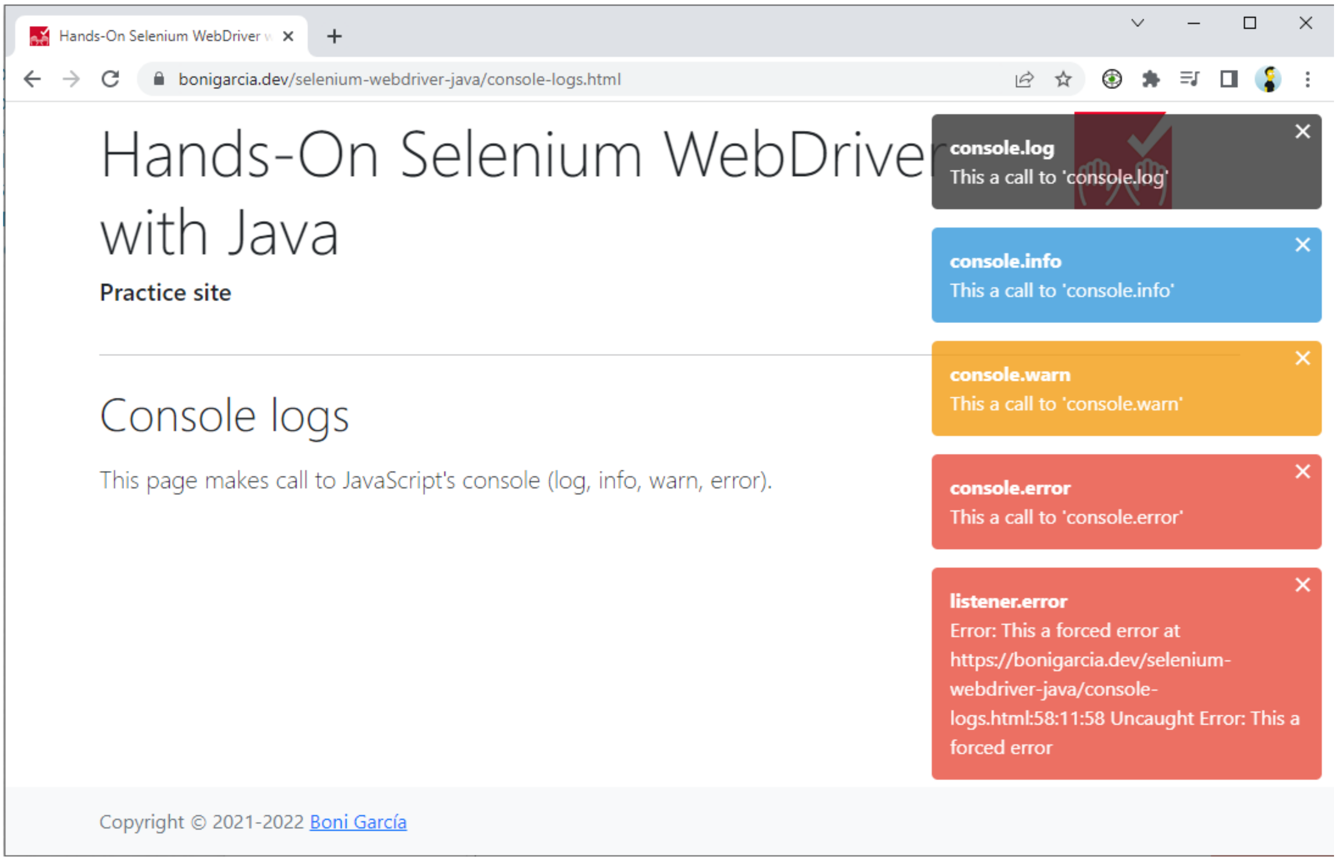
[redacted] , D 4.1.2.

3.5.2. Console Log

watchAndDisplay(), , :

```
import static java.lang.invoke.MethodHandles.lookup;
import static org.assertj.core.api.Assertions.assertThat;
import static org.slf4j.LoggerFactory.getLogger;

import java.time.Duration;
import java.util.List;
import java.util.Map;

import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.openqa.selenium.WebDriver;
import org.slf4j.Logger;

import io.github.bonigarcia.wdm.WebDriverManager;

class DisplayLogsChromeTest {

    static final Logger log = getLogger(lookup().lookupClass());

    WebDriverManager wdm = WebDriverManager.chromedriver().watchAndDisplay();
    WebDriver driver;

    @BeforeEach
    void setup() {
        driver = wdm.create();
    }
    @AfterEach
    void teardown() throws InterruptedException {
        // pause for manual browser inspection
        Thread.sleep(Duration.ofSeconds(3).toMillis());

        driver.quit();
    }

    @Test
    void test() {
        driver.get(
                "https://bonigarcia.dev/selenium-webdriver-java/console-logs.html");

        List<Map<String, Object>> logMessages = wdm.getLogs();

        assertThat(logMessages).hasSize(5);

        logMessages.forEach(map -> log.debug("[{}] [{}] {}",
                map.get("datetime"),
                String.format("%1$-14s",
                        map.get("source").toString().toUpperCase() + "."
                                + map.get("type").toString().toUpperCase()),
                map.get("message")));
    }

}
```

:


Hands-On Selenium WebDriver w
bonigarcia.dev/selenium-webdriver-java/console-logs.html

Hands-On Selenium WebDriver with Java

Practice site

Console logs

This page makes call to JavaScript's console (log, info, warn, error).

console.log
This a call to 'console.log'

console.info
This a call to 'console.info'

console.warn
This a call to 'console.warn'

console.error
This a call to 'console.error'

listener.error
Error: This a forced error at https://bonigarcia.dev/selenium-webdriver-java/console-logs.html:58:11:58 Uncaught Error: This a forced error

Copyright © 2021-2022 Boni García

Figure 3. Display logs example

3.5.3. Tab Recording

B .
(,). ,
`startRecording()` `stopRecording()`, , :

```java
import static java.lang.invoke.MethodHandles.lookup;
import static org.assertj.core.api.Assertions.fail;
import static org.slf4j.LoggerFactory.getLogger;

import java.io.File;
import java.time.Duration;
import java.util.concurrent.TimeUnit;

import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.slf4j.Logger;

import io.github.bonigarcia.wdm.WebDriverManager;

class RecordEdgeTest {
    static final Logger log = getLogger(lookup().lookupClass());

    static final int REC_TIMEOUT_SEC = 10;
    static final int POLL_TIME_MSEC = 100;
    static final String REC_FILENAME = "myRecordingEdge";
    static final String REC_EXT = ".webm";

    WebDriver driver;
    File targetFolder;
    WebDriverManager wdm = WebDriverManager.edgedriver().watch();

    @BeforeEach
    void setup() {
        driver = wdm.create();
        targetFolder = new File(System.getProperty("user.home"), "Downloads");
    }

    @AfterEach
    void teardown() {
        driver.quit();
    }

    @Test
    void test() throws InterruptedException {
        driver.get(
                "https://bonigarcia.dev/selenium-webdriver-java/slow-calculator.html"
);

        wdm.startRecording(REC_FILENAME);

        // 1 + 3
        driver.findElement(By.xpath("//span[text()='1']")).click();
        driver.findElement(By.xpath("//span[text()='+']")).click();
        driver.findElement(By.xpath("//span[text()='3']")).click();
        driver.findElement(By.xpath("//span[text()='=']")).click();

        // ... should be 4, wait for it
        WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));
        wait.until(ExpectedConditions.textToBe(By.className("screen"), "4"));

        wdm.stopRecording();

        long timeoutMs = System.currentTimeMillis()
                + TimeUnit.SECONDS.toMillis(REC_TIMEOUT_SEC);

        File recFile;
        do {
            recFile = new File(targetFolder, REC_FILENAME + REC_EXT);
            if (System.currentTimeMillis() > timeoutMs) {
                fail("Timeout of " + REC_TIMEOUT_SEC
                        + " seconds waiting for recording " + recFile);
                break;
            }
            Thread.sleep(POLL_TIME_MSEC);

        } while (!recFile.exists());

        log.debug("Recording available at {}", recFile);
    }

}
```

A C . , [redacted] A [redacted] .

3.5.4. Disa [redacted]

C (C) - ( ). C , B C . (. ., [redacted] D disableCsp(),

```
import static java.lang.invoke.MethodHandles.lookup;
import static org.assertj.core.api.Assertions.assertThat;
import static org.slf4j.LoggerFactory.getLogger;

import java.util.List;
import java.util.Map;

import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.openqa.selenium.WebDriver;
import org.slf4j.Logger;

import io.github.bonigarcia.wdm.WebDriverManager;

class DisableCspFirefoxTest {

    static final Logger log = getLogger(lookup().lookupClass());

    WebDriverManager wdm = WebDriverManager.firefoxdriver().watch()
            .disableCsp();
    WebDriver driver;

    @BeforeEach
    void setup() {
        driver = wdm.create();
    }

    @AfterEach
    void teardown() {
        driver.quit();
    }

    @Test
    void test() {
        driver.get("https://paypal.com/");
        List<Map<String, Object>> logMessages = wdm.getLogs();
        assertThat(logMessages).isNotNull();
    }

}
```

Chapter 4. Other Usages

, D .
.

4.1. WebDriverManager CLI

D C (C), . ., .
D C :

1. D *fat-JAR* (. ., D
A). A
`mvn` `compile` `assembly:single`,
D . - A .
, (`<args>`
,):

```
java -jar webdrivermanager-5.3.1-fat.jar <args>
```

2. . D .
, C ,
:

```
mvn exec:java -Dexec.args="<args>"
```

3. D . D
D D D - A .
D D .

```
docker run --rm -e ARGS="<args>" bonigarcia/webdrivermanager:5.3.1
```

D C .
C `<args>`). ,
D C :

```
[ERROR] The valid arguments for WebDriverManager CLI are:
[ERROR] 1. For resolving drivers locally:
[ERROR]         resolveDriverFor browserName <browserVersion>
[ERROR] (where browserName is: chrome|edge|firefox|opera|chromium|iexplorer)
[ERROR]
[ERROR] 2. For running a browser in a Docker (and use it trough noVNC):
[ERROR]         runInDocker browserName <browserVersion>
[ERROR] (where browserName is: chrome|edge|firefox|opera|safari|chrome-mobile)
[ERROR]
[ERROR] 3. For starting WebDriverManager Server:
```

```
[ERROR]        server <port>
[ERROR] (where the default port is 4444)
```

Option 1: Driver Resolver

D C (. ., ,)
.
. ,
D C (
):

- A :

```
java -jar webdrivermanager-5.3.1-fat.jar resolveDriverFor chrome
```

```
mvn exec:java -Dexec.args="resolveDriverFor chrome"
```

D :

```
docker run --rm -v ${PWD}:/wdm -e ARGS="resolveDriverFor chrome"
bonigarcia/webdrivermanager:5.3.1
```

Option 2: Browsers in Docker

D C D
C. *exploratory testing*
. ,
(C):

- A :

```
java -jar webdrivermanager-5.3.1-fat.jar runInDocker chrome
```

```
mvn exec:java -Dexec.args="runInDocker chrome"
```

D :

```
docker run --rm -it -v /var/run/docker.sock:/var/run/docker.sock -e ARGS="runInDocker
```

```
chrome" bonigarcia/webdrivermanager:5.3.1
```

C 92+ D
, D C [redacted]
, .

Option 3: Server

[redacted] , D C D ,
):

```
java -jar webdrivermanager-5.3.1-fat.jar server
```

[redacted]

```
mvn exec:java -Dexec.args="server"
```

D :

```
docker run --rm -p 4444:4444 -v /var/run/docker.sock:/var/run/docker.sock
bonigarcia/webdrivermanager:5.3.1
```

4.2. WebDriverManager Server

D [redacted] ,
(, , .). D
- A . D
. A
D , . ., 4444):

[redacted]

- :// :4444/ : .
- :// :4444/ : .
- :// :4444/ : .
- :// :4444/ : .
- :// :4444/ : .

D ( ) ,
wdm. [redacted] ,
100

:// :4444/ ? =100.

, D (. ., *hub*). WebDriver D ( ). . D D (, D , . ., :// :4444/):

```
var webdriver = require("selenium-webdriver");

async function wdmServerTest() {
    var wdmServerUrl = "http://localhost:4444/";
    var capabilities = {
        browserName : "chrome",
        version: "100"
    };

    try {
        var driver = await new webdriver.Builder().usingServer(wdmServerUrl)
                .withCapabilities(capabilities).build();

        var sutUrl = "https://bonigarcia.dev/selenium-webdriver-java/";
        await driver.get(sutUrl);

        await driver.getTitle().then(function(title) {
            console.log("The title of " + sutUrl + " is '" + title + "'")
        });

    } catch (err) {
        console.error("Something went wrong!\n", err.stack);

    } finally {
        if (driver) {
            driver.quit();
        }
    }
}

wdmServerTest();
```

4.3. WebDriverManager Agent

D *Java Agent.* , A , D . , D A . B D (ChromeDriver, FirefoxDriver, .), (, , .). , D (. .

`WebDriverManager.chromedriver.setup();`) , :

```java
import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

class ChromeAgentTest {

    WebDriver driver;

    @BeforeEach
    void setupTest() {
        driver = new ChromeDriver();
    }

    @AfterEach
    void teardown() {
        if (driver != null) {
            driver.quit();
        }
    }

    @Test
    void test() {
        // Test logic
    }

}
```

D A , D - A `-javaagent:/path/to/webdrivermanager-5.3.1-fat.jar`. A , ( ).

4.4. Selenium-Jupiter

D - , - 5 D . - 5 ( ) D

[redacted] , , `WebDriver` (`ChromeDriver`) . , - , :

```java
import static org.assertj.core.api.Assertions.assertThat;

import org.junit.jupiter.api.Test;
```

[redacted]

```
import org.junit.jupiter.api.extension.ExtendWith;
import org.openqa.selenium.chrome.ChromeDriver;

import io.github.bonigarcia.seljup.SeleniumJupiter;

@ExtendWith(SeleniumJupiter.class)
class ChromeTest {

    @Test
    void test(ChromeDriver driver) {
        driver.get("https://bonigarcia.dev/selenium-webdriver-java/");
        assertThat(driver.getTitle()).contains("Selenium WebDriver");
    }

}
```

- , D ,
(-), (
), ( ), ,
- .

[redacted]

4.5. Selenium Grid

D .
(, , .). , ,
D .

[redacted]

, ,
C . ,
, D C
, :

```
boni@ubuntu:~$ java -jar webdrivermanager-5.3.1-fat.jar resolveDriverFor chrome
[INFO] Using WebDriverManager to resolve chrome
[DEBUG] Detecting chrome version using online commands.properties
[DEBUG] Running command on the shell: [google-chrome, --version]
[DEBUG] Result: Google Chrome 103.0.5060.134
[DEBUG] Latest version of chromedriver according to
https://chromedriver.storage.googleapis.com/LATEST_RELEASE_103 is 103.0.5060.134
[INFO] Using chromedriver 103.0.5060.134 (resolved driver for Chrome 103)
[INFO] Reading https://chromedriver.storage.googleapis.com/ to seek chromedriver
[DEBUG] Driver to be downloaded chromedriver 103.0.5060.134
[INFO] Downloading
https://chromedriver.storage.googleapis.com/103.0.5060.134/chromedriver_linux64.zip
[INFO] Extracting driver from compressed file chromedriver_linux64.zip
[INFO] Driver location: /home/boni/chromedriver

boni@ubuntu:~$ java -jar selenium-server-4.3.0.jar standalone
```

```
01:31:52.806 INFO [LoggingOptions.configureLogEncoding] - Using the system default
encoding
01:31:52.810 INFO [OpenTelemetryTracer.createTracer] - Using OpenTelemetry for tracing
01:31:53.344 INFO [NodeOptions.getSessionFactories] - Detected 16 available processors
01:31:53.359 INFO [NodeOptions.discoverDrivers] - Discovered 1 driver(s)
01:31:53.382 INFO [NodeOptions.report] - Adding Chrome for {"browserName": "chrome"}
16 times
01:31:53.417 INFO [Node.<init>] - Binding additional locator mechanisms: id, relative,
name
01:31:53.436 INFO [GridModel.setAvailability] - Switching Node 3804a261-5889-44c5-
8cb1-d56162cf39ef (uri: http://172.18.0.1:4444) from DOWN to UP
01:31:53.436 INFO [LocalDistributor.add] - Added node 3804a261-5889-44c5-8cb1-
d56162cf39ef at http://172.18.0.1:4444. Health check every 120s
01:31:53.554 INFO [Standalone.execute] - Started Selenium Standalone 4.3.0 (revision
a4995e2c09*): http://172.18.0.1:4444
```

4.6. Appium

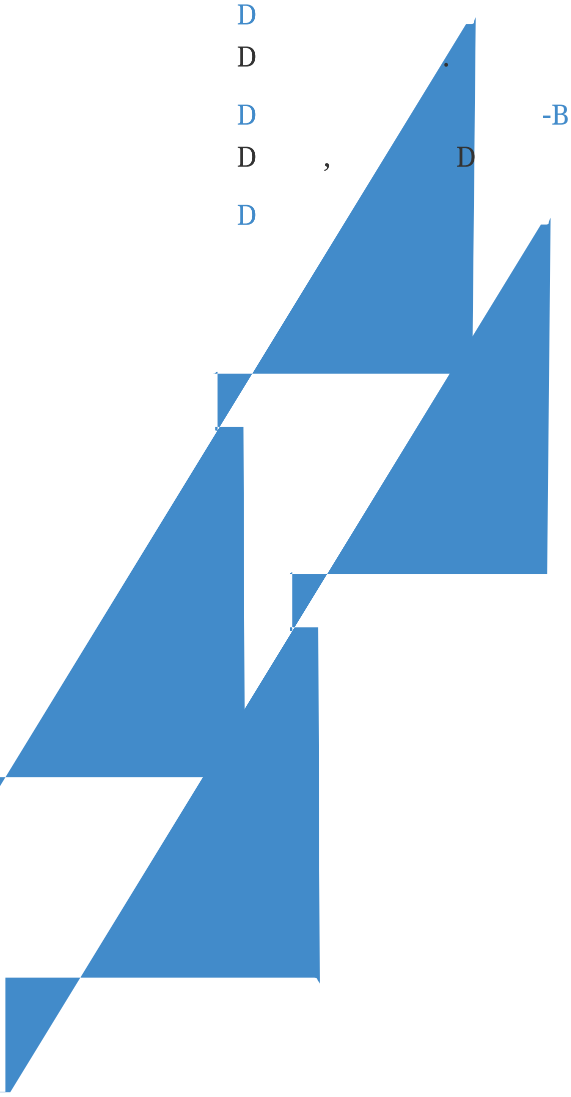
A .
D , A 3C D
[redacted]
,
A
A .
WebDriver C A :

```
public WebDriver createChromeAndroidDriver(String browserVersion,
        String deviceName, URL appiumServerUrl) {
    // Resolve driver and get its path
    WebDriverManager wdm = WebDriverManager.chromedriver()
            .browserVersion(browserVersion);
    wdm.setup();
    String chromedriverPath = wdm.getDownloadedDriverPath();

    // Create WebDriver instance using the driver path
    DesiredCapabilities capabilities = new DesiredCapabilities();
    capabilities.setCapability("browserName", "chrome");
    capabilities.setCapability("version", browserVersion);
    capabilities.setCapability("deviceName", deviceName);
    capabilities.setCapability("platformName", "android");
    capabilities.setCapability("chromedriverExecutable", chromedriverPath);

    return new AndroidDriver<WebElement>(appiumServerUrl, capabilities);
}
```

Chapter 5. Examples

A D .
, D , :

- D B : A 5, D ,
  D .
- D : D 5, D ,
  D .
- D -B : A -B , 5,
  D , D .
- D : D A .
  : A D 4
  .



Chapter 6. Advanced Configuration

D [redacted] , *Java API.* , (. ., chromedriver(), firefoxdriver(), A):

```
WebDriverManager.chromedriver().driverVersion("81.0.4044.138").setup();
WebDriverManager.firefoxdriver().browserVersion("75").setup();
WebDriverManager.operadriver().proxy("server:port").setup();
WebDriverManager.edgedriver().mac().setup();
```

, config() D . , D reset() .

D *Java system properties.* , D A *configuration k* [redacted] , A cachePath() ( ) wdm.cachePath. , , :

```
mvn test -Dwdm.cachePath=/custom/path/to/driver/cache
```

D *environmental variables.* (. ., wdm.cachePath) . _ (. ., WDM_CACHEPATH). . A , D D [redacted] :

```
docker run --rm -v ${PWD}:/wdm -e ARGS="resolveDriverFor chrome" -e
WDM_CHROMEVERSION=84 bonigarcia/webdrivermanager:5.3.1
```

( ) : 1) . 2) . 3) A .

D A : , D , D .

Table 1. C [redacted] *capabilities for driver management*

[redacted]

API method	Configuration key	Default value	Description
cachePath(String)	wdm.cachePath	~/.cache/selenium	[redacted]
resolutionCachePath(String)	wdm.resolutionCachePath	~/.cache/selenium	[redacted]
driverVersion(String)	wdm.chromeDriverVersion, wdm.operaDriverVersion, wdm.iExplorerDriverVersion, wdm.edgeDriverVersion, wdm.geckoDriverVersion, wdm.chromiumDriverVersion	"" ( )	[redacted]
browserVersion(String)	wdm.chromeVersion, wdm.operaVersion, wdm.edgeVersion, wdm.firefoxVersion, wdm.chromiumVersion, wdm.safariVersion	"" ( )	C ( )
forceDownload()	wdm.forceDownload=true	false (	[redacted]
useBetaVersions()	wdm.useBetaVersions=true	false ( )	[redacted])
architecture(Architecture)	wdm.architecture	"" (	[redacted]
arch32()	wdm.architecture=X32	"" (	[redacted]
arch64()	wdm.architecture=X64	"" ( )	[redacted]
arm64()	wdm.architecture=ARM64	"" ( )	A (A) 64-
operatingSystem(OperatingSystem)	wdm.os	"" ( )	[redacted] WIN LINUX MAC
win()	wdm.os=WIN	"" ( )	[redacted]
linux()	wdm.os=LINUX	"" ( )	[redacted]
mac()	wdm.os=MAC	"" ( )	[redacted]

API method	Configuration key	Default value	Description
driverRepositoryUrl(URL)	wdm.chromeDriverUrl, wdm.operaDriverUrl, wdm.edgeDriverUrl, wdm.geckoDriverUrl, wdm.iExplorerDriverUrl	D (.)	C
useMirror()	wdm.useMirror=true	false ( )	(, ,)
proxy(String)	wdm.proxy	"" ( )	proxy:port username:password@proxy:port). HTTPS_PROXY
proxyUser(String)	wdm.proxyUser	"")	. HTTPS_PROXY_USER
proxyPass(String)	wdm.proxyPass	"" ( )	. HTTPS_PROXY_PASS
gitHubToken(String)	wdm.gitHubToken	"" ( )	( ). GITHUB_TOKEN
ignoreVersions(String…)	wdm.ignoreVersions	"" ( )	( )
timeout(int)	wdm.timeout	30	( )

API method	Configuration key	Default value	Description
properties(String)	wdm.properties	webdrivermanager.properties	( )
avoidExport()	wdm.avoidExport=true	false ((. . webdriver.chrome.driver)	A 4 (, C)

exportParameter(String)
exportParameter(DriverManagerType)
	wdm.chromeDriverExport,
	wdm.geckoDriverExport,
	wdm.edgeDriverExport,
	wdm.iExplorerDriverExport,
	wdm.operaDriverExport
		(.)	. A DriverManagerType (. ., CHROME, FIREFOX, .) String

avoidOutputTree()	wdm.avoidOutputTree=true	false (, . ., chromedriver/linux64/2.37	A (, C)
avoidFallback()	wdm.avoidFallback=true	false ( )	A
avoidBrowserDetection()	wdm.avoidBrowserDetection =true	false (,)	[redacted]
avoidReadReleaseFromRepository()	wdm.avoidReadReleaseFrom Repository=true	false (, . ., - -)	[redacted]
avoidTmpFolder()	wdm.avoidTmpFolder=true	false (( ) ,	A ( )

API method	Configuration key	Default value	Description
`avoidShutdownHook()`	`wdm.avoidShutdownHook=true`	`false`	A `create()`
`browserVersionDetection Command(String)`	`wdm.browserVersionDetectionCommand`	`""` ( )	C ( )
`browserVersionDetection Regex(String)`	`wdm.browserVersionDetectionRegex`	`[^\\d\^\\.]`	
`useLocalVersionsProperties First()`	`wdm.versionsPropertiesOnlineFirst=true`	`false` ( )	
`useLocalCommandsProperties First()`	`wdm.commandsPropertiesOnlineFirst=true`	`false` ( )	
`versionsPropertiesUrl(URL)`	`wdm.versionsPropertiesUrl`		C
`commandsPropertiesUrl(URL)`	`wdm.commandsPropertiesUrl`		C
`clearDriverCache()`	`wdm.clearDriverCache=true`	`false` ( )	C
`clearResolutionCache()`	`wdm.clearResolutionCache =true`	`false` ( )	C
`ttl(int)`	`wdm.ttl`	`86400` (. ., 1)	.
`ttlBrowsers(int)`	`wdm.ttlForBrowsers`	`3600` (. ., 1)	( ).

Table 2. Configuration capabilities for browsers in Docker

API method	Configuration key	Default value	Description
dockerDaemonUrl(String)	wdm.dockerDaemonUrl	""	D
dockerTimezone(String)	wdm.dockerTimezone	Etc/UTC	
dockerNetwork(String)	wdm.dockerNetwork	bridge	D
dockerLang(String)	wdm.dockerLang	EN	
dockerShmSize(String)	wdm.dockerShmSize	256m	D . b ( ), k ( ), m ( ), g ( )
dockerTmpfsSize(String)	wdm.dockerTmpfsSize	128m	D - ( ).
dockerTmpfsMount(String)	wdm.dockerTmpfsMount	/tmp	-
dockerStopTimeoutSec(Integer)	wdm.dockerStopTimeoutSec	5	( )
enableVnc()	wdm.dockerEnableVnc=true	false	
viewOnly()	wdm.dockerViewOnly=true	false	(C) -
enableRecording()	wdm.dockerEnableRecording=true	false	D
dockerScreenResolution(String)	wdm.dockerScreenResolution	1280x1080x24	<width>x<height>x<colors-depth>
dockerRecordingFrameRate(int)	wdm.dockerRecordingFrameRate	12	

API method	Configuration key	Default value	Description

`dockerRecordingOutput(String)`
`dockerRecordingOutput(Path)`
	`wdm.dockerRecordingOutput`	`.`	. (`.mp4`)

`dockerRecordingPrefix(String)`	`wdm.dockerRecordingPrefix`	B	(. ., `_`)
`dockerCustomImage(String)`	`wdm.dockerCustomImage`	`""`	C D
`dockerVolumes(String[])`	`wdm.dockerVolumes`	`""`	D ( ) `"\local\path:\container\path"`
`dockerExtraHosts(String[])`	`wdm.dockerExtraHosts`	`""`	D ( ) `"hostname:IP"` (`"host1:192.168.48.82,host2:192.168.48.16"`)

API method	Configuration key	Default value	Description
`dockerPrivateEndpoint(String)`	`wdm.dockerPrivateEndpoint`	`""`	. - - . . - - . . / / (, ..), - - . . . / . , . .
`dockerAvoidPulling()`	`wdm.dockerAvoidPulling=true`	`false`	A D D

Table 3. Configuration capabilities for WebDriverManager Server

API method	Configuration key	Default value	Description
`serverPort(int)`	`wdm.serverPort`	`4444`	D
`serverPath(String)`	`wdm.serverPath`	`/`	D
`serverTimeoutSec(int)`	`wdm.serverTimeoutSec`	`60`	( ) D

Chapter 7. Known Issues

HTTP response code 403

(. .,) .
(D) ,
, 403 *(forbidden)*, :

```
io.github.bonigarcia.wdm.config.WebDriverManagerException: Error HTTP 403 executing
https://api.github.com/repos/mozilla/geckodriver/releases
    at io.github.bonigarcia.wdm.online.HttpClient.execute(HttpClient.java:172)
    at
io.github.bonigarcia.wdm.WebDriverManager.openGitHubConnection(WebDriverManager.java:1
266)
    at
io.github.bonigarcia.wdm.WebDriverManager.getDriversFromGitHub(WebDriverManager.java:1
280)
    at
io.github.bonigarcia.wdm.managers.FirefoxDriverManager.getDriverUrls(FirefoxDriverMana
ger.java:95)
    at
io.github.bonigarcia.wdm.WebDriverManager.createUrlHandler(WebDriverManager.java:1111)
    at io.github.bonigarcia.wdm.WebDriverManager.download(WebDriverManager.java:959)
    at io.github.bonigarcia.wdm.WebDriverManager.manage(WebDriverManager.java:877)
    at io.github.bonigarcia.wdm.WebDriverManager.fallback(WebDriverManager.java:1106)
    at
io.github.bonigarcia.wdm.WebDriverManager.handleException(WebDriverManager.java:1083)
    at io.github.bonigarcia.wdm.WebDriverManager.manage(WebDriverManager.java:883)
    at io.github.bonigarcia.wdm.WebDriverManager.setup(WebDriverManager.java:328)
```

, D
.

D .

A D 5.3.0, (
). 403 ,
A A .
D A .

Testing localhost

A .
, D , `localhost`
D ' . ,
. , -
. `172.17.0.1`, :

```
$ ip addr show docker0
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
group default
    link/ether 02:42:b4:83:10:c8 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 scope global docker0
       valid_lft forever preferred_lft forever
```

, D `host.docker.internal`, .

Chrome 92-94 in Docker

A C 92 94 D . `--disable-gpu` C . D 5 . , (. ., D D C) 92 94 D .

Chapter 8. Community

D ,
, *webdrivermanager_java*.
, -
D ,

Chapter 9. Support

D C ,
. (. .,
) (. .,).

Chapter 10. Further Documentation

- .

, :

[redacted]

- D . *O’Reilly Media, Inc.*, 2022.

. 5. *Packt Publishing Ltd*, 2017.

:

, ., . "A B A D ." *15th International Conference on the Quality of Information and Communications Technology* (A C) 2022, , . 2022.

, B , . " - : A 5 D ." *Journal of Systems and Software* (2022): 111298.

, B , . " D ." *Empirical Software Engineering* 26.5 (2021): 1-51.

, B , . " ." *Electronics* 9.7 (2020): 1067.

, B , . "A C - ." *Electronics* 9.3 (2020): 462.

, B , . " C ." *Electronics* 8.8 (2019): 854.

, B , . " C ." *IEEE Communications Standards Magazine* 1.2 (2017): 36-42.

, B , C D . " ." *Journal of web engineering* (2015): 474-502.

Chapter 11. About

D (C 2015-2022) -
B (@), A 2.0 . (
D) CC B - C- A 2.0.