

14. Java_QA (Selenium Grid)

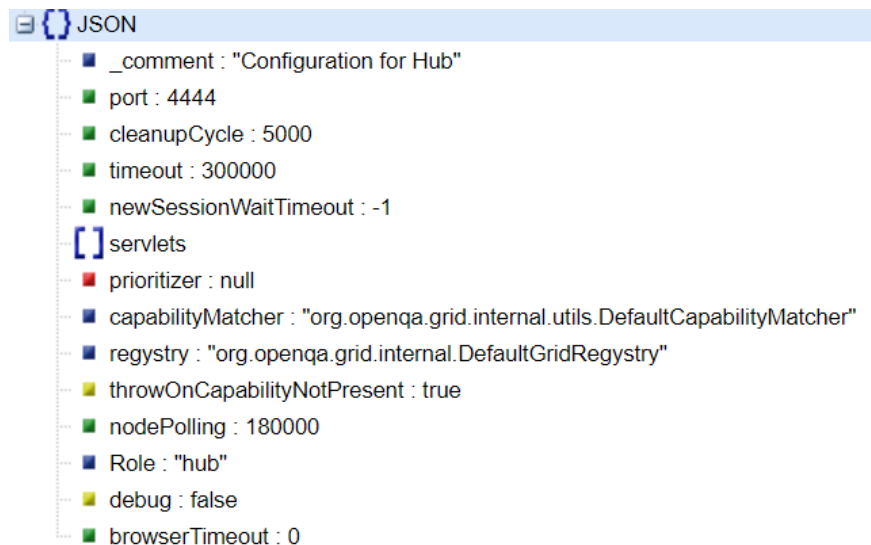
- Демонстрация запуска сценариев на удаленном сервере
- Способы конфигурации hub и nodes
- Использование Data Provider
- Запуск shell команд
- Конфигурация Selenium Grid с помощью Docker

Пример запуска Selenium Hub (запуск по умолчанию):

```
Linar@DESKTOP-OV3QR0S MINGW64 ~/Desktop/Project/14. QA (Selenium Grid, Docker)/data
$ java -jar selenium-server-standalone-3.141.59.jar -role hub
```

File hubConfig.json:

(конфигурация параметров в формате Json)



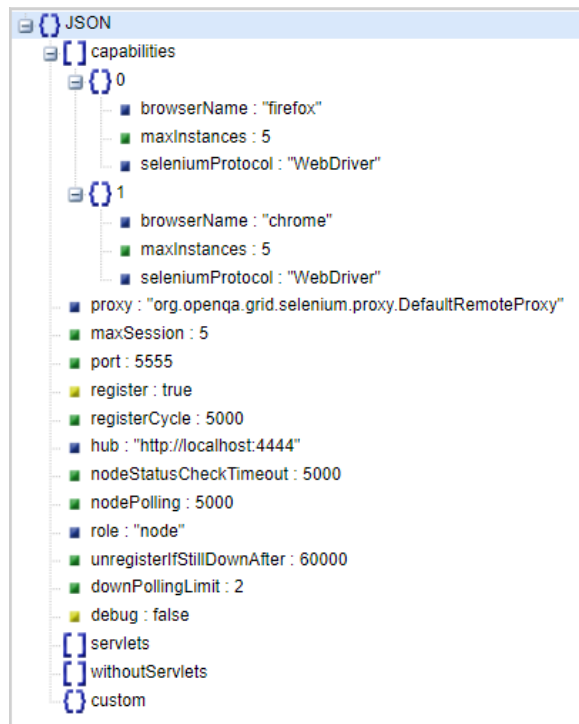
Упаковка в *.bat файл, и запуск с используемыми параметрами (-hubConfig):

```
Linar@DESKTOP-OV3QR0S MINGW64 ~/Desktop/Project/14. QA (Selenium Grid, Docker)/data
$ java -jar selenium-server-standalone-3.141.59.jar -role hub -hubConfig hubConfig.json
```

Запуск простого node (с указанием веб-драйвера и параметров -hub, -role, -port):

```
Linar@DESKTOP-0V3QR05 MINGW64 ~/Desktop/Project/14. QA (Selenium Grid, Docker)/data
$ java -Dwebdriver.chrome.driver=chromedriver.exe -jar selenium-server-standalone-3.141.59.jar \
> -role node -hub http://localhost:4444/grid/register -port 4546
```

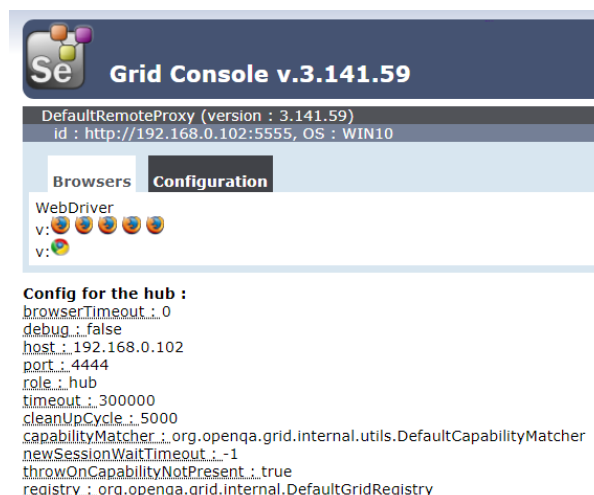
File nodeConfig.json (параметры node в формате json):



Упаковка в *.bat файл и запуск с параметрами (-nodeConfig):

```
Linar@DESKTOP-0V3QR05 MINGW64 ~/Desktop/Project/14. QA (Selenium Grid, Docker)/data
$ java -Dwebdriver.chrome.driver=chromedriver.exe -jar selenium-server-standalone-3.141.59.jar \
> -role node -nodeConfig nodeConfig.json
```

Переход на центральный узел, порт 4444:



Отработка сценариев:

Class ChromeTest: (простой запуск браузера на удаленной машине, использование объектов DesiredCapabilities, URL, RemoteWebDriver)

```
public class ChromeTest {
    public static void main(String[] args) throws MalformedURLException {
        DesiredCapabilities cap = new DesiredCapabilities().chrome();
        URL url = new URL("http://192.168.0.102:4444/wd/hub");
        WebDriver driver = new RemoteWebDriver(url, cap);
        driver.get("https://facebook.com");
        System.out.println(driver.getTitle());
        driver.quit();
    }
}
```

Class CrossBrowser: (запуск сценария на определённом узле в зависимости от номера порта - аннотация @Parameters, класс Platform, при этом используется Data Provider, который содержит данные для входа на сайт Facebook.com)

```
public class CrossBrowser {
    WebDriver driver;
    Platform ANY;
    String nodeUrl;

    @Parameters({"portNO", "appURL"})
    @BeforeMethod()
    public void setup(String portNO, String appURL) throws
        MalformedURLException {
        if (portNO.equalsIgnoreCase("4546")) {
            nodeUrl = "http://192.168.0.102:4546/wd/hub";
            System.out.println("Chrome Browser Environment test created");
            DesiredCapabilities cap = DesiredCapabilities.chrome();
            cap.setBrowserName("chrome");
            cap.setPlatform(Platform.ANY);

            driver = new RemoteWebDriver(new URL(nodeUrl), cap);
            driver.manage().window().maximize();
            driver.navigate().to(appURL);
            driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
        }
        else
            if (portNO.equalsIgnoreCase("4547")) {
                nodeUrl = "http://192.168.0.102:4547/wd/hub";
                System.out.println("Firefox Browser Environment test
created");

                DesiredCapabilities cap1 = DesiredCapabilities.firefox();
                cap1.setBrowserName("firefox");
                cap1.setPlatform(Platform.ANY);
                driver = new RemoteWebDriver(new URL(nodeUrl), cap1);
                driver.manage().window().maximize();
                driver.navigate().to(appURL);
                driver.manage().timeouts().implicitlyWait(10,
TimeUnit.SECONDS);
            }
    }
}
```

```

    }
    else
        System.err.println("Provide correct port no");
}

@Test(dataProvider="Authentication")
public void facebookLogin(String username, String password) throws
InterruptedException
{
    System.out.println("Trying with :"+username);
    System.out.println(driver.getTitle());

    driver.findElement(By.cssSelector("input[id='email']")).sendKeys(username);

    driver.findElement(By.cssSelector("input[id='pass']")).sendKeys(password);
    driver.findElement(By.cssSelector("button[name='login']")).click();
}

@DataProvider(name="Authentication")
public Object[][] credentials()
{
    Object[][] obj = new Object[3][2];
    obj[0][0]="Admin";
    obj[0][1]="Admin";

    obj[1][0]="Tester";
    obj[1][1]="Tester";

    obj[2][0]="Developer";
    obj[2][1]="developer";

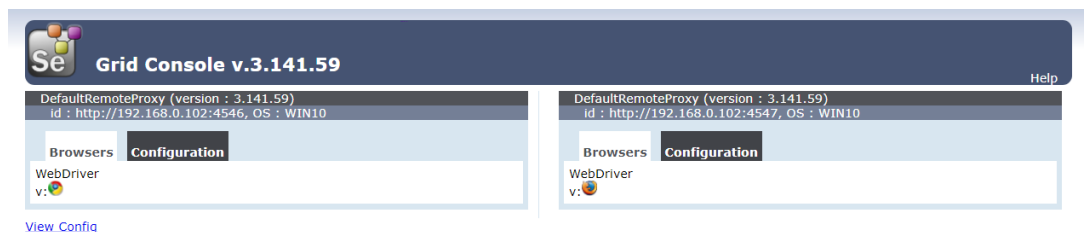
    return obj;
}

@AfterMethod()
public void tearDown()
{
    driver.quit();
    System.out.println("Closing the Browser");

    System.out.println("*****");
}
}

```

Запуск 2-х узлов:



Конфигурация файла CrossBrowser.xml для запуска сценариев:
(указаны параметры portNO и appURL)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="All Test Suite" parallel="tests">
  <test name="ChromeTest">
    <parameter name="portNO" value="4546"></parameter>
    <parameter name="appURL"
value="http://www.facebook.com"></parameter>
    <classes>
      <class name="seldocker.CrossBrowser"></class>
    </classes>
  </test>
  <test name="FirefoxTest">
    <parameter name="portNO" value="4547"></parameter>
    <parameter name="appURL"
value="https://www.facebook.com"></parameter>
    <classes>
      <class name="seldocker.CrossBrowser"></class>
    </classes>
  </test>
</suite>
```

Параллельный запуск сценариев:

✓ All Test Suite	1 m 29 s 45 ms
✓ FirefoxTest	18 s 419 ms
✓ ChromeTest	18 s 860 ms
✓ CrossBrowser	18 s 860 ms
✓ facebookLogin[Admin, Admin]	3 s 901 ms
✓ facebookLogin[Tester, Tester] (1)	2 s 930 ms
✓ FirefoxTest	7 s 728 ms
✓ CrossBrowser	7 s 728 ms
✓ facebookLogin[Admin, Admin] (2)	7 s 728 ms
✓ ChromeTest	7 s 280 ms
✓ CrossBrowser	7 s 280 ms
✓ facebookLogin[Developer, developer] (3)	2 s 428 ms
✓ FirefoxTest	36 s 758 ms
✓ CrossBrowser	36 s 758 ms
✓ facebookLogin[Tester, Tester] (4)	7 s 533 ms
✓ facebookLogin[Developer, developer] (5)	7 s 501 ms

Конфигурация SeleniumGrid с помощью Docker

Пример файла docker-compose.yml: (лежит в корне проекта)

```
version: "3"
services:
  selenium-hub:
    image: selenium/hub:latest
    container_name: selenium-hub
    ports:
      - "4444:4444"

  chrome:
    image: selenium/node-chrome:latest
    volumes:
      - /dev/shm:/dev/shm
    depends_on:
```

```

- selenium-hub
environment:
- HUB_HOST=selenium-hub
- HUB_PORT=4444

firefox:
image: selenium/node-firefox:latest
volumes:
- /dev/shm:/dev/shm
depends_on:
- selenium-hub
environment:
- HUB_HOST=selenium-hub
- HUB_PORT=4444

```

Class CommonMethods: (содержит метод для запуска shell команд, в данном случае “docker-compose up”)

```

public class CommonMethods {
    public static void runTerminalCommand(String command, String logText) {
        try {
            String path = System.getProperty("user.dir");
            ProcessBuilder builder = new ProcessBuilder(
                "cmd.exe", "/c", "cd \"\" + path + "\" && " + command);
            builder.redirectErrorStream(true);
            Process p = builder.start();
            BufferedReader r = new BufferedReader(new
InputStreamReader(p.getInputStream()));
            String line;
            while (true) {
                line = r.readLine();
                if (line.contains(logText)) {
                    Thread.sleep(5000);
                    System.out.println(line);
                    break;
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

Class GridTest: (аннотация @BeforeSuite – сборка docker image, выполнение команды “docker-compose up”, @AfterSuite – закрытие контейнера, параметр @Optional)

```

public class GridTest {

    WebDriver driver;

    @BeforeSuite
    public void startContainer()
    {
        CommonMethods.runTerminalCommand("docker-compose up","Registered a
node");
    }

    @BeforeTest

```

```

@Parameters({"browser"})
public void setup(@Optional("firefox")String browser) throws
MalformedURLException
{
    DesiredCapabilities cap = new DesiredCapabilities();
    URL url = new URL("http://192.168.0.102:4444/wd/hub");
    if(browser.equalsIgnoreCase("chrome"))
    {
        cap.setBrowserName("chrome");
        cap.setCapability("name", "ChromeTest"); //
        driver = new RemoteWebDriver(url, cap);
        driver.get("https://www.google.com");
    }else if(browser.equalsIgnoreCase("firefox"))
    {
        cap.setCapability(CapabilityType.BROWSER_NAME,BrowserType.FIREFOX);
        driver = new RemoteWebDriver(url, cap);
        driver.get("https://www.facebook.com");
    }
    System.out.println("Opening in the "+browser);
}

@Test
public void getTitle() throws MalformedURLException
{
    System.out.println(driver.getTitle());
}

@AfterTest
public void tearDown()
{
    driver.quit();
}

@AfterSuite
public void stopContainer()
{
    CommonMethods.runTerminalCommand("docker-compose down","Removing
selenium-hub");
}

```

Конфигурация файла GridTest.xml для запуска сценария:
(указание параметра browser)

```

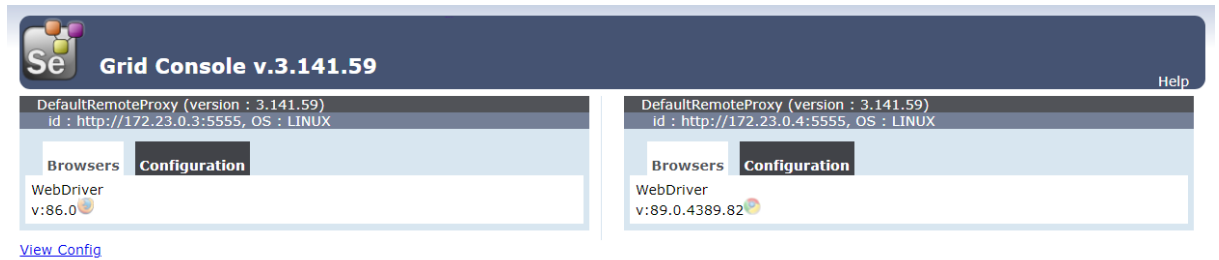
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Docker Grid" parallel="tests" thread-count="2">
    <test name="Chrome">
        <parameter name="browser" value="chrome"></parameter>
        <classes>
            <class name="seldocker.GridTest"></class>
        </classes>
    </test>
    <test name="Firefox">
        <parameter name="browser" value="firefox"></parameter>
        <classes>
            <class name="seldocker.GridTest"></class>
        </classes>
    </test>
</suite>

```

Pom.xml: (запуск сборки с помощью maven)

```
<configuration>
  <suiteXmlFiles>
    <!-- TestNG suite XML files -->
    <suiteXmlFile>TestSuite/GridTest.xml</suiteXmlFile>
  </suiteXmlFiles>
</configuration>
```

Запуск mvn clean test:



The screenshot displays the Selenium Grid Console v.3.141.59 interface. The top bar is dark blue with the Selenium logo and the title "Grid Console v.3.141.59". A "Help" link is visible in the top right corner. Below the header, there are two panels, each representing a node in the grid. Both panels show "DefaultRemoteProxy (version : 3.141.59)" and "id : http://172.23.0.3:5555, OS : LINUX". Each panel has a "Browsers" tab and a "Configuration" tab. The left panel's "Browsers" tab shows "WebDriver v:86.0". The right panel's "Browsers" tab shows "WebDriver v:89.0.4389.82". A "View Config" link is located below the left panel.