# **Lab: Test Automation Architectures**

# **Lab: Test Automation Architectures**

## Instructions

You have been given an application project named "PracticеProject" and a Test project:

Картина, която съдържа текст, екранна снимка, Шрифт, софтуер

Описанието е генерирано автоматично

The "PracticеProject" is a simple web application that you need to run locally and explore. There is no documentation about its functionality because it is mainly used for practicing different Front-End testing tools such as Selenium, Selenium Grid, Selenoid, etc.

The "TestProject", on the other hand, contains two Selenium tests for the "Practical Project" application:

* The first test checks if the title is as expected:

Картина, която съдържа текст, Шрифт, линия, екранна снимка

Описанието е генерирано автоматично

* The second test creates a new entity and then checks if the entity is displayed on the home page table with the correct data.

Картина, която съдържа текст, екранна снимка

Описанието е генерирано автоматично

You can explore the tests in the project by debugging them. Feel free to write more tests if you want to. Also, make sure that the URL is correct:

Картина, която съдържа текст, екранна снимка, Шрифт

Описанието е генерирано автоматично

To check the URL, simply run the application by pressing CTRL + F5 and check the URL in the browser's address bar:



If you get a different URL, change it in the setup method. Then run the tests from the Visual Studio Test Explorer:

Картина, която съдържа текст, екранна снимка, софтуер, Мултимедиен софтуер

Описанието е генерирано автоматично

But you already know these things. Let's add new features to our project to learn more and improve our efficiency and productivity in testing.

## Selenium Grid

Selenium Grid is part of the Selenium Suite that lets you run test scripts on multiple machines, browsers, and operating systems at the same time. This enables distributed test execution and parallel testing, which can significantly reduce the time needed for running tests.

Before we integrate Selenium Grid in our test project, we need to make a few things to ensure successful test execution later.

### **Install JDK**

We need to install JDK package to be able to run Selenium Server which we will need for Selenium Grid:

1. Check if you already have installed JDK on your computer by opening the CMD or PowerShell and running the command "**java --version**"

**NOTE:** Since you just finished the Appium lectures, the Java JDK should already be installed. However, if it's not, please refer to the "**09.Appium-Installations.docx**" document (Part 5. Setup Java JDK) in the resources for Lecture 09. Appium Testing for instructions on how to **install Java JDK** and set up **JAVA\_HOME"** and Update **"Path" variable**.

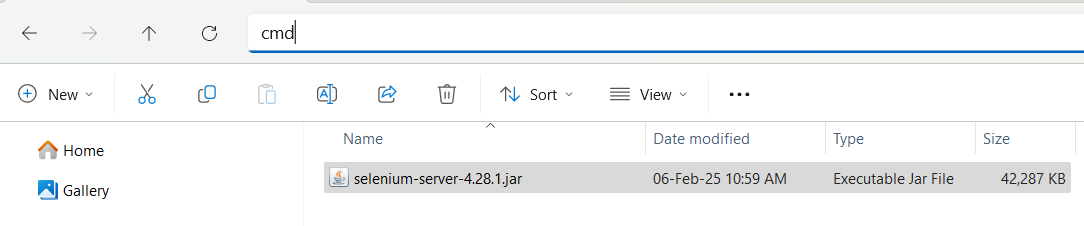
### **Download Selenium Server**

To be able to use Selenium Grid you need to download the Selenium Server jar file from the Selenium official website. Here is the link: [Downloads | Selenium](https://www.selenium.dev/downloads/)

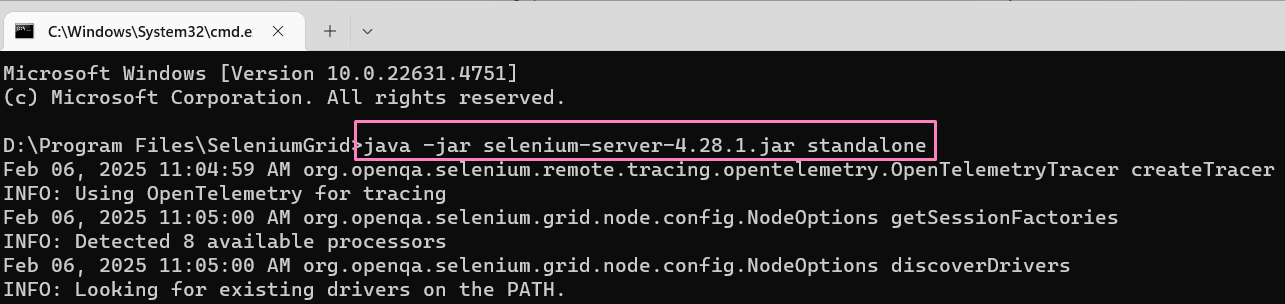


### **Run Selenium Server in Standalone Mode**

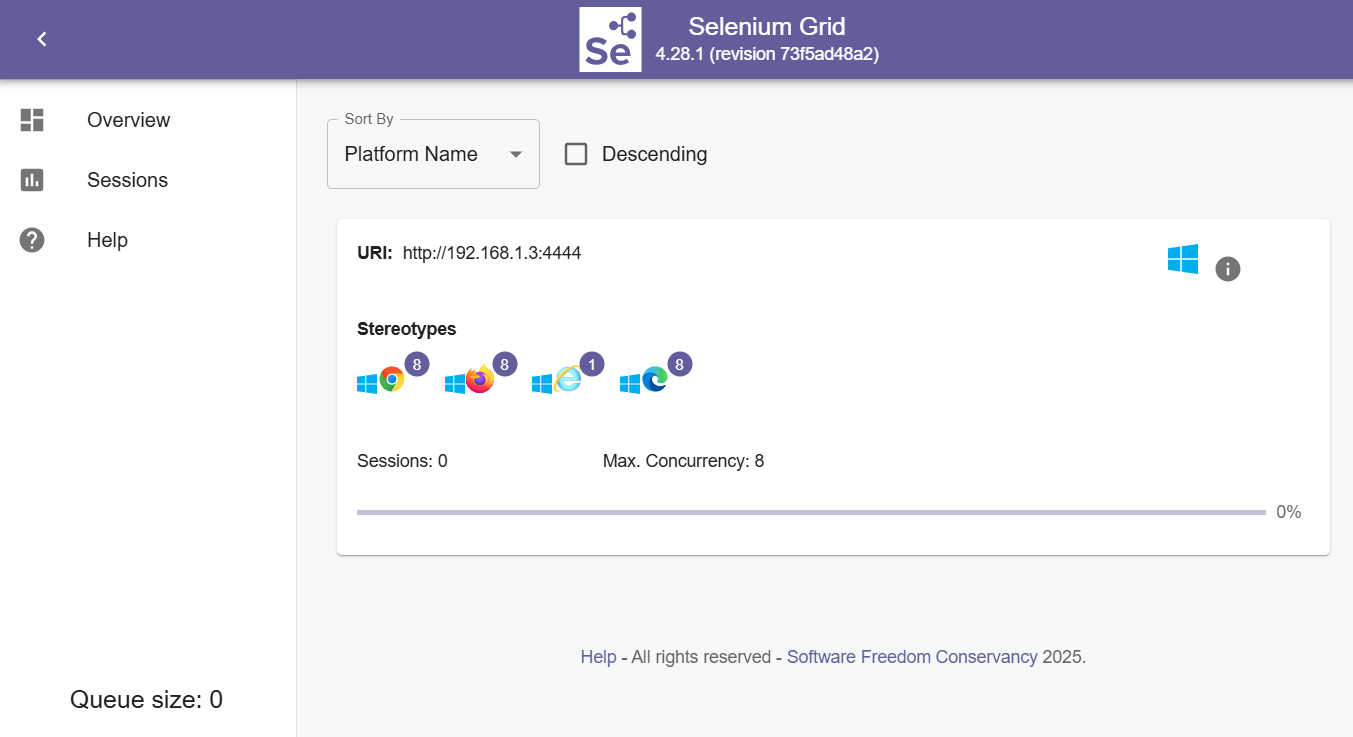
1. Navigate to the folder where you downloaded the Selenium server JAR file (selenium-server-4.28.1.jar).
2. In the file explorer, click on the address bar at the top of the folder window, type "cmd," and press "Enter." This will open a Command Prompt window with the folder's current path set as the working directory.



1. Run the command "**java -jar selenium-server-4.28.1.jar standalone**" to start Selenium Server in standalone mode:



1. Open the web browser and navigate to " <http://localhost:4444>" to see the Selenium Grid Console and verify that the Selenium Server is running correctly in standalone mode:

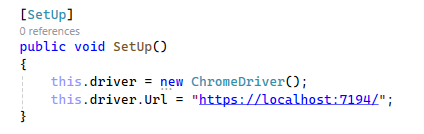


You can see in the browser that you have 4 OS and 4 type of browsers. Based on the processors on your computer Selenium Grid will create the number of instances for each browser type.

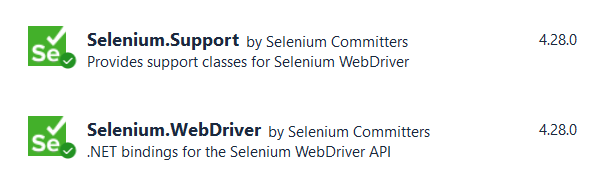
Also, your local machine is the Selenium Grid hub and each of the OS and the browser type attached to it is a Selenium Grid node.

### **Integrate Selenium Grid through the Code**

Let's change our test in the TestProject to see how Selenium Grid is working. At this moment we have the following setup configurations:



1. First, be sure that you have installed the following NuGet packages (they are installed in the current project, but is good to know):



1. Change the SetUp method as follows:
   * Add the ChromeOptions:



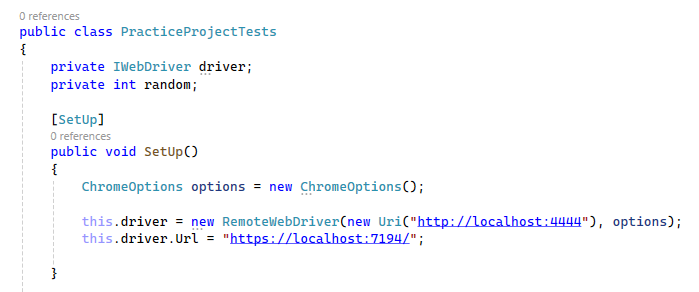
* + Change the IWebDriver to RemoteWebDriver.
  + The RemoteWebDriver constructor takes two **parameters**: the **URI of the Selenium Grid** (e.g., <http://localhost:4444>) and the **options** that define the browser's configuration settings.



* Add the following line to your setup, to specify the URL of the application you want to test.



1. The whole setup method should look like this:



1. Start the Practice Project with CTRL + F5.
2. Execute all tests and check the Selenium Grid Console. You can see in the Session tab the test execution:

Картина, която съдържа текст, екранна снимка, номер, Шрифт

Описанието е генерирано автоматично

### **Changing the Browser**

You can easily change the browser for test execution:

Картина, която съдържа текст, Шрифт, софтуер, Мултимедиен софтуер

Описанието е генерирано автоматично

Now the tests will run under the Edge bowser:

Картина, която съдържа текст, екранна снимка, линия, Шрифт

Описанието е генерирано автоматично

### **Changing the Test Execution on the Remote VM**

Till now we executed our test locally. But we can also execute them to the remote virtual machine of the Selenium Grid.

1. Go to Selenium Grid Console by typing in the browser address bar <http://localhost:4444>.
2. Go to "Overview" tab:

Картина, която съдържа текст, екранна снимка, номер, Шрифт

Описанието е генерирано автоматично

1. Check the URL and copy it:

Картина, която съдържа екранна снимка, текст, софтуер, Компютърна икона

Описанието е генерирано автоматично

1. Set it to the RemoteWebDriver in the code:

Картина, която съдържа текст, екранна снимка, линия, Шрифт

Описанието е генерирано автоматично

1. Execute the tests and check the results.

### **Execute All Test on More than One Browser**

1. Add more than one TestFixture attribute to the test class:

Картина, която съдържа текст, Шрифт, екранна снимка

Описанието е генерирано автоматично

1. Add a constructor and private variable "browserType":

Картина, която съдържа текст, екранна снимка, Шрифт

Описанието е генерирано автоматично

IMPORTANT: The constructor parameter "browserType" will get the value from given values in the TestFixture attribute.

1. Create a private method named GetOptions that returns DriverOptions and takes a parameter "browserType":

Картина, която съдържа текст, екранна снимка, Шрифт, софтуер

Описанието е генерирано автоматично

1. Change the code line for browser options as follows:

Картина, която съдържа текст, екранна снимка, софтуер, Мултимедиен софтуер

Описанието е генерирано автоматично

1. The whole test setup should look like this:

Картина, която съдържа текст, екранна снимка, софтуер

Описанието е генерирано автоматично

1. Execute the tests and check the results in Selenium Grid Console:

Картина, която съдържа екранна снимка, текст, номер, линия

Описанието е генерирано автоматично

### **Summary**

Selenium Grid allows you to run multiple tests simultaneously across different environments, enabling parallel execution and cross-browser testing. This increases the efficiency of test execution, ensures that the application works correctly in different environments and reduces the need for extensive local infrastructure.

Also, Selenium Grid is easy for setup and allows flexible ways of testing strategies and realization.

## Selenoid

Selenoid is a lightweight alternative to Selenium Grid. Installing Selenoid involves setting up Docker and using it to run Selenoid containers. Here's a step-by-step guide on how to install Selenoid using Docker Desktop and then interacting with it using C#:

### **Download Docker Desktop**

Navigate to the Docker website (<https://www.docker.com/products/docker-desktop>) and download Docker Desktop for your operating system (Windows or macOS):



### **Install and Run Docker Desktop**

Double-click Docker Desktop Installer.exe to run it (for this example, Docker Desktop 4.10.1 will be installed; you do not need to download the same version - it's best to download the latest one).

On the Configuration page, choose the "Use WSL 2 instead of Hyper-V" option if you have a choice. If your system only supports one of the options, you will not be able to select which backend to use:

Graphical user interface, text, application

Description automatically generated

Follow the installation instructions provided by Docker for your operating system. You may need to restart your computer after installation is complete:

Картина, която съдържа текст, екранна снимка, софтуер, Уеб страница

Описанието е генерирано автоматично

### **Install and Configure Selenoid**

1. Go to Selenoid releases page: [Release v1.8.8: Docker API 1.45 support · aerokube/cm (github.com)](https://github.com/aerokube/cm/releases/tag/1.8.8)
2. Download Selenoid Configuration Manager for your OS. For Windows is cm\_windows\_386.exe for 32- or 86-bit systems and cm\_windows\_amd64.exe for 64-bit systems

Картина, която съдържа текст, екранна снимка, Шрифт

Описанието е генерирано автоматично

1. Open a Terminal or CMD (ensure Docker Desktop is running):

Картина, която съдържа текст, екранна снимка, софтуер, мултимедия

Описанието е генерирано автоматично

1. Drag and drop the downloaded file into the Terminal to directly paste the file path:



1. Add the command "selenoid configure" after the file path and press "Enter":

Картина, която съдържа екранна снимка, Шрифт, Графика, текст

Описанието е генерирано автоматично

1. Execute the command "selenoid start --vnc”:



1. Wait for Selenoid and its browsers to install.

Картина, която съдържа текст, екранна снимка, Шрифт

Описанието е генерирано автоматично

1. Once done, you should see the following messages:



1. Go to Docker Desktop and check if all is installed in the "Image" section:

Картина, която съдържа текст, екранна снимка, софтуер, Мултимедиен софтуер

Описанието е генерирано автоматично

1. Check the "Container" section as well:

Картина, която съдържа екранна снимка

Описанието е генерирано автоматично

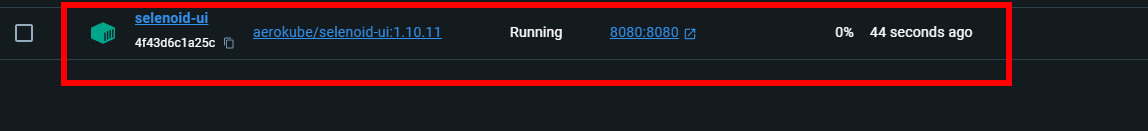
1. Start the Selenoid UI by executing the command "selenoid-ui start":



1. You should get this message:



1. In Docker Desktop, check for this container:



1. Note the port where the Selenoid UI is running. Click on it to open the Selenoid UI in the browser:

Картина, която съдържа екранна снимка, текст, Шрифт, линия

Описанието е генерирано автоматично

1. In the Selenoid UI in the browser, go to the "Stats" tab. We will need this tab soon:

Картина, която съдържа текст, екранна снимка, софтуер, Мултимедиен софтуер

Описанието е генерирано автоматично

Now, let’s proceed to Selenoid with C#.

### **Interact with Selenoid in C#**

We are going to create new test project:

1. First, go to Docker Desktop and check the images for Firefox and Chrome for their versions:

Картина, която съдържа текст, екранна снимка, Шрифт, номер

Описанието е генерирано автоматично

**IMPORTANT:** It's possible to have different versions of browsers. If you don’t have any installed browser in Docker for Selenoid, you need to do it manually:

* + For Chrome, execute this command in a Terminal: **docker pull selenoid/chrome:126.0**
  + For Firefox, execute this command in a Terminal: **docker pull selenoid/firefox:125.0**

1. Create an NUnit project:
   * First, create a new project:

Картина, която съдържа текст, екранна снимка, Шрифт, софтуер

Описанието е генерирано автоматично

* + Choose "NUnit Test Project":

Картина, която съдържа текст, екранна снимка, софтуер, Мултимедиен софтуер

Описанието е генерирано автоматично

* + Name the project and solution:

Картина, която съдържа текст, екранна снимка, софтуер, Мултимедиен софтуер

Описанието е генерирано автоматично

* + Choose .NET 8.0 as the framework.
  + Install the Selenium WebDriver NuGet package:

Картина, която съдържа текст, екранна снимка, софтуер, Мултимедиен софтуер

Описанието е генерирано автоматично

* + Rename the auto-generated class "UnitTest1.cs" to "SelenoidTests".

1. Use TestFixture attributes over the class name to set up browsers and browser versions:

Картина, която съдържа текст, Шрифт, екранна снимка

Описанието е генерирано автоматично

1. Create a constructor and private field to get the values given through the TestFixture attributes:

Картина, която съдържа текст, екранна снимка, Шрифт

Описанието е генерирано автоматично

1. Also, create a private field for the WebDriver:

Картина, която съдържа текст, екранна снимка, Шрифт

Описанието е генерирано автоматично

1. Use the SetUp method to make your configurations for Selenoid:

Картина, която съдържа текст, екранна снимка, софтуер, Шрифт

Описанието е генерирано автоматично

The GetOptions method is a helper method to set browser options:

Картина, която съдържа текст, екранна снимка

Описанието е генерирано автоматично

1. Don’t forget for the TearDown method:

Картина, която съдържа текст, екранна снимка, Шрифт, дизайн

Описанието е генерирано автоматично

1. Write your first test with Selenoid by following these steps:
   * Go to <https://en.wikipedia.org/>
   * Type "Quality Assurance" in the search input field
   * Press Search button
   * Check that article title (h1 tag) is as expected
2. Let’s make another test:
   * Go to <https://en.wikipedia.org/>
   * Click on the logo (top left corner)
   * Check that welcome message is as expected
3. Go to the Selenoid UI in the "Stats" tab. Make sure there are no sessions:

Картина, която съдържа текст, екранна снимка, Шрифт, визитка

Описанието е генерирано автоматично

1. Check if all needed containers are running:

Картина, която съдържа текст, екранна снимка, софтуер, Шрифт

Описанието е генерирано автоматично

1. Ensure everything is connected:

Картина, която съдържа текст, екранна снимка, Шрифт, линия

Описанието е генерирано автоматично

1. Run the tests:

Картина, която съдържа текст, електроника, екранна снимка, софтуер

Описанието е генерирано автоматично

1. Check again the "Stats" tab in Selenoid-ui:

Картина, която съдържа екранна снимка, текст

Описанието е генерирано автоматично

IMPORTANT: If you want to run the tests again, you should remove all sessions first:

Картина, която съдържа екранна снимка, текст, софтуер, Мултимедиен софтуер

Описанието е генерирано автоматично

### **Summary**

Selenoid is a lightweight alternative to Selenium Grid. It is easy to use and does not require many configurations. It has a browser UI, which provides additional information and an easy way to manage and track your test sessions.

## BrowserStack

Running tests using NUnit SDK on BrowserStack’s Selenium Grid of 3000+ real devices and desktop browsers.

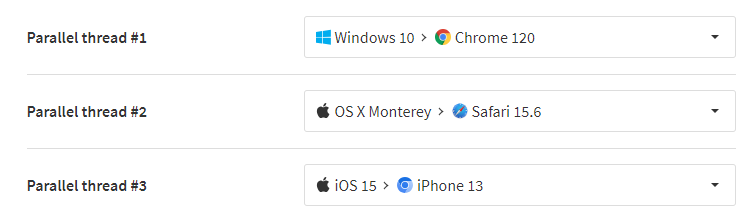
#### Prerequisites

BrowserStack Username and Access key. You can find this in your account profile section. If you do not have an account yet, sign up for a Free Trial.

#### Run a sample build

To execute Selenium tests using the NUnit SDK on BrowserStack Automate, follow these simple steps:

* **Get sample project**
  + Download [project](https://github.com/browserstack/nunit-browserstack/archive/refs/heads/master.zip)
  + After downloading the sample project, unzip it in a desired location.
  + Import the downloaded project
  + Click File > Open. Navigate to your project and open it.
* **Configure your browserstack.yml config file**
  + The browserstack.yml file holds all the required capabilities to run your tests on BrowserStack.
  + Set access credentials - Set **userName** and **accessKey** parameters in the **browserstack.yml** file, available in the root directory, to authenticate your tests on BrowserStack.
  + Select desktop browsers or real devices from a list of 3000+ available combinations:



* + Update the **browserstack.yml** config file - Replace the **platforms** object in the **browserstack.yml** file.

userName: YOUR\_USERNAME

accessKey: YOUR\_ACCESS\_KEY

platforms:

- os: Windows

osVersion: 10

browserName: Chrome

browserVersion: 120.0

- os: OS X

osVersion: Monterey

browserName: Safari

browserVersion: 15.6

- deviceName: iPhone 13

osVersion: 15

browserName: Chromium

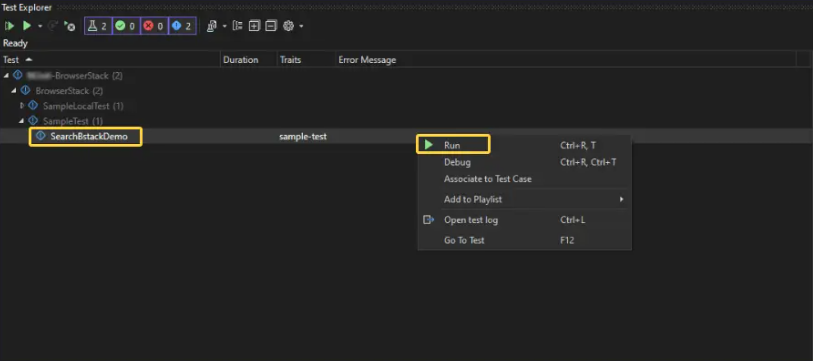
deviceOrientation: portrait

browserstackLocal: true

buildName: browserstack-build-1

projectName: BrowserStack Sample

* **Run sample tests on BrowserStack!** Follow these steps to run your test on BrowserStack:
* On the Visual Studio toolbar, select Test > Test Explorer.
* Right-click on your test and click Run.



Full Tutorial [Here](https://www.browserstack.com/docs/automate/selenium/getting-started/c-sharp/nunit)