EMERALD LABS

Contact@emerald-labs.com

**components of emerald labs test automation framework**

**MAIN Components of Emerald Labs test Automation Framework**

1. **Selenium** with Java binding
2. **Maven** as a build tool including dependencies:

|  |  |
| --- | --- |
| **Dependency Name** | **Version** |
| Selenium Java | *4.4.0* |
| Selenium API | *4.4.0* |
| TestNG | *7.6.1* |
| Webdriver Manager | *5.3.0* |
| Log4j-API | *2.18.0* |
| Log4j-Core | *2.18.0* |
| Apache POI | *5.2.2* |
| Poi-ooxml  Extent Reports | *5.2.2*  *5.0.9* |

Follow the given below procedure to add maven dependencies:

* Open pom.xml in the root directory of your project.

To quickly navigate to a file, press Ctrl+Shift+N and enter its name.

* In pom.xml, press Alt+Insert and select Add dependency.
* Locate the required dependencies, select its version in the search results, and click Add next to it.
* Apply the changes in the build script: press Ctrl+Shift+O or click Load Maven Changes in the notification that appears in the top-right corner of the editor.

1. **IntelliJ IDEA** 2022.2.1 (Community Edition) as an IDE

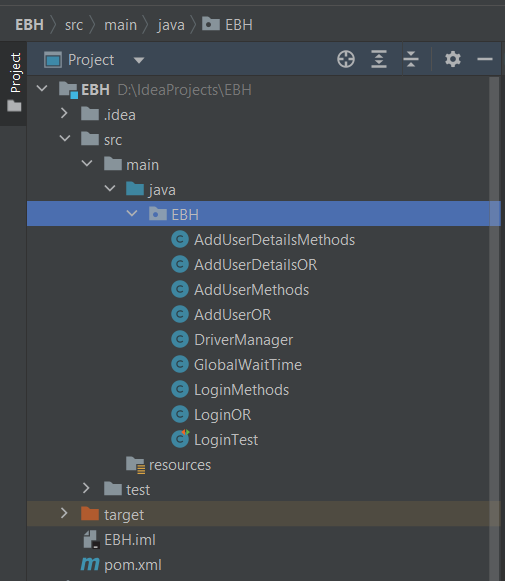
*Build #*IC-222.3739.54, *built on* August 16, 2022

*Runtime version:* 17.0.3+7-b469.37 amd64

*VM:* OpenJDK 64-Bit Server VM by Jet Brains s.r.o.

1. **Page Factory Model** as a Design Pattern

**Project structure/directory**



Note:

Driver Manager and Global Wait Time are global classes throughout the project.

How to Create a Test case

There are multiple steps involved to create a test case using automation framework. The steps are given below:

**Step-1**

Left click on project, create a new class with OR (object repository) abbreviation and apply any locating strategy w.r.t web element and build the OR’s (object repositories) of that particular page/functionality.

**Step-2**

After building OR’s (object repositories), again Left click on project and create a new class with methods abbreviation. Methods will include all the functions that will be use to perform certain actions mainly Keyboard & Mouse events such as **C**lick, **SendKeys, DoubleClick etc.**

**Step-3**

**Just Like step-1 and step-2, we have to create a class with test at the end of the name. Test class** Comprises of the business logic of the corresponding web page.

how to create test suite using xml

You can specify test suites in **.xml** or **.yaml** files. If you are going to use YAML, make sure to add the corresponding dependency to the build file.

1. Right-click the project root folder in the Project tool window, select New | File and enter the name of the file, for example **testing.xml**.
2. Fill in the file with information about your tests. For more information, refer to testing.xml.

how to create test suite using yaml

TestNG supports defining a test suite using YAML, however it doesn't contain the YAML parser by default. If you want to use a YAML file, add the snakeyaml dependency. The version should correspond with the TestNG version that you use.

1. Open the build file (**pom.xml** or **build.gradle**), press Alt+Insert, and select Add dependency.
2. In the tool window that opens, type snakeyaml in the search field.

Locate the necessary dependency, select its version in the search results, and click Add next to it.

1. Press Ctrl+Shift+O or click the notification that appears in the top-right corner of the editor to load the changes.
2. Right-click the project root folder in the Project tool window, select New | File and enter the name of the file, for example **testing.yaml**.

How to run testng tests

Run TestNG tests:﻿

* To run an individual test, click Run in the gutter and select Run.
* To run all tests in a test class, click Run against the test class declaration and select Run.
* You can view test results in the Run tool window.

How to run test suite

**﻿**

To be able to run a TestNG test suite, create a run configuration for this suite:

1. From the main menu, select Run | Edit Configurations.
2. In the left-hand pane, click Add New Configuration and from the list that opens, select TestNG. Name the new configuration.
3. From the Test kind list, select Suite.
4. The Suite field becomes available.
5. In the Suite field, click Browse and specify the path to the XML or YAML file in which the test suite is configured.
6. In the Output directory field, you can specify a folder in which the IDE will place output files, such as test reports.
7. By default, the IDE creates a new test-output folder in the project root directory, but you can use another folder or use different folders for different outputs.
8. Apply the changes and close the dialog.
9. On the toolbar, make sure that the newly created run configuration is selected and click Run next to it. Alternatively, press Shift+F10.

How to configure git