

[seleniumeasy.com](https://www.seleniumeasy.com)

# Selenium Automation Framework Example

## Introduction to Automation testing:

Testing is an essential part of a software development process. While testing intermediate versions of products/projects being developed, testing team needs to execute a number of test cases. In addition, prior to release every new version, it is mandatory that the version is passed through a set of “regression” and “smoke” tests. Most of all such tests are standard for every new version of product/project, and therefore can be automated in order to save human resources and time for executing them.

## Benefits of using automated testing are the following:

- Reduction of tests' time execution and human resources required
- Complete control over the tests' results (“actual results” vs “expected results”)
- Possibility to quickly change test's preconditions and input data, and re-run the tests dynamically with multiple sets of data

**Automation workflow for the application can be presented as**

**follows:**

- First of all it is required to identify tasks that an application has to accomplish.
- Second, a set of necessary input data has to be created.
- Third, expected results have to be defined in order one can judge that an application (a requested feature) works correspondingly.
- Fourth, Executes a test.
- Finally, Compares expected results with actual results, and decides whether the test has been passed successfully.

**Environment Specifications:**

1. Selenium Webdriver (Supports all major browsers, we use Mozilla, chrome and IE)
2. Eclipse IDE
3. Java
4. TestNG
5. AutoIT Tool (Used to handle Windows popups for Document [Uploads](#) and [Downloads](#).)
6. [JExcel](#) or [Apache POI](#) to perform operations with excel like read,

write and update the excel sheet

## **This Framework has the following tools:**

**1. Selenium** - Selenium is a well know open source testing framework, which is widely used for testing Web-based applications. It has different components and in that Webdriver has rendered the Selenium Remote Control obsolete, and is commonly referred to as Selenium 2.0.

Selenium Webdriver supports most of all browsers to run your test cases and many programming languages like C#, Java, Python, Ruby, .Net, Perl, PHP, etc.. to create and modify your test scripts.

**2. Eclipse IDE:** Eclipse is an integrated development environment (IDE) for Java. The Eclipse IDE is the most known product of the Eclipse Open Source project.

**3. [TestNG](#)** - Is a testing framework inspired from JUnit and NUnit. It has extended new functionalities which made it more powerful and easier than the other testing frameworks.

It supports [ReportNG \(simple HTML reporting plug-in\)](#) and XLST (Graphical / Pictorial reports) plug-ins to customize or extend the default TestNG reporting style.

TestNG also provides ability to implement 'IReporter' an interface which can be implemented to generate a [Customized TestNG report](#) by users. It has 'generateReport()' method which will be invoked after all the suite has completed its execution and gives the

report into the specified output directory.

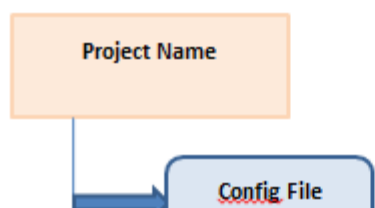
4. [AutoIT](#) - AutoIT v3 is a freeware BASIC-like scripting language designed for automating the Windows GUI and general scripting. It uses a combination of simulated keystrokes, mouse movement and window/control manipulation in order to automate tasks which is not possible with selenium.

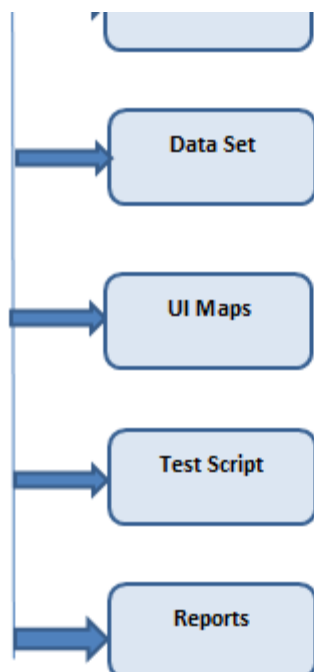
You can also refer the other frameworks available here [Keyword Driven Framework](#) and [Page Object Model Framework](#)

### File Formats Used in the Framework:

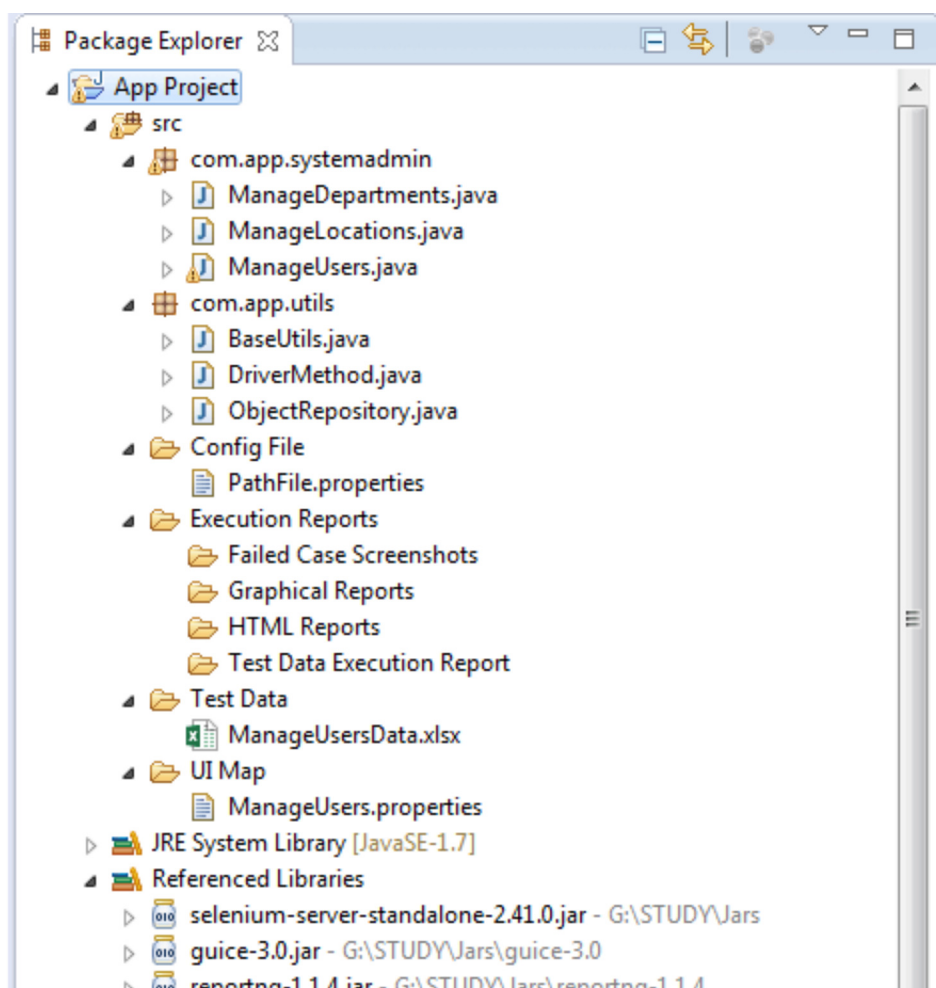
1. [Properties file](#) – We use properties file to store and retrieve the UI elements of an application or a website and data set file paths. It contains id of the elements, name, xpath or Css selector etc.
2. *Excel files* – Excel files are used to pass multiple sets of data to the application.
3. *Xml file* – Is used to execute the test scripts. Based on the package or classes or Tests mentioned in the xml file scripts will be executed.

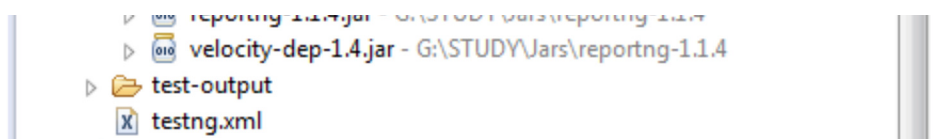
The following figure explains physical structure of files required for [Test Automation Framework](#)





**The Project Folder Structure:** All the basic required folders are created with the sub folders and classes under each folder:--





## The Following explains the structure in detail:-

### 1. [UI Map / Object Repository](#)

UIMap is a concept for defining, storing, and serving UI elements of an application or a website. The UIMap properties file contains a set of 'key-value' pairs, where key is an alias of the UI element, and a value is the locator. [Click here for more..](#)

### 2. [Data Set / Test Data](#)

Data set stores the data files, Script reads test data from external data sources and executes test based on it. Data sets increases test coverage by performing testing with various inputs and reduce the number of overall test scripts needed to implement all the test cases. [Click here for more..](#)

### 3. [Test Automation Scripts](#)

A test is considered as a single action or a sequence of actions, that defines whether a specific feature meets functional requirements. It has multiple test files / packages / class files which will be executed based on the configurations defined in testng.xml. [Click here for more..](#)

### 4. [Reports / Executed Results](#)

Test report/results is a document which contains summary of test activities. After execution is completed, it is very important to communicate the test results and findings to the project manager along with the [screenshots for failed tests](#) and with that decisions can be made for the release. [Click here for More..](#)

## 5. [TestNG xml file](#)

In order to create a test suite and run separate test cases, we need framework which drives the automation. Here testng.xml can be called as "driver" which drives several test cases automated using selenium code. Advantage of using TestNG with Selenium is of running multiple test cases from multiple classes using xml configuration file .[Click here for More..](#)