







**Data Driven Framework**

Data Driven Framework is one of the popular Automation Testing Framework in the current market. Data Driven automated testing is a method in which the test data set is created in the excel sheet, and is then imported into automation testing tools to feed to the software under test.

Selenium Webdriver is a great tool to automate web-based applications. But it does not support read and write operations on excel files.

Apache POI (Poor Obfuscation Implementation) is an API written in Java to support read and write operations – modifying office files. This is the most common API used for Selenium data driven tests. .xls .xlsx

**Why data drive tests?**

Often there might be may be a number of data sets that have to be used to test a feature of an application. Now running the same test with different data manually is time-consuming, error prone and a boring task.

***Let us understand this scenario with an example*.**

Suppose we need to test the Any form with multiple input fields with 100 different data sets.

**To test this you have three different approaches:**

**1)** Create 100 scripts one for each dataset and execute each test one by one.  
**2)** Change the data in the script and execute it multiple times.

**3)** Import the data from the excel sheet and execute the script multiple times with different data.

First two scenarios are laborious, time-consuming – implying low ROI. Hence, we must follow the third approach.

In the third approach, are implementing the Data Driven framework, where all our data resides in an excel sheet, where it is imported from and used to test the features of the application.

# **Java Array**

Normally, array is a collection of similar type of elements that have contiguous memory location.

Java array is an object the contains elements of similar data type. It is a data structure where we store similar elements. We can store only fixed set of elements in a java array.

Array in java is index based, first element of the array is stored at 0 index.



Following code snippet will show you the declaration of two dimensional array in [Java Programming](https://www.tutorialgateway.org/java-tutorial/) Language:

|  |  |
| --- | --- |
| 1 | Data\_Type[][] Array\_Name; |

* **Data\_type:** This will decide the type of elements it will accept. For example, If we want to store integer values then, the Data Type will be declared as int, If we want to store Float values then, the Data Type will be float etc
* **Array\_Name:**This is the name you want to give it to array. For example Car, students, age, marks, department, employees etc

## Create Two dimensional Array in Java

In order to create a two dimensional array we have to use the New operator as we shown below:

|  |  |
| --- | --- |
| 1 | Data\_Type[][] Array\_Name = new int[Row\_Size][Column\_Size]; |

If you observe the above code snippet,

* **Row\_Size:** Number of Row elements an array can store. For example, Row\_Size = 5 then array will have 5 rows.
* **Column\_Size:** Number of Column elements an array can store. For example, Column\_Size = 6 then array will have 6 Columns.

## Accessing Two Dimensional Array Elements

In Java programming, We can use the index position to access the two dimensional array elements. Using index we can access or alter/change each and every individual element present in a two dimensional array . Index value starts at 0 and end at n-1 where n is the size of a row or column. For example, if an int[][] Array\_name = new int[6][4] will stores 6 row elements and 4 column elements. To access or alter 1st value use Array\_name[0][0], to access or alter 2nd row 3rdcolumn value then use Array\_name[1][2] and to access the 6th row 4th column then use Array\_name[5][3].

**Interface in POI**

One of the most remarkable features of **Apache POI** is that it supports read and write operations on both .xls and .xlsx files.

Steps to create Data Driven Test

Step 1: Download Apache POI.zip and extract the files.

Step 2: Add all the below list of jar file to the project

* dom4j-1.6.1.jar
* poi-3.10-FINAL-20140208.jar
* poi-ooxml-3.10-FINAL-20140208.jar
* poi-ooxml-schemas-3.10-FINAL-20140208.jar
* xmlbeans-2.3.0.jar

Step 3: Create and Excel File and add data to the sheet

Step 4: Write a program having functionality where you will be importing data into the required fields.

## **What is Keyword Driven Framework?**

Keyword Driven Framework is a type of**Functional Automation Testing Framework** which is also known as **Table-Driven** testing or **Action Word based** testing. The basic working of the Keyword Driven Framework is to divide the Test Case in to four different parts. First is called as Test Step, second is Object of Test Step, third is Action on Test Object and fourth is Data for Test Object.

### The above categorization can be done and maintained with the help of Excel spread sheet:

**Test Step:** It is a very small description of the Test Step or the description of the Action going to perform on Test Object.  
**Test Object:** It is the name of the Web Page object/element, like Username & Password.  
**Action:** It is the name of the action, which is going to perform on any Object such as click, open browser, input etc.  
**Test Data:** Data can be any value which is needed by the Object to perform any action, like Username value for Username field.

The idea behind the Keyword Driven approach in automation is to separate the coding from the test case & test step.

**What is Hybrid Driven Framework?**

A combination of more than one framework and not specific to only data driven + Keyword driven only. It can be a combination of any two or more types of frameworks. When a project is designed with the help of using Page Object Model or Object Repository with a combination of Data Driven, Test Driven, Keyword Driven, Behaviour Driven, Modular Data driven, and many more… this combination makes the success of the project and both the types of framework are integrated with each other to make a smooth flow of execution process which leads to success of the project.

In laymen terms it is the combination of all the frameworks (Keyword + Data + Modular), which takes the best of all of the available frameworks to create a robust approach for automation.

Thus hybrid approach takes the features of all the available frameworks plus the user specific modification to achieve an easier route for automation testing of the application.

**Note: -** There is no standard definition or guidelines for Hybrid approach and its components are the vision of the automation tester to make automation task least maintainable, easy to implement and adapt.

Advantages:

The Hybrid framework is build with a number of reusable modules / function libraries that are developed with the following features in mind:

**Maintainability –** Hybrid framework significantly reduces maintenance effort

**Re-usability –** It allows to reuse test cases and library functions

**Manageability -** effective test design, execution, and traceability

**Accessibility –** easy to design, develop, modify and debug test cases while executing

**Availability –** Allows to schedule automation execution

**Reliability –** due to advanced error handling and scenario recovery

**Flexibility –** framework independent of system or environment under test

**Measurability –** customizable reporting of test results ensure the quality output

**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.

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.

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.

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.

**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

## **What is Maven?**

Maven is a build tool and it performs task just like Ant which is again a different build Tool. It is a software project management tool which provides new concept of project object model (POM). Maven allows the developer to automate the process of the creation of the initial folder structure, performing the compilation and testing and the packaging and deployment of the final product. It cuts down the good number of steps in build process and it makes it one step process to do a build.

## **Why Maven is Used?**

As I said above it cuts down the tasks in build process. To summarize, Maven simplifies and standardizes the project build process. It handles compilation, distribution, documentation, team collaboration and other tasks seamlessly. Maven increases reusability and takes care of most of build related tasks. It helps in bypassing my steps like adding jars to the project library, building reports, executing Junits test cases, creating Jar, War Ear files for the project deployment and many more. A very significant aspect of Maven is the use of repositories to manage jar files.

Sikuli: - <https://launchpad.net/sikuli/sikulix/x1.0-rc3>

Jenkins url: - <http://localhost:8080/>