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 into 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. This helps a non technical person to understand the automation very well. With this a manual tester can write Automation scripts. This does not mean that an automation tester is not needed but in any Keyword Driven project, a hard core technical coder is must needed for setting up the framework and to work on regular changes and updates of background automation code. But for an example an automation team can have two manual tester and a automation coder.

Let just understand this concept with the help of an example of our ***Demo Online Store*** Application. Consider that you have to automate a flow where you would need to do the following things:

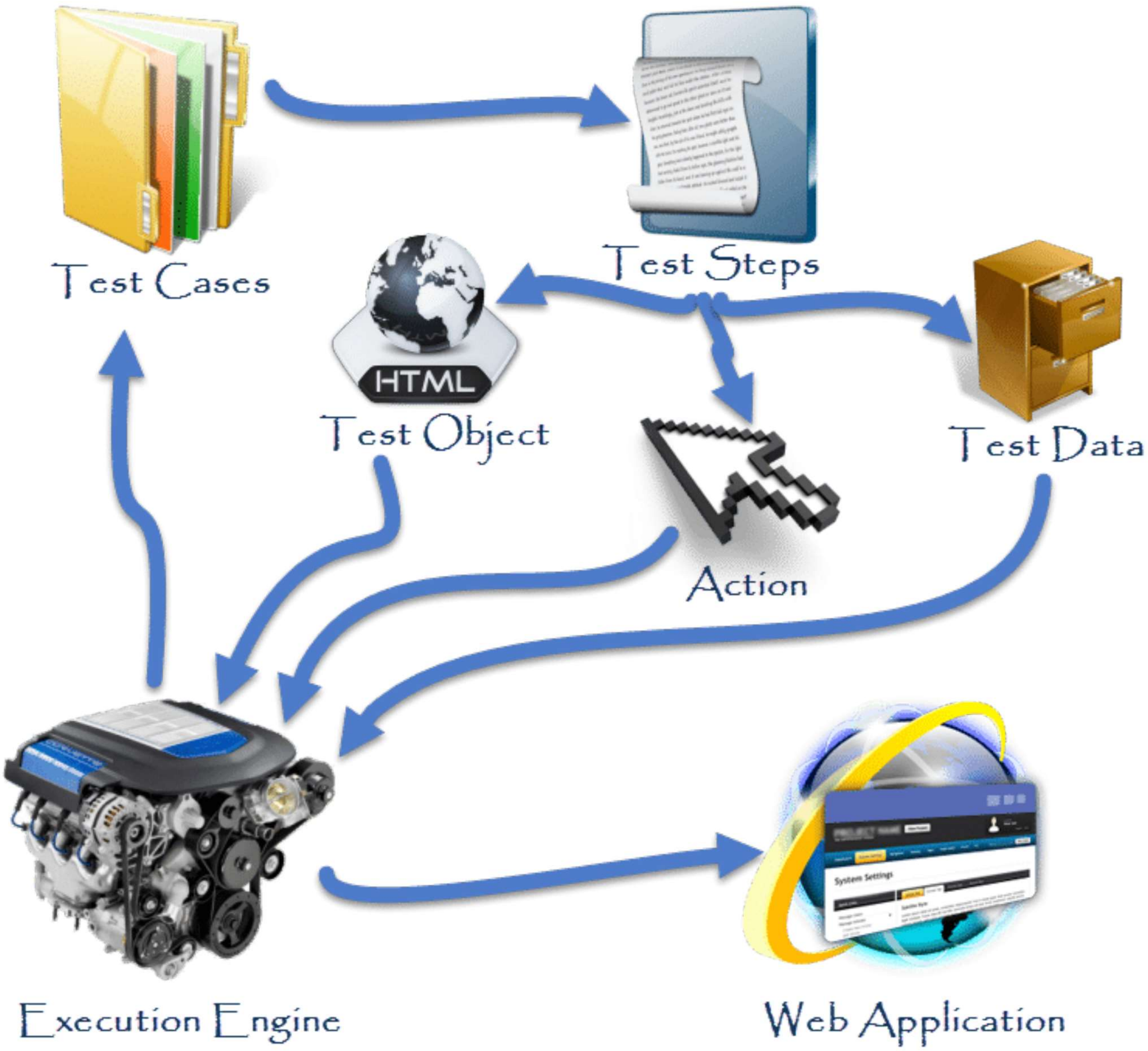
. Open a Browser
. Navigate to URL
. Click on My Account button
. Enter Username
. Enter Password

- Click on LogIn button
- Click on LogOut button
- Close the Browser

Below is the list of common components that you would required to achieve the task:

- ***Excel Sheet:*** This is the sheet which keeps most of the data for Keyword Driven which is needed for the test like Test Case, Test Steps, Test Objects and Actions.
- ***Object Repository:*** Property file to store the html elements properties of the web application, this property file will be linked with the Objects in the test
- ***Keyword Function Library:*** In keyword Driven Framework function file plays an important role, as it stores the working of the actions, so that each action can be callable from this file
- ***Data Sheet:*** Excel file to store the Data value needed by the object to perform some action on it
- ***Execution Engine:*** Test is the only test script we have in Keyword Framework and it contains all the code to drive the test from Excel sheet, Function Lib and Properties file.

The below image shows the generic work-flow of Keyword Driven Framework:



From the above figure, you can see that the generic flow of a *Keyword Driven Framework* is something like this -

Execution Engine starts the test and connect with the bundle of ***test cases*** and start executing one by one
Once Test Case is picked, linked ***test steps*** are followed sequentially
Test Steps are further connected with ***Page Objects***, ***Actions*** & ***Test Data***
Once Execution Engine gets all the required info to perform a test step, it connects with application and do the step.

Advantages:

. ***Less Technical Expertise:*** Once Framework is set up, manual testers or non technical testers can easily write test scripts for automation.

- ***Easy To Understand:*** As it is maintained in Excel sheet and no coding is exposed, the test scripts are easy to read and understand. Keywords & actions resemble so closely manual test cases, which become easier to write and maintain.
- ***Early Start:*** You can start building Keyword Driven test cases before the application is delivered, as Object Repository can be easily set up at the later stage. Using information gathered from Requirements or other documentation, keyword data tables can be created that mirror corresponding manual test procedures.
- ***Re-usability of component:*** With implementing modularization in Keyword Driven, Re-usability can be further increased.
- ***Re-usability of code:*** As there is just one Execution Engine in Keyword Driven Framework, it encourage extreme code re-usability.