

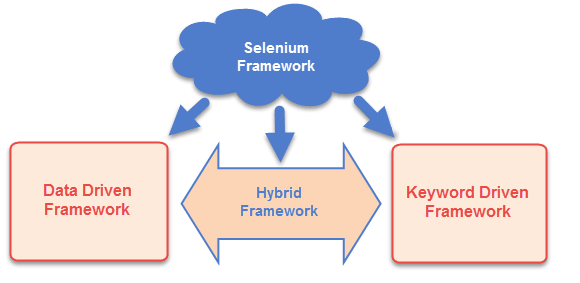
|  |  |  |  |
| --- | --- | --- | --- |
| **Revision History** | | | |
| **Version** | **Date** | **Auteur** | **Description** |
| 001 | 15/01/17 | Gregory Amirthanathan | First version  Keyword Driven Framework |

**Framework with Selenium**

* Frameworks help to structure our code and make maintenance easy.
* Without frameworks we will place all our code and data in same place, which is neither re-usable nor readable.
* Using Frameworks, produce beneficial outcomes like increase code re-usage, higher portability, reduced script maintenance cost etc.

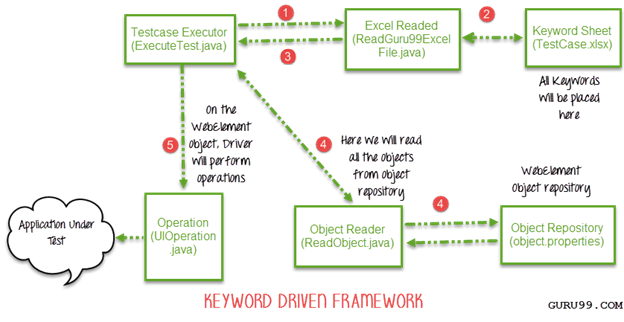
There are mainly three type of frameworks created by Selenium WebDriver to automate manual test cases,

* Data Driven Test Framework
* Keyword Driven Test Framework
* Hybrid Test Framework



**Keyword Driven Test Framework:**

In keyword driven test framework, all the operations and instructions are written in some external file like excel worksheet. Here is how the complete framework looks like



As you can see it's a 5-step framework. Let's study it stepwise in detail

Step 1)

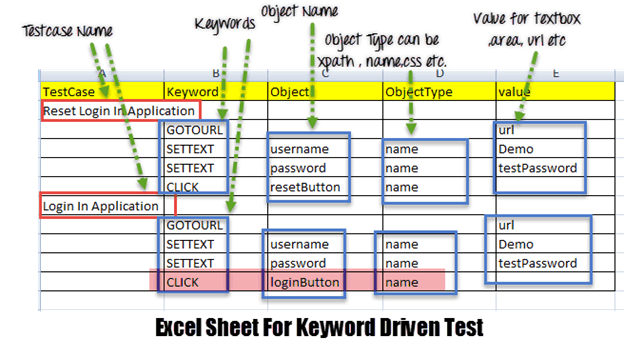
The driver script Execute.java will call ReadGuru99ExcelFile.java

ReadGuru99ExcelFile.java has POI script to read data from an Excel

Step 2)

ReadGuru99ExcelFile.java will read data from TestCase.xlsx

Here is how the sheet looks like-



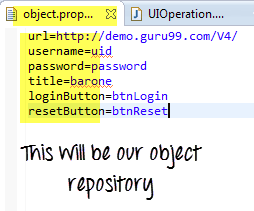
According to the keywords written in excel file, the framework will perform the operation on UI.

For example, we need to click a button 'Login'. Correspondingly, our excel will have a keyword 'Click'. Now the AUT can have hundreds of button on a page, to identify a Login button, in excel we will input Object Name as loginButton & object type as name (see highlighted row in above image). The Object Type could be Xpath, name, css or any other value

Step 3) ReadGuru99ExcelFile.java will pass this data to the driver script Execute.java

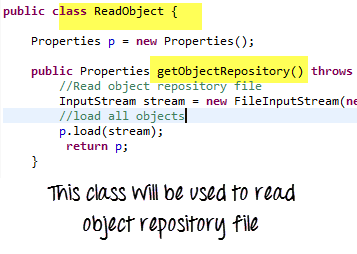
Step 4)

For all of our UI web elements we need to create an object repository where we will place their element locator (like Xpath, name, css path, class name etc.)



Execute.java (our driver script) will read the entire Object Repository and store it in a variable

To read this object repository we need a Read Object class, which has a getObjectRepository method to read it.



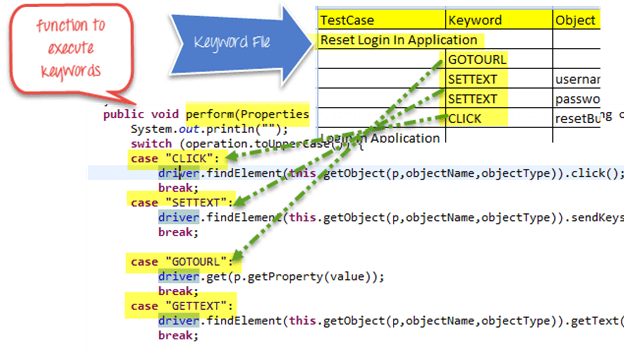
NOTE: You may think why do we need to create an object repository. The answer it helps in code maintenance. For example, we are using the button with name = btnlogin in 10 different test cases. In future, the developer decides to change the name from btnlogin to submit. You will have to make change in all the 10 test cases. In case of an object repository, you will make change just once in the repository.

Step 5)

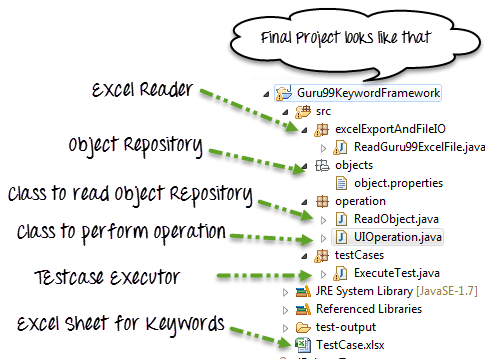
The driver will pass the data from Excel & Object Repository to UIOperation class

UIOperation class has functions to perform actions corresponding to keywords like CLICK, SETTEXT etc… mentioned in the excel

UIOperation class is a java class which has the actual implementation of the code to perform operations on web elements



The complete project will looks like-



**Data Driven Framework with Apache POI – Excel**