

Framework document

Definition: Framework is a well-organized structure of reusable components where one driver (.xml) file will take care of the execution without any manual intervention.

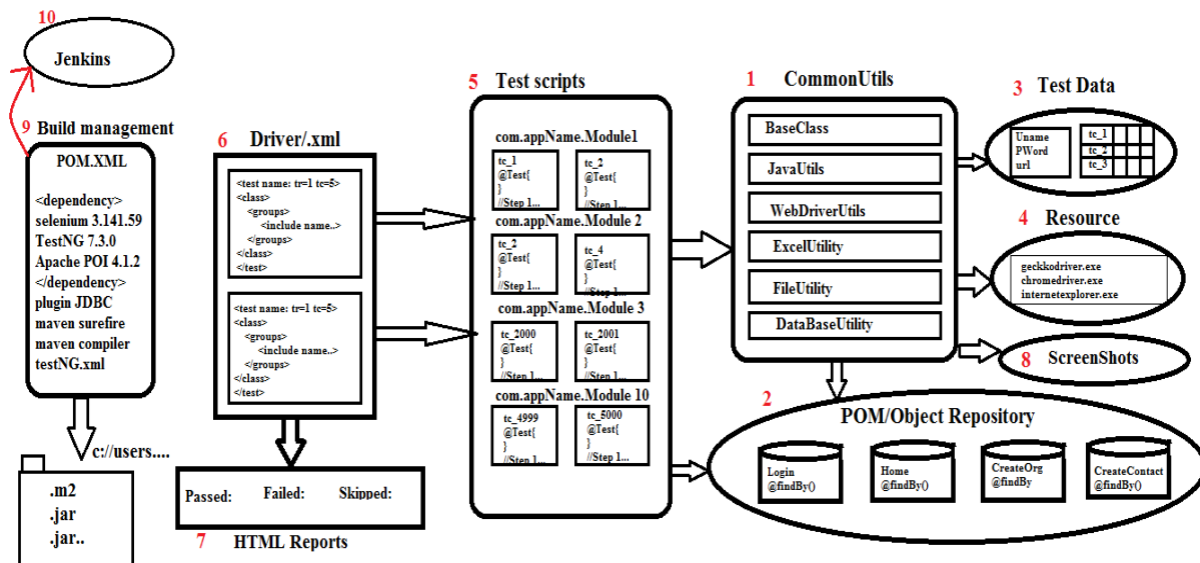
Definition: Framework is collection of reusable components that makes automation development execution and modification is easier and faster.

Definition: Framework is a set of instruction followed by every organization that makes automation test engineer life easy.

Framework has the following basic components:

1. Common Utils
2. POM/Object Repository
3. Test Data
4. Resource
5. Test Scripts
6. Run Me/driver/XML file
7. HTML Report
8. Screen Shots
9. Maven
10. Jenkins

Framework Architecture is depicted in the figure below.



1. **Common Utils:** It is one of the common components in my framework which can be used to any project. It contains several reusable classes like

- a. Base class: It contains common TestNG configuration annotations which is required for all the test script. As per the automation rule every test script should extend base class to use those annotations.

Real usage of testNG annotations:

- **@Before suite:** it is used to configure database connectivity and extend report, it will execute before <test> tag in xml file.
- **@Before test:** it is used to launch the browser in case of parallel execution before executing each <test> tag in xml file.
- **@Before Class:** it is used to launch the browser in sequential execution, it will get executed before executing each class
- **@Before Method:** it is used to login into application which is executed before executing every @test method
- **@After method:** it is used to logout into application which is executed after executing every @test method
- **@After Class:** it is used to close the browser in sequential execution; it will get executed after executing each class
- **@After suite:** it is used close the database connectivity.

- b. Java Utils: It contains java specific methods which can be used for all the test scripts like get-Random-number, get-Date etc.

- c. Web Driver Utils: It contains web-drivers action which is common for all the test scripts like select, switch-To-Window, switch-To-Frame etc., in real framework we use web driver utility to perform action on the browser so that the test script level is easy and no need to remember the syntax.

- d. Excel Utility: It is developed using Apache POI which is used to read and write data from Excel sheet. As per the rule of the automation, the data should not be hard coded in the test script so that we will take the help of Excel utility to interact with Excel file.

- e. File Utility: It is used to capture common data from Property file which is developed using java.

- f. Database Utility: This is developed using JDBC which is used to connect to any database from the automation test script. In real time few scenarios require to connect to database for validation to execute precondition.

- g. Listener Implementation: It is a implementation class of TestNG ITest Listener which is used to capture failure event in run time and capture the Screenshot in screenshot folder.
 - h. Retry Listener: It is an implementation class of testNG IRetry Analyser which is used to execute same script multiple times when ever test script is failed.
 - i. IConstants: It contains common global variable which is used across the framework in Implicit-Wait, Excel-Path, Property-Path, Explicit-wait.
2. **POM/ Object Repository**: It is a collection of reusable web elements and business library which can be used to specific business or project. In order to develop we have used POM design pattern as per the rule automation element locator should not be hard coded in test script because maintenance and modification is tedious job when ever requirement is changed. In real time most of the organization follow Agile process , where frequent requirement changes should be handled, in such cases POM repository helps us to maintain and modification becomes easier.
 3. **Test data**: It is one of the components in the framework it contains the data which is required to run the framework. There are two types of data:
 - a. Common data:
 - i. URL: used to run test scripts in different environment
 - ii. User Name / Password: which is used run all the data in different Credentials
 - iii. Browser: used to run all the test script in different browsers
 - b. Test Script Data: data which is required for all the test script. Every test script is dedicated with one row with in Excel file. As per the rule of automation data should be externalized because modification maintenance is easier
 4. **Resource**: It is one of the components in the framework which contains resources to run the Framework like IE-Driver-Server, Chrome-Driver-Server, Gecko-Driver-Server, User-Guide Document.
 5. **Test scripts**: It contains collection of TestNG test scripts which is automated using @Test. During test script development make sure generic libraries, object repository is being used. In real time when ever manual test cases is allocated to us we should create a separate package for a module and develop a test script by taking utility

6. **Driver/XML File:** It is a TestNG component used to execute all the test scripts in batch, parallel, group. In real time whenever we get a new build we should create testNG.xml and trigger the execution.
7. **HTML Report:** Whenever TestNG execution is completed it automatically generates HTML report which helps us to know the status of application. In Real time reporting component is very much important because it provides the quality of application and same report is being submitted to customer. In real time report helps the engineer to know the root cause of the issue whenever test Script is getting failed.
8. **Screen Shots:** Whenever any test script is failed during execution we will have screenshot in screen shot folder, same screenshot can be used for debugging and rise defect in JIRA.

Advantages of Framework:

1. Test script development is faster and easier because of reusability
2. Modification and maintenance of data is easy because data is stored in external resource
3. Modification and maintenance of element is easy because we have used POM design pattern to maintain then elements in well-organized way
4. POM design pattern is the perfect fit for agile Process
5. Framework provide automatic screenshot for failed script
6. Framework provides flexibility to achieve cross browser testing, distributed environment testing, smoke testing, regression testing, regional regression testing
7. Framework provide accurate execution report for every execution
8. Framework provides generic reusable utility for all actions like Excel Utility, base class, database utility.
9. Test script can be re used for every new build
10. Test script is optimized

Disadvantages of Framework:

1. Should be good in programming

Phases of Framework: There are 3 phases in Framework

1. Framework design Phase: In this phase framework developer will design the framework which contain common utility like generic libraries, test data template, POM classes with partial elements. This phase is executed in Sprint-1 or Release-1
2. Framework Implementation Phase: This phase start from Sprint-2, all automation engineer participate in test script development implementation
3. Framework Execution phase: Execution phase is always handled by Jenkins tool, that is whenever we get a new build to testing environment JENKINS will automatically execute and send email to the concerned engineer