

FRAMEWORK

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

(OR)

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

ADVANTAGES OF FRAMEWORK

- Test script development is faster and easy because of reusability.
- Improves test efficiency.
- Optimize test scripts.
- Modification and maintenance of data is easy.
- It supports multiple executions like batch execution, regional regression, distributed parallel execution and cross browser execution.
- It generates accurate test reports for each execution.
- It provides automatic screenshot for failed scripts.

TYPES OF AUTOMATION FRAMEWORK

DATA DRIVEN FRAME WORK

- Reading data from any external resources like properties file, excel, Jason, database, TestNG(.xml), .cmd(Maven) is called Data driven frame work.

MODULAR DRIVEN FRAME WORK

- Maintaining the Test scripts, Test data and Suite xml file in module wise in order to make the debugging process easy is called Modular driven frame work

METHOD DRIVEN FRAME WORK

- Developing the reusable methods for all repetitive action functionalities in the application and calling those methods in Test scripts is called Method driven frame work.

HYBRID DRIVEN FRAME WORK

- Combination of two or more frame work is called Hybrid driven frame work.

KEYWORD DRIVEN FRAME WORK

- Creating a keyword library and utilizing these keywords to develop test scripts is called Keyword driven frame work.

LINEAR (OR RECORD-AND-PLAYBACK) FRAMEWORK

A Linear Framework is the simplest type of Selenium test automation framework.

- It is also called a record-and-playback or scripted framework.
- The test scripts are written linearly, meaning step-by-step in a sequential manner.

Key point: Each test script is independent and contains all the test steps, including browser setup, navigation, actions, and validations.

Features

- Simple and easy to implement.
- No modularization: Everything is written in a single script.
- Suitable for small projects.
- No reuse of code: If a change occurs, you need to update each script individually.
- No integration with external data sources (Excel, CSV, DB) in basic form.

How It Works

- Open the browser.
- Navigate to the application.
- Perform test steps (click, type, select, etc.).
- Validate the result.
- Close the browser.

Advantages

- Easy to create and understand.
- Quick for small, one-time test scripts.
- No need for advanced programming skills.

Disadvantages

- Not scalable for large projects.
 - Code duplication if multiple tests use the same steps.
 - Difficult to maintain: Changes in UI require updating all scripts.
 - No separation of test data and scripts.
 - Cannot handle data-driven or keyword-driven testing.
-

MODULAR FRAMEWORK

A Modular Framework is an automation framework in which the application under test is divided into separate logical modules, and each module is automated as an independent script.

- Each module contains specific functions related to that part of the application.
- These modules are reusable across multiple test cases.

Key point: It emphasizes reusability by creating functions for repetitive actions.

How It Works

1. Divide the application into modules (e.g., Login, Registration, Checkout).
2. Create functions for each action in the module (e.g., login_user(), add_item_to_cart()).
3. Write test scripts by calling these modular functions.

Advantages

- Reusability: Common functions can be reused in multiple test scripts.
- Maintainability: Changes in one module do not affect others.
- Scalable: Suitable for medium to large projects.
- Reduces code duplication.
-

Disadvantages

- Requires programming knowledge to create reusable functions.

- Initial setup time is higher than Linear Framework.
- Still not fully data-driven; test data is often hardcoded unless combined with Data-Driven approach.

DATA DRIVEN TESTING

What is Data Driven Testing?

Reading the data from external resource & run the test is called Data driven testing (parameterization)

Why Data Driven Testing?

As per the rule of the automation data shouldn't not hardcoded(fixed) with in a test scripts, because data modification & maintenance is tedious job when you want to run the test with different data, instead we should get the data from external resource like xlsx, .properties file, db, XML, JSON, CMD Line Data

What are the advantages of Data Driven Testing?

1. Maintenance of the test data is easy
 2. Modification of the test data in external recourse is easy
 3. Cross browser /platform testing is easy (means change the browser in property File)
 4. Running test scripts in different Environment is easy
 5. Running test scripts in different credentials is easy
-
-

PAGE OBJECT MODEL

Why Page Object Model (POM)?

When we automate web applications using Selenium, we often deal with multiple web pages, each having many elements. Without a proper structure:

- Test scripts become **long, hard to read, and difficult to maintain**.
- Locators are scattered across scripts; if the UI changes, we must update each script individually.

POM solves this by separating the page structure from the test logic.

What is Page Object Model (POM)?

Page Object Model (POM) is a design pattern in Selenium that encourages:

- Creating a separate class for each web page.
 - Storing locators (WebElements) and page actions (methods) in that class.
 - Writing test scripts that interact with the page through these page classes, not directly with WebDriver.
-

Page Object Model (POM) Architecture

POM Architecture typically consists of three layers:

1. Test Layer (Tests)
 - Contains test scripts that perform actions on the pages.
 - Should not contain locators or UI logic.
 2. Page Layer (Page Classes)
 - Contains WebElements (locators) and methods to interact with them.
 - One class per web page.
 3. Utility Layer (Optional)
 - Contains reusable helper functions, like waiting, reading config, logging, screenshots, etc.
-

Advantages of Page Object Model (POM)

Advantage	Explanation
Reusability	Page classes can be used across multiple test scripts.
Maintainability	Only page classes need to be updated if UI changes.
Readability	Tests are more readable, showing “what” is tested, not “how”.
Separation of Concerns	Keeps test logic separate from page structure.
Scalability	Supports large projects efficiently.

5. Overview of Selenium Design Patterns

- Page Object Model (POM): Encapsulates page elements and actions.
- Page Factory: An extension of POM that uses `init_elements()` in Python to initialize WebElements automatically.
- Singleton Pattern: Ensures only one WebDriver instance exists.
- Factory Pattern: Creates objects without exposing creation logic to the test.

Design patterns help in writing scalable, maintainable, and reusable automation frameworks.

6. Importance of Design Patterns in Selenium Automation Testing

- Reduces code duplication
- Improves readability
- Simplifies maintenance when application UI changes
- Encourages modular testing

In essence, design patterns make automation projects professional and enterprise-ready.

GENERIC UTILITIES

What is Generic components in Automation Framework?

- It's one of the automation framework components which is common for all the application
- Its collection of generic class contains reusable methods / libraries
- The methods which can be used to any application is called Generic/common methods

What are the advantages of Generic Components?

- Reusability of code
- Code Optimization
- Test script development is faster
- Test Code readability
- Generic libraries are common to all automation projects
- Avoid duplicate Code
- No need to remember the syntax every time , just create once & use multiple times

Excel Utility libraries

As per the rule of automation, data should not be hardcoded with in the test scripts, so that to get the data from external file like Excel & .properties file We go for ExcelUtility & FileUtility

We use `xlrd` to read the data from the excel file

File Utility Libraries

FileUtility is used to get the data from .properties file