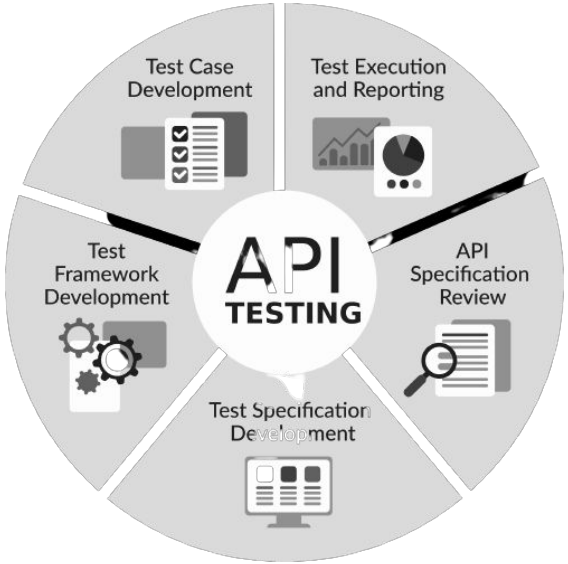
Rest Assured Basics



Pramod Dutta
Lead SDET.

Coding + Rest Assured

Agenda

- Best Practices For Writing Automated Testing Code.
- Jenkins Pipeline Job
- Jenkins Other Concepts
- Overview of Automation Framework.
- Folder Structure For API Testing Framework
- TestNG Demos

Best Practices For Writing Automation Test Code

- Follow Programming Language Guidelines(method name-getPayload(), class name- API Endpoints, package name-com.testingsumo.backend.tests, variable name-authToken

- Follow Oops Concepts Wherever Possible- Abstraction(base classes), Inheritance(multiple implementation of same things/multiple inheritance), Polymorphism(many forms with something different), Data Hiding(hide unnecessary/sensitive info), Encapsulation(Bind small entities into a single large entity)

Best Practices For Writing Automation Test Code

- Reduce code duplicacy (think before writing new code, can i use/make change in existing code?)
- Increase code reusability
- Make your code generic wherever possible
- Leave no hardcoded data in source code
- Keep your static data outside the source code
- Keep your dynamic data dynamic in testcode (fetch it from db queries or scripts)
- Test your code properly, use IDE options such as call heirarcy or show usage to test your changes end 2 end

Best Practices For Writing Automated Testing Code Cont...

- Use Extensive logging- everything which is part of source code should be analyzed from logs without looking at the source code
- Generate and save failure proofs outside the src code- videos/data/screenshots/logs
- Focus on making your code scalable and faster without compromising the code quality
- Your code should be platform and system independent
- Use as many assertions as possible focus on automated testing rather than automation

Best Practices For Writing Automated Testing Code Cont...

- Leave no hardcoded data in source code
- Always think for the future, separate out tech dependencies so that migration to new tech is easy in case it is needed
- Keep your tests independent for better results in multithreading unless they are related (example publisher subscriber related tests)
- Use Proper Documentation
- Create code which is can be easily read and modified by others

TestNG Demos

- Annotations
- Group Run
  - Include
  - Exclude
- Listeners
- ReportNG
- Priority
- Allure Report
- TestNG – Parallel Test Execution

TestNG Demos

- Annotations
- Priority
- Param
- Assertions

TestNG Assertion

Assertion in TestNG is way to verify expected value with actual. If values are not same it will mark test fail.

Assertion Type:

1. Hard Assertion
2. Soft Assertion

Hard Assertion: Hard assertion stop the execution when assertion fail, and remaining steps of test will not execute. Hard assertions are default assertion in TestNG.

Soft Assertion: Soft assertions are opposite of hard assertion, it will not stop the execution of test even when assertion fail.

Listeners

Listeners are TestNG annotations that literally “listen” to the events in a script and modify TestNG behaviour accordingly

IAnnotationTransformer
IExecutionListener
IHookable
IInvokedMethodListener
IMethodInterceptor
IReporter
ISuiteListener
ITestListener

Listeners

Demo of IExecutionListener

CustomListener.java

```java
package com.thetestingacademy.testng.Listener;

import org.testng.IExecutionListener;

public class CustomListener implements IExecutionListener {
    @Override
    public void onExecutionFinish() {
        long endTime= System.currentTimeMillis();
        System.out.println("****   *** Finished execution at- "+ endTime +"

    }

    @Override
    public void onExecutionStart() {
        long startTime= System.currentTimeMillis();
        System.out.println(" ****   *** Started execution at - "+ startTime

    }
}
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="All Test Suite">
    <listeners>
        <listener class-name="com.thetestingacademy.testng.Listener.CustomLis
    </listeners>
    <test name="LearnTestNG">

        <groups>
            <run>
                <include name = "sanity"></include>
<!--                <exclude name="smoke"></exclude>-->
            </run>
        </groups>
        <classes>
            <class name="com.thetestingacademy.testng.Groups"></class>
        </classes>
    </test>
</suite>
```

Data Provider CSV File

```
<dependency>

<groupId>com.codoid.products</gr
oupId>

<artifactId>fillo</artifactId>
   <version>1.15</version>
</dependency>
```

User, pass x4

Passed as array to function()

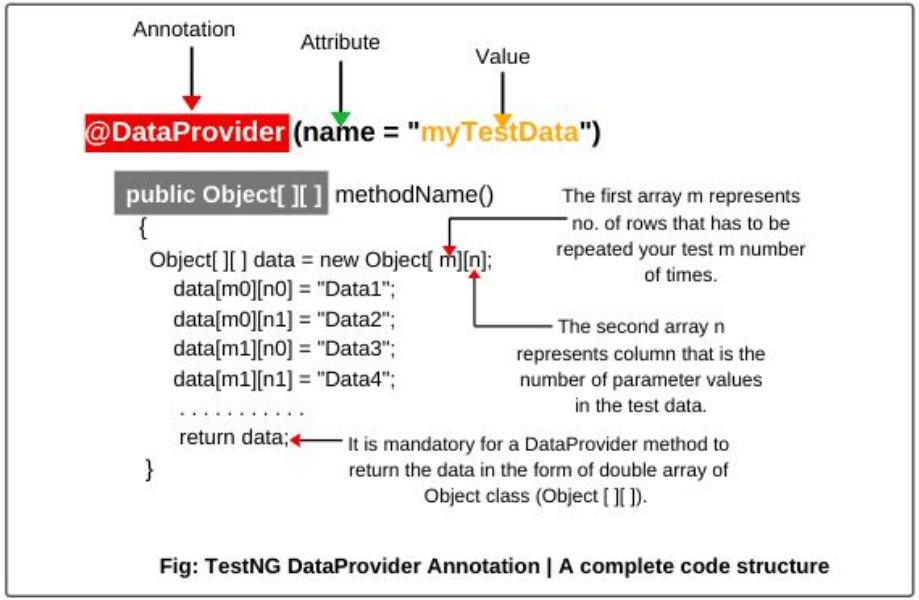


Fig: TestNG DataProvider Annotation | A complete code structure

Data Provider CSV File

```
Run All
public class BasicTest2 {

    @Test(dataProvider = "dp1")
    Run | Debug
    public void TestLogin(String[] s) throws Exception {
        System.out.println(s[0] +" >> "+s[1]);
    }

    @DataProvider()
    public String[][] dp1() {
        String[][] data= new String[][] {
            {"hyr","123"},
            {"pqr","456"},
            {"xyz","789"}
        };
        return data;
    }
```

ReportNG

Easy to use Reporting

```
<dependency>
        <groupId>org.testng</groupId>
        <artifactId>reportng</artifactId>
        <version>1.2.2</version>
        <scope>test</scope>
    </dependency>
```

TestNG – Parallel Test Execution

TestNG parallel execution of tests, classes and suites with examples. Learn how to run testng tests and suites in parallel or single test in multiple threads.

Reduces execution time

Allows multi-threaded tests

Test Level
Class Level
Method Level
D**ata Provider**
Thread-Count

Jenkins Pipeline Job

- Run POSTMAN collection as Pipeline Job.

Pipeline script

Script

```
pipeline {
    agent any
    stages {
        stage('Example 0') {
            steps {
                echo 'Hello World'
                echo "${branch}"
            }
        }
        stage('Example 1') {
            steps {
                echo 'Hello World'
                echo "${branch}"
            }
        }
        stage('Example 2') {
            steps {
```

☑ Use Groovy Sandbox

Pipeline Syntax

Save | Apply

Pipeline Hello Pipeline

Hi

Recent Changes

Stage View

	Example 0	Example 1	Example 2	Example 3
Average stage times: (Average full run time: ~879ms)	92ms	85ms	88ms	88ms
#8 Aug 29 23:22 No Changes	92ms	85ms	88ms	88ms

trend

Permalinks

Jenkins Pipeline Job

```
pipeline {
    agent any

    tools {
        // Install the Maven version configured as "M3" and add it to the path.
        maven "mvn"
    }

    stages {
        stage('Build and Run') {
            steps {
                // Get some code from a GitHub repository
                git branch: 'main', url: 'https://github.com/apitestingco/Java-TestNG-Selelnium.git'
                bat "mvn clean test -Dsuite=single.xml"
            }

            post {
                // If Maven was able to run the tests, even if some of the test
                // failed, record the test results and archive the jar file.
                success {
                    publishHTML([allowMissing: false, alwaysLinkToLastBuild: false, keepAll: false, reportDir: 'target/surefire-reports/', reportFiles: 'emailable-report.html', reportName: 'HTML
Report', reportTitles: '])
                }
            }
        }
    }
}
```

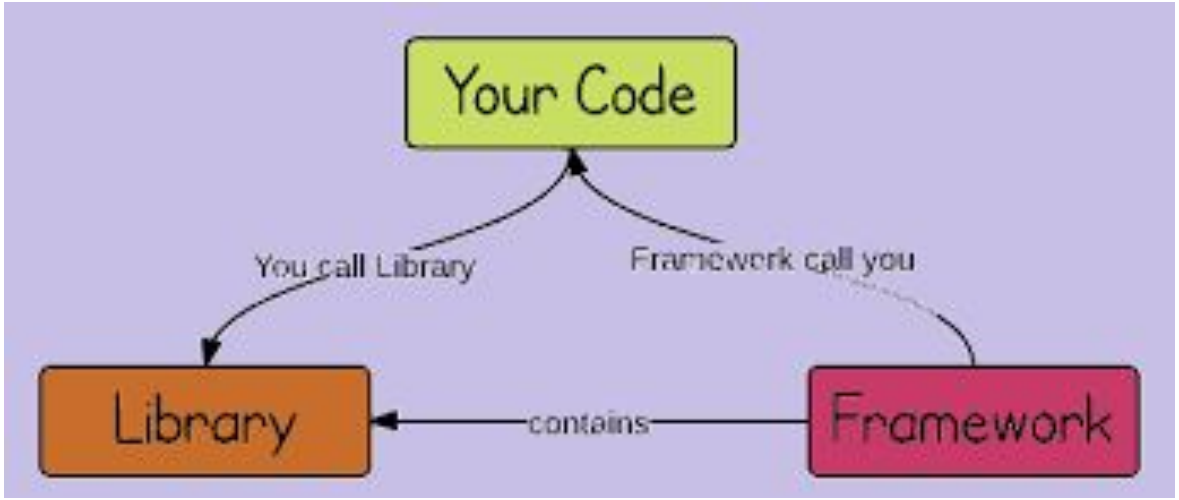
https://github.com/robsonbittencourt/jenkins-pipeline-example/blob/master/jenkinsfiles/pipeline/Jenkinsfile

https://github.com/kitconcept/jenkins-pipeline-examples

Why Framework

Why Framework

A framework defines the organization's way of doing things - a 'Single Standard'

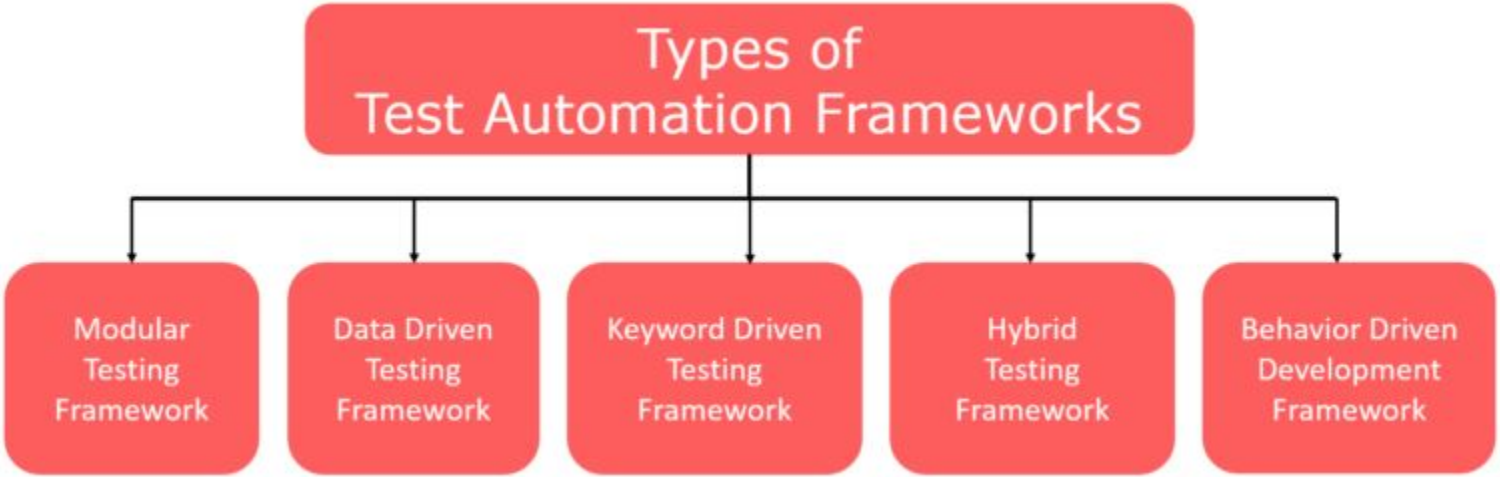


Framework vs Library

Basis	LIBRARY	FRAMEWORK
Meaning	Library is collection of reusable functions used by computer program	Framework is a piece of code that dictates the arcitecture of project
Inversion of control	With a library , you are in-charge, means you can choose where and when you want to insert or use the library.	In a framework, the framework is in-charge, not you, means a framework tells you where to put a specific part of your code
Function	They are important in program linking and binding process	In a framework , provided standard way to build and deploy applications
Flexiblity	Libraries are more flexible with greater degree of control	Frameworks are enforced structure and standards.
Example	React.js, Jquery is a javascript library	Angular js, Vue js is javascript framework

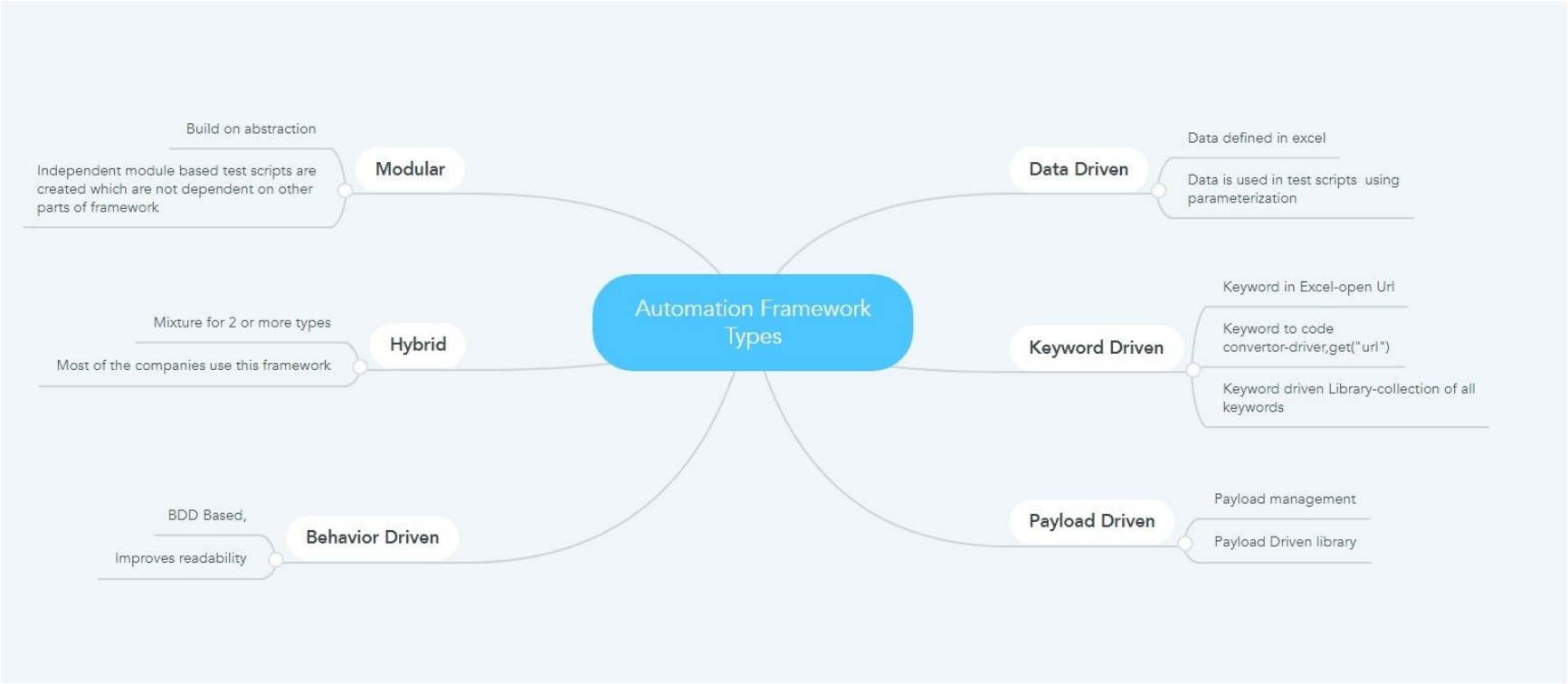
IS Selenium is Framework / Lib/ Tool?

https://stackoverflow.com/questions/67014328/in-automation-testing-is-selenium-a-framework-or-tool-or-library-as-few-refer


Types of
Test Automation Frameworks
Modular Testing Framework
Data Driven Testing Framework
Keyword Driven Testing Framework
Hybrid Testing Framework
Behavior Driven Development Framework

Types of Automation Frameworks

- Linear Scripting Framework
- Modular Testing Framework
- Data Driven Testing Framework
- Keyword Driven Testing Framework
- Hybrid Testing Framework
- Behavior Driven Development Framework


Automation Framework Types
Modular
Build on abstraction
Independent module based test scripts are created which are not dependent on other parts of framework
Hybrid
Mixture for 2 or more types
Most of the companies use this framework
Behavior Driven
BDD Based,
Improves readability
Data Driven
Data defined in excel
Data is used in test scripts using parameterization
Keyword Driven
Keyword in Excel-open Url
Keyword to code convertor-driver,get("url")
Keyword driven Library-collection of all keywords
Payload Driven
Payload management
Payload Driven library

Modular

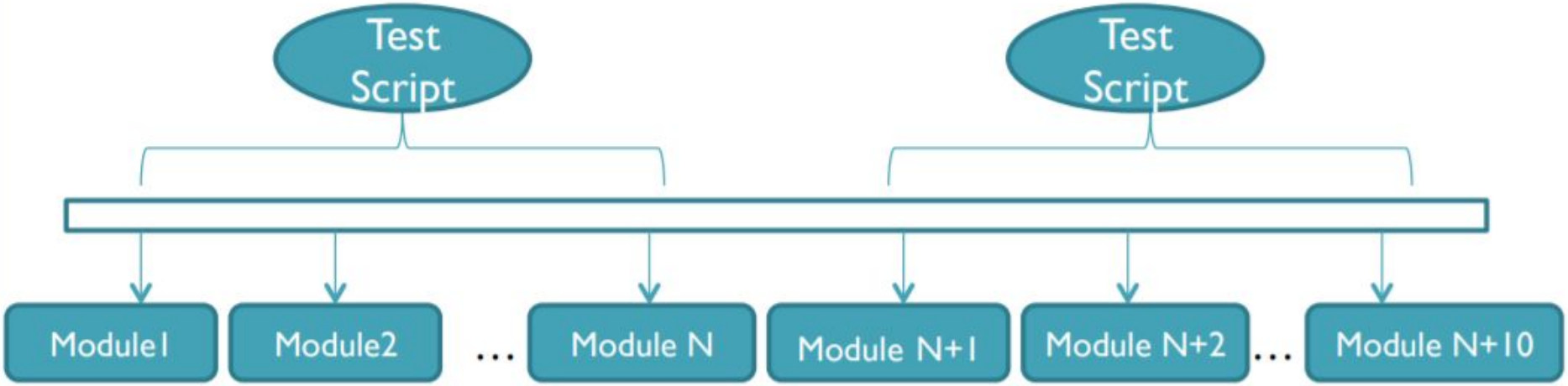
Modular frameworks divides the test scripts into small modules where modules are small scripts written to perform certain tasks.

Modular framework is like creation of small, independent scripts that represents modules, sections and functions of the application under test

individual test scripts can be combined to make larger test scripts by using a master script to achieve the required scenarios.

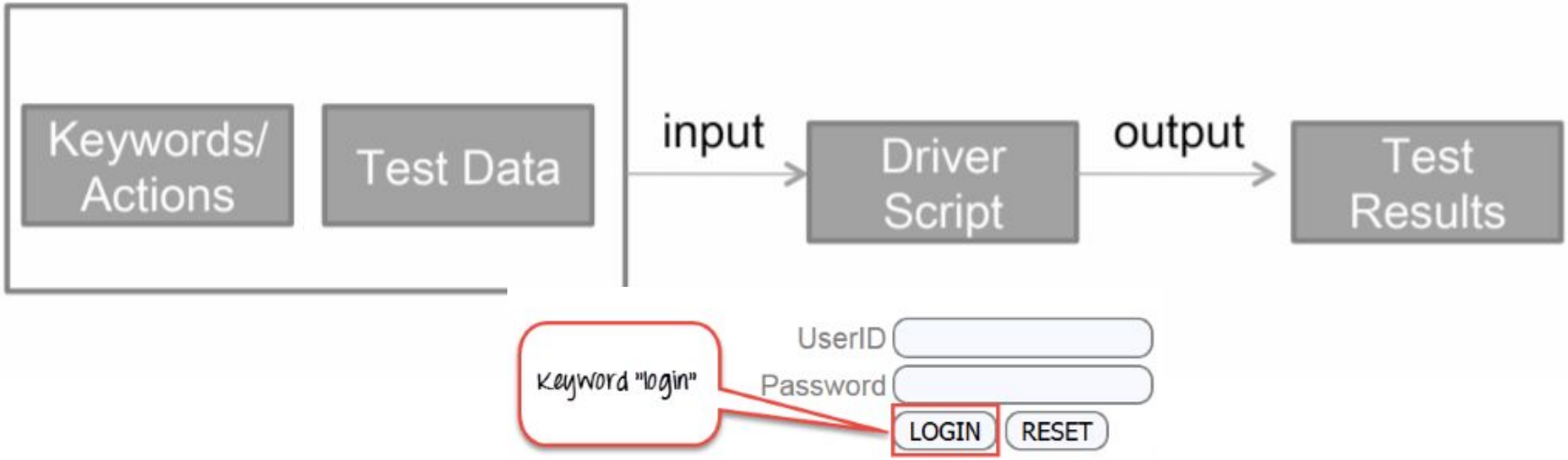
Master script is used to invoke the individual modules to run end to end test scenarios

Modular


Test Script
Test Script
Module1
Module2
...
Module N
Module N+1
Module N+2
...
Module N+10

Keyword Driven

Keyword Driven Framework is a functional automation testing framework that divides test cases into four different parts in order to separate coding from test cases and test steps for better automation

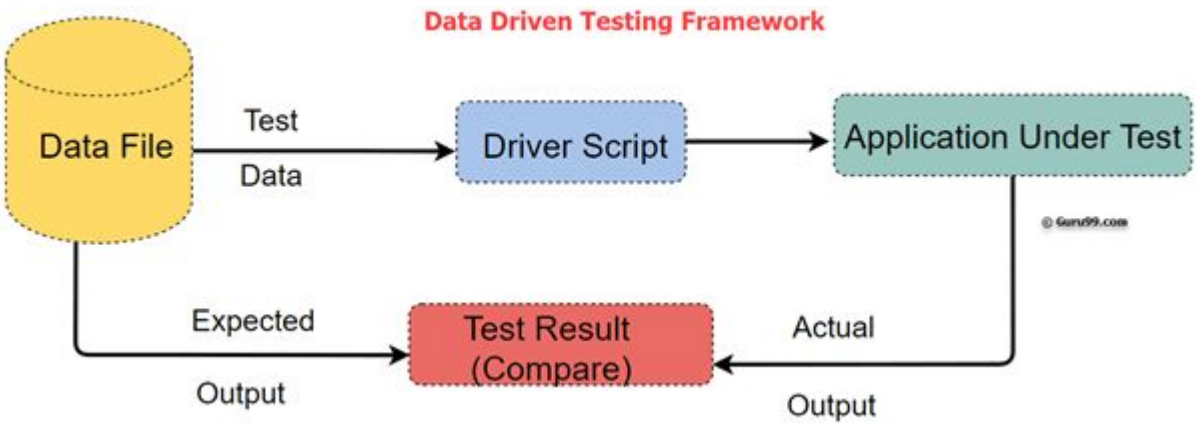


Keyword Driven

ID	Name	Step_ID	Description	Keyword
1	Login to Application	Step 1	Open browser	openBrowser
		Step 2	Navigate to URL	navigate
		Step 3	Enter Email	enterEmail
		Step 4	Enter Password	enterPassword
		Step 5	Click on Sign in button	clickSignIn
		Step 6	Click on Logout button	logout
		Step 7	Close browser	closeBrowser

Data Driven

Data driven testing allows testers to input a single test script that can execute tests for all test data from a table and expect the test output in the same table

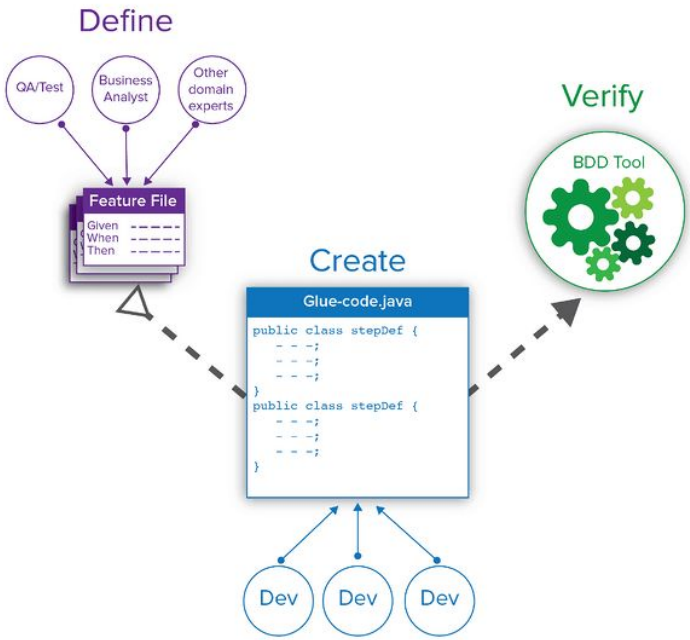


Behavior Driven

Runner File

Feature Files(Gherkin Language)

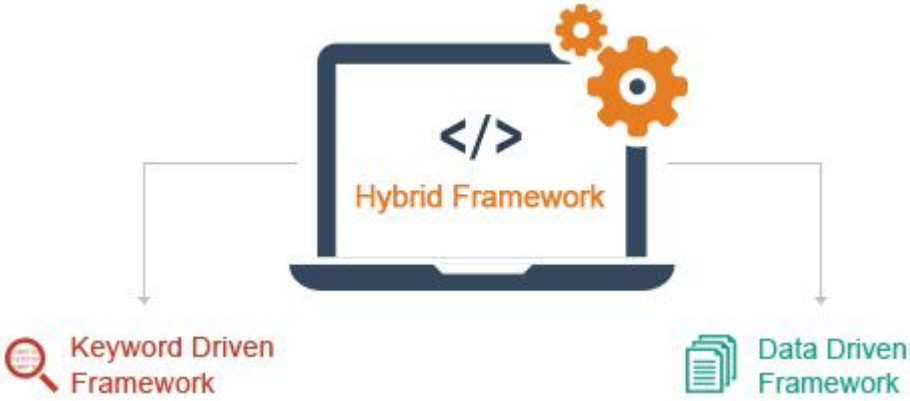
GivenWhenThen Methods/Test Scripts



Hybrid

Combination of multiple types

=>Data Driven+BDD+Keyword



Understand Requirements

- Is Framework Needed?
- Select Programming Language and tool
- Choose Your Framework Design.
- Create your framework
- Implement CI/CD

Framework Components

- Manage Dependencies/Projects(Maven/Gradle/PIP/NPM/Nuget)
- Manage Data(Excel/json files/prop files/xml files) - Apache POI/ Fillow
- Manage Payload and Endpoints(json strings/jsonmaps/pojos/serialize/deserialize/gson)
- Manage Tests (testng and allure for this)(precondition/postcondition/set/config/teardown/steps,description,priority,severity/execution)
- Reuse Components(Keywords, Abstraction, Inheritance, Generics, Configs. Specs. setups/helpers)
- Logger-Report Loggers(Testng. allure), text loggers(log4J)
- Reports-Test summary, percentage, steps, description, failure reason, logs-allure
- Utils -String manipulators, Json Manipulators, Data Manipulators, Readers/Writers/custom code/tools - Faker
- CI/CT-Version Controlling-github/git, Continuous Integration and Testing-jenkins/teamcity/travisci - Jenkins File + GIT + GIthub

Restful Booker Full CRUD

- Post Request using Rest Assured
- Patch Request Using Rest Assured
- Delete , Put
- One Integration Scenario
    - Create Booking, Update the , Delete it and Verify it via GET Request

Thanks, for attending Class

I hope you liked it.

Fin.