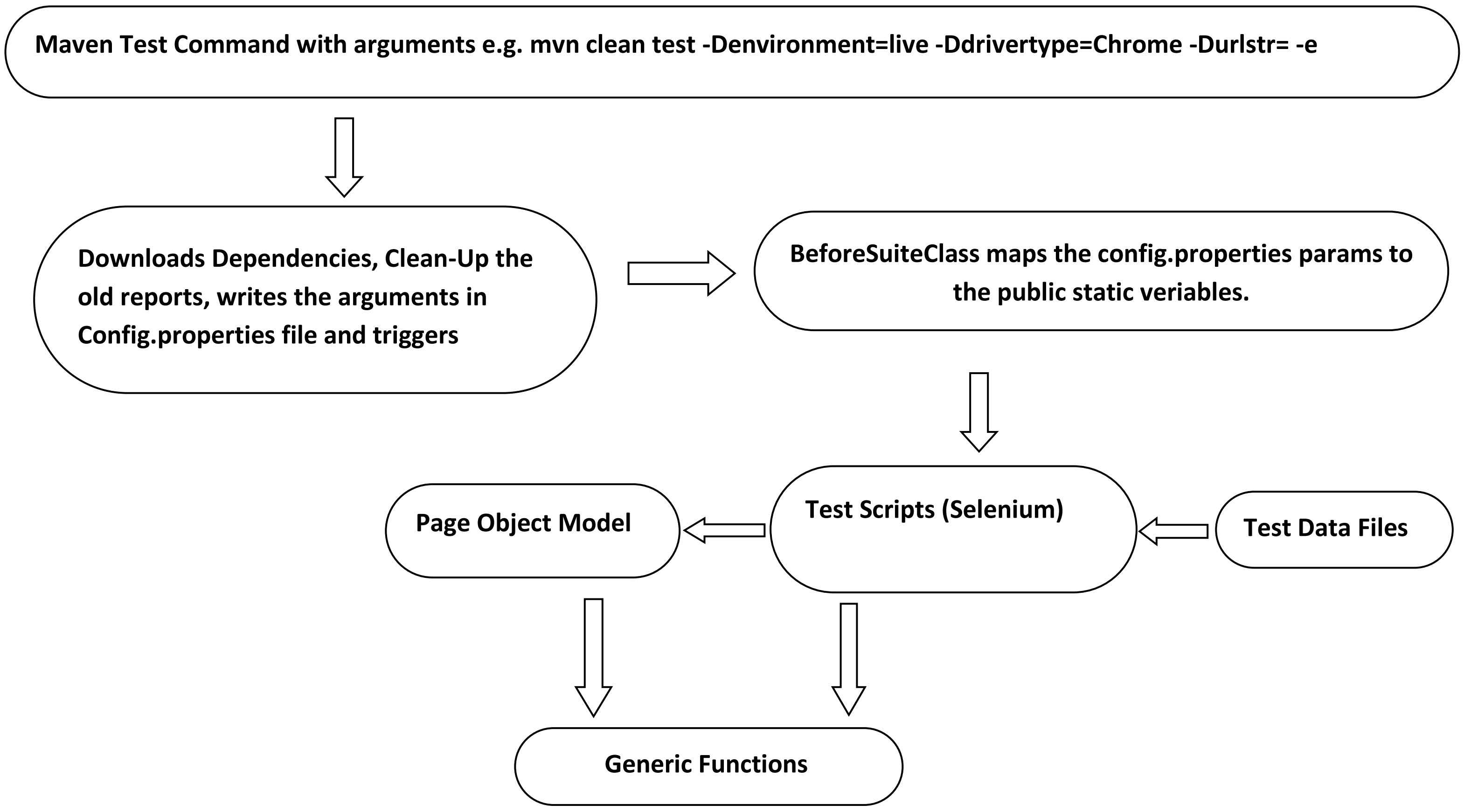
AUTOMATION TESTING TASK

Tools Used: - Selenium , TestNG, Maven plugins (Compiler, Ant run and Executor)

Platform: - Java JDK 1.8

Framework Architecture


Maven Test Command with arguments e.g. mvn clean test -Denvironment=live -Ddrivertype=Chrome -Durlstr= -e
Downloads Dependencies, Clean-Up the old reports, writes the arguments in Config.properties file and triggers
BeforeSuiteClass maps the config.properties params to the public static veriables.
Test Scripts (Selenium)
Page Object Model
Test Data Files
Generic Functions

Key Highlights:-

1. Thread Local Objects are used to provide multi-threading support. Code is clean and self-explanatory and needs minimal comments/explanation that too provided wherever required.

2. Test Cases are robust and stable, No Hard waits/sleep are used in any test case.

3. Test Suite is separated in layers (Test cases, test data, page objects, generic functions, configurations etc.)

Implementations:-

1) Logging ---- **Used log4j.**
2) Human Readable Reporting --- **Implemented Extent Reports with some tweaks.**
3) Taking Screenshot on failure --- **Done it using ITestListener along with Extent Report Implementation.**
4) WebDriver Factory --- **Implemented.**
5) Encapsulation of Test Data , Actions and Logic --- **Implemented Using Separation of Test Data in form of xls File /Json Payload and Using Page Object Pattern .**
6) Ability to run test on different Browsers --- **Done using Maven CLI arguments.**
7) Ability to run test on different Environments --- **Done using Maven CLI arguments.**
8) Reading Test Data from File ---- **Reading test data from xls File /Json Payload.**
9) Run Test in parallel mode --- **Done using modifying testng manually.**

Procedure to Run Selenium Web Automation Project:-

1) Take a git clone.
2) Go to the project directory and make sure that pom.xml file is lying there.
3) Command To Run :- mvn clean test -Denvironment=live -Ddrivertype=Chrome -Durlstr= -e

   Environment -> A flag used to determine whether to use the custom url (urlstr) in the testsuite or not.

   Drivertype -> Can be Firefox or Chrome , used for testing scripts on these browsers.

   Urlstr - > custom url support, to use it, Environment should not be live (it should be test)

Improvements that could be done in framework:-

1) Handling Test Case retry case: - To reduce flakiness in our test reports, we could have an option to retry the failed test cases in our framework that would require a retry Analyzer class implementing the IRetryAnalyzer Interface.
2) Logging the Test Case Steps to Extent Reports: - By utilizing the generic functions, we can record almost all the steps of the test cases into our extent reports. It would involve writing logs steps in each function of Generic Functions class and by reflection finding the variable name that get passed as argument.
3) Custom Assertions :- We can add our own assertions to the test suite , e.g. implementing assertions in a way that it assert the element , but does not stop the test cases (Verification of items instead of hard assertion)