

How to build your framework from scratch

- Testing Framework - guidelines and rules used for creating and designing test cases

1. Set up Environment ; install JDK, MAVEN, ECLIPSE IDE

2. Create Maven project

- ArtifactID - name of your project
- GroupID - identifies your project uniquely across all projects

3. Add Dependencies from Maven Repository

- Selenium Java
- Cucumber Sandwich
- JRE System library 1.8 in <properties>
- TestNG
- WebDriverManager_BoniGarcia
- Cucumber Java
- Cucumber TestNG
- Apache POI.XML
- APACHE POI
- JDBC
- Restassured
- Gson
- Log4j

4. Create framework Structure (packages)

- Pages
 - i. Webelements and methods
- POJO / Beans
 - i. Custom classes
- Runner
 - i. Cuckesrunner - generates codes and stores html report to target
 - ii. Smoke
 - iii. Regression
- StepDefinitions
 - i. Actual codes and hook class
- Tests
 - i. Data driven tests
- JDBC
- API
- Utilities
 - i. ConfigurationReader
 - ii. Driver class (Singleton)
 - iii. browserUtils
 - iv. DBUtils
 - v. ApiUtils
- Configuration.properties
- Testng_runner.xml
 - i. One of the runner class Call the packages with com.app.utilities

5. Create feature folder with feature file (.feature) in resources

- File executable specification written in a gherkin language

6. After you finish writing your scenario, run your cukesRunner with dryRun=false, this will give you implementation code that you will store in StepDefinition class

7. How to run your code?

- Use runner class - run codes and generate cucumber report and html reports

8. Send framework to GitHub or SVN

- Create new repo
- Copy git url and go to eclipse
- Configure git repo and add the url
- Right click project, click team, click commit and you'll be in git staging
- NOW YOUR CODE IS IN GITHUB

9. Next is JENKINS INTEGRATION

- Jenkins
 - i. Open source automation server
 - ii. Helps to automate the non-human part of the software development process
 - iii. Allows continuous integration
 - iv. Development practice that requires dev to integrate code into a shared repo at reg intervals
 - v. Port is 8081 localhost
- To run your project in jenkins
 - i. Login to jenkins account
 - ii. Create project - freestyle
 - iii. Install plugins -cucumber report and git
 - iv. Under source code management choose git and past git url
 - v. Build trigger - choose to build periodically
 - vi. Invoke top-level maven
 1. Maven version ; MAVEN_HOME
 2. Goals ; clean verify -Drunner=smoke_runnerxml
 - vii. Under post-build actions
 1. Choose cucumber reports
 2. Choose editable email notify
 - viii. Editable email notification
 1. Attach build log ; choose build log
 2. Click advanced settings
 - ix. Failure-Any
 1. Click advanced
 2. Recipient list - email address who will receive the report. Add comma if multiple
 3. Click add trigger - like failure always
 4. Attach build log; select attach build log
 5. Save
 - x. Final Step
 1. Click build now and test will run and gives your cucumber report

Talk about cucumber report after the build

1. What is Page Object Model

- a. Reduces code redundancy and organizes code
- b. Helps identify elements and store it as a page object variable
- c. You can link it to where it was stored
- d. Added PageFactory design pattern

2. Selenium WebDriver as my automation tool

- e. Manual test it first by:
 - i. Front end
 - Functional testing
 - ii. Back-end
 - Database- SQL Developer IDE
 - API - Postman
- f. Integrate Selenium with:
 - i. Maven
 - Test package
 - Utility package
 - a. UI
 - b. DB
 - c. API
 - Configuration file
 - a. Properties
 - Driver class
 - a. Singleton design pattern
 - a. Have a private constructor
 - ii. Cucumber BDD
 - Facilitate the collaboration during the BDD process
 - Enables explaining the story and the acceptance criteria in easy language.
 - iii. Git - source control
 - iv. Jenkins
 - v. Java
 - Collections Framework
 - Apache POI
 - JDBC
 - Rest-Assured

3. Behavior Driven Development

- Developing together with the customer to ensure it meets the right standards for the customer

4. Data Driven Development

- Executing same test case against different sets of data
- Test flow should not change based on data

5. Cucumber reporting

- Target folder
- Jenkins