Cucumber Framework Advanced

- Build Maven Project with all the framework dependencies
- Develop end to end selenium web scenarios in the feature file
- Implement TestNG/ Junit Runner to run the framework
- Understand the single responsibility principle to distribute the implementation into multiple step definitions
- Importance of Dependency injection to share the data between the Step file
- Understand the Page object pattern to drive the locators from page files
- Implement Factory Design Pattern by writing PageObjectManager class to create objects of all PO classes
- How driver can be configured and distributed across the files in the framework
- How to run Cucumber scenarios in parallel mode using Cucumber TestNG runner
- Build Test utilities for reusable Selenium methods
- Implement Cucumber Hooks to capture screenshots on test failures
- Implement Cucumber tags to run selected tests in framework
- Parameterizing the test data to run the scenarios with multiple data sets
- Creating HTML and Extent reports to run Cucumber Selenium Tests
- Running tests in parallel mode and generate Extent reports with screenshots attached for failed scenarios
- Run the cucumber tests using maven & command line options
- Integrate the framework to CI/ CD Jenkins and schedule the jobs on regular time intervals
- Create Parameterized Jenkins job to dynamically send the global properties at run time of job execution

https://rahulshettyacademy.com/seleniumPractise/#/

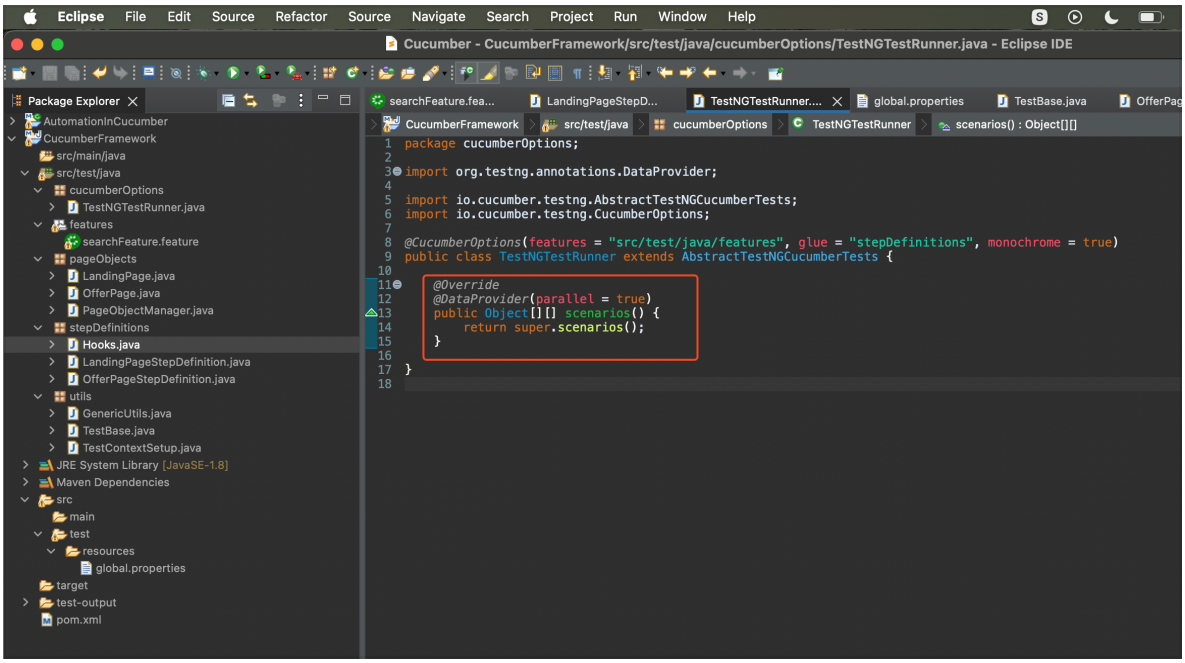
What is Cucumber dependency injection? Cucumber PicoContainer use

To run tests parallel in cucumber selenium:

- To obtain parallel execution in cucumber test runner,
  -we need to call **scenarios()** from its parent class(**AbstractTestNGCucumberTests**) from cucumber testng runner.
  -set **parallel** attribute to 'true' for @DataProvider annotation

```
@Override
```

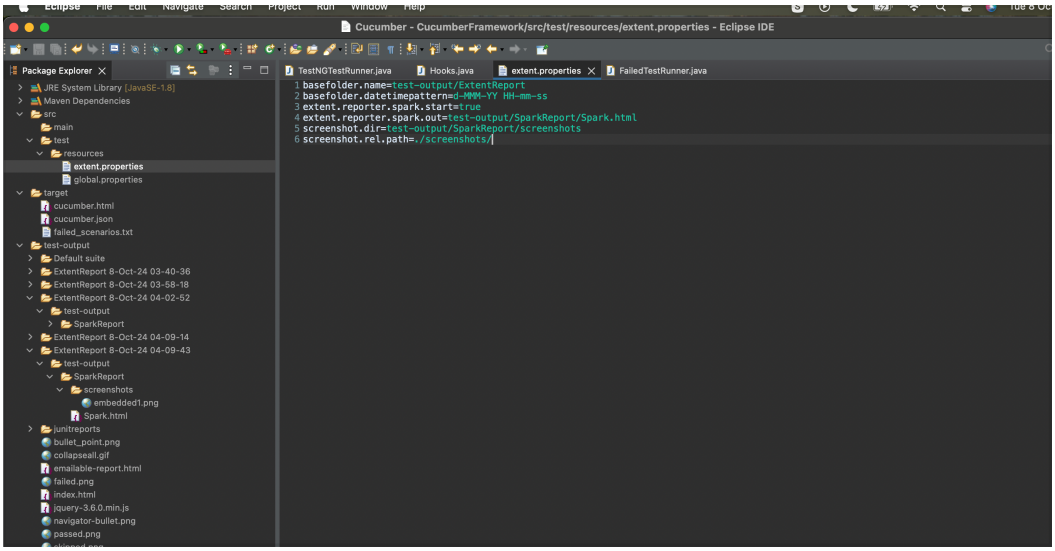
```
    @DataProvider(parallel = true)
    public Object[][] scenarios() {
        return super.scenarios();
    }
```



Execution of scenarios along with cucumber & extent reports attaching failed screenshots & configured .properties:

```
package cucumberOptions;

import org.testng.annotations.DataProvider;

import io.cucumber.testng.AbstractTestNGCucumberTests;
import io.cucumber.testng.CucumberOptions;

@CucumberOptions(features = "src/test/java/features", glue =
"stepDefinitions", monochrome = true, tags = "@SearchProduct
or @PlaceOrder", plugin = {
        "html:target/cucumber.html", "json:target/
cucumber.json",
        "com.aventstack.extentreports.cucumber.adapter.Extent
CucumberAdapter:",
        "rerun:target/failed_scenarios.txt"})
public class TestNGTestRunner extends
AbstractTestNGCucumberTests {

    @Override
    @DataProvider(parallel = true)
    public Object[][] scenarios() {
        return super.scenarios();
```

```
    }
}
```



Execution of only failed scenarios from test runner

```
package cucumberOptions;

import org.testng.annotations.DataProvider;

import io.cucumber.testng.AbstractTestNGCucumberTests;
import io.cucumber.testng.CucumberOptions;

@CucumberOptions(features = "@target/failed_scenarios.txt",
glue = "stepDefinitions", monochrome = true, plugin = {
        "html:target/cucumber.html", "json:target/
cucumber.json",
        "com.aventstack.extentreports.cucumber.adapter.Extent
CucumberAdapter:" })
public class FailedTestRunner extends
AbstractTestNGCucumberTests {

    @Override
    @DataProvider(parallel = true)
    public Object[][] scenarios() {
        return super.scenarios();
    }
}
```

Run our maven cucumber framework from command line

- If we want to run all of our scenarios present in the cucumber project from command line,

```
>mvn test
```

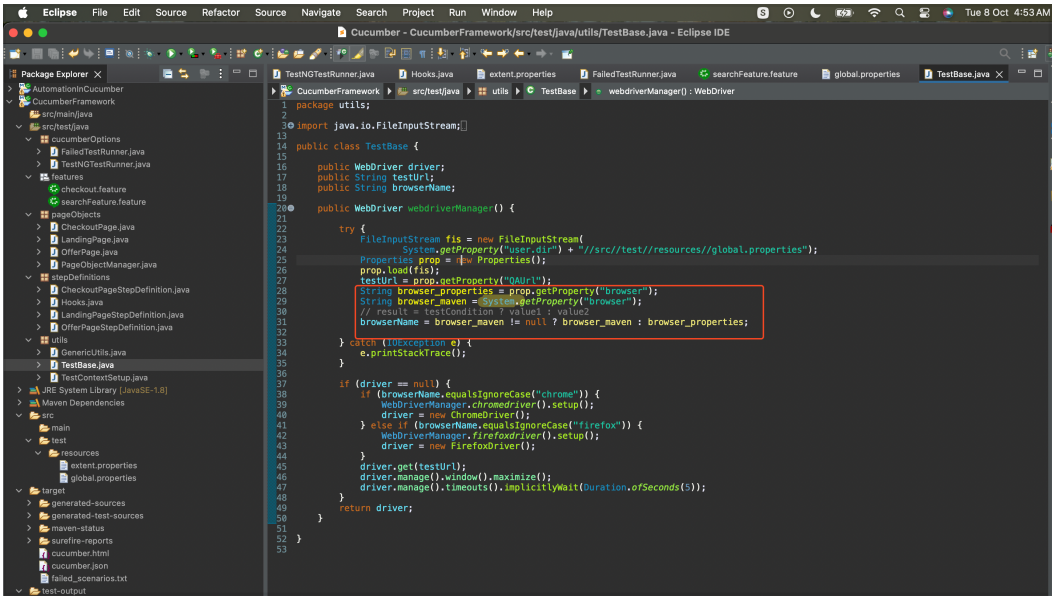
- If we want to run subset of scenarios of a project from command line,

```
>mvn test -Dcucumber.filter.tags="@SearchProduct"
>mvn test -Dcucumber.filter.tags="@PlaceOrder"
-Dbrowser=edge
```

- If we want to run specific feature file from project from command line,

```
>mvn test -Dcucumber.features="/Users/krishnabros/Desktop/
Projects/Cucumber/CucumberFrameworkTestNG/src/test/java/
features/checkout.feature"
>mvn test -Dcucumber.features="/Users/krishnabros/Desktop/
Projects/Cucumber/CucumberFrameworkTestNG/src/test/java/
features/searchFeature.feature"
```

- If we want to pass global properties through command line, we need to first configure or address our property variable preference in the code as highlighted in below screenshot and then run the command in command line as below. (Preference of variable as below
  First : command line
  Second : code)
  -**Browser properties** from **command line** is interpreted/ fetched in cucumber selenium java code using **System.getProperty("browser")**



Run scripts in Chrome, Edge browser -

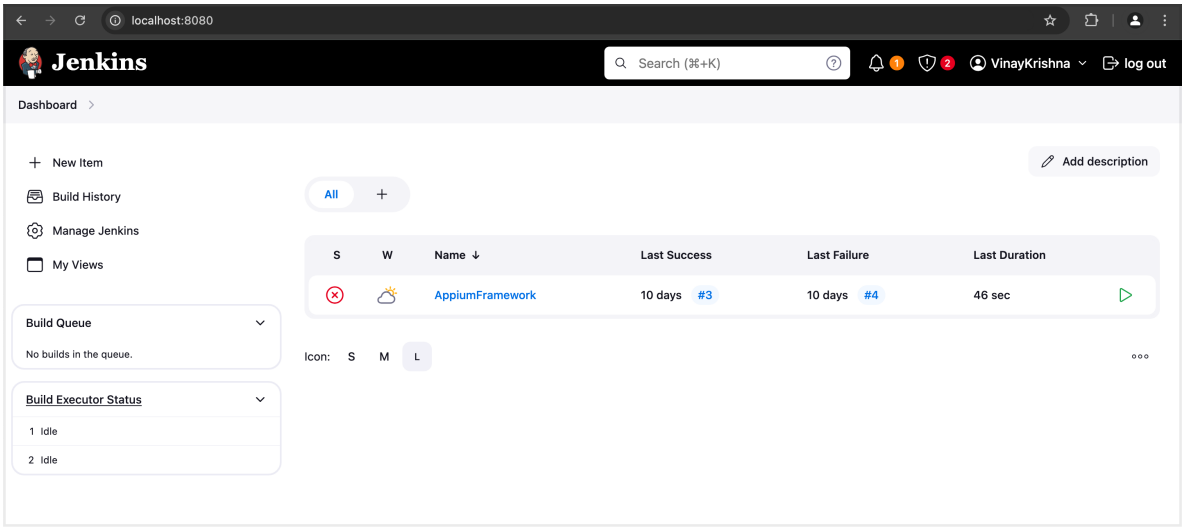
```
>mvn test -Dcucumber.features="/Users/krishnabros/Desktop/
Projects/Cucumber/CucumberFrameworkTestNG/src/test/java/
features/searchFeature.feature" -Dbrowser=chrome
>mvn test -Dcucumber.features="/Users/krishnabros/Desktop/
Projects/Cucumber/CucumberFrameworkTestNG/src/test/java/
features/searchFeature.feature" -Dbrowser=edge
```

```
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:3.0.0-M5:test (default-test) @ CucumberFramework ---
[INFO]
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running TestSuite
SLF4J(W): No SLF4J providers were found.
SLF4J(W): Defaulting to no-operation (NOP) logger implementation
SLF4J(W): See https://www.slf4j.org/codes.html#noProviders for further details.
Tomato is extracted from home page
Strawberry is extracted from home page
Beetroot is extracted from home page
Strawberry is extracted from landing page
Tomato is extracted from landing page
No data is extracted from landing page
[ERROR] Tests run: 3, Failures: 1, Errors: 0, Skipped: 0, Time elapsed: 9.788 s <<< FAILURE! - in TestSuite
[ERROR] cucumberOptions.TestNGTestRunner.runScenario["Search experience for product search in both home and offers page", "Search for Products in home and offers page"](3)  Time
elapsed: 9.124 s  <<< FAILURE!
java.lang.AssertionError: expected [Beetroot] but found [No data]

[INFO]
[INFO] Results:
[INFO]
[ERROR] Failures:
[ERROR] cucumberOptions.TestNGTestRunner.runScenario["Search experience for product search in both home and offers page", "Search for Products in home and offers page"](3)
[INFO]   Run 1: PASS
[INFO]   Run 2: PASS
[ERROR]   Run 3: TestNGTestRunner.runScenario expected [Beetroot] but found [No data]
[INFO]
[INFO]
[ERROR] Tests run: 1, Failures: 1, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD FAILURE
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  11.117 s
[INFO] Finished at: 2024-10-08T04:55:56+05:30
[INFO] ------------------------------------------------------------------------
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-surefire-plugin:3.0.0-M5:test (default-test) on project CucumberFramework: There are test failures.
[ERROR]
[ERROR] Please refer to /Users/krishnabros/Desktop/Projects/Cucumber/CucumberFramework/target/surefire-reports for the individual test results.
[ERROR] Please refer to dump files (if any exist) [date].dump, [date]-jvmRun[N].dump and [date].dumpstream.
[ERROR] -> [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/MojoFailureException
krishnabros@Krishnas-MacBook-Air CucumberFramework %
```

Run Jenkins.war file in local

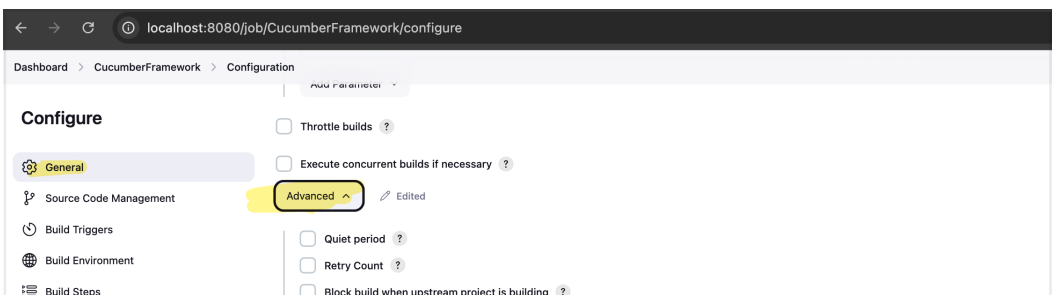
`>java –jar jenkins.war –httpPort=9090`

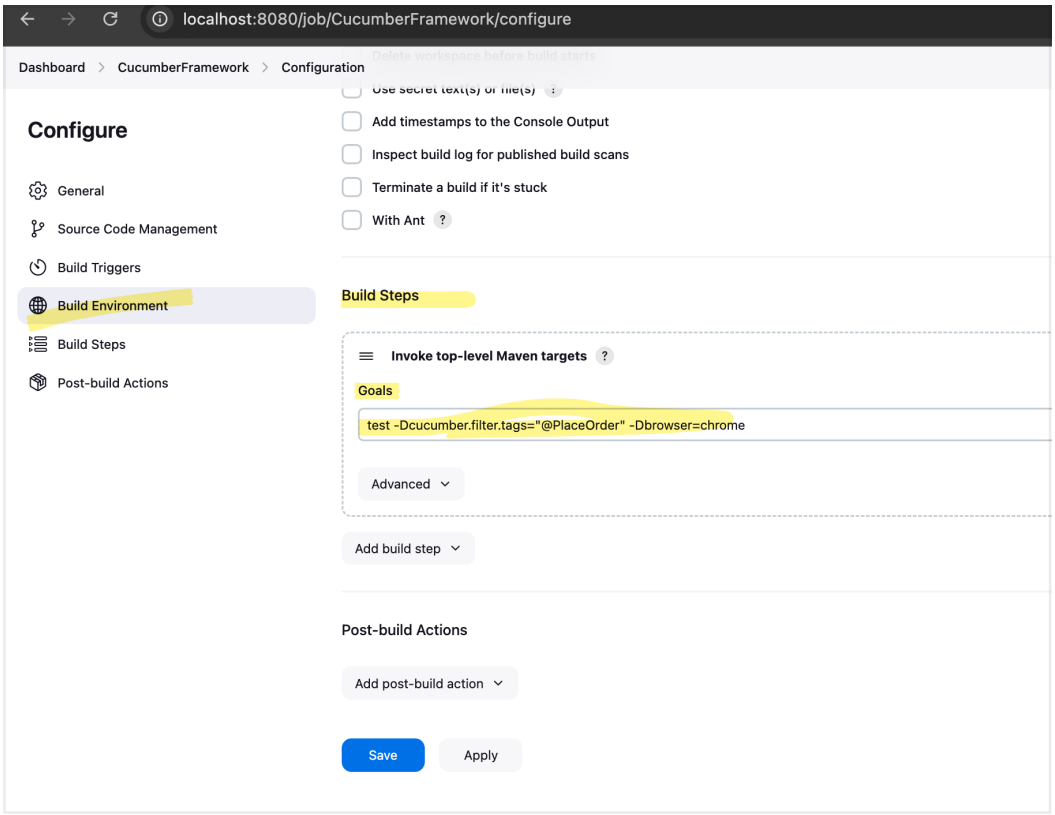
By default my jenkins runs in port 8080



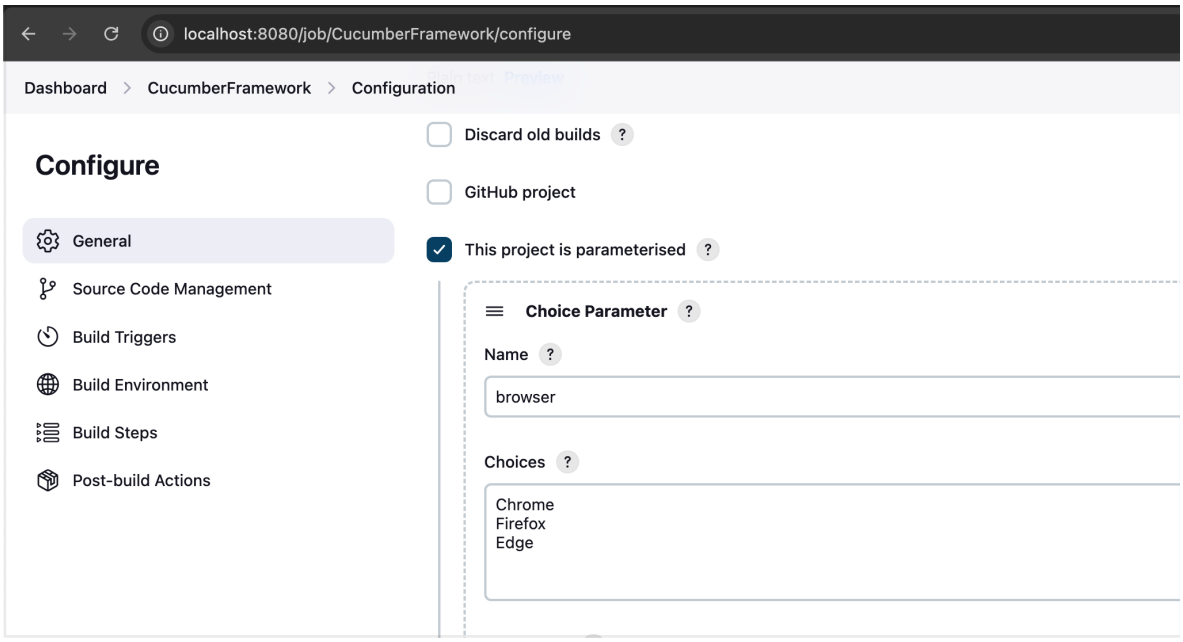
Create Jenkin Job to run Selenium Cucumber Framework Scenarios :

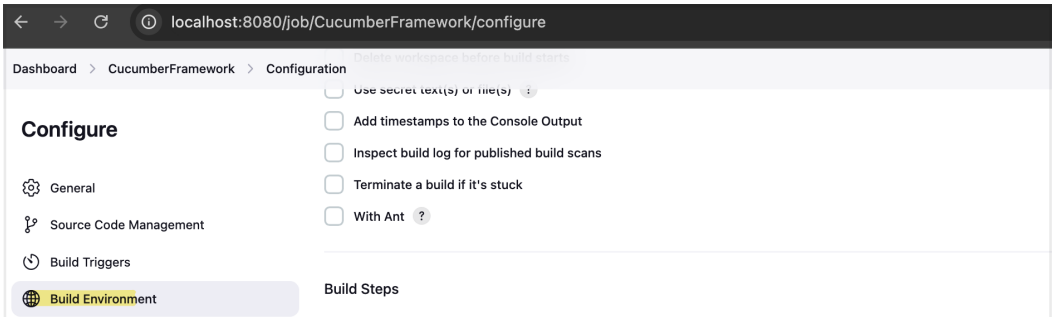
1. We can create a jenkins Job for local project with below configured maven goals
   test -Dcucumber.filter.tags="@PlaceOrder" -Dbrowser=chrome

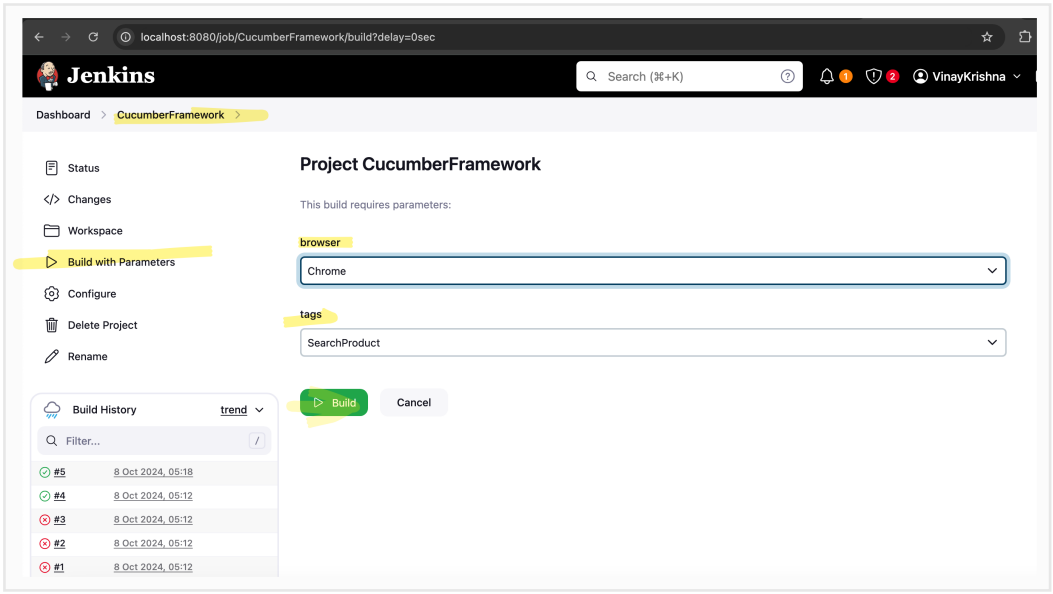


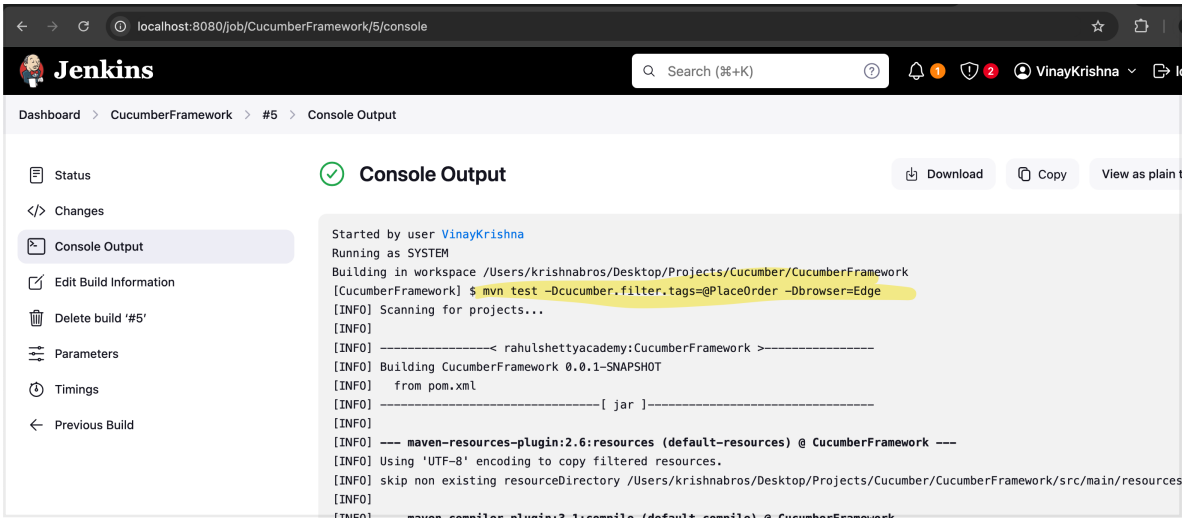


2. We can parameterise our jenkins job by adding choice parameters and below configured maven goals
test -Dcucumber.filter.tags="@"$tags"" -Dbrowser="$browser"

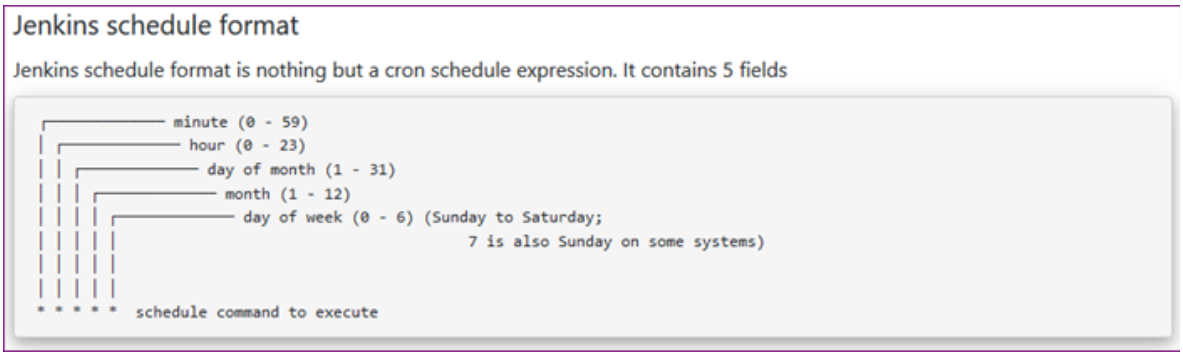
3. 





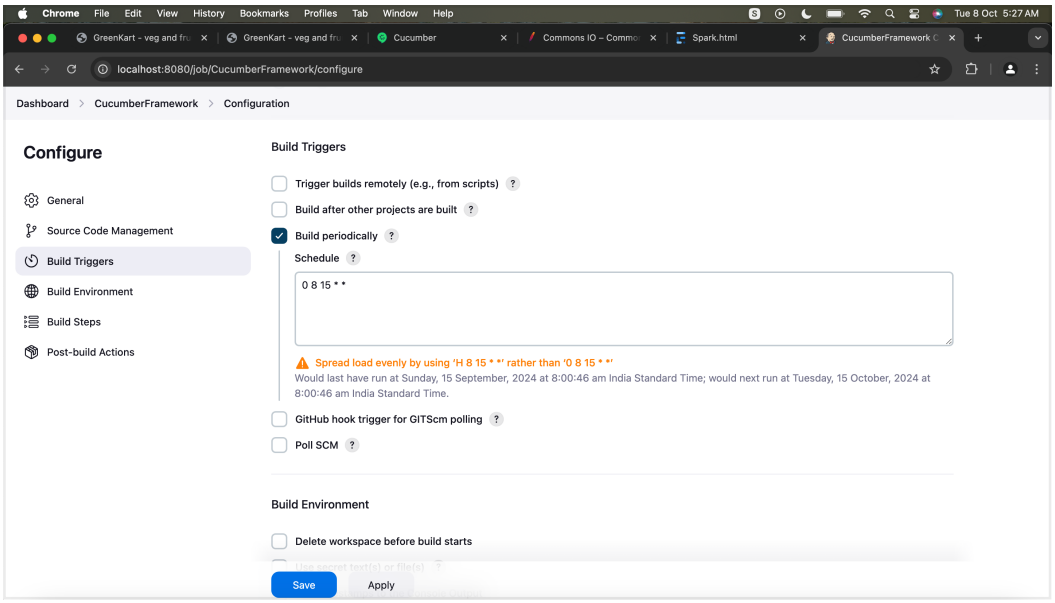


2. We can schedule our jenkin job
   Jenkins used a **cron expression** and the different fields are:

3. 

1. MINUTES Minutes in one hour (0–59)
2. HOURS Hours in one day (0–23)
3. DAYMONTH Day in a month (1–31)
4. MONTH Month in a year (1–12)
5. DAYWEEK Day of the week (0–7) where 0 and 7 are Sunday

- Represents every time and no restriction

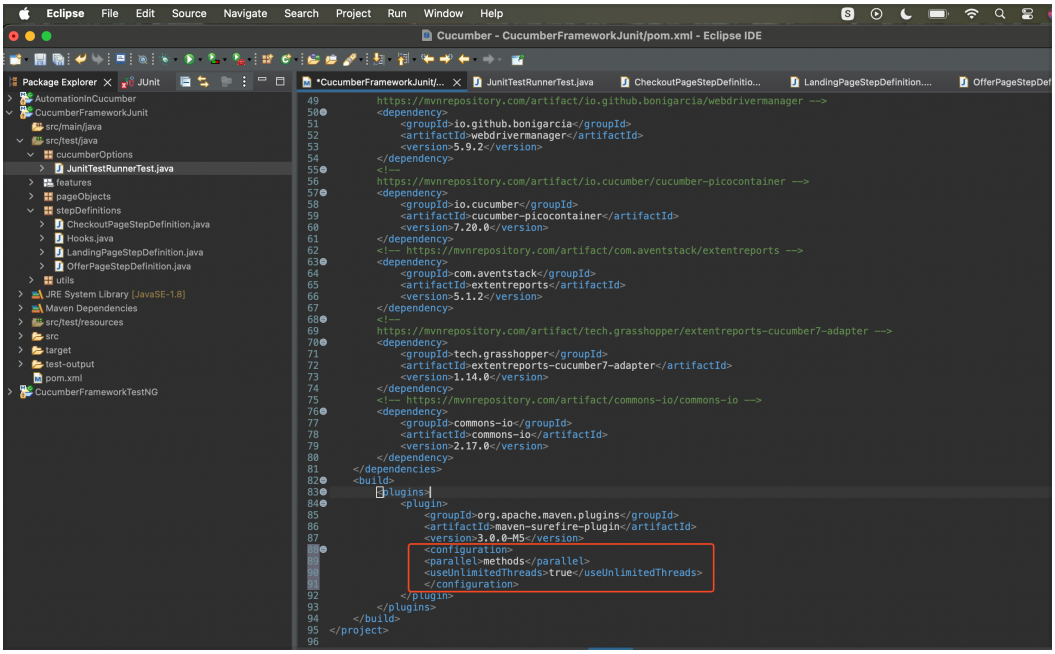


Running cucumber project using Junit:

- For achieving parallel execution of scenarios using Junit:

Note:

- In JUnit, feature files are run in parallel rathan than scenarios, which means all the scenarios in a feature file will be executed by the same Thread. You can use Maven surefire to execute the runners
- In TestNG, feature files & its scenarios are run in parallel by different Threads.



- Execute junit tests from command line

```
>mvn test
```