

BDD framework design

1 For Eclipse IDE

Install open JDK and set the environment variables in the system

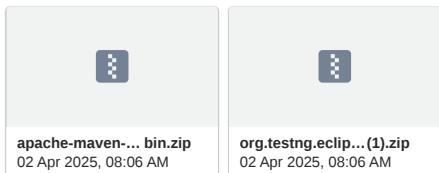
To check java version in command prompt run the command: `java -version`

Download maven and set the environment variables in the system

To check maven version in command prompt run the command : `- mvn -version`

Install eclipse IDE

install testNG plugin in eclipse from eclipse market place or install from zip file offline



Create a maven project in eclipse IDE , provide relevant group id and artifact id

click on New - Project - maven -

check the checkbox "create a simple project (skip archetype selection)

click on next

provide relevant group ID, artifact ID and packaging should be jar

click on Finish

A maven project will be displayed in the project workspace directory

open pom.xml file and add https to remove errors

Note - for IntelliJ convert https to http

File- Maven - update project

Now Add testNG library to the maven project

Right click > build path > configure build path >

select libraries tab

Click on Add library

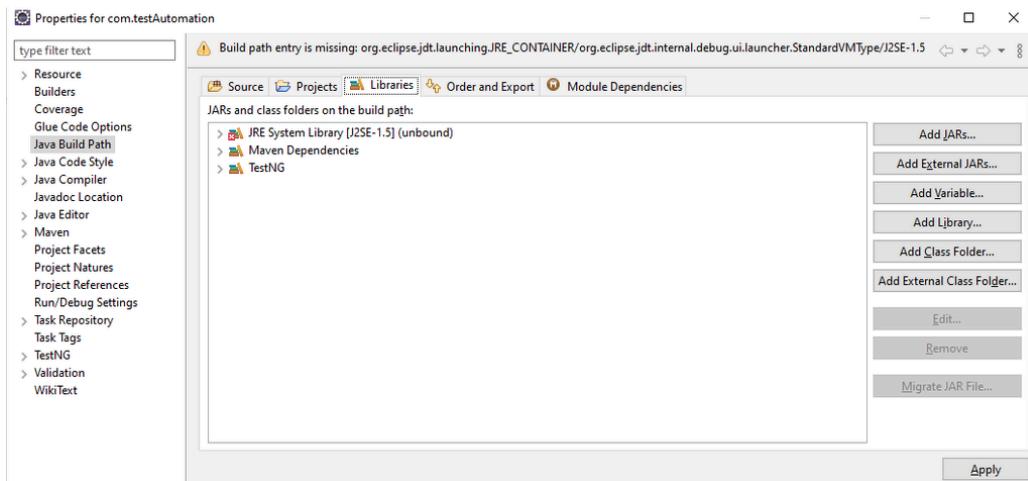
Select testNG in the popup window

click on next and finish

click on Apply And Close

user should be able to see the TestNG with lib icon in the project structure

Update the JRE system lib to latest one



right click > build path > configure build path >

click and delete the old jre lib

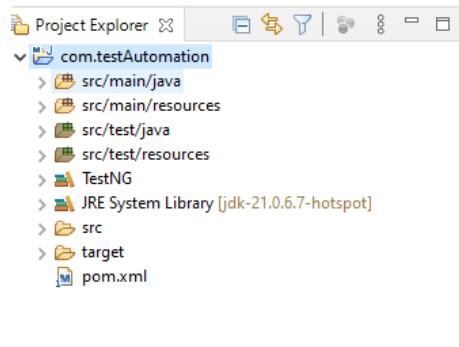
Again click on Add Library

JRE System Library

Click on Next

Select the radio button alternate JRE (it will fetch the installed jdk)

click on finish > Apply And Close



updated system library will be shown like this.

Convert the project to cucumber project

Right click > configure > Convert To cucumber project

project will have one green icon

Now its time to update the pom.xml file with all the required dependencies

Format 1

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
5 <modelVersion>4.0.0</modelVersion>
6
7 <groupId>com.anhtester</groupId>
8 <artifactId>anhtester-cucumber-testng</artifactId>
9 <version>1.8.5</version>
10 <name>Test Automation Framework Cucumber TestNG</name>
11 <url>https://github.com/anhtester/AutomationFrameworkCucumberTestNG</url>
12
13 <properties>
14     <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
15     <project.build.resourceEncoding>UTF-8</project.build.resourceEncoding>
16     <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
17     <java-compiler.version>17</java-compiler.version>
18     <maven-surefire-plugin.version>3.5.2</maven-surefire-plugin.version>
19
20     <cucumber.java.version>7.21.1</cucumber.java.version>
21     <gherkin.version>32.0.0</gherkin.version>
22     <cucumber-reporting.version>5.8.4</cucumber-reporting.version>
23     <selenium.version>4.29.0</selenium.version>
24     <testng.version>7.11.0</testng.version>
25     <aspectjweaver.version>1.9.22.1</aspectjweaver.version>
26     <log4j.version>2.24.3</log4j.version>
27     <slf4j.version>2.0.16</slf4j.version>
28     <allure-testing.version>2.29.1</allure-testing.version>
29     <allure-maven.version>2.15.2</allure-maven.version>
30     <allure-environment-writer.version>1.0.0</allure-environment-writer.version>
31     <allure.cmd.download.url>
32         https://repo.maven.apache.org/maven2/io/qameta/allure/allure-commandline
33     </allure.cmd.download.url>
34     <extentreports.version>5.1.2</extentreports.version>
35     <apache-poi.version>5.4.0</apache-poi.version>
36     <apache-poi-xml.version>5.2.5</apache-poi-xml.version>
37     <commons-io.version>2.18.0</commons-io.version>
38     <owner.version>1.0.12</owner.version>
39     <assertj.version>3.24.2</assertj.version>
40     <monte-screen-recorder.version>0.7.7.0</monte-screen-recorder.version>
41     <mysql-connector-java.version>8.0.33</mysql-connector-java.version>
42     <lombok.version>1.18.36</lombok.version>
43     <data.supplier.version>1.9.7</data.supplier.version>
44     <javax.mail.version>1.6.2</javax.mail.version>
45     <zip.version>1.17</zip.version>
46     <rest-assured.version>5.5.1</rest-assured.version>
47     <gson.version>2.12.1</gson.version>
48     <jackson-databind.version>2.18.2</jackson-databind.version>
49     <datafaker.version>2.4.2</datafaker.version>
50     <telegram.bot.version>8.3.0</telegram.bot.version>
51     <commons-lang3.version>3.17.0</commons-lang3.version>
52     <extentreports-cucumber7-adapter.version>1.14.0</extentreports-cucumber7-adapter.version>
53     <mysql-connector-j.version>9.2.0</mysql-connector-j.version>
54
55     <!-- Suite XML path -->
56     <suite.feature>src/test/resources/suites/SuiteFeatureByTag.xml</suite.feature>
57     <suite.feature.login.cms>src/test/resources/suites/SuiteFeatureLoginCMS.xml</suite.feature.login.cms>
58 </properties>
```

```
59
60 <dependencies>
61
62     <!-- https://mvnrepository.com/artifact/io.rest-assured/rest-assured -->
63     <dependency>
64         <groupId>io.rest-assured</groupId>
65         <artifactId>rest-assured</artifactId>
66         <version>${rest-assured.version}</version>
67     </dependency>
68     <!-- https://mvnrepository.com/artifact/io.rest-assured/json-schema-validator -->
69     <dependency>
70         <groupId>io.rest-assured</groupId>
71         <artifactId>json-schema-validator</artifactId>
72         <version>${rest-assured.version}</version>
73     </dependency>
74     <!-- https://mvnrepository.com/artifact/com.google.code.gson/gson -->
75     <dependency>
76         <groupId>com.google.code.gson</groupId>
77         <artifactId>gson</artifactId>
78         <version>${gson.version}</version>
79     </dependency>
80
81     <!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind -->
82     <dependency>
83         <groupId>com.fasterxml.jackson.core</groupId>
84         <artifactId>jackson-databind</artifactId>
85         <version>${jackson-databind.version}</version>
86     </dependency>
87
88     <!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-core -->
89     <dependency>
90         <groupId>com.fasterxml.jackson.core</groupId>
91         <artifactId>jackson-core</artifactId>
92         <version>${jackson-databind.version}</version>
93     </dependency>
94
95     <!-- https://mvnrepository.com/artifact/org.apache.commons/commons-lang3 -->
96     <dependency>
97         <groupId>org.apache.commons</groupId>
98         <artifactId>commons-lang3</artifactId>
99         <version>${commons-lang3.version}</version>
100    </dependency>
101
102    <!-- https://mvnrepository.com/artifact/net.datafaker/datafaker -->
103    <dependency>
104        <groupId>net.datafaker</groupId>
105        <artifactId>datafaker</artifactId>
106        <version>${datafaker.version}</version>
107    </dependency>
108
109    <!-- https://mvnrepository.com/artifact/com.github.pengrad/java-telegram-bot-api -->
110    <dependency>
111        <groupId>com.github.pengrad</groupId>
112        <artifactId>java-telegram-bot-api</artifactId>
113        <version>${telegram.bot.version}</version>
114    </dependency>
115
116    <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-java -->
```

```
117     <dependency>
118         <groupId>io.cucumber</groupId>
119         <artifactId>cucumber-java</artifactId>
120         <version>${cucumber.java.version}</version>
121     </dependency>
122     <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-testng -->
123     <dependency>
124         <groupId>io.cucumber</groupId>
125         <artifactId>cucumber-testng</artifactId>
126         <version>${cucumber.java.version}</version>
127     </dependency>
128
129     <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-picocontainer -->
130     <dependency>
131         <groupId>io.cucumber</groupId>
132         <artifactId>cucumber-picocontainer</artifactId>
133         <version>${cucumber.java.version}</version>
134     </dependency>
135
136     <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-gherkin -->
137     <dependency>
138         <groupId>io.cucumber</groupId>
139         <artifactId>cucumber-gherkin</artifactId>
140         <version>${cucumber.java.version}</version>
141     </dependency>
142
143     <!-- https://mvnrepository.com/artifact/io.cucumber/gherkin -->
144     <dependency>
145         <groupId>io.cucumber</groupId>
146         <artifactId>gherkin</artifactId>
147         <version>${gherkin.version}</version>
148     </dependency>
149
150     <!-- https://mvnrepository.com/artifact/com.aventstack/extentreports -->
151     <dependency>
152         <groupId>com.aventstack</groupId>
153         <artifactId>extentreports</artifactId>
154         <version>${extentreports.version}</version>
155     </dependency>
156
157     <!-- https://mvnrepository.com/artifact/tech.grasshopper/extentreports-cucumber7-adapter -->
158     <dependency>
159         <groupId>tech.grasshopper</groupId>
160         <artifactId>extentreports-cucumber7-adapter</artifactId>
161         <version>${extentreports-cucumber7-adapter.version}</version>
162     </dependency>
163
164     <!-- https://mvnrepository.com/artifact/io.qameta.allure/allure-cucumber7-jvm -->
165     <dependency>
166         <groupId>io.qameta.allure</groupId>
167         <artifactId>allure-cucumber7-jvm</artifactId>
168         <version>${allure-testng.version}</version>
169     </dependency>
170
171     <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
172     <dependency>
173         <groupId>org.seleniumhq.selenium</groupId>
174         <artifactId>selenium-java</artifactId>
```

```
175      <version>${selenium.version}</version>
176    </dependency>
177
178    <!-- https://mvnrepository.com/artifact/org.testng/testng -->
179    <dependency>
180      <groupId>org.testng</groupId>
181      <artifactId>testng</artifactId>
182      <version>${testng.version}</version>
183    </dependency>
184
185    <!--https://mvnrepository.com/artifact/com.sun.mail/javax.mail/ -->
186    <dependency>
187      <groupId>com.sun.mail</groupId>
188      <artifactId>javax.mail</artifactId>
189      <version>${javax.mail.version}</version>
190    </dependency>
191
192    <!-- https://mvnrepository.com/artifact/org.zeroturnaround/zt-zip -->
193    <dependency>
194      <groupId>org.zeroturnaround</groupId>
195      <artifactId>zt-zip</artifactId>
196      <version>${zip.version}</version>
197    </dependency>
198
199    <!-- https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-api -->
200    <dependency>
201      <groupId>org.apache.logging.log4j</groupId>
202      <artifactId>log4j-api</artifactId>
203      <version>2.24.3</version>
204    </dependency>
205    <!-- https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-core -->
206    <dependency>
207      <groupId>org.apache.logging.log4j</groupId>
208      <artifactId>log4j-core</artifactId>
209      <version>2.24.3</version>
210    </dependency>
211
212    <!-- https://mvnrepository.com/artifact/org.slf4j/slf4j-api -->
213    <dependency>
214      <groupId>org.slf4j</groupId>
215      <artifactId>slf4j-api</artifactId>
216      <version>2.0.16</version>
217    </dependency>
218    <!-- https://mvnrepository.com/artifact/org.slf4j/slf4j-simple -->
219    <dependency>
220      <groupId>org.slf4j</groupId>
221      <artifactId>slf4j-simple</artifactId>
222      <version>2.0.16</version>
223    </dependency>
224
225    <!-- https://mvnrepository.com/artifact/io.qameta.allure/allure-testng -->
226    <dependency>
227      <groupId>io.qameta.allure</groupId>
228      <artifactId>allure-testng</artifactId>
229      <version>${allure-testng.version}</version>
230    </dependency>
231    <dependency>
232      <groupId>io.qameta.allure</groupId>
```

```
233         <artifactId>allure-attachments</artifactId>
234         <version>${allure-testng.version}</version>
235     </dependency>
236
237     <!-- https://mvnrepository.com/artifact/com.github.automatedowl/allure-environment-writer -->
238     <dependency>
239         <groupId>com.github.automatedowl</groupId>
240         <artifactId>allure-environment-writer</artifactId>
241         <version>${allure-environment-writer.version}</version>
242     </dependency>
243
244     <!-- https://mvnrepository.com/artifact/org.apache.maven.plugins/maven-compiler-plugin -->
245     <dependency>
246         <groupId>org.apache.maven.plugins</groupId>
247         <artifactId>maven-compiler-plugin</artifactId>
248         <version>3.14.0</version>
249         <type>maven-plugin</type>
250     </dependency>
251
252     <!-- https://mvnrepository.com/artifact/org.apache.maven.plugins/maven-surefire-plugin -->
253     <dependency>
254         <groupId>org.apache.maven.plugins</groupId>
255         <artifactId>maven-surefire-plugin</artifactId>
256         <version>${maven-surefire-plugin.version}</version>
257     </dependency>
258
259     <!-- https://mvnrepository.com/artifact/org.apache.poi/poi -->
260     <dependency>
261         <groupId>org.apache.poi</groupId>
262         <artifactId>poi</artifactId>
263         <version>${apache-poi.version}</version>
264     </dependency>
265     <!-- https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml -->
266     <dependency>
267         <groupId>org.apache.poi</groupId>
268         <artifactId>poi-ooxml</artifactId>
269         <version>${apache-poi.version}</version>
270     </dependency>
271
272     <!-- https://mvnrepository.com/artifact/commons-io/commons-io -->
273     <dependency>
274         <groupId>commons-io</groupId>
275         <artifactId>commons-io</artifactId>
276         <version>${commons-io.version}</version>
277     </dependency>
278
279     <!-- https://mvnrepository.com/artifact/com.github.stephenc.monte/monte-screen-recorder -->
280     <dependency>
281         <groupId>com.github.stephenc.monte</groupId>
282         <artifactId>monte-screen-recorder</artifactId>
283         <version>${monte-screen-recorder.version}</version>
284     </dependency>
285
286     <!-- https://mvnrepository.com/artifact/org.aeonbits.owner/owner -->
287     <dependency>
288         <groupId>org.aeonbits.owner</groupId>
289         <artifactId>owner</artifactId>
290         <version>${owner.version}</version>
```

```
291      </dependency>
292
293      <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
294      <dependency>
295          <groupId>org.projectlombok</groupId>
296          <artifactId>lombok</artifactId>
297          <version>${lombok.version}</version>
298          <scope>provided</scope>
299      </dependency>
300
301      <!-- https://mvnrepository.com/artifact/com.mysql/mysql-connector-j -->
302      <dependency>
303          <groupId>com.mysql</groupId>
304          <artifactId>mysql-connector-j</artifactId>
305          <version>${mysql-connector-j.version}</version>
306      </dependency>
307
308      <!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
309      <dependency>
310          <groupId>mysql</groupId>
311          <artifactId>mysql-connector-java</artifactId>
312          <version>${mysql-connector-java.version}</version>
313      </dependency>
314
315      <!-- https://mvnrepository.com/artifact/org.aspectj/aspectjweaver -->
316      <dependency>
317          <groupId>org.aspectj</groupId>
318          <artifactId>aspectjweaver</artifactId>
319          <version>${aspectjweaver.version}</version>
320      </dependency>
321
322  </dependencies>
323
324  <!-- This profile execute the TestNG suite inside the suites folder on test/resources/suites -->
325  <profiles>
326
327      <profile>
328          <id>feature-gherkin</id>
329          <build>
330              <plugins>
331                  <plugin>
332                      <groupId>org.apache.maven.plugins</groupId>
333                      <artifactId>maven-surefire-plugin</artifactId>
334                      <version>${maven-surefire-plugin.version}</version>
335                      <configuration>
336                          <suiteXmlFiles>
337                              <suiteXmlFile>${suite.feature.login.cms}</suiteXmlFile>
338                          </suiteXmlFiles>
339                      </configuration>
340                  </plugin>
341              </plugins>
342          </build>
343      </profile>
344
345  </profiles>
346
347  <build>
348
```

```

349     <plugins>
350
351         <plugin>
352             <groupId>org.apache.maven.plugins</groupId>
353             <artifactId>maven-compiler-plugin</artifactId>
354             <version>3.13.0</version>
355             <configuration>
356                 <source>${java.compiler.version}</source>
357                 <target>${java.compiler.version}</target>
358             </configuration>
359         </plugin>
360
361         <plugin>
362             <groupId>org.apache.maven.plugins</groupId>
363             <artifactId>maven-surefire-plugin</artifactId>
364             <version>${maven-surefire-plugin.version}</version>
365             <configuration>
366                 <suiteXmlFiles>
367                     <!-- Call Suite name global set up above -->
368                     <suiteXmlFile>${suite.feature.login.cms}</suiteXmlFile>
369                 </suiteXmlFiles>
370
371                 <rerunFailingTestsCount>1</rerunFailingTestsCount>
372                 <argLine>
373                     -Dfile.encoding=UTF-8
374                     -
375                     javaagent:"${settings.localRepository}/org/aspectj/aspectjweaver/${aspectjweaver.version}/aspectjweaver-${aspectjweaver.version}.jar"
376                     </argLine>
377                     <reportsDirectory>target/test-output</reportsDirectory>
378                     <testFailureIgnore>true</testFailureIgnore>
379                     <systemPropertyVariables>
380                         <allure.results.directory>target/allure-results</allure.results.directory>
381                     </systemPropertyVariables>
382
383             </configuration>
384         </plugin>
385
386         <plugin>
387             <groupId>io.qameta.allure</groupId>
388             <artifactId>allure-maven</artifactId>
389             <version>${allure-maven.version}</version>
390             <configuration>
391                 <reportVersion>${allure-testng.version}</reportVersion>
392                 <allureDownloadUrl>
393                     ${allure.cmd.download.url}/${allure-testng.version}/allure-commandline-${allure-testng.version}.zip
394                 </allureDownloadUrl>
395             </configuration>
396         </plugin>
397
398         <plugin>
399             <!--https://mvnrepository.com/artifact/net.masterthought/maven-cucumber-reporting-->
400             <groupId>net.masterthought</groupId>
401             <artifactId>maven-cucumber-reporting</artifactId>
402             <version>5.8.5</version>
403             <executions>
404                 <execution>
```

```

404             <id>generate-cucumber-report</id>
405             <phase>test</phase>
406             <goals>
407                 <goal>generate</goal>
408             </goals>
409             <configuration>
410                 <projectName>AutomationFrameworkCucumberTestNG</projectName>
411                 <skip>false</skip>
412                 <outputDirectory>exports/reports/CucumberReports/cucumber-reports
413                 </outputDirectory>
414                 <inputDirectory>target/cucumber-reports</inputDirectory>
415                 <jsonFiles>
416                     <param>**/*.json</param>
417                 </jsonFiles>
418             </configuration>
419         </execution>
420     </executions>
421 </plugin>
422
423     <plugin>
424         <groupId>org.apache.maven.plugins</groupId>
425         <artifactId>maven-javadoc-plugin</artifactId>
426         <version>3.6.3</version>
427         <configuration>
428             <source>${java-compiler.version}</source>
429             <encoding>UTF-8</encoding>
430         </configuration>
431     </plugin>
432
433 </plugins>
434
435 </build>
436
437 </project>

```

format 2

```

1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4
5   <groupId>org.example</groupId>
6   <artifactId>Selenium4TestAutomation</artifactId>
7   <version>1.0-SNAPSHOT</version>
8   <packaging>jar</packaging>
9
10  <name>Selenium4TestAutomation</name>
11  <url>http://maven.apache.org</url>
12
13  <properties>
14    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
15    <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
16    <testNGFile></testNGFile>
17    <runReport></runReport>
18    <exeflag></exeflag>
19    <run></run>
20    <envValue></envValue>

```

```
21 <browserType></browserType>
22 <!--      <maven.compiler.source>1.8</maven.compiler.source>
23 <maven.compiler.target>1.8</maven.compiler.target>  -->
24
25 </properties>
26
27 <build>
28   <plugins>
29     <plugin>
30       <groupId>org.apache.maven.plugins</groupId>
31       <artifactId>maven-compiler-plugin</artifactId>
32       <version>3.5.1</version>
33       <configuration>
34         <source>1.8</source>
35         <target>1.8</target>
36       </configuration>
37     </plugin>
38     <plugin>
39       <groupId>org.apache.maven.plugins</groupId>
40       <artifactId>maven-surefire-plugin</artifactId>
41       <version>2.22.0</version>
42       <configuration>
43         <suiteXmlFiles>
44           <suiteXmlFile>src/test/resources/TestNGSuiteFiles/${testNGFile}.xml</suiteXmlFile>
45
46       </suiteXmlFiles>
47
48       <systemPropertyVariables>
49         <EnvironmentValue>${envValue}</EnvironmentValue>
50         <browserType>${browserType}</browserType>
51       </systemPropertyVariables>
52     </configuration>
53   </plugin>
54 </plugins>
55 </build>
56
57 <dependencies>
58
59
60
61
62 <!-- https://mvnrepository.com/artifact/tech.grasshopper/extentreports-cucumber7-adapter -->
63 <dependency>
64   <groupId>tech.grasshopper</groupId>
65   <artifactId>extentreports-cucumber7-adapter</artifactId>
66   <version>1.7.0</version>
67 </dependency>
68 <!-- https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager -->
69 <dependency>
70   <groupId>io.github.bonigarcia</groupId>
71   <artifactId>webdrivermanager</artifactId>
72   <version>5.9.2</version>
73 </dependency>
74
75 <!-- https://mvnrepository.com/artifact/org.json/json -->
76 <!-- https://mvnrepository.com/artifact/org.json/json -->
77 <dependency>
78   <groupId>org.json</groupId>
```

```
79      <artifactId>json</artifactId>
80      <version>20090211</version>
81    </dependency>
82
83    <dependency>
84      <groupId>junit</groupId>
85      <artifactId>junit</artifactId>
86      <version>4.12</version>
87      <scope>compile</scope>
88    </dependency>
89    <dependency>
90      <groupId>com.deque</groupId>
91      <artifactId>axe-selenium</artifactId>
92      <version>2.1</version>
93    </dependency>
94
95    <!-- https://mvnrepository.com/artifact/info.cukes/cucumber-java
96    <dependency>
97      <groupId>info.cukes</groupId>
98      <artifactId>cucumber-java</artifactId>
99    </dependency> -->
100
101   <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-java -->
102   <dependency>
103     <groupId>io.cucumber</groupId>
104     <artifactId>cucumber-java</artifactId>
105     <version>7.3.4</version>
106   </dependency>
107
108
109
110  <!-- https://mvnrepository.com/artifact/info.cukes/cucumber-core
111  <dependency>
112    <groupId>info.cukes</groupId>
113    <artifactId>cucumber-core</artifactId>
114    <version>1.2.5</version>
115  </dependency> -->
116
117
118  <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-core -->
119  <dependency>
120    <groupId>io.cucumber</groupId>
121    <artifactId>cucumber-core</artifactId>
122    <version>7.3.4</version>
123  </dependency>
124
125
126
127  <!-- https://mvnrepository.com/artifact/info.cukes/cucumber-testng
128  <dependency>
129    <groupId>info.cukes</groupId>
130    <artifactId>cucumber-testng</artifactId>
131    <version>1.2.5</version>
132  </dependency> -->
133
134
135  <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-testng -->
136  <dependency>
```

```
137      <groupId>io.cucumber</groupId>
138      <artifactId>cucumber-testng</artifactId>
139      <version>7.3.4</version>
140    </dependency>
141
142
143    <!-- https://mvnrepository.com/artifact/info.cukes/cucumber-jvm-deps -->
144    <dependency>
145      <groupId>info.cukes</groupId>
146      <artifactId>cucumber-jvm-deps</artifactId>
147      <version>1.0.5</version>
148      <scope>provided</scope>
149    </dependency>
150
151    <!-- https://mvnrepository.com/artifact/info.cukes/gherkin -->
152    <dependency>
153      <groupId>info.cukes</groupId>
154      <artifactId>gherkin</artifactId>
155      <version>2.12.2</version>
156      <scope>provided</scope>
157    </dependency>
158    <!-- https://mvnrepository.com/artifact/junit/junit -->
159    <dependency>
160      <groupId>junit</groupId>
161      <artifactId>junit</artifactId>
162      <version>4.12</version>
163      <scope>test</scope>
164    </dependency>
165    <!-- https://mvnrepository.com/artifact/info.cukes/cucumber-junit -->
166    <dependency>
167      <groupId>info.cukes</groupId>
168      <artifactId>cucumber-junit</artifactId>
169      <version>1.2.5</version>
170      <scope>test</scope>
171    </dependency>
172
173
174
175
176    <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-server -->
177    <dependency>
178      <groupId>org.seleniumhq.selenium</groupId>
179      <artifactId>selenium-server</artifactId>
180      <version>3.141.59</version>
181    </dependency>
182
183    <!-- https://mvnrepository.com/artifact/net.bytebuddy/byte-buddy-->
184    <dependency>
185      <groupId>net.bytebuddy</groupId>
186      <artifactId>byte-buddy</artifactId>
187      <version>1.8.15</version>
188    </dependency>
189    <!-- https://mvnrepository.com/artifact/log4j/log4j -->
190    <dependency>
191      <groupId>log4j</groupId>
192      <artifactId>log4j</artifactId>
193      <version>1.2.12</version>
194    </dependency>
```

```
195
196    <!-- https://mvnrepository.com/artifact/com.github.2gis.winium/winium-webdriver -->
197    <dependency>
198        <groupId>com.github.2gis.winium</groupId>
199        <artifactId>winium-webdriver</artifactId>
200        <version>0.1.0-1</version>
201    </dependency>
202
203    <!-- https://mvnrepository.com/artifact/com.github.2gis.winium/winium-elements-desktop -->
204    <dependency>
205        <groupId>com.github.2gis.winium</groupId>
206        <artifactId>winium-elements-desktop</artifactId>
207        <version>0.2.0-1</version>
208    </dependency>
209
210    <!-- https://mvnrepository.com/artifact/net.sourceforge.jexcelapi/jxl -->
211    <dependency>
212        <groupId>net.sourceforge.jexcelapi</groupId>
213        <artifactId>jxl</artifactId>
214        <version>2.6.10</version>
215    </dependency>
216    <!-- https://mvnrepository.com/artifact/com.codoid.products/fillo -->
217    <dependency>
218        <groupId>com.codoid.products</groupId>
219        <artifactId>fillo</artifactId>
220        <version>1.18</version>
221    </dependency>
222
223
224    <!-- https://mvnrepository.com/artifact/com.aventstack/extentreports
225    <dependency>
226        <groupId>com.aventstack</groupId>
227        <artifactId>extentreports</artifactId>
228        <version>3.0.7</version>
229        <scope>provided</scope>
230    </dependency>  -->
231
232
233    <!-- https://mvnrepository.com/artifact/org.freemarker/freemarker -->
234    <dependency>
235        <groupId>org.freemarker</groupId>
236        <artifactId>freemarker</artifactId>
237        <version>2.3.23</version>
238    </dependency>
239    <!-- https://mvnrepository.com/artifact/org.mongodb/mongo-java-driver -->
240    <dependency>
241        <groupId>org.mongodb</groupId>
242        <artifactId>mongo-java-driver</artifactId>
243        <version>2.10.1</version>
244    </dependency>
245    <!-- https://mvnrepository.com/artifact/joda-time/joda-time -->
246    <dependency>
247        <groupId>joda-time</groupId>
248        <artifactId>joda-time</artifactId>
249        <version>2.10</version>
250    </dependency>
251    <!-- https://mvnrepository.com/artifact/com.cedarsoftware/json-io -->
252    <dependency>
```

```
253     <groupId>com.cedarsoftware</groupId>
254     <artifactId>json-io</artifactId>
255     <version>4.10.0</version>
256   </dependency>
257   <!-- https://mvnrepository.com/artifact/commons-lang/commons-lang -->
258   <dependency>
259     <groupId>commons-lang</groupId>
260     <artifactId>commons-lang</artifactId>
261     <version>2.6</version>
262   </dependency>
263
264
265   <!-- <dependency>
266     <groupId>info.cukes</groupId>
267     <artifactId>cucumber-java</artifactId>
268     <version>1.2.5</version>
269   </dependency> -->
270   <!-- <dependency>
271     <groupId>com.ram.cucumber</groupId>
272     <artifactId>cucumber</artifactId>
273     <version>0.0.1</version>
274   </dependency> -->
275   <!-- https://mvnrepository.com/artifact/org.testng/testng -->
276   <dependency>
277     <groupId>org.testng</groupId>
278     <artifactId>testng</artifactId>
279     <version>6.14.3</version>
280   </dependency>
281
282   <!-- https://mvnrepository.com/artifact/io.rest-assured/rest-assured -->
283   <dependency>
284     <groupId>io.rest-assured</groupId>
285     <artifactId>rest-assured</artifactId>
286     <version>3.0.0</version>
287     <scope>test</scope>
288   </dependency>
289
290   <!-- <dependency>
291     <groupId>groovy</groupId>
292     <artifactId>groovy</artifactId>
293     <version>2.4.6</version>
294   </dependency> -->
295   <!-- https://mvnrepository.com/artifact/org.codehaus.groovy/groovy-json -->
296   <dependency>
297     <groupId>org.codehaus.groovy</groupId>
298     <artifactId>groovy-json</artifactId>
299     <version>2.4.6</version>
300   </dependency>
301
302   <!-- https://mvnrepository.com/artifact/org.codehaus.groovy/groovy-xml -->
303   <dependency>
304     <groupId>org.codehaus.groovy</groupId>
305     <artifactId>groovy-xml</artifactId>
306     <version>2.4.6</version>
307   </dependency>
308
309   <!-- https://mvnrepository.com/artifact/org.hamcrest/hamcrest-core -->
310   <dependency>
```

```
311     <groupId>org.hamcrest</groupId>
312     <artifactId>hamcrest-core</artifactId>
313     <version>1.3</version>
314     <scope>test</scope>
315   </dependency>
316
317
318   <!-- https://mvnrepository.com/artifact/org.hamcrest/hamcrest-library -->
319   <dependency>
320     <groupId>org.hamcrest</groupId>
321     <artifactId>hamcrest-library</artifactId>
322     <version>1.3</version>
323     <scope>test</scope>
324   </dependency>
325
326   <!-- https://mvnrepository.com/artifact/org.apache.httpcomponents/httpclient -->
327   <dependency>
328     <groupId>org.apache.httpcomponents</groupId>
329     <artifactId>httpclient</artifactId>
330     <version>4.5.2</version>
331   </dependency>
332
333   <!-- https://mvnrepository.com/artifact/org.apache.httpcomponents/httpcore -->
334   <dependency>
335     <groupId>org.apache.httpcomponents</groupId>
336     <artifactId>httpcore</artifactId>
337     <version>4.4.4</version>
338   </dependency>
339
340   <!-- https://mvnrepository.com/artifact/org.apache.httpcomponents/httpmime -->
341   <dependency>
342     <groupId>org.apache.httpcomponents</groupId>
343     <artifactId>httpmime</artifactId>
344     <version>4.5.1</version>
345   </dependency>
346
347   <!-- https://mvnrepository.com/artifact/io.rest-assured/json-path -->
348   <dependency>
349     <groupId>io.rest-assured</groupId>
350     <artifactId>json-path</artifactId>
351     <version>3.0.0</version>
352   </dependency>
353
354   <!-- https://mvnrepository.com/artifact/io.rest-assured/rest-assured-common -->
355   <dependency>
356     <groupId>io.rest-assured</groupId>
357     <artifactId>rest-assured-common</artifactId>
358     <version>3.0.0</version>
359   </dependency>
360
361   <!-- https://mvnrepository.com/artifact/io.rest-assured/xml-path -->
362   <dependency>
363     <groupId>io.rest-assured</groupId>
364     <artifactId>xml-path</artifactId>
365     <version>3.0.0</version>
366   </dependency>
367
368   <!-- https://mvnrepository.com/artifact/org.ccil.cowan.tagsoup/tagsoup -->
```

```
369 <dependency>
370   <groupId>org.ccil.cowan.tagsoup</groupId>
371   <artifactId>tagsoup</artifactId>
372   <version>1.2.1</version>
373 </dependency>
374
375 <!-- https://mvnrepository.com/artifact/commons-codec/commons-codec -->
376 <dependency>
377   <groupId>commons-codec</groupId>
378   <artifactId>commons-codec</artifactId>
379   <version>1.9</version>
380 </dependency>
381
382 <!-- <dependency>
383   <groupId>commons-lang</groupId>
384   <artifactId>commons-lang</artifactId>
385   <version>3.3.4</version>
386 </dependency> -->
387 <!-- https://mvnrepository.com/artifact/commons-logging/commons-logging -->
388 <dependency>
389   <groupId>commons-logging</groupId>
390   <artifactId>commons-logging</artifactId>
391   <version>1.2</version>
392 </dependency>
393
394 <dependency>
395   <groupId>com.microsoft.azure</groupId>
396   <artifactId>azure-functions-java-core</artifactId>
397   <version>1.0.0-beta-3</version>
398 </dependency>
399 <dependency>
400   <groupId>com.opencsv</groupId>
401   <artifactId>opencsv</artifactId>
402   <version>4.1</version>
403 </dependency>
404
405
406 <dependency>
407   <groupId>com.relevantcodes</groupId>
408   <artifactId>extentreports</artifactId>
409   <version>2.41.2</version>
410 </dependency>
411
412
413
414 <dependency>
415   <groupId>org.slf4j</groupId>
416   <artifactId>slf4j-api</artifactId>
417   <version>1.7.5</version>
418 </dependency>
419 <dependency>
420   <groupId>junit</groupId>
421   <artifactId>junit</artifactId>
422   <version>4.12</version>
423   <scope>compile</scope>
424 </dependency>
425 </dependencies>
426
```

```

427 <repositories>
428   <repository>
429     <id>sonatype-nexus-snapshots</id>
430     <url>https://oss.sonatype.org/content/repositories/snapshots</url>
431   </repository>
432 </repositories>
433
434 <pluginRepositories>
435   <pluginRepository>
436     <id>sonatype-nexus-snapshots</id>
437     <url>https://oss.sonatype.org/content/repositories/snapshots</url>
438   </pluginRepository>
439 </pluginRepositories>
440
441
442
443
444 </project>
445

```

Current xml file

List of dependencies needed

```

1 <!-- https://mvnrepository.com/artifact/tech.grasshopper/extentreports-cucumber7-adapter -->
2 <dependency>
3   <groupId>tech.grasshopper</groupId>
4   <artifactId>extentreports-cucumber7-adapter</artifactId>
5   <version>1.7.0</version>
6 </dependency>

```

```

1 <!-- https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager -->
2 <dependency>
3   <groupId>io.github.bonigarcia</groupId>
4   <artifactId>webdrivermanager</artifactId>
5   <version>6.0.0</version>
6 </dependency>

```

```

1 <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
2 <dependency>
3   <groupId>org.seleniumhq.selenium</groupId>
4   <artifactId>selenium-java</artifactId>
5   <version>4.30.0</version>
6 </dependency>

```

```

1 <!-- https://mvnrepository.com/artifact/org.testng/testng -->
2 <dependency>
3   <groupId>org.testng</groupId>
4   <artifactId>testng</artifactId>
5   <version>7.11.0</version>
6   <scope>test</scope>
7 </dependency>

```

```

1 <!-- https://mvnrepository.com/artifact/org.apache.poi/poi -->
2 <dependency>

```

```
3   <groupId>org.apache.poi</groupId>
4     <artifactId>poi</artifactId>
5     <version>5.4.0</version>
6   </dependency>
```

```
1 <!-- https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml -->
2 <dependency>
3   <groupId>org.apache.poi</groupId>
4     <artifactId>poi-ooxml</artifactId>
5     <version>5.4.0</version>
6 </dependency>
```

```
1 <!-- https://mvnrepository.com/artifact/org.apache.poi/poi-scratchpad -->
2 <dependency>
3   <groupId>org.apache.poi</groupId>
4     <artifactId>poi-scratchpad</artifactId>
5     <version>5.4.0</version>
6 </dependency>
```

```
1 <!-- https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml-schemas -->
2 <dependency>
3   <groupId>org.apache.poi</groupId>
4     <artifactId>poi-ooxml-schemas</artifactId>
5     <version>4.1.2</version>
6 </dependency>
```

```
1 <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-java -->
2 <dependency>
3   <groupId>io.cucumber</groupId>
4     <artifactId>cucumber-java</artifactId>
5     <version>7.21.1</version>
6 </dependency>
```

```
1 <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-core -->
2 <dependency>
3   <groupId>io.cucumber</groupId>
4     <artifactId>cucumber-core</artifactId>
5     <version>7.21.1</version>
6 </dependency>
```

```
1 <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-testng -->
2 <dependency>
3   <groupId>io.cucumber</groupId>
4     <artifactId>cucumber-testng</artifactId>
5     <version>7.21.1</version>
6 </dependency>
```

```
1 <!-- https://mvnrepository.com/artifact/com.aventstack/extentreports -->
2 <dependency>
3   <groupId>com.aventstack</groupId>
4     <artifactId>extentreports</artifactId>
5     <version>5.1.2</version>
6 </dependency>
```

```
1 <!-- https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-core -->
2 <dependency>
3   <groupId>org.apache.logging.log4j</groupId>
```

```
4     <artifactId>log4j-core</artifactId>
5         <version>2.24.3</version>
6     </dependency>
```

```
1 <!-- https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-slf4j-impl -->
2 <dependency>
3     <groupId>org.apache.logging.log4j</groupId>
4     <artifactId>log4j-slf4j-impl</artifactId>
5     <version>2.24.3</version>
6     <scope>compile</scope>
7 </dependency>
```

```
1 <!-- https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-api -->
2 <dependency>
3     <groupId>org.apache.logging.log4j</groupId>
4     <artifactId>log4j-api</artifactId>
5     <version>2.24.3</version>
6 </dependency>
```

Now add other tags

```
1 <repositories>
2     <repository>
3         <id>sonatype-nexus-snapshots</id>
4         <url>https://oss.sonatype.org/content/repositories/snapshots</url>
5     </repository>
6 </repositories>
7
8 <pluginRepositories>
9     <pluginRepository>
10        <id>sonatype-nexus-snapshots</id>
11        <url>https://oss.sonatype.org/content/repositories/snapshots</url>
12    </pluginRepository>
13 </pluginRepositories>
14
```

```
1 <properties>
2     <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
3     <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
4     <testNGFile></testNGFile>
5     <runReport></runReport>
6     <exeflag></exeflag>
7     <run></run>
8     <envValue></envValue>
9     <browserType></browserType>
10    <!--      <maven.compiler.source>1.8</maven.compiler.source>
11    <maven.compiler.target>1.8</maven.compiler.target>  -->
12
13 </properties>
14
15 <build>
16     <plugins>
17         <plugin>
18             <groupId>org.apache.maven.plugins</groupId>
19             <artifactId>maven-compiler-plugin</artifactId>
20             <version>3.5.1</version>
21             <configuration>
```

```

22      <source>1.8</source>
23      <target>1.8</target>
24    </configuration>
25  </plugin>
26  <plugin>
27    <groupId>org.apache.maven.plugins</groupId>
28    <artifactId>maven-surefire-plugin</artifactId>
29    <version>2.22.0</version>
30    <configuration>
31      <suiteXmlFiles>
32        <suiteXmlFile>src/test/resources/TestNGSuiteFiles/${testNGFile}.xml</suiteXmlFile>
33
34      </suiteXmlFiles>
35
36      <systemPropertyVariables>
37        <EnvironmentValue>${envValue}</EnvironmentValue>
38        <browserType>${browserType}</browserType>
39      </systemPropertyVariables>
40    </configuration>
41  </plugin>
42 </plugins>
43 </build>
44

```

1. extentreports-cucumber7-adapter
2. webdrivermanager
3. selenium-java
4. testng
5. poi
6. poi-ooxml
7. poi-scratchpad
8. poi-ooxml-schemas
9. cucumber-java
10. cucumber-core
11. cucumber-testng
12. extentreports
13. log4j-core
14. log4j-slf4j-impl
15. log4j-api

```

1 <project xmlns="https://maven.apache.org/POM/4.0.0"
2   xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
4   <modelVersion>4.0.0</modelVersion>
5   <groupId>com.testAutomation</groupId>
6   <artifactId>com.testAutomation</artifactId>
7   <version>0.0.1-SNAPSHOT</version>
8
9 <dependencies>
10
11   <!-- https://mvnrepository.com/artifact/tech.grasshopper/extentreports-cucumber7-adapter -->

```

```
12     <dependency>
13         <groupId>tech.grasshopper</groupId>
14         <artifactId>extentreports-cucumber7-adapter</artifactId>
15         <version>1.7.0</version>
16     </dependency>
17
18     <!-- https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager -->
19     <dependency>
20         <groupId>io.github.bonigarcia</groupId>
21         <artifactId>webdrivermanager</artifactId>
22         <version>5.9.2</version>
23     </dependency>
24     <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
25     <dependency>
26         <groupId>org.seleniumhq.selenium</groupId>
27         <artifactId>selenium-java</artifactId>
28         <version>4.28.1</version>
29     </dependency>
30
31     <!-- https://mvnrepository.com/artifact/org.testng/testng -->
32     <dependency>
33         <groupId>org.testng</groupId>
34         <artifactId>testng</artifactId>
35         <version>7.10.1</version>
36         <scope>test</scope>
37     </dependency>
38
39     <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-java -->
40     <dependency>
41         <groupId>io.cucumber</groupId>
42         <artifactId>cucumber-java</artifactId>
43         <version>7.20.1</version>
44     </dependency>
45
46     <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-core -->
47     <dependency>
48         <groupId>io.cucumber</groupId>
49         <artifactId>cucumber-core</artifactId>
50         <version>7.20.1</version>
51     </dependency>
52
53     <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-testng -->
54     <dependency>
55         <groupId>io.cucumber</groupId>
56         <artifactId>cucumber-testng</artifactId>
57         <version>7.20.1</version>
58     </dependency>
59
60     <!-- https://mvnrepository.com/artifact/org.apache.poi/poi -->
61     <dependency>
62         <groupId>org.apache.poi</groupId>
63         <artifactId>poi</artifactId>
64         <version>5.3.0</version>
65     </dependency>
66
67     <!-- https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml -->
68     <dependency>
69         <groupId>org.apache.poi</groupId>
```

```
70         <artifactId>poi-ooxml</artifactId>
71         <version>5.3.0</version>
72     </dependency>
73
74     <!-- https://mvnrepository.com/artifact/org.apache.poi/poi-scratchpad -->
75     <dependency>
76         <groupId>org.apache.poi</groupId>
77         <artifactId>poi-scratchpad</artifactId>
78         <version>5.3.0</version>
79     </dependency>
80
81     <!-- https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml-schemas -->
82     <dependency>
83         <groupId>org.apache.poi</groupId>
84         <artifactId>poi-ooxml-schemas</artifactId>
85         <version>4.1.2</version>
86     </dependency>
87
88     <!-- https://mvnrepository.com/artifact/com.aventstack/extentreports -->
89     <dependency>
90         <groupId>com.aventstack</groupId>
91         <artifactId>extentreports</artifactId>
92         <version>5.1.2</version>
93     </dependency>
94
95     <!-- https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-core -->
96     <dependency>
97         <groupId>org.apache.logging.log4j</groupId>
98         <artifactId>log4j-core</artifactId>
99         <version>2.24.3</version>
100    </dependency>
101
102    <!-- https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-api -->
103    <dependency>
104        <groupId>org.apache.logging.log4j</groupId>
105        <artifactId>log4j-api</artifactId>
106        <version>2.24.3</version>
107    </dependency>
108
109    </dependencies>
110
111    <build>
112        <plugins>
113            <plugin>
114                <groupId>org.apache.maven.plugins</groupId>
115                <artifactId>maven-compiler-plugin</artifactId>
116                <version>3.5.1</version>
117                <configuration>
118                    <source>21</source>
119                    <target>21</target>
120                </configuration>
121            </plugin>
122            <plugin>
123                <groupId>org.apache.maven.plugins</groupId>
124                <artifactId>maven-surefire-plugin</artifactId>
125                <version>2.22.0</version>
126                <configuration>
127                    <suiteXmlFiles>
```

```

128         <suiteXmlFile>src/test/resources/TestNGSuiteFiles/${testNGFile}.xml</suiteXmlFile>
129
130     </suiteXmlFiles>
131
132     <systemPropertyVariables>
133         <EnvironmentValue>${envValue}</EnvironmentValue>
134         <browserType>${browserType}</browserType>
135     </systemPropertyVariables>
136
137     </configuration>
138
139 </plugins>
140
141 </build>
142
143 </project>

```

i Create Maven project in Eclipse IDE

Creating Feature Files and run features from run configuration

Installing cucumber eclipse IDE plugin. Restart eclipse after plugin installation

Right click on the project > configure > convert to cucumber project, cucumber green symbol will be displayed in project.

Create feature files and keep it under src/test/resources with .feature extension. Feature files doesn't have any code, its plain English using gherkin language/syntax

Creating Scenarios in feature files and run them using run configurations

Right click and select pretty format yellow highlight as glue code is not there

Run the feature file , step def methods will be shown in console

To Run - Run Configuration > select feature option > enter feature file path

create package features, create file login.feature

```

1 Feature: Login Feature
2
3 In order to perform successful login
4 As a user I have to enter correct username and password
5
6 Scenario: Login to the Application
7 Given user navigates to the application
8 When user validates the home page title
9 Then user enters username
10 And user enters password
11 And user clicks on SignIn button
12
13

```

- Create steps package under src/test/java under this create a new java class > loginSteps.java and paste the auto suggested glue code from console here.

- Import the relevant packages
- file > maven > update project, yellow sign will go
- Again run the feature file and check, you will see green sign for each step once feature file is mapped to steps definitions
- CTRL+click on feature steps will navigate to steps
- code written inside steps will have selenium/ api/ appium any code will run

Now create the Runner file (java file) to run our feature file

- Under src/test/java create a new package called runner
- Create java class RunLogin.java

```

1 package runner;
2 import io.cucumber.testng.AbstractTestNGCucumberTests;
3 import io.cucumber.testng.CucumberOptions;
4 @CucumberOptions(features="src/test/resources/features/login.feature",
5 glue= {"steps"}
6 )
7 public class RunLogin extends AbstractTestNGCucumberTests {
8 }
9
10 //run the runner class as TestNG

```

Cucumber Report Implementation

After the runner class is implemented lets implement cucumber report in our framework modified runner after implementing plugin

```

1 package runner;
2 import io.cucumber.testng.AbstractTestNGCucumberTests;
3 import io.cucumber.testng.CucumberOptions;
4 @CucumberOptions(features="src/test/resources/features/login.feature",
5 glue= {"steps"},
6 plugin= {"html:target/cucumber-reports/cucumber-html-report.html"}
7 )
8 public class RunLogin extends AbstractTestNGCucumberTests {
9 }
10 //run the runner class and check for the report in Target folder after refreshing the project

```

Extent-Report Implementation

- extent report implementation from tech.grasshopper
- search for - extent reports-cucumber 7 Adapter
- Add the latest maven dependency in pom.xml file
- add extent report latest version dependency also in pom.xml file
- Now add extent reports-cucumber 7 Adapter as plugin in your runner file
 - ↳ <https://extentsreports.com/docs/versions/4/java/cucumber4.html>
- ref- ↳ <https://extentsreports.com/docs/versions/4/java/spark-reporter.html>

```

1 package runner;
2 import io.cucumber.testng.AbstractTestNGCucumberTests;
3 import io.cucumber.testng.CucumberOptions;
4 @CucumberOptions(features="src/test/resources/features/login.feature",
5 glue= {"steps"},
6 plugin= {"html:target/cucumber-reports/cucumber-html-report.html",
7 "com.aventstack.extentreports.cucumber.adapter.ExtentCucumberAdapter:"}
8 )
9 public class RunLogin extends AbstractTestNGCucumberTests {
10 }

```

Once the adapter plugin is configured in the runner class, under src/test/resources/ create extent.properties file so basically the extent plugin is going to refer this extent.properties file.

So report activation write

Note- There are two files extent.properties and extent-config.xml, keep these files under src/test/resources/ to make it work extent-config.xml file will be referred from extent.properties file

sample extent.properties file

```

1 # For report Activation
2 extent.reporter.spark.start=true
3 extent.reporter.spark.out=MyReports/Spark.html
4 extent.reporter.spark.config=src/test/resources/extent-config.xml

```

sample extent-config.xml file

```

1 sample extent-config.xml
2 <?xml version="1.0" encoding="UTF-8"?>
3 <extentsreports>
4 <configuration>
5 <!-- report theme -->
6 <!-- standard, dark -->
7 <theme>standard</theme>
8 <!-- enables timeline -->
9 <!-- defaults to true -->
10 <enableTimeline>true</enableTimeline>
11 <!-- document encoding -->
12 <!-- defaults to UTF-8 -->
13 <encoding>UTF-8</encoding>
14 <!-- protocol for script and stylesheets -->
15 <!-- defaults to https -->
16 <protocol>https</protocol>
17 <!-- title of the document -->
18 <documentTitle>Extent Framework</documentTitle>
19 <!-- report name - displayed at top-nav -->
20 <reportName>Build 1</reportName>
21 <!-- custom javascript -->
22 <scripts>
23 <![CDATA[
24 $(document).ready(function() {
25 });
26 ]]>
27 </scripts>
28 <!-- custom styles -->
29 <styles>

```

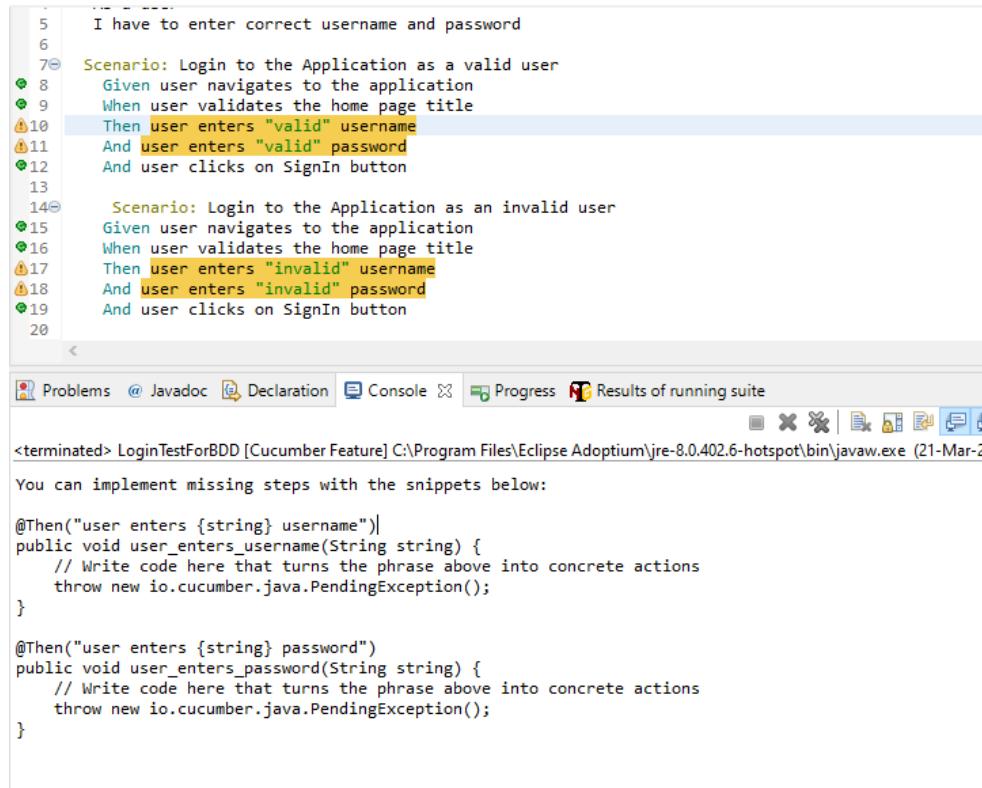
```

30  <![CDATA[
31  ]]>
32  </styles>
33  </configuration>
34  </extentreports>

```

>>

Adding multiple scenarios to a feature file



```

5   I have to enter correct username and password
6
7@ Scenario: Login to the Application as a valid user
8  Given user navigates to the application
9  When user validates the home page title
10 Then user enters "valid" username
11 And user enters "valid" password
12 And user clicks on SignIn button
13
14@ Scenario: Login to the Application as an invalid user
15 Given user navigates to the application
16 When user validates the home page title
17 Then user enters "invalid" username
18 And user enters "invalid" password
19 And user clicks on SignIn button
20

```

Problems @ Javadoc Declaration Console X Progress Results of running suite

<terminated> LoginTestForBDD [Cucumber Feature] C:\Program Files\Eclipse Adoptium\jre-8.0.402.6-hotspot\bin\javaw.exe (21-Mar-2)

You can implement missing steps with the snippets below:

```

@Then("user enters {string} username")
public void user_enters_username(String string) {
    // Write code here that turns the phrase above into concrete actions
    throw new io.cucumber.java.PendingException();
}

@Then("user enters {string} password")
public void user_enters_password(String string) {
    // Write code here that turns the phrase above into concrete actions
    throw new io.cucumber.java.PendingException();
}

```

```

@Then("user enters {string} username")
public void user_enters_username(String username) {
    System.out.println("@Then--- user enters "+username+"username");
}

@Then("user enters {string} password")
public void user_enters_password(String password) {
    System.out.println("@Then--- user enters"+password+"password");
}

```

No of {string} counts = no of parameters in the method.

```

Scenario Outline: Login to the Application with different combinations
Given user navigates to the application
When user validates the home page title
Then user enters "<username>" username
And user enters "<password>" password
And user clicks on SignIn button

Examples:
| username | password |
| valid    | valid    |
| invalid  | invalid  |
| valid    | invalid  |
| invalid  | valid    |

```

```

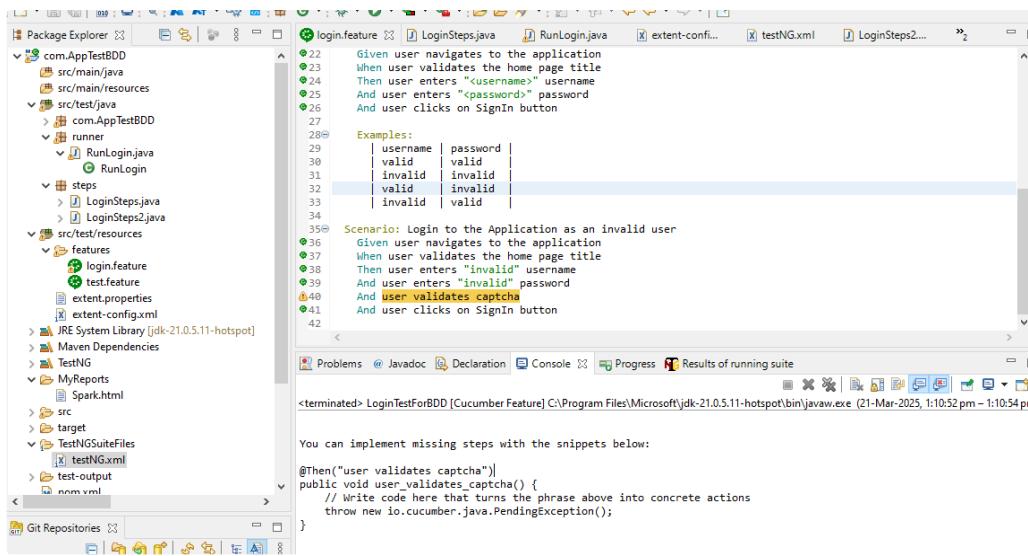
@Then("user enters {string} username")
public void user_enters_username(String username) {
    System.out.println("@Then---- user enters "+username+"username");
}

@Then("user enters {string} password")
public void user_enters_password(String password) {
    System.out.println("@Then---- user enters"+password+"password");
}

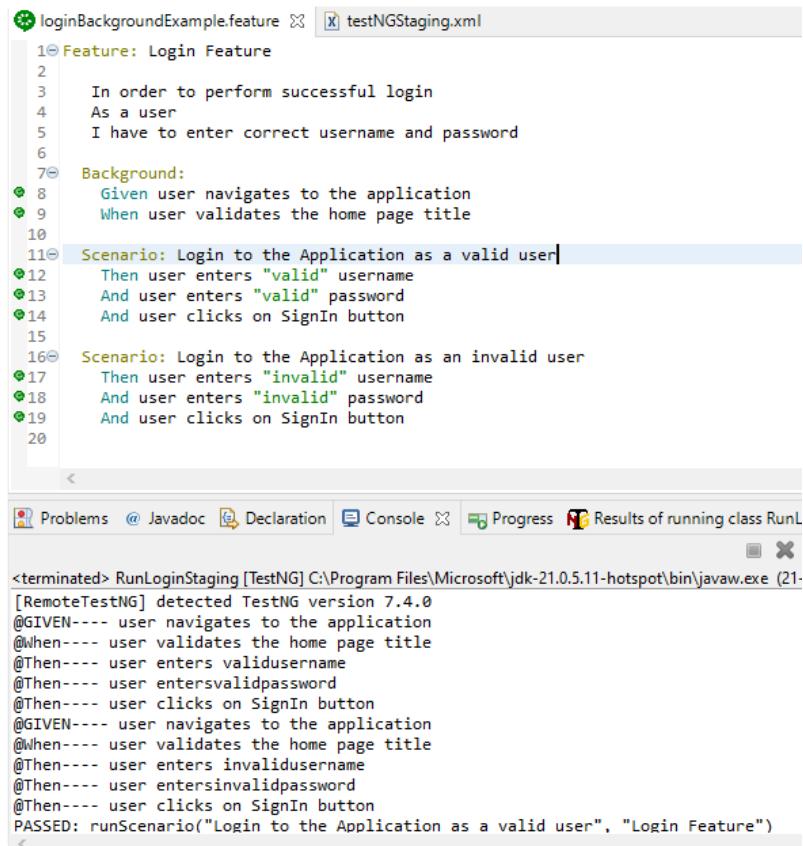
```

Adding multiple features and Step Definitions

Under the steps package we can add multiple java classes for steps definitions and all the steps will be visible throughout the features.



Background



Cucumber Tags

Used when we want to run some set of selected scenarios smoke,sanity, modulewise
To execute limited features, we can tag our features and scenarios.

Tags="@production or @staging or @sanity" run both

Tags="@production or @staging and not @sanity"

```
1 @regression
2 Feature: Login Feature
3
4   In order to perform successful login
5     As a user
6     I have to enter correct username and password
7
8 @ Scenario: Login to the Application as a valid user
9   Given user navigates to the application
10  When user validates the home page title
11  Then user enters "valid" username
12  And user enters "valid" password
13  And user clicks on SignIn button
14
15 @regression
16 @ Scenario: Login to the Application as an invalid user
17   Given user navigates to the application
18   When user validates the home page title
19   Then user enters "invalid" username
20   And user enters "invalid" password
21   And user clicks on SignIn button
22
23 @smoke
24 @ Scenario Outline: Login to the Application with different combinations
25   Given user navigates to the application
26   When user validates the home page title
27   Then user enters "<username>" username
28   And user enters "<password>" password
29   And user clicks on SignIn button
30
31 Examples:
32   | username | password |
33   | valid    | valid    |
34   | invalid  | invalid  |
35   | '        | '       |
```

To make this work under @cucumberOptions add tags="regression"

If you don't mention any tags then we can run all the features in features folder in case if we don't want to run any specific feature file.

```
1 package runner;
2
3 @import com.aventstack.extentreports.ExtentReports;
4
5 @CucumberOptions(features="src/test/resources/features",
6   glue= {"steps"},
7   tags=@staging,
8   plugin= {"html:target/cucumber-reports/cucumber-html-report.html",
9             "com.aventstack.extentreports.cucumber.adapter.ExtentCucumberAdapter:"})
10
11 public class RunLogin extends AbstractTestNGCucumberTests {
```

The screenshot shows an IDE interface with several tabs at the top: loginBackgroundExample.feature, login.feature, RunLogin.java, and testNG.xml. The main area displays a Cucumber feature file with the following content:

```
1 @staging
2 Feature: Login Feature staging
3
4   In order to perform successful login
5   As a user
6   I have to enter correct username and password
7
8   Background:
9     Given user navigates to the application
10    When user validates the home page title
11
12   @staging
13   Scenario: Login to the Application as a valid user on staging
14     Then user enters "valid" username
15     And user enters "valid" password
16     And user clicks on SignIn button
17
18   @staging
19   Scenario: Login to the Application as an invalid user on staging
20     Then user enters "invalid" username
21     And user enters "invalid" password
22     And user clicks on SignIn button
23
```

Below the code editor, there is a toolbar with icons for Problems, Javadoc, Declaration, Console, Progress, and Results of running class RunLogin. The Results tab is selected, showing the output of the run:

```
<terminated> RunLogin [TestNG] C:\Program Files\Microsoft\jdk-21.0.5.11-hotspot\bin\javaw.exe (21-Mar-2025, 2:45:53 pm - 2:45:57 pm)
@Then---- user enters invalid password
@Then---- user clicks on SignIn button
PASSED: runScenario("Login to the Application as an invalid user on staging", "Login Feature staging")
  Runs Cucumber Scenarios
PASSED: runScenario("Login to the Application as a valid user on staging", "Login Feature staging")
  Runs Cucumber Scenarios
=====
Default test
```

The screenshot shows an IDE interface with several tabs at the top: loginBackgroundExample.feature, login.feature, RunLogin.java, testNG.xml, and RunLoginStaging.java. The main area displays a Java code file with the following content:

```
1 package runner;
2
3 import com.aventstack.extentreports.ExtentReports;
4
5
6 @CucumberOptions(features="src/test/resources/features",
7   glue= {"steps"},
8   tags="@production and not @staging",
9   plugin= {"html:target/cucumber-reports/cucumber-html-report.html",
10     "com.aventstack.extentreports.cucumber.adapter.ExtentCucumberAdapter:"}
11 )
12 public class RunLoginStaging extends AbstractTestNGCucumberTests {
13
14
15 }
```

```

1 @production
2 Feature: Login Feature production
3
4 In order to perform successful login |
5 As a user
6 I have to enter correct username and password
7
8 Scenario: Login to the Application as a valid user on production
9 Given user navigates to the application
10 When user validates the home page title
11 Then user enters "valid" username
12 And user enters "valid" password
13 And user clicks on SignIn button
14
15 Scenario: Login to the Application as an invalid user on production
16 Given user navigates to the application
17 When user validates the home page title
18 Then user enters "invalid" username
19 And user enters "invalid" password
20 And user clicks on SignIn button
21
22 Scenario Outline: Login to the Application with different combinations on production
23 Given user navigates to the application
24 When user validates the home page title
25 Then user enters "<username>" username
26 And user enters "<password>" password
27 And user clicks on SignIn button
28
29 Examples:
30 | username | password |
31 | valid | valid |
32 | invalid | invalid |
33 | valid | invalid |
34 | invalid | valid |

```

Cucumber Hooks

- i** `@BeforeAll & @AfterAll` -static method - works before and after all the scenarios in a feature file
- `@Before & @After` - works before and after each scenario in a feature file
- `@BeforeStep and @AfterStep` – works before and after each steps

Hooks is similar to `@Before` and `@After` annotation in TestNG and junit

- i** Hooks in Cucumber BDD are special methods that run before or after each scenario. They are useful for setting up and tearing down the test environment. In Java, you can use annotations to define these hooks.

Types of Hooks

1. **Before Hooks:** These run before each scenario.
2. **After Hooks:** These run after each scenario.
3. **BeforeStep Hooks:** These run before each step.
4. **AfterStep Hooks:** These run after each step.

Usage of Hooks with Examples

Before and After Hooks

You can use `@Before` and `@After` annotations to define methods that should run before and after each scenario.

BeforeStep and AfterStep Hooks

Similarly, you can use `@BeforeStep` and `@AfterStep` annotations to define methods that should run before and after each step.

Explanation

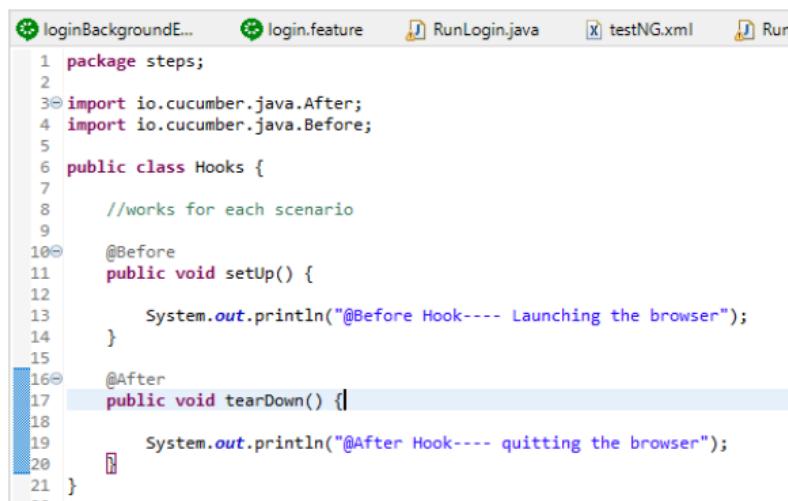
- **Before Hooks:** Useful for initializing WebDriver, opening a browser, or setting up test data.
- **After Hooks:** Useful for closing the browser, cleaning up test data, or logging results.
- **BeforeStep Hooks:** Useful for actions that need to be performed before each step, like logging or setting up preconditions.
- **AfterStep Hooks:** Useful for actions that need to be performed after each step, like taking screenshots or logging results.

The `@BeforeAll` annotation is used to define a method that runs once before all scenarios. This is useful for initializing resources that are needed for all tests, such as database connections or starting a server.

AfterAll Annotation

The `@AfterAll` annotation is used to define a method that runs once after all scenarios. This is useful for cleaning up resources that were used during the tests, such as closing database connections or stopping a server.

Create hooks class in your steps package



```
1 package steps;
2
3 import io.cucumber.java.After;
4 import io.cucumber.java.Before;
5
6 public class Hooks {
7
8     //works for each scenario
9
10    @Before
11    public void setUp() {
12
13        System.out.println("@Before Hook---- Launching the browser");
14    }
15
16    @After
17    public void tearDown() {
18
19        System.out.println("@After Hook---- quitting the browser");
20    }
21}
```

Note- `@Before` and `@After` – work before and after each scenario in a feature file.

```
[RemoteTestNG] detected TestNG version 7.4.0
@Before Hook---- Launching the browser
@GIVEN---- user navigates to the application
@When---- user validates the home page title
@Then---- user enters validusername
@Then---- user entersvalidpassword
@Then---- user clicks on SignIn button
@After Hook---- quitting the browser
@Before Hook---- Launching the browser
@GIVEN---- user navigates to the application
@When---- user validates the home page title
@Then---- user enters invalidusername
@Then---- user entersinvalidpassword
@Then---- user clicks on SignIn button
@After Hook---- quitting the browser
PASSED: runScenario("Login to the Application as a valid user on staging", "Login Feature staging")
    Runs Cucumber Scenarios
PASSED: runScenario("Login to the Application as an invalid user on staging", "Login Feature staging")
    Runs Cucumber Scenarios

=====
Default test
Tests run: 1, Failures: 0, Skips: 0
=====
```

Note - Create hooks class in your steps package, if you are creating a separate package for Hooks then it should be defined in the runner class similar to how we define steps.

```
1 package com.anhtester.runners;
2
3 import io.cucumber.testng.AbstractTestNGCucumberTests;
4 import io.cucumber.testng.CucumberOptions;
5 import org.testng.annotations.DataProvider;
6 import org.testng.annotations.Test;
7
8 @CucumberOptions(
9     features = "src/test/resources/features/CMS/Category/CategoryCMS.feature",
10    glue = {"com.anhtester.stepdefinitions",
11            "com.anhtester.common",
12            "com.anhtester.hooks"
13        },
14    plugin = {
15        "com.anhtester.hooks.CucumberListener",
16        "pretty",
17        "html:target/cucumber-reports/TestRunnerCategoryCMS.html",
18        "json:target/cucumber-reports/TestRunnerCategoryCMS.json",
19        "io.qameta.allure.cucumber7jvm.AllureCucumber7Jvm"
20    },
21    tags = "@AddCategory"
22)
23 @Test
24 public class TestRunnerCategoryCMS extends AbstractTestNGCucumberTests {
25     //Parallel Execution Scenario
26     @Override
27     @DataProvider(parallel = true)
28     public Object[][] scenarios() {
29         return super.scenarios();
30     }
31 }
```

Make the BeforeAll and AfterAll method as static.

```

1 package steps;
2 import io.cucumber.java.After;
3 import io.cucumber.java.AfterAll;
4 import io.cucumber.java.Before;
5 import io.cucumber.java.BeforeAll;
6
7 public class Hooks {
8
9     @BeforeAll
10    public void before_All() {
11
12        System.out.println("@BeforeAll --- Open DB connection ");
13    }
14
15    @AfterAll
16    public void after_All() {
17        System.out.println("@AfterAll --- close DB connection ");
18    }
19
20    @Before
21    public void setUp() {
22
23        System.out.println("@Before Hook--- Launching the browser");
24    }
25
26    @After
27    public void tearDown() {
28
29        System.out.println("@After Hook--- quitting the browser");
30    }
31
32 }
33

```

```

<terminated> RunLogin [TestNG] C:\Program Files\Microsoft\jdk-21.0.5.11-hotspot\bin\javaw.exe (21-Mar-2025, 3:38:29 pm - 3:38:35 pm)
[RemoteTestNG] detected TestNG version 7.4.0
FAILED CONFIGURATION: @BeforeClass setUpClass(org.testng.TestRunner@49bf29c6)
io.cucumber.java.InvalidMethodSignatureException: A method annotated with BeforeAll or AfterAll must have one of the:
* public static void before_or_after_all()
at steps.Hooks.before_All()

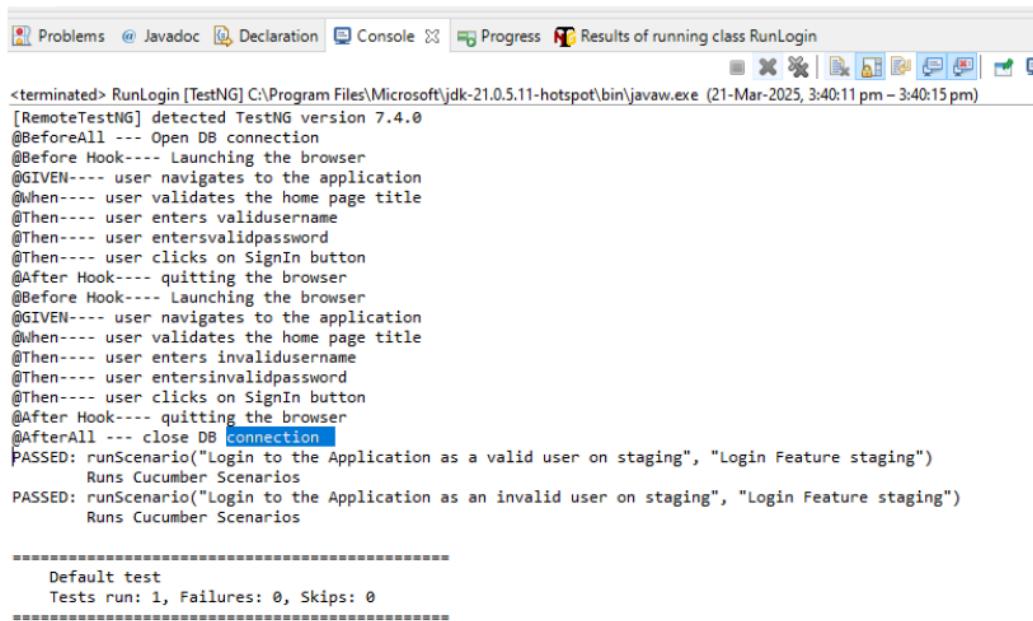
```

Make the BeforeAll and AfterAll method as static.

```

1 package steps;
2 import io.cucumber.java.After;
3 import io.cucumber.java.AfterAll;
4 import io.cucumber.java.Before;
5 import io.cucumber.java.BeforeAll;
6
7 public class Hooks {
8
9     @BeforeAll
10    public static void before_All() {
11
12        System.out.println("@BeforeAll --- Open DB connection ");
13    }
14
15    @AfterAll
16    public static void after_All() {
17        System.out.println("@AfterAll --- close DB connection ");
18    }
19
20 }
21

```



The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output window displays the execution results of a TestNG test named 'RunLogin'. The log shows various annotations like @BeforeAll, @Before, @Given, @When, @Then, @After, and @AfterAll being triggered at different points of the scenario. The results indicate two passed scenarios: one for a valid user and one for an invalid user, each running cucumber scenarios.

```

<terminated> RunLogin [TestNG] C:\Program Files\Microsoft\jdk-21.0.5.11-hotspot\bin\javaw.exe (21-Mar-2025, 3:40:11 pm - 3:40:15 pm)
[RemoteTestNG] detected TestNG version 7.4.0
@BeforeAll --- Open DB connection
@Before Hook---- Launching the browser
@Given---- user navigates to the application
@When---- user validates the home page title
@Then---- user enters validusername
@Then---- user entersvalidpassword
@Then---- user clicks on SignIn button
@After Hook--- quitting the browser
@Before Hook---- Launching the browser
@Given---- user navigates to the application
@When---- user validates the home page title
@Then---- user enters invalidusername
@Then---- user entersinvalidpassword
@Then---- user clicks on SignIn button
@After Hook--- quitting the browser
@AfterAll --- close DB connection
PASSED: runScenario("Login to the Application as a valid user on staging", "Login Feature staging")
    Runs Cucumber Scenarios
PASSED: runScenario("Login to the Application as an invalid user on staging", "Login Feature staging")
    Runs Cucumber Scenarios

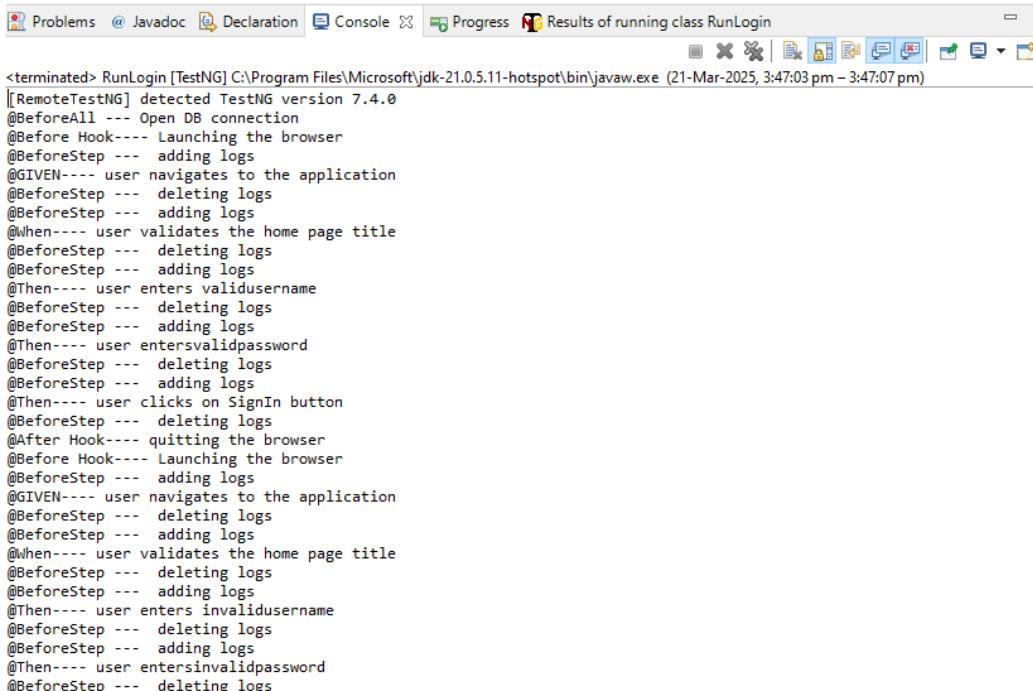
=====
Default test
Tests run: 1, Failures: 0, Skips: 0
=====
```

@BeforeStep and @AfterStep – works before and after each steps

```

@BeforeStep
public void before_step() {
    System.out.println("@BeforeStep --- adding logs");
}

@AfterStep
public void after_step() {
    System.out.println("@BeforeStep --- deleting logs");
}
```



The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output window displays the execution results of the 'RunLogin' test. The log includes numerous instances of the @BeforeStep and @AfterStep annotations, showing them being executed before and after each step definition in the feature file. This demonstrates how these annotations can be used to perform setup and cleanup operations for each individual step.

```

<terminated> RunLogin [TestNG] C:\Program Files\Microsoft\jdk-21.0.5.11-hotspot\bin\javaw.exe (21-Mar-2025, 3:47:03 pm - 3:47:07 pm)
[RemoteTestNG] detected TestNG version 7.4.0
@BeforeAll --- Open DB connection
@Before Hook--- Launching the browser
@BeforeStep --- adding logs
@Given---- user navigates to the application
@BeforeStep --- deleting logs
@BeforeStep --- adding logs
@When---- user validates the home page title
@BeforeStep --- deleting logs
@BeforeStep --- adding logs
@Then---- user enters validusername
@BeforeStep --- deleting logs
@BeforeStep --- adding logs
@Given---- user navigates to the application
@BeforeStep --- deleting logs
@BeforeStep --- adding logs
@When---- user validates the home page title
@BeforeStep --- deleting logs
@BeforeStep --- adding logs
@Then---- user enters invalidusername
@BeforeStep --- deleting logs
@BeforeStep --- adding logs
@Then---- user entersinvalidpassword
@BeforeStep --- deleting logs
```

Defining the execution order in Cucumber Hooks

We can have multiple @Before hook and we can define the execution order

User order=0 , 0 being max priority for Before in case of after it will be reverse.

```
10 public class Hooks {
11
12     @BeforeAll
13     public static void before_All() {
14
15         System.out.println("@BeforeAll --- Open DB connection ");
16
17     }
18
19     @AfterAll
20     public static void after_All() {
21
22         System.out.println("@AfterAll --- close DB connection ");
23
24     }
25     @Before(order = 0)
26     public void setUp1() {
27
28         System.out.println("@Before Hook 1---- Launching the browser");
29
30     }
31     @Before(order = 1)
32     public void setUp2() {
33
34         System.out.println("@Before Hook 2---- Launching the browser");
35
36     }
37     @After(order = 1)
38     public void tearDown1() {
39
40         System.out.println("@After Hook 1---- quitting the browser");
41
42     }
43     @After(order = 0)
44     public void tearDown2() {
45
46 }
```

Tagged Hooks

Before or after condition for a particular tag, make sure runner class is also updated.

```
1 @Before("@staging")
2     public void setUp() {
3
4         System.out.println("@Before Hook 1---- Launching the browser");
5
6 }
```

```
1 package runner;
2
3     import com.aventstack.extentreports.ExtentReports;
4
5
6     @CucumberOptions(features="src/test/resources/features",
7                     glue= {"steps"},
8                     tags="@production and not @staging",
9                     plugin= {"html:target/cucumber-reports/cucumber-html-report.html",
10                            "com.aventstack.extentreports.cucumber.adapter.ExtentCucumberAdapter:"}
11
12     )
13     public class RunLoginStaging extends AbstractTestNGCucumberTests {
14
15
16 }
```

Data Driven approach in feature files

Example of Passing Data Directly from Scenario Step to Glue Code

```
1 Feature: User Registration
2
3 Scenario: Register a new user
4   Given the user is on the registration page
5   When the user registers with username "testUser" and password "testPass"
6   Then the user should see a confirmation message
```

In this example, the scenario step `When the user registers with username "testUser" and password "testPass"` contains the data that we want to pass to the glue code.

```
1 import io.cucumber.java.en.Given;
2 import io.cucumber.java.en.When;
3 import io.cucumber.java.en.Then;
4 import org.openqa.selenium.By;
5 import org.openqa.selenium.WebDriver;
6 import org.openqa.selenium.chrome.ChromeDriver;
7
8 public class RegistrationSteps {
9
10    WebDriver driver;
11
12    @Given("the user is on the registration page")
13    public void the_user_is_on_the_registration_page() {
14        // Initialize the WebDriver and navigate to the registration page
15        System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");
16        driver = new ChromeDriver();
17        driver.get("http://example.com/registration");
18    }
19
20    @When("the user registers with username {string} and password {string}")
21    public void the_user_registers_with_username_and_password(String username, String password) {
22        // Code to enter the username and password
23        driver.findElement(By.id("username")).sendKeys(username);
24        driver.findElement(By.id("password")).sendKeys(password);
25        driver.findElement(By.id("registerButton")).click();
26    }
27
28    @Then("the user should see a confirmation message")
29    public void the_user_should_see_a_confirmation_message() {
30        // Code to verify that the confirmation message is displayed
31        String confirmationMessage = driver.findElement(By.id("confirmationMessage")).getText();
32        assert confirmationMessage.equals("Registration successful!");
33        driver.quit(); // Close the browser
34    }
35 }
```

Passing multiple data sets from Scenario Outline and Examples to glue code

It allows you to run the same test scenario multiple times with different sets of data.

This is a Cucumber feature that allows you to define a scenario template that can be executed multiple times with different sets of data.

You define the placeholders in the scenario and provide the data in a separate Examples table.

```
1 Feature: User Login
2
3 Scenario Outline: Successful login with valid credentials
4   Given the user is on the login page
5   When the user enters "<username>" and "<password>"
6   Then the user should be redirected to the dashboard
7
8 Examples:
9   | username | password |
10  | user1    | pass1    |
11  | user2    | pass2    |
12  | user3    | pass3    |
```

In this example, the scenario outline allows you to test the login functionality with different usernames and passwords.

```
1 import io.cucumber.java.en.Given;
2 import io.cucumber.java.en.When;
3 import io.cucumber.java.en.Then;
4
5 public class LoginSteps {
6
7     @Given("the user is on the login page")
8     public void the_user_is_on_the_login_page() {
9         // Code to navigate to the login page
10    }
11
12     @When("the user enters {string} and {string}")
13     public void the_user_enters_and(String username, String password) {
14         // Code to enter username and password
15         // For example:
16         driver.findElement(By.id("username")).sendKeys(username);
17         driver.findElement(By.id("password")).sendKeys(password);
18         driver.findElement(By.id("loginButton")).click();
19    }
20
21     @Then("the user should be redirected to the dashboard")
22     public void the_user_should_be_redirected_to_the_dashboard() {
23         // Code to verify that the user is on the dashboard
24    }
25 }
```

Data Table as List

Create a feature file "dataTableAsList.feature"

```
1 @datatable
2 Feature: Login Feature production
3
4 In order to perform successful login
5 As a user
6 I have to enter correct username and password
7
8 @datatable
9 Scenario Outline: Login to the Application with different combinations on production
10 Given user navigates to the application
11 When user validates the home page title
12 Then user enters "valid" username
13 And user enters "valid" password
14 And user validates captcha
15 And user enters firstname and lastname
16 | Rahul | Ritesh |
17 | john | cena |
18 And user clicks on SignIn button
19
20
21 @Then("user enters firstname and lastname")
22 public void user_enters_firstname_and_lastname(io.cucumber.datatable.DataTable dataTable) {
23     // Write code here that turns the phrase above into concrete actions
24     // For automatic transformation, change DataTable to one of
25     // E, List<E>, List<List<E>>, List<Map<K,V>>, Map<K,V>
26     // Map<K, List<V>>. E,K,V must be a String, Integer, Float,
27     // Double, Byte, Short, Long, BigInteger or BigDecimal.
28     //
29     // For other transformations you can register a DataTableType.
30     // data in table is stored in rows and Columns
31     List<List<String>> data = dataTable.asLists();
32     String firstName = data.get(0).get(0);
33     System.out.println("First Name is: " + firstName);
34
35     String lastName = data.get(0).get(1);
36     System.out.println("Last Name is: " + lastName);
37 }
```

It is a data table so we can read all the data as row column format.

i `DataTable` is a powerful feature that allows you to pass structured data to your step definitions. You can convert `DataTable` to a list of objects or a list of lists, depending on your needs.

Example 1: Converting DataTable to a List of Lists ↗

Suppose you have a feature file with a scenario that includes a table:

```
1
2 Feature: User Login
3
4 Scenario: Successful login
5     Given the following users exist:
6     | username | password |
7     | user1    | pass1    |
8     | user2    | pass2    |
9
```

You can convert this `DataTable` to a list of lists in your step definition:

```
1 import io.cucumber.datatable.DataTable;
2 import io.cucumber.java.en.Given;
3
4 import java.util.List;
5
6 public class LoginSteps {
```

```

7
8     @Given("the following users exist:")
9     public void the_following_users_exist(DataTable dataTable) {
10         List<List<String>> users = dataTable.asLists(String.class);
11         for (List<String> user : users) {
12             System.out.println("Username: " + user.get(0) + ", Password: " + user.get(1));
13         }
14     }
15 }
16

```

Example 2: Converting DataTable to a List of Objects ↗

If you have a more complex scenario, you might want to convert `DataTable` to a list of custom objects. First, define a class to represent the data:

```

1 public class User {
2     private String username;
3     private String password;
4
5     // Constructors, getters, and setters
6     public User(String username, String password) {
7         this.username = username;
8         this.password = password;
9     }
10
11    public String getUsername() {
12        return username;
13    }
14
15    public void setUsername(String username) {
16        this.username = username;
17    }
18
19    public String getPassword() {
20        return password;
21    }
22
23    public void setPassword(String password) {
24        this.password = password;
25    }
26 }
27

```

Then, use `DataTable` to convert the table to a list of `User` objects:

```

1 import io.cucumber.datatable.DataTable;
2 import io.cucumber.java.en.Given;
3
4 import java.util.List;
5
6 public class LoginSteps {
7
8     @Given("the following users exist:")
9     public void the_following_users_exist(DataTable dataTable) {
10         List<User> users = dataTable.asList(User.class);
11         for (User user : users) {

```

```

12         System.out.println("Username: " + user.getUsername() + ", Password: " + user.getPassword());
13     }
14 }
15 }
16

```

Explanation

- DataTable as Lists:** This method is useful for simple tables where each row represents a list of values. You can access each value using its index.
- DataTable as List of Objects:** This method is more powerful and flexible, allowing you to map each row to a custom object. This is particularly useful for scenarios with more complex data structures.

These examples should help you understand how to use `DataTable` in Cucumber BDD with Selenium and Java. If you have any specific questions or need further clarification, feel free to ask!

Data Table As Map

```

1 @datatableAsMap
2 Feature: Login Feature production
3
4 In order to perform successful login
5 As a user
6 I have to enter correct username and password
7
8 Scenario Outline: Login to the Application with different combinations on production
9   Given user navigates to the application
10  When user validates the home page title
11  Then user enters "valid" username
12  And user enters "valid" password
13  And user validates captcha
14  And user enters firstname and lastname as map
15    | firstName | lastName |
16    | Rahul    | Ritesh   |
17    | john     | cena      |
18  And user clicks on SignIn button
19

@Then("user enters firstname and lastname as map")
public void user_enters_firstname_and_lastname_as_map(io.cucumber.datatable.DataTable dataTable) {
    List<Map<String, String>> map = dataTable.asMaps();
    String firstName = map.get(0).get("firstName");
    String lastName = map.get(0).get("lastName");

    System.out.println("First Name is: " + firstName + "Last Name is: " + lastName);
    System.out.println("First Name for 2 nd row: " + map.get(1).get("firstName")
        + "Last Name for 2 nd row: " + map.get(1).get("lastName"));

}

```

Parameterize data table as a Map

```
@Then("user enters firstname and lastname as map")
public void user_enters_firstname_and_lastname_as_map(io.cucumber.datatable.DataTable dataTable) {
/*
 * List<Map<String, String>> map = dataTable.asMaps(); String firstName =
 * map.get(0).get("firstName"); String lastName = map.get(0).get("lastName");
 *
 * System.out.println("First Name is: "+ firstName + "Last Name is: "+lastName);
 * System.out.println("First Name for 2 nd row: "+ map.get(1).get("firstName") +
 * "Last Name for 2 nd row: "+ map.get(1).get("lastName"));
 */
for(Map<String, String> data : dataTable.asMaps(String.class, String.class)) {
    System.out.println("FirstName is----"+data.get("firstName")
    + " LastName is----"+data.get("lastName"));
}
}
```

```
package steps;
|
import io.cucumber.java.Before;
import utils.SeleniumDriver;

public class BeforeActions {

    @Before
    public static void setUp() {
        //ExtentCucumberFormatter.initiateExtentCucumberFormatter();
        System.out.println("Before");
        SeleniumDriver.setUpDriver();
    }
}
```

Taking Screenshot on failure

```
package steps;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;

import io.cucumber.java.Scenario;
import io.cucumber.java.After;
import utils.SeleniumDriver;

public class AfterActions {

    @After
    public static void tearDown( @NotNull Scenario scenario) {

        WebDriver driver=SeleniumDriver.getDriver();
        System.out.println(scenario.isFailed());
        if (scenario.isFailed()) {
            byte[] screenshotBytes = ((TakesScreenshot) driver).getScreenshotAs( target: OutputType.BYTES);
            scenario.attach( data: screenshotBytes, mediaType: "image/png", name: "image");
        }
        SeleniumDriver.tearDown();
    }
}
```