Selenium Cucumber Java BDD

Project creation Steps:

1. Create Maven project.
2. Add dependencies in pom.xml: Add maven dependencies.

Cucumber: <https://mvnrepository.com/artifact/io.cucumber/cucumber-java/7.18.0>

1. Create a folder Feature under src/test/resources.
2. Under features folder create a new feature file login.feature.
3. Download Cucumber plugin from Eclipse Marketplace.
4. Create feature file and add contents.

Feature, Scenario, Steps, Scenario Outline, Example, Tags, Comments.

1. Try to run the feature file
2. Add Step Definitions/Guide Code under src/test/java package.
3. Regular expression: <https://regexone.com/>

|  |  |  |
| --- | --- | --- |
|  | [abc…](https://regexone.com/lesson/introduction_abcs) | [*Letters*](https://regexone.com/lesson/introduction_abcs) |
|  | [123…](https://regexone.com/lesson/letters_and_digits) | [*Digits*](https://regexone.com/lesson/letters_and_digits) |
|  | [\d](https://regexone.com/lesson/letters_and_digits) | [*Any Digit*](https://regexone.com/lesson/letters_and_digits) |
|  | [\D](https://regexone.com/lesson/letters_and_digits) | [*Any Non-digit character*](https://regexone.com/lesson/letters_and_digits) |
|  | [.](https://regexone.com/lesson/wildcards_dot) | [*Any Character*](https://regexone.com/lesson/wildcards_dot) |
|  | [\.](https://regexone.com/lesson/wildcards_dot) | [*Period*](https://regexone.com/lesson/wildcards_dot) |
|  | [[abc]](https://regexone.com/lesson/matching_characters) | [*Only a, b, or c*](https://regexone.com/lesson/matching_characters) |
|  | [[^abc]](https://regexone.com/lesson/excluding_characters) | [*Not a, b, nor c*](https://regexone.com/lesson/excluding_characters) |
|  | [[a-z]](https://regexone.com/lesson/character_ranges) | [*Characters a to z*](https://regexone.com/lesson/character_ranges) |
|  | [[0-9]](https://regexone.com/lesson/character_ranges) | [*Numbers 0 to 9*](https://regexone.com/lesson/character_ranges) |
|  | [\w](https://regexone.com/lesson/character_ranges) | [*Any Alphanumeric character*](https://regexone.com/lesson/character_ranges) |
|  | [\W](https://regexone.com/lesson/character_ranges) | [*Any Non-alphanumeric character*](https://regexone.com/lesson/character_ranges) |
|  | [{m}](https://regexone.com/lesson/repeating_characters) | [*m Repetitions*](https://regexone.com/lesson/repeating_characters) |
|  | [{m,n}](https://regexone.com/lesson/repeating_characters) | [*m to n Repetitions*](https://regexone.com/lesson/repeating_characters) |
|  | [\*](https://regexone.com/lesson/kleene_operators) | [*Zero or more repetitions*](https://regexone.com/lesson/kleene_operators) |
|  | [+](https://regexone.com/lesson/kleene_operators) | [*One or more repetitions*](https://regexone.com/lesson/kleene_operators) |
|  | [?](https://regexone.com/lesson/optional_characters) | [*Optional character*](https://regexone.com/lesson/optional_characters) |
|  | [\s](https://regexone.com/lesson/whitespaces) | [*Any Whitespace*](https://regexone.com/lesson/whitespaces) |
|  | [\S](https://regexone.com/lesson/whitespaces) | [*Any Non-whitespace character*](https://regexone.com/lesson/whitespaces) |
|  | [^…$](https://regexone.com/lesson/line_beginning_end) | [*Starts and ends*](https://regexone.com/lesson/line_beginning_end) |
|  | [(…)](https://regexone.com/lesson/capturing_groups) | [*Capture Group*](https://regexone.com/lesson/capturing_groups) |
|  | [(a(bc))](https://regexone.com/lesson/nested_groups) | [*Capture Sub-group*](https://regexone.com/lesson/nested_groups) |
|  | [(.\*)](https://regexone.com/lesson/more_groups) | [*Capture all*](https://regexone.com/lesson/more_groups) |

1. Create a Runner class.
2. Add Junit dependencies: <https://mvnrepository.com/artifact/junit/junit/4.13.2>
3. Add Cucumber Junit Dependencies: <https://mvnrepository.com/artifact/io.cucumber/cucumber-junit/7.15.0>
4. Create Add Cucumber Options for generating reports: HTML|JSON|JUNIT|XML.
5. Run and verify results.

Selenium Test:

1. Add selenium Java maven dependencies: <https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java/4.17.0>
2. Create feature file and add scenario and steps
3. Create Step Definition/Glue Code for the steps