BDD (Behavior Driver Development)

Pre-defined frameworks:

* Junit (TDD)
* TestNg (TDD)
* Cucumber (BDD)

User defined frameworks

* Data driven framework
* Keyword driven framework
* Hybrid framework

Tools available for BDD:

* + Cucumber
  + Jbehave
  + Nbehave
  + SpecFlow

Gherkin is plain text

Syntax of Gherkin:

1. Feature
2. Background
3. Scenario
4. Given
5. When
6. Then
7. And
8. But
9. Scenario outline
10. Exmaples
11. Scenario Templates
12. Installation of new version cucumber eclipse plugin. Visit below website:

<https://github.com/cucumber/cucumber-eclipse-update-site-snapshot>

<https://cucumber.io/>

1. Add cucumber dependency in pom.xml

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

<dependency>

<groupId>io.cucumber</groupId>

<artifactId>cucumber-java</artifactId>

<version>4.7.2</version>

</dependency>

<!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-testng -->

<dependency>

<groupId>io.cucumber</groupId>

<artifactId>cucumber-testng</artifactId>

<version>4.7.2</version>

</dependency>

<dependency>

<groupId>io.cucumber</groupId>

<artifactId>cucumber-jvm-deps</artifactId>

<version>1.0.6</version>

<scope>provided</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/net.masterthought/cucumber-reporting -->

<dependency>

<groupId>net.masterthought</groupId>

<artifactId>cucumber-reporting</artifactId>

<version>4.10.0</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.assertj/assertj-core -->

<dependency>

<groupId>org.assertj</groupId>

<artifactId>assertj-core</artifactId>

<version>3.13.2</version>

<scope>test</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/net.masterthought/maven-cucumber-reporting -->

<dependency>

<groupId>net.masterthought</groupId>

<artifactId>maven-cucumber-reporting</artifactId>

<version>4.10.0</version>

</dependency>

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>6.14.3</version>

<scope>test</scope>

</dependency>

Feature file:

It contains all the features of an application. To specify steps in feature file we need to use special language called **Gherkin**

Sample feature file:

#Author: your.email@your.domain.com

#Keywords Summary :

#Feature: List of scenarios.

#Scenario: Business rule through list of steps with arguments.

#Given: Some precondition step

#When: Some key actions

#Then: To observe outcomes or validation

#And,But: To enumerate more Given,When,Then steps

#Scenario Outline: List of steps for data-driven as an Examples and <placeholder>

#Examples: Container for s table

#Background: List of steps run before each of the scenarios

#""" (Doc Strings)

#| (Data Tables)

#@ (Tags/Labels):To group Scenarios

#<> (placeholder)

#""

## (Comments)

#Sample Feature Definition Template

@tag

Feature: Title of your feature

I want to use this template for my feature file

@tag1

Scenario: Title of your scenario

Given I want to write a step with precondition

And some other precondition

When I complete action

And some other action

And yet another action

Then I validate the outcomes

And check more outcomes

@tag2

Scenario Outline: Title of your scenario outline

Given I want to write a step with <name>

When I check for the <value> in step

Then I verify the <status> in step

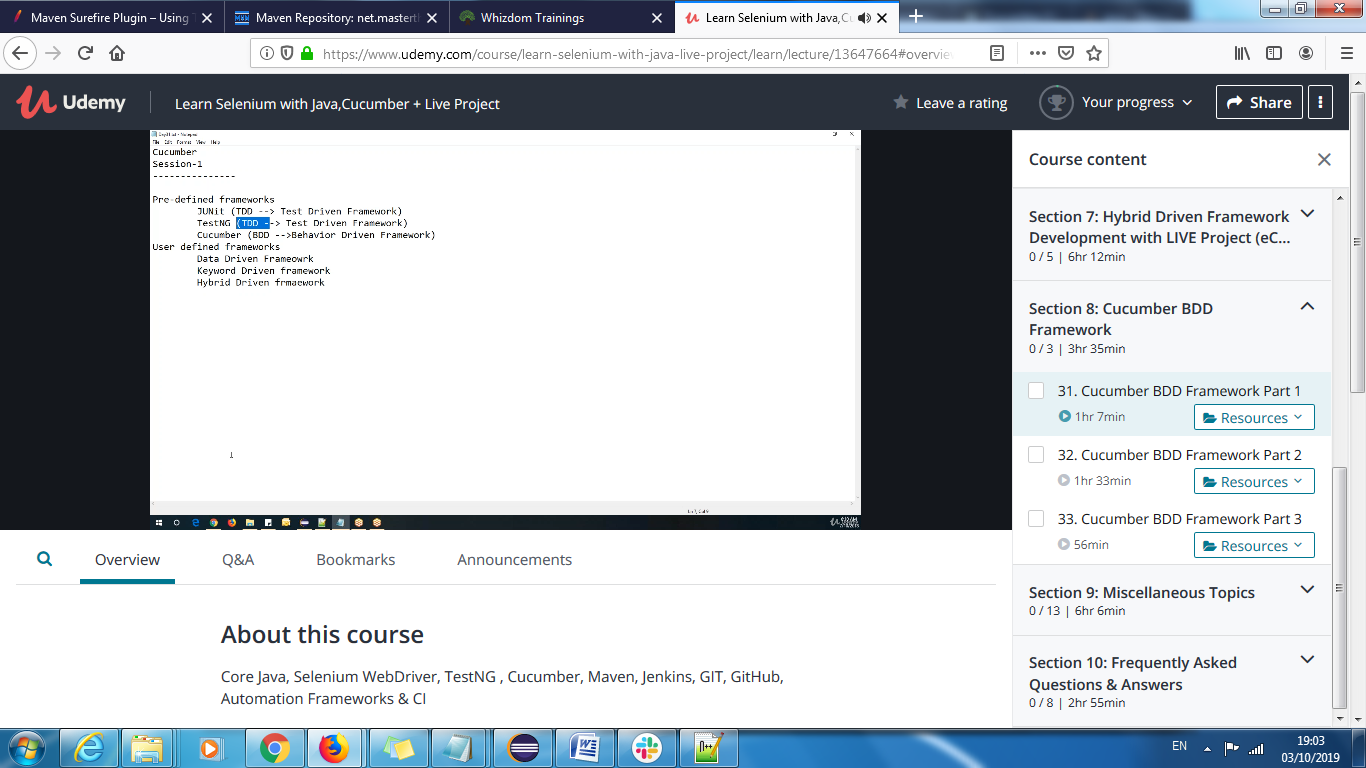
Examples:

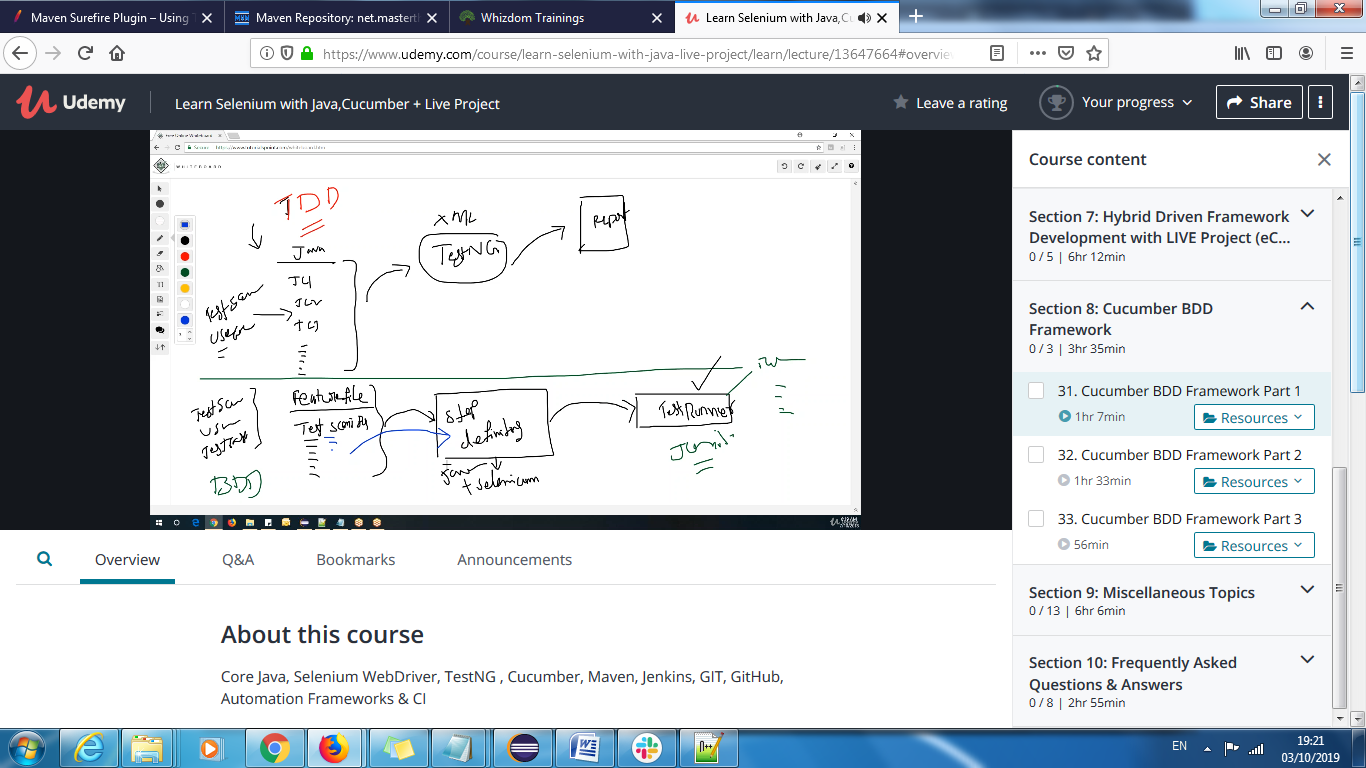
| name | value | status |

| name1 | 5 | success |

| name2 | 7 | Fail |

Udemy





Qtpselenium TestNg Dependencies

