**What is Cucumber?**

Cucumber is **software tool** that supports **Behaviour-Driven-Development**.

Cucumber is a software tool used for testing software. It **runs automated acceptance tests written in a behavior-driven development (BDD) style**.

**Cucumber reads executable specifications written in plain text** and **validate that the software does what those specifications say.**

The **specifications consist of multiple example or scenarios**.

**Each scenario is a list of steps for cucumber to work through**. **Cucumber verifies** that the software conforms with the **specification** and **generate a report** indicating **✅ success or ❌ failure for each scenario**.

In order for **cucumber to understand the scenarios**, they **must follow some basic syntax rules called Gherkin**.

**Gherkin** documents are stored in **.feature** text files.

**@smoke @regression**

**Feature:** Login functionality

**Background:**

**Scenario Outline:** Valid username and password

**Given** As a end user I am on login page

**When** user enter <username> and “<password>”

**Then** user should land on homepage

Examples

| username | password !

! [ballachakri@yahoo.com](mailto:ballachakri@yahoo.com) ! Khalifa12 |

**Step Definitions**

In **addition to feature files**, **Cucumber needs a set of Step definitions**. **Step definitions map each Gherkins step into runnable programming code to carry out what action should be performed by the step.**

**Step definitions can be written in many programming languages.**

[ Cucumber was originally written in the Ruby programming languagemnhhyrf6u95310 ].

--------------------------------------------------------------------------------------------------------------------------------------

To create Page Object Model Framework: using Cucumber BDD format

* New Maven project with QuickStart.
* Add below three main dependencies.

*<!-- info.cukes cucumber -java dependency -->* <**dependency**>  
 <**groupId**>info.cukes</**groupId**>  
 <**artifactId**>cucumber-java</**artifactId**>  
 <**version**>1.2.5</**version**>  
 </**dependency**>  
*<!-- info.cukes cucumber -junit dependency -->* <**dependency**>  
 <**groupId**>info.cukes</**groupId**>  
 <**artifactId**>cucumber-junit</**artifactId**>  
 <**version**>1.2.5</**version**>  
 </**dependency**>  
*<!-- seleniumhq selenium java -->* <**dependency**>  
 <**groupId**>org.seleniumhq.selenium</**groupId**>  
 <**artifactId**>selenium-java</**artifactId**>  
 <**version**>3.14.0</**version**>  
 </**dependency**>

* Create a resources Package in src/test/java folder for feature files
* Create another Package src/test/java folder for stepdefinitions

To run test we need to create CucumberTestRunner class

**@RunWith** (Cucumber.**class**)  
**@CucumberOptions**(features = **"src/test"**, *//// set: the path of the feature files -- { }* tags={**"@sort"**}, */// to instruct what tags in te feature files should be executed -- { }* dryRun = **false**, */// true: checks if all the steps have step definition -- default false* strict = **true**, */// true : will fail execution if there are undefined or pending steps -- default false* plugin = **"json: cath.json"**,  
 monochrome = **true**,format = {**"html:src/test/java/cucReports"** ,**"json:src/test/java/cucReports/cath1.json"**,**"pretty"**})  
 *//format = {"pretty"})  
 /// glue -- set: the path of the step definition files --- {}  
 /// monochrome -- true : display the console output in much readable way -- default false  
 /// format -- set: what all report formaters to use -- default false*

**public class** CucumberTestRunner {  
  
}