

# Cucumber Basics Key Points



**Scripting Logic**

# Topics to be covered

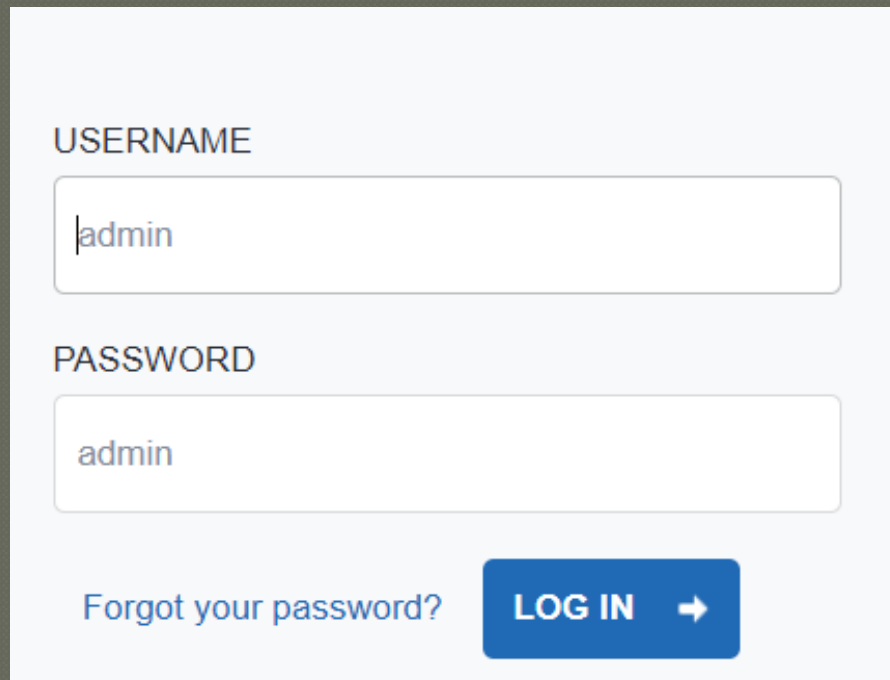
---

- ◉ Gherkin syntax
- ◉ Setting up cucumber
- ◉ Feature file
- ◉ Step definitions
- ◉ TestRunner
- ◉ Parameterization :
  - Reuse function with different data
  - Data Tables
  - Parameterization using example Keyword
- ◉ Tags
- ◉ Background & Hooks
- ◉ Cucumber Reports



# Gherkin syntax

- Write a test case of login page for testing the functionality of login button



USERNAME

PASSWORD

[Forgot your password?](#)

**LOG IN** →



# Gerkin syntax

× Below is the test case

Objective	To test the functionality of login button for valid user name and password
Precondition	Login page should be opened
Test steps	1.enter username 2.enter password 3.click on login button
Test input	username= admin Pasword=admin123
Expected result	we should land on the home page

Given



When



Then



# Gerkin syntax example

---

**Feature:** Login to the billing page

**Scenario:** Successful login for correct username and password

*Given* I am on login page of billing

*When* I enter valid username and password

*And* click on login button

*Then* I should be redirected to home page



# Setting up cucumber project

---

- ◉ Install Natural plug-in in the eclipse
- ◉ Create a maven project in eclipse using quick-start template
- ◉ Add maven code / dependencies  
cucumber-java 5.3 , cucumber-junit 5.3 & junit
- ◉ Optionally selenium & testng
- ◉ All set



# Feature file

- Create a feature file with .feature extension in eclipse
- Write the scenarios in gherkin language

```
Feature: Login to the billing page
```

```
Scenario: Successful login for correct username and password
```

```
    Given I am on login page of billing
```

```
    When I enter valid username and password
```

```
    And click on login button
```

```
    Then I should be redirected to home page
```



# Step definition

- Write step definition for all the steps you wrote in feature files

```
@Given("^I am on login page of billing$")
public void i_am_on_login_page_of_billing() { // throw new

    System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\

    driver= new ChromeDriver();
    driver.manage().window().maximize();
    driver.get("http://billing.scriptinglogic.net");

}
```





# Test runner

- Write test runner to run your test

```
import org.junit.runner.RunWith;
import io.cucumber.junit.Cucumber;
import io.cucumber.junit.CucumberOptions;

@RunWith(Cucumber.class)
@CucumberOptions
(features = "src/test/java/Feature",
glue="stepDefinition")
public class LoginTestRunner {

}
```

- Run the test, that's it !!!



# Ready made step definition - Option 1

```
import org.junit.runner.RunWith;
import io.cucumber.junit.Cucumber;
import io.cucumber.junit.CucumberOptions;

@RunWith(Cucumber.class)
@CucumberOptions
(features = "src/test/java/Feature",
glue="stepDefinition")
public class LoginTestRunner {

}
```

- Run the testRunner
- Get the skeleton of code in the output

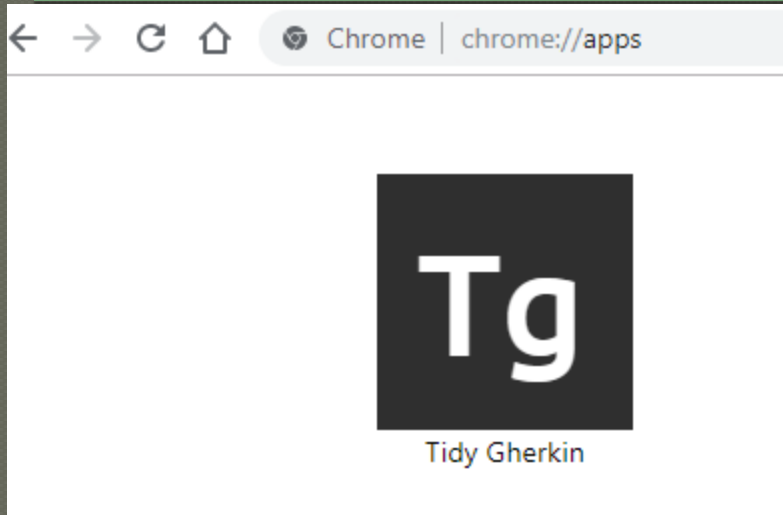
```
@Given("^I am on login page of billing$")
public void i_am_on_login_page_of_billing() { // throw new

    System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\\\

    driver= new ChromeDriver();
    driver.manage().window().maximize();
    driver.get("http://billing.scriptinglogic.net");

}
```

# Ready made step definition - Option 2



- Install Tidy Gherkin app in chrome
- Get the skeleton of code in the output



# Ready made step definition - Option 2

TIDY! FILE + EDIT + INSERT + SUPPORT + ABOUT + SETTINGS Pg Try Pretty Gherkin

rows columns INSERT TABLE

```
1 Feature: Login to the billing page
2
3
4
5 Scenario: Successful login for correct username and password
6   Given I am on login page of billing
7   When I enter valid username and password
8   And click on login button
9   Then I should be redirected to home page
10
```

Copy gherkin code here

TIDY PREVIEW • JAVA STEPS • RUBY STEPS • JS STEPS

```
1 package my.package.name
2
3 import cucumber.api.PendingException;
4 import cucumber.api.java.en.Given;
5 import cucumber.api.java.en.When;
6 import cucumber.api.java.en.Then;
7 import cucumber.api.java.en.And;
8 import cucumber.api.junit.Cucumber;
9 import org.junit.runner.RunWith;
10
11 @RunWith(Cucumber.class)
12 public class MyStepDefinitions {
13
14     @Given("^I am on login page of billing$")
15     public void i_am_on_login_page_of_billing() throws Throwable {
16         throw new PendingException();
17     }
18
19     @When("^I enter valid username and password$")
20     public void i_enter_valid_username_and_password() throws Throwable {
21         throw new PendingException();
22     }
23
24     @Then("^I should be redirected to home page$")
25     public void i_should_be_redirected_to_home_page() throws Throwable {
26         throw new PendingException();
27     }
28
29     @And("^click on login button$")
30     public void click_on_login_button() throws Throwable {
31         throw new PendingException();
32     }
33
34 }
```

select Java steps

see the code

# Watch the video up to this

---

Cucumber Set up

<https://youtu.be/69c7k71P-vQ>

Tidy Gherkin plug-in

<https://youtu.be/7RILnwdGwvE>

# Parameterization

## Reuse function with different data

**Scenario:** Successful login for correct username and password  
**Given** I am on login page of billing  
**When** I enter "amolujagare@gmail.com" and "admin123"  
**And** click on login button  
**Then** I should be redirected to home page

```
@When("^I enter \"([^\"]*)\" and \"([^\"]*)\"$")  
    public void i_enter_something_and_something(String username, String password)  
    {  
        driver.findElement(By.name("email")).sendKeys(username);  
        driver.findElement(By.name("password")).sendKeys(password);  
    }
```

You can generate using any readymade options)



Scripting Logic

# Parameterization (Data tables)

---

First Name	<input type="text"/>
Last Name	<input type="text"/>
Phone no.	<input type="text"/>
Email Id	<input type="text"/>

- When : I enter “fname” “lname” “pnno” “email” & /or many fields
- Previous option is Not a good Choice



# Parameterization (Data tables)

**Scenario:** parameterization checking

*Given* I am on login page of billing

*When* I have following details

amol	ujagare	9975709708	amolujagare@gmail.com
------	---------	------------	-----------------------

```
@When("^I have following details$")
public void i_have_following_details(DataTable data) {

    List<List<String>> obj= data.asLists();

    System.out.println(obj.get(0).get(0));
    System.out.println(obj.get(0).get(1));
    System.out.println(obj.get(0).get(2));
    System.out.println(obj.get(0).get(3));

}
```



# Parameterization (Like Data Provider)

## Parameterization using example Keyword

**Scenario Outline:** Another parameterization checking  
**Given** I am on login page of billing  
**When** I enter <Username> and <password>

### Examples:

Username	password	
am011	uagare1	
am012	uagare2	
am013	uagare3	
am014	uagare4	
am015	uagare5	



# Parameterization (Like Data Provider)

- Implementation : Parameterization using example Keyword
- If this code is not getting generated after running the testRunner then use TIDY gherking plugin and remove ^ symbol at start & \$ symbol at the end

```
@When("^I enter (.+) and (.+)$")  
public void i_enter_and(String username, String password)  
{  
    System.out.println(username+" "+password);  
}
```



# Tagging feature to control test cases

**Feature:** Application Login

@RegTest

**Scenario:** Home page default login

**Given** User is on NetBanking landing page

**When** User login into application with "jin" and password "1234"

**Then** Home page is populated

**And** Cards displayed are "true"

@SmokeTest

**Scenario:** Home page default login

**Given** User is on NetBanking landing page

**When** User login into application with "john" and password "4321"

**Then** Home page is populated

**And** Cards displayed are "false"

@SmokeTest

**Scenario:** Home page default login

**Given** User is on NetBanking landing page

**When** User login into application with "john" and password "4321"

**Then** Home page is populated

**And** Cards displayed are "false"

@RegTest

**Scenario:** Home page default login

**Given** User is on NetBanking landing page

**When** User sign up with following details

| jenny | abcd | john@abcd.com | Australia | 3242353 |

# Tagging feature to control test cases

```
import org.junit.runner.RunWith;

@RunWith(Cucumber.class)
@CucumberOptions(
    features = "src/test/java/features",
    glue="stepDefinitions", tags="@SmokeTest,@RegTest")
public class TestRunner {
}
```



# Background keyword for prerequisite

## Background:

*Given* validate the browser

*When* Browser is triggered

*Then* check if browser is started

1. write this in feature file
2. generate the code
3. write implementation
4. this implementation will run before every scenario



# Background keyword

## –Key Points to remember

---

- Background keyword belongs to that feature file only where it is written
- This has to be written at the top only. If you try to write it anywhere in the feature file it will give you an error
- This case is only applicable when prerequisite is common for all scenarios (not for different prerequisites for different scenarios)





# Hooks

- We can write different prerequisites for different tags using hooks
- If we do not specify tag it will work like background
- Define it inside step definition
- Multiple tags can be used separated by comma e.g. `@Before("@tag1,@tag2")`

```
public class Hooks {  
  
    @Before("@MobileTest")  
    public void beforevalidation()  
    {  
        System.out.println("Before Mobile hook");  
    }  
  
    @After("@MobileTest")  
    public void Aftervalidation()  
    {  
        System.out.println(" After Mobile before hook");  
    }  
}
```



# Cucumber reports

---

- `plugin={}`
- Pretty – for colors
- `"html:target/cucumber"` for html reports in target folder
- `"json:target/cucumber.json"` for json reports in target folder (full implementation need to be done by developer)
- `"junit:target/cuckes.xml"` xml reports in jenkins





# Cucumber reports

## target

### cucumber

formatter.js

index.html

jquery-3.4.1.min.js

report.js

style.css

cuckes.xml

cucumber.json

"html:target/cucumber"

"json:target/cucumber.json"

"junit:target/cuckes.xml"



# Cucumber reports

```
@RunWith(Cucumber.class)
@CucumberOptions
(features = "src/test/java/Feature", dryRun=true,
glue="stepDefinition",
plugin= {"pretty", "html:target/cucumber",
        "json:target/cucumber.json", "junit:target/cuckes.xml"})

public class LoginTestRunner {

}
```



# Dry run

- It will just tell in the console if any step is implemented or not

```
@RunWith(Cucumber.class)
@cucumberOptions
(features = "src/test/java/Feature", dryRun=true,
glue="stepDefinition", plugin= {"pretty", "html:target/cucumber"})
public class LoginTestRunner {

}
```

There were undefined steps. You can implement missing steps with the snippets below

```
@Given("I am on login page of billing")
public void i_am_on_login_page_of_billing() {
    // Write code here that turns the phrase above into concrete actions
    throw new cucumber.api.PendingException();
}
```



# Other attributes

- `monochrome=true` : well formatted output
- `strict=true` : if you don't want to pass the test if the implementation of step definition is not done

# More Free content

---

- **Free Text Tutorials**

<https://scriptinglogic.com>

- **Free Courses**

<https://scriptinglogic.in>

- **YouTube channel**

<https://www.youtube.com/amolujagare>



# Contact for training

---

- **Selenium**
- **Cucumber**
- **API Testing (Soap UI / REST assured / POSTMAN)**
- **Call / whatsApp 9529802751, 9657852251**
- **Visit <https://scriptinglogic.org/>**



# Scripting Logic

- ❖ Get Trained in person individually.
- ❖ Complete Practical Approach.
- ❖ Training from industry Expert
- ❖ Get confidence to do programming irrespective of your background
- ❖ Life time access to video Tutorials.
- ❖ Pre & post job support.

## Automation Testing

- Selenium
- Cucumber
- API Testing  
(SoapUI/REST Assured/  
POSTMAN)

**Call / WhatsApp: 9529802751, 9657852251**

**Visit : <https://scriptinglogic.org>**



**Scripting Logic**