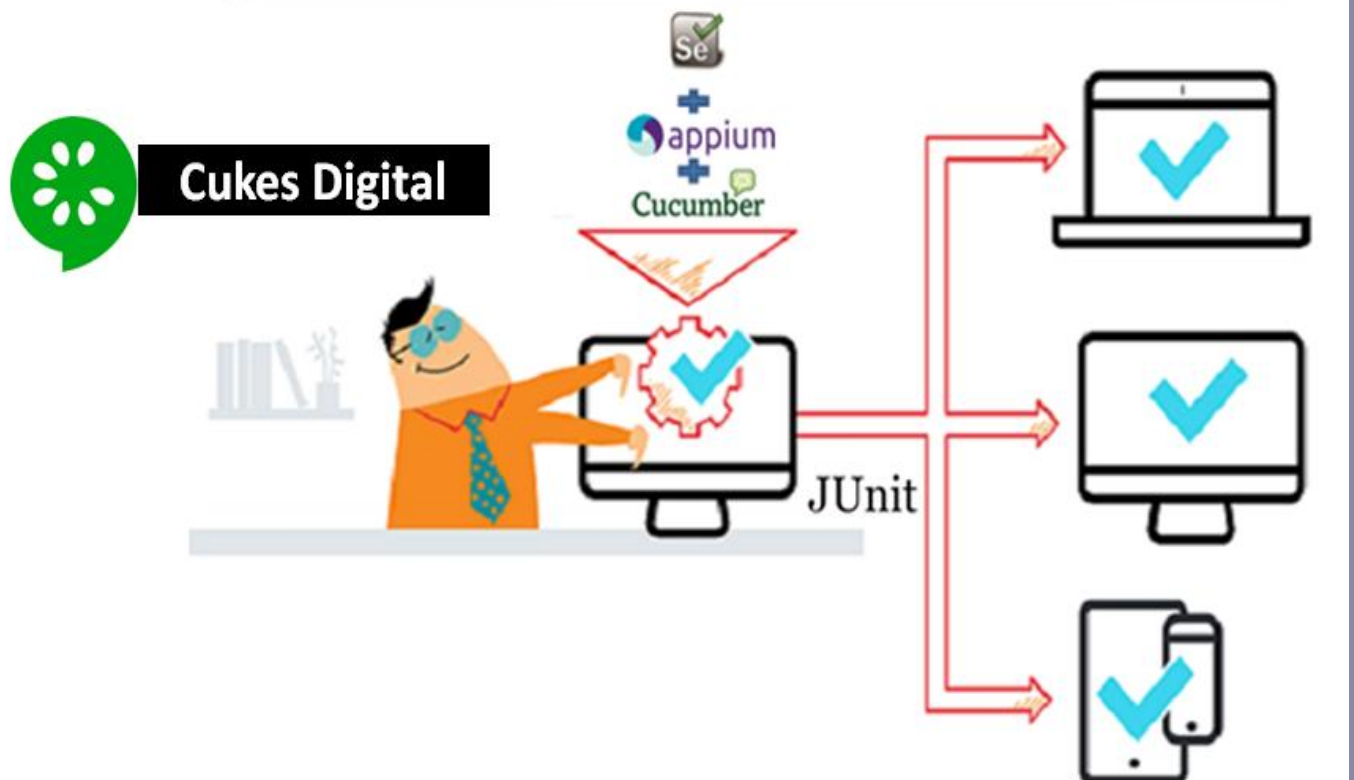


2016

# User Guide for Cukes-Digital



Author: Aniruddh Bajpai  
Guided By: Arun Pakkan  
7/22/2016

## Table of Contents

Summary: .....	3
Skill set required for using Cukes - Digital: .....	3
Tool set required for Cukes- Digital: .....	3
Setting Up Prerequisites for Android: .....	3
Setting Up Prerequisites for iOS: .....	5
Setting Up Cukes - Digital framework: .....	5
Cukes- Mobile Testing: .....	6
Cukes- Mobile: CREATE: .....	7
Cukes- Mobile: Create Feature and Step Definition: .....	8
Cukes- Mobile: Create Step Definition: .....	10
Cukes- Mobile: Tag: .....	11
Cukes- Mobile: Execute: .....	12
Cukes- Mobile: Execute: Real Device .....	12
Cukes- Web Testing: .....	15
Cukes- Web: CREATE: .....	15
Cukes- Web: Create Feature and Step Definition: .....	16
Cukes- Web: Create Step Definition: .....	19
Cukes- Web: Execute: Tags .....	21
Cukes- Web: Execute: .....	21

## Summary:

Capgemini Cukes Digital Framework provides a comprehensive Test Automation solution for Cross channel applications leveraging the Agile-BDD methodologies and it allows seamless integration with Cloud , DevOps tools and services

## Skill set required for using Cukes - Digital:

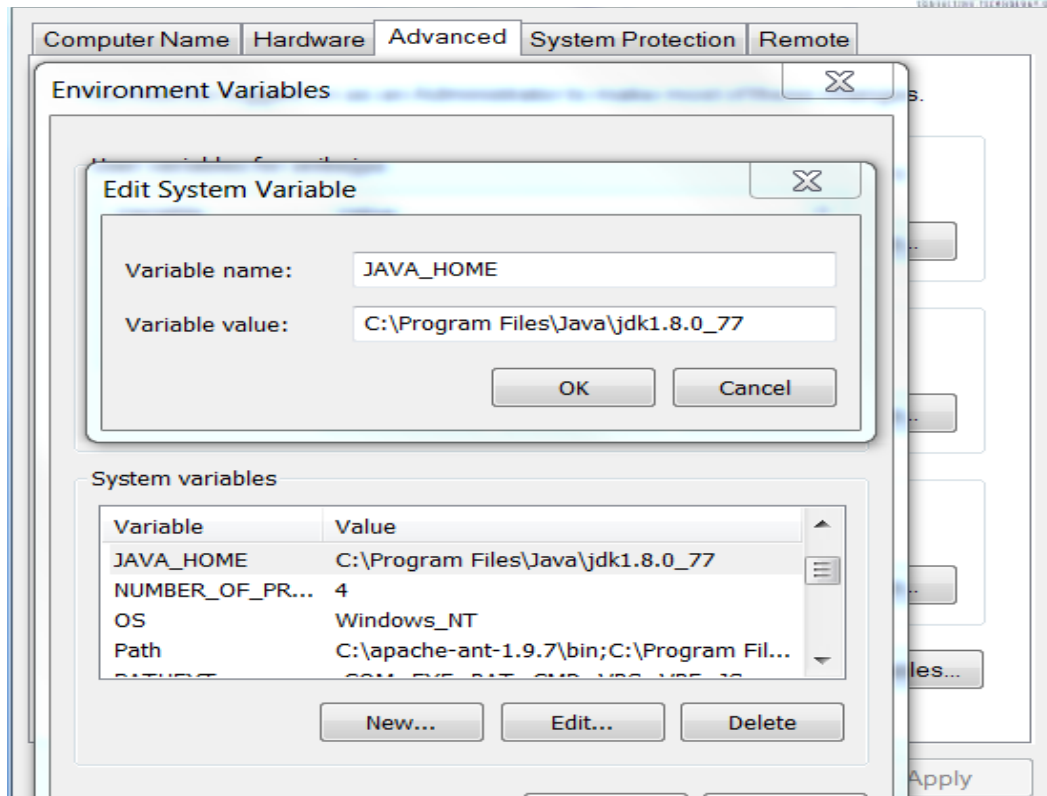
- 1) Basics of Automation Testing
- 2) Basics of Java
- 3) Basics of Selenium Web Driver
- 4) Basics of Appium
- 5) Understanding in POM
- 6) Understanding of Cucumber

## Tool set required for Cukes- Digital:

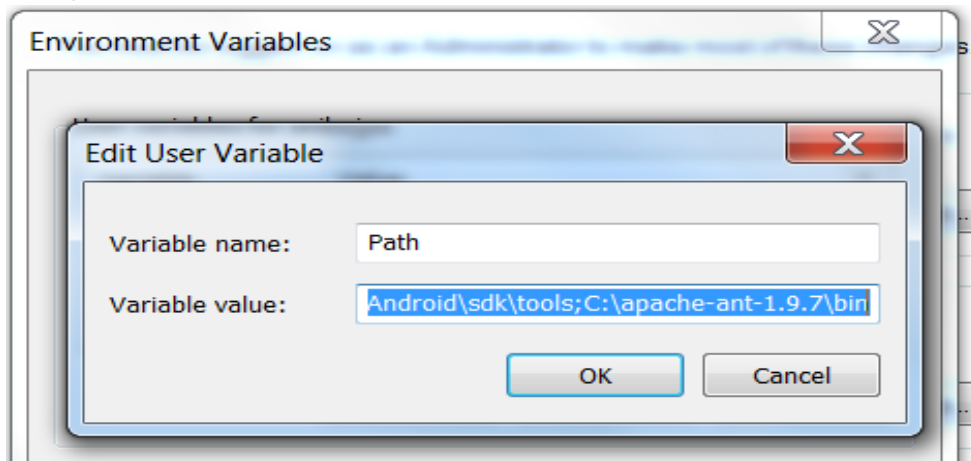
- 1- Eclipse latest version
- 2- JDK 1.8, JRE 8.0
- 3- Android SDK for Android
- 4- Xcode for iOS
- 5- In Windows we can perform execution only on android device
- 6- In Mac we can perform Execution on both devices android as well as ios.
- 7- Selenium Jar files
- 8- Appium Java-Client
- 9- Appium Desktop Application
- 10- Extent-Report Jar file
- 11- Cucumber Jar File

## Setting Up Prerequisites for Android:

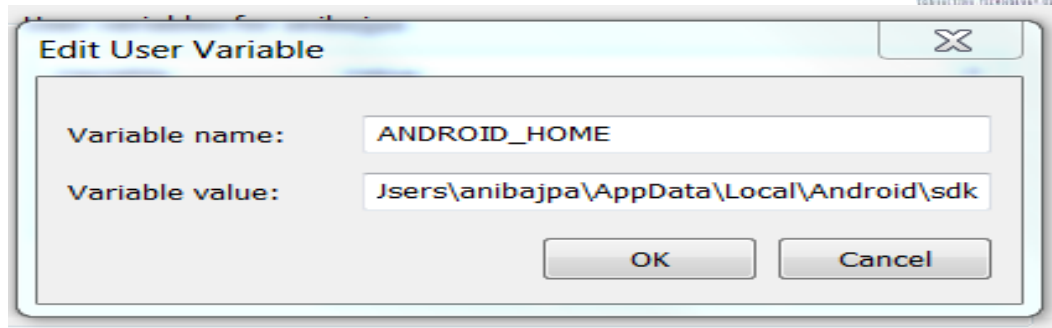
- 1- Set up JAVA\_HOME in environment variable.



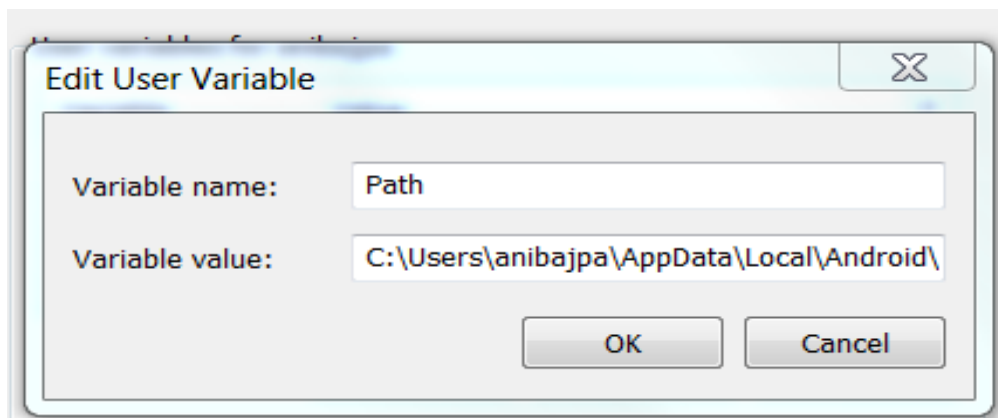
2- Set up Java Path in environment variable.



3- Set up ANDROID\_HOME in environment variable.



- 4- Set up Android Path in environment variable:  
Path will be till tools & Platform tools folder in Android SDK.



- 5- Enable debugging option in device.
- Go to setting
  - Go to about device
  - Click multiple time on build number and this will enable Developer Option
  - Go to Developer Option and enable USB Debugging.
- 6- Connect device with system and run cmd and type command `adb devices` it will list connected device in system.

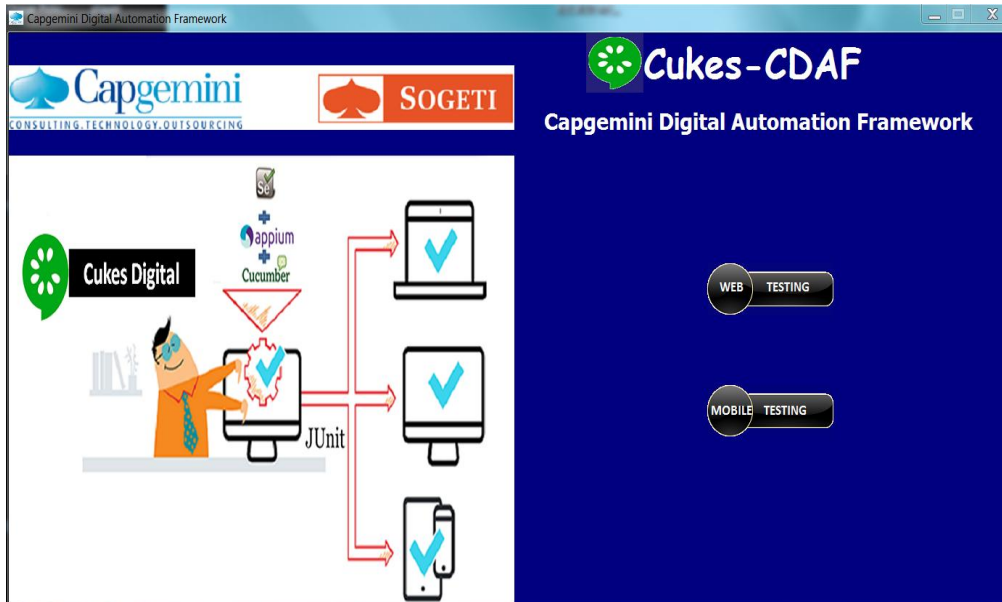
### Setting Up Prerequisites for iOS:

- 1- MAC OS-10.11 recommended
- 2- Xcode 6.0, 7.1.1 recommended
- 3- Apple Developer Tools (iPhone simulator SDK, command line tools)

### Setting Up Cukes - Digital framework:

- 1) Extract The Cukes-Digital Zip file on your local machine provided by Mobile COE.

- 2) Create a local workspace in Eclipse and import Cukes- Digital into it.
- 3) Open the Cukes – Digital Project in eclipse.
- 4) Launch the Cukes – Digital front screen will display which looks like below



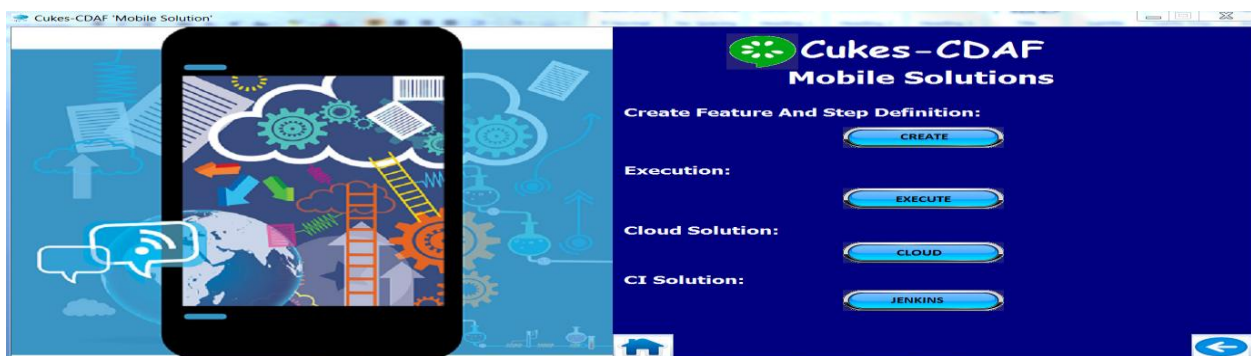
- 5) Select the present option accordingly, If you want to go with Mobile Testing (Native, Hybrid, Web) then select Mobile Testing. If you want to go Web Testing then go with Web Testing

### Cukes- Mobile Testing:

Cukes – Digital offer to perform mobile automation testing in BDD methodology with various options like cloud based testing, real device testing, Jenkins for CI integration.

You can directly go to different options with Mobile- Cukes digital window.

### CUKES-DIGITAL MOBILE WINDOW:



Once you will navigate to CUKES MOBILE WINDOW you will get different options:

- 1- Create
- 2- Execute
- 3- Cloud
- 4- Jenkins

We will talk about all these options how they work and how to maintain them.

### Cukes- Mobile: CREATE:

Click on CREATE option and navigate to 'Create' window, on this window you will get multiple options like:

- 1- Create Feature Definition
- 2- Create Step Definition
- 3- Tag

### Feature Definition:

As we following BDD approach for this framework so we required to create feature definition file in gherkin language with different annotations like Scenario , Feature, Back ground ,Given, When , Then , And , But ,\* , Outline , Example ,tags etc

Let's have overview what this annotation does:

**GIVEN:** Given defines a precondition to the test.

**WHEN:** When keyword defines the test action that will be executed. By test action we mean the user input action.

**THEN:** Then keyword defines the Outcome of previous steps.

**AND:** And keyword is used to add conditions to your steps.

**BUT:** But keyword is used to add negative type comments.

\* **Keyword:** This keyword is very special. This keyword defies the whole purpose of having Given, When, Then and all the other keywords. Basically Cucumber doesn't care about what Keyword you use to define test steps, all it cares about what code it needs to execute for each step.

**Scenario:** This cucumber tag use to define what test case we going to script.

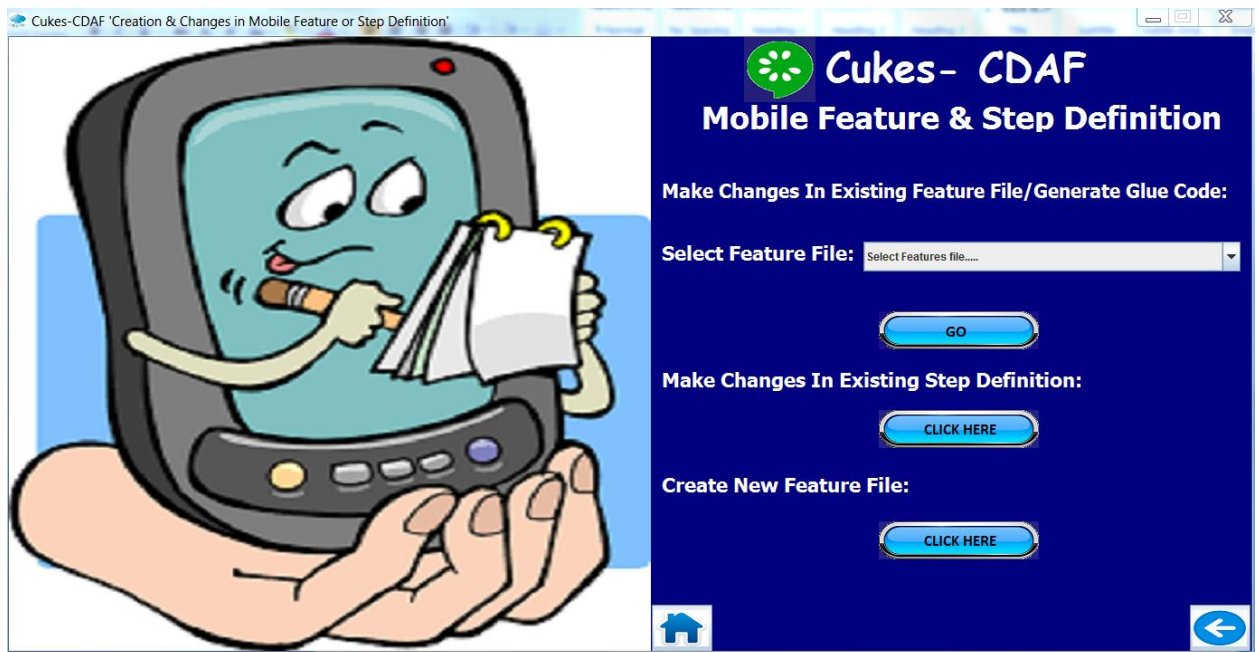
**Feature:** This cucumber tag use to define what kind of scenario we are going to cover.

**Scenario Outline and Example:** This cucumber tag use to cover data driven approach.

### **Cukes- Mobile: Create Feature and Step Definition:**

Click on Create Feature and Step Definition option and navigate to 'Create Feature and Step Definition' window, on this window you will get multiple options like:

- 1- Make changes to existing feature and generate glue code
- 2- Make changes in step definition
- 3- Create new feature



### **Make changes to existing feature and generate glue code:**

If you want to make changes in existing feature file then select feature from drop down which you had previously created.

### **Make changes in existing step definition:**

If you want to make changes in existing step definition then click on 'CLICK HERE' it will take to new window and you can make change accordingly and Save the changes



```
Viewing Mobile Step Definition Details
```

```
    }

    @Given("the user is on landing page$")
    public void the_user_is_on_landing_page() throws Throwable {
        test = extent.startTest("Launching the browser", "Launching the browser");
        test.log(LogStatus.INFO, "Launching the browser");
        driver.get("http://www.tgifindia.com");
        test.log(LogStatus.PASS, "Browser Launch successfully and user navigated to TGIF home screen");
        test.log(LogStatus.INFO, test.addScreenCapture(Report.CaptureScreen(driver,"Results/Screenshot")));
    }

    @Then("user click on menu button$")
    public void user_click_on_menu_button() throws Throwable {
        new Mobilelauncher(driver).user_chooses_to_menu();
        test.log(LogStatus.PASS, "User click on menu button");
        test.log(LogStatus.INFO, test.addScreenCapture(Report.CaptureScreen(driver,"Results/Screenshot")));
    }

    @Then("user chooses to our story$")
    public void user_chooses_to_our_story() throws Throwable {
        new Mobilelauncher(driver).user_chooses_our_story();
        Thread.sleep(10000);
        test.log(LogStatus.PASS, "User navigate to Our story screen");
        test.log(LogStatus.INFO, test.addScreenCapture(Report.CaptureScreen(driver,"Results/Screenshot")));
    }
}
```

Save Changes  
Go Back

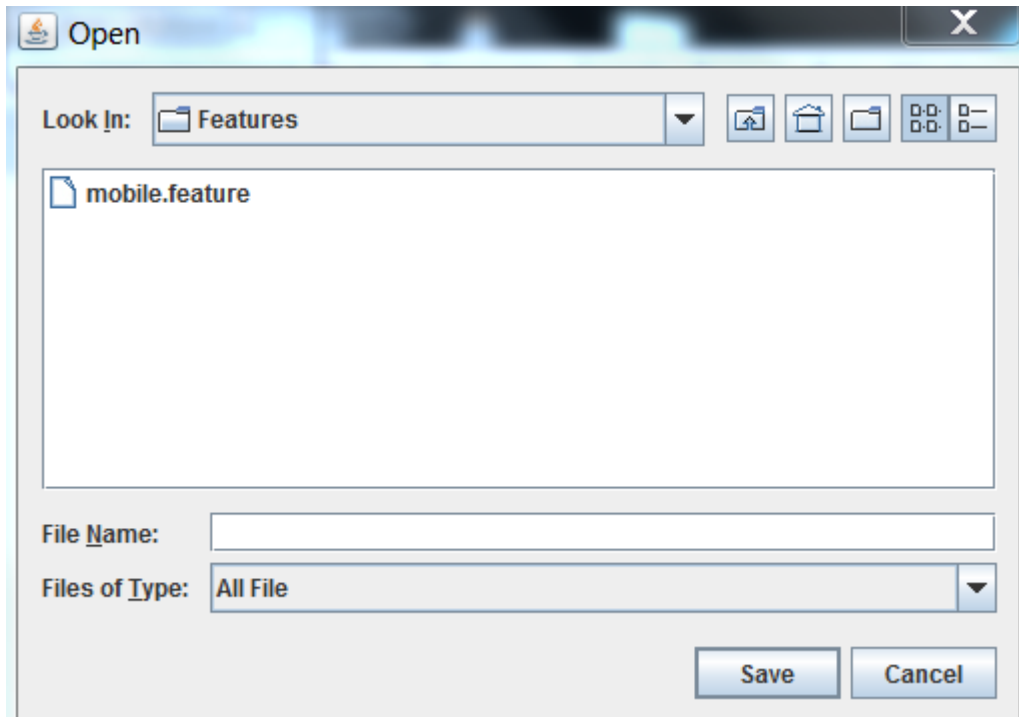
## Create New Feature File:

If you want to create new feature file then 'CLICK HERE' it will take to new window and you write down new feature and save accordingly what name you required with extension .feature

## CREATE WINDOW



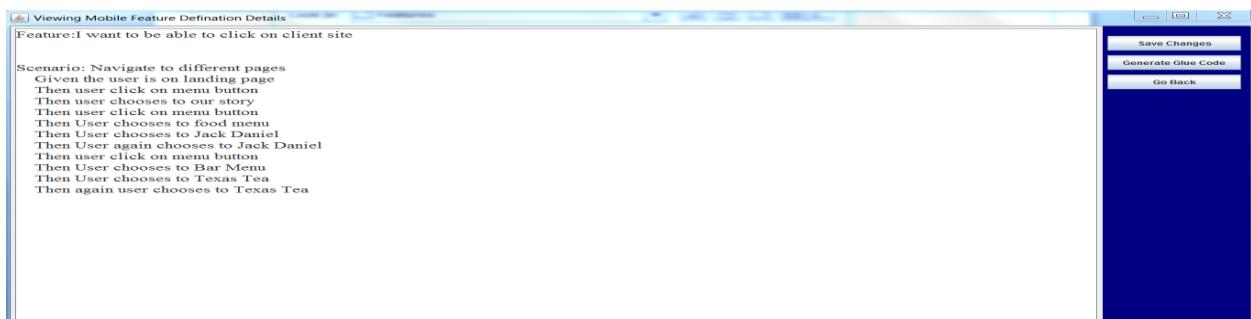
## SAVE WINDOW:




Once creation or modification done go back to Mobile window and generate step definition to make feature run able and interact with application

## Cukes- Mobile: Create Step Definition:

To create step definition you required to generate glue code. Glue code in empty structure of methods with annotation of cucumber. It allows us to make feature definition file run able and interact with applications. Click on generate glue code button and copy that code and paste in step definition field and write down code accordingly to make run able.





```

new Mobile_Bar(driver).user_click_on_Bar_Menu();
test.log(LogStatus.PASS, "User click on bar menu");
test.log(LogStatus.INFO, test.addScreenCapture(Report.CaptureScreen(driver,"Results/Screenshot")));
}

@Then("^User chooses to Texas Tea$")
public void user_chooses_to_Texas_Tea() throws Throwable {

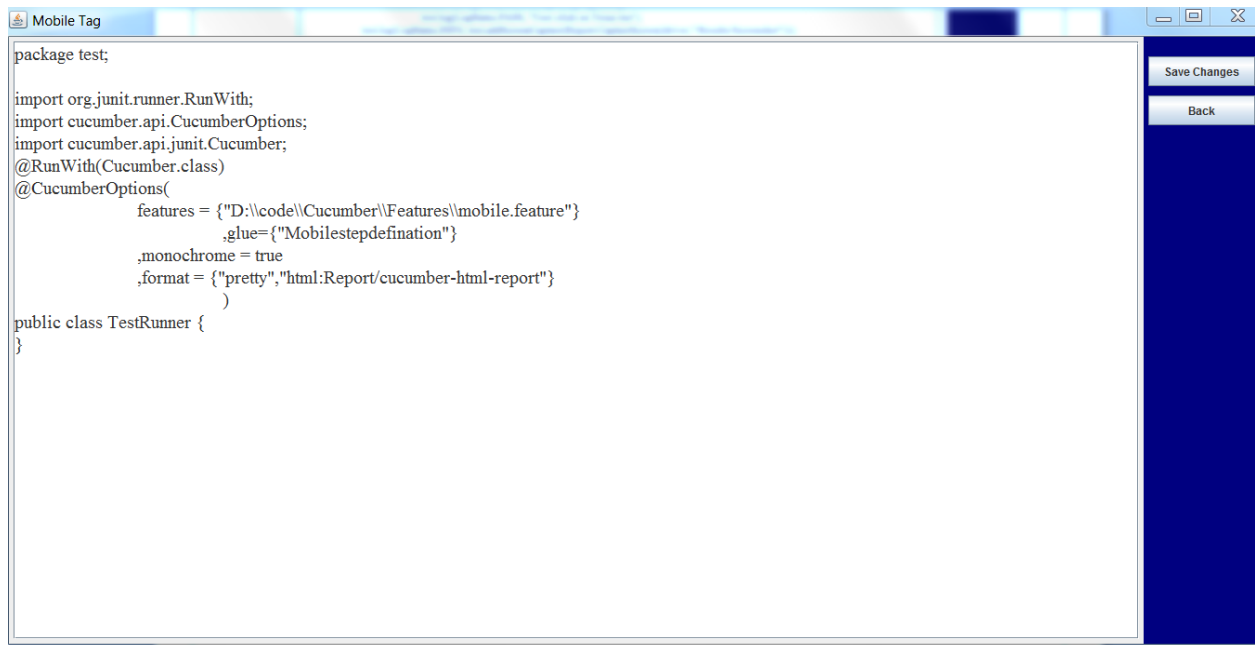
    //String text = driver.findElement(By.xpath("/html/body/div/div[1]/div[2]/div[2]/ul/li[1]/a")).getText();
    Thread.sleep(10000);
    new Mobile_Bar(driver).user_click_on_Texas_Tea();
    test.log(LogStatus.PASS, "User click on Texas tea");
test.log(LogStatus.INFO, test.addScreenCapture(Report.CaptureScreen(driver,"Results/Screenshot")));
}

@Then("^again user chooses to Texas Tea$")
public void again_user_chooses_to_Texas_Tea() throws Throwable {
    new Mobile_Bar(driver).user_again_click_on_Texas_Tea();
    test.log(LogStatus.PASS, "User again click on texas tea");
test.log(LogStatus.INFO, test.addScreenCapture(Report.CaptureScreen(driver,"Results/Screenshot")));
}
}

```

## Cukes- Mobile: Tag:

Click on tag button it take you to runner class and define the scenario tag you want run



```

package test;

import org.junit.runner.RunWith;
import cucumber.api.CucumberOptions;
import cucumber.api.junit.Cucumber;
@RunWith(Cucumber.class)
@CucumberOptions(
    features = {"D:\\code\\Cucumber\\Features\\mobile.feature"}
    ,glue={"Mobilestepdefination"}
    ,monochrome = true
    ,format = {"pretty","html:Report/cucumber-html-report"}
)
public class TestRunner {
}

```

### Cukes- Mobile: Execute:

Navigate back to Cukes mobile window and select option 'EXECUTE'.

In execute window you will different fields like tag , tag specifies that you want to execute specific feature or scenario.

### Cukes- Mobile: Execute: Real Device

Provide required data and click on EXECUTE button. After clicking execution will began on real device which is connected to system.

If you want to cast your device in screen click on 'SCREEN-CAST'. This will display device on desktop.

### Cukes –Cloud:

Right now cukes –digital support Browser Stack(Mobile Web) and Perfecto (Native , Hybrid), to perform cloud test select the device and provide data



For native and hybrid navigate to Cukes perfecto window and provide data accordingly and perform execution:

Cukes-CDAF 'Perfecto Cloud Solution'



### Cukes-CDAF Perfecto Cloud Solution

Provide Data For Execution:

Platform Name:

Platform Version:

Manufacturer:

Model:

Package Name:

Activity Name:



For mobile web navigate to Cukes browser stack window and provide data accordingly and perform execution:

Cukes-CDAF 'Browser-Stack Solution'




### Cukes-CDAF Browser Stack Cloud Solution

Provide Data For Execution :

Browser Name:

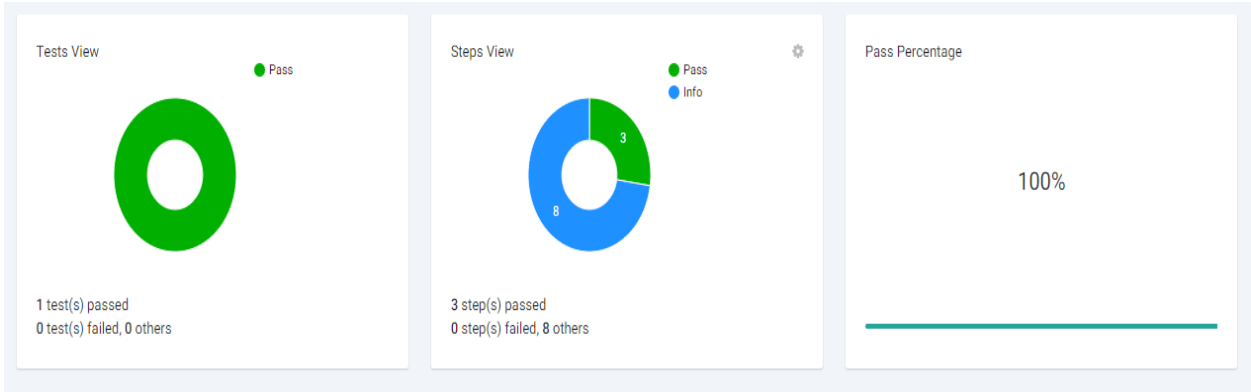
Platform:

Device:



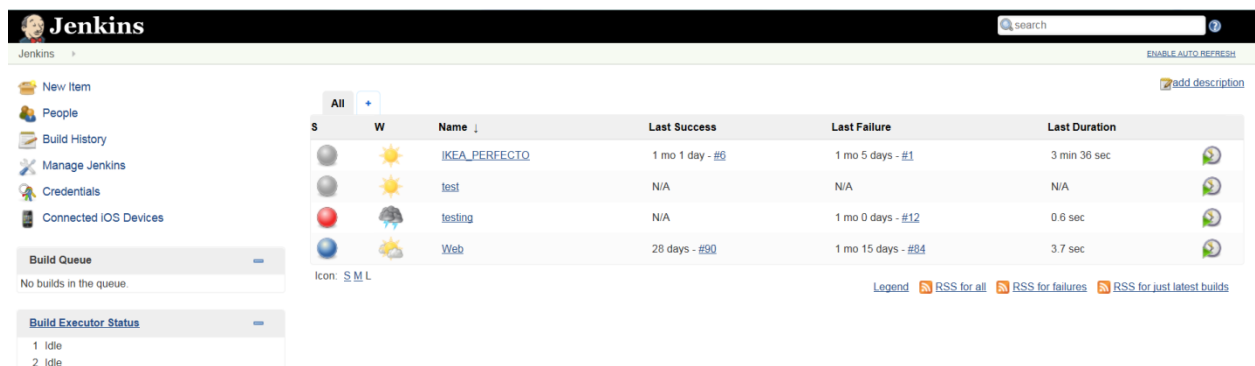
## Cukes –Mobile Results:

After completion of automation suite, module wise folder structure will be created through automation scripts to hold html reports. HTML reports will have step wise description of action performed along with link to the screenshot for that specific action with complete screenshot.



## Cukes –Mobile Jenkins: \

Jenkins use to provide CI integration solution, If you need to perform CI solution then click on Jenkins option and will take you to Jenkins window and began the execution.

The screenshot shows the Jenkins web interface. The main content area displays a table of build history:

S	W	Name ↓	Last Success	Last Failure	Last Duration
●	☀	<a href="#">IKEA_PERFECTO</a>	1 mo 1 day - #6	1 mo 5 days - #1	3 min 36 sec
●	☀	test	N/A	N/A	N/A
●	☁	testing	N/A	1 mo 0 days - #12	0.6 sec
●	☀	Web	28 days - #90	1 mo 15 days - #84	3.7 sec

Additional interface elements include a search bar, navigation menu (New Item, People, Build History, Manage Jenkins, Credentials, Connected iOS Devices), Build Queue (No builds in the queue), and Build Executor Status (1 Idle, 2 Idle).

## Cukes- Web Testing:

Cukes – Digital offer to perform web automation testing in BDD methodology with various options like cloud based testing, real device testing, Jenkins for CI integration.

You can directly go to different options with Mobile- Cukes digital window.

### CUKES-DIGITAL WEB WINDOW:



Once you will navigate to CUKES WEB WINDOW you will get different options:

- 5- Create
- 6- Execute
- 7- Cloud
- 8- Jenkins

We will talk about all these options how they work and how to maintain them.

### Cukes- Web: CREATE:

Click on CREATE option and navigate to 'Create' window, on this window you will get multiple options like:

- 4- Create Feature Definition
- 5- Create Step Definition
- 6- Tag

### Feature Definition:

As we following BDD approach for this framework so we required to create feature definition file in gherkin language with different annotations like Scenario , Feature, Back ground ,Given, When , Then , And , But ,\* , Outline , Example ,tags etc

Let's have overview what this annotation does:

**GIVEN:** Given defines a precondition to the test.

**WHEN:** When keyword defines the test action that will be executed. By test action we mean the user input action.

**THEN:** Then keyword defines the Outcome of previous steps.

**AND:** And keyword is used to add conditions to your steps.

**BUT:** But keyword is used to add negative type comments.

\* **Keyword:** This keyword is very special. This keyword defies the whole purpose of having Given, When, Then and all the other keywords. Basically Cucumber doesn't care about what Keyword you use to define test steps, all it cares about what code it needs to execute for each step.

**Scenario:** This cucumber tag use to define what test case we going to script.

**Feature:** This cucumber tag use to define what kind of scenario we are going to cover.

**Scenario Outline and Example:** This cucumber tag use to cover data driven approach.

### Cukes- Web: Create Feature and Step Definition:

Click on Create Feature and Step Definition option and navigate to 'Create Feature and Step Definition' window, on this window you will get multiple options like:

- 4- Make changes to existing feature and generate glue code
- 5- Make changes in step definition
- 6- Create new feature





### **Make changes to existing feature and generate glue code:**

If you want to make changes in existing feature file then select feature from drop down which you had previously created.

### **Make changes in existing step definition:**

If you want to make changes in existing step definition then click on 'CLICK HERE' it will take to new window and you can make change accordingly and Save the changes

Viewing Web Step Definition Details

```

@Then("^User chooses to Bar Menu$")
public void user_chooses_to_Bar_Menu() throws Throwable {
    new Bar(driver).user_click_on_Bar_Menu();
    test.log(LogStatus.PASS, "User click on bar menu");
}

@Then("^User chooses to Texas Tea$")
public void user_chooses_to_Texas_Tea() throws Throwable {
    new Bar(driver).user_click_on_Texas_Tea();
    test.log(LogStatus.PASS, "User click on Texas tea");
}

@Then("^again user chooses to Texas Tea$")
public void again_user_chooses_to_Texas_Tea() throws Throwable {
    new Bar(driver).user_again_click_on_Texas_Tea();
    test.log(LogStatus.PASS, "User again click on texas tea");
}

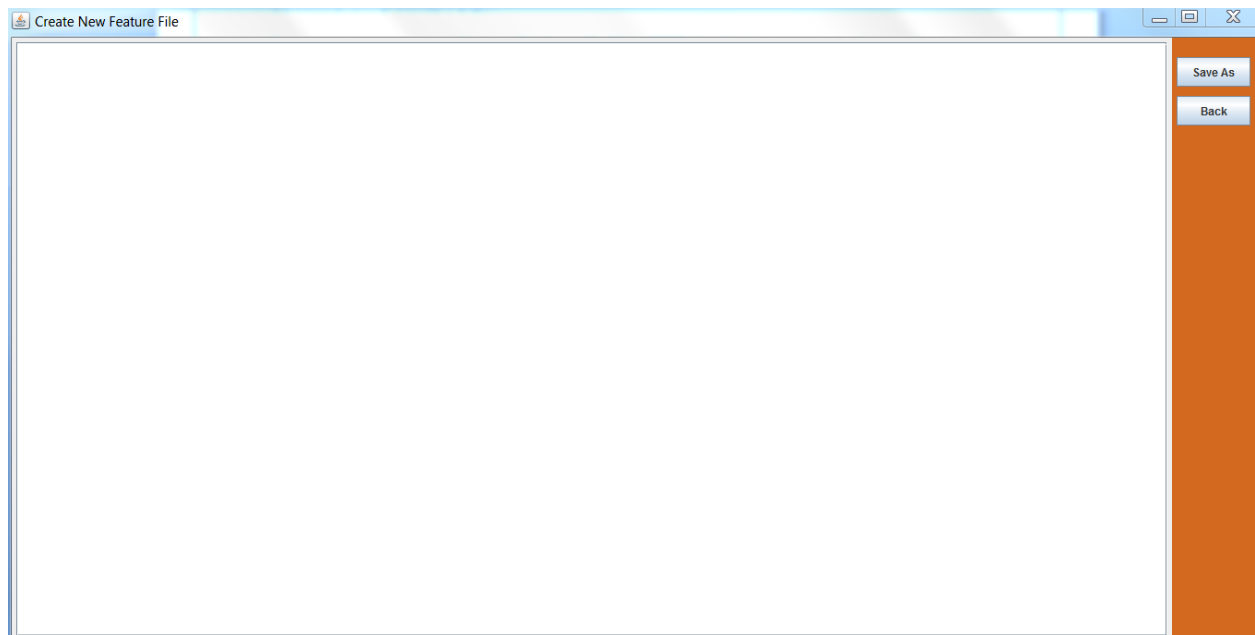
```

Save Changes  
Back

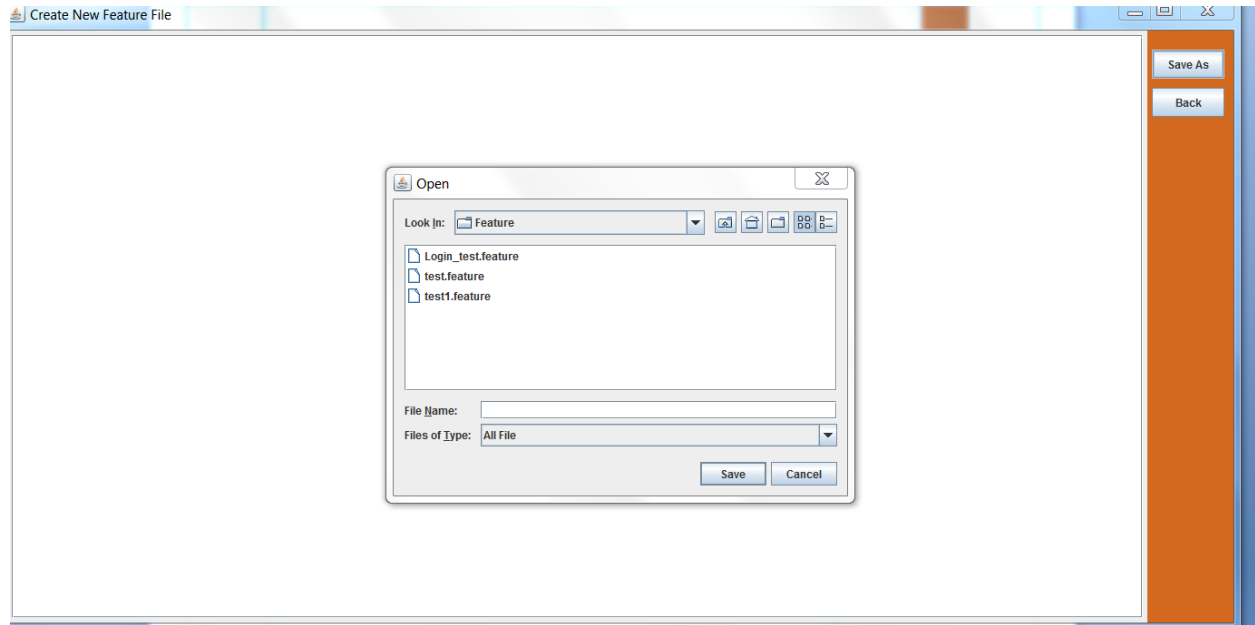
## Create New Feature File:

If you want to create new feature file then 'CLICK HERE' it will take to new window and you write down new feature and save accordingly what name you required with extension .feature

## CREATE WINDOW



## SAVE WINDOW:



Once creation or modification done go back to Web window and generate step definition to make feature run able and interact with application

## Cukes- Web: Create Step Definition:

To create step definition you required to generate glue code. Glue code in empty structure of methods with annotation of cucumber. It allows us to make feature definition file run able and interact with applications. Click on generate glue code button and copy that code and paste in step definition field and write down code accordingly to make run able.

Viewing Web Step Definition Details

```

@Then("^User chooses to Bar Menu$")
public void user_choses_to_Bar_Menu() throws Throwable {
    new Bar(driver).user_click_on_Bar_Menu();
    test.log(LogStatus.PASS, "User click on bar menu");
    test.log(LogStatus.INFO, test.addScreenCapture(Report.CaptureScreen(driver,"Results/Screenshot")));
}

@Then("^User chooses to Texas Tea$")
public void user_choses_to_Texas_Tea() throws Throwable {
    new Bar(driver).user_click_on_Texas_Tea();
    test.log(LogStatus.PASS, "User click on Texas tea");
    test.log(LogStatus.INFO, test.addScreenCapture(Report.CaptureScreen(driver,"Results/Screenshot")));
}

@Then("^again user chooses to Texas Tea$")
public void again_user_choses_to_Texas_Tea() throws Throwable {
    new Bar(driver).user_again_click_on_Texas_Tea();
    test.log(LogStatus.PASS, "User again click on texas tea");
    test.log(LogStatus.INFO, test.addScreenCapture(Report.CaptureScreen(driver,"Results/Screenshot")));
}
}

```

Save Changes  
Back

## Cukes- WEB: Tags:

Click on tag button it take you to runner class and define the scenario tag you want run

Web Tag

```

package cucumberTest;
import org.junit.runner.RunWith;
import cucumber.api.CucumberOptions;
import cucumber.api.junit.Cucumber;
@RunWith(Cucumber.class)
@CucumberOptions(
    features = {"D:\\code\\Cucumber\\Feature\\Login_test.feature"}
    ,glue={"stepDefinition"}
    ,monochrome = true
    ,format = {"pretty","html:Report/cucumber-html-report"}
)
public class TestRunner {
}

```

Save Changes  
Back

## Cukes- Web: Execute: Tags

Navigate back to Cukes mobile window and select option 'EXECUTE'.

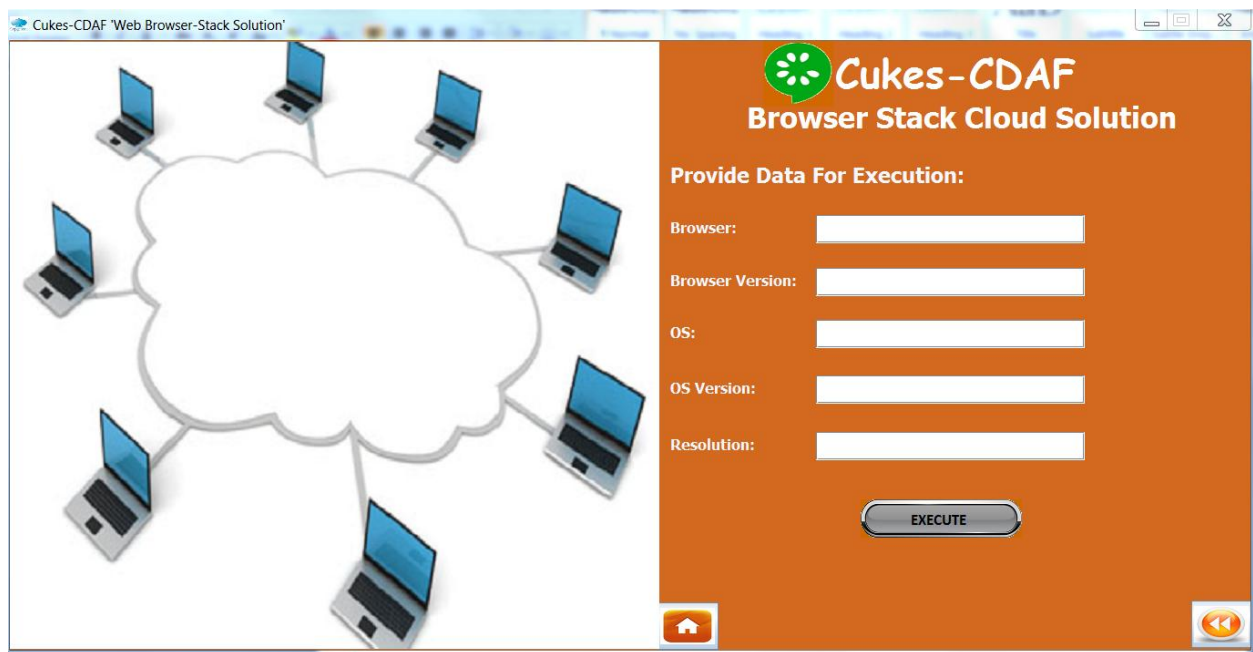
In execute window you will different fields like tag , tag specifies that you want to execute specific feature or scenario.

## Cukes- Web: Execute:

Provide required data and click on EXECUTE button. After clicking execution will began on all browser which is selected.

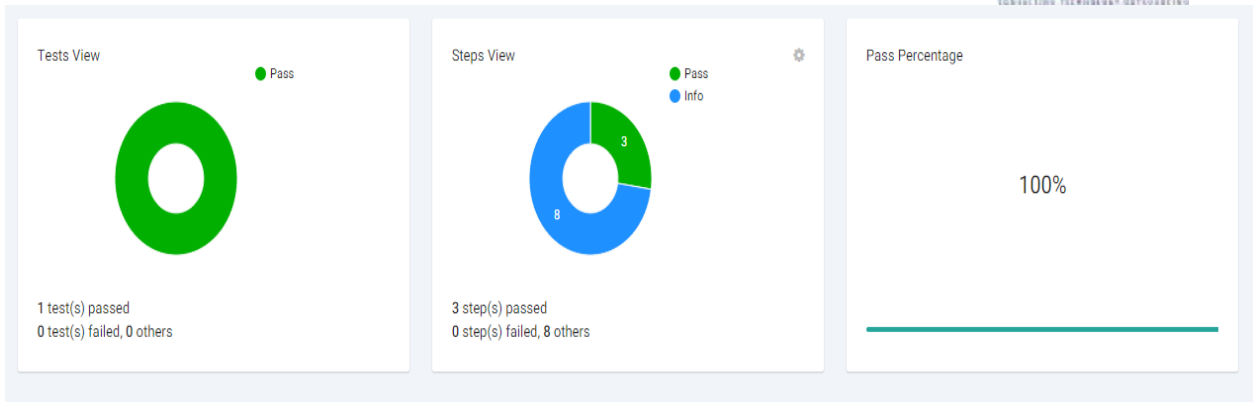
## Cukes –WEB: Cloud:

Right now cukes –digital support Browser Stack(Web) select the browser and provide data



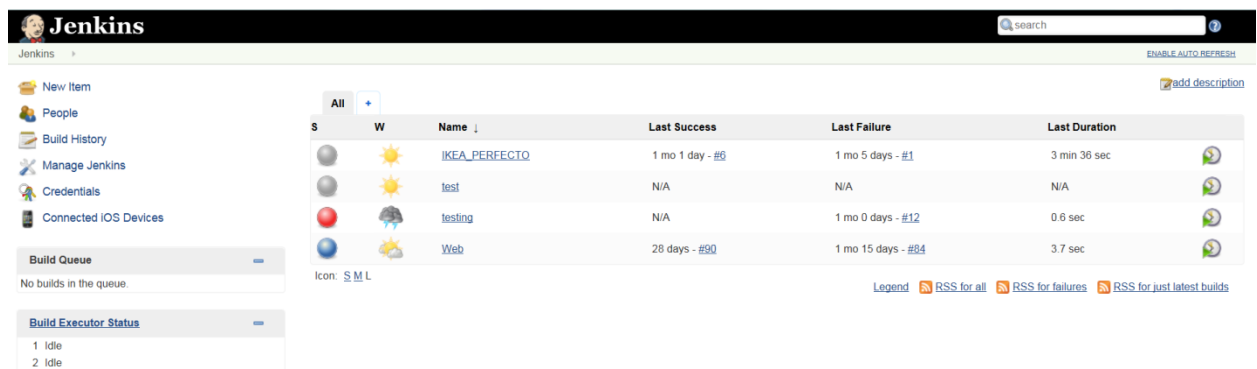
## Cukes – Web Results:

After completion of automation suite, module wise folder structure will be created through automation scripts to hold html reports. HTML reports will have step wise description of action performed along with link to the screenshot for that specific action with complete screenshot.



## Cukes –WEB Jenkins: \

Jenkins use to provide CI integration solution, If you need to perform CI solution then click on Jenkins option and will take you to Jenkins window and began the execution.

**Jenkins**

Build Queue: No builds in the queue.

Build Executor Status: 1 Idle, 2 Idle.

S	W	Name ↓	Last Success	Last Failure	Last Duration
☉	☀	IKEA_PERFECTO	1 mo 1 day - #6	1 mo 5 days - #1	3 min 36 sec
☉	☀	test	N/A	N/A	N/A
☹	☁	testing	N/A	1 mo 0 days - #12	0.6 sec
🌐	☀	Web	28 days - #90	1 mo 15 days - #84	3.7 sec

Icon: S M L

Legend: RSS for all, RSS for failures, RSS for just latest builds