**Java Framework Solution**

**Instructions for the installation of the Java Framework**

**On MAC**

1. **Install the JDK 8.**

[**http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html**](http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html)

1. **Install the JRE.**

[**http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html**](http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html)

1. **Install the XCODE from the Apple Store.**

**Note:** You need an Apple ID. For create it go to the following link:

<https://appleid.apple.com/#!&page=signin>

1. **Install brew package manager.**

Open a terminal and execute the command:

*/usr/bin/ruby -e "$(curl -fsSL* [*https://raw.githubusercontent.com/Homebrew/install/master/install*](https://raw.githubusercontent.com/Homebrew/install/master/install)*)"*

1. **Install Node.**

In the terminal execute: *brew install node*

1. **Install Appium.**

In a terminal run the following command: *npm install appium@latest –g*

1. **Install Carthage.**

In the same terminal execute the command: *brew install carthage*

1. **Install Maven.**

Download the binary file from <https://maven.apache.org/download.cgi>

1. **Install Intelllij IDEA.**

Download the Community version from <https://www.jetbrains.com/idea/download>

1. **Install Android Studio.**

a. Download the Android Studio: https://developer.android.com/studio/install.html

b. Go to Android Studio >> Tools >> Android >> SDK Tools and

1. From SDK Platforms, Download the Android SDK (e.g Android SDK Platform 24)

and the System image (Intel x86 Atom\_64) you need to create the emulators

2. From SDK Tools, make sure that "Intel x86 Emulator Accelerator (HAXM installer)"

is installed and updated.

c. Set ANDROID\_SDK\_ROOT

(source: <https://developer.android.com/studio/command->line/variables.html)

1. Check SDK installation folder. The path should be similar to:

/Users/<your\_user>/Library/Android/sdk

2. Run the following command: **open -e .bash\_profile**

3. You should have something similar than:

# ANDROID SDK

export PATH=$PATH:/Users/<your\_user>/Library/Android/sdk/platform-tools

export PATH=$PATH:/Users/<your\_user>/Library/Android/sdk/emulator

export PATH=$PATH:/Users/<your\_user>/Library/Android/sdk/tools/bin

export PATH=$PATH:/Users/<your\_user>/Library/Android/sdk/build-tools

export ANDROID\_HOME=/Users/user/Library/Android/sdk

# JAVA HOME DEFAULT

export JAVA\_HOME=$(/usr/libexec/java\_home)

# JAVA HOME SPECIFIC VERSION

export JAVA\_HOME=$ (/usr/libexec/java\_home -v 1.8.0\_66)

# MAVEN

export PATH=$PATH:/Users/user/apache-maven-3.5.2/bin

# ANDROID\_SDK\_ROOT

export ANDROID\_SDK\_ROOT=/Users/<your\_user>/Library/Android/sdk

# APPIUM

export PATH=$PATH:/Users/user/.npm-global/bin

4. Save and close the changes.

5. Apply changes on mac Os: **source ~/.bash\_profile**

6. You may need to close all consoles and apps which work with JAVA

**Other Tools**

**Install MySQL Server and Workbench**

1. Download the Mysql Server application from:

[**https://dev.mysql.com/downloads/mysql/**](https://dev.mysql.com/downloads/mysql/)

1. Download the Workbench application from:

[**https://dev.mysql.com/downloads/workbench/**](https://dev.mysql.com/downloads/workbench/)

**Install Mongodb**

1. [**https://www.mongodb.com/blog/post/mongodb-3210-is-released**](https://www.mongodb.com/blog/post/mongodb-3210-is-released)
2. **Create the folders** /data/db in the Home

**Note:** The user should has permissions.

**Run:** *mongod --storageEngine=mmapv1*

**On WINDOWS**

1. **Install the JDK 8.**

[**http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html**](http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html)

1. **Install the JRE.**

[**http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html**](http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html)

1. **Install Node and Appium**

Execute the following commands:

*npm i -g npm*

*npm install --global --production windows-build-tools*

*npm install appium@latest –g*

1. **Install Apache Maven**

Download the binary file from: <https://maven.apache.org/download.cgi>

1. **Install Intelllij IDEA**

Download the Community version from <https://www.jetbrains.com/idea/download>

1. **Install Android Studio.**

Download the Android Studio: https://developer.android.com/studio/install.html

**Other Tools**

**Install MySQL Server and Workbench**

1. Download the Mysql Server application from:

**<https://dev.mysql.com/downloads/mysql/>**

1. Download the Workbench application from:

**<https://dev.mysql.com/downloads/workbench/>**

**Install Mongodb**

1. **<https://www.mongodb.com/blog/post/mongodb-3210-is-released>**
2. **Create the folders** /data/db in the root.
3. **Go to C:\Program Files (x86)\MongoDB\Server\3.2\bin**

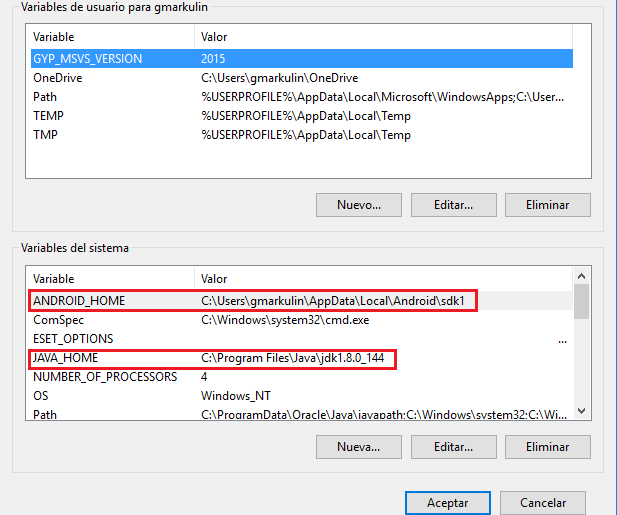
**Run:** *mongod --storageEngine=mmapv1*

**Set the Environment Variables**

1. Go to the **Control Panel** / **System Properties**.
2. Click on the **Enviroment Variables** button.
3. Create the following variables:

* **ANDROID\_HOME**
* **JAVA\_HOME**

Fill them with the paths as appropriate.

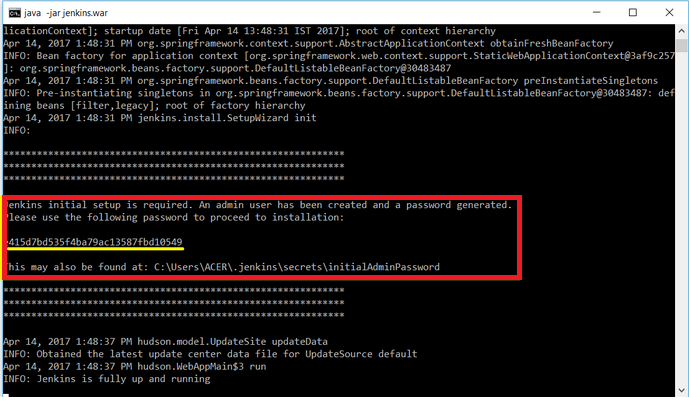


**Jenkins**

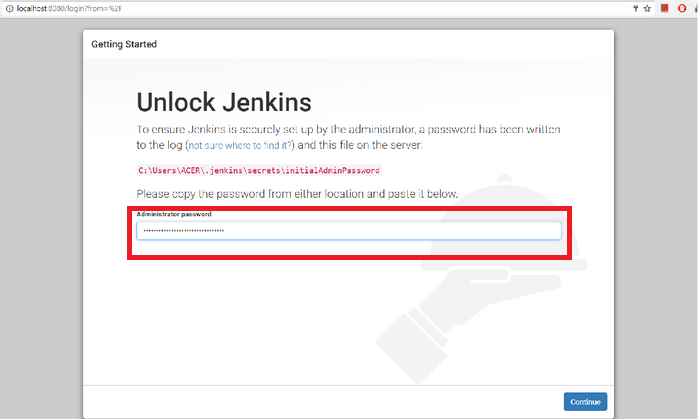
1. **Install Jenkins**
2. Download the war file **(**https://updates.jenkins-ci.org/download/war/)
3. Open a terminal and run the following command:

**java –jar Jenkins.war**

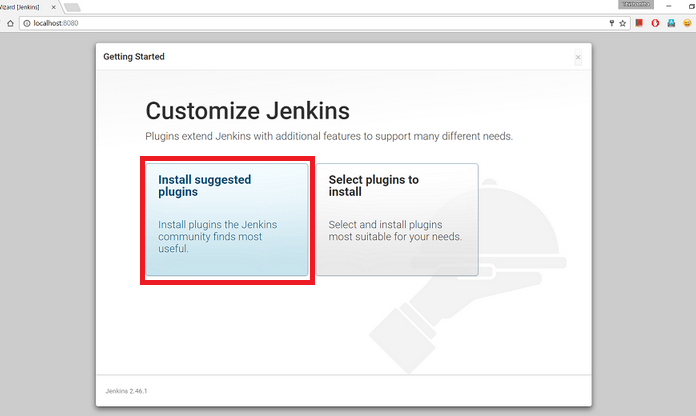
1. The installation will start and the setup will generate a randown password as you see in the screenshot (save password).

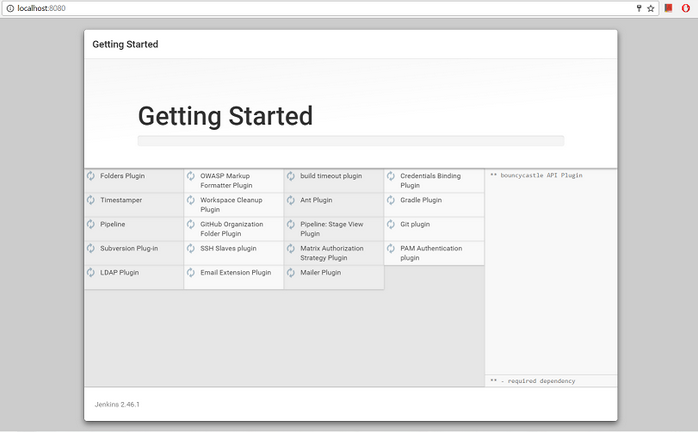


1. Go to your browser and type the following url: **[http://localhost:8080/](http://localhost:8080/login?from=%2F" \t "_blank)**
2. Complete **Administrator password** field with the password of the point c.

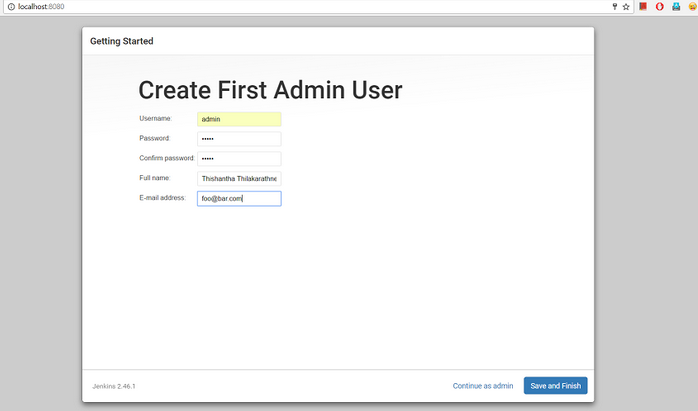


1. In the **Customize Jenkins** Windows click on the **Install suggested plugins** option.

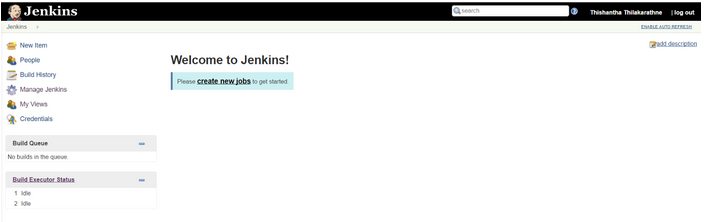




1. Once finished the installation of the plugins. Complete the fields to create the new user.



1. Click on the **Save and Finish** button.
2. Jenkins will shows the Dashboard where you can start creating jobs for your projects.

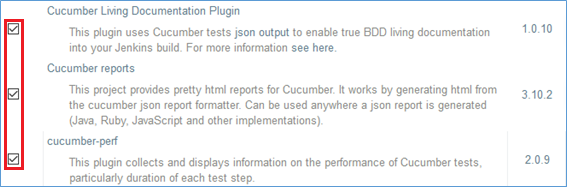


1. **Install the necessary plugins**
2. Go to the Plugin Manager through **Jenkins** / **Manage–Jenkins / Manage Plugins**

and select the **Available** Tab. In the Filter field search the following plugins:

* [Cucumber Living Documentation Plugin](https://wiki.jenkins-ci.org/display/JENKINS/Cucumber+Living+Documentation+Plugin)
* [Cucumber reports](https://github.com/jenkinsci/cucumber-reports-plugin)
* [cucumber-perf](http://wiki.jenkins-ci.org/display/JENKINS/Cucumber+Performance+Reports+Plugin)

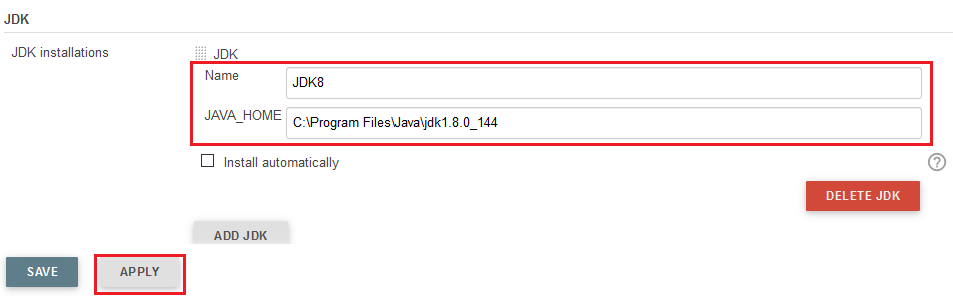
After that select your checkboxes.



1. Select the **Download Now And Install After Restart** button or **The Install Without Restart** button.

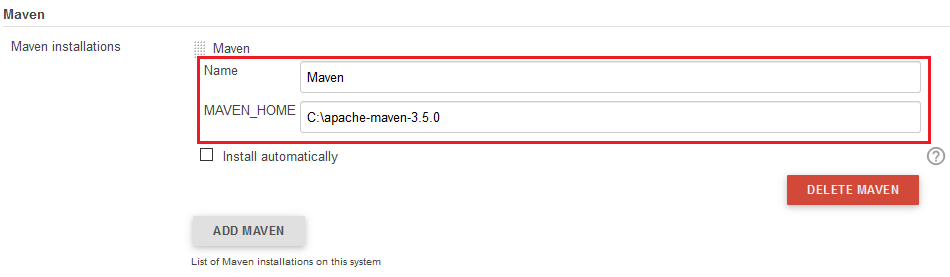


1. **Set the Global Tool Configuration**
2. **Define the JDK**



Fill the **Name** Field and complete the **JAVA\_HOME** with the URL of the installed JDK.

1. **Define the Maven**

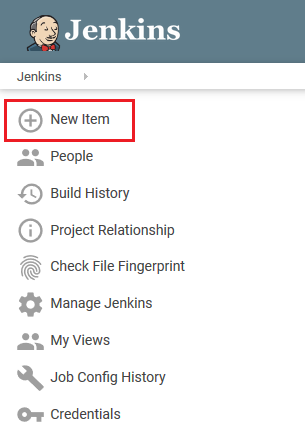


Fill the **Name** field and complete the **MAVEN\_HOME** with the path of the installed apache maven.

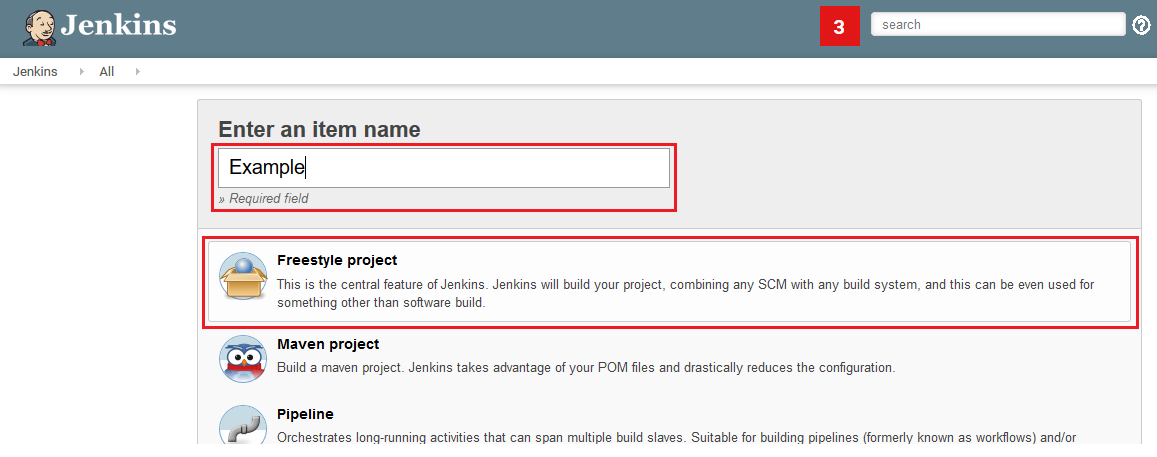
1. Click on the **Save** button.



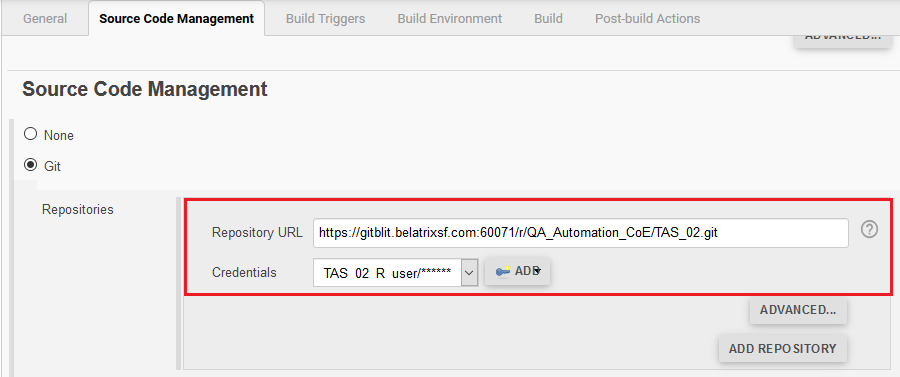
1. **Create a new job**
2. Click on the New Item option as you see in the screenshots.



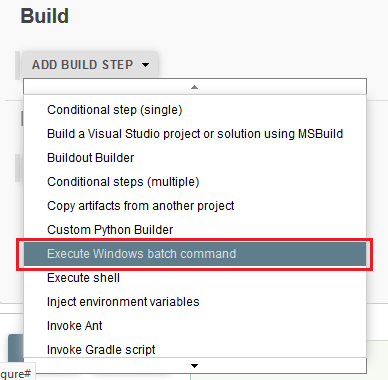
1. Complete an item name of the Job and select Freestyle project.



1. Click on the **OK** button.
2. **Configure the created job.**
3. Go to **Source Code Management** section and select Git Repositories. After that fill the **Repository URL** and **Credentials** fields.

****

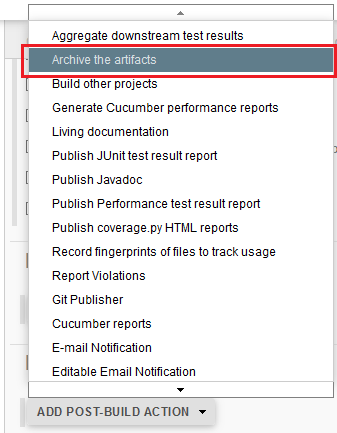
1. Go to **Build** section and click on the **Add Build Step** dropdown. Select the **Execute Windows batch command** item.



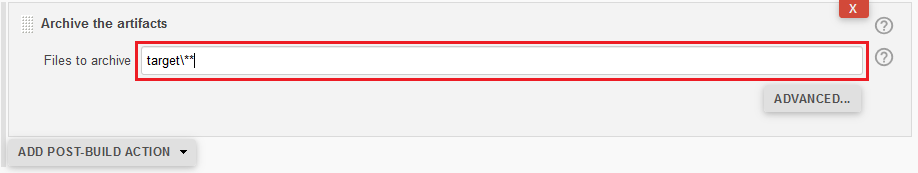
In the command field, complete the maven with the cucumber options as you see in the screenshot.



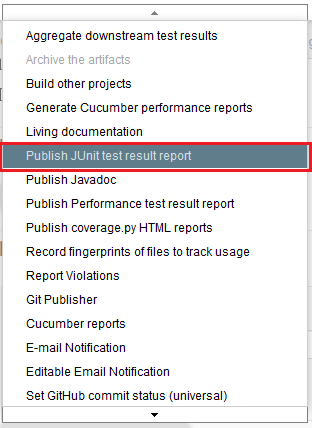
1. Go to the **Post-build-Actions** section and click on the **Add Post-Build-Actions** dropdown. After that select the **Archive the artifacts** item.



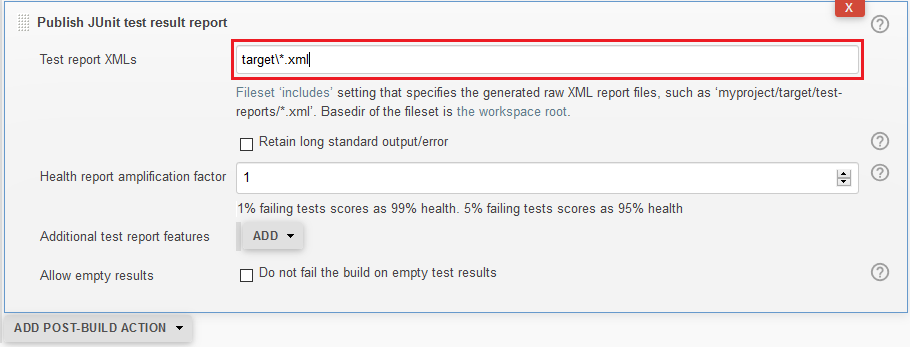
In the **Files to archive** field complete with the folder where you wish save the generated files. These will be used to generate the reports.

****

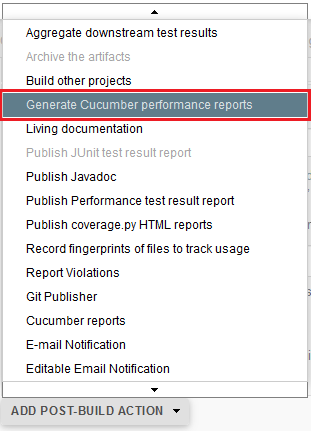
1. In the **Add Post-Build-Actions** dropdown select the **Publish Junit test result report** item.



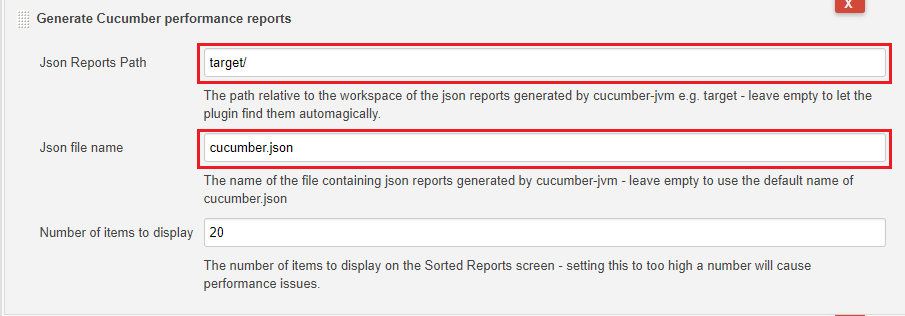
Complete the **Test report XMLs** field with the path appropriate.



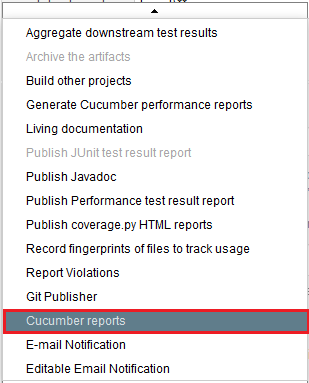
1. In the **Add Post-Build-Actions** dropdown select the **Generate Cucumber performance reports** item.



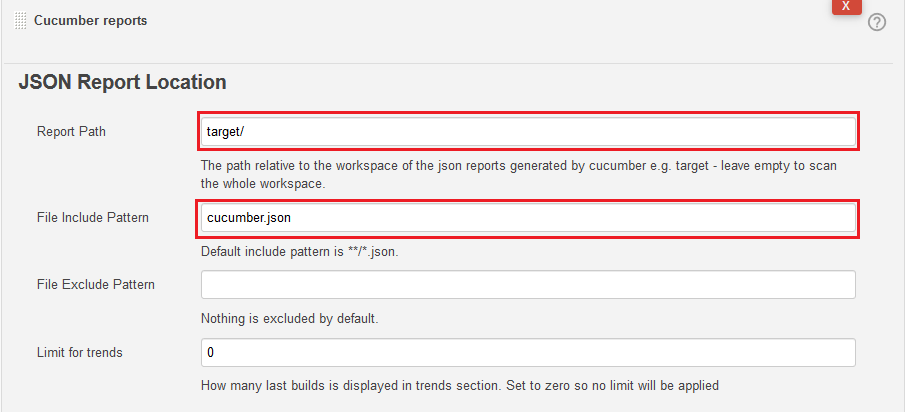
Complete the **Json Reports Path** field with the folder and the **Json file name** with cucumber.json



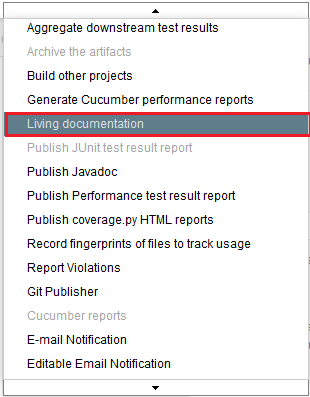
1. In the **Add Post-Build-Actions** dropdown select the **Cucumber reports** item.



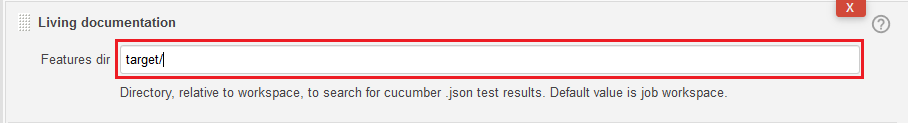
Complete the **Report Path** field with the folder appropriate and **the File Include Pattern** with cucumber.json



1. In the **Add Post-Build-Actions** dropdown select the **Living documentation** item.

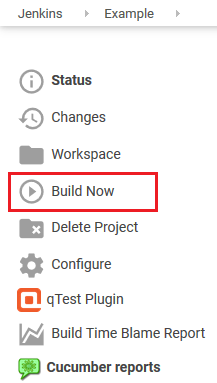


Fill the Features dir with the folder appropriateas you see in the screenshot.



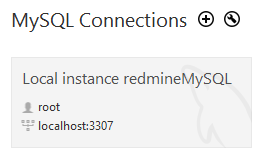
1. **Execute the Job**

In the Jenkins section click on the **Build Now** option.

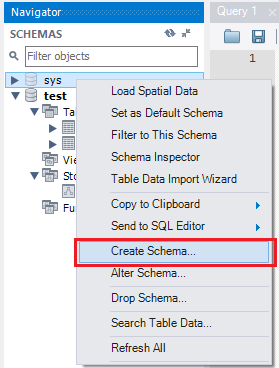


**Mysql Workbench– Configuration**

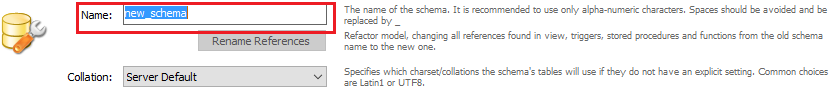
1. Open the application.
2. Go to MySQL Connections and click on the Local instance (use root user and port by default).



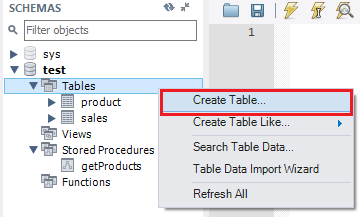
1. Go to the **Schema** section and do right click on sys schema.
2. In the contextual menu click on the **Create Schema…** option



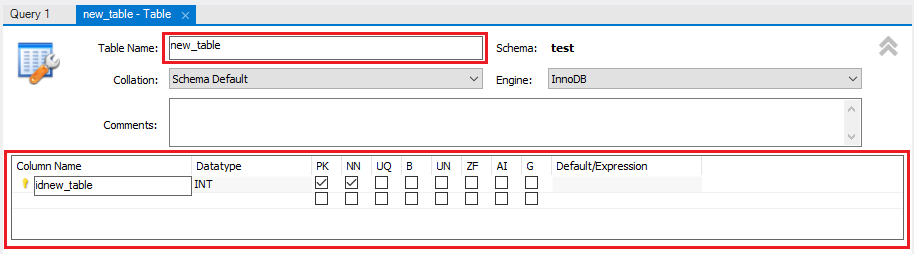
1. Complete the name of the schema in the **Name** field(In this case use **test**).



1. Expand the created **test** schema and do right click on the **Tables** option. After that select the **Create Table…** option.



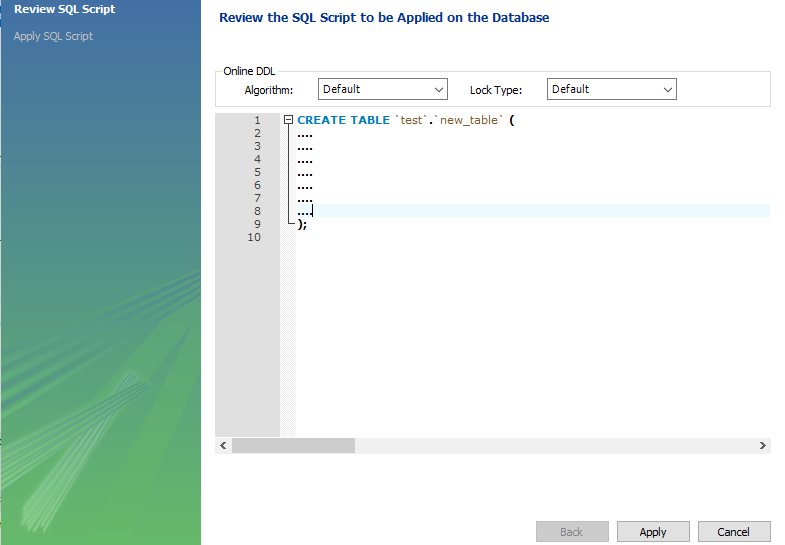
1. Complete the **Table Name** field with the value: **product**.



1. Create the columns for the table according to the following list:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Datatype** | **PK** | **NN** | **Defaul/Expression** |
| id | int | yes | yes |  |
| name | varchar | no | no | NULL |
| price | int | no | no | NULL |
| stock | Int | no | no | NULL |
| length | Int | no | no | NULL |
| width | Int | no | no | NULL |
| height | Int | no | no | NULL |

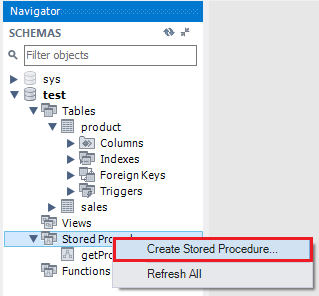
1. Click on the **Apply** button



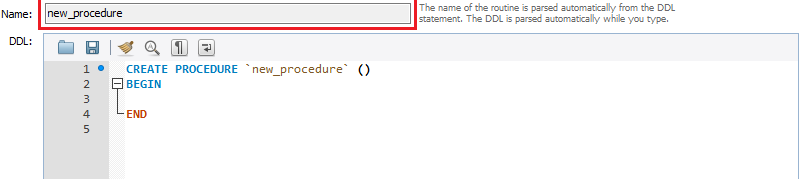
1. Click on the **Apply** button again.
2. Create another table called sales with the following columns:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Datatype** | **PK** | **NN** | **Defaul/Expression** |
| id | int | yes | yes |  |
| amount | int | no | no | NULL |

1. On the **Stored Procedures** option do right click and select the **Create Stored Procedure…** item.



1. Complete the **Name** field with the value: getProducts.

****

1. Edit the store procedure and put the following sentences:

CREATE DEFINER=`root`@`localhost` PROCEDURE `getProducts`()

BEGIN INSERT INTO `test`.product (id, name, price, stock, length, width, height) VALUES (2, "radio", 30, 200, 10,20, 18); END

1. Click on the Apply button twice times.

**Rest – Source Configuration**

The **Given Setup “Rest” Handler** Step has 2 sections:

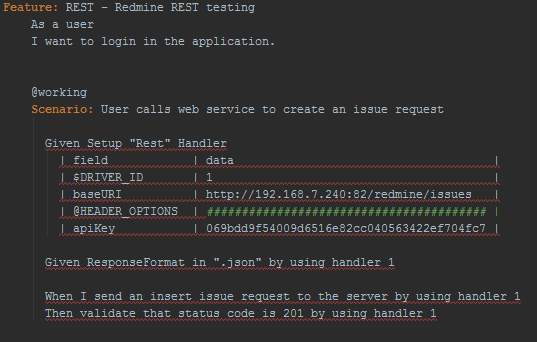
**First**:

* $**DRIVER\_ID**: is a number to identify the driver.
* **baseURI**: is the url of the endpoint to be tested.

**HEADER\_OPTIONS**:

* **apiKey**: is the key of the endpoint.

In this section you can included another options necessary for the header.

****

**Run the Feature/s for console**

1. Open a terminal.
2. Execute the command mvn:

mvn test -Dlog4j.debug -Dcucumber.options="--glue com\company\test\steps\rest src\test\java\com\company\test\features\rest"

**Selenium Grid - Set up Hub**

1. Open a terminal.
2. Go to the folder where you have the selenium server standalone file.
3. Execute the following command:

java -jar selenium-server-standalone-3.6.0.jar -role hub -hubConfig hub\_config.json

**Selenium Grid - Set up Web Node**

1. Open a terminal.
2. Go to the folder where you have the selenium server standalone file.
3. Execute the following command:

java -jar selenium-server-standalone-3.6.0.jar -role node -nodeConfig web\_nodes\_win.json

**Selenium Grid - Set up Appium Node**

1. Open a terminal.
2. Go to the folder where you have the selenium server standalone file.
3. Execute the following command:

appium -p 4723 --nodeconfig appium\_nodes.json

**Selenium Grid – Open Console in browser**

1. Open a browser.
2. Write the following url:

http://127.0.0.1:4444/grid/console?config=true&configDebug=true

****

**Selenium Grid – Open Console**

1. Open a terminal.
2. Go to Jenkins folder where you have Jenkins.war file.
3. Run the following command:

Java –jar Jenkins.war