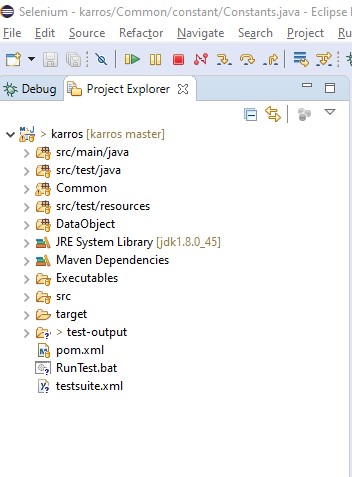
Karros Selenium Framework (FW)

1. **Description**

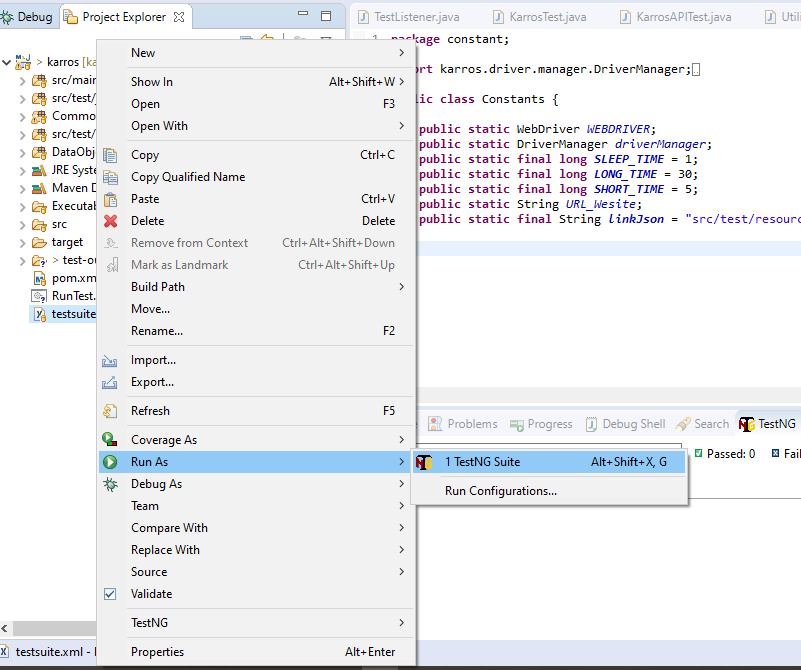
This is the demo framework which base on the Karros – QA Test Assignment. This FW can run automation on UI and API of the Karros demo.

1. **Setup environment**
2. Setup Eclipse.
3. Setup Java JDK (this project is using 1.8.045 version.)
4. Setup plugins Maven on Eclipse.
5. Setup TestNG on Eclipse.
6. Get the latest version chrome (version 79).
7. **Run Karros Project**
8. **Using Eclipse**

* You can get the Karros FW project at location https://github.com/nhatlong8905/karrosproject.git
* Add Karros project as the picture below



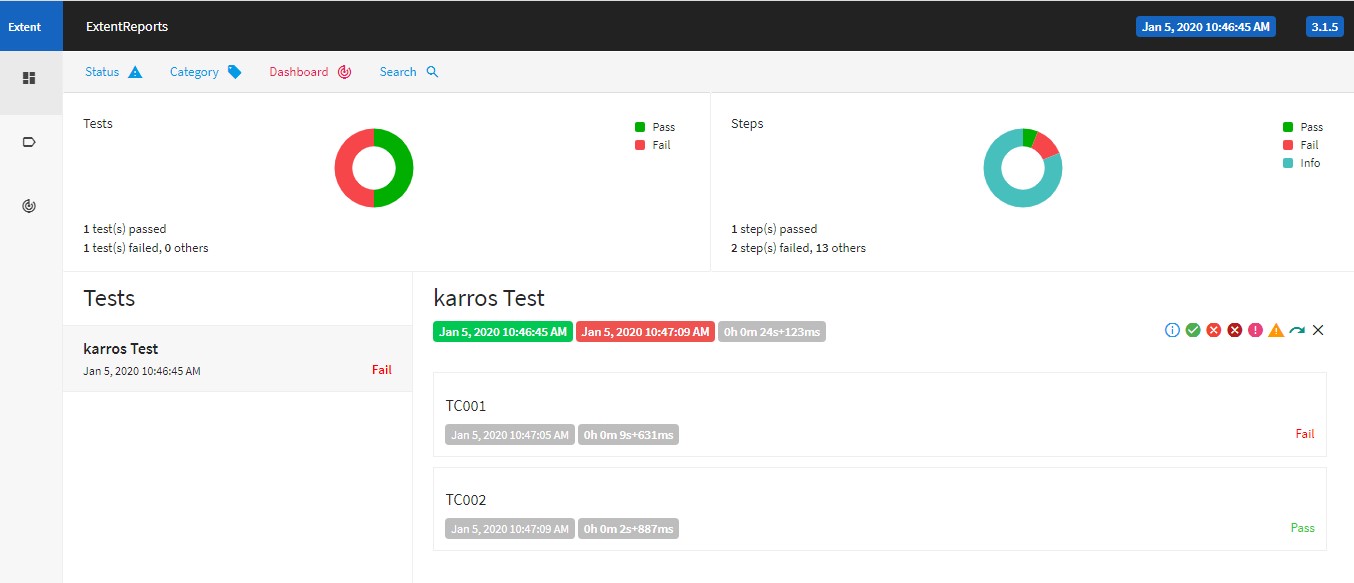
* Right click on testsuite.xml and run as this image below



* You get the test result in the folder

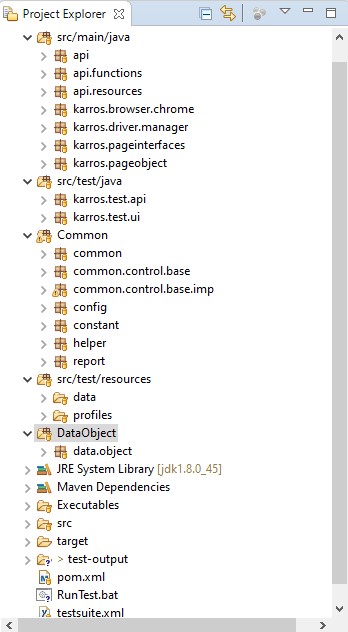
*..\ karros\test-output\ Report + ‘day’ - + ‘time’ (ex: Report 01.05.2020 - 10.46.45) \ ReportResults.html*

The result as below



1. **Using .bat file**
2. **Structure FW**

Here is the structure of Karros Framework



* **src/main/java**: The folder consists the main packet for the automation on application.

+ **api**: This packet will initialize the API which the project will be test.

+ **api.resources**: This packet handle for using methods GET/DELETE/POST/PUT… on API.

+ **api.functions**: The Project creates keyword for testing API which base on api.resources packet.

+ **karros.driver.manager**: This packet manages many browsers for running automation on application.

+ **karros.driver.chrome**: Chrome will be initialize at this packet.

+ **karros.pageinterfaces**: This packet handles the locators of pages.

+ **karros.pageobject**: The page on each screen will be handle as an object.

* **src/test/java**: Test case of project will be saved this folder.

+ **karros.test.api**: Test script about API testing.

+ **karros.test.ui**: Test script about UI testing.

* **Common**: this folder holds packets which will support for the automation on website.

+ **common**: include the functions about system, web element, time, … and they will be used many times in project.

+ **common.control.base**: the interface of control UI.

+ **common.control.base.imp**: Controls UI will implement and save at this packet.

+ **config**: Change the config of testNG report.

+ **constant**: variable common.

+ **helper**: Logger and properties mode on project.

+ **report**: custom report project (com.aventstack).

* **src/test/resources:** this folder holds the resources of project as data (json, excel…), profiles (properties, …).
* **Data Object:** the data of project will be packet as an object. Ex: Account, Customer, …
* **Executables**: the driver of browsers will be kept at this folder.

1. **Technique**

* Here are the techniques which FW is using

+ Page object model.

+ Design pattern: Factory.

+ Keyword driven testing.

+ Modular driven testing.

+ Data driven testing.

+ UI type layer.