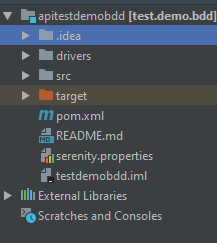
# **API Automation Framework Skeleton**

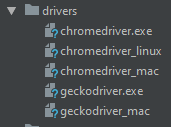
### API Automation skeleton structure:



### Structure contains the following folders at base level:

* **drivers**
* **src**
* test
* **Test**
* java
* resources
* **Java**
* helpers
* model
* pages
* services
* steps
* utilities
* Runtests.java
* **resources**
* features
* . conf
* **features**
* api
* web

**drivers** contain all the drivers which are required at the run time.

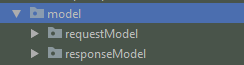


**helpers** contain all the helpers required.



* ApiHelper.java contains different functions which return different base api urls.

**model**contains all the models which can be used in API as request model, response model or in web automation to store dynamic data if needed.



* **pages**contain all the page object(locators) and their relative functions.
* **services**contain functions which are used to hit different end points from different micro services and services.

**Steps**contain all the definitions of the steps which are specified in the feature file for both web and API.

Configloader.java contain function to load the url from the env.conf file.

**CsvReader**

* CSV Reader.java contain function to read the data from the csv files.

**MongoDbConnection**

* MongoDbConnection.java contain functions to connect/disconnect to the mongoDb and get the data which further used in the api automation scripts assertions.

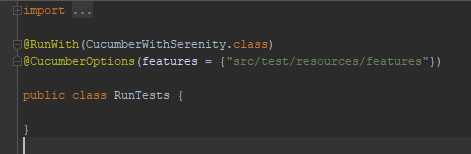
**PropertyFileOperations**

* PropertyFileOperations.java contains a function to handle the data from the properties files which is further used as input in the api automation scripts.

**RandomGenerator**

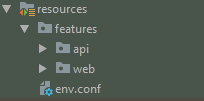
* RandomGenerator.java contains functions to handle the random values which are further used as input in the api automation scripts.

**Runtests.java**

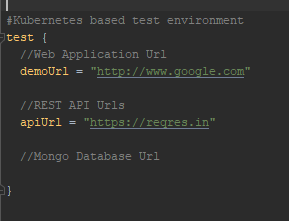
****

Runtests.java is also called a runnable file as it contains the path which the user needs to execute.

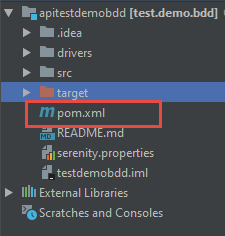
**resources**



**resources** contain the api/web .feature files and the env.conf file.

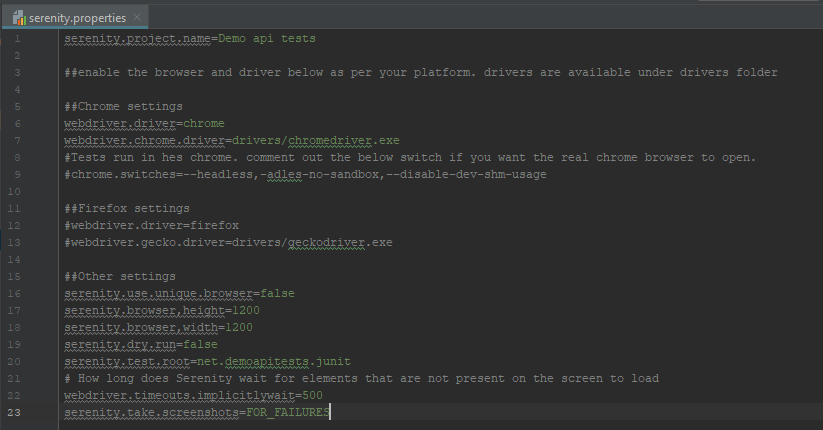


**pom.xml**



pom.xml file contains information of project and configuration information for the maven to build the project such as dependencies, build directory, source directory, test source directory, plugin, goals etc. Maven reads the pom.xml file, then executes the goal.

### serenity.properties

****

serenity.property file contain all the serenity system properties and configurations.

# How to run?

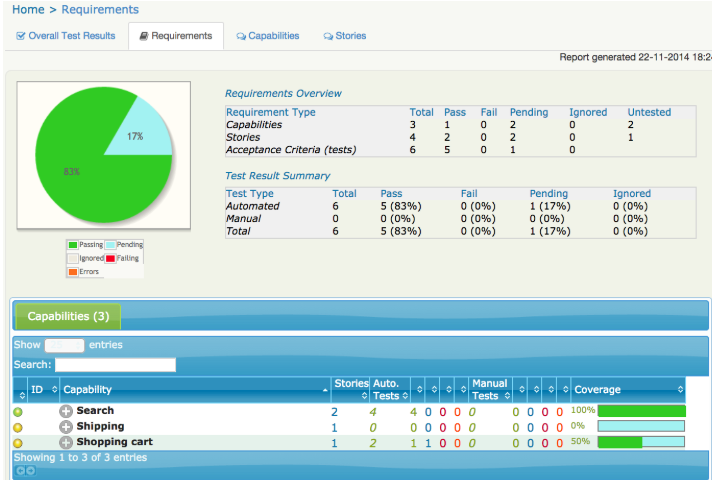
We can directly run scripts by the command: **mvn clean verify -Dcucumber.options="src/test/resources/features"**

**For more details click on**[**Link**](http://thucydides.info/docs/serenity-staging/#_serenity_system_properties_and_configuration)

**target**

After successful execution reports can be found in the target folder.

Path to get the reports: **target/site/serenity/index.html**

****

**Framework Link:**