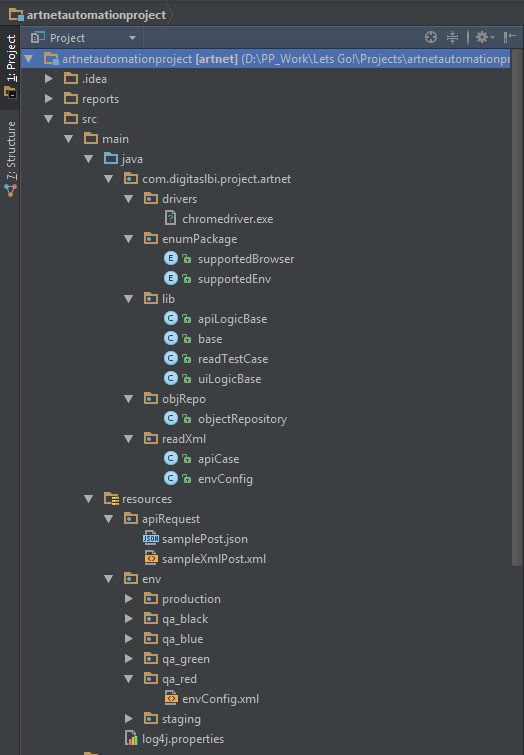


Prachi Pawaskar

ARTNET AUTOMATION PROJECT GUIDE

# **Project structure:**



# 1. Reports:

* Dir location: artnetautomationproject\reports
* File format: html
* New report is created after every run.
* Naming convention used for reports: “detailTestReport\_[timestamp].html”
* Image is added to the report for a failed step, these images are locally stored under: “artnetautomationproject\reports\reportImages”.
* As the failed step images are locally stored emailing the report excludes the images from report.
* **Note: Do not commit reports or report images to the repo.**

# 2. Source Code:

* Dir location: artnetautomationproject\src\main\**java**\
* This directory contains the source code for the framework

## 2.1 Drivers:

* Dir location: \src\main\java\com\digitaslbi\project\artnet\drivers
* Holds “chrome driver” exe file to support Chrome browser.

## 2.2 Enum Packages:

* Dir location: \src\main\java\com\digitaslbi\project\artnet\enumPackage
* Holds three enum class to define supported Browser, Environments and Tags by framework.
* Files:
  + supportedBrowser.java – Supported Browser.
  + supportedEnv.java – Supported environments.
  + supportedTag.java - Supported tags.

## 2.3 Lib:

* Dir location: \src\main\java\com\digitaslbi\project\artnet\lib
* Following are the class files and its function:
  + base.java:
    - Kicks off creation of report for the run.
    - Reads env files to set up the configs like run environment, browser, tag and default URLs and credentials.
    - Holds data provider method to read active test scripts that is to be executed.
  + apiLogicBase.java:
    - Holds method to support API testing.
    - Contains method to disable SSL certification check.
    - Contains doGet method to support GET api calls.
    - Contains doPost method to support POST api calls.
    - Jersey client used for REST calls.
  + uiLogicBase.java
    - Holds method to start and stop driver.
  + readTestCase.java
    - Holds method to read API and UI test scripts.
    - Parses xml test case and performs actions.

## 2.4 objRepo:

* Dir location: \ src\main\java\com\digitaslbi\project\artnet\objRepo
* Holds objectRepository.java class file.
* Class file contains all the object repositories for web driver in variable format.

## 2.5 readXml:

* Dir location: \src\main\java\com\digitaslbi\project\artnet\readXml
* Holds two class files:
  + apiCase.java – Read api test scripts tags.
  + envConfig.java – Read envConfig.xml tags for environment variables.

# 3. Resources:

* Dir location: artnetautomationproject\src\main\resources
* This dir contains resource files.

## 3.1 apiRequest:

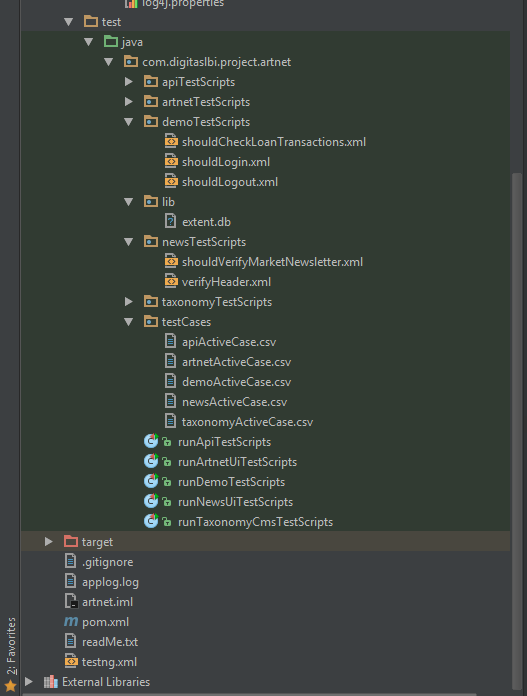
* Dir location: \src\main\resources\apiRequest
* Holds POST api request files.
* Can be in either json or xml format.

## 3.2 env:

* Dir location: \src\main\resources\env
* Holds envConfig files: \env\[envName]\envConfig.xml

# 4. Logger file:

* “log4j.properties”
* Do not touch unless necessary.



# 5. Test:

* Dir location: artnetautomationproject\src\**test**\
* Holds all the test scripts to be executed.

## 5.1 Test scripts:

* Dir location: src\test\java\com\digitaslbi\project\artnet\[TestScripts\_package]
* Holds all API and UI test scripts in xml format.

## 5.2 Test cases:

* Dir location: \src\test\java\com\digitaslbi\project\artnet\testCases
* Holds a csv file for each module eg: apiActiveCase.csv
* CSV file contains [active/inactive],[bat/sanity/negative/regression],testscriptName
* Only test scripts marked as active under these will be executed for the module.

## 5.3 Run class:

* Dir location: \src\test\java\com\digitaslbi\project\artnet\
* Naming convention: run[Module\_name]TestScripts.java
* This class is defined in testing.xml and is responsible to trigger execution for test scripts.
* Each class has a run method, with naming convention as “[module]TestCase”

# 6. app.log

* Dir location: artnetautomationproject\applog.log
* Log file created after execution.
* Only one file is maintained, hence old logs are replaced with new ones.

# 7. readMe.txt

* Dir location: artnetautomationproject\
* Holds information about project.
* Gives execution directions.

# 8. pom.xml

* Dir location: artnetautomationproject\
* Holds all dependencies.
* DO NOT touch unless necessary.

# 9. testng.xml

* Dir location: artnetautomationproject\
* Defines all the run classes which needs to be executed.
* Also defines env and browser to use in execution.

# **Adding new module:**

Following steps are to be followed to add new module eg: “newModule”:

1. Create a new test script package [refer 5.1 Test scripts].

Eg: src\test\java\com\digitaslbi\project\artnet\newModuleTestScripts

1. Add test scripts in xml format under the new test script package.

Eg: src\test\java\com\digitaslbi\project\artnet\newModuleTestScripts\sampleTest.xml

1. Add new test case csv file [refer 5.2 Test cases]

Eg: src\test\java\com\digitaslbi\project\artnet\testCases\newModuleActiveCase.csv

1. Add the test script name to the csv and mark as active with tag name, no space or file extension to be added.

Eg: active, regression,sampleTest

1. Create a new run class [refer 5.3 Run class]

Eg: \src\test\java\com\digitaslbi\project\artnet\runNewModuleTestScripts.java

1. Copy/paste other run class content and change the method name.

Eg: newModuleTestCase

1. Add the data provider declaration in to the base class [refer 2.3 Lib >> base.java] under “//Choose csv to run test scripts” , line #154, section in the following format:

else if (type == "[run class method name]") {temp = "csv file name";}

Eg: else if (type == " newModuleTestCase ") {temp = " newModuleActiveCase ";}

1. Add the run class declaration to testng.xml [refer 9. testng.xml]

Eg: <class name="com.digitaslbi.project.artnet.runNewModuleTestScripts " />

1. Run testng.xml and check reports [refer 1. Reports] and logs [refer 6. app.log] for execution results.

# **Adding new test script to existing module:**

Following steps are to be followed to add new test script eg: “newModuleScript.xml”:

1. Add test script in xml format under the respective test script package.

Eg: src\test\java\com\digitaslbi\project\artnet\newModuleTestScripts\newModuleScript.xml

1. Add the test script name to the csv and mark as active with tag name no space or file extension to be added.

Eg: active, regression,newModuleScript

1. Run respective run class testng.xml and check reports [refer 1. Reports] and logs [refer 6. app.log] for execution results.

Eg: <class name="com.digitaslbi.project.artnet.runNewModuleTestScripts " />