**Pet Store API Automation using Rest Assured**

This automation suite is developed to automate the APIs mentioned in the following swagger file <https://petstore.swagger.io/>.

**How to get the code?**

1. The code is uploaded in the following GIT repository. User can clone the code and run it using java IDE like eclipse, intellij, etc.

<https://github.com/tpjohnmarshal/AutomationAssignment>

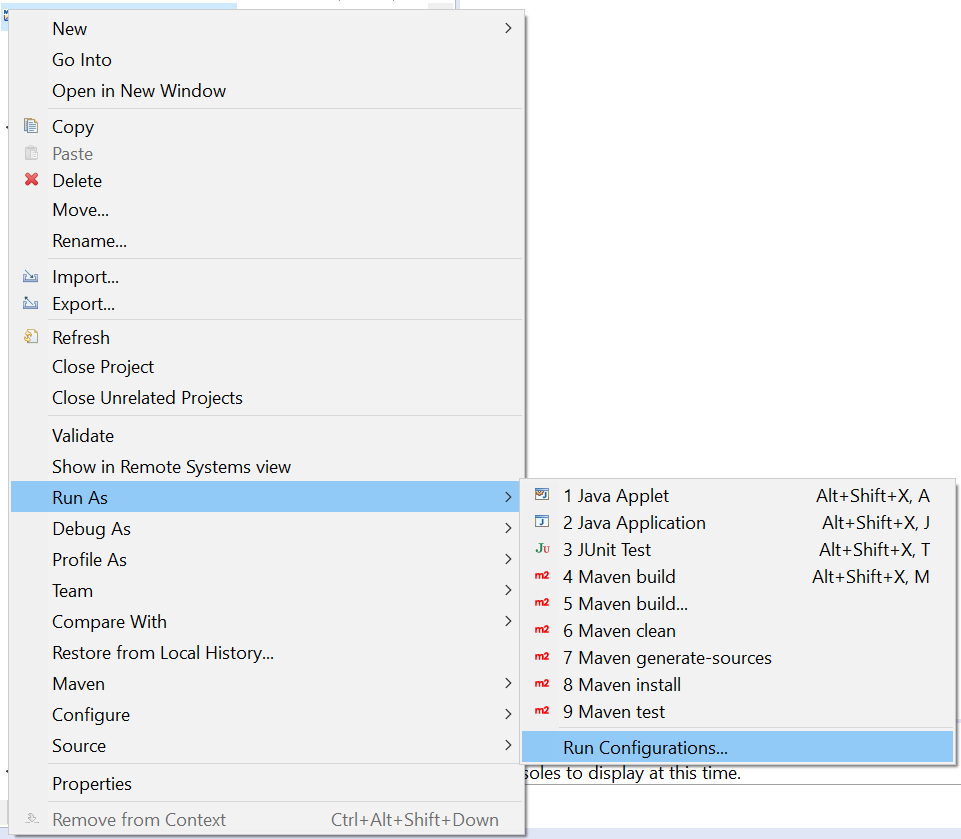
1. User can import the petStoreApiAutomation.zip file into the IDE and run.

**What is the Framework Structure?**

1. All the required dependencies are provided in the pom.xml.
2. The src/test folder will contain the code to automate the test cases.
3. configs folder contains the required configurations for Logger and the automation framework.
4. Logs generated are stored in the file present inside the logs folder.
5. resources folder contains the required feature file and json resources.
6. Src/test/java directory will contain the Test Runner class with the required cucumber options.
7. Pojo classes created to handle the Pet object smoothly.
8. Context technique is used to avoid creating of instances and objects unnecessarily.
9. Step definitions required to execute the BDD test steps are written and placed inside the StepDefinitions directory.
10. To handle the api calls and response the required code and configurations are coded inside the apiEngine directory.
11. SureFire utility is used to generate the test rest reports.

**How to execute the code?**

1. Import the project into an IDE.
2. Verify all required dependencies(.jar files) are downloaded in the .m2 folder
3. Verify that NO compile errors in the code.
4. This framework will execute all the test cases since we are not having any specific tags. If you wish to execute specific test cases, then kindly add a tag (ex: @name) in the respective testcase in feature file and provide the same tag in the cucumber options and run the framework.
5. In order to execute the automation right click on the project and navigate to Run As -> Run Configurations.



1. Create a new Maven Build configuration with goal “**clean install test surefire-report:report site -DgenerateReports=false**” and run the framework.
2. Once the build is completed the test result reports are generated inside the target directory.

**How to add a new test case?**

1. Open the respective feature file and add a new scenario.
2. Add the required step definitions in the StepDefinitions directory.
3. Compile the code and run.