**Ex. 1 SWAGGER API- Pet Store**

**Execution instructions:**

Project can be run from runner class.

Project can run from testing.xml file.

Project can be run from command line.

**Approach to be implemented at Actual scenario**

All the points provided below can be implemented considering the timeline of the task but in the real scenario these are important and should be the part of the approach.

*Authorization check:* real time will check the authenticity of API and the authorisation key to consume and interact. It will provide security to data.

*Roles & access:*  real environment will define roles for each user based on authentication. Different access be given to each users and based on that tests approach will be performed.

*Method testing:* All the REST methods require testing if they return a proper structured response and code.

*Signature verification:* Important step that will be taken care of in real time about the header which is coming from client and it will be verified by signature verification.

*Throttling & Burst testing:* Limit and load checks like user hitting per second will be carried out initially to know the limits of the API.

*Masker:* If dealing with sensitive data we will use masker to identify critical and sensitive data from the client/user to mask it.

*Header check:* Mandatory header check will be done for each methods.

*Database:*  Very important step in API testing. DB Connection will be established to verify each data with DB.

*Detailed Logging :* In real time we will use more loggings. Logs containing each transaction will help in identifying the records for which the system behaved weirdly

*Reporting:* Important point to know how the system is performing in the real world. It helps in knowing the efficiency and accuracy of the system in terms of percentage. Furthermore, reporting helps in checking whether the goals are met. *Extent reporting* can be used and can be easily integrated with selenium. Or any similar reporting tool.

*Test scenarios:* More negative test scenarios and cases can be used to do both positive n negative testing for more coverage. Use of real data and variety is must

*Validation and checks:*  Detailed validation and checks are provided already but in real scenario are required more and at many places related to the data and process.

*Exceptions:* Detailed exception handling related to business and application can be used to have more robustness and knowing the system.

*Environment duplication:* Separate environment for development, QA, SIT UAT

*Code review:* peer review with the subordinates for codes, testscripts and data.

*Version control(Jenkin):*  Important step to have, we will do version control in real scenario to achieve automated CI/CD approach and ease deployment.