**API Script Approach & Execution**

I have used open source JMeter tool for script design and for execution purpose and I have followed below steps.

**Tool Setup: -**

As its open source we can download it from any web browser and use it – we just need Java as prerequisite and I already had it in my system.

**Test Plan Setup: -**

Added via templates > recording > then all the required thread group, header manager and listeners will be added automatically no need to add all components separately

**OR**

We can add all components individual using Test Plan > Thread Group > HTTP Requests > Listeners (whatever required)

**Script Design: -**

* Added HTTP request
* Added CategoryId as a Parameter File ( Via CSV data set config) – Use csv path if the jmx file is not in the same folder
* Done Correlation using boundry and JSON extractor (Variable name startes from Cor\_)
* Assertion check points

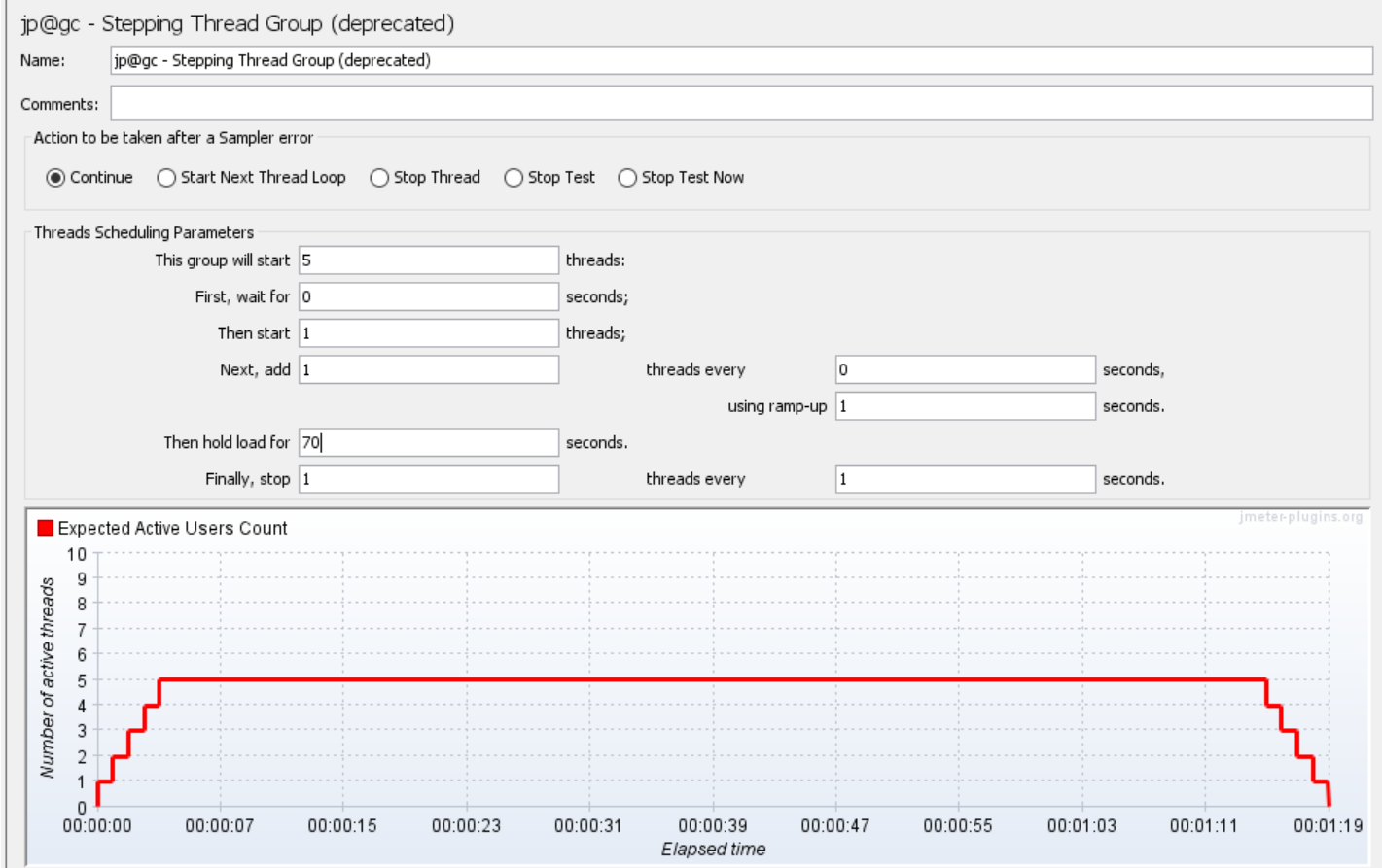
**Test Assertion:**

* + Check for the Response Status - **Added response status 200 code as assertion**
  + On a successful response, validate the response with below criteria:
    - Parameter Check: Category ID –**Written a code in BeanSheel assertion** **Validating ParamCategoryId == Response CategoryId – if its equal log as true or if it not log as false**
    - Text Check: "CanRelist": true – **Added response assertion = CanRelist": true**
* Beanshell script for witing into a file
  + Print following values in a csv file: Category ID, Name, Path, Promotion ID, Price
    - **Correlated all the above ID’s mentioned and written BeanShell Post Processer to write into a CSV file**
    - **Added Once only Controller to print the heading for 1st time**
  + Please print all Promotion IDs and respective Prices per Category ID
    - **Correlated Promoion ID’s/Price using JSON extractor and written file opetion funciton code in Beanshell and added forloop.**
    - **Given filepath to store/write the extracated values autoamtically**

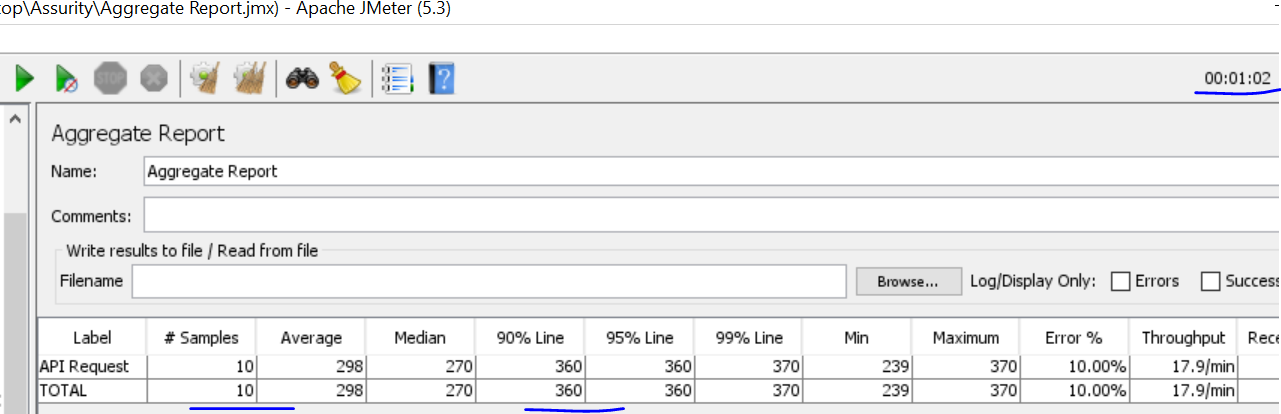
**Non-Functional Requirements**

|  |  |
| --- | --- |
| **NFR #** | **Description** |
| NFR-01 | Test should support Vusers (Threads) half the count of Category IDs shared in Test Data – Added Test data and parameterized in the script and I have added 5 threads half of the Category ID as mentioned |
| NFR-02 | The test should ramp up at one VUser (Thread) per second – We can design using no of thread groups. I have done two approaches here.   1. In normal thread group I added total Vuser as 5 and ramp up second as one and duration as 60sec 2. In Stepping Thread Group – added total 5 Vuser, ramp-up 1 user/sec and execute the test for 60 secs (steady state) and ramp- down 1user/sec (added 10 sec 1for ramup and down) – refer screenshot 1. |
| NFR-03 | Test should achieve 10 API calls in total for the 1-minute Steady State duration – We can archive this in multiple ways – I have followed two approaches and achieved in both   1. Precise Throughput Timer – We can just specify how much thorughtput we wanted to archive in XX duration – I have achieved using this timer 2. Workload model – I have calculated pacing using little’s law and added constant timer to achieve 10API hits in 1min. - I have achieved |
| NFR-04 | 90 percent of the times the API is expected to perform within 500 ms – All response time (Min , Avg , 90% and Max ) are less than ,500ms Please refer screenshot 2 below for the exact output |

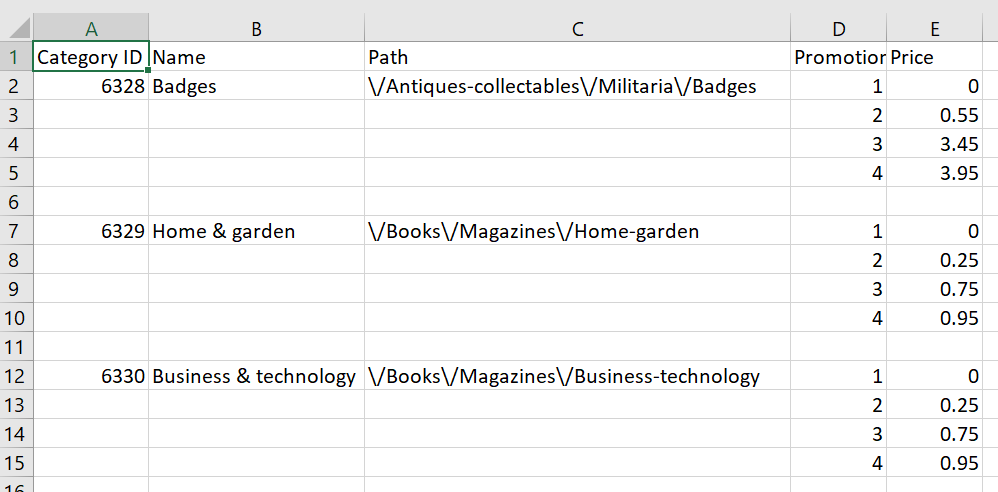
**Screenshot 1**



**Screenshot 2**



**Screenshot 3**



Performance Test Observation/Report.

* Could see all format of response time is under SLA (500ms)
* Per SLA – the Avg and 90 percentile response time is under 300sec
* Generated HTML report - using JMeter plugins
* Reports are saved to csv file, please refer screenshot 3 and also I will attach the files
* For server related observation, we need server details or third-party tools like
  + APPD
  + Azure Insights
  + AWS cloud watch