Jmeter Test Performance Report

1. Test Plan Configuration
    - Created a Test plan in Jmeter to test the Performance of the api https://httpbin.org/delay/{seconds_to_delay}
    - Added Thread Group, HTTP Request, Header Manager, Cookie Manager, Assertions, and Listeners.

2. Thread Group Settings
    - Number of Users(Threads):
    - Ramp Up Period:
    - Loop Count:

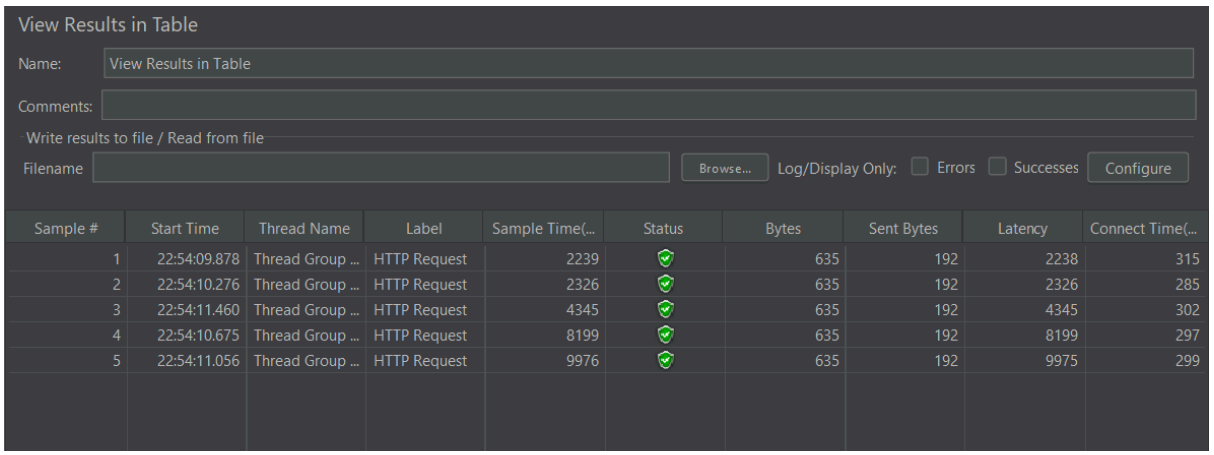
We defined each field with different values and check for the performance of the api.

3. HTTP Request Details.
    - Defined the headers, cookie and variables needed for the HTTP request.
    - Defined base url as a variable and used as **${baseurl}** in the request.
    - Defined a customised token and defined in the Header Manager.
    - Defined a variable to generate random number between 1-10.
    - Checked for Response Code 200, Response Time ≤ 20s, and JSON content validation.

Observed Result:

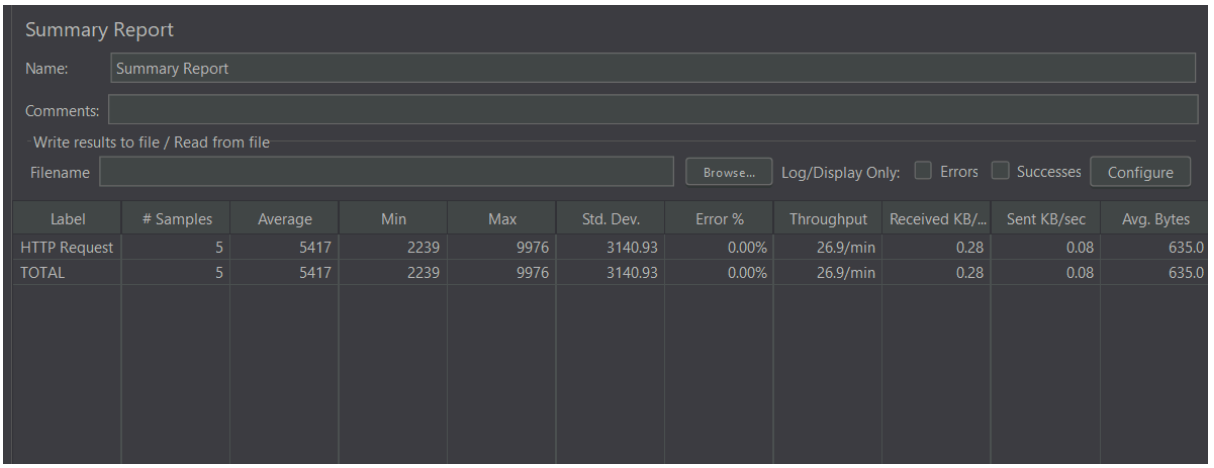
- Checked the Results of the Performance Test using Listeners like View Results Tree, View Results in Table, Aggregate Report , Summary Report.

1. Result when No. of users is 5 with a ramp up period of 2sec.



Sample #	Start Time	Thread Name	Label	Sample Time(...	Status	Bytes	Sent Bytes	Latency	Connect Time(...
1	22:54:09.878	Thread Group ...	HTTP Request	2239		635	192	2238	315
2	22:54:10.276	Thread Group ...	HTTP Request	2326		635	192	2326	285
3	22:54:11.460	Thread Group ...	HTTP Request	4345		635	192	4345	302
4	22:54:10.675	Thread Group ...	HTTP Request	8199		635	192	8199	297
5	22:54:11.056	Thread Group ...	HTTP Request	9976		635	192	9975	299

- The recorded response times range from 2239 ms (~2.2s) to 9976 ms (~9.9s), confirming that the delay mechanism works as expected.
- The green shield icons indicate successful responses (HTTP 200 OK) for all requests.
- Latency time is between 2238 ms - 9975 ms, consistent with the expected delay.
- Connection time is around ~300 ms, showing minimal network.
- Response size and Request Size are consistent all around.



Summary Report

Name: Summary Report

Comments:

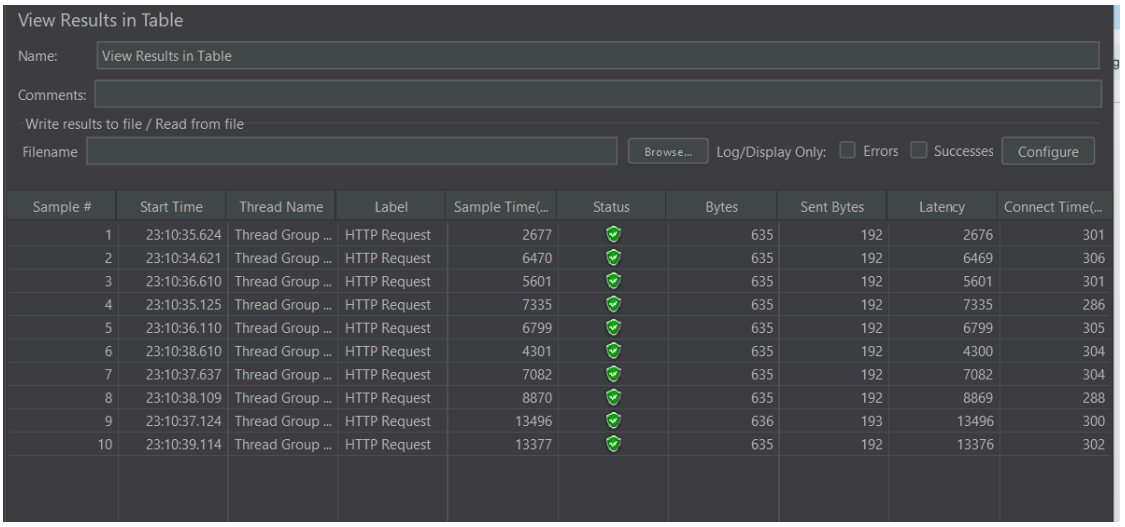
Write results to file / Read from file

Filename Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/...	Sent KB/sec	Avg. Bytes
HTTP Request	5	5417	2239	9976	3140.93	0.00%	26.9/min	0.28	0.08	635.0
TOTAL	5	5417	2239	9976	3140.93	0.00%	26.9/min	0.28	0.08	635.0

- A total of 5 API requests were executed.
- Throughput is low due to the API's forced delay (1-10 sec per request).
- No failed requests, indicating stable API performance.
- The API response time varies between 2-10 seconds, confirming the expected delay behavior.

2. Result when No. of users is 10 with a ramp up period of 5 sec.



View Results in Table

Name: View Results in Table

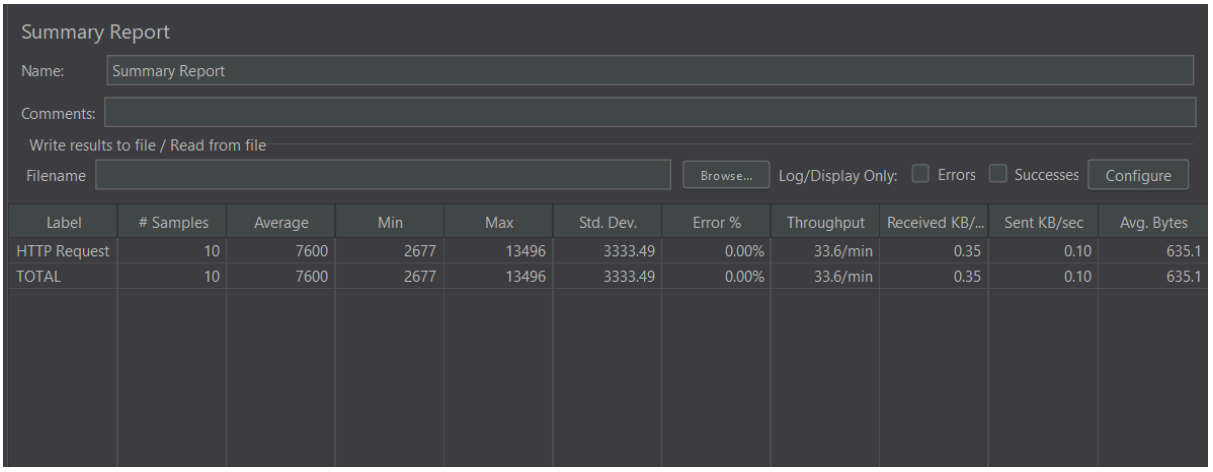
Comments:

Write results to file / Read from file

Filename Browse... Log/Display Only: Errors Successes Configure

Sample #	Start Time	Thread Name	Label	Sample Time(...	Status	Bytes	Sent Bytes	Latency	Connect Time(...
1	23:10:35.624	Thread Group ...	HTTP Request	2677		635	192	2676	301
2	23:10:34.621	Thread Group ...	HTTP Request	6470		635	192	6469	306
3	23:10:36.610	Thread Group ...	HTTP Request	5601		635	192	5601	301
4	23:10:35.125	Thread Group ...	HTTP Request	7335		635	192	7335	286
5	23:10:36.110	Thread Group ...	HTTP Request	6799		635	192	6799	305
6	23:10:38.610	Thread Group ...	HTTP Request	4301		635	192	4300	304
7	23:10:37.637	Thread Group ...	HTTP Request	7082		635	192	7082	304
8	23:10:38.109	Thread Group ...	HTTP Request	8870		635	192	8869	288
9	23:10:37.124	Thread Group ...	HTTP Request	13496		636	193	13496	300
10	23:10:39.114	Thread Group ...	HTTP Request	13377		635	192	13376	302

- The recorded response times range from 26677ms (~2.6s) to 13377ms (~13.37s), indicating a response with a higher delay.
- The green check marks in the "Status" column indicate that all requests were successfully completed without errors.
- Latency values closely match the sample times, suggesting that the processing overhead is minimal.
- The connection time varies between 286 ms to 306 ms, indicating a relatively stable network connection.



Summary Report

Name: Summary Report

Comments:

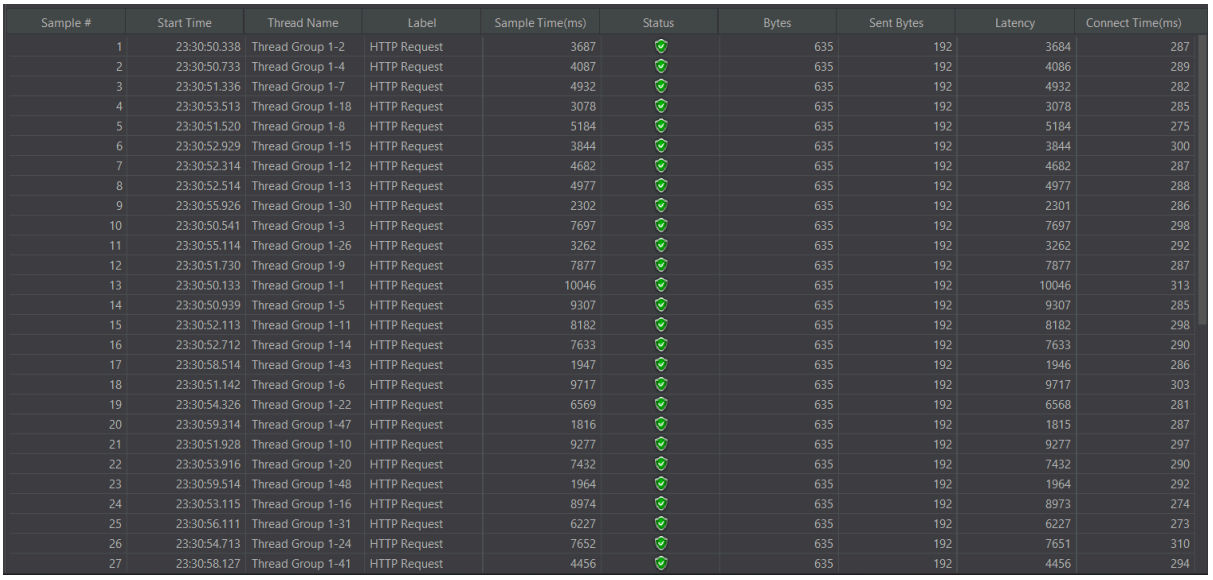
Write results to file / Read from file

Filename | Browse... | Log/Display Only: Errors Successes | Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/...	Sent KB/sec	Avg. Bytes
HTTP Request	10	7600	2677	13496	3333.49	0.00%	33.6/min	0.35	0.10	635.1
TOTAL	10	7600	2677	13496	3333.49	0.00%	33.6/min	0.35	0.10	635.1

- No failed requests, meaning all HTTP requests were successfully processed.
- 33.6 requests per minute , this means JMeter was able to send approximately 33.6 HTTP requests per minute.

3. Result when No. of users is 50 with a ramp up period of 10 sec.



Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	23:30:50.338	Thread Group 1-2	HTTP Request	3687		635	192	3684	287
2	23:30:50.733	Thread Group 1-4	HTTP Request	4087		635	192	4086	289
3	23:30:51.336	Thread Group 1-7	HTTP Request	4932		635	192	4932	282
4	23:30:53.513	Thread Group 1-18	HTTP Request	3078		635	192	3078	285
5	23:30:51.520	Thread Group 1-8	HTTP Request	5184		635	192	5184	275
6	23:30:52.929	Thread Group 1-15	HTTP Request	3844		635	192	3844	300
7	23:30:52.314	Thread Group 1-12	HTTP Request	4682		635	192	4682	287
8	23:30:52.514	Thread Group 1-13	HTTP Request	4977		635	192	4977	288
9	23:30:55.926	Thread Group 1-30	HTTP Request	2302		635	192	2301	286
10	23:30:50.541	Thread Group 1-3	HTTP Request	7697		635	192	7697	298
11	23:30:55.114	Thread Group 1-26	HTTP Request	3262		635	192	3262	292
12	23:30:51.730	Thread Group 1-9	HTTP Request	7877		635	192	7877	287
13	23:30:50.133	Thread Group 1-1	HTTP Request	10046		635	192	10046	313
14	23:30:50.939	Thread Group 1-5	HTTP Request	9307		635	192	9307	285
15	23:30:52.113	Thread Group 1-11	HTTP Request	8182		635	192	8182	298
16	23:30:52.712	Thread Group 1-14	HTTP Request	7633		635	192	7633	290
17	23:30:58.514	Thread Group 1-43	HTTP Request	1947		635	192	1946	286
18	23:30:51.142	Thread Group 1-6	HTTP Request	9717		635	192	9717	303
19	23:30:54.326	Thread Group 1-22	HTTP Request	6569		635	192	6568	281
20	23:30:59.314	Thread Group 1-47	HTTP Request	1816		635	192	1815	287
21	23:30:51.928	Thread Group 1-10	HTTP Request	9277		635	192	9277	297
22	23:30:53.916	Thread Group 1-20	HTTP Request	7432		635	192	7432	290
23	23:30:59.514	Thread Group 1-48	HTTP Request	1964		635	192	1964	292
24	23:30:53.115	Thread Group 1-16	HTTP Request	8974		635	192	8973	274
25	23:30:56.111	Thread Group 1-31	HTTP Request	6227		635	192	6227	273
26	23:30:54.713	Thread Group 1-24	HTTP Request	7652		635	192	7651	310
27	23:30:58.127	Thread Group 1-41	HTTP Request	4456		635	192	4456	294

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
24	23:30:53.115	Thread Group 1-16	HTTP Request	8974		635	192	8973	274
25	23:30:56.111	Thread Group 1-31	HTTP Request	6227		635	192	6227	273
26	23:30:54.713	Thread Group 1-24	HTTP Request	7652		635	192	7651	310
27	23:30:58.127	Thread Group 1-41	HTTP Request	4456		635	192	4456	294
28	23:30:54.128	Thread Group 1-21	HTTP Request	8586		635	192	8586	277
29	23:30:55.313	Thread Group 1-27	HTTP Request	7569		635	192	7569	288
30	23:30:56.512	Thread Group 1-33	HTTP Request	6472		635	192	6471	301
31	23:30:57.529	Thread Group 1-38	HTTP Request	5788		635	192	5788	287
32	23:30:54.926	Thread Group 1-25	HTTP Request	8734		635	192	8733	274
33	23:30:57.923	Thread Group 1-40	HTTP Request	5826		635	192	5826	292
34	23:30:59.727	Thread Group 1-49	HTTP Request	4051		635	192	4051	276
35	23:30:57.144	Thread Group 1-36	HTTP Request	7446		635	192	7446	284
36	23:30:53.713	Thread Group 1-19	HTTP Request	10910		635	192	10910	289
37	23:30:54.514	Thread Group 1-23	HTTP Request	10112		635	192	10111	274
38	23:30:59.116	Thread Group 1-46	HTTP Request	5587		635	192	5587	290
39	23:30:58.726	Thread Group 1-44	HTTP Request	6203		635	192	6202	291
40	23:30:57.717	Thread Group 1-39	HTTP Request	7907		635	192	7907	286
41	23:30:53.313	Thread Group 1-17	HTTP Request	12325		636	193	12325	275
42	23:30:55.712	Thread Group 1-29	HTTP Request	10788		635	192	10788	289
43	23:30:59.919	Thread Group 1-50	HTTP Request	7037		635	192	7037	290
44	23:30:56.312	Thread Group 1-32	HTTP Request	10695		635	192	10695	290
45	23:30:56.712	Thread Group 1-34	HTTP Request	10347		635	192	10347	291
46	23:30:57.335	Thread Group 1-37	HTTP Request	9882		635	192	9882	296
47	23:30:55.513	Thread Group 1-28	HTTP Request	11887		636	193	11887	288
48	23:30:56.942	Thread Group 1-35	HTTP Request	11667		636	193	11667	284
49	23:30:58.315	Thread Group 1-42	HTTP Request	11410		636	193	11410	289
50	23:30:58.928	Thread Group 1-45	HTTP Request	11252		635	192	11252	286

- The green check marks in the "Status" column indicate that all requests were successfully completed without errors.
- The highest recorded response time is around 13 sec, indicating a response with a higher delay.
- The connection times range between 273 ms and 313 ms, which is consistent with previous results.
- Latency values are close to the sample time, confirming that most of the response time is due to the API's built-in delay.

Overall Conclusion of Jmeter Test:

- The API response time changes a lot, from 1.8 seconds to 13.4 seconds, which matches the random delay (1 to 10 seconds) used in the test.
- All requests were completed successfully (0% error rate), showing that the API is very stable and has no failures.
- The API can handle only a certain number of requests per minute,so it is not good for high applications.
- The throughput is quite low (33.6 requests per minute) because of the built-in delay in the API.
- If faster responses are required, API optimizations should be done.
- The API is not suitable for real-time applications, but it performs reliably within its intended design.

Report Generation of the Jmeter Test.

- We can generate a HTML report from the Jmeter Test Plan
- Reports can be generated through CLI option or Jmeter GUI mode.
- To generate the report from CLI option the following script is needed:

jmeter -n -t{jmx file} -l C:\Users\Admin\OneDrive\Desktop\result.csv -e -o C:\Users\Admin\OneDrive\Desktop\HtmlReport

[jmeter -n -t Assignment.jmx -l C:\Users\Admin\OneDrive\Desktop\result.csv -e -o C:\Users\Admin\OneDrive\Desktop\HtmlReport]

- The above script will generate a HTML report of the Test Plan.

Reports