

Load Test Plan - Conclusions

Author - Roxana Stefania Pruteanu

SINGLE METHOD(GET) with multiple users

#1st scenario - Listing all resources, 10 users/s with 1 GET requests = 10REQ/s x 10 seconds - 100

# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput
100	98	0	171	18.01	0.00%	10.0/sec
100	98	0	171	18.01	0.00%	10.0/sec

- Expected results:

Resp time <= 2ms | STATUS 200

- Throughput handles 10 requests per second
- All tests have PASSED with 0% error rate

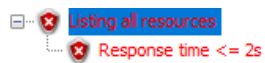
#2nd scenario - Listing all resources, 50 users/s with 1 GET requests = 50REQ/s x 10 seconds - 500

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput
:Listing all resources	500	98	0	201	19.58	0.20%	49.6/sec
TOTAL	500	98	0	201	19.58	0.20%	49.6/sec

- Expected results:

Resp time <= 2ms | STATUS 200 | Resp content

- Throughput handles 50 requests per second
- All tests have PASSED with 0.20% error rate



Thread Name: 1-177
Sample Start: 2023-09-04 17:01:11 EEST
Load time: 201

#3rd scenario - Listing all resources, 60 users/s with 1 GET requests = 60REQ/s x 10 seconds - 600

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput
:Listing all resources	600	108	0	274	35.63	0.00%	58.5/sec
TOTAL	600	108	0	274	35.63	0.00%	58.5/sec

- Expected results:

Resp time <= 2ms | STATUS 200

- Throughput handles slightly lower requests per second than expected
- All tests have PASSED with 0% error rate

#4th scenario - Listing all resources, 66 users/s with 1 GET requests = 66REQ/s x 10 seconds - 660

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput
:Listing all resources	660	703	0	10232	1210.99	1.67%	35.1/sec
TOTAL	660	703	0	10232	1210.99	1.67%	35.1/sec

- Expected results:

Resp time <= 2ms | STATUS 200

- Throughput nearing to half of the expected results

Error rate due to "Response code:Non HTTP response code: java.net.SocketException"

- Tests suspended for SINGLE METHOD

MULTIPLE METHODS(GET,POST, PUT, PATCH, DELETE) with multiple users

#1st scenario - 60 users/s with 2 requests = 120REQ/s x 10 seconds - 1200

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput
:Listing all resources	600	365	0	3601	572.31	0.00%	47.3/sec
:Creating a resource	600	249	0	5069	197.61	0.00%	38.5/sec
TOTAL	1200	307	0	5069	432.04	0.00%	76.6/sec

- Expected results:

STATUS 200 | 201

- All tests have PASSED with 0% error rate

#2nd scenario - 60 users/s with 3 requests = 180REQ/s x 10 seconds - 1800

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput
:Listing all resources	600	117	0	268	43.65	0.00%	58.1/sec
:Creating a resource	600	244	0	1336	48.29	0.00%	57.4/sec
:Updating a resource	600	234	0	289	8.31	0.00%	57.5/sec
TOTAL	1800	199	0	1336	68.94	0.00%	166.8/sec

- Great results here, it seems I can't rely on my environment, lucky strikes..

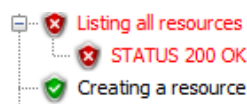
#3rd scenario - 60 users/s with 5 requests = 300REQ/s x 10 seconds - 3000

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput
:Listing all resources	600	105	0	738	45.02	0.00%	58.4/sec
:Creating a resource	600	237	0	312	13.68	0.00%	57.7/sec
:Updating a resource	600	233	0	303	7.93	0.00%	57.8/sec
:Patching a resource	600	232	0	283	7.23	0.00%	57.9/sec
:Deleting a resource	600	232	0	280	7.25	0.00%	57.9/sec
TOTAL	3000	208	0	738	55.95	0.00%	268.1/sec

- Fantastic, let's ramp up with a heavier load next time in 3rd scenario

#3rd scenario - 120 users/s with 5 requests = 600REQ/s x 10 seconds - 3000

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput
:Listing all resources	600	5941	0	50589	5680.39	18.83%	11.5/sec
:Creating a resource	600	1188	0	13122	2056.11	0.00%	11.4/sec
:Updating a resource	600	434	0	15939	1072.18	0.00%	11.4/sec
:Patching a resource	600	264	0	3553	259.77	0.00%	11.4/sec
:Deleting a resource	600	253	0	2449	165.48	0.00%	11.4/sec
TOTAL	3000	1616	0	50589	3513.05	3.77%	56.3/sec



```
Assertion error:false
Assertion failure:true
Assertion failure message:'Status fails for 200 OK'
```

Error rate due to "Response code:Non HTTP response code: java.net.SocketException"

- Tests suspended for SINGLE METHOD