

# Performance test report - Jun 30, 2024 (#48)

Open in Postman

Postman collection: MasterThesisOriginal  
Report exported on: Jun 30, 2024, 21:30:17 (GMT+1)

## Test setup

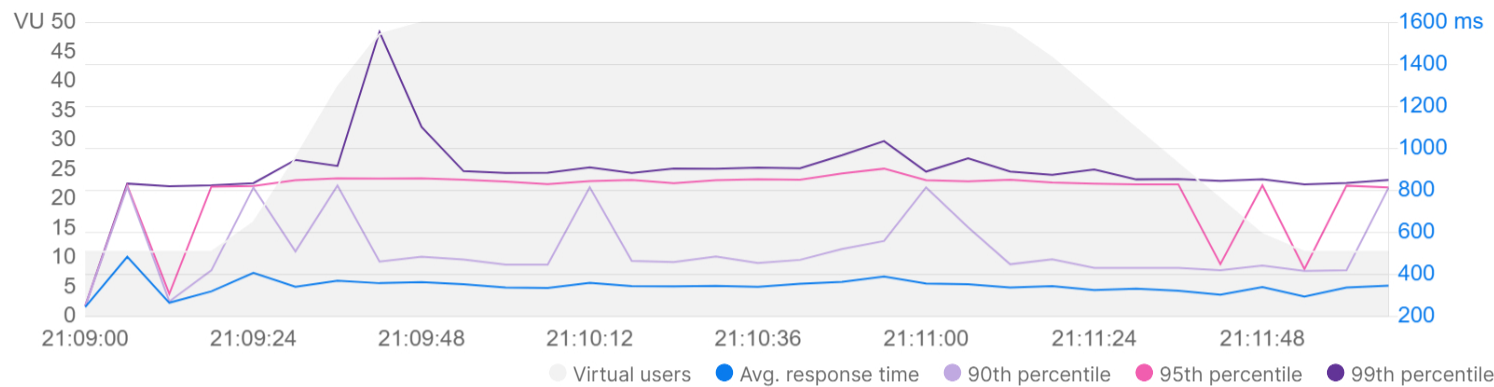
Virtual users	Start time	Load profile
50 VU	Jun 30, 21:09:01 (GMT+1)	Peak
Duration	End time	Environment
3 minutes	Jun 30, 21:12:10 (GMT+1)	-

## 1. Summary

Total requests sent	Throughput	Average response time	Error rate
14,937	79.04 requests/second	348 ms	0.01 %

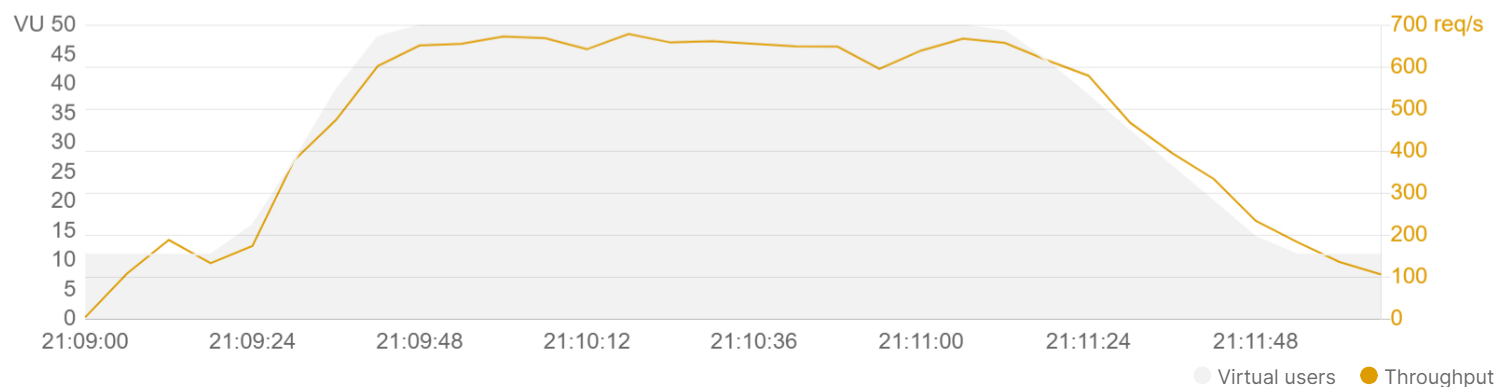
### 1.1 Response time

Response time trends during the test duration.



### 1.2 Throughput

Rate of requests sent per second during the test duration.



### 1.3 Requests with slowest response times

Top 5 slowest requests based on their average response times.

Request	Resp. time (Avg ms)	90th (ms)	95th (ms)	99th (ms)	Min (ms)	Max (ms)
<b>POST</b> 4- /api/checkout http://{{IP}}:{{PORT}}/api/checkout	885	917	991	2,228	810	2,595
<b>POST</b> 4- /api/checkout Copy http://{{IP}}:{{PORT}}/api/checkout	873	935	978	1,255	809	2,172
<b>POST</b> 4- /api/checkout Copy 2 http://{{IP}}:{{PORT}}/api/checkout	870	919	970	1,269	808	2,253
<b>POST</b> 3- /api/cart Copy 5 http://{{IP}}:{{PORT}}/api/cart	434	464	483	561	413	712
<b>POST</b> 3- /api/cart Copy 3 http://{{IP}}:{{PORT}}/api/cart	432	453	483	552	413	730

### 1.4 Requests with most errors

Top 5 requests with the most errors, along with the most frequently occurring errors for each request.

Request	Total error count	Error 1	Error 2	Other errors
<b>GET</b> 13- /api/products/2ZYFJ3GM2N http://{{IP}}:{{PORT}}/api/products/2ZYFJ3GM2N	1	503 Service Unavailable (1)	-	0

## 2. Metrics for each request

The requests are shown in the order they were sent by virtual users.

Request	Total requests	Requests/s	Min (ms)	Avg (ms)	90th (ms)	Max (ms)	Error %
<b>GET</b> 1- / http://{{IP}}:{{PORT}}/	469	2.48	220	246	276	453	0

<b>GET</b> 2- /api/cart http://{{IP}}:{{PORT}}/api/cart	469	2.48	218	233	247	565	0
<b>POST</b> 3- /api/cart http://{{IP}}:{{PORT}}/api/cart	460	2.43	414	430	451	525	0
<b>POST</b> 3- /api/cart Copy http://{{IP}}:{{PORT}}/api/cart	442	2.34	414	429	447	566	0
<b>POST</b> 4- /api/checkout http://{{IP}}:{{PORT}}/api/checkout	427	2.26	810	885	917	2,595	0
<b>POST</b> 3- /api/cart Copy 2 http://{{IP}}:{{PORT}}/api/cart	427	2.26	414	431	455	600	0
<b>POST</b> 3- /api/cart Copy 3 http://{{IP}}:{{PORT}}/api/cart	427	2.26	413	432	453	730	0
<b>POST</b> 4- /api/checkout Copy http://{{IP}}:{{PORT}}/api/checkout	427	2.26	809	873	935	2,172	0
<b>POST</b> 3- /api/cart Copy 4 http://{{IP}}:{{PORT}}/api/cart	427	2.26	414	432	456	597	0
<b>POST</b> 3- /api/cart Copy 5 http://{{IP}}:{{PORT}}/api/cart	427	2.26	413	434	464	712	0
<b>POST</b> 4- /api/checkout Copy 2 http://{{IP}}:{{PORT}}/api/checkout	427	2.26	808	870	919	2,253	0
<b>GET</b> 5- /api/data http://{{IP}}:{{PORT}}/api/data	427	2.26	220	233	248	418	0
<b>GET</b> 6- /api/data?contextKeys=assembly http://{{IP}}:{{PORT}}/api/data?contextKeys=assembly	427	2.26	240	257	277	641	0
<b>GET</b> 7- /api/data?contextKeys=binoculars http://{{IP}}:{{PORT}}/api/data?contextKeys=binoculars	427	2.26	242	260	282	625	0
<b>GET</b> 8- /api/data?contextKeys=books http://{{IP}}:{{PORT}}/api/data?contextKeys=books	427	2.26	236	252	271	440	0

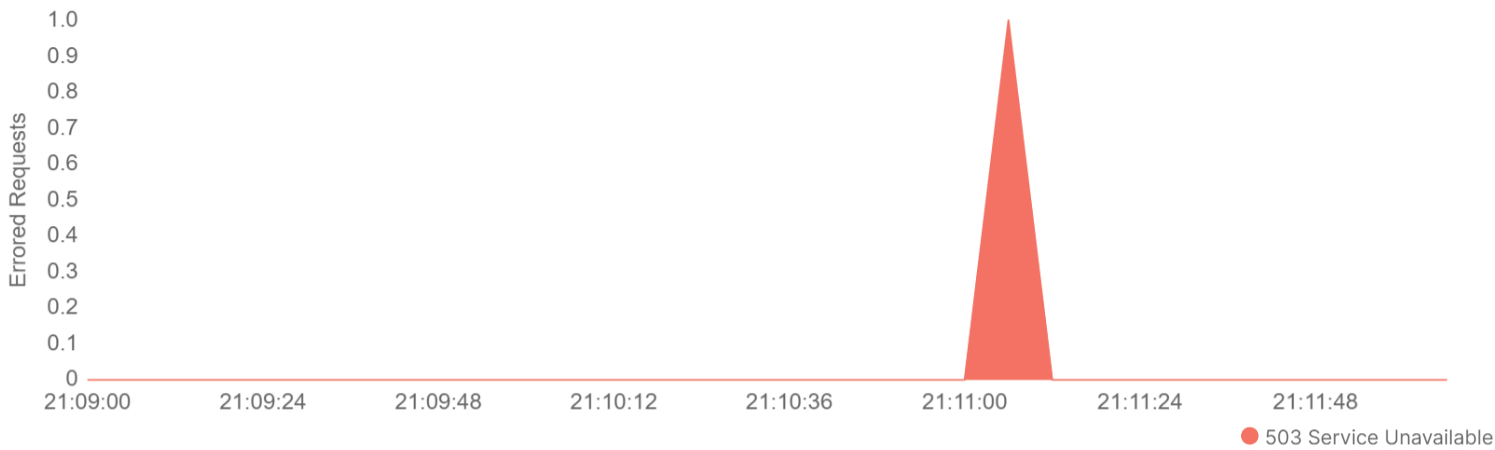
<b>GET</b> 9- /api/data?contextKeys=telescopes http://{{IP}}:{{PORT}}/api/data?contextKeys=telescopes	425	2.25	243	259	278	479	0
<b>GET</b> 10- /api/data/?contextKeys=travel http://{{IP}}:{{PORT}}/api/data/?contextKeys=travel	422	2.23	300	315	333	491	0
<b>GET</b> 11- /api/products/OPUK6V6EV0 http://{{IP}}:{{PORT}}/api/products/OPUK6V6EV0	421	2.23	234	247	258	631	0
<b>GET</b> 12- /api/products/1YMWWN1N4O http://{{IP}}:{{PORT}}/api/products/1YMWWN1N4O	421	2.23	235	245	254	530	0
<b>GET</b> 13- /api/products/2ZYFJ3GM2N http://{{IP}}:{{PORT}}/api/products/2ZYFJ3GM2N	420	2.22	235	248	260	551	0.24
<b>GET</b> 14- /api/products/66VCHSJNUP http://{{IP}}:{{PORT}}/api/products/66VCHSJNUP	420	2.22	234	247	258	502	0
<b>GET</b> 15- /api/products/6E92ZMYFZ http://{{IP}}:{{PORT}}/api/products/6E92ZMYFZ	420	2.22	221	248	260	551	0
<b>GET</b> 16- /api/products/9SIQT8TOJO http://{{IP}}:{{PORT}}/api/products/9SIQT8TOJO	420	2.22	234	246	258	647	0
<b>GET</b> 17- /api/products/HQTGWGPNH4 http://{{IP}}:{{PORT}}/api/products/HQTGWGPNH4	420	2.22	233	246	257	515	0
<b>GET</b> 18- /api/products/L9ECAV7KIM http://{{IP}}:{{PORT}}/api/products/L9ECAV7KIM	420	2.22	234	244	257	451	0
<b>GET</b> 19- /api/products/LS4PSXUNUM http://{{IP}}:{{PORT}}/api/products/LS4PSXUNUM	420	2.22	234	246	262	342	0
<b>GET</b> 20- /api/products/OLJCESPC7Z http://{{IP}}:{{PORT}}/api/products/OLJCESPC7Z	419	2.22	232	256	277	471	0
<b>GET</b> 21- /api/recommendations? productId=0PUK6V6EV0 http://{{IP}}:{{PORT}}/api/recommendations? productId=0PUK6V6EV0	419	2.22	249	301	367	645	0

GET 22- /api/recommendations? productId=1YMWWN1N4O http://{{IP}}:{{PORT}}/api/recommendations? productId=1YMWWN1N4O	419	2.22	261	308	381	647	0
GET 23- /api/recommendations? productId=2ZYFJ3GM2N http://{{IP}}:{{PORT}}/api/recommendations? productId=2ZYFJ3GM2N	419	2.22	260	303	376	623	0
GET 24- /api/recommendations? productId=66VCHSJNUP http://{{IP}}:{{PORT}}/api/recommendations? productId=66VCHSJNUP	419	2.22	261	308	387	712	0
GET 25- /api/recommendations? productId=6E92ZMYFZ http://{{IP}}:{{PORT}}/api/recommendations? productId=6E92ZMYFZ	419	2.22	261	306	371	694	0
GET 26- /api/recommendations? productId=9SIQT8TOJO http://{{IP}}:{{PORT}}/api/recommendations? productId=9SIQT8TOJO	419	2.22	257	309	380	746	0
GET 27- /api/recommendations? productId=HQTGWGPNH4 http://{{IP}}:{{PORT}}/api/recommendations? productId=HQTGWGPNH4	419	2.22	261	310	391	627	0
GET 28- /api/recommendations? productId=L9ECAV7KIM http://{{IP}}:{{PORT}}/api/recommendations?productId=L9ECAV7KIM	419	2.22	250	293	365	542	0

### 3. Errors

#### 3.1 Error distribution over time

Top 5 error classes observed during the test duration.



#### 3.2 Error distribution for requests

Errored requests grouped by error class, along with the error count for each class.

Error class	Total counts
503 Service Unavailable	1
GET 13- /api/products/2ZYFJ3GM2N	1



#### Testing API performance on Postman

Postman enables you to simulate user traffic and observe how your API behaves under load. It also helps you identify any issues or bottlenecks that affect performance.

Learn more about [testing API performance](#).