

Postman collection: NBA GraphQL

Report exported on: Mar 21, 2024, 0:32:42 (CDT)

Test setup

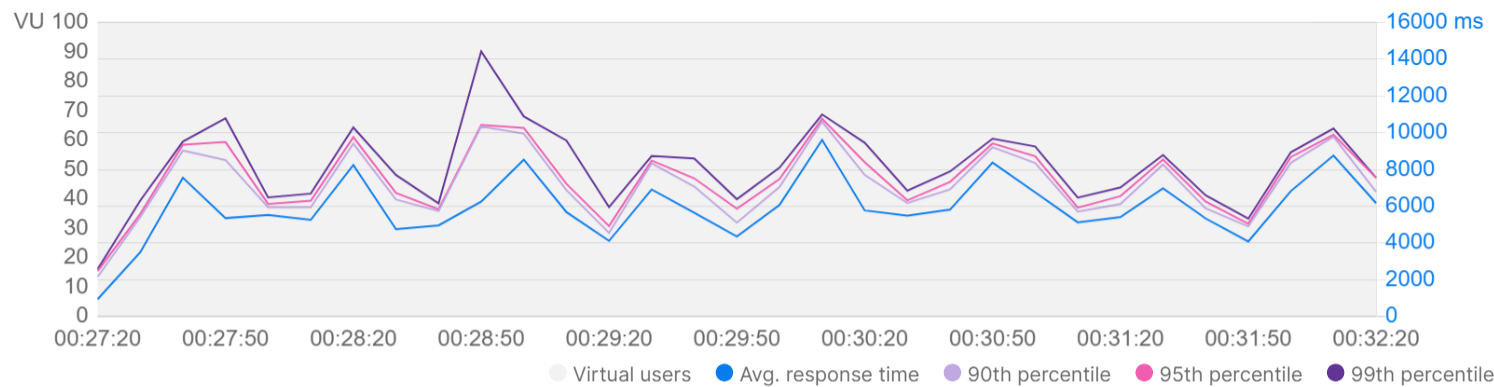
Virtual users	Start time	Load profile
100 VU	Mar 21, 0:27:14 (CDT)	Fixed
Duration	End time	Environment
5 minutes	Mar 21, 0:32:20 (CDT)	GraphQL

1. Summary

Total requests sent	Throughput	Average response time	Error rate
4,664	15.21 requests/second	5,808 ms	0.00 %

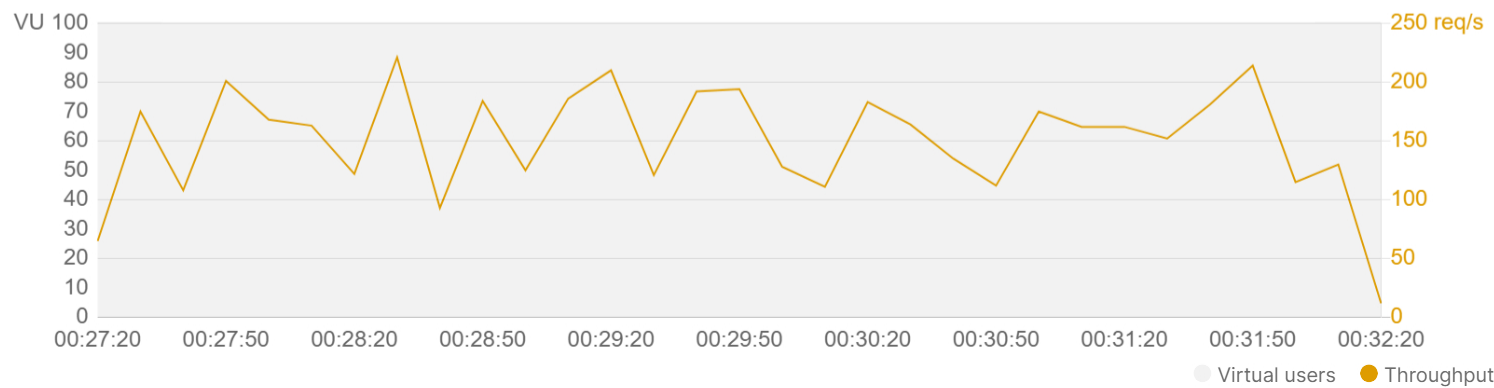
1.1 Response time

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>GET</b> Season - Combined Data {{url}}/graphql/	7,230	9,625	10,264	10,847	819	11,837
<b>GET</b> Playoffs - Advanced Data {{url}}/graphql/	6,346	8,646	9,263	10,026	2,150	10,633
<b>GET</b> Team Data http://209.38.172.107/graphql/	5,441	7,923	8,587	10,042	435	10,420
<b>GET</b> Season - Totals Data {{url}}/graphql/	5,432	7,918	8,851	10,250	171	14,685
<b>GET</b> Playoffs - Totals Data http://209.38.172.107/graphql/	5,223	7,011	7,997	9,559	364	10,082

## 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> Season - Advanced Data {{url}}/graphql/	816	2.66	260	5,147	7,845	14,426	0
<b>GET</b> Season - Totals Data {{url}}/graphql/	807	2.63	171	5,432	7,918	14,685	0
<b>GET</b> Season - Combined Data {{url}}/graphql/	790	2.58	819	7,230	9,625	11,837	0
<b>GET</b> Playoffs - Advanced Data {{url}}/graphql/	782	2.55	2,150	6,346	8,646	10,633	0
<b>GET</b> Playoffs - Totals Data http://209.38.172.107/graphql/	746	2.43	364	5,223	7,011	10,082	0
<b>GET</b> Team Data http://209.38.172.107/graphql/	723	2.36	435	5,441	7,923	10,420	0

### 3. Errors

This run has no errors

All requests were sent successfully and returned a 2xx response code.



## 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](#).