



## When Postman Goes That Extra Mile to Deliver Performance to APIs

---

V S Rini Susan

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

April 3, 2024

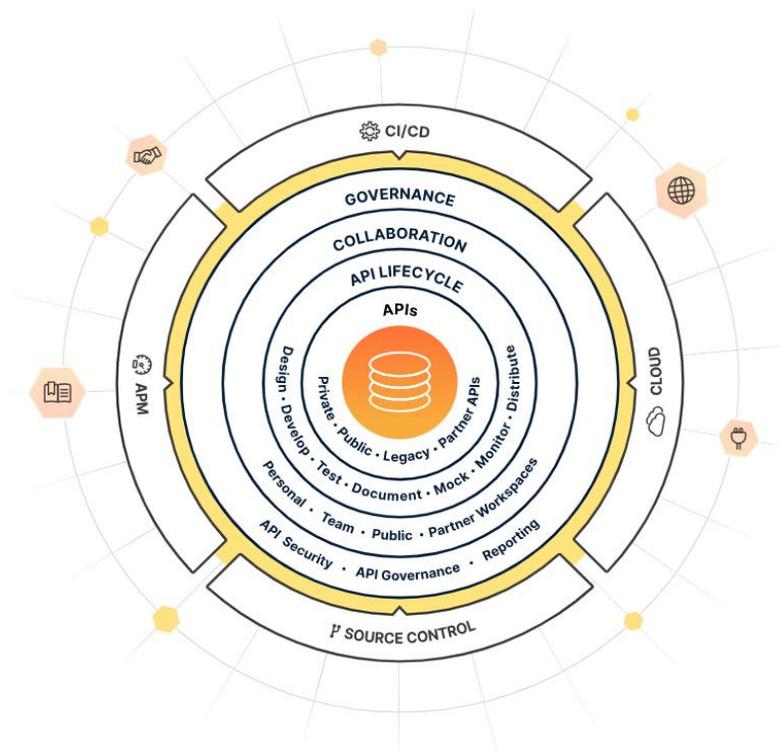
# When Postman goes that extra mile to deliver Performance to APIs

By

Rini Susan V S

## Background

Postman is an API platform for building and testing Application Programming Interfaces [API]. Postman streamlines the stages in the API lifecycle and promotes collaboration to generate APIs, better and faster. Postman supports API development with the API Builder and also provides the option to write scripts to perform API tests. Postman has a Collections feature that helps to organize a group of saved API requests. Postman also provides integrations with various CI/CD, APM, Cloud, and Collaboration tools like Jenkins, Splunk, NewRelic, AWS API Gateway, GitHub, Slack, etc.



Wikipedia defines software testing as, an investigation conducted to provide stakeholders with information about the quality of the software product or service under test. Performance testing is a type of testing in which the speed, responsiveness, and stability of a software, product, or network is evaluated under peak workload. Performance testing checks and validates an application's capacity and ensures that it works well within the acceptable Service Level Agreements.

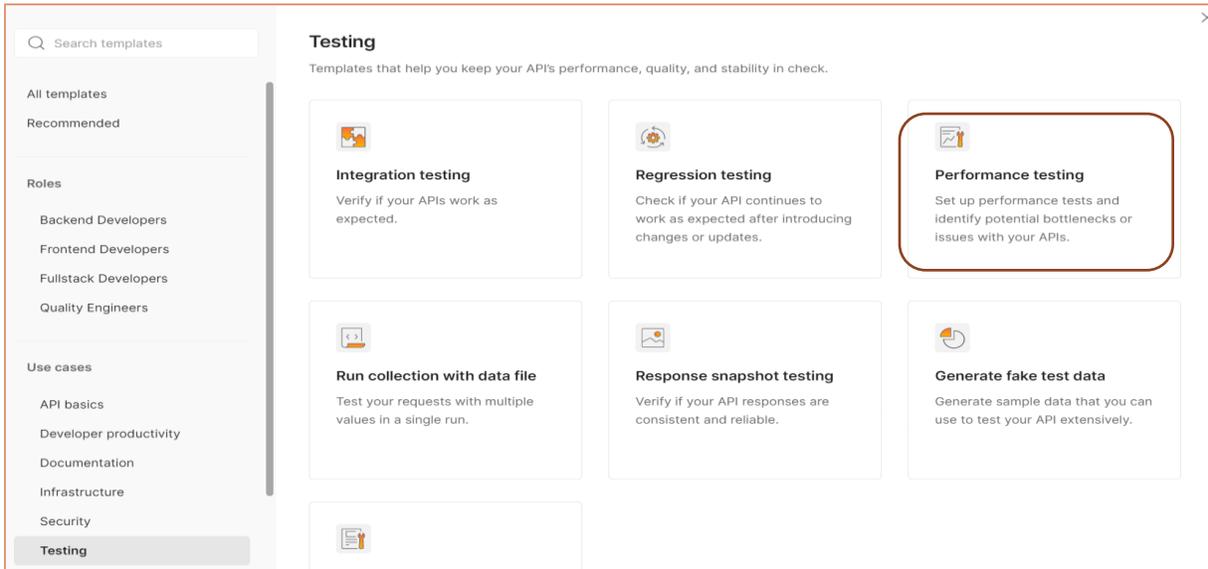
## API Performance Testing

Performance testing is done to evaluate API performance and system capacity under expected traffic/load. It helps to identify performance bottlenecks under unusual traffic. Some of the essential information regarding API performance testing is provided below,

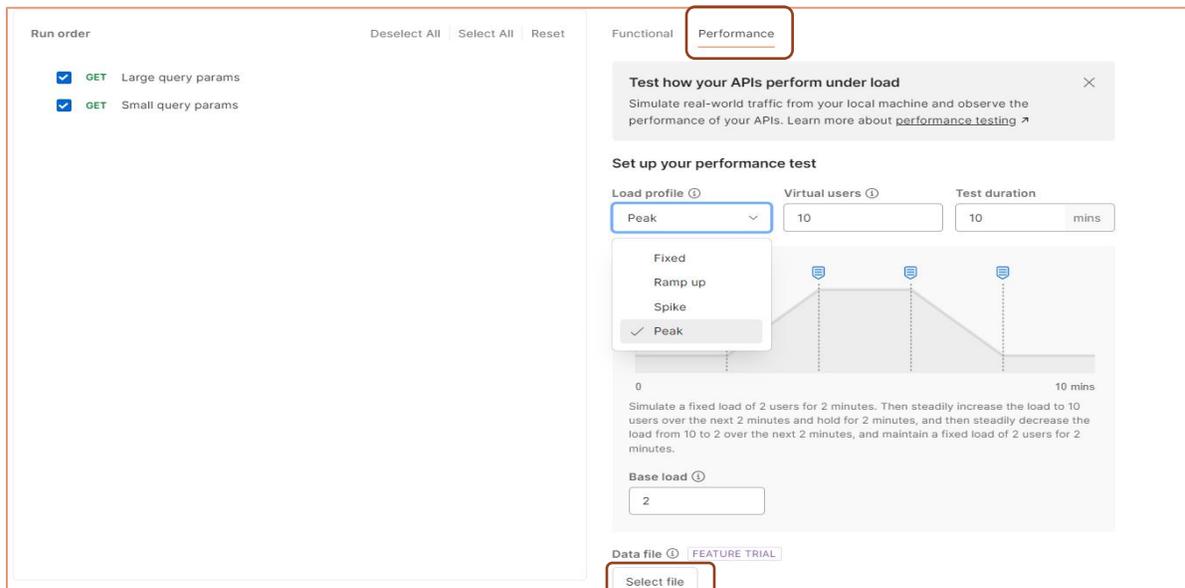
- Types of APIs tested – REST, SOAP
- Types of API performance tests - Smoke, Load, Stress, Spike, Endurance
- Tools used for API testing - Postman, SoapUI, Apache JMeter, LoadRunner, Gatling

## Performance Testing using Postman

The performance testing feature in Postman is designed to test the API endpoints under varying loads and to assess how well the APIs handle different loads. Accordingly, potential performance bottlenecks in API can be identified and performance-tuning recommendations can be provided by the performance team. It helps to optimize and enhance the overall API performance.

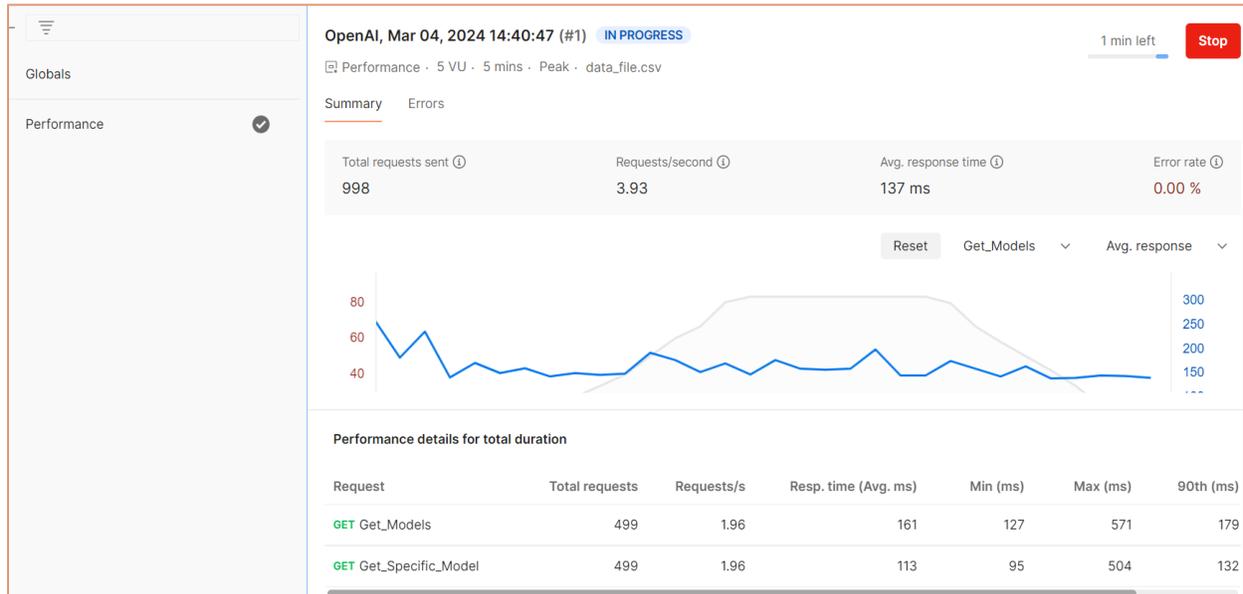


Currently, the load profile feature in the tool enables testers to execute performance tests with fixed, ramp up, spike, and peak loads. The data file feature enables the testers to use the dataset file required to load test API with different data sets in each iteration.

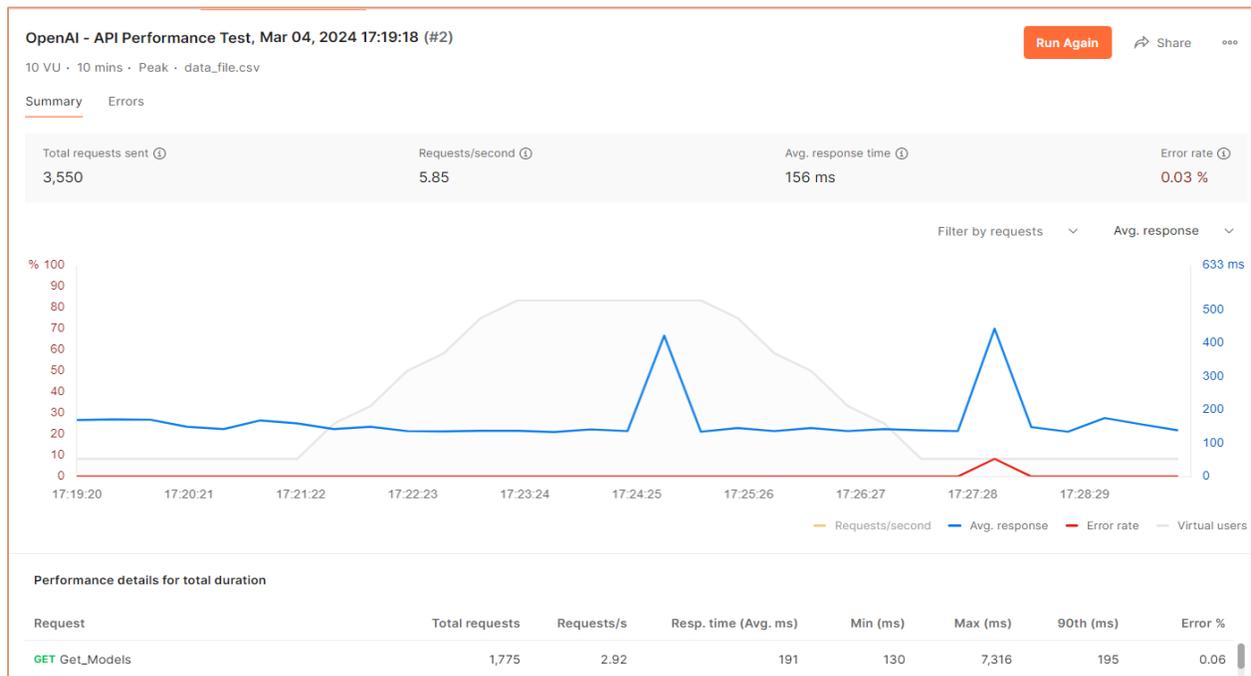


## Test Monitoring

Load test execution can be monitored in real-time through the Postman Summary tab. A summary of performance metrics is available in tabular and graphical format and includes test duration, virtual user count, the total request count, requests/second, average response time, and errors of APIs.



Test summary will be displayed after execution in the Summary tab as shown below.



# Test Report

Reports are generated after test execution and can be downloaded in pdf or html format. Test dashboard details can also be shared as a link.

## Performance test report - Mar 4, 2024 (#2)

[Open in Postman](#)

Postman collection: OpenAI - API Performance Test  
Report exported on: Mar 4, 2024, 17:35:33 (PST)

### Test setup

Virtual users 10 VU	Start time Mar 4, 17:19:19 (PST)	Load profile Peak
Duration 10 minutes	End time Mar 4, 17:29:26 (PST)	Environment -

### 1. Summary

Total requests sent 3,550	Throughput 5.85 requests/second	Average response time 156 ms	Error rate 0.03 %
------------------------------	------------------------------------	---------------------------------	----------------------

### 2. Metrics for each request

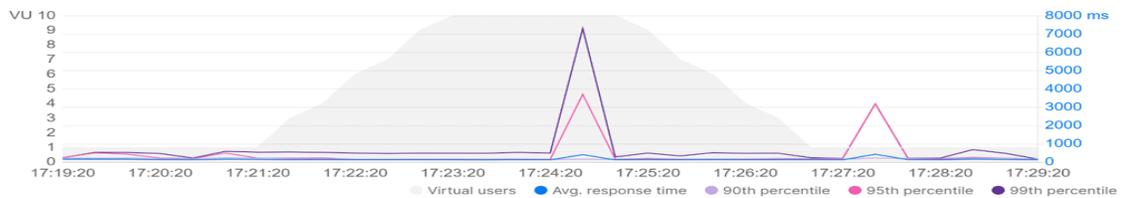
The requests are shown in the order they were sent by virtual users.

Request	Total requests	Requests/s	Min (ms)	Avg (ms)	90th (ms)	Max (ms)	Error %
<b>GET</b> Get_Models <code>https://api.openai.com/v1/models</code>	1,775	2.92	130	191	195	7,316	0.06
<b>GET</b> Get_Specific_Model <code>https://api.openai.com/v1/models/{{model_name}}</code>	1,775	2.92	95	122	123	7,290	0

Details of each performance metric are also available in the test report. Transaction response time and throughput are displayed in graphical format, while the top 5 requests with the slowest response times and errors are displayed in tabular format.

#### 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> Get_Models https://api.openai.com/v1/models	191	195	250	580	130	7,316
<b>GET</b> Get_Specific_Model https://api.openai.com/v1/models/{{model_name}}	122	123	134	195	95	7,290

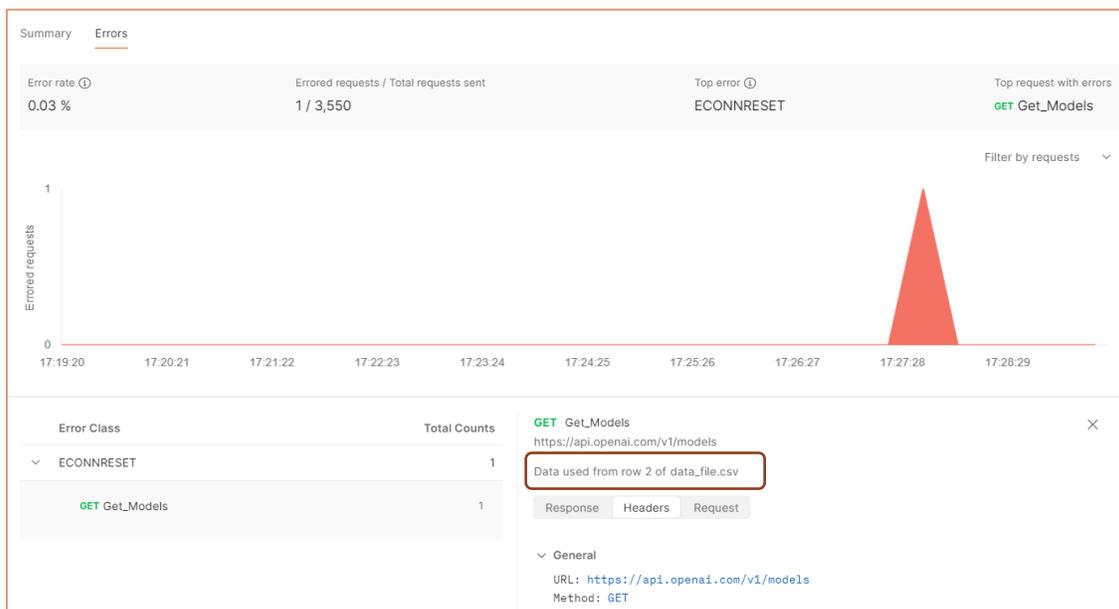
### 1.4 Requests with most errors

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> Get_Models https://api.openai.com/v1/models	1	ECONNRESET (1)	-	0

## Error Analysis

Detailed error analysis is available in the Errors tab and provides information about the API name, error class, error code, number of errors, and data set that caused the error. The error code and error class help testers easily identify the root cause of the error, whether it is an issue caused by incorrect data or a code error. In case of a data issue, the ability to view the request and response of the specific error request in the Error tab is very beneficial. It helps to pinpoint the exact data that caused the error during execution.



## Closing Thoughts

The Postman tool provides options to develop and test different aspects of API efficiently. The tool improves collaboration; team members can collaborate in team workspaces to develop and test different aspects of API. To sum up, the performance testing capability introduced in the Postman is a valuable feature that enhances the overall scalability feature of API. It provides option to reuse existing API collections for performance testing with minimal effort. From a testing tool standpoint, there is still scope for improvement in terms of features and capabilities.

## Beneficial Aspects

- The performance test team can use the API collections created by the development or functional testing team as a base and make necessary modifications. It would reduce the overall scripting effort and time taken in the project.
- The capability to test with 100 virtual users even in the Postman Free plan is very beneficial for small projects with budget and resource constraints.
- The test reports available in shareable PDF and HTML formats help in easy test reporting and analysis.

## Areas of Improvement

- Timer features to manage the frequency of requests based on test requirements and sleep time option to introduce a delay between the requests to emulate real-world scenarios are unavailable, unlike other load-testing tools.
- The test scenario can have only one data file, which is an unlikely scenario in load testing. Normally, load test scenarios have multiple data set files and require options to control and format the data.

## References

<https://www.postman.com/product/what-is-postman/>

<https://learning.postman.com/docs/introduction/overview/>