8 ways to improve API performance

Description

In this blog post we are going to see eight ways to improve API performance. Have you ever seen the testers or API consumers complaining about the API being too slow or responding very slowly? It can be really painful specially when you know the API that you have written follows all the coding principles and what not but still performing poorly. Specially this can be a nightmare when you get to know that your APIs are performing poorly just weeks before production. I have noted eight different points which you can consider reevaluating in your application which may improve API performance

Lets get started ðŸ˜Š

Tip#1 Pagination for API performance

There is a possibility that a REST API might return a huge result set based on a specific criteria. If you think that there is absolutely no chance that your application will return millions of records back to the client, then you need to rethink. There can be a risk of system crash if not implemented on the right time.

Pagination is a good practice and it should be implemented so that more results are returned on demand basis.

Tip#2 Payload Compression

API responses can be compressed in order to reduce the time taken during data transmission. We can use GZIP in order to make the upload and download processes quicker. The only thing required to do is, pass **‘gzip’** in the **‘Accept-Encoding’** header. The client may spend a bit of CPU power to decompress the file, but this is better than downloading a large file which will take time.

Tip#3 Connection Pooling

Opening and closing a database connection is one of the most time-consuming operation, because it adds additional processing time. Having a database connection pool saves us from opening and closing a connection for every API call. The pool manages the lifecycle of connections for efficient resource use.

Tip#4 Asynchronous Logging:

In Asynchronous logging, the logs are sent to a lock free buffer instead of dealing with the disk onevery call. The logs from the buffer are periodically flushed to the disk which indirectly reduces the I/O overhead.

Tip#5 Data Caching:

Caching is used to return the results stored in the cache rather than returning it from the database. Cache is faster as compared to the database. Most frequently used data can be stored in the cache for faster data access. If certain requests always return the same response, then putting the response in the cache prevents unnecessary database querying. We should consider spending time, monitoring and assessing your API to see if certain resources are getting used more heavily than others. We should consider including those resources in the cache if itâ€™s possible.

Tip#6 Simplify Database Queries

There can be a APIs which return data from various database tables. We should try and optimize database queries to return results in less time.

Tip#7 Use PATCH When Possible

Developers often think PUT and PATCH are one and the same, but they are actually different. A PUT request interacts with the entire resource. A PATCH request only interacts with a small portion, making it suitable for updating files or versions. PATCH requests deal with smaller payloads, which will help optimize API performance and make your network as efficient as possible.

While PATCH can reduce the size of each request, note that itâ€™s not idempotent, hence it can yield different results with multiple requests. So carefully consider evaluating the use case for PATCH requests and ensure theyâ€™re idempotently built if needed, or use PUT requests in those cases.

Tip#8 Rate Limiting

We can rate limit our APIs to prevent DDOS (Distributed denial of service) attacks. Rate limiting means allowing only ‘n’ no of requests for consumers or IP address in a specific duration to prevent the server from getting overwhelmed with huge traffic.

If you want to read more on Rate limiting then you can click here

Conclusion on improve API Performance

I hope you liked this post talks about how to improve the API performance. Apart from expecting the desired output from the APIs , we should also give importance to see if the APIs are responding real quick. Apart from Unit testing and Integration testing, a proper Performance testing should also bedone.

If you have come across any other pointers which helped you improve the API performance, then you can let me know in the comments section below. Until then Happy Learning! ðŸ˜Š

Category

1. Design

Date Created
December 28, 2023
Author
kk-ravi144gmail-com