



How to Improve Node.js API Performance?

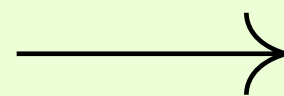
@princemeghani

Use Asynchronous Functions

Asynchronous functions are the heart of JavaScript. They perform non-blocking I/O operations, allowing the CPU to handle multiple requests simultaneously.



@princemeghani	→
----------------	---

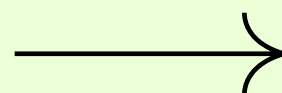


Optimize Database Queries

Efficient database queries can significantly reduce response time. Use logging to find which query takes the most time and optimize it.

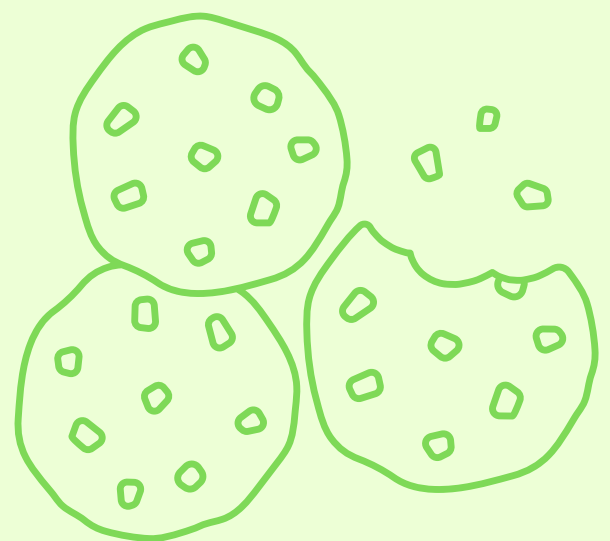


@princemeghani	→
----------------	---

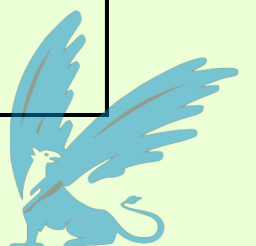
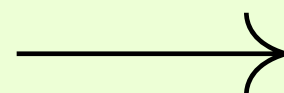


Avoid Sessions and Cookies in APIs

Stateless APIs are common and provide JWT, OAuth, and other authentication mechanisms. These authentication tokens are kept on the client side, reducing the load on server.



@princemeghani	→
----------------	---

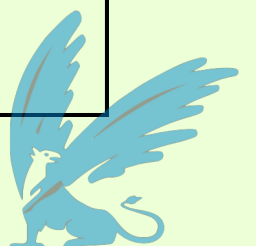
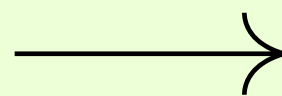


Use Caching

Caching can handle frequent common requests, removing the need to make additional queries.



@princemeghani	→
----------------	---

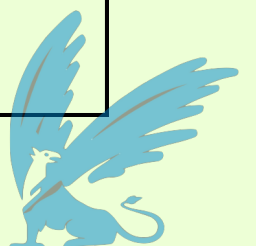
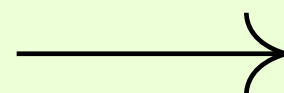


Efficient Code and Architecture Design

Breaking your application into smaller, self-contained modules can reduce complexity and make it easier to manage and scale.



@princemeghani	→
----------------	---

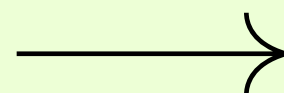


Use the Latest Version of Node.js

Newer versions of Node.js generally have performance improvements and bug fixes.

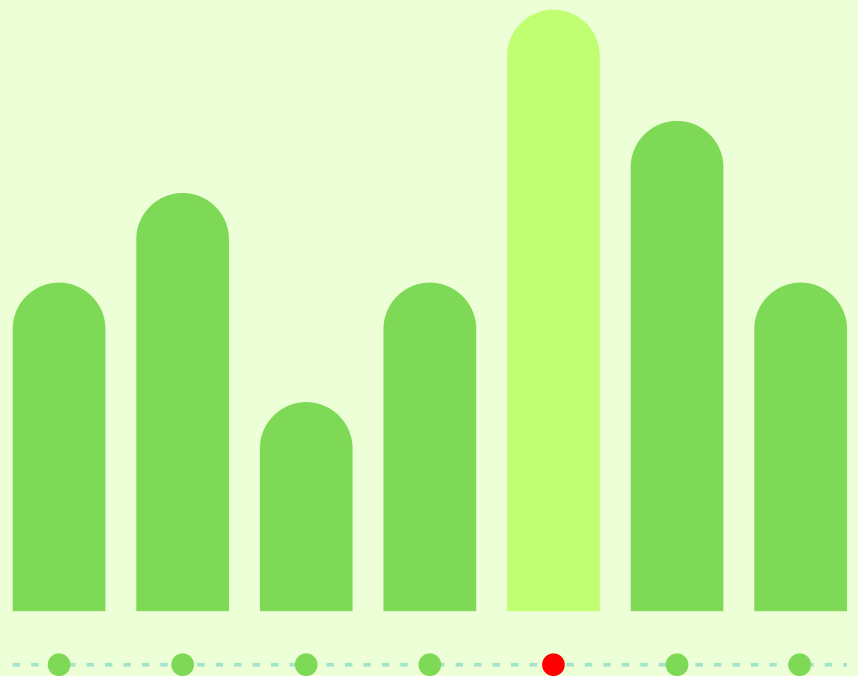


@princemeghani	→
----------------	---

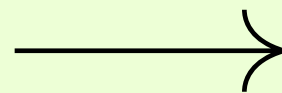


Use a Profiler

A profiler can help you identify areas of your code that are causing performance bottlenecks, such as slow function calls or memory leaks.

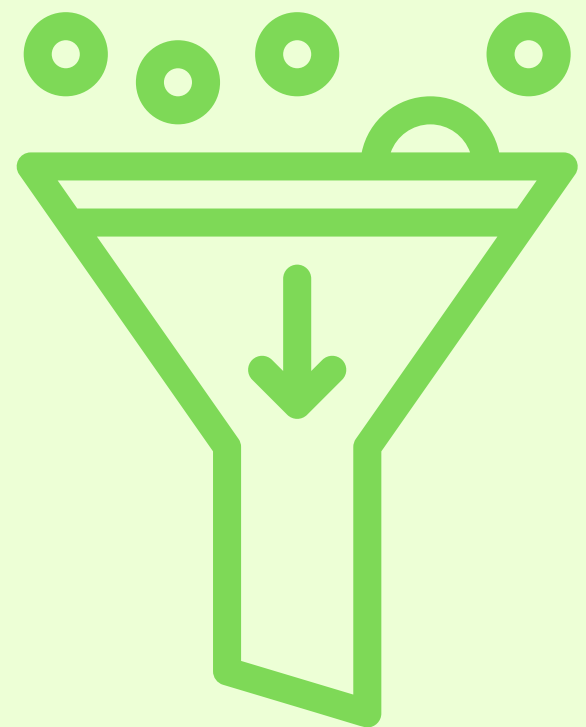


@princemeghani	→
----------------	---

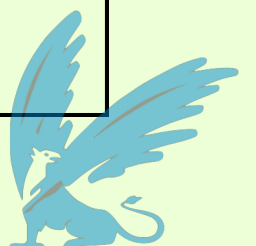
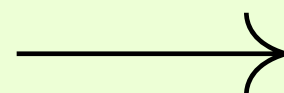


Use Throttling

Throttling can prevent your API from being overwhelmed by too many requests at once.

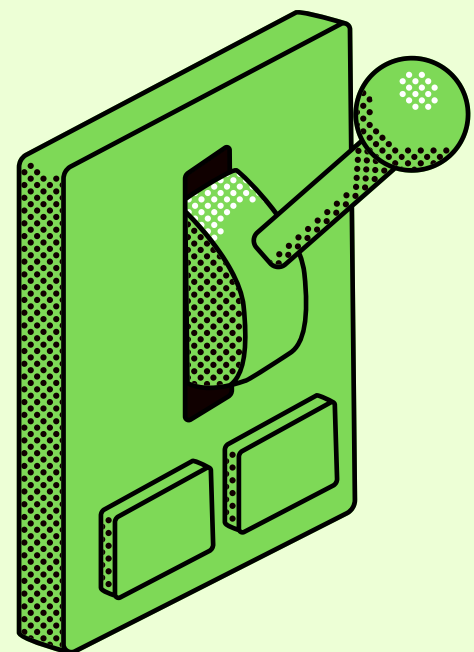


@princemeghani	→
----------------	---

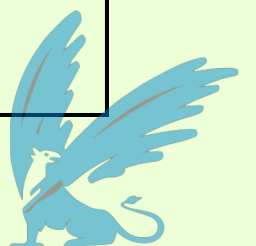
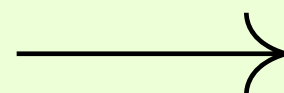


Fail Fast with Circuit Breaker

This technique can prevent a function that is likely to fail from being executed, which can improve the overall performance of your API.



@princemeghani	→
----------------	---

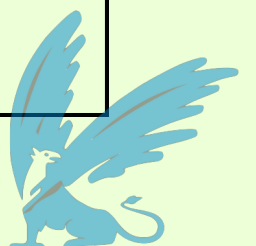
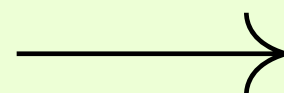


Use HTTP/2 Instead of HTTP

HTTP/2 can provide performance benefits over HTTP, such as header compression and multiplexing.

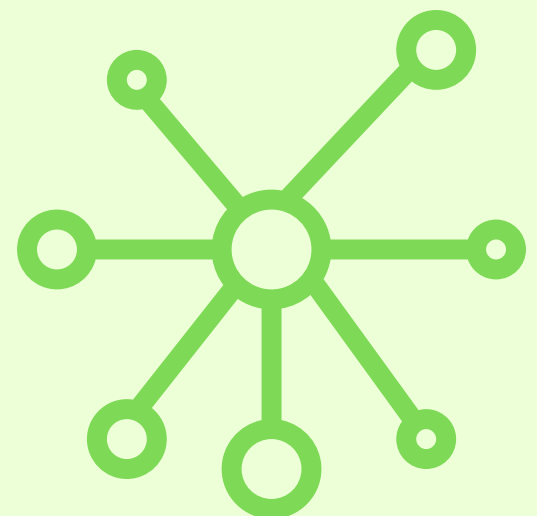


@princemeghani	→
----------------	---

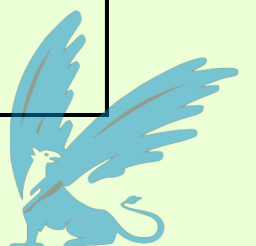
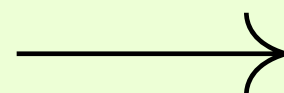


Optimize APIs with PM2 Clustering

PM2 is a production process manager for Node.js applications with a built-in load balancer. It allows you to keep applications alive forever and reload them without downtime.



@princemeghani	→
----------------	---

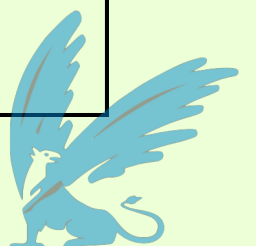
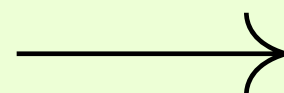


Reduce TTFB (Time to First Byte)

This is the time it takes for the client to receive the first byte of data from the server. Reducing TTFB can improve the perceived performance of your API.



@princemeghani	→
----------------	---

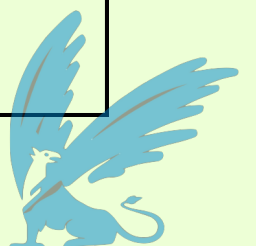
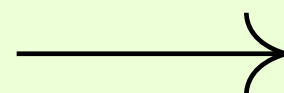


Run Tasks in Parallel

This can help to improve the performance of I/O operations.



@princemeghani	→
----------------	---

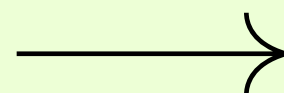


Use Error Scripts with Logging

This can help you quickly identify and fix issues.

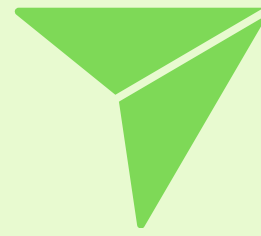


@princemeghani	→
----------------	---



Thank you for Reading

Follow me for more useful information!



@princemeghani