

Overview

[Question 4, responses to What n](#)
[Question 3, Elaboration on Cons](#)

Sheet 1: Question 4, responses to What n

What needs do you have which are not covered in the list?

To know that there is sufficient mindshare and economic stability that node will survive for the next 10 years

Good Type Information

Game develop

The above selected would be good enough, for starters.

a built in module that can replace http backend frameworks

not having politics injected into every facet of my life

Bring standard library such other languages like PHP(SPL).

Multi-version support on the same system (eg. nvm) is critical to my workflow

typescript execution with transparent compilation at runtime

Easy way to scale project under highload

Keeping up with the pace of JS ecosystem

First class support for yarn

Innovation as fast as possible. Quick v2 of some API's, less C-like APIs and code practices, safe types, promotion of best practices in terms of maintainable and portable code

I want Node.js to migrate to TypeScript

Remote debugging in k8s

enterprise dependency management to build predictable dependency trees

simplification of APIs

Epic

More inbuilt packages like Auth

Embedded GDB (debugging server) browser agnostic - VueJs like debug plugin with debug module plugin api

t

1. Cleaner syntax and ecosystem 2. Cleaner async, both promises as well as async a wait suck

Nothing else to add

CJS can require native ESM modules

Scalability and Performance

Ability to "compile" into a single executable file.

support of framework meteor

Longer LTS. Minimization of BC breaks.

Good interoperability with other languages/binaries

Good reference documentation

add more support library for hardware support e.d fingerprint

Proper Multithreading, and more detailed Documentation, more easy to browse and understand

Sustainable computing

Faster innovation

Improved core reliability generally (I have frequently found & reported bugs)

Thats all

ability to get together in-person with the community

better version managemens especially at windows machines

First class typescript / Optional types

Some of the items above are not clear!

Do NOT include more "relevant APIs" in core

Further documentation of "hidden" APIs.

Compatibility with web APIs and/or browser code

Management of the runtime and security updates

Rust native addons

More core tools to minimize dependencies and ecosystem issues.

Modules permissions (ex allow "axios" to run HTTP requests, but deny "is-even")

Stable Language! Stable Language! Stable Language!

Stable Foreign Language integration

Keeping core reasonably small; better consistency between APIs; security features (isolates, disable core modules, risk mitigation in dependencies, code signing); static analysis tooling

Simplified & newer API's (Example: Switch all API's to promise based)

Cross platform, easy, reliable installation/build of native modules

Improved ways to manage 3rd party code risks (i.e. to make sure modules that don't need FS access don't get it)

solid support ES standard, specially modules import/export

shared memory multithreading

Easy to use typescript

6666

Performance improvement or check test for every Node release

type safety

official binary compile support

package security

Backward compatibility

Framework ecosystem development

I think it would be great if something like nvm were part of node itself

Bundle a linter in core

Examples in documentation - very lacking, always have to search google for full usage examples from a 3rd party

TypeScript support

optional central package cache on computer to reduce disk usage (like deno does)

A way to add community contributed examples to core documentation, see PHP documentation. PHP sucks, but their docs are top tier.

Anything that supports the hapijs framework as the most comprehensive back end node framework

Confidence in security of widely used "should really be core" npm packages

Security from the outset.

Improved VM sandboxing

Package manager ecosystem improvements

Better communication of the roadmap with non-maintainers

Integration with web api

Packaging and distribution for deployments

That's it

- Analysis tools
- NA
- Fast native code speed in my nodejs code. Webassembly maybe?
- Performance!
- easy concurrency like go, compile to executable
- Ability to provide Feedback or Report Bugs (security or normal) WITHOUT LOGGING IN -- telling me to go to GitHub or HackerOne is NOT good enough /// Unrelated, but: cPanel / CloudLinux / LiteSpeed use old versions of node! (my host doesn't have v14 or v15)
- Native Typescript support?
- Better support for canary builds for Windows
- Take leadership in package managers (yarn Vs npm) and builders/bundlers to have one standard, not new one every year (like in many other programming languages)
- Strict and comprehensive adherence to ECMAScript standards
- High performance
- More batteries included

Sheet 2: Question 3, Elaboration on Cons

If you answered yes to question 2, please provide details about which constituency you belong to that is not covered in the list.
Open-Ended Response

Service and infrastructure providers

Student

Manage team that does Node.js development.

Nothing

Vue

Game develop

Students who are learning

Avid Fan. ;)

Developer from Africa

Application Security

I have submitted a few PRs that were merged into core, but I do not consider myself a core "maintainer" at this time.

requirements management

throw new Error("HELLO WORLD ") ;

None that I can think of.

Lok Sabha Constituency 550 India

Founder/think tank

Data analytics

Application developer

Alternative Javascript Runtime Maintainer

Teaching Node.js at university level

api custom service

Instrumentation developers working to create application runtime diagnostic tools

Command line tools and applications

Décision maker on use of technologies including nodeJS

Technical executive (CTO) whos team utilizes Node.js in application development, tooling development and devops pipelines.

6666

Scripting users - I feel that people who are using Node for scripting (especially small scripts ie. bash replacement) should be captured in a separate constituency.

Guten morgen

Support and Issues Analysis

Standards groups (like TC39) that node implements, strives/claims to support, and who rely on node's input and use cases.