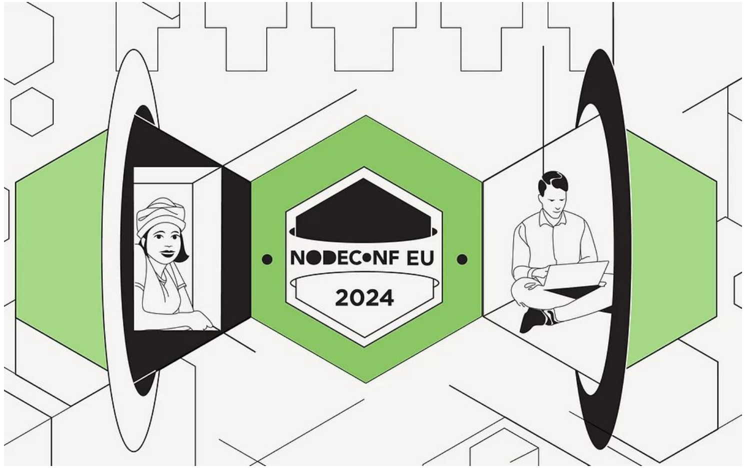


What’s new and what’s next

About Michael Dawson

Node.js lead for Red Hat and IBM

Active Node.js community member

- Node.js Collaborator, Node.js Technical Steering Committee,
- Active in a number of Working group(s)

Active OpenJS Foundation member

- Voting Cross Project Council Member
- Community Director 2021-2022

Twitter: @mhdawson1

GitHub: @mhdawson

Linkedin: https://www.linkedin.com/in/michael-dawson-6051282

Overview

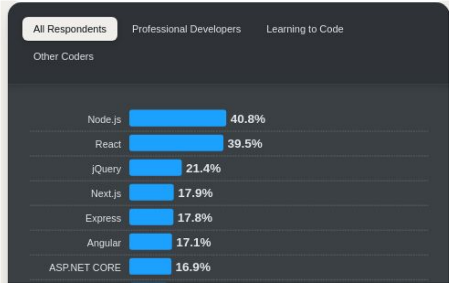
- What’s Not New
- Recent survey
- What’s new
- What’s next

What’s not new

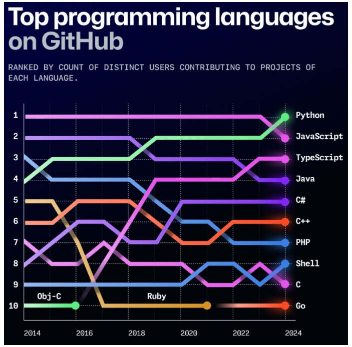
* Even after docker activities pushing people to cache

- Massive usage in last year
  - Over 1B downloads from nodejs.org
  - Almost 800M docker pulls* dockerhub

- Tops recent surveys



https://survey.stackoverflow.co/2024/technology/

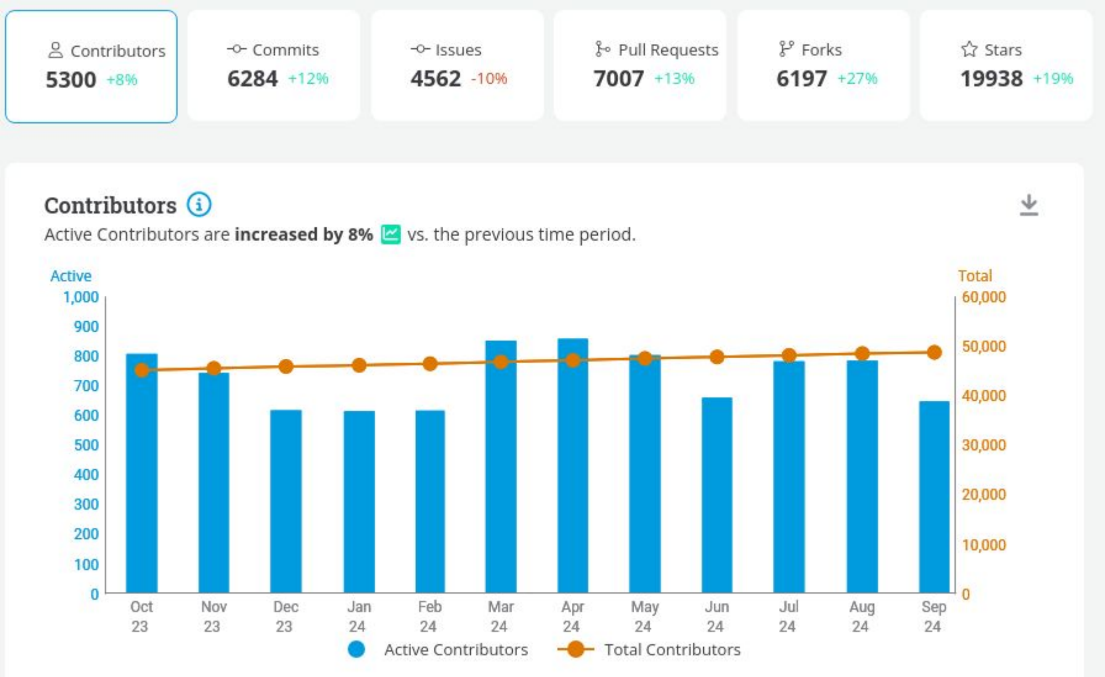


2 - JavaScript
3 - typeScript
Notes that combined still higher than Python

https://github.blog/news-insights/octoverse/octoverse-2024/

What’s not new

- Active and vibrant project



Contributors	Commits	Issues	Pull Requests	Forks	Stars
5300 +8%	6284 +12%	4562 -10%	7007 +13%	6197 +27%	19938 +19%

Contributors

Active Contributors are **increased by 8%** vs. the previous time period.

https://insights.lfx.linuxfoundation.org/foundation/ojsf/overview/github?project=ojsf-nodejs (Oct 1 2023 - Sep 30 2024)

What’s not new

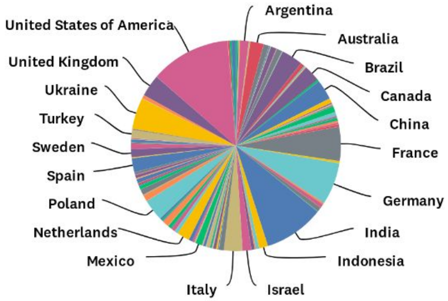
- Relevant to major trends - Large Language Models
  - developer libraries often JavaScript/Typescript 2nd language supported

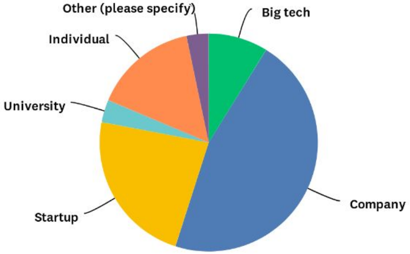
LangChain LlamaIndex ollama-js

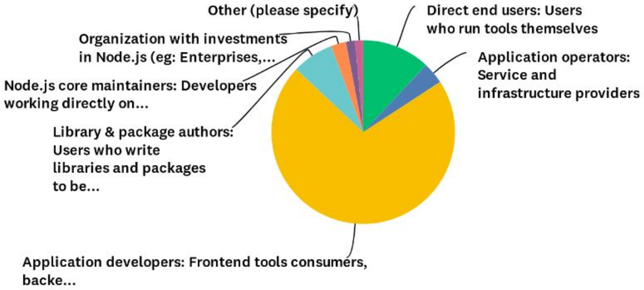
* Even after docker activities pushing people to cache

Recent Survey

- 2978 respondents, 81% more than last year
- Average of 6 years using Node.js

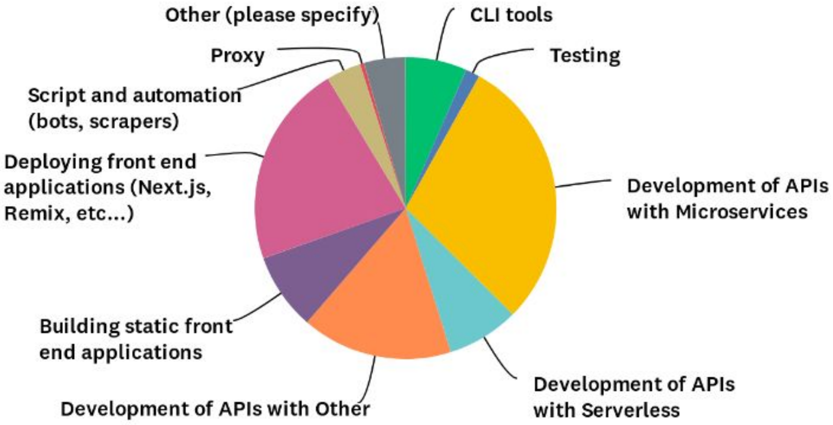






https://github.com/nodejs/next-10/blob/main/surveys/2024-04

Recent Survey

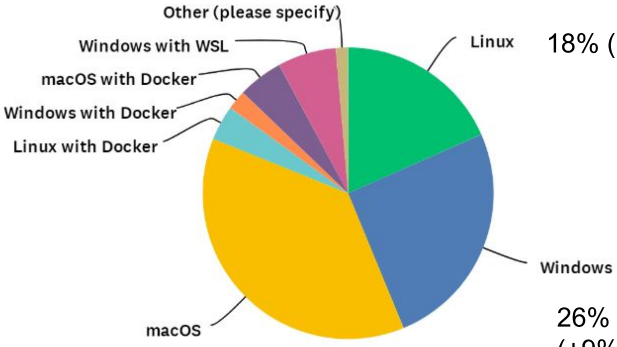


Use Cases

- 53% back end services
- 22% deploying front end apps
- 21% building, and scripting

https://github.com/nodejs/next-10/blob/main/surveys/2024-04

Recent Survey

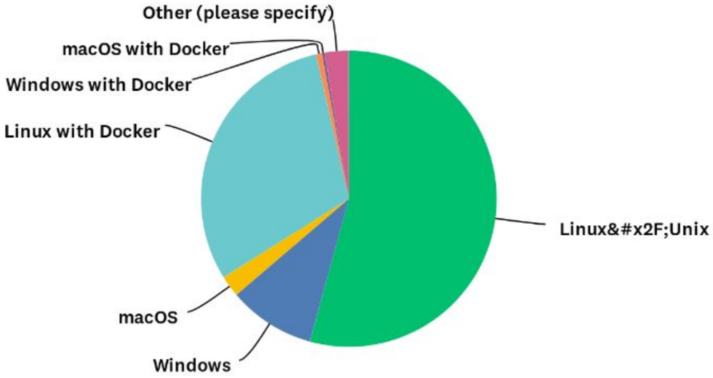


18% (+11%)

26%
(+9%)

38%
(+5%)

Local Development



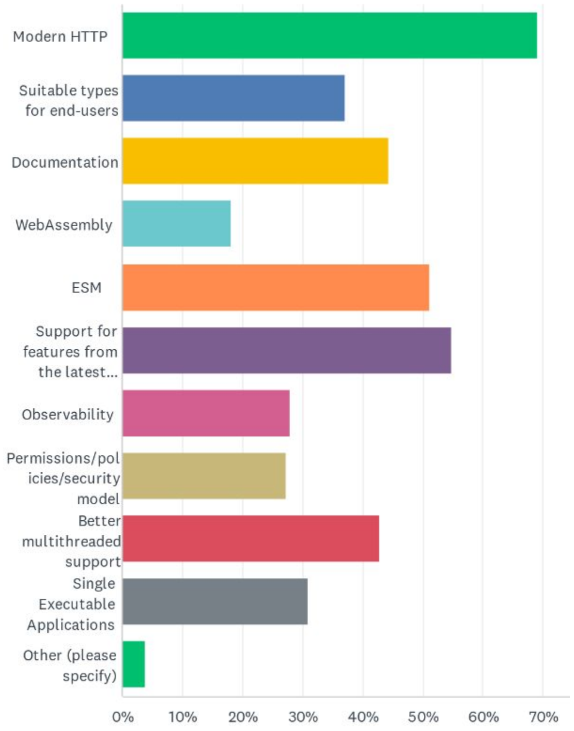
Production

x86 dominates but Arm is growing

https://github.com/nodejs/next-10/blob/main/surveys/2024-04

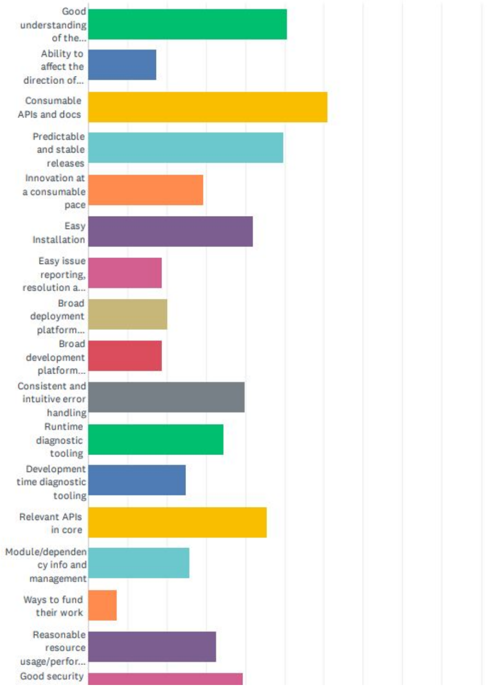
Recent Survey

Technical Priorities



What's important to you

(incomplete list shown)



Recent Survey

- Key Takeaways
  - Most common pain points
    - ESM integration
    - TypeScript
    - Docs
    - Monitoring/Diagnostics
    - Performance
  - ESM widely used, 71% reported code with ES module syntax
  - New stable features well used
  - Experimental features well used

https://github.com/nodejs/next-10/blob/main/surveys/2024-04

What's new

- 2024 was Exciting year
- Some Highlights
  - Progress on ESM compatibility
  - Experimental Limited TypeScript Support
  - Experimental Web Storage API
  - Experimental SQLite API
  - Continued progress on performance
  - Node run
- Regular Releases

Most common pain points

- ESM integration ✅
- TypeScript ✅
- Docs
- Monitoring/Diagnostics
- Performance ✅

Releases

Last Major Release

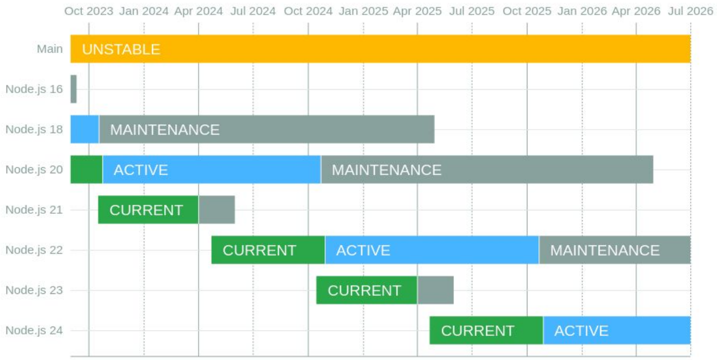
- 23.x in October 2024 (Curent)

Ongoing LTS Releases

- Node.js 18.x - ends April 2025
- Node.js 20.x - ends April 2026
- Node.js 22.x - ends April 2027

Next current

- 24.x in April 2024



https://github.com/nodejs/Release

ESM Compatibility

hello.js

```
const greeter = require('./greeter.js').default;

const greeting = greeter('World');
console.log(greeting);
```

CJS

greeter.js

```
import util from 'node:util';
export default function (greeting) {
  return 'Hello ' + greeting;
};
```

ESM

It just works!

```
>node hello.js
Hello World
```

The Fine Print:

Depends on two features that are still experimental but stability is “release candidate”, and no experimental warnings.

Options to turn of in case of issues

--no-experimental-require-module

-> ESM must be synchronous

--no-experimental-detect-module

-> depends on syntax detection - https://nodejs.org/api/packages.html#syntax-detection
-> may load file twice
-> recommended to use "type": "module" in package.json where possible

Experimental Limited Typescript Support

- At a glance
    - --experimental-strip-types
        - replaces TypeScript with whitespace
            - no need for source maps
        - .ts, .mts, .cts files supported
    - --experimental-transform-types
- Key limitations
    - Only supports a subset of TypeScript
    - Does not handle files in node_modules (design choice)
    - Does not read tsconfig.json
    - type keyword needed in imports

https://nodejs.org/api/typescript.html - Stability: 1.0 - Early Development

Experimental Limited Typescript Support

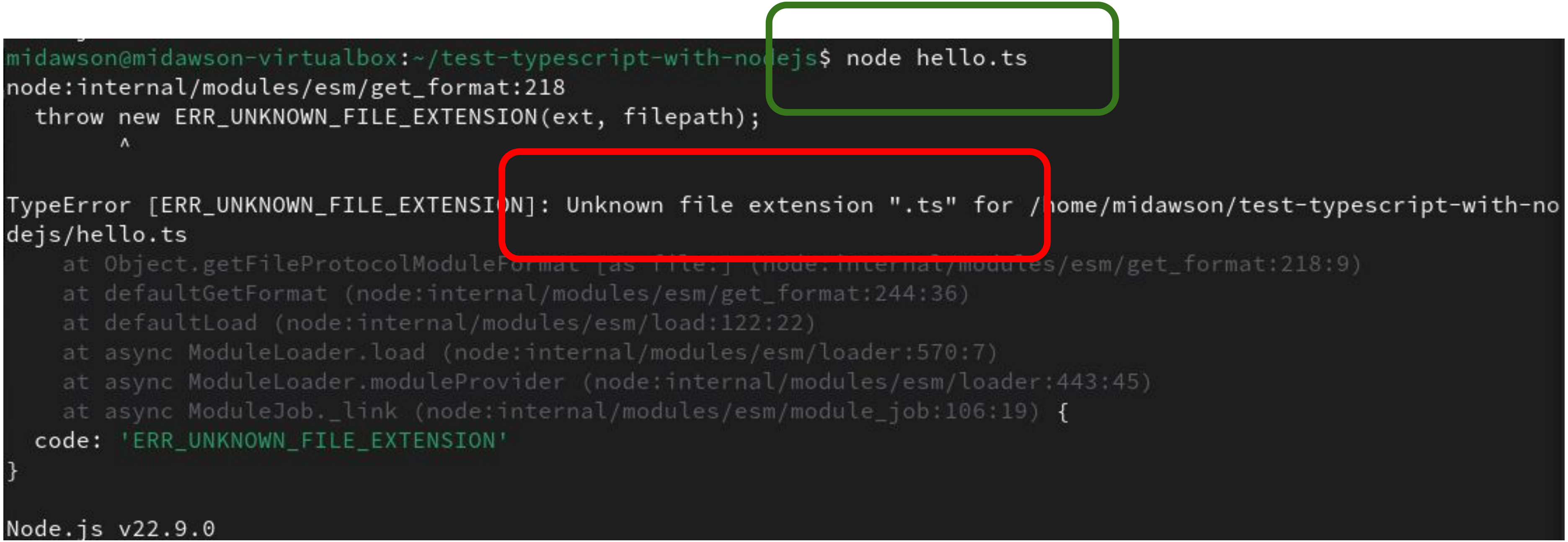
hello.ts

```
import { GreetOptions } from
'./greeter.ts'
import greeter from './greeter.ts'

const message: string = 'World';
const greeting: string =
greeter(GreetOptions.Hello,
        message);

console.log(greeting);
```

greeter.ts

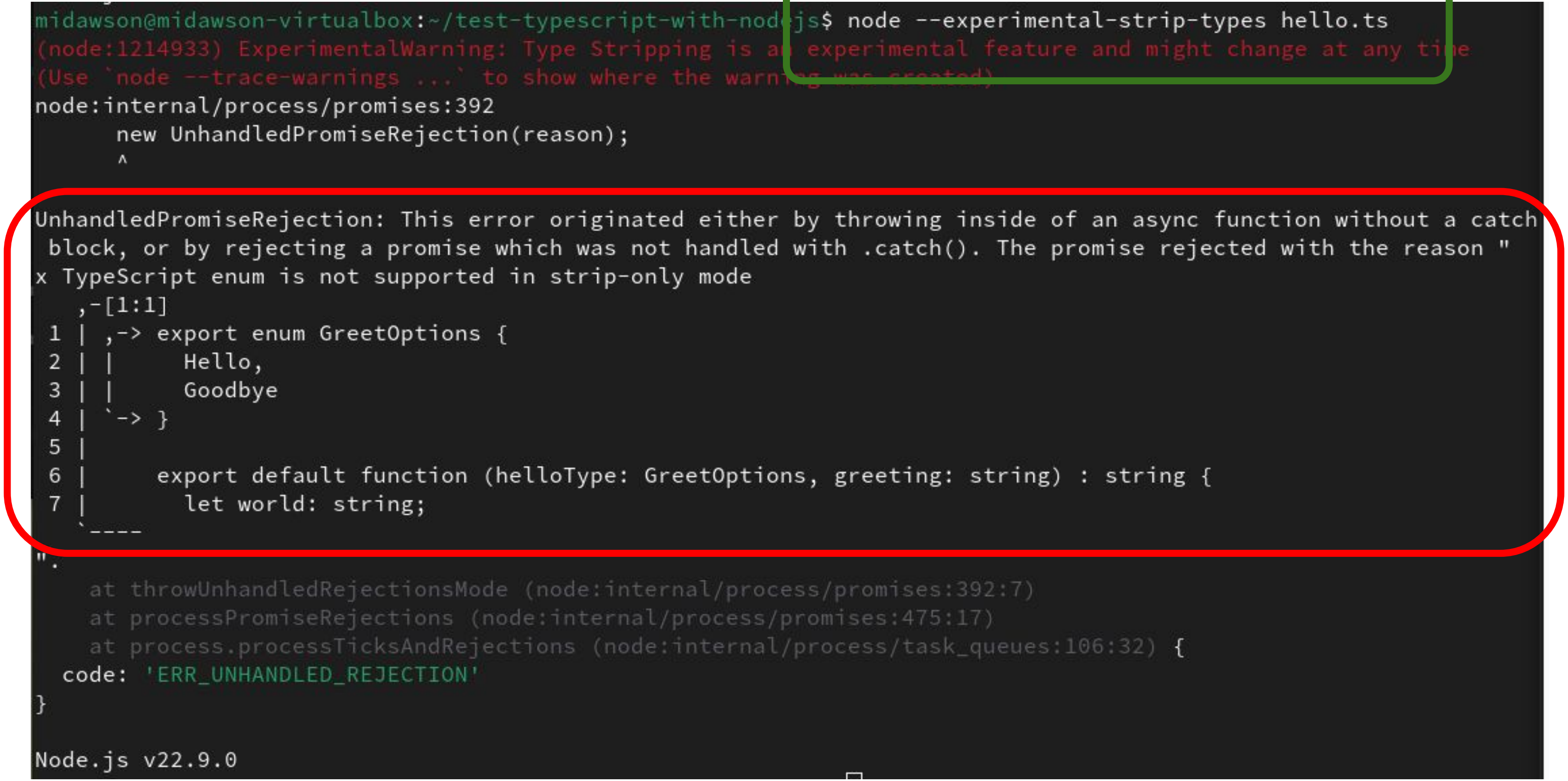
```
export enum GreetOptions {
  Hello,
  Goodbye
};

export default function (helloType:
GreetOptions, greeting: string) : string {
  let world: string;
  if (helloType === GreetOptions.Hello) {
    world = 'Hello ' + greeting;
    return world;
  };
  throw Error('Unknown GreetOption');
};
```

Experimental Limited Typescript Support



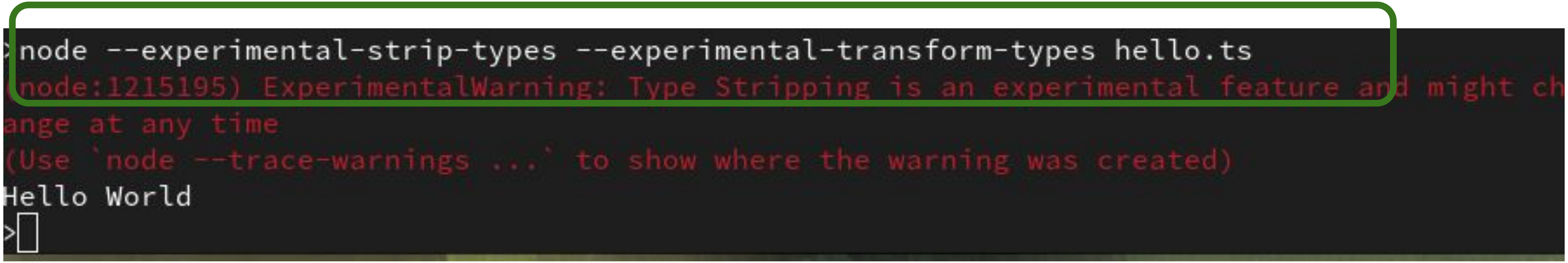
```
midawson@midawson-virtualbox:~/test-typescript-with-nodejs$ node hello.ts
node:internal/modules/esm/get_format:218
  throw new ERR_UNKNOWN_FILE_EXTENSION(ext, filepath);
        ^

TypeError [ERR_UNKNOWN_FILE_EXTENSION]: Unknown file extension ".ts" for /home/midawson/test-typescript-with-no
dejs/hello.ts
    at Object.getFileProtocolModuleFormat [as file:] (node:internal/modules/esm/get_format:218:9)
    at defaultGetFormat (node:internal/modules/esm/get_format:244:36)
    at defaultLoad (node:internal/modules/esm/load:122:22)
    at async ModuleLoader.load (node:internal/modules/esm/loader:570:7)
    at async ModuleLoader.moduleProvider (node:internal/modules/esm/loader:443:45)
    at async ModuleJob._link (node:internal/modules/esm/module_job:106:19) {
  code: 'ERR_UNKNOWN_FILE_EXTENSION'
}

Node.js v22.9.0
```

Experimental Limited Typescript Support



```
midawson@midawson-virtualbox:~/test-typescript-with-nodejs$ node --experimental-strip-types hello.ts
(node:1214933) ExperimentalWarning: Type Stripping is an experimental feature and might change at any time
(Use `node --trace-warnings ...` to show where the warning was created)
node:internal/process/promises:392
      new UnhandledPromiseRejection(reason);
      ^

UnhandledPromiseRejection: This error originated either by throwing inside of an async function without a catch
 block, or by rejecting a promise which was not handled with .catch(). The promise rejected with the reason "
x TypeScript enum is not supported in strip-only mode
   ,-[1:1]
 1 | ,-> export enum GreetOptions {
 2 | |     Hello,
 3 | |     Goodbye
 4 | `-> }
 5 |
 6 |     export default function (helloType: GreetOptions, greeting: string) : string {
 7 |       let world: string;
   `----
".
    at throwUnhandledRejectionsMode (node:internal/process/promises:392:7)
    at processPromiseRejections (node:internal/process/promises:475:17)
    at process.processTicksAndRejections (node:internal/process/task_queues:106:32) {
  code: 'ERR_UNHANDLED_REJECTION'
}

Node.js v22.9.0
```

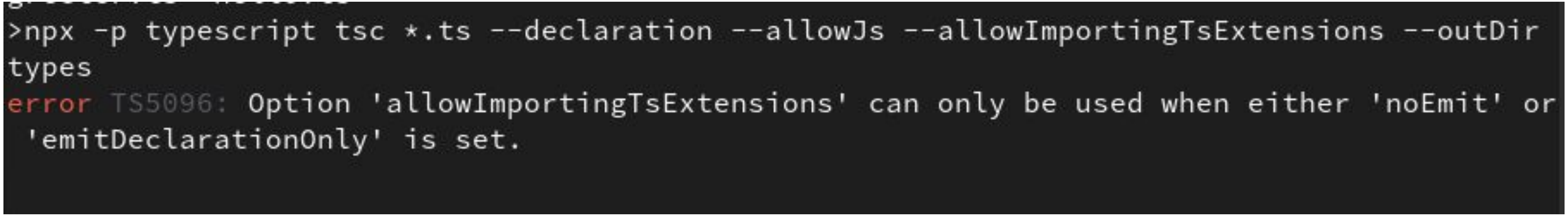
Experimental Limited Typescript Support



```
>node --experimental-strip-types --experimental-transform-types hello.ts
(node:1215195) ExperimentalWarning: Type Stripping is an experimental feature and might ch
ange at any time
(Use `node --trace-warnings ...` to show where the warning was created)
Hello World
>
```

Experimental Limited Typescript Support

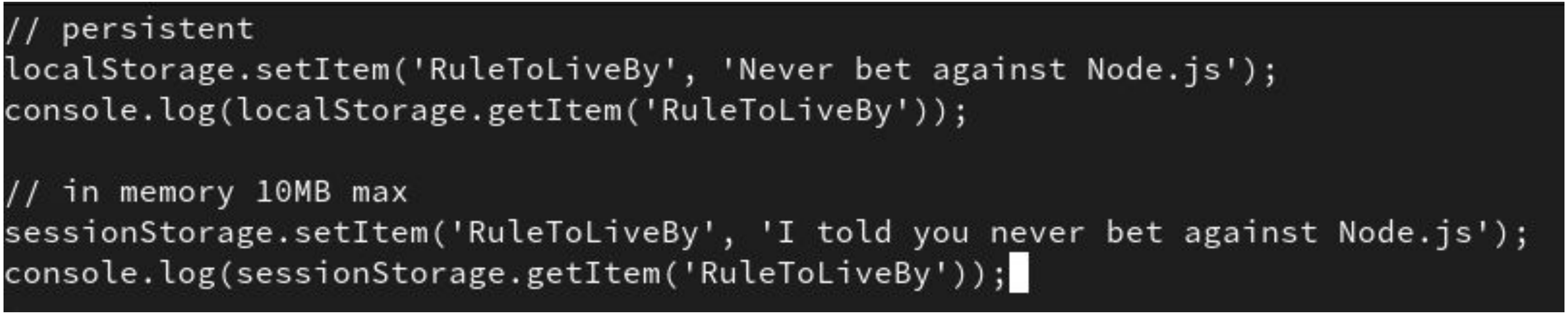
- TSC ~~has~~/had trouble with .ts extensions in imports



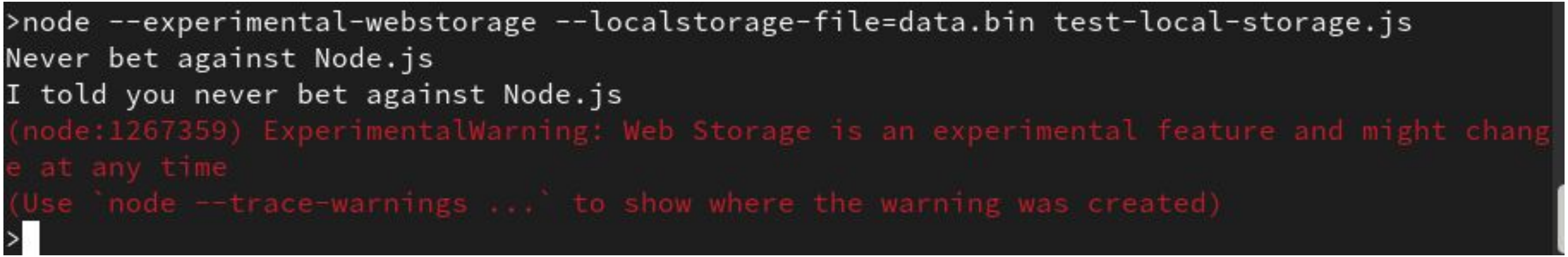
```
>npx -p typescript tsc *.ts --declaration --allowJs --allowImportingTsExtensions --outDir
types
error TS5096: Option 'allowImportingTsExtensions' can only be used when either 'noEmit' or
 'emitDeclarationOnly' is set.
```

- Node.js Typescript team -
nodejs / typescript
  - Members from typescript,Node.js collaborating on issue
  - PR landed to handle https://github.com/microsoft/TypeScript/pull/59767
    - --rewriteRelativeImportExtensions

Experimental Web Storage API

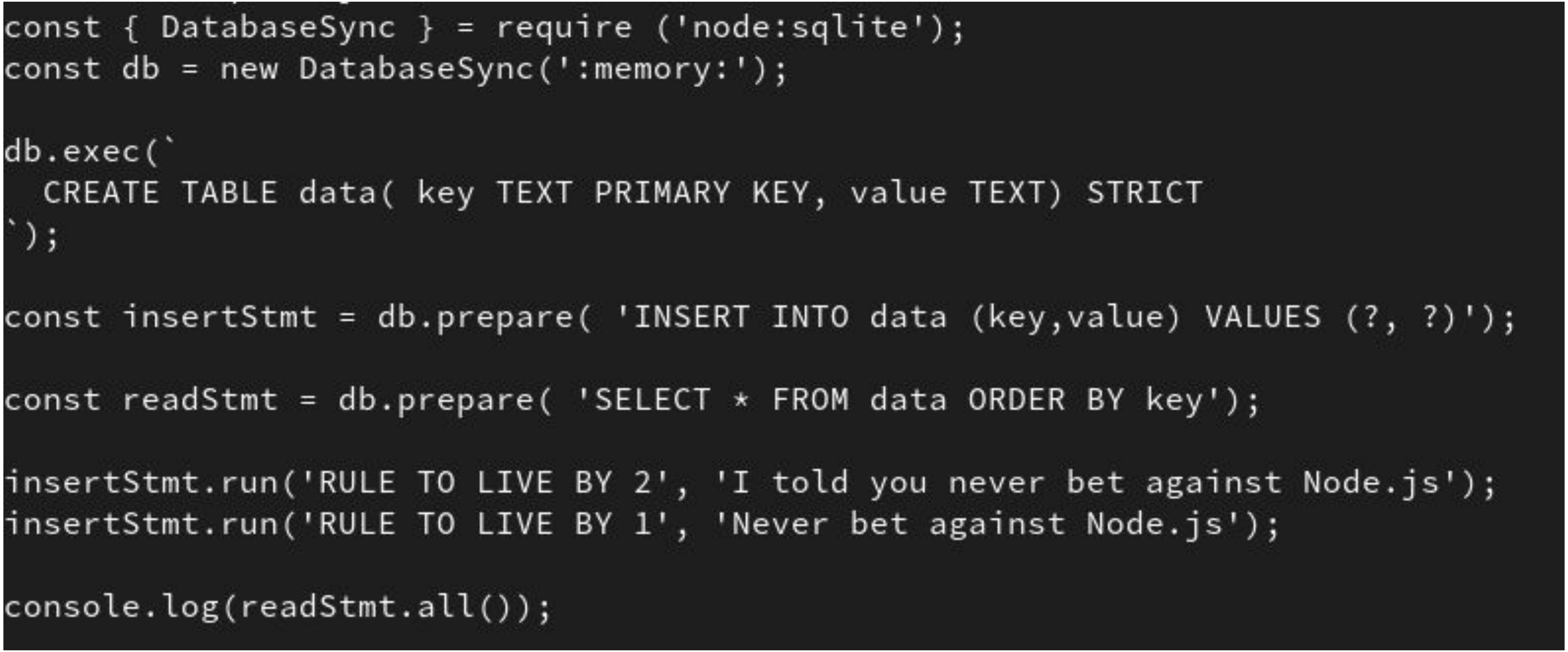


```
// persistent
localStorage.setItem('RuleToLiveBy', 'Never bet against Node.js');
console.log(localStorage.getItem('RuleToLiveBy'));

// in memory 10MB max
sessionStorage.setItem('RuleToLiveBy', 'I told you never bet against Node.js');
console.log(sessionStorage.getItem('RuleToLiveBy'));
```



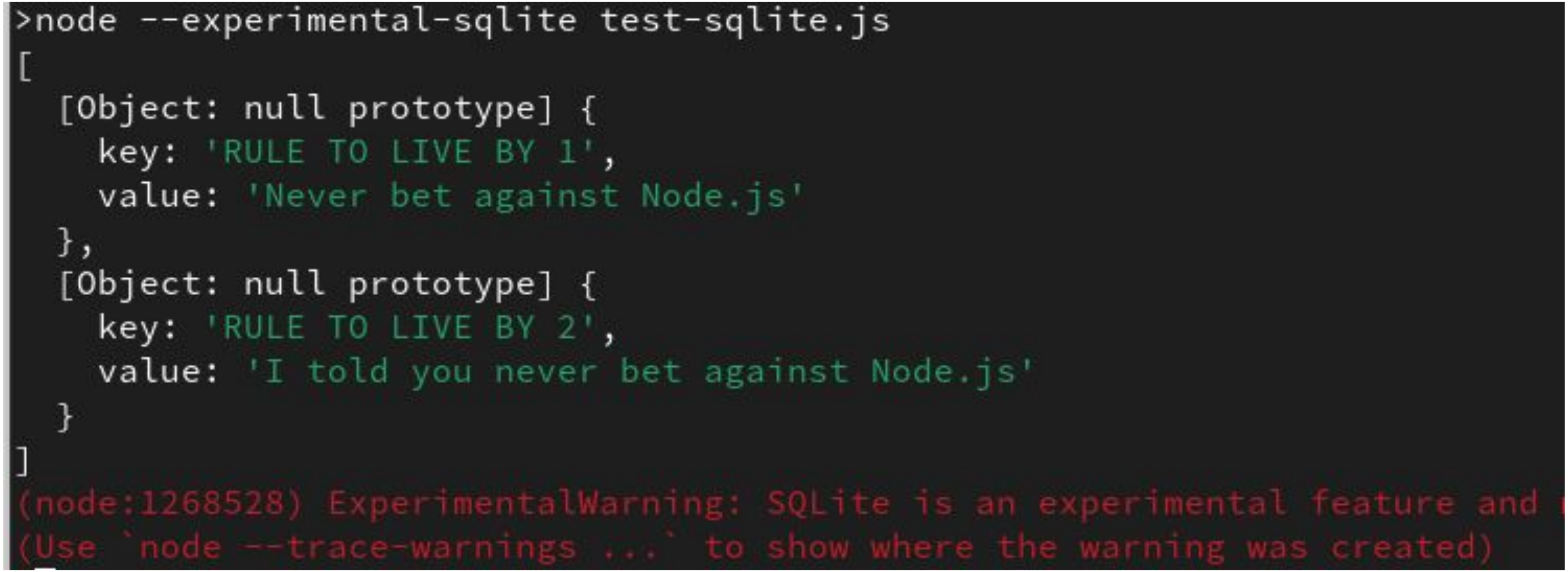
```
>node --experimental-webstorage --localstorage-file=data.bin test-local-storage.js
Never bet against Node.js
I told you never bet against Node.js
(node:1267359) ExperimentalWarning: Web Storage is an experimental feature and might chang
e at any time
(Use `node --trace-warnings ...` to show where the warning was created)
>
```

Experimental SQLite API



```
const { DatabaseSync } = require ('node:sqlite');
const db = new DatabaseSync(':memory:');

db.exec(`
  CREATE TABLE data( key TEXT PRIMARY KEY, value TEXT) STRICT
`);

const insertStmt = db.prepare( 'INSERT INTO data (key,value) VALUES (?, ?)');

const readStmt = db.prepare( 'SELECT * FROM data ORDER BY key');

insertStmt.run('RULE TO LIVE BY 2', 'I told you never bet against Node.js');
insertStmt.run('RULE TO LIVE BY 1', 'Never bet against Node.js');

console.log(readStmt.all());
```

Experimental SQLite API



```
>node --experimental-sqlite test-sqlite.js
[
  [Object: null prototype] {
    key: 'RULE TO LIVE BY 1',
    value: 'Never bet against Node.js'
  },
  [Object: null prototype] {
    key: 'RULE TO LIVE BY 2',
    value: 'I told you never bet against Node.js'
  }
]
(node:1268528) ExperimentalWarning: SQLite is an experimental feature and
(Use `node --trace-warnings ...` to show where the warning was created)
```

Continued progress on performance

- Newer versions of V8
- On-disk caching to improve startup
  - NODE_COMPILE_CACHE, NODE_DISABLE_COMPILE_CACHE
- Buffer improvements
- FS improvements

Node --run

Package.json

```
{
  "name": "my-great-app",
  "version": "0.0.1",
  "scripts": {
    "start": "node doit.js"
  }
}
```

```
>node --run start
I ran
```

What’s Next

What’s Next

- GitHub Notifications
- Working Groups and Teams - https://nodejs.org/calendar
- Strategic Initiatives - https://github.com/nodejs/node/blob/master/doc/guides/strategic-initiatives.md
- Next-10 - https://github.com/nodejs/next-10
- Releases - https://github.com/nodejs/release
- Node.js news feed https://nodejs.github.io/nodejs-news-feeder/feed.xml

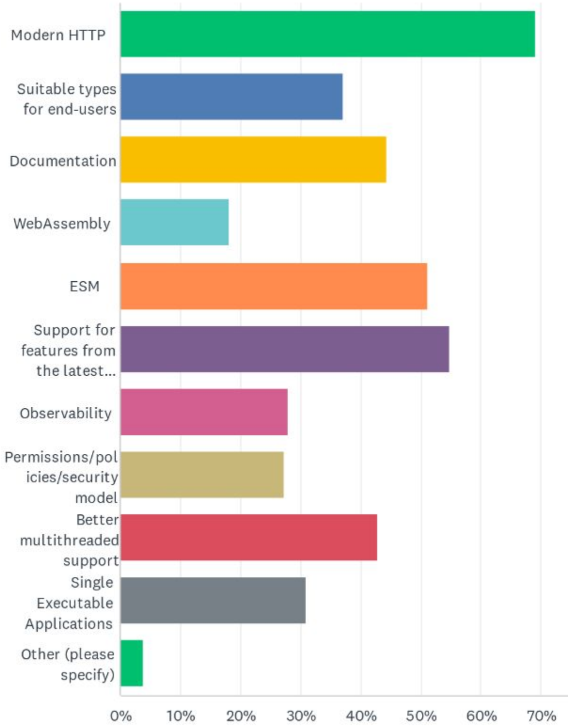
Next 10 team

- Technical priorities well defined
  - Aligned based on survey results
- Current focus areas less technical
  - Advocacy and promotion
    - Ambassador program
    - Corporate usage stories
    - News Feed - https://nodejs.github.io/nodejs-news-feeder/feed.xml
  - Funding
  - Collaborator Health

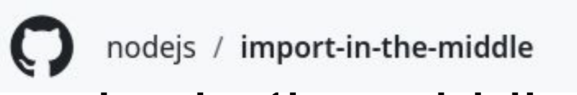
https://github.com/nodejs/node/blob/main/doc/contributing/advocacy-ambasador-program.md

What’s next

- Continued work on
  - ESM
  - Typescript
  - Performance
  - QUIC
  - …



What’s Next?

- What about other pain points ?
  - Docs Initiative - https://github.com/nodejs/next-10/issues/166
  - Monitoring/Diagnostics
    - diagnostics team
      - AsyncLocalStorage rewritten not to use Async Hooks
    - Key components moved into org
      - 
        nodejs / import-in-the-middle
      - require-in-the middle ?

Most common pain points

- ESM integration
- TypeScript
- Docs ✅
- Monitoring/Diagnostics✅
- Performance

Thank You

Thank You

Questions?

Copyright and Trademarks