

# Fully agentic coding with OpenAI Codex, JavaScript's new superpower, Deno's update next-gen web framework, and more - Week #35



JAM.DEV  
MAY 18, 2025

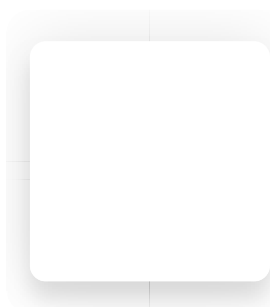


Hello JavaScript Enthusiasts!

Welcome to a new edition of "This Week in JavaScript"!

Today, we're exploring OpenAI's groundbreaking Codex system, JavaScript's new resource management capabilities, and Basecoat's framework-agnostic UI components, plus five exciting new tools to code and develop faster!

Can't read the newsletter? Listen on the go!



May 18 · [This Week in JavaScript](#)

Save on Spotify

03:22

© 2025 This Week In JavaScript · [Privacy](#) · [Terms](#) · [Collection notice](#)  
[Substack](#) is the home for great culture

support our work.

Subscribe

## OpenAI Codex is Finally Here

[Codex](#) is OpenAI's software engineering agent that helps developers, especially web and JavaScript developers-automate coding tasks, boost productivity, and handle multiple tasks securely. Each task runs in its own cloud sandbox environment, preloaded with your repository.

### Built for developers:

Codex excels in JavaScript and frameworks like React, Node.js, and TypeScript. It can generate, refactor, debug code, write tests, and help understand codebases for both frontend and backend projects.

### Technical details:

Powered by the codex-1 model and trained via reinforcement learning on millions of coding tasks, Codex runs in secure, isolated cloud containers. It takes natural language or code prompts, generates code, and runs tests and linters automatically until pass

### Parallel task handling:

Codex can manage multiple tasks at once, such as adding features, fixing bugs, and running tests, speeding up development and reducing interruptions.

### Use cases:

- Prototyping React components with accessibility
- Migrating JavaScript to TypeScript
- Automating Node.js CI/CD pipelines

- Enhancing test coverage and debugging
- Helping non-developers with small fixes
- Supporting large refactors in companies like Cisco

### Security and compliance:

Runs in sandboxed environments without internet access, protecting sensitive data and including automatic vulnerability scanning.

### Limitations & future plans:

Some tasks take up to 30 minutes; no visual design or real-time collaboration yet. Future updates include browser-based execution, Figma integration, and multi-age coordination.

## JavaScript's New Superpower: Explicit Resource Management in V8 v13.8

[The V8 engine](#) now introduces Explicit Resource Management (ERM) — a feature long requested by developers seeking more control over resource cleanup.

Traditionally, JavaScript relied on garbage collection, making resource disposal non-deterministic. ERM changes that.

- **using Blocks:** Wrap resources like file handles or streams in `using/await using` blocks for automatic cleanup.
- **Custom Disposable Protocol:** Objects can implement a `[Symbol.dispose]()` or `[Symbol.asyncDispose]()` method to define how they release external resources.
- **Synchronous + Async Support:** This covers both sync and async resource management needs.

This gives developers more deterministic control — especially for scenarios involving file handles, streams, sockets, or GPU resources — all without waiting for GC to ca

up.

## Deno Fresh is (Still) Here, with a v2 Upgrade

[The Deno team just dropped a major update on Fresh](#), their edge-first full-stack web framework built for performance and simplicity. With V1.6 and beyond, Fresh is moving into a more mature phase of development.

- **New Build System:** Fresh now uses a faster Rust-based builder (Island Builder) to speed up dev experience and production builds.
- **Improved DX:** A simplified CLI, better error messages, and smarter routing improve developer onboarding and iteration speed.
- **New Plugin Architecture:** Official support for plugins and integrations, starting with Tailwind and Deno KV.

Focused on "island-based" rendering, Fresh delivers fast pages by sending minimal JavaScript to the client — ideal for edge environments like Deno Deploy. Fresh is increasingly becoming a strong choice for those who want performance without giving up on Developer Experience.

## Node.js Released Critical Security Updates

Heads up, Node.js developers! [Critical security patches](#) are out for multiple versions — including v24.0.2, v23.11.1, v22.15.1, and v20.19.2. If you're running on Node, this one's important.

- **Cryptographic Fix:** Resolves crashes during TLS and encryption operations, essential for secure data handling.
- **HTTP Header Bug:** Prevents request manipulation and protects against

accidental data leaks.

- **Memory Leak Patch:** Improves performance for long-running servers and background services.

If you're on an LTS version, this is a must-do update. Don't skip it!

## Tools & Releases You Should Know About

Let's speed-run through some of the other big tool updates this week!

- [diff v8.0.1](#): The popular text diffing library now offers improved performance for large datasets, better error handling, and an updated online demo—perfect for JavaScript-based CLIs or code editors.
- [React Aria Components v1.3](#): Enhanced accessibility for React apps with new components for modals and tooltips, plus better keyboard navigation—ideal for building inclusive user experiences.
- [UnJS Nitro v2.9](#): This server toolkit now offers even faster cold starts, enhanced TypeScript support for stricter type safety, and improved middleware handling—perfect for JavaScript server-side projects.
- [Tamagui v1.126.13](#): A UI library for React Native and web, now with improved performance for complex layouts, new adaptive components for responsive design, and enhanced theming options—your cross-platform UI just got even better.

And that's it for the thirty-fifth issue of "This Week in JavaScript", brought to you by [jam.dev](#)—the tool that makes it impossible for your team to send you bad bug reports.

Feel free to share this newsletter with a fellow developer, and make sure you're subscribed to get notified about the next issue.

Until next time, happy coding!

Thanks for reading This Week in JavaScript!  
Subscribe for free to receive new posts and  
support our work.

Type your email...	Subscribe
--------------------	-----------

[Previous](#)

[Next](#)

## Discussion about this post

Comments	Restacks
----------	----------



Write a comment...

