Oct 24, 2025 • 8 min

Why developers are leaving Next.js for TanStack Start, and loving it

The React world is full of opinions, and every few years a new framework changes how we think about building apps. Next.js has been the default choice for a long time, but recently, more developers have been moving toward TanStack Start.

Tessa Mero
Head of Developer Relations

SHARE

The React world is full of opinions, and every few years a new framework changes how we think about building apps. Next.js has been the default choice for a long time. It offered a clear path to production and made React easier to manage. But recently, more developers have been moving toward TanStack Start, a newer full-stack framework built by the same team behind TanStack Query and TanStack Router.

They're not leaving because Next.js is broken. They're leaving because TanStack feels lighter, clearer, and closer to plain React, and it gives them back control over how things actually work.

The growing frustration with Next.js

Many developers used to love Next.js for its simplicity. But as it grew, so did its complexity. The shift to the App Router, the introduction of React Server Components, and constant new patterns have made the framework harder to follow.

In community discussions, one developer summed it up: *"Next.js lost me with the constant changes and complexity. The benefits don't justify the cognitive overload."* Others mentioned how debugging has become confusing and how new features often break old patterns.

Another recurring concern is that Next.js feels tied to Vercel. Many devs appreciate Vercel as a company, but they feel the framework has become more about the platform than the developer. As one user said, *"It's tough to separate Next.js from Vercel. Most advantages people list are actually Vercel features."*

TanStack's foundation of trust

TanStack didn't appear out of nowhere. It grew from the success of TanStack Query (formerly React Query), a library that completely changed how developers handled state and server data. It replaced complex Redux setups with something that "just made sense."

That same idea of simplicity without shortcuts is what's driving interest in TanStack Start. Developers already trust the TanStack team because their tools consistently solve real problems without hiding what's going on behind the scenes.

As one developer put it, *"They did for routing and server logic what they already did for state management — made it simple, predictable, and type-safe."*

What TanStack Start actually is

TanStack Start is a full-stack React framework that builds on TanStack Router, powered by Vite for fast builds and Vite Env APIs, which supports Nitro among other deployment plugins. It combines client-side interactivity with server-side rendering, all while keeping things explicit and type-safe.

Key features developers like:

- Type-safe routing: Everything is validated at compile time. If a route or parameter changes, you'll know immediately.
- Server functions: You can write backend logic (like database queries or API calls) in the same codebase and call it safely from the client.
- Client-first rendering: You get SSR for performance, but navigation feels like a single-page app.

- Simple deployment: Works on Netlify, Cloudflare, Vercel, or your own Node server with minimal setup.

It's not trying to reinvent React. It's giving React a modern backbone that's easy to understand.

Why developers prefer it

Developers in the community consistently mention a few themes:

1. Less magic, more control.

   Frameworks like Next.js and Remix rely on "magic" conventions which is behaviors that happen automatically. TanStack Start avoids that. You choose how data loads, where it runs, and what gets rendered.
2. Feels closer to React.

   Many devs say that once they start using TanStack Start, they forget they're even in a framework. It feels like normal React with some helpful extras.
3. A smoother developer experience.

   Type-safe routes, integrated server functions, and predictable data fetching make debugging faster. One Reddit user said, *"When I first tried it, I thought I was missing something because it was so easy — it just worked."*
4. Flexible hosting and tooling.

   Using Vite and Nitro means you can build and deploy anywhere. You're not locked into one company's stack or runtime.

Real-world use cases

The developers experimenting with TanStack Start aren't just building toy projects. They're using it for production apps, internal dashboards, and even company websites.

A few examples mentioned in community threads:

- A real-time quiz app built for an event, where switching from React Router 7 to TanStack Start fixed hydration issues right before launch.
- A multi-language analytics dashboard powered by Laravel on the backend and TanStack Start on the frontend, praised for its easy SSR setup.
- Startups and indie projects migrating from Next.js because they wanted more predictable code and simpler local hosting options.

Many devs say TanStack Start lets them scale from side projects to production without losing control or adding complexity.

Why this shift matters

This shift isn't just about replacing one framework with another. It reflects a broader change in what developers value.

For years, frameworks have chased automation and abstractions and trying to do more for the developer. But that often came with confusion, slower builds, and less transparency. TanStack's approach goes the other way. It gives developers clarity, type safety, and freedom to understand and shape their stack.

As one developer wrote, *"TanStack brings back sanity. It's a framework that 99% of devs actually want to use."*

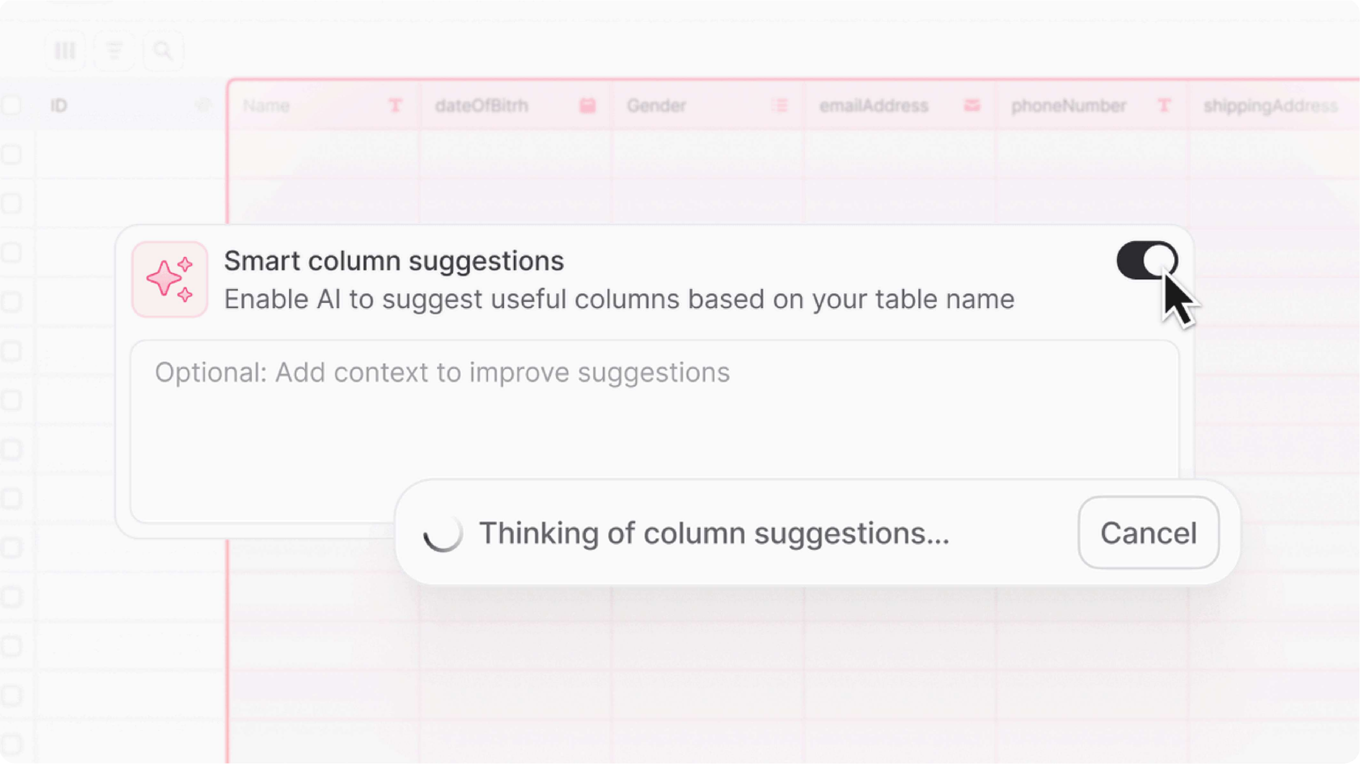
The takeaway

TanStack Start is still young, but it's already making an impact. Its community is growing quickly, its documentation is improving fast, and it's backed by a team with a proven record of building tools developers trust.

Next.js isn't going anywhere, but developers are looking for something that feels simpler and more honest. TanStack Start gives them that: a framework that's powerful without being heavy, modern without being controlling, and familiar without being boring.

If you've ever wished React felt like React again, you'll probably understand why developers are leaving Next.js for TanStack Start, and loving it.

Read next



Smart column suggestions

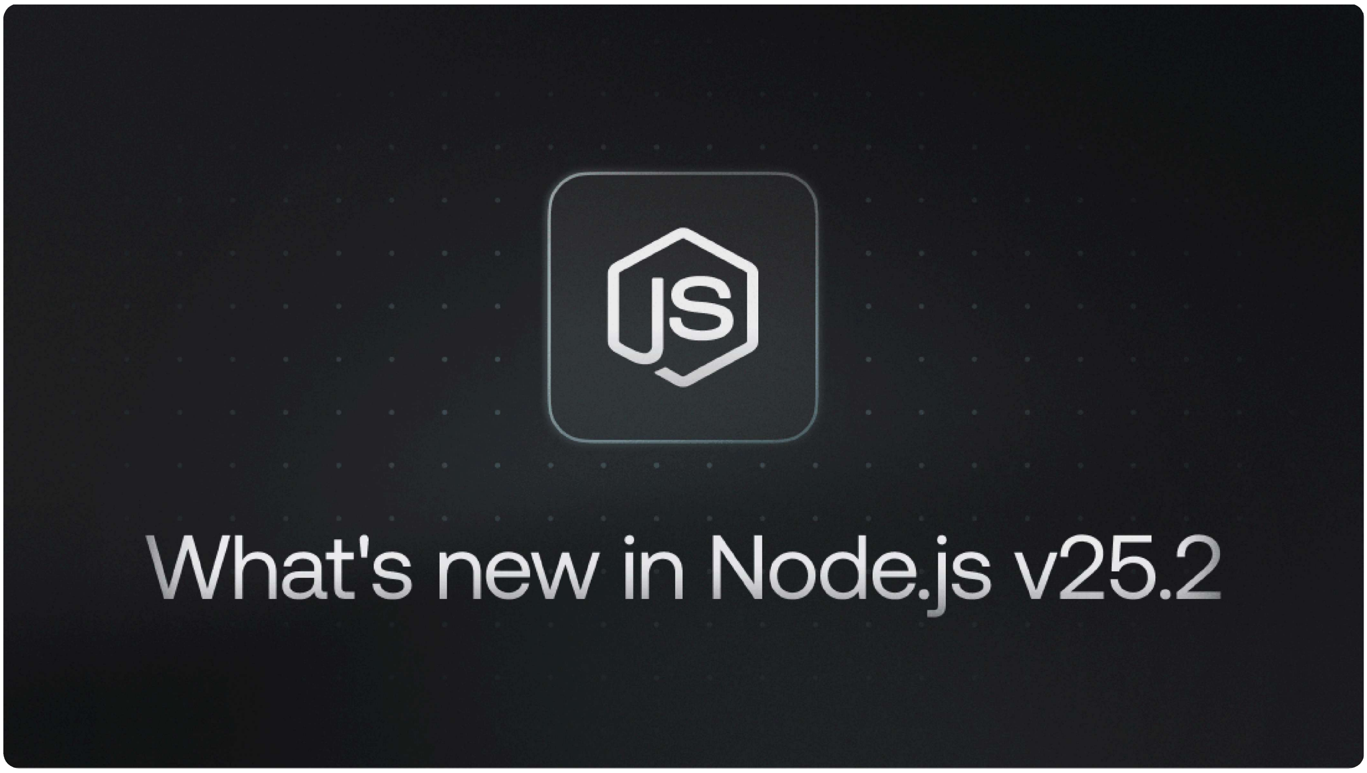
Enable AI to suggest useful columns based on your table name

Optional: Add context to improve suggestions

Thinking of column suggestions...

Announcing Database AI suggestions: from table name to schema in one click

Darshan Pandya Nov 28, 2025 5 min



What's new in Node.js v25.2: Web Storage, V8 14.1, permissions and more

Ebenezer Don Nov 25, 2025 9 min

Remix 3: what's changing and why it matters

Ebenezer Don Nov 18, 2025 8 min

Subscribe to our newsletter

Sign up to our company blog and get the latest insights from Appwrite. Learn more about engineering, product design, building community, and tips & tricks for using Appwrite.

Enter your email Sign up

appwrite

QUICK STARTS

PRODUCTS

LEARN

PROGRAMS

ABOUT

✓ All services are online

Terms Privacy Cookies