ucing WebContainers: ode.js natively in your ser

AY 20, 2021 NEWS & ANNOUNCEMENTS

e StackBlitz, an online IDE tailored specifically for web development. You might've peek of us during the Google I/O keynote this week.

xcited to announce a new technology we've been working on in concert with the .js and Google.

> You want it to be a mini-operating system.
>
> - Alan Kay
> on his vision for web browsers

A few years ago we realized that the web was heading towards a key inflection point. The advent of WebAssembly and new capabilities APIs made it seem possible to write a WebAssembly-based operating system powerful enough to run Node.js, entirely inside your browser. We envisioned a superior development environment that was **faster**, **more secure** and **consistent** than local environments, to enable seamless code collaboration without ever having to set up a local environment.

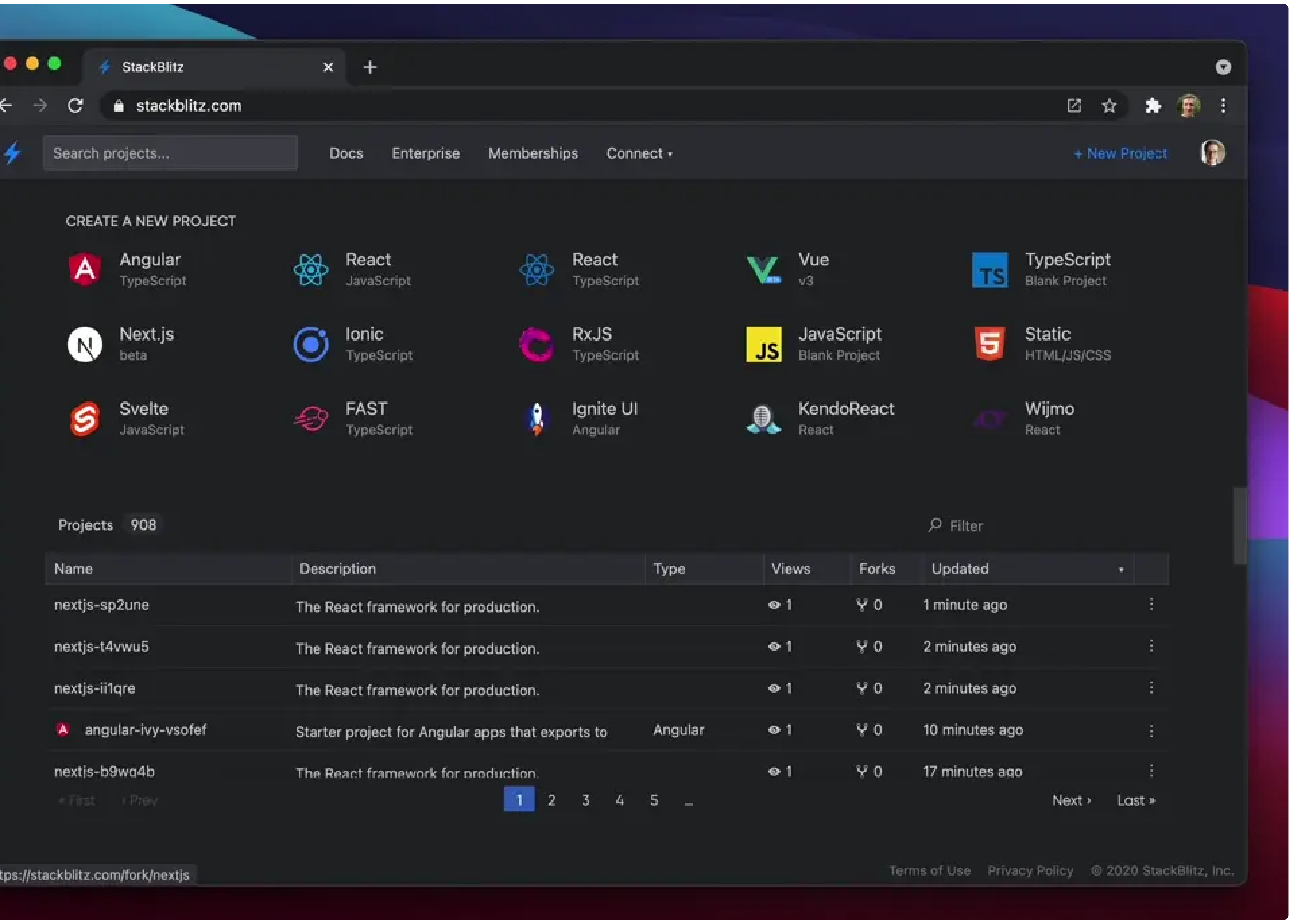
This sounded far-fetched. But if the web now runs full environments for graphic designers, video editors, and rich document editing, we wondered: **is it finally possible for developers to *use* the web to *build* the web?**

We decided to give it a shot. We hoped for the best, and expected the worst. Two years later (time flies 😅), the result has shaped up to be unexpectedly phenomenal.

Today we're excited to announce WebContainers.

WebContainers allow you to create fullstack Node.js environments that boot in milliseconds and are immediately online & link shareable—in just one click. The environment loads with VS Code's powerful editing experience, a full terminal, npm and more. It also runs *entirely inside your browser*, which yields some key benefits:

- **Faster than your local environment**. Builds complete up to 20% faster and package installs complete >= 5x faster than yarn/npm.
- **Node.js debugging in-browser**. Seamless integration with Chrome DevTools enables native back-end debugging, no installs or extensions required.
- **Secure by default**. All code execution happens *inside* the browser's security sandbox, not on remote VMs or local binaries.



Name	Description	Type	Views	Forks	Updated
nextjs-sp2une	The React framework for production.		1	0	1 minute ago
nextjs-t4vwu5	The React framework for production.		1	0	2 minutes ago
nextjs-ii1qre	The React framework for production.		1	0	2 minutes ago
angular-ivy-vsofef	Starter project for Angular apps that exports to	Angular	1	0	10 minutes ago
nextjs-b9wq4b	The React framework for production.		1	0	17 minutes ago

That's right: the Node.js runtime itself is running natively, inside the browser, for the first time ever.

Again, these environments are not running on remote servers. Instead, each environment is completely contained within your web browser. That's right: the Node.js runtime itself is running natively, inside the browser, for the first time ever.

You can try it out for yourself over at StackBlitz.com or by clicking one of the starter projects below.

Boot a Node.js project in your browser in milliseconds (don't blink!):

NEXT.JS GRAPHQL HTTP SERVER NODE.JS STARTER

As of today’s launch, WebContainers are now in **public beta**. Current support includes Next.js, GraphQL, and Vanilla Node.js and we're working with additional open source projects to expand support. (If you want to work with us check out our repo).

Why?

Setting up local environments is a huge buzzkill—especially if you want to rapidly prototype a cool idea, try out a new open source library, create a bug reproduction or collaborate with a coworker ("hey, can you check out this branch locally really quick?" 😒). This problem is more common than ever as web development moves towards fullstack SSR and SSG toolchains like Next.js.

Running user-submitted code for bug reproductions is also becoming a major security risk for open source maintainers and Fortune 100 companies alike, and these types of supply chain attacks are on the rise.

StackBlitz solves these problems by leveraging the decades of speed and security innovations built into your browser. All computation in StackBlitz happens instantly within the browser security sandbox and cannot break out to your local machine. This model also unlocks some key development & debugging benefits (more on these in a sec).

What about Code Spaces/Sandbox/Repls/...?

Legacy online IDEs run your entire dev environment on a remote server and stream the results back across the internet to your browser. The problem with this approach is that it yields few security benefits and provides a worse experience than your local machine in nearly every way: it takes minutes to spin up containers, is prone to network latency, cannot work offline, often results in network timeouts, debugging frozen/broken containers is nearly impossible, and hitting refresh just reconnects you to the broken container again.

> StackBlitz is the first online IDE whose compute model makes sense to me.

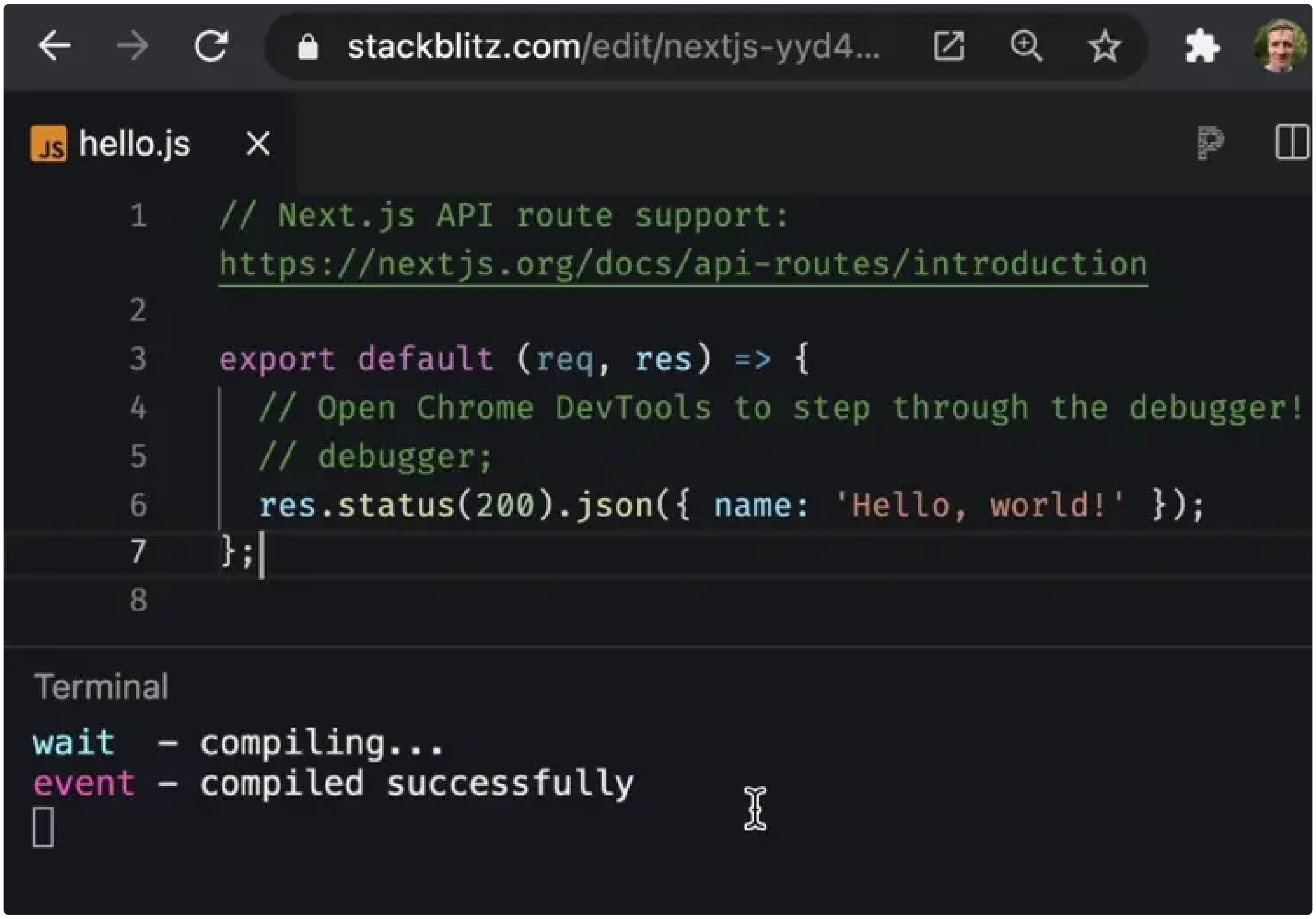
TOM PRESTON-WERNER

founder of GitHub & investor in StackBlitz

Unleashing the power of your browser.

Seamless Node.js debugging with Chrome DevTools.

It turns out, browsers are really good at debugging Javascript. Shocking, I know ;) By executing Node.js inside the browser, the integration with Chrome DevTools "just works" out of the box. No installs, no extensions, just **native back-end debugging** right in the browser:



```js
// Next.js API route support: https://nextjs.org/docs/api-routes/introduction

export default (req, res) => {
  // Open Chrome DevTools to step through the debugger!
  // debugger;
  res.status(200).json({ name: 'Hello, world!' });
};
```

```
wait  - compiling...
event - compiled successfully
```

Being able to leverage your browser's built in capabilities to develop and debug Next.js applications is a game changer.

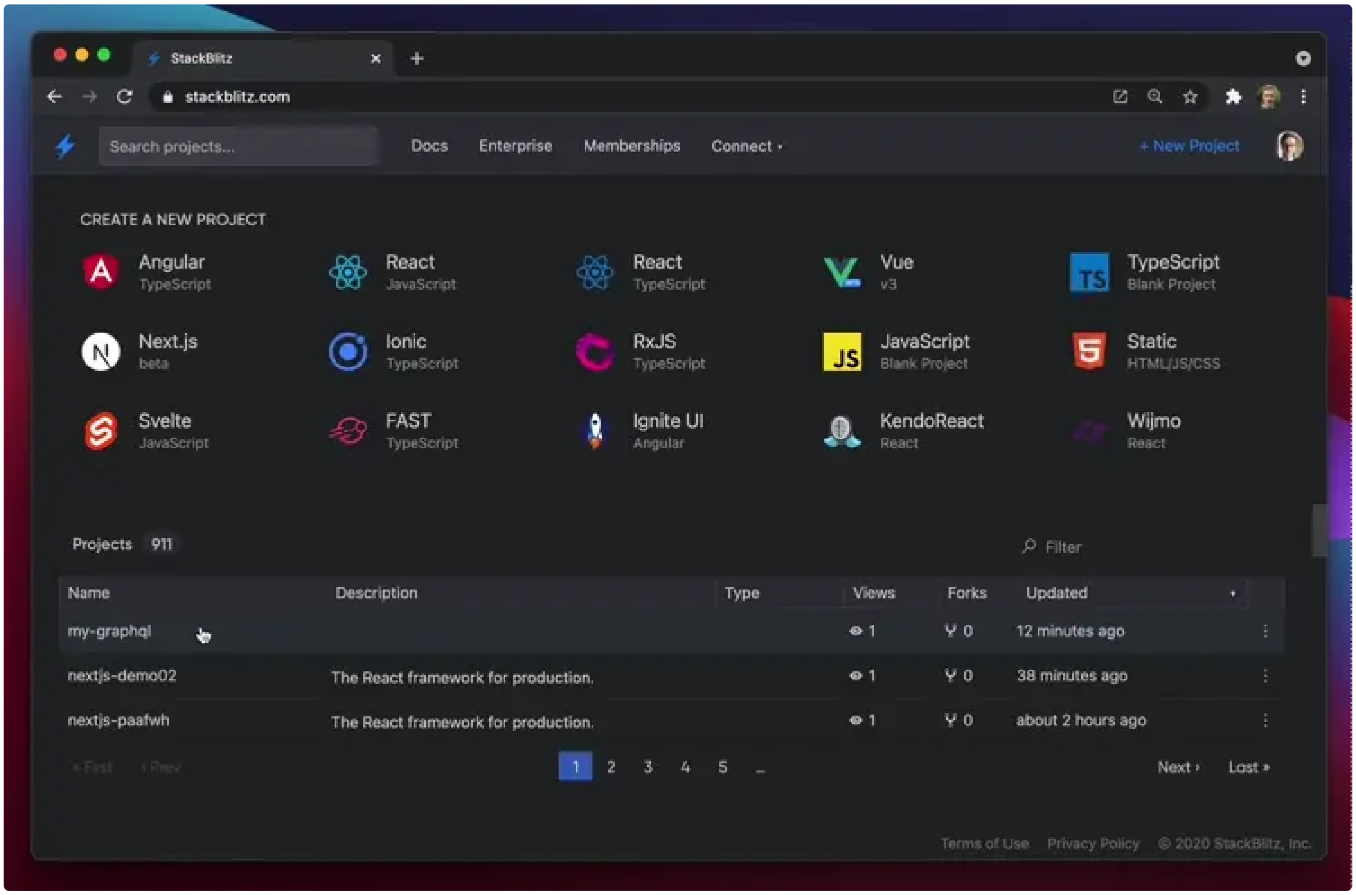
We're excited to partner with the StackBlitz team to make Next.js and Vercel more accessible to developers.

GUILLERMO RAUCH

founder of Vercel & creator of Next.js

Run servers. In your browser.

Yes, actually. WebContainers include a virtualized TCP network stack that's mapped to your browser's ServiceWorker API, enabling you to instantly create live Node.js servers on-demand that continue to work even when you go offline. Because it runs entirely within the browser security sandbox, server responses have *less latency than localhost* (!) and protects your web servers from localhost scraping attacks:



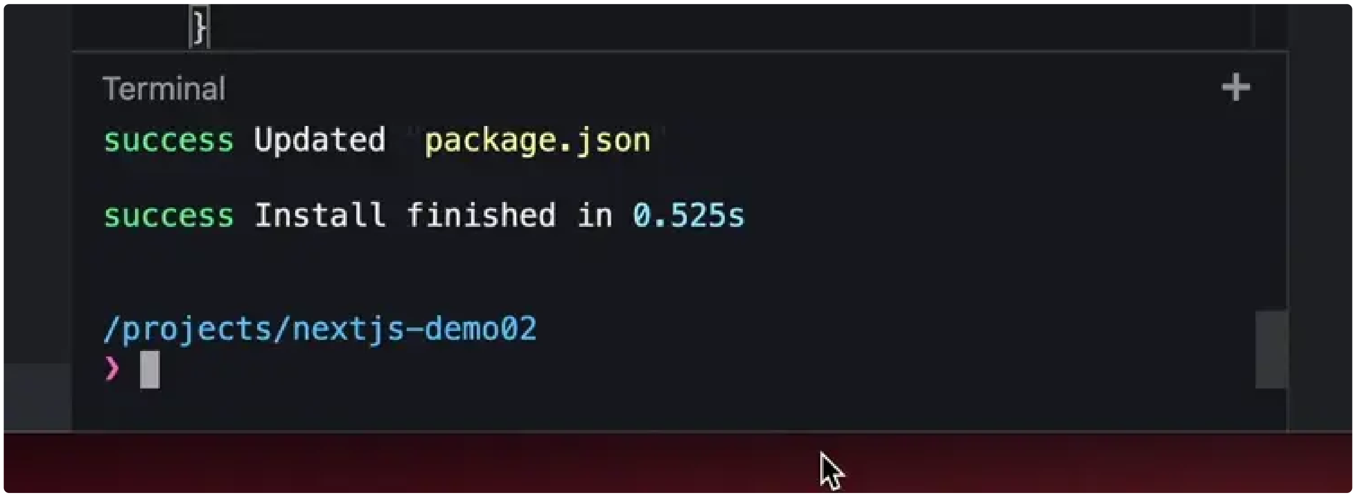
Name	Description	Type	Views	Forks	Updated
my-graphql			1	0	12 minutes ago
nextjs-demo02	The React framework for production.		1	0	38 minutes ago
nextjs-paafwh	The React framework for production.		1	0	about 2 hours ago

Zero footprint. Boots in milliseconds.

Browsers are incredibly fast at executing Javascript and WebAssembly. We leverage this to create an instant development OS that uses no server resources, and doesn't create a `node_modules` black hole on your computer.

A fresh environment on every page load.

Goodbye, `rm -rf node_modules`! WebContainer's built in npm client is so fast that it runs a fresh install on every page load ensuring you get a clean environment every single time. If something *does* goes wrong with your environment, you can get back to a clean state the same way you do any other web app: hit the refresh button.



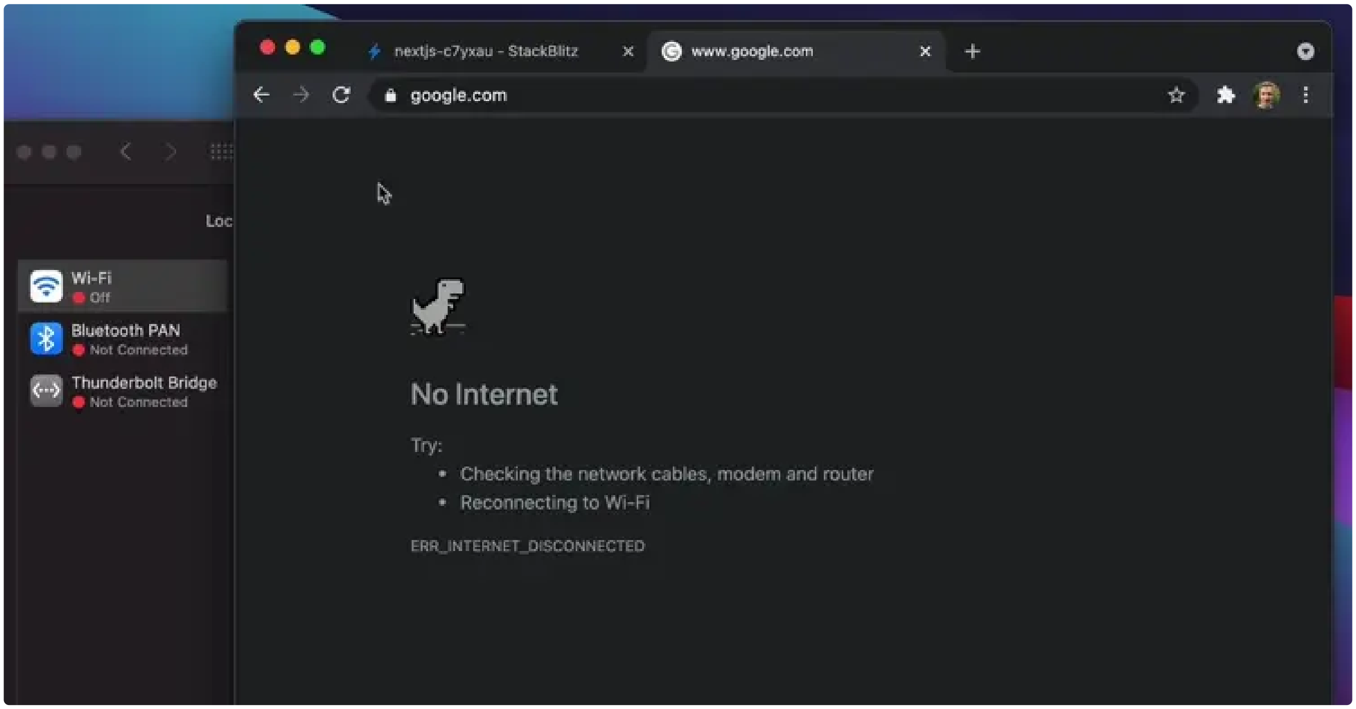
```
success Updated package.json

success Install finished in 0.525s


/projects/nextjs-demo02
❯ 
```

Zero latency. Works offline.

If the work-from-home pivot has taught us anything, it's that network blips happen—often. ISPs go down—a lot. With StackBlitz you can keep working, without an internet connection, regardless of whether you're on a train, in a plane, or backseat uber-ing in the rain:



No Internet

Try:

- Checking the network cables, modem and router
- Reconnecting to Wi-Fi

ERR_INTERNET_DISCONNECTED

Secure by default.

With StackBlitz's novel compute model, 100% of code execution occurs in the browser security sandbox. This results in a much faster and *less* restrictive development environment than local while at the same time delivering far *more* security, a very rare combination.

In fact, the default security posture is so solid, that our embedded package manager is the first publicly available tool that solves the long unaddressed npm vulnerability Sam Saccone discovered over five years ago.

Let's pause for a sec.

Because this is where the story gets really mind bending.

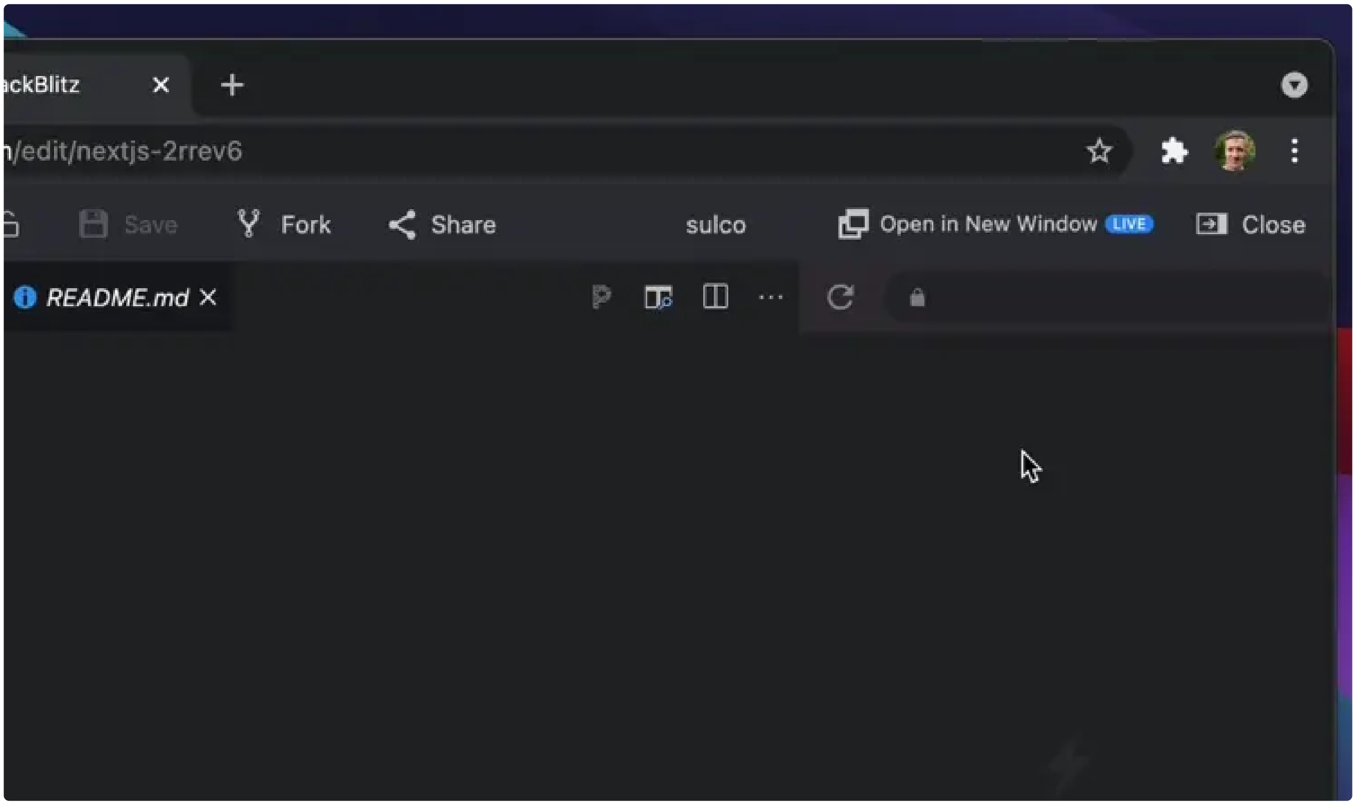
At Google I/O, we were excited to show how StackBlitz is using the latest web capabilities to deliver an experience that blurs web apps and desktop apps.

DION ALMAER
Director of Engineering on Google Chrome

What's the difference between a 'web' app and a 'native' app? The Chrome team has been shipping new capabilities APIs to close this gap and the delta is rapidly approaching zero.

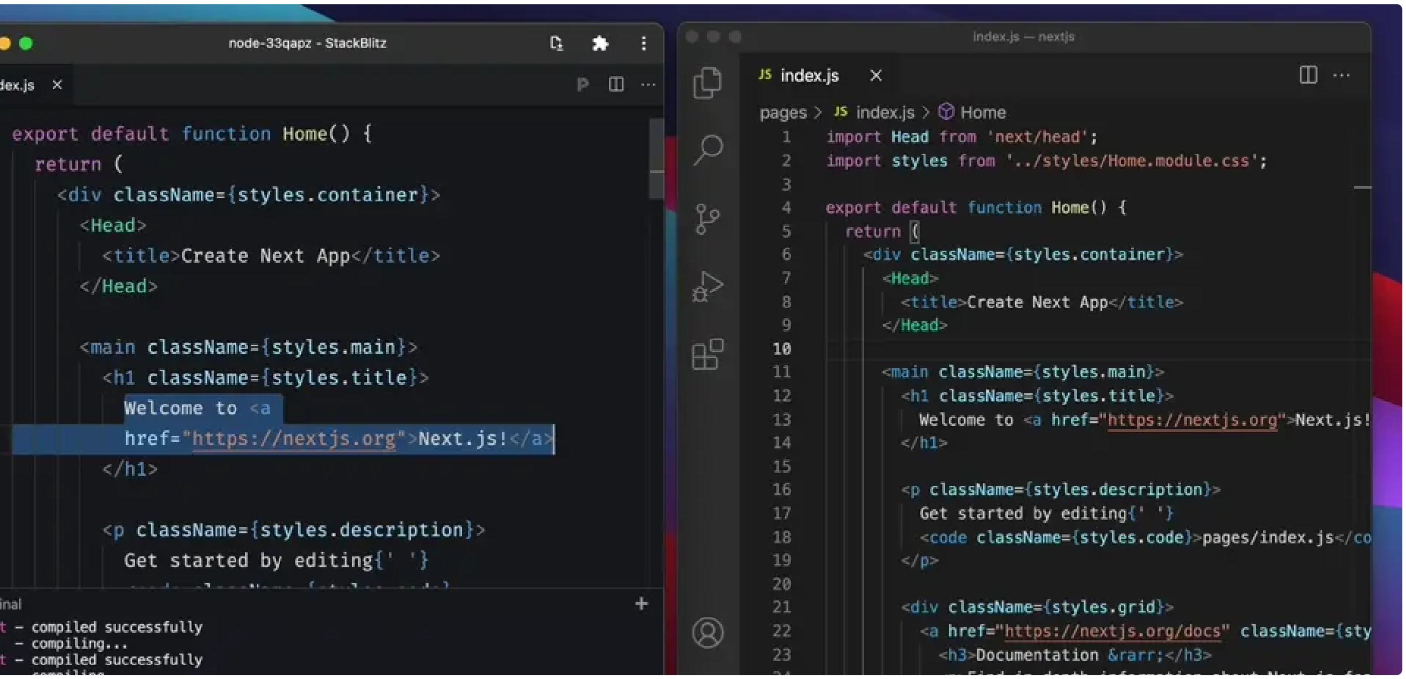
Desktop grade editing. Instant desktop app install.

Thanks to Chrome's PWA functionality, installing StackBlitz is as simple as a single click. Milliseconds later you have a desktop IDE you can launch from your dock. The keybindings you rely on for daily productivity like CMD + W and CMD + T "just work". Additionally, just like on local, you have the ability to open & debug your dev servers in a completely separate window.



Read and write from your local filesystem.

The Chrome team recently shipped the File System Access API. This gives PWA's the ability to request persistent read and write access to portions of your local file system. Paired with StackBlitz WebContainers, this hints at a potential future without the need for node, npm, git, VS Code or anything else installed on your hard drive. You just need a web browser:



```
export default function Home() {
  return (
    <div className={styles.container}>
      <Head>
        <title>Create Next App</title>
      </Head>

      <main className={styles.main}>
        <h1 className={styles.title}>
          Welcome to <a
          href="https://nextjs.org">Next.js!</a>
        </h1>

        <p className={styles.description}>
          Get started by editing{' '}
```

```
inal
t - compiled successfully
  - compiling...
t - compiled successfully
```

```
import Head from 'next/head';
import styles from '../styles/Home.module.css';

export default function Home() {
  return (
    <div className={styles.container}>
      <Head>
        <title>Create Next App</title>
      </Head>

      <main className={styles.main}>
        <h1 className={styles.title}>
          Welcome to <a href="https://nextjs.org">Next.js!
        </h1>

        <p className={styles.description}>
          Get started by editing{' '}
          <code className={styles.code}>pages/index.js</co
        </p>

        <div className={styles.grid}>
          <a href="https://nextjs.org/docs" className={sty
            <h3>Documentation &rarr;</h3>
```

Trick question: which one of these is StackBlitz, and which one is actually VS Code? 🙃

What's next?

We're spending the next quarter or two in beta as we work with open source maintainers to bring full compatibility to their userbases and stabilize the core WebContainer technology. What's coming after that is a fully-featured StackBlitz v2.

Wouldn’t it be great if you could:

- Open environments (editor + live preview!) on every PR. You can browse, play, test, and thus perform a really solid code review, without closing other projects you're working on.
- Check out multiple branches at the same time to compare them side by side. (Would you ever consider that for a local environment? In StackBlitz it means just opening a new tab).
- Update your Docusaurus documentation or Gatsby blog right from the browser.
- Learn literally any JavaScript front-end or back-end framework without installing a single thing!

@ericsimons40

ERIC SIMONS

CEO at StackBlitz making web development fast & secure. Viva la Web!

RECENT POSTS

StackBlitz September 2022 Update

OCTOBER 02, 2022

New team members, an avalanche of community updates, and 19,000 registered tickets for ViteConf ⚡💙

StackBlitz August 2022 Update

SEPTEMBER 02, 2022

Two new features released, 14,000 registrations for ViteConf, and a wonderful community shoutouts!

StackBlitz July 2022 Update

AUGUST 19, 2022

As we are preparing ViteConf, our community has been sharing amazing projects, and our engineers fixed a handful of issues. Have a look!

Introducing: Collections and Social Previews!

AUGUST 04, 2022

Two of the features most requested by the StackBlitz community are finally here!

Down the caching-hole: adventures in 'HTTP

Announcing ViteConf

JULY 14, 2022

caching and Faraday' land

JULY 18, 2022

Learn about HTTP caches, how Faraday middleware work, and how to use them to automagically start caching API calls.

Coming this October: Join the web ecosystem for a free online event featuring 28 all-star speakers and a StackBlitz surprise ⚡

Home / Blog / Posts / Introducing WebContainers: Run Node.js natively in your browser

Code the future.