Announcing Vite 2.0

- What's New in 2.0
  - Framework Agnostic Core
  - New Plugin Format and API
  - esbuild Powered Dep Pre-Bundling
  - First-class CSS Support
  - Server-Side Rendering (SSR) Support
  - Opt-in Legacy Browser Support
- Give it a Try!

February 16, 2021 - Check out the Vite 3.0 announcement



Today we are excited to announce the official release of Vite 2.0!

Vite (French word for "fast", pronounced `/vit/`) is a new kind of build tool for frontend web development. Think a pre-configured dev server + bundler combo, but leaner and faster. It leverages browser's native ES modules support and tools written in compile-to-native languages like esbuild to deliver a snappy and modern development experience.

To get a sense of how fast Vite is, check out this video comparison of booting up a React application on Repl.it using Vite vs. `create-react-app` (CRA).

If you've never heard of Vite before and would love to learn more about it, check out the rationale behind the project. If you are interested in how Vite differs from other similar tools, check out the comparisons.

What's New in 2.0

Since we decided to completely refactor the internals before 1.0 got out of RC, this is in fact the first stable release of Vite. That said, Vite 2.0 brings about many big improvements over its previous incarnation:

Framework Agnostic Core

The original idea of Vite started as a hacky prototype that serves Vue single-file components over native ESM. Vite 1 was a continuation of that idea with HMR implemented on top.

Vite 2.0 takes what we learned along the way and is redesigned from scratch with a more robust internal architecture. It is now completely framework agnostic, and all framework-specific support is delegated to plugins. There are now official templates for Vue, React, Preact, Lit Element, and ongoing community efforts for Svelte integration.

New Plugin Format and API

Inspired by WMR, the new plugin system extends Rollup's plugin interface and is compatible with many Rollup plugins out of the box. Plugins can use Rollup-compatible hooks, with additional Vite-specific hooks and properties to adjust Vite-only behavior (e.g. differentiating dev vs. build or custom handling of HMR).

The programmatic API has also been greatly improved to facilitate higher level tools / frameworks built on top of Vite.

esbuild Powered Dep Pre-Bundling

Since Vite is a native ESM dev server, it pre-bundles dependencies to reduce the number browser requests and handle CommonJS to ESM conversion. Previously Vite did this using Rollup, and in 2.0 it now uses `esbuild` which results in 10-100x faster dependency pre-bundling. As a reference, cold-booting a test app with heavy dependencies like React Material UI previously took 28 seconds on an M1-powered MacBook Pro and now takes ~1.5 seconds. Expect similar improvements if you are

switching from a traditional bundler based setup.

First-class CSS Support

Vite treats CSS as a first-class citizen of the module graph and supports the following out of the box:

- **Resolver enhancement**: `@import` and `url()` paths in CSS are enhanced with Vite's resolver to respect aliases and npm dependencies.
- **URL rebasing**: `url()` paths are automatically rebased regardless of where the file is imported from.
- **CSS code splitting**: a code-split JS chunk also emits a corresponding CSS file, which is automatically loaded in parallel with the JS chunk when requested.

Server-Side Rendering (SSR) Support

Vite 2.0 ships with experimental SSR support. Vite provides APIs to efficiently load and update ESM-based source code in Node.js during development (almost like server-side HMR), and automatically externalizes CommonJS-compatible dependencies to improve development and SSR build speed. The production server can be completely decoupled from Vite, and the same setup can be easily adapted to perform pre-rendering / SSG.

Vite SSR is provided as a low-level feature and we are expecting to see higher level frameworks leveraging it under the hood.

Opt-in Legacy Browser Support

Vite targets modern browsers with native ESM support by default, but you can also opt-in to support legacy browsers via the official @vitejs/plugin-legacy. The plugin automatically generates dual modern/legacy bundles, and delivers the right bundle based on browser feature detection, ensuring more efficient code in modern browsers that support them.

Give it a Try!

That was a lot of features, but getting started with Vite is simple! You can spin up a Vite-powered app literally in a minute, starting with the following command (make sure you have Node.js >=12):

```
npm init @vitejs/app
```

Then, check out the guide to see what Vite provides out of the box. You can also check out the source code on GitHub, follow updates on Twitter, or join discussions with other Vite users on our Discord chat server.

□□□ #3
□ □□□ □□□ 2026-10-31 13:12:47 UTC
□ cocobook-sync □□□ 2026-11-05 06:06:10 UTC