It's time. After five months and hundreds of commits, you're finally invited to try out the SvelteKit beta. It's not finished — there are a few known bugs and several missing features — but we're really happy with how it's shaping up and can't wait for you to try it.

Starting a new project is easy:

```
# create the project
mkdir my-app
cd my-app
npm init svelte@next

# install dependencies
npm install

# start dev server and open a browser tab
npm run dev -- --open
```

You'll find documentation at [kit.svelte.dev/docs](#). If you have a [Sapper](#) app that you'd like to migrate to SvelteKit, you'll find instructions at [kit.svelte.dev/docs/migrating](#).

The source code is available at [github.com/sveltejs/kit](#). Issues and pull requests are disabled while we finish getting our house in order, but we'll be making it fully open in the near future.

Wait, what is SvelteKit?

Think of it as [Next](#) for Svelte. It's a framework for building apps with Svelte, complete with server-side rendering, routing, code-splitting for JS and CSS, adapters for different serverless platforms and so on.

If you're familiar with [Sapper](#), SvelteKit is Sapper's successor.

From Snowpack to Vite

One thing that might seem surprising after the [announcement video](#), in which I waxed lyrical about [Snowpack](#), is that SvelteKit uses [Vite](#) under the hood. When we tried Snowpack back when we started thinking about what form SvelteKit should take, it was love at first sight.

Snowpack created an entirely new category of dev tooling. Rather than *bundling* your app in development, as we've been doing with webpack and Rollup for the last several years, Snowpack is an *unbundled dev server* that uses the browser's native `import` and does 1:1 transformations of things like Svelte components on the fly. As a result you get quick startup, simple caching and instant hot module reloading. Once you experience this way of working, it will ruin you for anything else.

Vite falls into the same category as Snowpack. While Vite 1 wasn't suitable for SvelteKit — it was Vue-centric (Vite and Vue are both created by [Evan You](#)) and made server-side rendering difficult — Vite 2 is framework-agnostic and designed with SSR at the core. It also has powerful features, like CSS code-splitting, that we previously had to implement ourselves. When we evaluated the two technologies side-by-side we were forced to conclude that Vite is a closer match for SvelteKit's requirements and would give us the best chance to deliver the framework of our imaginations.

We owe a deep debt of gratitude to the Snowpack team, both for the close collaboration earlier in development and for lighting the path that web development will take over the next few years. It's a wonderful tool, and you should

absolutely try it out.

Dogfooding as extreme sport

SvelteKit is very much in beta, but that doesn't mean it hasn't been used in production.

My day job is at the New York Times, where I've spent much of the last twelve months working on our [coronavirus tracker](#). It uses a customised version of the workflow that powers the majority of graphics at the Times, which isn't designed for large multi-page projects. When we decided late last year to create pages for each of the ~3,000 counties in the US, we quickly realised we would need to completely rearchitect the project.

Even though it was far from ready, SvelteKit was the only framework that matched our esoteric requirements. (Anyone who has worked in a newsroom and done battle with their CMS will know what I'm talking about.) Today it powers our [county risk pages](#) and we're in the process of migrating existing pages to the SvelteKit app.

I am eternally grateful for my coworkers' forbearance.

Using unfinished software to build an app that will be seen by millions of people is a risk, and in general I don't recommend it. But it has enabled us to develop the app much faster, and has made the framework itself much stronger than it otherwise would be.

The road to 1.0

You can see the list of outstanding issues with the 1.0 milestone on our [issue tracker](#). Alongside that work, we plan to upgrade the documentation and add more [adapters](#).

Most importantly though, we need your feedback to help us make the best possible app framework. Try it out, and let us know which pieces are missing.

Many thanks to everyone who has tried SvelteKit out despite the 'here be dragons' warnings and lack of documentation; your back-channel feedback has been invaluable. In particular, I want to acknowledge the work of [GrygrFlzr](#), who maintained unofficial docs and a fork that added Windows support when we lacked it; and [dominikg](#) whose work on [Svite](#) laid essential groundwork for SvelteKit's Vite integration. Both have now been welcomed onto the team.