



Static site generation

ON THIS PAGE



To use SvelteKit as a static site generator (SSG), use [adapter-static](#) .

This will prerender your entire site as a collection of static files. If you'd like to prerender only some pages and dynamically server-render others, you will need to use a different adapter together with [the prerender option](#).

Usage

Install with `npm i -D @sveltejs/adapter-static` , then add the adapter to your `svelte.config.js` :

svelte.config.js



```
import adapter from '@sveltejs/adapter-static';

export default {
  kit: {
    adapter: adapter({
      // default options are shown. On some platforms
      // these options are set automatically – see below
      pages: 'build',
      assets: 'build',
      fallback: undefined,
      precompress: false,
      strict: true
    })
  }
};
```



```
// This can be false if you're using a fallback (i.e. SPA mode)
export const prerender = true;
```

You must ensure SvelteKit's `trailingSlash` option is set appropriately for your environment. If your host does not render `/a.html` upon receiving a request for `/a` then you will need to set `trailingSlash: 'always'` in your root layout to create `/a/index.html` instead.

Zero-config support

Some platforms have zero-config support (more to come in future):

Vercel

On these platforms, you should omit the adapter options so that `adapter-static` can provide the optimal configuration:

svelte.config.js

```
export default {
  kit: {
    adapter: adapter({...})
  }
};
```

Options

pages

The directory to write prerendered pages to. It defaults to `build`.

assets

pages and assets to separate locations.

fallback

Specify a fallback page for SPA mode, e.g. `index.html` or `200.html` or `404.html`.

precompress

If `true`, precompresses files with brotli and gzip. This will generate `.br` and `.gz` files.

strict

By default, `adapter-static` checks that either all pages and endpoints (if any) of your app were prerendered, or you have the `fallback` option set. This check exists to prevent you from accidentally publishing an app where some parts of it are not accessible, because they are not contained in the final output. If you know this is ok (for example when a certain page only exists conditionally), you can set `strict` to `false` to turn off this check.

GitHub Pages

When building for GitHub Pages, if your repo name is not equivalent to `your-username.github.io`, make sure to update `config.kit.paths.base` to match your repo name. This is because the site will be served from `https://your-username.github.io/your-repo-name` rather than from the root.

You'll also want to generate a fallback `404.html` page to replace the default 404 page shown by GitHub Pages.

A config for GitHub Pages might look like the following:

```
svelte.config.js
```

```
import adapter from '@sveltejs/adapter-static';
```

```

    adapter: adapter({
      fallback: '404.html'
    }),
    paths: {
      base: process.argv.includes('dev') ? '' : process.env.BASE_PATH
    }
  }
};

export default config;

```

You can use GitHub actions to automatically deploy your site to GitHub Pages when you make a change. Here's an example workflow:

.github/workflows/deploy.yml

name: Deploy to GitHub Pages

on:

push:

branches: 'main'

jobs:

build_site:

runs-on: ubuntu-latest

steps:

- name: Checkout

uses: actions/checkout@v4

If you're using pnpm, add this step then change the commands and cache key below

- name: Install pnpm

uses: pnpm/action-setup@v3

with:

version: 8

- name: Install Node.js

uses: actions/setup-node@v4

with:

node-version: 20

cache: npm

```
env:
  BASE_PATH: '/${{ github.event.repository.name }}'
  run: |
    npm run build

- name: Upload Artifacts
  uses: actions/upload-pages-artifact@v3
  with:
    # this should match the `pages` option in your adapter-static options
    path: 'build/'

deploy:
  needs: build_site
  runs-on: ubuntu-latest

permissions:
  pages: write
  id-token: write

environment:
  name: github-pages
  url: ${{ steps.deployment.outputs.page_url }}

steps:
  - name: Deploy
    id: deployment
    uses: actions/deploy-pages@v4
```

If you're not using GitHub actions to deploy your site (for example, you're pushing the built site to its own repo), add an empty `.nojekyll` file in your `static` directory to prevent Jekyll from interfering.

[🔗 Edit this page on GitHub](#)

PREVIOUS

Node servers

NEXT

Single-page apps