



Writing adapters

If an adapter for your preferred environment doesn't yet exist, you can build your own. We recommend looking at the source for an adapter to a platform similar to yours and copying it as a starting point.

Adapter packages implement the following API, which creates an Adapter :

```
/** @param {AdapterSpecificOptions} options */
export default function (options) {
  /** @type {import('@sveltejs/kit').Adapter} */
  const adapter = {
    name: 'adapter-package-name',
    async adapt(builder) {
      // adapter implementation
    },
    async emulate() {
      return {
        async platform({ config, prerender }) {
          // the returned object becomes `event.platform` during dev, build and
          // preview. Its shape is that of `App.Platform`
        }
      }
    },
    supports: {
      read: ({ config, route }) => {
        // Return `true` if the route with the given `config` can use `read`
        // from `$app/server` in production, return `false` if it can't.
        // Or throw a descriptive error describing how to configure the deployment
      }
    }
  };

  return adapter;
}
```



Clear out the build directory

Write SvelteKit output with `builder.writeClient`, `builder.writeServer`, and `builder.writePrerendered`

Output code that:

Imports `Server` from `${builder.getServerDirectory()}/index.js`

Instantiates the app with a manifest generated with `builder.generateManifest({ relativePath })`

Listens for requests from the platform, converts them to a standard Request if necessary, calls the `server.respond(request, { getClientAddress })` function to generate a Response and responds with it

expose any platform-specific information to SvelteKit via the `platform` option passed to `server.respond`

Globally shims `fetch` to work on the target platform, if necessary. SvelteKit provides a `@sveltejs/kit/node/polyfills` helper for platforms that can use `undici`

Bundle the output to avoid needing to install dependencies on the target platform, if necessary

Put the user's static files and the generated JS/CSS in the correct location for the target platform

Where possible, we recommend putting the adapter output under the `build/` directory with any intermediate output placed under `.svelte-kit/[adapter-name]`.

[✎ Edit this page on GitHub](#)

PREVIOUS

Vercel

NEXT

Advanced routing