SVELTEKIT • ADVANCED

Server-only modules

ON THIS PAGE

Like a good friend, SvelteKit keeps your secrets. When writing your backend and frontend in the same repository, it can be easy to accidentally import sensitive data into your front-end code (environment variables containing API keys, for example). SvelteKit provides a way to prevent this entirely: server-only modules.

Private environment variables

The `$env/static/private` and `$env/dynamic/private` modules can only be imported into modules that only run on the server, such as `hooks.server.js` or `+page.server.js` .

Server-only utilities

The `$app/server` module, which contains a `read` function for reading assets from the filesystem, can likewise only be imported by code that runs on the server.

Your modules

You can make your own modules server-only in two ways:

- adding `.server` to the filename, e.g. `secrets.server.js`
- placing them in `$lib/server` , e.g. `$lib/server/secrets.js`

Any time you have public-facing code that imports server-only code (whether directly or indirectly)...

$lib/server/secrets.js

```js
export const atlantisCoordinates = [/* redacted */];
```

src/routes/utils.js

```js
export { atlantisCoordinates } from '$lib/server/secrets.js';

export const add = (a, b) => a + b;
```

src/routes/+page.svelte

```svelte
<script>
  import { add } from './utils.js';
</script>
```

...SvelteKit will error:

```
Cannot import $lib/server/secrets.js into public-facing code:
- src/routes/+page.svelte
  - src/routes/utils.js
    - $lib/server/secrets.js
```

Even though the public-facing code — `src/routes/+page.svelte` — only uses the `add` export and not the secret `atlantisCoordinates` export, the secret code could end up in JavaScript that the browser downloads, and so the import chain is considered unsafe.

This feature also works with dynamic imports, even interpolated ones like `` await import(`./${foo}.js`) ``, with one small caveat: during development, if there are two or more dynamic imports between the public-facing code and the server-only module, the illegal import will not be detected the first time the code is loaded.

Docs

code. For this reason, illegal import detection is disabled when running tests, as determined by `process.env.TEST === 'true'` .

Further reading

- Tutorial: Environment variables

Edit this page on GitHub

PREVIOUS
Service workers

NEXT
Snapshots