



Static files

Fresh automatically serves static assets placed in a `static/` directory in the project root. These assets are served at the root of the webserver, with a higher priority than routes. This means that if a given request matches a file in the `static/` folder, it is always served, even if there is a route that would also match the request.

Static asset responses automatically get a `content-type` header assigned based on the file extension of the file on disk. Assets are also automatically streamed from disk to the client to improve performance and efficiency for both user and server.

Fresh also adds an `etag` header to assets automatically and handles the `If-None-Match` header for incoming requests.

Caching

By default, no caching headers are added to assets. This can be disadvantageous in many scenarios, so Fresh makes it easy to serve assets with long cache lifetimes too.

The first approach to do this is manual. The client runtime exports an `asset` function that takes an absolute path to the static asset and returns a "locked" version of this path that contains a build ID for cache busting. When the asset is requested at this "locked" path, it will be served with a cache lifetime of one year.

```
import { asset } from "$fresh/runtime.ts";

export default function Page() {
  return (
    <p>
      <a href={asset("/brochure.pdf")}>View brochure</a>
    </p>
  );
}
```

Fresh also does this automatically for `src` and `srcset` attributes in `` and `<source>` HTML tags. These will automatically use "locked" paths if fresh deems it safe to do so. You can always opt out of this behaviour per tag, by adding the `data-fresh-disable-lock` attribute.

```
;
```

← Previous
Concepts: Interactive islands

Next →
Concepts: Route middleware

Source License Code of Conduct

© 2022 the fresh authors