

Helper

August 17, 2026

Helper

What it is responsible for

Helper manages request- and response-adjacent decisions and transformations for the codebase. Its entry points include `Accept`, `acceptsConfig`, `acceptsOptions`, and `defaultMatch`; `Accept` matches the accept header with given options, while cookie members read, generate, sign, set, and delete cookies. `GetConnInfo` exposes HTTP Connection information, and selector constants such as `PSEUDO_GLOBAL_SELECTOR`, `DEFAULT_STYLE_ID`, and `SELECTOR` identify CSS-related handling. The subsystem also names route matching and registered-route-path access, `basePath` lookup, proxy Fetch API wrapping, plugins for recommended behavior and redirect HTML pages, plus WebSocket contexts, listeners, upgrade types, ready states, and [adapter](#) creation. These responsibilities place protocol matching, request metadata, cookies, route helpers, CSS naming hooks, and WebSocket control under a helper boundary.

What it refuses

It rejects: Error (when `value && /\r\n]/.test(value)`).

What it needs, and who needs it

Helper depends on `Utils`, `src/jsx`, `src/request`, and `src/client`. `src/middleware`, `Adapter`, `src/jsx`, `Utils`, and `src/validator` depend on Helper. Without this subsystem, those consumers lose access to its acceptance matching, cookie operations, connection information, and the other named helper interfaces they call; the evidence does not identify which individual consumer calls which member.

Notable members

`Accept` is a work entry point and matches the accept header with given options. `acceptsConfig` and `acceptsOptions` are also work entry points; their names distinguish

configuration and options forms, but the supplied evidence assigns no further behavior to either. `defaultMatch`, the remaining entry point, similarly has no supplied description beyond its name. `GetConnInfo` carries the HTTP Connection information responsibility. Cookie members `getCookie`, `getSignedCookie`, `generateCookie`, `setCookie`, `generateSignedCookie`, `setSignedCookie`, and `deleteCookie` together cover the stated cookie actions, although the evidence does not specify their individual contracts. Their validation behavior is evidenced by the refusal above.

109 entities in `src/helper`. **5 other subsystems depend on it**, which makes it the 2nd most depended-upon part of this codebase.

What it is made of

Its 109 entities sit in 21 files under `src/helper`: 54 functions, 19 constants, 18 interfaces, 15 type aliases and 3 more.

`common.ts` holds 20 of them — more than any other file here.

Where work enters

- `Accept` — `src/helper/accepts/accepts.ts :5`
- `acceptsConfig` — `src/helper/accepts/accepts.ts :11`
- `acceptsOptions` — `src/helper/accepts/accepts.ts :17`
- `defaultMatch` — `src/helper/accepts/accepts.ts :21`

How it refuses and fails

1 of its components records a refusal or a failure handler.

It refuses work outright, under a condition written into the component itself.

Boundaries

5 other subsystems depend on this one — `Middleware`, `Adapter`, `Jsx`, `Utils`, `Validator`. Changing what it exposes changes them.

Those 5 hold 37 edges between them, unevenly: `Jsx` reaches in across 18 edges, while 2 of them hold one each. 37 edges arrive against 27 leaving — more of this repository reaches into it than it reaches out to. What they reach is narrower than the folder: 21 of its 109 members carry every inbound edge — `html` (9), `getCookie` (4) and `defineWebSocketHelper` (3), plus 18 more. Of the 27 it sends out, 23 go to `Utils` — more than to any other.

It depends on `Utils` , `Jsx` , `Request` , `Client` , and on nothing else in this repository.