

Documentation health

August 17, 2026

Documentation health

706 symbols have a reference page in this documentation. This page says how each of them came to be described, because "documented" covers four situations a reader should not have to treat alike.

Where the descriptions come from

	Symbols	What it means for a reader
Written by the author	129 of 706	A doc comment in the source. Nothing generated improves on this.
Generated, with citations	31 of 706	Points at specific <code>file:line</code> ranges you can open and check.
Generated, without citations	38 of 706	Often right, but nothing anchors it. Treat as a starting point.
Not described	508 of 706	The page shows structure only — signature, members, references.

160 of 706 descriptions can be checked against source — 129 because the author wrote them next to the code, 31 because the generator cited the lines it read. The rest are shown, and marked, for what they are.

Why there is no coverage percentage

A percentage counts whether a comment exists, so the cheapest way to raise it is to add empty ones. SonarQube withdrew its comment-density metric over this; studies of "covered" API documentation classify 43–51% of it as carrying no information; and a missing-comment warning has been measured producing 1,774 junk comments in a single codebase. The counts

above cannot

be moved that way — a doc comment that only restates the symbol name is not counted as one.

Where the gaps are

Undescribed symbols by the part of the system they sit in — worst first, so the next unit

of documentation effort has an address. This is the table to bring to planning.

Area	Undescribed	Of	Share
<code>src/jsx</code>	131	147	89%
<code>src/utils</code>	122	148	82%
<code>src/helper</code>	79	109	72%
<code>src/middleware</code>	62	110	56%
<code>src/types.ts</code>	37	41	90%
<code>src/adapters</code>	36	71	51%
<code>src/client</code>	19	25	76%
<code>src/router</code>	11	21	52%
<code>src/validator</code>	4	6	67%
<code>src/context.ts</code>	2	9	22%
<code>src/preset</code>	2	2	100%
<code>src/router.ts</code>	1	10	10%
<code>src/hono-base.ts</code>	1	1	100%
<code>src/request</code>	1	1	100%

What can be checked

A claim on these pages is anchored when it points at something a reader can open: a `file:line` range in the source, or a `[census]` tag naming the arithmetic behind a number. Anything else is prose you would have to take on trust. These counts say which is which. They do not say the anchored claims are right — only that they can be checked.

	Count	What it means
Pages read by this audit	717 [census]	Every page in the set as it stood before this section was written. This section's own claims are not counted below.
Pages carrying an anchored claim	38 of 717 [census]	The page points at source or at a census at least once.
Anchored claims	829 [census]	One per <code>file:line</code> reference and per census tag. Brackets inside code samples are code, and are not counted.
... pointing at source	814 of 829 [census]	A path and a line you can open.
... pointing at a census	15 of 829 [census]	A number computed before anything was written.

Whether those citations land in files that exist was NOT CHECKED for this build: the audit was given no way to look a path up. The count above is of citations made, not of citations verified — and reporting them as sound on that basis would be the invention this page exists to catch.

Every count in this section and the two below is arithmetic over these pages, redone on each build; `[census]` marks a figure read from that arithmetic rather than written by hand.

Where the docs and the code disagree

One lint reads every page for a single pattern: a sentence stating runtime behaviour — what listens, what validates, what retries — whose only evidence is a prose document. A README saying the server listens on a port is testimony ABOUT the code, not the behaviour itself, and a sentence citing it for the behaviour has quietly turned one into the other. The honest forms either cite the source that listens, or keep the attribution in the sentence.

1 of 717 pages carries a sentence like that, 4 sentences in all [census]. Each is quoted below exactly as written, with the page it is on and the document it leans on. This lint flags innocent sentences and misses guilty ones — it is a list to review, not a list of errors.

```
page: Docs (subsystem-docs)
cites: docs/CONTRIBUTING.md
claim: [docs/CONTRIBUTING.md:27-31] Dependency setup then runs `bun install --frozen-lockfile`,
```

```
page: Docs (subsystem-docs)
cites: docs/MIGRATION.md
claim: [docs/MIGRATION.md:39-49] Other changes in that same upgrade include a changed default f
```

```
page: Docs (subsystem-docs)
cites: docs/MIGRATION.md
claim: [docs/MIGRATION.md:76-103] For Cloudflare Workers, the guide marks `serveStatic` in Serv
```

```
page: Docs (subsystem-docs)
cites: docs/CODE_OF_CONDUCT.md
claim: [docs/CODE_OF_CONDUCT.md:39-44] They may remove, edit, or reject comments, commits, code
```

What we have not written yet

The evidence census lists, for each part of this repository, the reader-topics that part's contents warrant: a part holding database models owes its readers a data model however few models it holds. This section holds that list against the headings the pages for each part actually carry.

Every topic the census says this repository warrants — 19 [census] across 9 parts — has a page carrying a heading that answers for it. That is a statement about headings, not about depth: a section can exist and still be thin.

A topic counts as written when any page for its part carries a heading naming it — "Fields" answers for a data model, "Endpoint" for an API surface, "Dependencies" for wiring. The match is deliberately loose in that direction: a topic covered under a heading nobody predicted must not be reported as missing, so the error this makes is silence, never a false accusation of absence.

3 pages name no source path [census], so they were matched to no part and can cover no topic above — the pages about this documentation itself are among them.