

Documentation health

September 17, 2026

Documentation health

1087 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	158 of 1087	A doc comment in the source. Nothing generated improves on this.
Generated, with citations	104 of 1087	Points at specific <code>file:line</code> ranges you can open and check.
Generated, without citations	293 of 1087	Often right, but nothing anchors it. Treat as a starting point.
Not described	532 of 1087	The page shows structure only — signature, members, references.

262 of 1087 descriptions can be checked against source — 158 because the author wrote them next to the code, 104 because the generator cited the lines it read. The rest are shown, and marked, for what they are.

32 symbols carried a description that is the same sentence repeated across many symbols — a README or manifest description copied per package. Those are not counted as authored: a sentence pasted per package is presence, not authorship.

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>packages/microservices</code>	227	399	57%
<code>packages/common</code>	131	322	41%
<code>packages/core</code>	120	253	47%
<code>packages/websockets</code>	23	57	40%
<code>packages/platform-express</code>	14	22	64%
<code>packages/platform-fastify</code>	9	14	64%
<code>tools/gulp</code>	5	6	83%
<code>packages/testing</code>	2	11	18%
<code>hooks/mocha-init-hook.ts</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	1099 [census]	Every page in the set as it stood before this section was written. This section's own claims are not counted below.

	Count	What it means
Pages carrying an anchored claim	105 of 1099 [census]	The page points at source or at a census at least once.
Anchored claims	1081 [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	1066 of 1081 [census]	A path and a line you can open.
... pointing at a census	15 of 1081 [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.

No page states runtime behaviour on the authority of a prose document alone. That is an absence of the pattern rather than proof the pages and the code agree: the lint reads verbs and file extensions, and it misses what it was not taught to see.

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.

1 of 22 reader-topics, across 10 parts of this repository, have no page carrying a heading that answers for them [census].

Part	Topics no heading answers for	Pages for this part
overview	wiring	7 [census]

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.