

Docs

August 17, 2026

Docs

Overview

The `docs` part is a human-facing policy and guidance surface: the code of conduct defines participation standards and reporting, the contribution guide describes how to prepare and submit work, and the migration guide records upgrade-specific source changes. [docs/CODE_OF_CONDUCT.md:3-13; docs/CONTRIBUTING.md:3-10; docs/MIGRATION.md:3-19]

Nothing in these Markdown files receives an HTTP request, dispatches a job, or imports application modules; their operational role is to tell contributors and application maintainers what actions to take outside this directory. [docs/CODE_OF_CONDUCT.md:39-67; docs/CONTRIBUTING.md:27-60; docs/MIGRATION.md:25-295]

For a contributor, the material starts with accepted ways to participate: opening an issue, submitting a pull request, creating external middleware, sharing feedback, or building an application with Hono. [docs/CONTRIBUTING.md:3-10] The guide also sets expectations about acceptance: the project began as Yusuke Wada's hobby proposal, and an idea may not be accepted even when it is considered good. [docs/CONTRIBUTING.md:12-19]

For an existing Hono user, the migration document is organized by source and target release ranges, then narrows into concrete API, import, middleware, type, and deployment changes. [docs/MIGRATION.md:3-19; docs/MIGRATION.md:21-57; docs/MIGRATION.md:59-123] For a community participant reporting misconduct, the code of conduct routes a report to the listed enforcement contact and describes review, investigation, privacy expectations, and possible moderation consequences. [docs/CODE_OF_CONDUCT.md:59-67; docs/CODE_OF_CONDUCT.md:69-113]

How it works

Contribution path

A contributor first chooses a participation route, with issues intended for feature proposals and bug reports, while pull requests cover bug fixes, typos, and refactoring.

[docs/CONTRIBUTING.md:3-8] Before working locally, the guide tells developers to install Bun because the project uses it as its package manager.

[docs/CONTRIBUTING.md:27-31] Dependency setup then runs `bun install --frozen-lockfile`, and the local-development example combines cloning the repository, changing into it, and running that same install command.

[docs/CONTRIBUTING.md:33-35; docs/CONTRIBUTING.md:56-60]

The stated pre-submission check is `bun run test`; the guide asks that pull requests pass that command. [docs/CONTRIBUTING.md:37-39] The AI policy applies whether or not AI was actually used: a contribution must not waste maintainer time or make the work unpleasant, and a maintainer may close a pull request without notice and block the account. [docs/CONTRIBUTING.md:21-25]

Middleware that depends on other libraries or only works in a particular environment is explicitly outside the core boundary described here. [docs/CONTRIBUTING.md:41-44]

The guide names GraphQL Server, Firebase Auth, and Sentry as examples, says developers can create such middleware independently, and directs prospective work toward an issue and the `honojs/middleware` monorepo. [docs/CONTRIBUTING.md:43-54] This means that a contribution in that category follows a different destination than a core-code change: it may live under the `honojs` organization and the `@honojs` namespace rather than in the core. [docs/CONTRIBUTING.md:50-54]

Upgrade path

The migration guide works as a decision record keyed by the version boundary being crossed. [docs/MIGRATION.md:3-7; docs/MIGRATION.md:21-25;

docs/MIGRATION.md:59-62] At the newer Deno boundary, it says the module is no longer published from `deno.land/x`, directs users to JSR, and shows replacing the URL import with `jsr:@hono/hono`. [docs/MIGRATION.md:5-19]

For the major-version upgrade section, the guide first lists removed or obsolete APIs and their replacements. [docs/MIGRATION.md:25-37] Examples include switching the Next.js adapter import from `hono/nextjs` to `hono/vercel`, using `c.json()` rather than `c.jsonT()`, accessing request-native properties through `req.raw`, and using `app.fetch()` rather than `app.handleEvent()`. [docs/MIGRATION.md:27-37] It separately states that `app.get()` implicitly handles `HEAD`, so an application should not retain a separate `app.head()` usage from the older API. [docs/MIGRATION.md:32-35]

The Cloudflare Workers static-file migration has a concrete configuration consequence: the `serve-static` call needs a `manifest` option, with the example importing `__STATIC_CONTENT_MANIFEST` and passing it alongside the asset root. [docs/MIGRATION.md:39-49] Other changes in that same upgrade include a changed default for JSX renderer `docType`, the absence of `children` from `FC`, removed MIME types, and type-sensitive route and validator chaining. [docs/MIGRATION.md:51-57]

The earlier major-version guidance changes the meaning of `c.req`: it becomes `HonoRequest`, while code that needs the underlying platform `Request` must use `c.req.raw`. [docs/MIGRATION.md:64-74] It marks `StaticRouter` unavailable, describes a changed validator API, and directs static serving imports to runtime adapters such as the Cloudflare Workers, Bun, or Deno adapter paths. [docs/MIGRATION.md:76-103] For Cloudflare Workers, the guide marks `serveStatic` in Service Worker mode obsolete, identifies that mode with `app.fire()`, and recommends Module Worker mode using `export default app`. [docs/MIGRATION.md:105-110]

Type migration is treated as source editing rather than runtime configuration: the guide requires a `type` declaration, not an `interface`, when declaring generics for `new Hono`. [docs/MIGRATION.md:112-123] Another example describes the generic constructor's `Variables` and `Bindings`: bindings type Cloudflare Workers environment values, while variables type values stored and read through `c.set` and `c.get`. [docs/MIGRATION.md:184-210]

The migration guide also records behavioral failure cases that callers must correct. [docs/MIGRATION.md:150-182] Nested Basic Auth or Bearer Auth middleware must return `auth(c, next)` rather than await it in the shown handler pattern. [docs/MIGRATION.md:154-163] `c.req.parseBody()` is limited to multipart and URL-encoded form data in the documented change; JSON, text, and array-buffer bodies instead go through `c.req.json()`, `c.req.text()`, and `c.req.arrayBuffer()`. [docs/MIGRATION.md:169-182]

The oldest covered instructions preserve historical alternatives but label several middleware paths obsolete. [docs/MIGRATION.md:217-295] Deno middleware imports move out of `hono/mod.ts` to `hono/middleware.ts`; cookie handling and request-body parsing move to request or context methods in the examples; GraphQL Server and Mustache middleware are identified as obsolete. [docs/MIGRATION.md:217-231; docs/MIGRATION.md:233-295]

Conduct and enforcement path

The code of conduct begins by committing community members, contributors, and leaders to harassment-free participation and to an open, welcoming, diverse, inclusive, and healthy community. [docs/CODE_OF_CONDUCT.md:3-13] It distinguishes constructive conduct, such as empathy, respect, feedback, accountability, and focus on the community, from unacceptable conduct including sexualized behavior, attacks, harassment, publishing private information without permission, and other professionally inappropriate behavior. [docs/CODE_OF_CONDUCT.md:15-37]

When a standards issue occurs, community leaders are assigned responsibility for clarification, enforcement, and corrective action. [docs/CODE_OF_CONDUCT.md:39-44] They may remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that do not align with the policy, and they communicate moderation reasons when appropriate. [docs/CODE_OF_CONDUCT.md:46-49] The policy applies both in community spaces and when an individual officially represents the community publicly. [docs/CODE_OF_CONDUCT.md:51-57]

A report of abusive, harassing, or otherwise unacceptable behavior goes to `yusuke@kamawada.com`, after which complaints are reviewed and investigated promptly and fairly. [docs/CODE_OF_CONDUCT.md:59-64] Leaders must respect the reporter's privacy and security. [docs/CODE_OF_CONDUCT.md:66-67] The stated consequence path ranges from a private written correction, through a warning that restricts interaction, to a temporary ban and a permanent ban for sustained or severe patterns of violation. [docs/CODE_OF_CONDUCT.md:74-113]

Configuration

The documentation itself does not declare environment-variable readers, command-line flag parsing, or a runtime settings format; its actionable setup instruction is the Bun dependency-install command with the frozen lockfile option. [docs/CONTRIBUTING.md:27-35]

The migration examples describe application-side settings that may require source changes during an upgrade. [docs/MIGRATION.md:39-49; docs/MIGRATION.md:184-210] In particular, Cloudflare static serving takes a `manifest` option, and the generic `Bindings` type is described as the type surface for Cloudflare Workers environment values such as `KV` and `Storage`. [docs/MIGRATION.md:39-49; docs/MIGRATION.md:186-210] The contribution guide's test command is a contributor workflow requirement, not a setting read by the documentation. [docs/CONTRIBUTING.md:37-39]

Wiring

The repository README is the visible entry point into this part for migration and contribution work: it links directly to `docs/MIGRATION.md` under its Migration section and to `docs/CONTRIBUTING.md` under its Contributing section. [README.md:53-55; README.md:61-71] The README points general product documentation to `hono.dev`, which separates the repository-local migration and contribution guides from the broader documentation site. [README.md:49-55]

Within the contribution guide, Bun is an external package-management dependency named for local setup. [docs/CONTRIBUTING.md:27-35] The third-party middleware route points outward to the `honojs/middleware` monorepo and describes possible distribution in the `@honojs` namespace. [docs/CONTRIBUTING.md:50-54] The migration guide also points outward to the Deno getting-started page and to the Hono documentation for validator details. [docs/MIGRATION.md:19; docs/MIGRATION.md:80-84]

The conduct document's boundary is community governance rather than application execution: its inputs are reports and contributions in community spaces, and its outputs are investigation, moderation, correction, interaction restrictions, or bans. [docs/CODE_OF_CONDUCT.md:46-67; docs/CODE_OF_CONDUCT.md:74-113] Its text attributes the policy to Contributor Covenant and links to that project's FAQ and translations, while its enforcement ladder is described as inspired by Mozilla's code-of-conduct enforcement ladder. [docs/CODE_OF_CONDUCT.md:115-128]

73 entities in `docs`. Nothing else in this repository depends on it.

What it is made of

Its 73 entities sit in 3 files under `docs`: 70 doc comments and 3 markdown docs.

Where work enters

- Contributor Covenant Code of Conduct — `docs/CODE_OF_CONDUCT.md` :1
- Contribution Guide — `docs/CONTRIBUTING.md` :1
- Migration Guide — `docs/MIGRATION.md` :1