

Platform Fastify

August 3, 2026

Platform Fastify

What it is responsible for

Platform Fastify manages the Fastify-facing work entry points named `FastifyAdapter`, `MiddlewareFn`, and `FASTIFY_ROUTE_CONFIG_METADATA`. Its documented surface is tied to Nest, which the authors describe as a framework for building Node.js server-side applications. The available evidence identifies these symbols as the paths through which work enters this subsystem, without describing their internal control flow or integration details.

What it refuses

Platform Fastify rejects a version constraint when its value is neither a string nor an array of strings. The rejection message is: “Version constraint should be a string or an array of strings.”

Notable members

- `FastifyAdapter` is a named entry point for work entering Platform Fastify.
- `MiddlewareFn` is another named entry point for work entering the subsystem.
- `FASTIFY_ROUTE_CONFIG_METADATA` is a named constant and work-entry symbol; the evidence does not state its stored value or how consumers interpret it.

44 entities in `packages/platform-fastify`. Nothing else in this repository depends on it.

What it is made of

Its 44 entities sit in 11 files under `packages/platform-fastify`: 29 doc comments, 5 interfaces, 3 constants, 3 functions and 4 more.

`constants.ts` holds 3 of them — more than any other file here.

`FastifyAdapter` declares 50 methods, the widest surface here.

Where work enters

- `Readme.md` — `packages/platform-fastify/Readme.md` :1
- `FastifyAdapter` — `packages/platform-fastify/adapters/fastify-adapter.ts` :124
- `MiddlewareFn` — `packages/platform-fastify/adapters/middie/fastify-middie.ts` :17
- `FASTIFY_ROUTE_CONFIG_METADATA` — `packages/platform-fastify/constants.ts` :1

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.

Its `catch` blocks handle a failure that already happened in 2 places.