

Host

July 31, 2026

Host

What it is responsible for

Host manages the entry points represented by `HostController` and `HostModule`, including `greeting`, `asyncGreeting`, `streamGreeting`, and `localPipe`. It also owns named services and classes including `HostService`, `UserService`, `UserByIdPipe`, and `TestDto`. The evidence does not identify endpoint-to-service links, configuration, inputs, outputs, or execution order explicitly here.

What it needs, and who needs it

Host needs no dependencies from elsewhere in this repository. `Integration` needs Host: it is listed as depending on this subsystem. Without Host, `Integration` would lack that declared dependency, and `HostController`, `HostModule`, and `HostService` would not be available from it.

10 entities in `integration/hello-world/src/host`. **1 other subsystem depends on it**, which makes it the 5th most depended-upon part of this codebase.

What it is made of

Its 10 entities sit in 6 files under `integration/hello-world/src/host`: 4 HTTP endpoints, 3 services, 1 controller, 1 module and 1 more.

`host.controller.ts` holds 5 of them — more than any other file here.

Where work enters

1 controller publishes 4 `GET` endpoints. None of them declares a guard.

- `HostController` — `integration/hello-world/src/host/host.controller.ts` :6
- `greeting` — `integration/hello-world/src/host/host.controller.ts` :13
- `asyncGreeting` — `integration/hello-world/src/host/host.controller.ts` :19
- `streamGreeting` — `integration/hello-world/src/host/host.controller.ts` :24

- `localPipe` — `integration/hello-world/src/host/host.controller.ts` :29
- `HostModule` — `integration/hello-world/src/host/host.module.ts` :6

Boundaries

1 other subsystem depends on this one — `Integration` . Changing what it exposes changes them.

They hold 1 edge into it between them. What they reach is narrower than the folder: 1 of its 10 members carries every inbound edge — `HostModule` (1).

It depends on no other subsystem in this repository — it is a leaf.