

# Integration

July 31, 2026

---

## Integration

---

### What it is responsible for

Integration manages entry handling through `AppController`, `getGlobals`, `AppV1Controller`, `helloWorldV1`, `paramV1`, and `AppV2Controller`, and it turns incoming work into controller, module, service, class, and API-endpoint activity. Its named members indicate responsibility spans application setup and request-facing behavior: `AppModule` and `AppController` sit alongside `RecipesModule`, `RecipesResolver`, `RecipesService`, `AuthGuard`, and `DataInterceptor`. The authors explicitly identify HTTP Proxy entries supporting non-stream find methods and client-side stream find methods, define a “Date custom scalar type,” and mark one decorator with: “This decorator must not be used anywhere!” These statements bound documented behavior without assigning unstated behavior to every listed member.

### What it refuses

Integration rejects a `gqlContext` with `UnauthorizedException`. It rejects a missing recipe (`!recipe`) with `NotFoundException`. It rejects `isNaN(val)` with “Validation failed.” It rejects a missing `this.helperSvc.request` with `error`, an undefined durable service with “Durable service is undefined,” and `requestPayload.forceError` with “Forced error.” These are explicit refusal conditions; callers reaching the relevant paths receive the stated failures rather than continued processing.

### What it needs, and who needs it

Integration depends on `integration/discovery/src/my-webhook`, `Core`, `Host`, `Host Array`, `integration/injector/src/exports`, `Dogs`, `integration/inspector/src/external-svc`, `integration/inspector/src/chat`, `Microservices`, and `Common`. The supplied dependency record names no subsystem that depends on Integration. Therefore, the evidence identifies Integration as the needing side, but does not identify an external consuming side or establish which external component breaks without it. Within Integration, the

record names entry symbols and modules, but does not assign any dependency to an individual work path.

## Notable members

`AppController` is a named work entry. `RecipesService` appears with `RecipesResolver` and `RecipesModule`; absent recipes are rejected with `NotFoundException`. `AuthGuard` is named alongside a `gqlContext` rejection: `UnauthorizedException` in the evidence.

188 entities in `integration`. Nothing else in this repository depends on it.

## What it is made of

Its 188 entities sit in 95 files under `integration`: 54 HTTP endpoints, 48 modules, 25 controllers, 17 services and 44 more.

`utils.ts` holds 13 of them — more than any other file here.

`AppService` declares 7 methods, the widest surface here.

## Where work enters

25 controllers publish 54 HTTP endpoints — 40 `GET` and 14 `POST`. They answer under `/foo` and `/lazy`. `AppController` carries 30 of them; the remaining 24 are split across 14 other controllers. None of them declares a guard.

- `AppController` — `integration/cors/src/app.controller.ts` :3
- `getGlobals` — `integration/cors/src/app.controller.ts` :5
- `AppV1Controller` — `integration/inspector/src/app-v1.controller.ts` :3
- `helloWorldV1` — `integration/inspector/src/app-v1.controller.ts` :7
- `paramV1` — `integration/inspector/src/app-v1.controller.ts` :12
- `AppV2Controller` — `integration/inspector/src/app-v2.controller.ts` :3

## Boundaries

It depends on `My Webhook`, `Core`, `Host`, `Host Array`, `Exports`, `Dogs`, `External Svc`, `Chat`, `Microservices`, `Common`, and on nothing else in this repository.

## How this code is named

These conventions cover most of the codebase. Learning them is faster than reading an index —

each one lets you find any member of its family without looking it up.

| Pattern         | Where                 | Count | Examples                                                                        |
|-----------------|-----------------------|-------|---------------------------------------------------------------------------------|
| *.module.ts     | across the repository | 100   | a.module.ts , b.module.ts , c.module.ts , app.module.ts                         |
| *.service.ts    | across the repository | 80    | bar.service.ts , foo.service.ts , app.service.ts , cats.service.ts              |
| *.controller.ts | across the repository | 71    | app.controller.ts , rmq.controller.ts , host.controller.ts , cats.controller.ts |
| *.dto.ts        | across the repository | 26    | sum.dto.ts , test.dto.ts , user.dto.ts , business.dto.ts                        |