

Autodiscovery

Autodiscovery

Anonymous · 10/10/2026

Autodiscovery

One shared criterion for what counts as “visible”, applied consistently across both surfaces — because both are walked from the exact same source:

Explorer composes over `derafu/backbone-dispatcher`’s own `ExplorerInterface` rather than reimplementing its walk, only adding HATEOAS `_links` on top of whatever it returns — `GET /api/billing/document/builder` returns the worker’s `_links` plus its operations. Every id, name, description and policy-based visibility rule (see [Backbone Dispatcher](#)) comes straight from the delegate: with a policy wired into it, a worker nobody may call does not show up here either.

Documenter generates the OpenAPI 3.1 document served at `GET /api/openapi-docs.json`, walked from `Explorer::tree()` — the exact same nested, policy-pruned structure `Explorer` itself is built on, not a separate traversal of the package registry. This used to not be true: `Documenter` had its own, independent notion of “visible” (any method tagged with Backbone’s `#[Operation]` attribute, optionally narrowed by a *second*, separately-injected policy instance), which could silently drift from what a real `DirectDispatcher` would actually accept — an operation the active `OperationPolicyInterface` allowed, but nobody had gotten around to tagging, was dispatchable and browsable yet absent from the spec, understating the real attack surface. There is no way to configure `Documenter` differently from `Explorer` anymore: whatever `OperationPolicyInterface` was wired into the `ExplorerInterface` `Explorer` composes over is the one and only thing that decides what gets documented.

`#[Operation]` still exists and is still useful — for overriding a parameter’s reflected type/description/example, or the operation’s own name/description, when the reflected PHPDoc isn’t enough (see [Backbone Dispatcher](#)) — it just no longer gates *whether* something gets documented, only how it looks once it is. If you want a real, non-`#[Operation]`-tagged public method to disappear from both `Explorer` and `Documenter` at once, that is exactly what [TaggedOperationPolicy](#) is for — wired once, where the dispatch chain itself is built, never in `backbone-api`.

Every documented operation is generated as a single OpenAPI post entry (there’s no GET/PUT/DELETE distinction), and its request/response schema is built from the reflected parameter types, following the same type-name vocabulary as `backbone-dispatcher`’s `Caster::resolveType()` (string, number, integer, boolean, array, object).

Operation ≠ Job

Same terminology note as [Backbone Dispatcher](#): an “operation” here is any public method found via reflection, unrelated to Backbone’s own `JobInterface#[Job]` concept.

Last updated on 10/10/2026