

# Introduction

Generic invocation and introspection of Backbone operations by id, with plain array parameters and structured failures

Anonymous · 10/10/2026

## Safely Invoke Any Backbone Operation, From Anywhere

|             |                            |             |             |           |               |           |           |
|-------------|----------------------------|-------------|-------------|-----------|---------------|-----------|-----------|
| github      | derafu/backbone-dispatcher | last commit | last monday | CI        | passing       | code size | 320.6 KiB |
| open issues | 0                          | downloads   | 233         | downloads | 65 this month |           |           |

Generic invocation and introspection for [Backbone](#) services: turn any worker's operation into something safely callable by id, with plain array parameters, from outside PHP's own type system.

## Why

A [Backbone](#) worker's public methods are ordinary, statically-typed PHP methods — great for an in-process PHP caller, useless for anything else. Backbone Dispatcher adds a generic invocation layer on top: it turns a string operation id (`"package.component.worker::operation"`) plus a plain associative array of parameters into a real, reflection-resolved, type-coerced method call on the right worker, and turns whatever comes back — a return value **or** an uncaught exception — into a plain, serializable shape.

That is what makes it possible for a caller with no concept of PHP reflection, PHP's type system, or PHP exceptions — Python code across a [phpy](#) boundary, an HTTP client through [Backbone API](#), or anything else — to invoke any Backbone operation and get back a predictable, serializable answer either way.

## Installation

```
composer require derafu/backbone-dispatcher
```

## Requirements

PHP 8.5+. Depends on [derafu/backbone](#), [derafu/cache](#) and [derafu/translation](#).

Last updated on 10/10/2026