

Introduction

Python bridge for any Backbone Dispatcher PHP library through swoole/phpy, with no phpy code on the side of the caller

Anonymous · 10/10/2026

Call Any Backbone-Dispatcher Library From Python

github	derafu/backbone-bridge-python	last commit	september	CI	passing	code size	64.2 KiB
open issues	0	pypi	v0.2.1	downloads	53/month		

Generic Python bridge for any [derafu/backbone-dispatcher](#)-based PHP library, via [swoole/phpy](#) — without a caller ever having to know phpy exists.

Why

`derafu/backbone-dispatcher`'s `SafeDispatcherInterface` already turns any Backbone operation into something that never throws a raw PHP exception and always returns a serializable value — exactly the contract a foreign-language caller needs. `swoole/phpy` lets Python call into a real, embedded PHP interpreter in the same process. This package is the glue between the two: it boots a real `SafeDispatcherInterface` and exposes it as `dispatch(operation_id, **params)`, translating PHP exceptions into typed Python exceptions on the way back.

It knows nothing about any specific library. A specific PHP library's own bridge subclasses `GenericDispatcher` (and, for exploring a package tree, `GenericExplorer` — see [Exploring the Package Tree](#) below), pre-wiring how its own `SafeDispatcherInterface`/`SafeExplorerInterface` gets booted — this is exactly what this package's own test suite does, against a real, minimal PHP fixture (no mocks):

```
class ExampleDispatcher(GenericDispatcher):
    _BOOTSTRAP_CLASS = 'Derafu\\TestsBackboneBridgePython\\Fixture\\Bootstrap'

    def __init__(self, autoload_path=None):
        super().__init__(self._BOOTSTRAP_CLASS, autoload_path=autoload_path)
```

Installation

```
pip install derafu-backbone-bridge
```

phpy itself is **not** a normal pip dependency and never will be resolvable from PyPI: it requires a PHP build with `--enable-embed`, so it must already be present system-wide wherever this package runs — see [Docker with Python and Caddy](#) for a ready-made image with an optional `PHPY_ENABLED` build. Install this package into a virtualenv created with `--system-site-packages` so it can see the system `phpy`.

PyPI has an unrelated package also named ``phpy``

There is a long-abandoned, unrelated package on PyPI called `phpy` (“call legacy PHP functions from Python”, 2013). It is **not** `swoole/phpy`. This package deliberately does not declare `phpy` as a dependency at all, specifically so `pip install` never silently resolves that decoy instead of the real, system-provided one.

Requirements

Python 3.14+. `phpy` itself requires PHP built with `--enable-embed`.

Last updated on 10/10/2026