

FRANKENPHP

And PHP came back
from the dead!



Kévin Dunglas: Polyglot Dev

→ PHP

API Platform creator,
Symfony Core Team

→ Go

Mercure.rocks creator,
Caddy maintainer,
Go core contributor

→ C

PHP core contributor

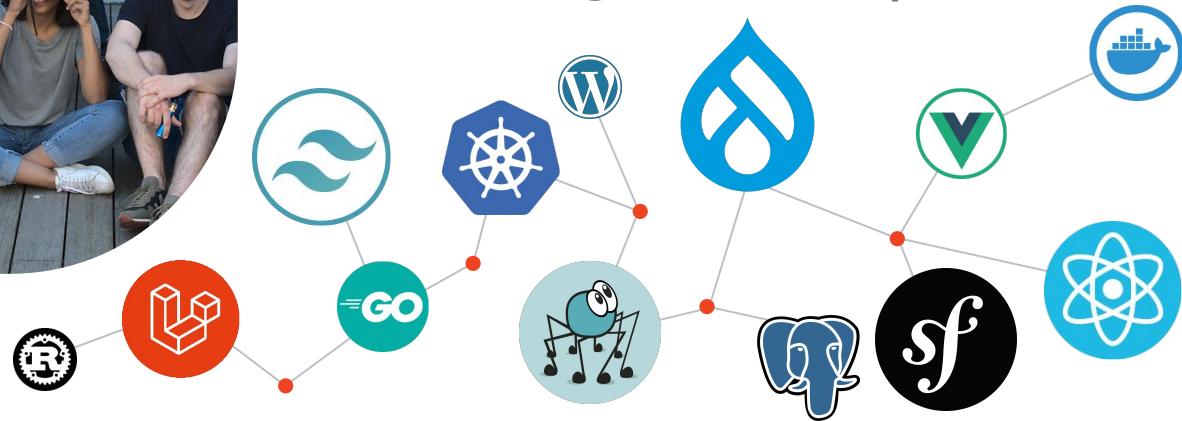


@dunglas
X  mastodon.social



Web and Cloud Experts

- Web development (PHP, JS, Go, Rust...)
- Support & hosting (FrankenPHP) apps
- Consultancy & maintenance
- UX & UI design
- contact@les-tilleuls.coop ❤️



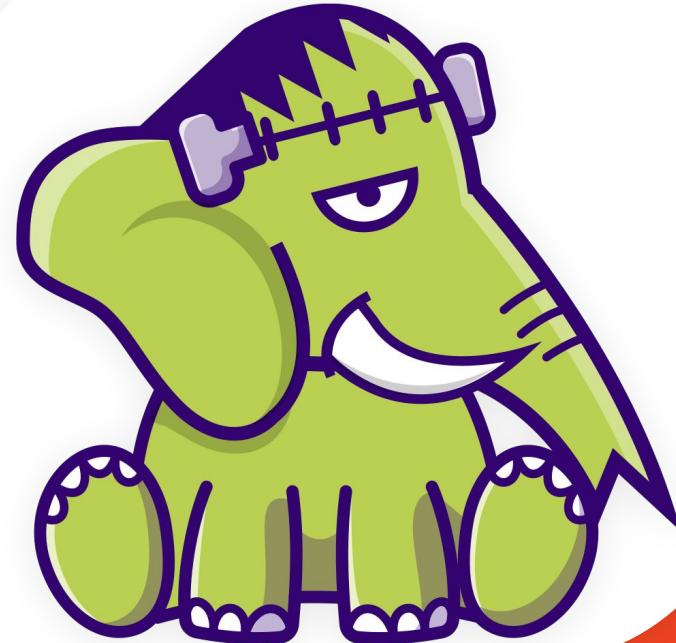


FRANKENPHP?!

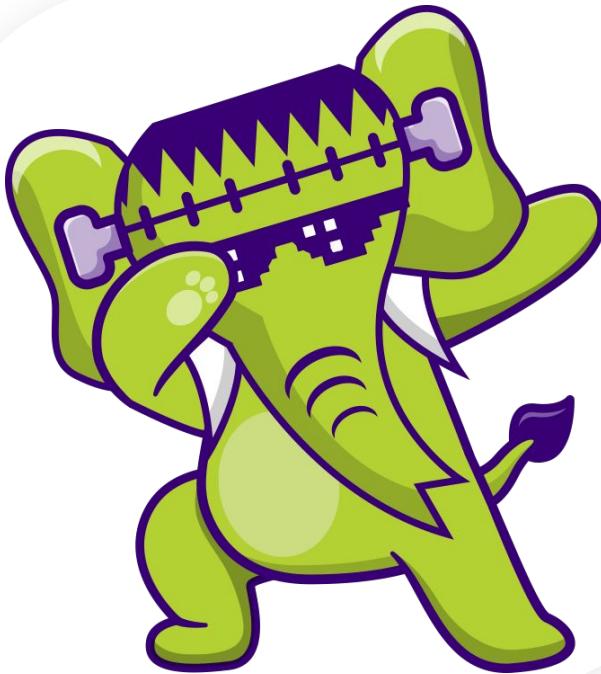
FrankenPHP

A modern **app server** for Drupal and PHP apps:

- **Replaces** NGINX+FPM or Apache+mod_php
- **Built for easy deployments:** compatible with Docker, can **embed your app** in the binary!
- **The fastest** PHP engine
- **103 Early Hints** support
- Built-in **real-time** support (Mercure)
- Compatible with **all existing PHP apps**: progressive enhancement



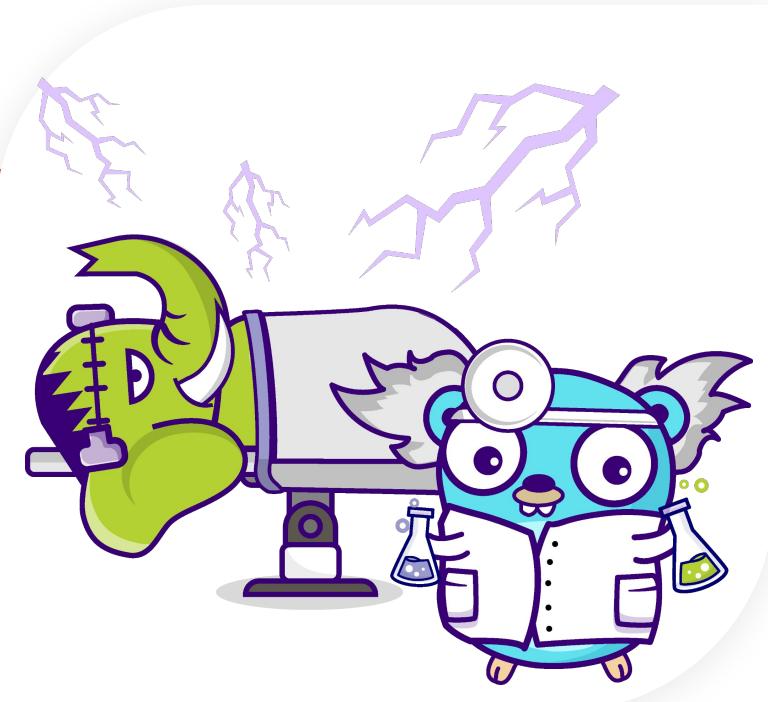
FrankenPHP: Modern PHP App Server



- Super simple to use:
 - ◆ Standalone binary (that can contain your app)
 - ◆ No external dependencies
 - ◆ **Batteries included**
- Built on top of the **Caddy** web server
 - ◆ All Caddy features and modules
 - ◆ Benefits from Go features
 - ◆ Extensible: in **Go**, in **C**, in **PHP**
- Designed for prod, CI and dev envs



The ❤️ of FrankenPHP



- **Standalone Go library:**
calls libphp (the official PHP engine as a C library) using `cgo`
- **Embed PHP in any Go program (static build)**
- **New SAPI for Go net/http**
 - ◆ **Caddy, Traefik, Kubernetes...**
 - ◆ **Your custom Go app**
- **Caddy module** using the library
- **Unique features**



FrankenPHP: Features

BASED ON
caddy®



HTTPS AUTOMATION

Automatic HTTPS certificate generation, renewal and revocation.



HTTP/2 & HTTP/3

Native support for HTTPS, HTTP/2 and HTTP/3.



BROTLI, ZSTANDARD & GZIP COMPRESSION

Modern compression formats are supported out-of-the-box.



STRUCTURED LOGGING

Bring a more defined format and details to your logging.



PROMETHEUS METRICS & TRACING

Built-in Prometheus support!



GRACEFUL RELOAD

Deploy your apps with zero downtime thanks to graceful reloads.



FrankenPHP: At a Glance



- Compatible with **existing** PHP apps (including Drupal, Laravel, Symfony and even WordPress)! 🐘
- **Static binary** ✨
- **Single Docker image** 🐳
- Entirely configurable 🛠
 - ◆ Caddyfile
 - ◆ php.ini
- Free software (as in free speech 🤝, and free beer 🍺)



[main](#) [1 Branch](#) [0 Tags](#)[Go to file](#)[Code](#)

 Thomthomars	Update Dockerfile to match with new caddyfile path (#17)	a3987c5 · 6 months ago	12 Commits
	Install pg_trgm extension as it's hard requirement for Dru...	2 years ago	
	Update Dockerfile to match with new caddyfile path (#17)	6 months ago	
	Drupal 10.1 (#16)	7 months ago	
	Install pg_trgm extension as it's hard requirement for Dru...	2 years ago	
	Install pg_trgm extension as it's hard requirement for Dru...	2 years ago	

[README](#) [Code of conduct](#)

Drupal on FrankenPHP

Run the popular [Drupal CMS](#) on top of [FrankenPHP](#), the modern app server for PHP.

Getting Started

```
git clone https://github.com/dunglas/frankenphp-drupal
cd frankenphp-drupal
docker compose pull --include-deps
docker compose up
```

Drupal is available on <https://localhost>.

About

Drupal on FrankenPHP

[docker](#) [php](#) [drupal](#) [docker-compose](#)
[frankenphp](#)[Readme](#)[Code of conduct](#)[Activity](#)[44 stars](#)[4 watching](#)[6 forks](#)[Report repository](#)

Sponsor this project

 **dunglas** Kévin Dunglas[Sponsor](#)[Learn more about GitHub Sponsors](#)

Contributors 3

-  **back-2-95** Marko Korhonen
-  **dunglas** Kévin Dunglas
-  **Thomthomars** Mars0test



Getting Started

```
git clone https://github.com/dunglas/frankenphp-drupal  
cd frankenphp-drupal  
docker compose up
```

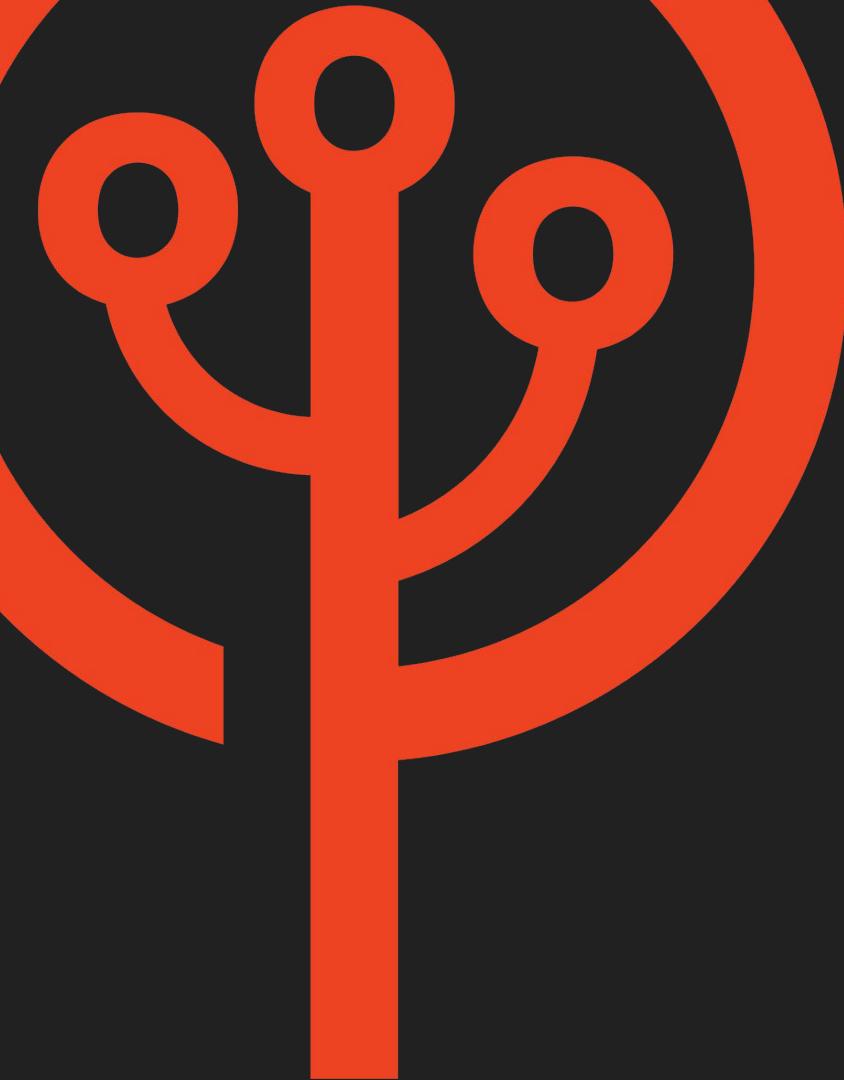


Non-Drupal apps (ex: Laravel)



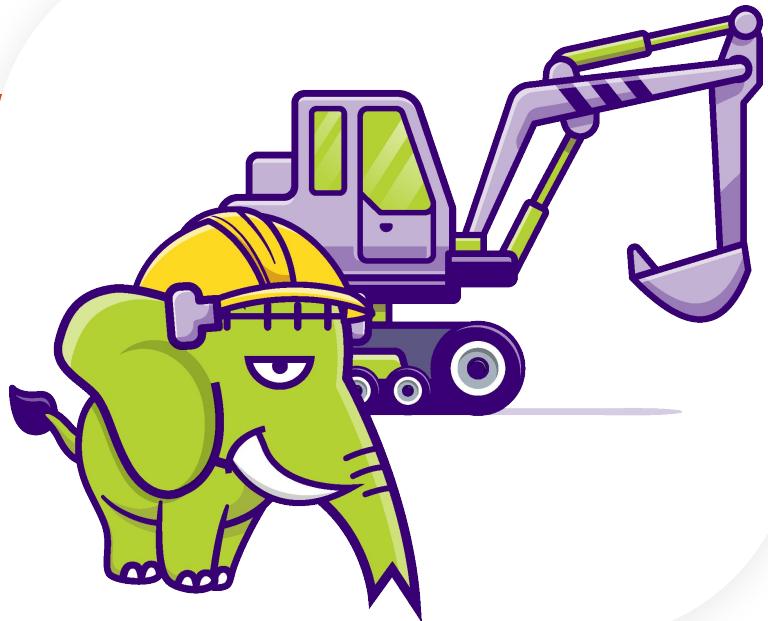
```
# Go in the root directory of your Laravel app  
and run:  
./frankenphp php-server --root=public/
```





THE WORKER MODE

FrankenPHP Superpower: The Worker Mode



- Boot your application **once**
- Keep it **in memory**
- **Process incoming requests** without having to boot your app again
- Relies on **goroutines and channels**
- Somewhat similar to RoadRunner
- Unlike RoadRunner
 - ◆ **runs in process:** no network calls, no gRPC
 - ◆ **uses plain old superglobals:** no PSR-7



The Runtime Component



Search Symfony Docs

Documentation Book Reference Bundles Cloud SymfonyCasts

Symfony stands with Ukraine.

Version: 6.1 (current) ▾

GETTING STARTED

Setup / Installation
Creating Pages
Routing / Generating URLs
Controllers
Templates / Twig
Configuration / Env Vars

ARCHITECTURE

Requests / Responses
Kernel
Services / DI
Events
Contracts
Bundles

THE BASICS

Databases / Doctrine
Forms
Tests

The Runtime Component decouples the bootstrapping logic from any global state to make sure the application can run with runtimes like PHP-PM, ReactPHP, Swoole, etc. without any changes.

Installation

```
$ composer require symfony/runtime
```



If you install this component outside of a Symfony application, you must require the `vendor/autoload.php` file in your code to enable the class autoloading mechanism provided by Composer. Read [this article](#) for more details.

Usage

The Runtime component abstracts most bootstrapping logic as so-called *runtimes*, allowing you to write front-controllers in a generic way. For instance, the Runtime component allows Symfony's `public/index.php` to look like this:

```
1 <?php
2 // public/index.php
3 use App\Kernel;
4
5 require_once dirname(__DIR__).'/vendor/autoload_runtime.php';
```



FrankenPHP worker for Symfony

Install

```
$ composer require \
    php-runtime/frankenphp-symfony
```

Run

```
$ frankenphp \
    -r public/ -w public/index.php
```



Worker mode: Benchmark (Laravel Octane)



Concurrency 1 for 5 seconds...

```
./vendor/bin/pest stress http://127.0.0.1:8000
```

Swoole Medium Request Duration 4.94 ms

RoadRunner Medium Request Duration 2.61 ms

FrankenPHP Medium Request Duration 0.88 ms

Concurrency 8 for 5 seconds...

```
./vendor/bin/pest stress http://127.0.0.1:8000  
--concurrency=8
```

Swoole Medium Request Duration 5.39 ms

RoadRunner Medium Request Duration 4.00 ms

FrankenPHP Medium Request Duration 1.59 ms

Benchmark by Nuno Maduro (Laravel Core Team)
using Pest's stress testing plugin on a M1 pro.





103 EARLY HINTS

103 Early Hints



Colin Bendell
@colinbendell · [Follow](#)



Today, HTTP 103 Early Hints ships with Chrome 103!

Why is this important for [#webperf](#)? How did [@Shopify](#) help make all merchant sites faster? (LCP over 500ms faster at p50!) 

Hint: A little collaboration w/ [@cloudflare](#) & [@googlechrome](#)



8:59 PM · Jun 21, 2022



[Read the full conversation on Twitter](#)



862



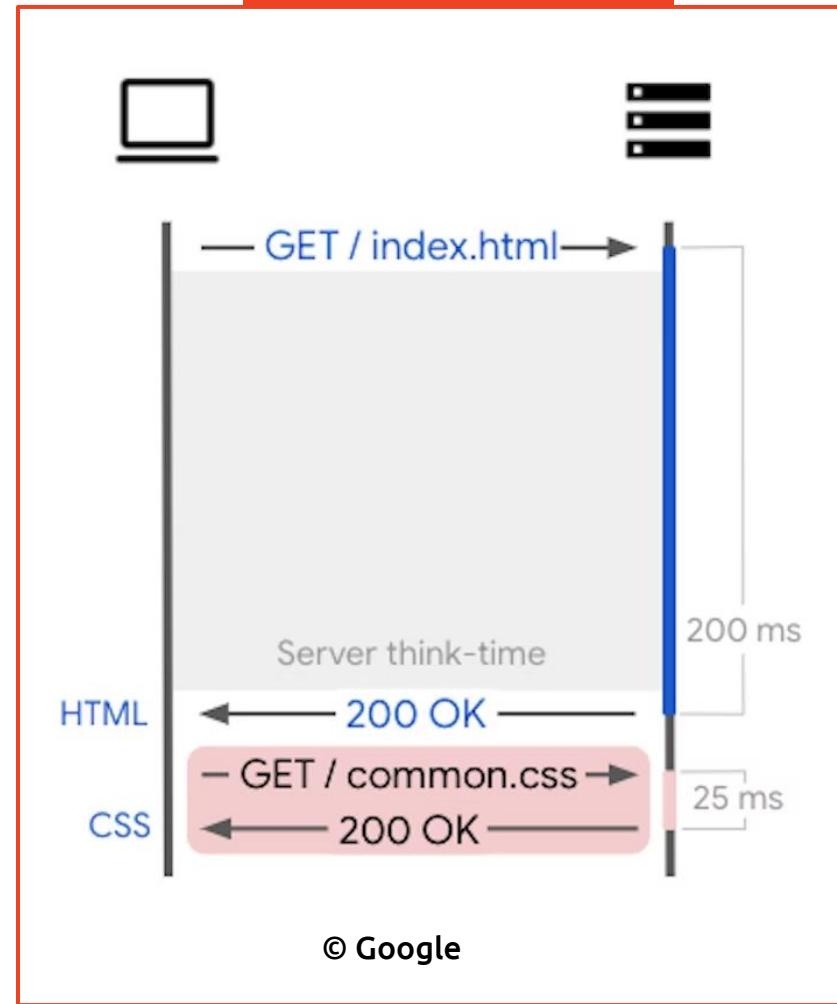
Reply



Copy link

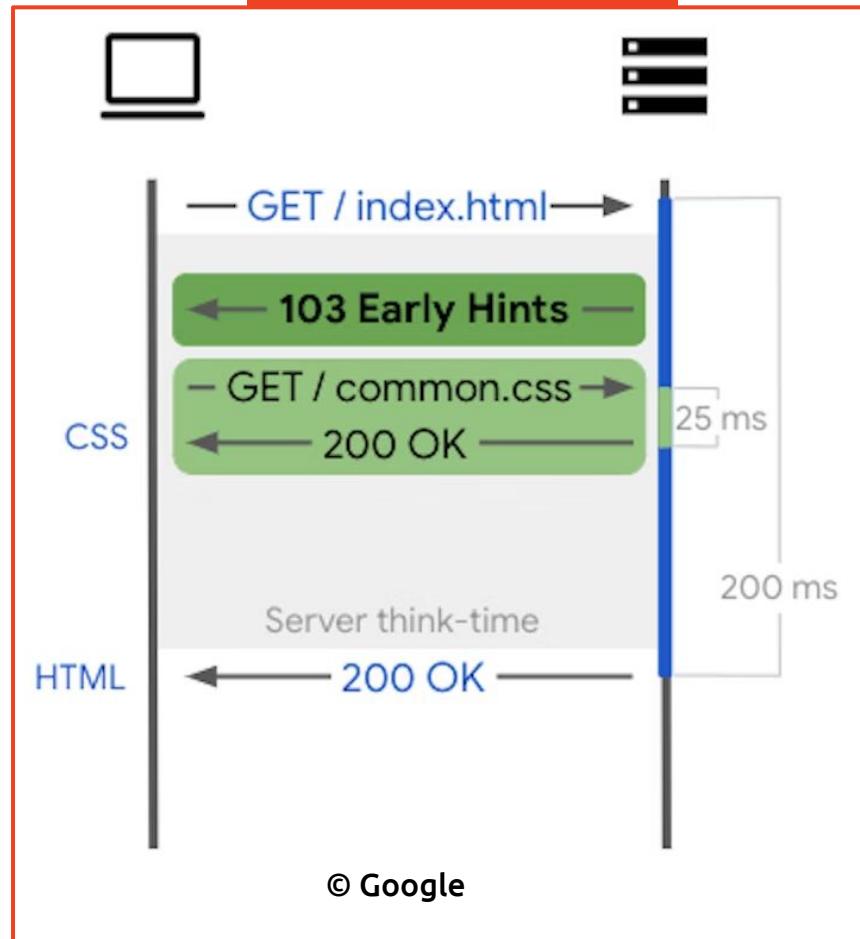


Typical Web Page



103 Early Hints

~30% LCP improvement 



Early Hints: Not Possible With PHP



nikic commented on May 21, 2021

Member



...

This is intended to be used in conjunction with HTTP Early Hints and other 1xx status codes, which require sending multiple sets of headers.



dunglas commented on May 25, 2021 • edited

Contributor



...

@**nikic**, I wonder if this patch really works:

This works for sure (but maybe by accident) with HTTP/1. However, with HTTP/2 and HTTP/3 things get harder: HTTP/2 and 3 are binary protocols. Responses must be sent using "special" binary frames (for instance, here is the Go implementation of the 103 status code for HTTP/3: [lucas-clemente/quic-go#3047](#)).

According to the CGI spec (which is inherited by FastCGI), a CGI script can only send one response, and this response cannot have a 1XX status code.

I fear that with the current implementation, the web server will send the second (final) response as part of the body frame instead of being "converted" in a new response frame. I've not tested this, but from my understanding of the spec, I don't see how it could behave differently. While it's not an issue with HTTP/1.1 (because it's a text protocol), it will break sites using HTTP/2 or 3.



Early Hints: Possible With Go

Go CHANGES ▾ YOUR ▾ DOCUMENTATION ▾ BROWS

Merged ★ 269997 ▾ net/http: allow sending 1xx responses 📱

Change Info		SHOW ALL ▾	REPLY
Submitted	May 17		
Owner	 GerritBot		net/http: a
Uploader	 Damien Neil +2		Currently, 103 Early H
Author	 Kévin Dunglas		



Early Hints: Supported by FrankenPHP ⚡

```
header('Link: </style.css>; rel=preload; as=style');  
headers_send(103);
```

// Your slow algorithms and SQL queries 😅

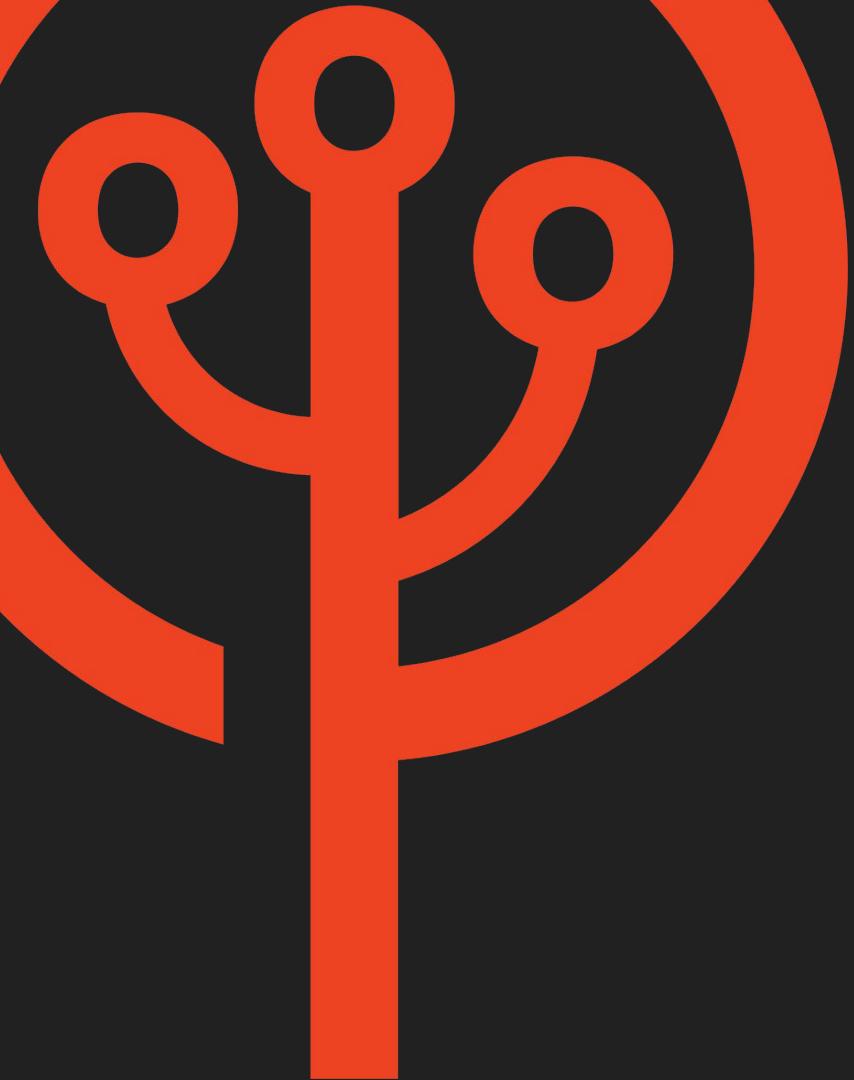
```
echo <<<'HTML'  
<!DOCTYPE html>  
<title>Hello FrankenPHP</title>  
<link rel="stylesheet" href="style.css">  
HTML;
```



Early Hints: Drupal + FrankenPHP ⚡

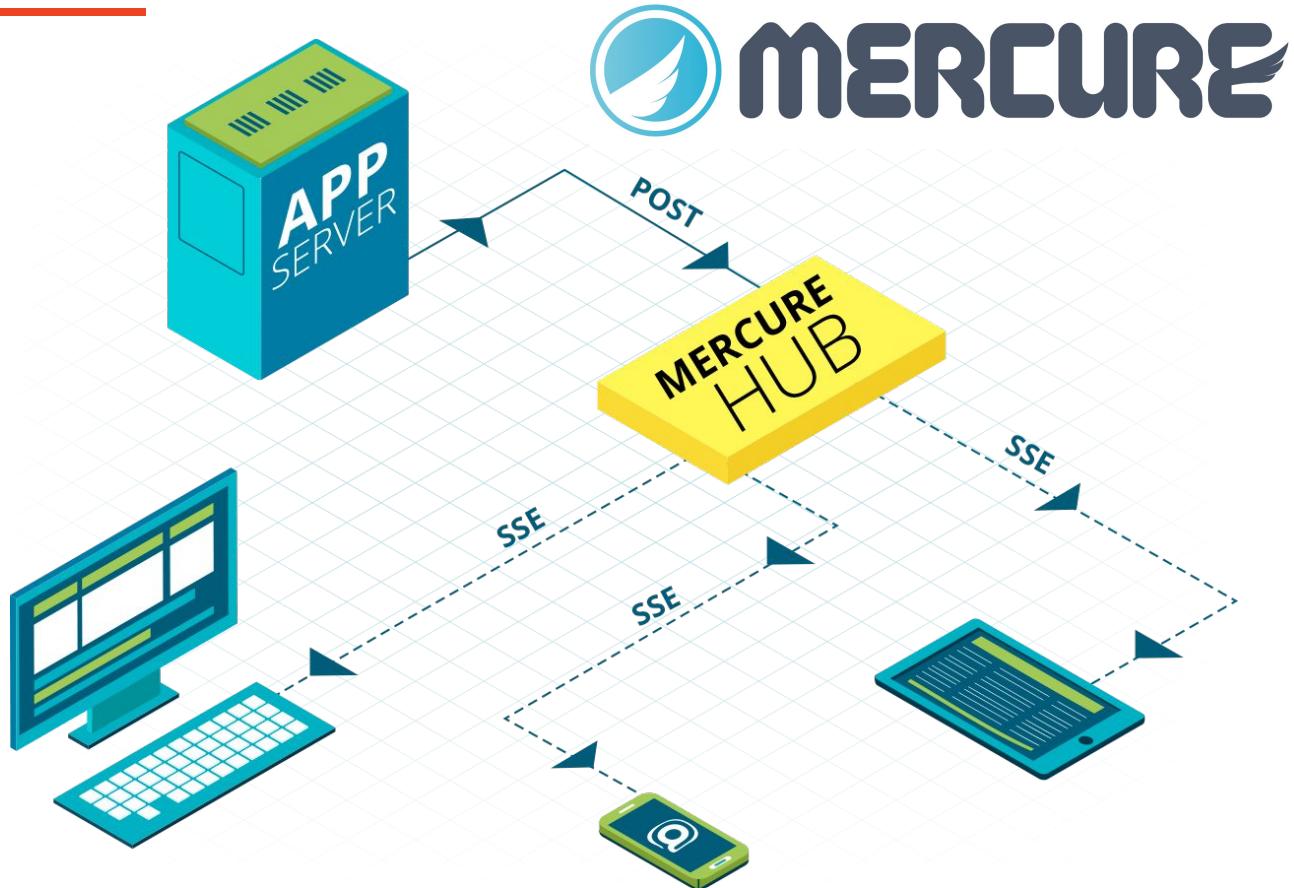
```
class ExampleController extends ControllerBase {  
    public function content() {  
        $r = new Response();  
        $r->headers->set('Link',  
            '</style.css>; rel=preload; as=style');  
        $r->sendHeaders(103);  
  
        // Your slow algorithms and SQL queries 😱  
        $r->setContent('The result of your computations');  
  
        return $r;  
    }:
```



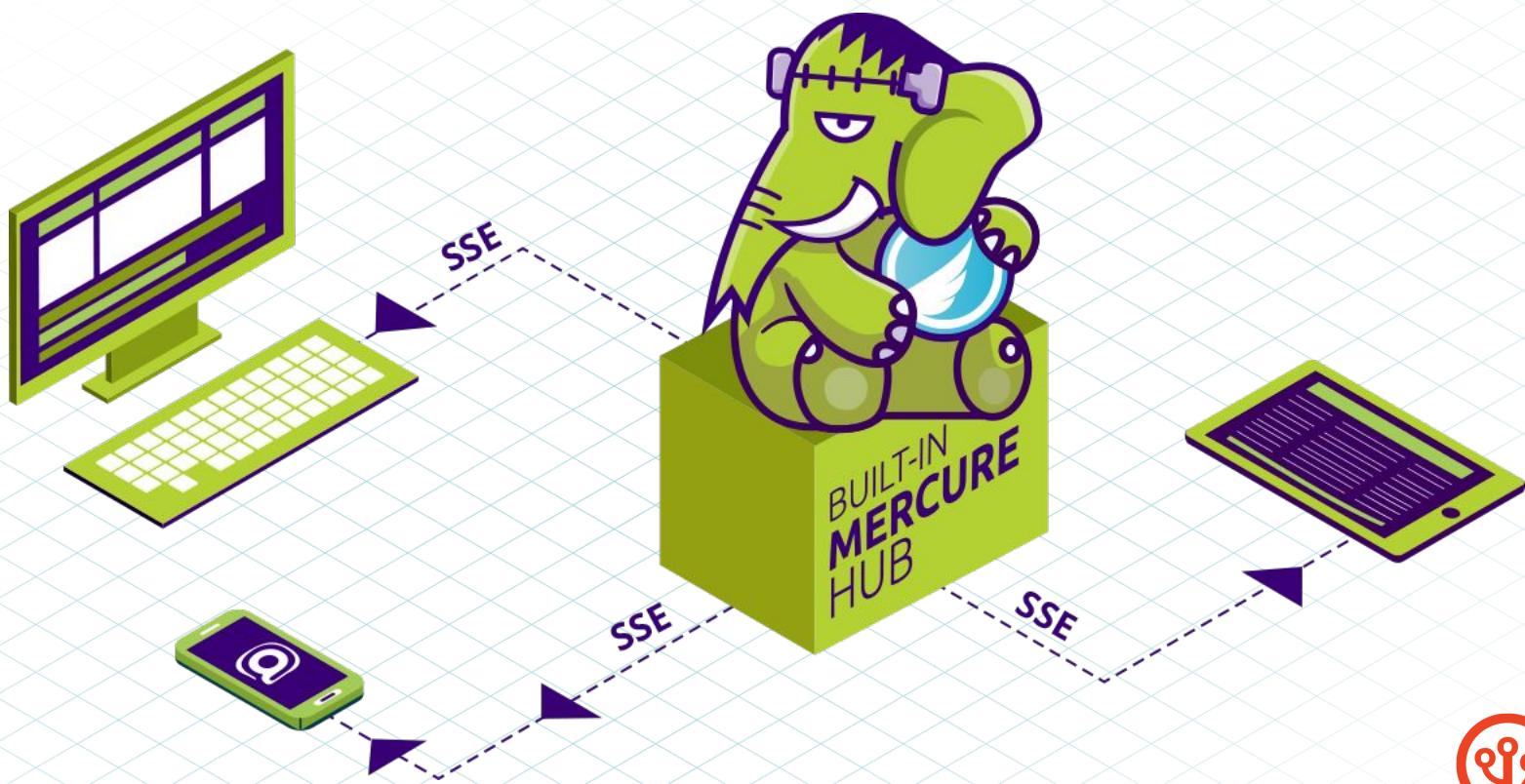
A large, stylized orange graphic of the planet Mercury is positioned on the left side of the slide. It features a central vertical axis with three circular rings extending from it at different angles, resembling a three-legged stool or a minimalist tree. The background behind the planet is black.

MERCURE INTEGRATION

Mercure: Alternative To WebSockets



FrankenPHP: Built-In Mercure Hub



FrankenPHP includes a Mercure hub!

```
// composer require symfony/mercure

class PublishController extends AbstractController
{
    #[Route("/publish", name: "publish")]
    public function publish(HubInterface $hub): Response
    {
        $update = new Update(
            'https://example.com/books/1',
            json_encode(['status' => 'OutOfStock'])
        );

        // The update is sent to the built-in Mercure hub
        $hub->publish($update);

        return new Response('A JS event is triggered in connected browsers!');
    }
}
```

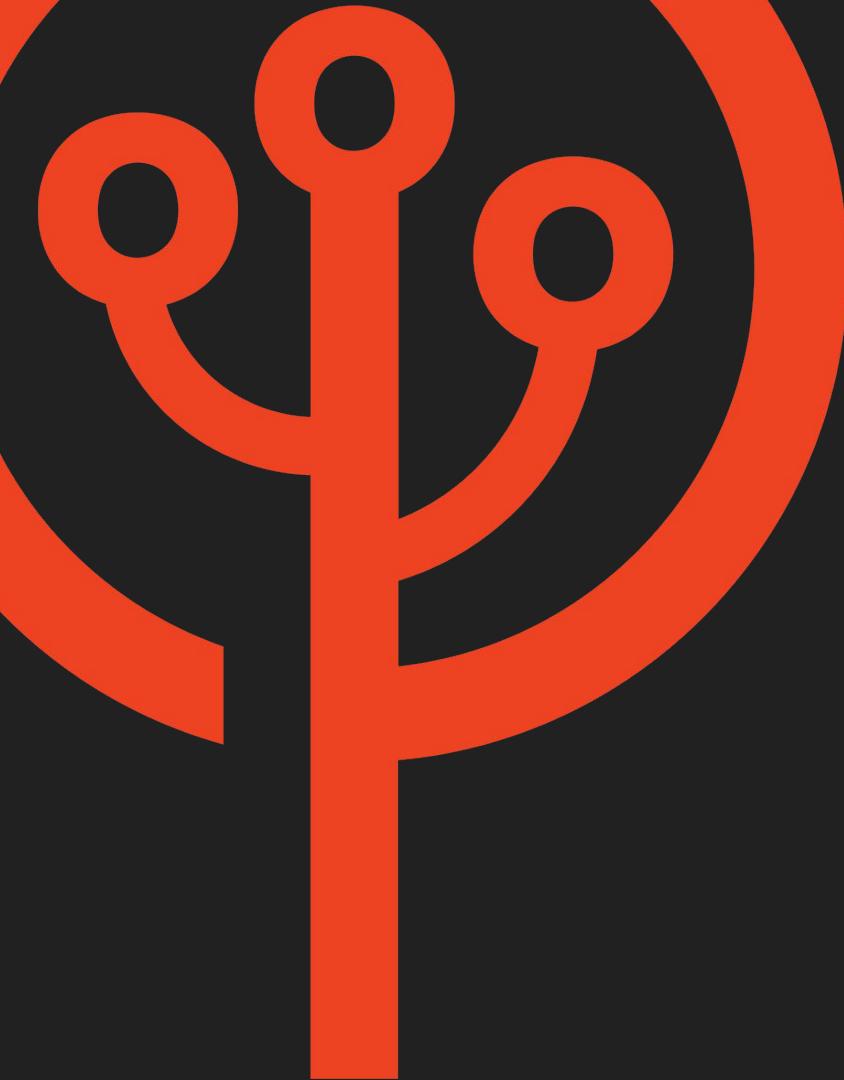


Broadcasting: Subscribe and Dispatch

```
// Subscribe in the browser, no 3rd-party lib required!
// Compatible with Livewire and all JS frameworks
const eventSource = new EventSource(
    '/.well-known/mercure?topic=https://example.com/books/1'
);

eventSource.onmessage = e => console.log(e);
```



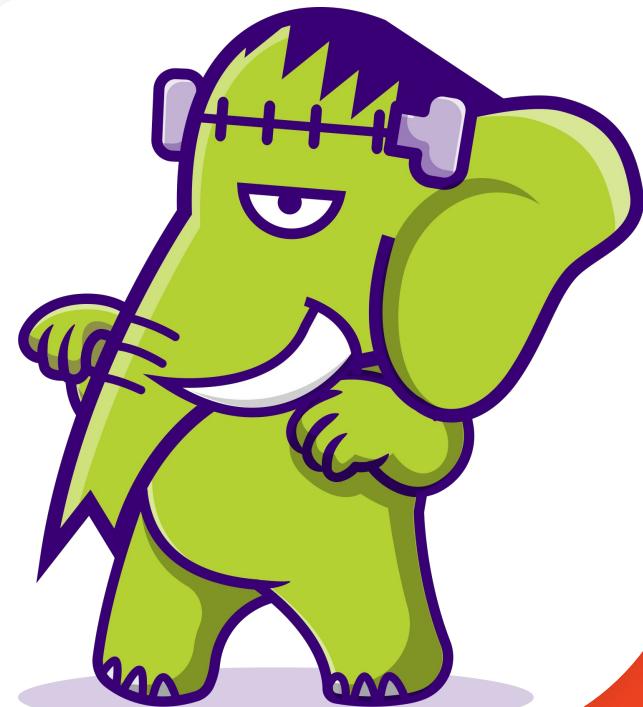


Package Your Apps As a Standalone Binary

A standalone PHP App:

We'll create a **static** binary containing:

- A web server (Caddy)
- PHP (and its dependencies)
- The extensions we need (and their dependencies)
- FrankenPHP
- Our Drupal/PHP app
- The dependencies of our Drupal app
- Our assets



PHP App As a Standalone Binary

```
# Prepare your app
$ composer install --no-dev -a

# Build
$ git clone https://github.com/dunglas/frankenphp && cd frankenphp
$ EMBED=/path/to/my/app ./build-static.sh

# Run in standard mode 🎉
$ ./dist/frankenphp-<os>-<arch> php-server
# Run a command
$ ./dist/frankenphp-<os>-<arch> php-cli my-command
```





Thanks!

Want to know more about
FrankenPHP or how
Les-Tilleuls.coop can help you?

contact@les-tilleuls.coop



frankenphp.dev



@dunglas