

Documentación » Dispatcher » Introducción

Introducción

Dispatcher para la Biblioteca PHP

Anonymous · 08/10/2026

Dispatcher para la Biblioteca PHP

 CI passing license AGPL-3.0+ downloads 161

Conector que expone las operaciones de `libredte/libredte-lib-core` de forma segura y agnóstica al transporte, usando [derafu/backbone-dispatcher](#) como base. Pensado como la capa intermedia entre LibreDTE y un bridge hacia otro lenguaje, aunque no depende de ninguno en particular. Repositorio: [libredte-lib-core-dispatcher](#).

Importante

El Dispatcher corre dentro de tu propio proceso PHP (lo llamas directo, no por red): tu software se convierte en un trabajo derivado y queda sujeto a los términos de la [licencia AGPL](#), igual que si usaras la [Biblioteca PHP](#) directamente. Si buscas integrarte sin ese efecto, revisa los [Servicios Web \(API\)](#), la [Imagen Docker \(API\)](#) o la [Imagen Docker \(Consola\)](#).

Instalación

Se instala como cualquier paquete de Composer:

```
composer require libredte/libredte-lib-core-dispatcher
```

Ejemplo

Un uso mínimo: arrancar el dispatcher, despachar una operación y revisar si tuvo éxito.

```
use libredte\lib\CoreDispatcher\Bootstrap;
use Derafu\BackboneDispatcher\ValueObject\OperationRequest;

// SafeDispatcherInterface.
$dispatcher = Bootstrap::boot();

$request = new OperationRequest(
    'billing',
```

```
'identifier',
'caf_loader',
'load',
['xml' => base64_encode($cafXml)],
);

// Nunca lanza excepciones.
$result = $dispatcher->dispatch($request);

if ($result->isSuccess()) {
    // Ya serializado (array), listo para cruzar un límite de
    // proceso o lenguaje.
    $caf = $result->getValue();
} else {
    // ProblemDetail (RFC 7807-like).
    $problem = $result->getProblem();
}
```

Nota

Revisa [Uso del Dispatcher](#) para el detalle completo de `Bootstrap::boot()`, `OperationRequest` y el manejo de errores.

Términos y condiciones de uso

Licenciado bajo AGPL-3.0+. Disponible en el archivo [COPYING](#).

Última actualización el 08/10/2026