

Documentación » Dispatcher » Arquitectura

Arquitectura

Cómo está armado este paquete

Anonymous · 08/10/2026

Arquitectura

Así está resuelto el cableado interno del paquete:

- `Bootstrap::boot(string $environment = 'prod', bool $debug = false)` es el único punto de entrada público. Todo el resto del cableado (`Deserializers`, la cadena `DirectDispatcher` → `TypedDispatcher` → `SafeDispatcher` de [derafu/backbone-dispatcher](#)) está declarado en `config/services.yaml`, no en PHP.
- `config/services.yaml` importa el de `libredte-lib-core` y sobrescribe `libredte.lib.core.project_dir: kernel.project_dir` se resuelve contra el paquete que actúa como punto de entrada, y ese rol lo cumple este paquete, no `libredte-lib-core`.
- `Derafu\BackboneDispatcher\Contract\ObjectFactoryInterface` está registrado `lazy: true` porque `DocumentBagDeserializer/SiiRequestDeserializer` necesitan ese mismo registro (para deserializar sus campos anidados) y a la vez son entradas de su propio mapa de `deserializers` — una referencia circular que `lazy: true` rompe.

Desarrollo

Para modificar este paquete, instala sus dependencias y corre las validaciones antes de proponer un cambio:

```
composer install

# PHPUnit con cobertura.
composer tests

# Nivel 5.
composer phpstan

# php-cs-fixer (dry-run).
composer phpcs
```

Última actualización el 08/10/2026