

Documentación » Dispatcher » Deserializadores

Deserializadores

Qué tipos cubren y cómo agregar uno nuevo

Anonymous · 08/10/2026

Deserializadores

Los 9 Deserializers de este paquete cubren únicamente interfaces/tipos de `libredte-lib-core` (o de paquetes Derafu genéricos que Core usa): `DocumentBagInterface`, `XmlDocumentInterface`, `CafInterface`, `CertificateInterface`, `EmisorInterface`, `ReceptorInterface`, `MandatarioInterface`, `DocumentEnvelopeInterface`, `SiiRequestInterface`.

Agregar soporte a una clase nueva

Sigue este orden para decidir qué construir:

1. Revisar si la clase ya tiene un `fromArray(array $data): self` estático. Si lo tiene, no hace falta escribir nada: `FromArrayDeserializer` (el `$fallback` del registro) ya la cubre automáticamente, sin registrar nada en `services.yaml`.
2. Si no lo tiene, ver si ya existe una Factory a la que delegar (ver `EmisorDeserializer`, `ReceptorDeserializer`, `MandatarioDeserializer`).
3. Si no hay Factory, construir el objeto directamente, delegando los campos que a su vez son objetos en el `ObjectFactoryInterface` inyectado (ver `DocumentBagDeserializer`).
4. Si la construcción no tiene una única forma fija, decidir según qué claves están presentes (ver `CertificateDeserializer`, que elige entre `loadFromData()/loadFromKeys()`).
5. Todo Deserializer empieza narrowing su entrada con `assertArray()/assertString()` (heredados de `Abstract\AbstractDeserializer`).

`DocumentBag` no tiene Deserializer propio para sus campos `document` y `documentType`: siempre reciben `null`. Agregar los suyos cuando se necesiten.

Última actualización el 08/10/2026