

Symfony Integration

Symfony Integration

Anonymous · 09/09/2026

Symfony Integration

`derafu/translation` doesn't *integrate with* `symfony/translation` the way a bridge or an adapter would — it's built directly **on top of** it. `TranslatorFactory::create()` returns a real `Symfony\Component\Translation\Translator` instance, not a custom wrapper. Everything this library adds (translatable exceptions, resource discovery across directories) sits alongside that instance, not between your code and it.

No Framework Dependency

The only hard dependencies are `symfony/translation`, `symfony/translation-contracts`, `symfony/yaml`, and the `intl` PHP extension. There is no dependency on `symfony/framework-bundle` or `symfony/http-kernel`, and there will not be one — this library works in any PHP application, framework or not.

Using the Real `Translator` Directly

Because `TranslatorFactory::create()` gives you the actual Symfony class, every native method is available — this library doesn't hide or wrap it:

```
use Derafu\Translation\TranslatorFactory;

$translator = TranslatorFactory::create('es', ['es', 'en']);

// Native Symfony\Component\Translation\Translator methods, all available.
$translator->setLocale('en');
$translator->getFallbackLocales();
$translator->getCatalogue('es')->all('errors+intl-icu');
```

If you need a loader this library doesn't pre-wire (for example, `IcuResFileLoader` or `IcuDatFileLoader`, or your own custom loader — see [Resource Providers](#)), add it the normal Symfony way:

```
$translator->addLoader('my_format', new MyCustomLoader());
```

Locale-Aware Contracts

`Symfony\Component\Translation\Translator` also implements `Symfony\Component\Translation\TranslatorBagInterface` and `Symfony\Contracts\Translation\LocaleAwareInterface`, so it can be used anywhere those contracts are expected, not just where `Symfony\Contracts\Translation\TranslatorInterface` is required — which is the interface `TranslatableExceptionTrait::trans()` itself expects.

Dependency Injection

If your application uses `symfony/dependency-injection` (with or without the rest of the Symfony Framework), see [Advanced Usage](#) for the wiring recipe this package ships.

Last updated on 09/09/2026