

Docs » Views and Templates Category » Twig Components Project » Symfony

Symfony

Symfony

Anonymous · 20/10/2026

Symfony

The package is not a bundle: three settings of your application register the components, their templates and their translations. They work with Symfony 8.

Requirements

- `symfony/framework-bundle` and `symfony/twig-bundle`.
- `symfony/ux-twig-component`, with its bundle enabled (its recipe does it when you install it).

```
composer require derafu/twig-components
```

Settings

The templates

The templates of the components are in the `components` directory of `resources/templates`. Give that path to Twig, and Symfony UX finds them:

```
# config/packages/twig.yaml
twig:
  paths:
    '%kernel.project_dir%/vendor/derafu/twig-components/resources/templates': ~
```

The translations

The texts of the components are translated in the catalogues of `resources/translations`. Give that path to the translator:

```
# config/packages/translation.yaml
framework:
  translator:
    paths:
      - '%kernel.project_dir%/vendor/derafu/twig-
```

```
components/resources/translations'
```

The catalogues are in Spanish (the texts of the components are their English ids). Set the locale of your application to use them:

```
# config/packages/framework.yaml
framework:
    default_locale: es
```

The components

The classes of the components are services that Symfony UX discovers by their `#[AsTwigComponent]` attribute. Register the directory with `autoconfigure`:

```
# config/services.yaml
services:
    _defaults:
        autowire: true
        autoconfigure: true

    Derafu\TwigComponents\Component\:
        resource: '../vendor/derafu/twig-components/src/Component/'
```

Use

With the three settings, the tags work in any template of the application:

```
<!-- TWIG_BLOCK_0 -->
```

The page also needs the styles and the scripts of Bootstrap and Font Awesome, and the CSS of the components: see [Styles](#).

Strict Mode

The templates are written for `strict_variables`, the mode that Symfony has in `dev`: they read the optional keys of an item in a way that does not fail when the key is not there. The tests of the package render every component and every example of this documentation in that mode, in a Symfony kernel.