

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

Derafu

Derafu

Anonymous · 20/10/2026

Derafu

[derafu/twig](#) creates a Twig environment without Symfony and registers Symfony UX Twig Components in it. This is how the components are used in a Derafu site.

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

[derafu/twig](#) does not know the components of this package, and this package does not depend on [derafu/twig](#): the one that builds the environment is the one that gives them.

Directly

Give the environment the templates of the package in [paths](#), and the components in [components](#). The directory of the classes has all of them:

```
use Derafu\Twig\Provider\DirectoryComponentProvider;
use Derafu\Twig\Service\TwigService;

$vendor = __DIR__ . '/vendor/derafu/twig-components';

$twig = new TwigService([
    'paths' => [__DIR__ . '/templates', $vendor . '/resources/templates'],
    'components' => new DirectoryComponentProvider($vendor . '/src/Component'),
]);

$data = [];

echo $twig->renderFromstring('<!--TWIG_BLOCK_0-->', $data);
```

The [components](#) option also takes the classes of the components, or their instances (see [Components](#)).

With the Container

The file `resources/config/twig-components-services.yaml` registers every component with the tag

`derafu twig.component`, and the translations of the package with the tag `derafu_translation.resource_provider`. Import it in your services:

```
imports:
  - { resource: '../vendor/derafu/twig-components/resources/config/twig-components-services.yaml' }
```

The `components` option of `derafu/twig` takes that iterable (see [Components](#)), so the creator of the environment gives it with:

```
components: !tagged_iterator derafu twig.component
```

With the Renderer

[derafu/renderer](#) gives its `components` option to its Twig environment:

```
use Derafu\Renderer\Factory\RendererFactory;
use Derafu\Twig\Provider\DirectoryComponentProvider;

$vendor = __DIR__ . '/vendor/derafu/twig-components';

$renderer = RendererFactory::create([
    'paths' => [__DIR__ . '/templates', $vendor . '/resources/templates'],
    'components' => new DirectoryComponentProvider($vendor . '/src/Component'),
]);

echo $renderer->renderFromString('<!--TWIG_BLOCK_1-->');
```

Last updated on 20/10/2026