Programowanie

# Symfony 7 — routing i kontrolery od podstaw

Admin Główny 21.09.2026 504 wyświetleń 3 komentarzy

Dostępne w językach: PL EN



## Routing w Symfony 7

Symfony 7 używa atrybutów PHP 8 do definiowania tras. To zdecydowanie czytelniejsze niż YAML czy XML.

### Podstawowy kontroler

```
<?php

namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\Abst
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Attribute\Route;
```

```
class BlogController extends AbstractController
{
    #[Route('/blog', name: 'app_blog_index', metho
    public function index(): Response
    {
        return $this->render('blog/index.html.twig
    }

    #[Route('/blog/{slug}', name: 'app_blog_show',
    public function show(string $slug): Response
    {
        // $slug jest automatycznie wstrzykiwany z
        return $this->render('blog/show.html.twig',
    }
}
```

## Wymagania dla parametrów

Możesz ograniczyć format parametrów za pomocą `requirements`:

```
#[Route(
    '/blog/{id}',
    name: 'blog_show_by_id',
    requirements: ['id' => '\d+'],
    methods: ['GET']
)]
public function showById(int $id): Response
{
    // ...
}
```

## Grupy tras

Atrybut `#[Route]` można też ustawiać na klasie — wszystkie trasy w kontrolerze dziedziczą prefiks:

```php
#[Route('/api/v1', name: 'api_v1_')]
class ApiController extends AbstractController
{
    #[Route('/users', name: 'users')]    // → /api
    public function users(): JsonResponse { /* ...
}
```

## Dependency Injection w kontrolerze

### Constructor injection (zalecane)

```php
class ArticleController extends AbstractController
{
    public function __construct(
        private readonly ArticleRepository $articl
        private readonly SluggerInterface  $slugge
    ) {}

    #[Route('/articles')]
    public function index(): Response
    {
        $all = $this->articles->findPublished();
        return $this->render('article/index.html.t
    }
}
```

### Action injection (dla jednorazowych zależności)

```php
#[Route('/articles/new')]
public function new(Request $request, EntityManage
{
    // $request i $em wstrzyknięte tylko do tej ak
}
```

## Zwracanie danych

| Metoda | Klasa | Kiedy używać |
|---|---|---|
| `render()` | `Response` | Strony HTML (Twig) |
| `json()` | `JsonResponse` | API endpoints |
| `redirect()` | `RedirectResponse` | Przekierowania |
| `file()` | `BinaryFileResponse` | Pliki do pobrania |

### Przykład JSON

```
#[Route('/api/articles', format: 'json')]
public function list(ArticleRepository $repo): Jso
{
    return $this->json(
        $repo->findAll(),
        Response::HTTP_OK,
        [],
        ['groups' => 'article:list']
    );
}
```

## Debugowanie tras

```
# lista wszystkich tras
php bin/console debug:router

# szczegóły konkretnej trasy
php bin/console debug:router app_blog_show
```

```
# dopasowanie URL do trasy
php bin/console router:match /blog/moj-artykul
```

> ***Tip:*** *W środowisku deweloperskim pasek Symfony Profiler (na dole strony) pokazuje aktualną trasę, kontroler i czas wykonania.*

Tagi: #PHP #Symfony #Twig

Udostępnij: Facebook X LinkedIn Kopiuj link

## Komentarze (3)

J

**Jan Kowalski** 21.10.2026, 14:27

Świetne wyjaśnienie grup routingu. Używam prefiksów na /api i gra muzyka.

A

**Anna Nowak** 21.10.2026, 14:27

Warto dodać sekcję o route requirements z wyrażeniami regularnymi.

A

**Admin Główny** 21.10.2026, 14:27

Masz rację, dodam w następnej edycji. Dzięki za feedback!

Zaloguj się, aby dodać komentarz.

## FactoryCode

Projekt demonstracyjny zbudowany na Symfony 8, PHP 8.4 i PostgreSQL 16. Tłumaczenia zarządzane przez panel admina z AI.

### NAWIGACJA

- Start
- Blog
- Projekty
- Kontakt
- Polityka prywatności

### KONTAKT

- kontakt@factorycode.pl
- GitHub

---

© 2026 FactoryCode. Wszelkie prawa zastrzeżone.

Zbudowane z Symfony 8 & Claude AI