
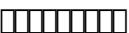



Symfony

-  _____
- Flex _____
-  _____



10 Symphony

 Symfony

Composer PHP 8.1

```
composer create-project symfony/skeleton quick_tour
```

quick_tour/ Symphony

```
quick_tour/
├── .env
├── bin/console
├── composer.json
├── composer.lock
├── config/
├── public/index.php
├── src/
├── symfony.lock
├── var/
└── vendor/
```

```
symfony server:start
```

http://localhost:8000

15

Symfony

API

Web

src/Controller

DefaultController

index

```
// src/Controller/DefaultController.php
namespace App\Controller;

use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Attribute\Route;

class DefaultController
{
    #[Route('/', name: 'index')]
    public function index(): Response
    {
        return new Response('Hello!');
    }
}
```

index()

http://localhost:8000/ Symfony URL

Symfony Response

JSON HTML

```
// src/Controller/DefaultController.php
namespace App\Controller;

use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Attribute\Route;

class DefaultController
{
-   #[Route('/', name: 'index')]
+   #[Route('/hello/{name}', name: 'index')]
    public function index(): Response
    {
```



```
        return new Response('Hello!');
    }
}
```

URL /hello/* {name}

```
<?php
// src/Controller/DefaultController.php
namespace App\Controller;

use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Attribute\Route;

class DefaultController
{
    #[Route('/hello/{name}', name: 'index')]
    - public function index()
    + public function index(string $name): Response
    {
        - return new Response('Hello!');
        + return new Response("Hello $name!");
    }
}
```

http://localhost:8000/hello/Symfony Hello Symfony URL {name} \$name

DefaultController

```
// src/Controller/DefaultController.php
namespace App\Controller;

use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Attribute\Route;

class DefaultController
{
    // ...
}
```



```
#[Route('/simplicity', methods: ['GET'])]

public function simple(): Response
{
    return new Response('Simple! Easy! Great!');
}
```



Diagram illustrating the relationship between Twig, PHP, and HTML:

- Twig** (represented by a small box) is used to generate **HTML** (represented by a large box).
- Twig** is also used to generate **PHP** (represented by a large box).
- PHP** is used to generate **HTML**.

```
"require": {
    "...",
    "symfony/console": "^6.1",
    "symfony/flex": "^2.0",
    "symfony/framework-bundle": "^6.1",
    "symfony/yaml": "^6.1"
}
```

composer require twig

RECIPIES.md

config/bundles.php

3

• config/packages/twig.yaml Twig

• config/packages/test/twig.yaml

Twig

• templates/

base.html.twig

Twig

Flex

Twig

```
<?php
// src/Controller/DefaultController.php
namespace App\Controller;

use Symfony\Component\Routing\Attribute\Route;
use Symfony\Component\HttpFoundation\Response;
+ use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;

- class DefaultController
+ class DefaultController extends AbstractController
{
    #[Route('/hello/{name}', methods: ['GET'])]
    public function index(string $name): Response
    {
-     return new Response("Hello $name!");
+     return $this->render('default/index.html.twig', [
+         'name' => $name,
+     ]);
    }
}
```

AbstractController

render()

```
{# templates/default/index.html.twig #}
<h1>Hello {{ name }}</h1>
```

{{ name }}

name

Twig

h1

HTML

base.html.twig


```
{# templates/default/index.html.twig #}

{% extends 'base.html.twig' %}


{% block body %}

    <h1>Hello {{ name }}</h1>

{% endblock %}
```

base.html.twig ☐ HTML ☐Profiler



--	--	--	--	--

Symfony

```
composer require profiler
```

Flex

 Flex

 Symfony  Profiler

--	--	--	--	--	--	--	--	--

[illegible][illegible][illegible]

Web

--	--	--	--	--	--	--	--

[illegible]

API

API []

JSON

```
// src/Controller/DefaultController.php

namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\JsonResponse;
use Symfony\Component\Routing\Attribute\Route;

class DefaultController extends AbstractController
{
    // ...

    #[Route('/api/hello/{name}', methods: ['GET'])]
```



```
public function apiHello(string $name): JsonResponse
{
    return $this->json([
        'name' => $name,
        'symfony' => 'rocks',
    ]);
}
```

```
composer require api
```

api-platform/api-pack Symfony pack Doctrine ORM Flex 5

product API Product

#[ApiResource]

```
// src/Entity/Product.php

namespace App\Entity;

use ApiPlatform\Core\Annotation\ApiResource;
use Doctrine\ORM\Mapping as ORM;

#[ORM\Entity]
#[ApiResource]
class Product
{
    #[ORM\Id]
    #[ORM\GeneratedValue(strategy: 'AUTO')]
    #[ORM\Column(type: 'integer')]
    private int $id;

    #[ORM\Column(type: 'string')]
    private string $name;

    #[ORM\Column(type: 'integer')]
    private int $price;

    // ...
}
```




```
php bin/console debug:router
```

```
-----
```

Name	Method	Path
api_products_get_collection	GET	/api/products.{_format}
api_products_post_collection	POST	/api/products.{_format}
api_products_get_item	GET	/api/products/{id}.{_format}
api_products_put_item	PUT	/api/products/{id}.{_format}
api_products_delete_item	DELETE	/api/products/{id}.{_format}
...		

```
-----
```



composer remove api

[illegible][illegible]

Symfony
Symfony



Flex



composer require logger

[illegible]

LoggerInterface



```
// src/Controller/DefaultController.php

namespace App\Controller;

use Psr\Log\LoggerInterface;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Attribute\Route;

class DefaultController extends AbstractController
{
    #[Route('/hello/{name}', methods: ['GET'])]
    public function index(string $name, LoggerInterface $logger): Response
    {
        $logger->info("Saying hello to $name!");

        // ...


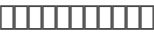
    }
}
```



```
var/log/dev.log
```





Symfony  LoggerInterface
Logger 

Symfony 



Logger 

Twig



Symfony



php bin/console debug:autowiring

Describes a logger instance.

Psr\Log\LoggerInterface - alias:monolog.logger

Request stack that controls the lifecycle of requests.


Symfony\Component\HttpFoundation\RequestStack - alias:request_stack

RouterInterface is the interface that all Router classes must implement.

Symfony\Component\Routing\RouterInterface - alias:router.default

[...]



“Hello”  “Yo”



```
// src/GreetingGenerator.php
namespace App;

class GreetingGenerator
{
    public function getRandomGreeting(): string
    {
        $greetings = ['Hey', 'Yo', 'Aloha'];
        $greeting = $greetings[array_rand($greetings)];
    }
}
```



```
    return $greeting;
  }
}
```



```
// src/Controller/DefaultController.php

namespace App\Controller;

use App\GreetingGenerator;
use Psr\Log\LoggerInterface;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Attribute\Route;

class DefaultController extends AbstractController
{
    #[Route('/hello/{name}', methods: ['GET'])]
    public function index(string $name, LoggerInterface $logger, GreetingGenerator $generator): Response
    {
        $greeting = $generator->getRandomGreeting();

        $logger->info("Saying $greeting to $name!");

        // ...
    }
}
```

GreetingGenerator

[illegible]

```
<?php
    // src/GreetingGenerator.php
+ use Psr\Log\LoggerInterface;

    class GreetingGenerator
    {
+     public function __construct(
+         private LoggerInterface $logger,
```



```
+ ) {  
+ }  
  
public function getRandomGreeting(): string  
{  
    // ...  
  
+     $this->logger->info('Using the greeting: '.$greeting);  
  
    return $greeting;  
}  
}
```

Twig

--	--	--	--	--	--

Diagram illustrating the layout of a 100-byte memory block:

- Symfony** (0-19): 20 bytes, divided into two 10-byte sub-sections.
- Twig** (20-79): 60 bytes, divided into two 30-byte sub-sections.
- greet** (80-99): 20 bytes, divided into two 10-byte sub-sections.

```
// src/Twig/GreetExtension.php

namespace App\Twig;

use App\GreetingGenerator;
use Twig\Extension\AbstractExtension;
use Twig\TwigFilter;

class GreetExtension extends AbstractExtension
{
    public function __construct(
        private GreetingGenerator $greetingGenerator,
    ) {
    }

    public function getFilters(): array
    {
        return [
            new TwigFilter('greet', [$this, 'greetUser']),
        ];
    }
}
```



```
public function greetUser(string $name): string
{
    $greeting = $this->greetingGenerator->getRandomGreeting();

    return "$greeting $name!";
}
```

```
{# templates/default/index.html.twig #}


{# Will print something like "Hey Symfony!" #}

<h1>{{ name|greet }}</h1>
```

Diagram illustrating the relationship between Symfony, AbstractExtension, Twig, and another Symfony instance:

- Row 1: A 6-box block, the text "Symfony", a 6-box block, the text "AbstractExtension", a 6-box block, the text "Twig", and a 6-box block.
- Row 2: A single continuous block of 30 boxes.
- Row 3: A 6-box block followed by a 2-box block.

--	--	--	--	--	--	--

The diagram illustrates the relationship between a 'Symfony' component and its 'Dependency'. It consists of two horizontal bars. The top bar is labeled 'Symfony' and is composed of 20 small squares. The bottom bar is labeled 'Dependency' and is composed of 10 small squares. The 'Dependency' bar is positioned below the 'Symfony' bar, and its length is half that of the 'Symfony' bar, indicating a 1:2 ratio.

A horizontal number line with arrows at both ends. It is marked with numbers from 0 to 100 in increments of 10 (0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100). A vertical tick mark is placed at the number 10.

[illegible]

Symfony [] [] [] [] [] [] [] [] [] [] Symfony [] [] [] [] [] [] [] [] [] [] dev [] prod [] test []
[] [] [] [] [] [] [] [] [] [] when@[] [] [] [] config/[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] []


```
# config/packages/routing.yaml
```

```
framework:
```

```
    router:
```

```
        utf8: true
```

```
when@prod:
```

```
    framework:
```

```
        router:
```

```
            strict_requirements: null
```

```
#####
```

```
█
```

```
#####
```

```
APP_ENV=
```

```
dev
```

```
prod
```

```
# .env
```

```
- APP_ENV=dev
```

```
+ APP_ENV=prod
```

```
#####
```

```
dev
```

```
███
```

```
#####
```

```
-
```

```
#####
```

```
Symfony
```

```
(PaaS)
```

```
Docker
```

```
Symfony
```

```
#####
```

```
.env
```

```
#####
```

```
# .env
```

```
###> symfony/framework-bundle ###
```

```
APP_ENV=dev
```

```
APP_SECRET=cc86c7ca937636d5ddf1b754beb22a10
```

```
###< symfony/framework-bundle ###
```

```
#####
```

```
#####
```

```
ORM
```

```
Doctrine ORM
```

```
composer require doctrine
```


Flex .env

```
###> symfony/framework-bundle ###
APP_ENV=dev
APP_SECRET=cc86c7ca937636d5ddf1b754beb22a10
###< symfony/framework-bundle ###

+ ###> doctrine/doctrine-bundle ###
+ # ...
+ DATABASE_URL=mysql://db_user:db_password@127.0.0.1:3306/db_name
+ ###< doctrine/doctrine-bundle ###
```

DATABASE_URL doctrine.yaml
Flex

Symfony
Symfony