



Symphony

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

[illegible][illegible]

composer require logger

```

classDiagram
    class Monolog {
        array[16]
        array[7]
    }
    class LoggerInterface {
        array[16]
    }

```

```
// 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!");

        // ...


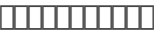
    }
}
```

```

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!");

        // ...
    }
}
```



```
<?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 memory page. The page is divided into three rows. The first row contains 'Symfony' followed by a block of 16 small squares. The second row contains a block of 24 small squares, followed by 'Twig' followed by a block of 5 small squares, and 'greet' at the end. The third row contains a block of 10 small squares, followed by 'AbstractExtension' followed by a block of 3 small squares.

```
// 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 24-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.

[illegible][illegible]

```
# 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