

Framework Web

CM : Jean-Michel Couvreur (10h), Frédéric Loulergue (8h)

TP (20h) : Jean-Michel Couvreur (MIAGE+Ing.), Frédéric Loulergue (Ing.),
Aurélien Emmanuel (MIAGE), Mouhamad Said (MIAGE+Ing)

Organisation du cours

- Symfony : 5 cours + 5 TP
- CC Projet Symfony par équipe de 3-4 étudiants sur une semaine
- Angular : 4 cours + 5 TP
- CT 2-3 jours par équipe de 3-4 étudiants

Plan du cours

- (1) Tour de Symfony
- (2) Les entités et formulaires
- (3) L'authentification
- (4) API Platform



<https://symfony.com/doc/current/the-fast-track/fr/index.html>



<https://symfony.com>

Blog de [Lior CHAMLA](#)



https://youtube.com/playlist?list=PLIxQJeQRaKDTxU_kTI_0Ltn4WGZFIB1xtT

La boite à outils



<https://www.php.net>



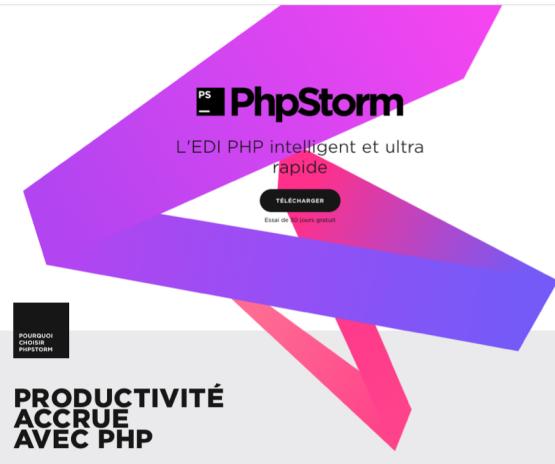
A Dependency Manager for PHP
<https://getcomposer.org>



Symfony

Symfony is a set of reusable
PHP *components*...
... and a PHP *framework* for web projects

<https://symfony.com/download>



<https://www.jetbrains.com/fr-fr/phpstorm/>



<https://www.docker.com>

1. Environnement de développement



Terminal connecté au conteneur docker



Un linux minimal avec PHP 8,
Composer et Symfony

Editeur sur le dossier



Un dossier partagé contenant le
code du projet

Créons rapidement l'environnement : Docker

(1) Créer un dossier de travail

(2) Dans ce dossier, copier les fichiers :

docker-compose.yml

Dockerfile

et créer un dossier dev

(3) Dans un terminal, lancer les commandes :

docker-compose build

docker-compose up -d

docker ps

docker exec -ti my-web-container bash

(4) Dans le conteneur, lancer les commandes :

symfony new monprojet --full

cd monprojet

symfony server:start --no-tls --d

(5) Ouvrir avec Chrome <http://localhost:5000>



Un linux minimal avec PHP 8,
Composer et Symfony

Un dossier partagé contenant le
code du projet

Welcome to Symfony!

localhost:5000

⚠ You're seeing this page because you haven't configured any homepage URL and debug mode is enabled.



Welcome to
Symfony 6.0.1

/var/www/html/cours1/monprojet/

Your application is now ready and you can start working on it.

Documentation
Guides, components, references

Tutorials
Create your first page

Community
Connect, get help, or contribute

404 965 ms 20.0 MiB 2 n/a 25 ms

7

Symfony Server 6.0.1 X

Le Dockerfile

```
1  FROM php:8-fpm-alpine3.14
2
3  # installation bash
4  RUN apk --no-cache update && apk --no-cache add bash git npm\
5      && git config --global user.email "jean-michel.couvreur@univ-orleans.fr" \
6      && git config --global user.name "Jean-Michel Couvreur"
7
8  # installation de composer
9  RUN php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
10 && php composer-setup.php --install-dir=/usr/local/bin \
11 && php -r "unlink('composer-setup.php');"
12
13 # installation de symfony
14 RUN wget https://get.symfony.com/cli/installer -O - | bash \
15     && mv /root/.symfony/bin/symfony /usr/local/bin/symfony
16
17 WORKDIR /var/www/html
18
```

Docker-compose.yml

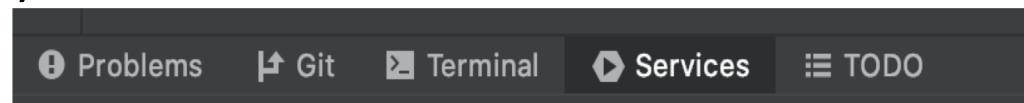
```
docker-compose.yml Dockerfile
1   version: '3'
2   services:
3     php-fpm:
4       build: .
5       container_name: my-web-container
6       volumes:
7         - ./dev:/var/www/html
8       ports:
9         - "5000:8000"
10      environment:
11        PHP_IDE_CONFIG: serverName=toto
12
```

Ouvrons le projet avec PhpStorm



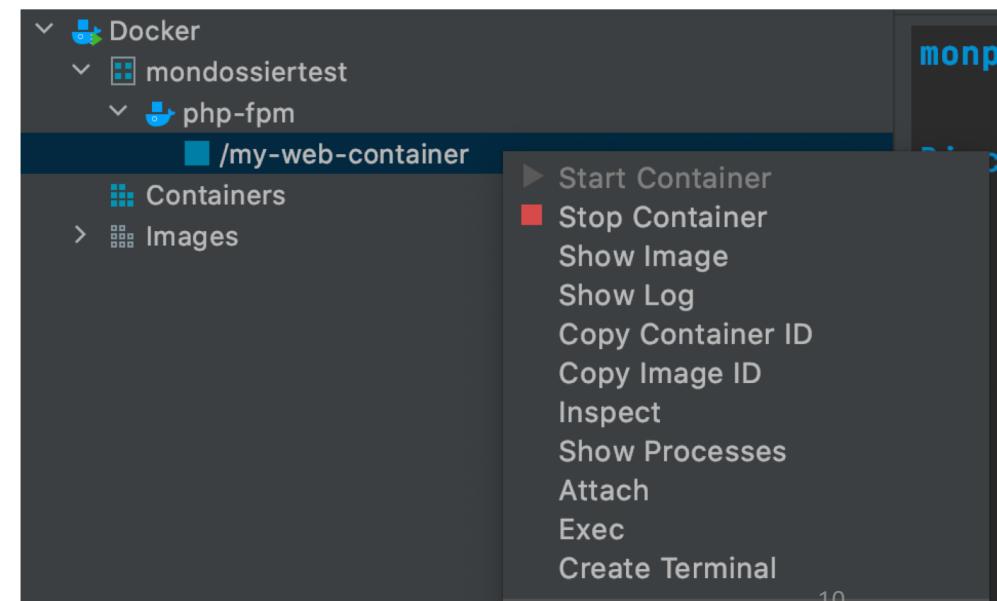
- (1) Faire File -> Open
- (2) Ouvrir le projet qui est dans le dossier dev

- (3) Installer les plugins Docker.
- (4) Sans Preferences -> Build, Execution, Deployment -> Docker : vérifier les paramètres
- (5) Lancer Services



- (6) Ouvrir un terminal

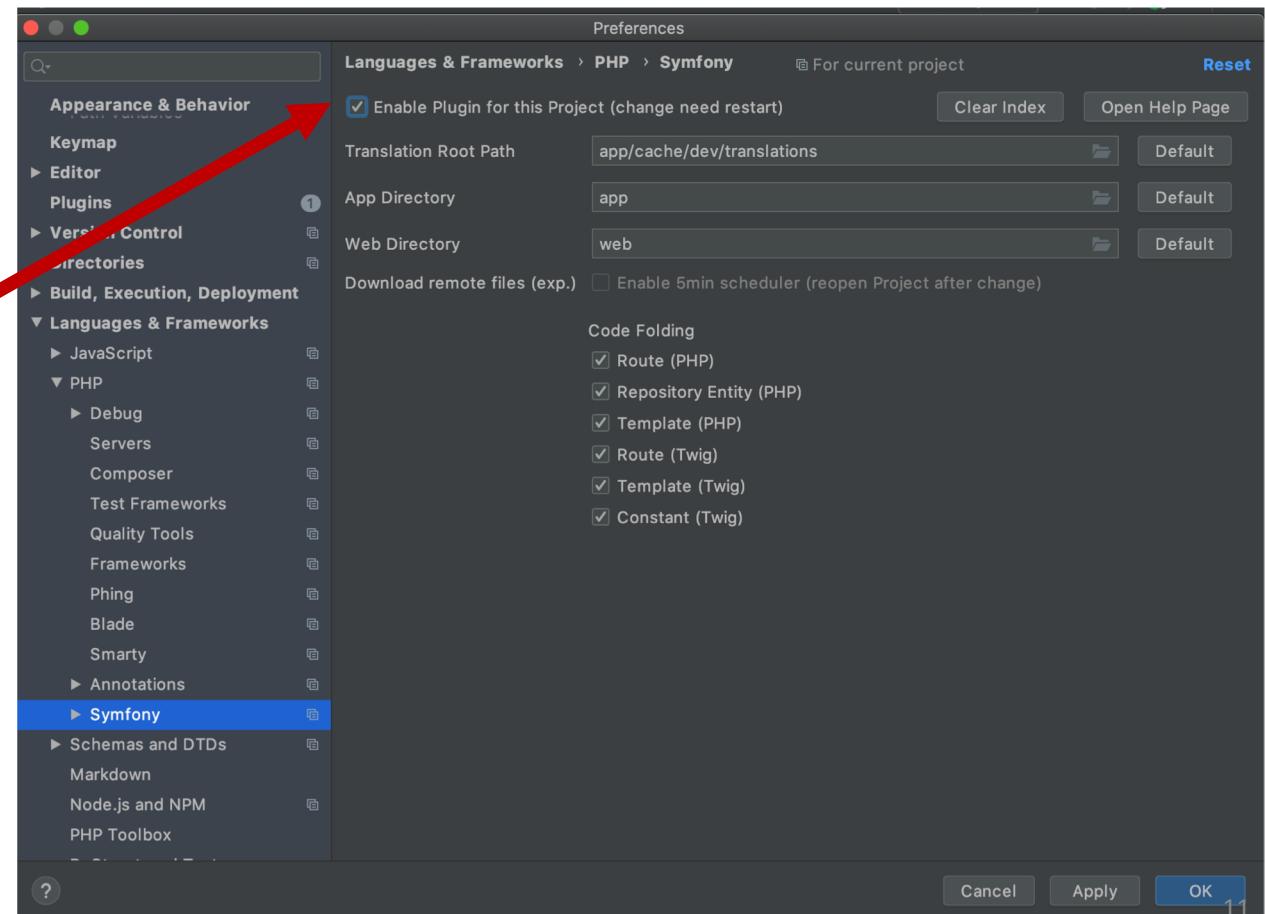
<https://www.jetbrains.com/help/phpstorm/docker.html>



Quelques plugins utiles : PhP Annotations, PhP Toolbox et Symfony Support

(1) Ouvrir le projet que nous venons de créer

(2) Activer le plugin Symfony



Tout est ok !



Terminal connecté au conteneur
docker



Un linux minimal avec PHP 8,
Composer et Symfony

Editeur sur le dossier



Un dossier partagé contenant le
code du projet

Un container est suffisant pour créer tous vos projets symfony

- Crédit d'un projet

```
symfony new --full monautreprojet
```

- Lancer un serveur local

```
cd monautreprojet
```

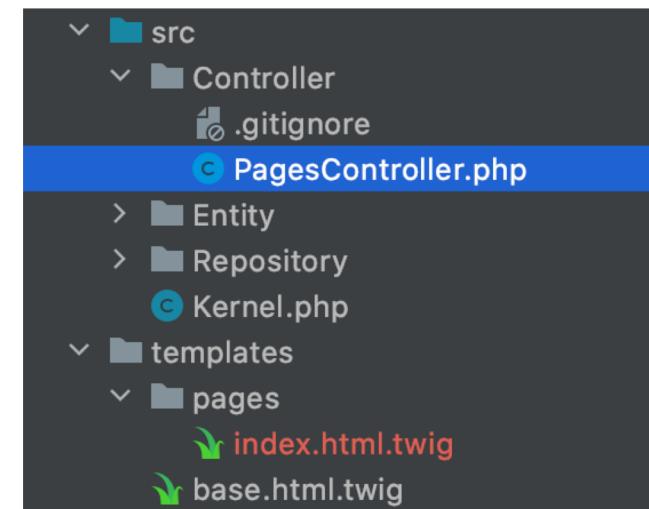
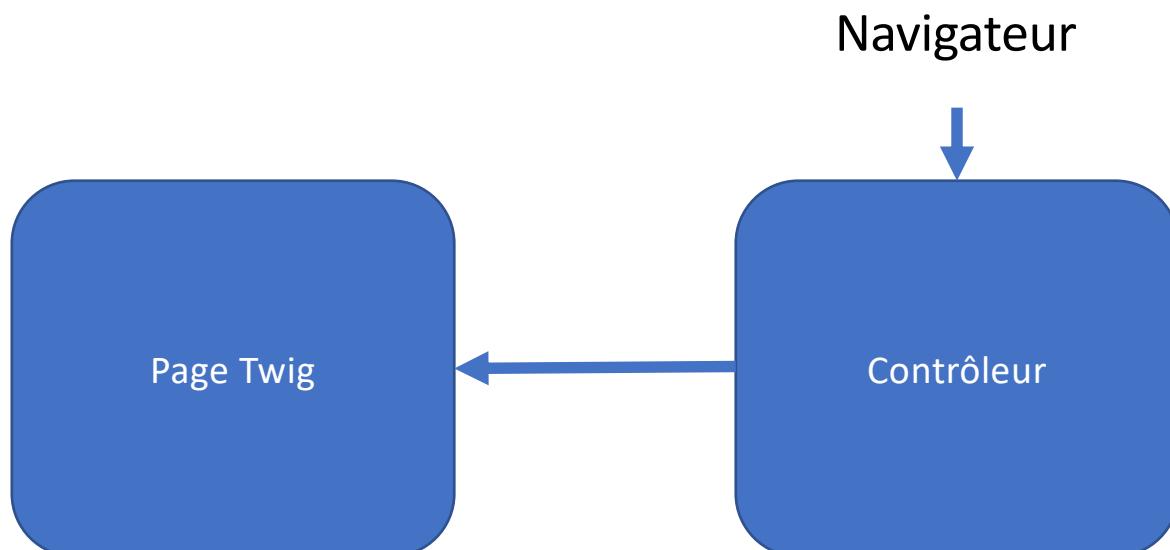
```
symfony server:start -d --no-tls
```

- Ouvrir votre navigateur

```
http://127.0.0.1:5000 ou http://localhost:5000
```

<https://symfony.com/doc/current/setup.html>

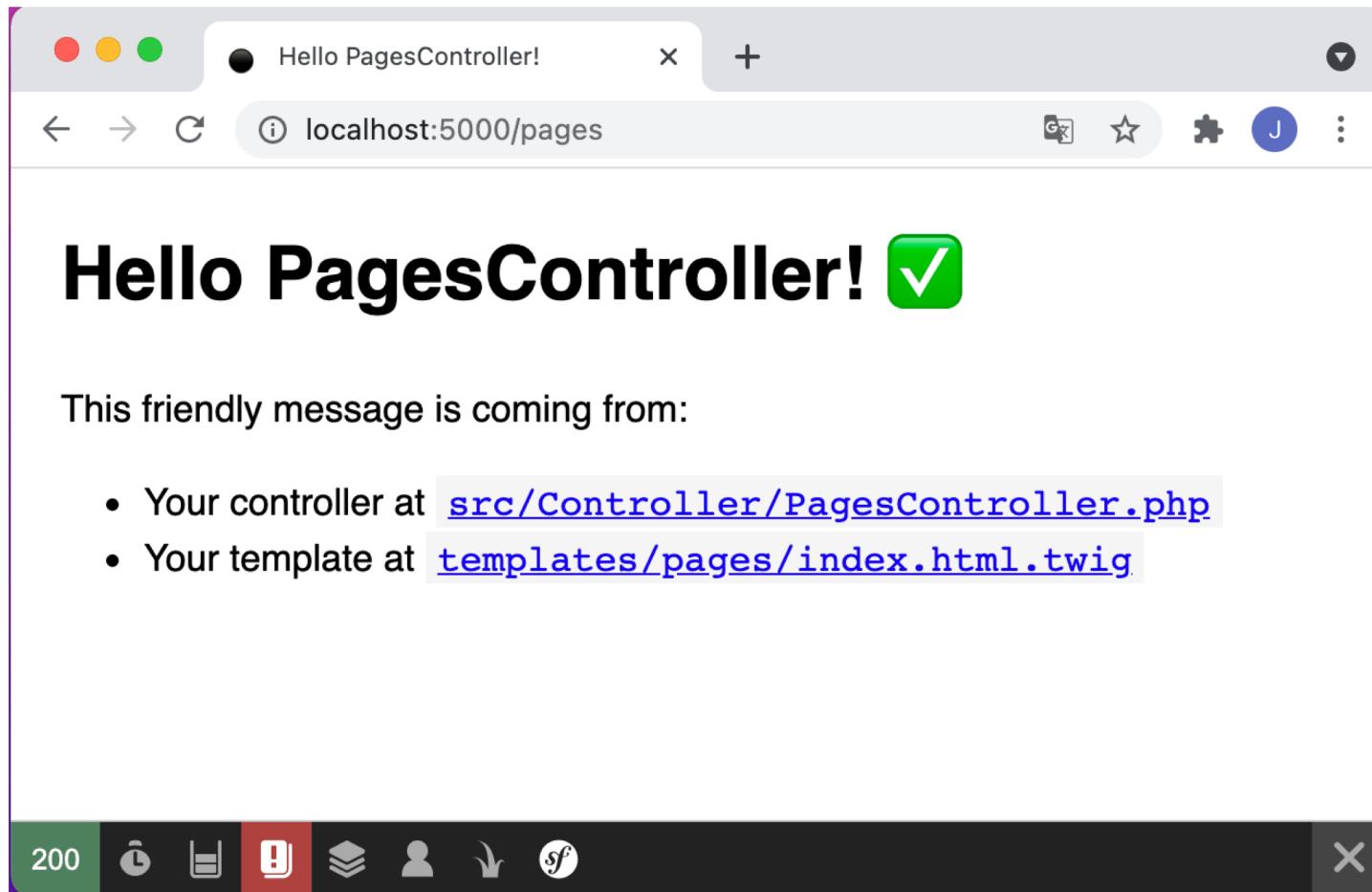
2. Mon premier programme



```
bash-5.0# symfony console make:controller pages
```

https://symfony.com/doc/current/page_creation.html

Cela donne



```
<?php

namespace App\Controller;

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

class PagesController extends AbstractController
{
    /**
     * @Route("/pages", name="pages")
     */
    public function index(): Response
    {
        return $this->render('pages/index.html.twig', [
            'controller_name' => 'PagesController',
        ]);
    }
}
```

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

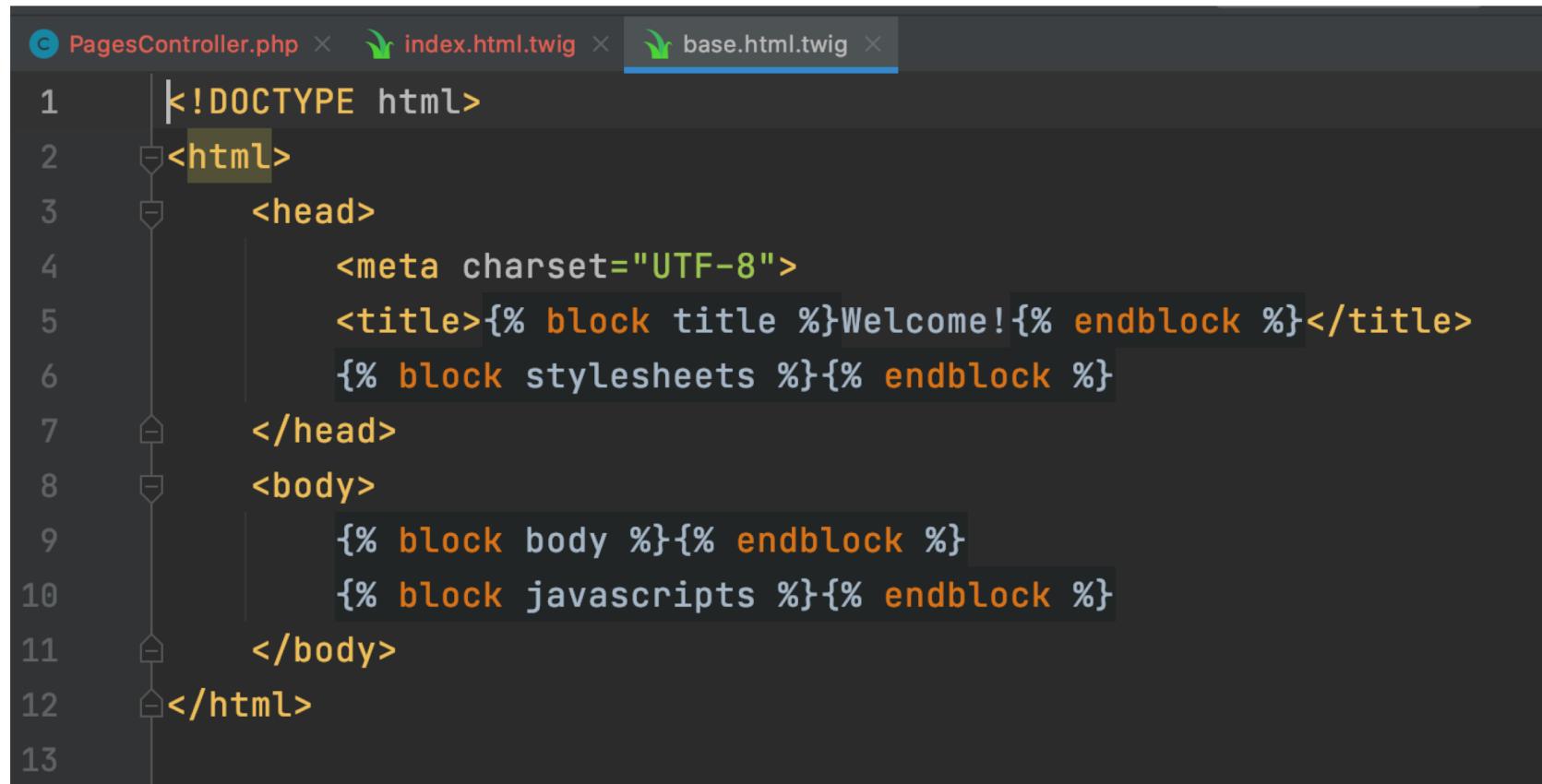
{% block title %}Hello PagesController!{% endblock %}

{% block body %}
<style>
    .example-wrapper { margin: 1em auto; max-width: 800px; width: 95%; }
    .example-wrapper code { background: #F5F5F5; padding: 2px 6px; }
</style>

<div class="example-wrapper">
    <h1>Hello {{ controller_name }}! ✓</h1>

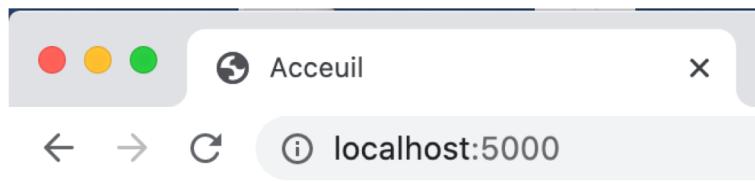
    This friendly message is coming from:
    <ul>
        <li>Your controller at <code><a href="{{ '/var/www/html/monproje" />
        <li>Your template at <code><a href="{{ '/var/www/html/monprojet/" />
    </ul>
</div>
{% endblock %}
```

Le template « base.html.twig »



```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="UTF-8">
5          <title>{% block title %}Welcome!{% endblock %}</title>
6          {% block stylesheets %}{% endblock %}
7      </head>
8      <body>
9          {% block body %}{% endblock %}
10         {% block javascripts %}{% endblock %}
11     </body>
12 </html>
```

Faisons quelques modifications



Bonjour !

A screenshot of a code editor showing a Twig template file named "index.html.twig". The code defines a title block and a body block, both containing the text "Bonjour !".

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

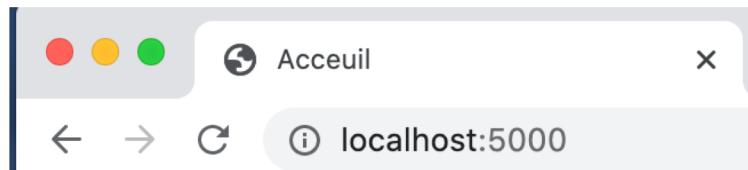
{% block title %}Acceuil{% endblock %}

{% block body %}
    <h1>Bonjour ! </h1>
{% endblock %}
```

A screenshot of a code editor showing a PHP controller file named "PagesController". It contains annotations for routes and a constructor.

```
class PagesController extends AbstractController
{
    /**
     * @Route("/pages", name="pages")
     * @Route("/", name="home")
     */
}
```

Faisons quelques modifications



Bonjour !

Il est 01:34

© Copyright 2021

A screenshot of a code editor with two tabs: "PagesController.php" and "index.html.twig". The "index.html.twig" tab is active. The code is a Twig template with the following content:

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

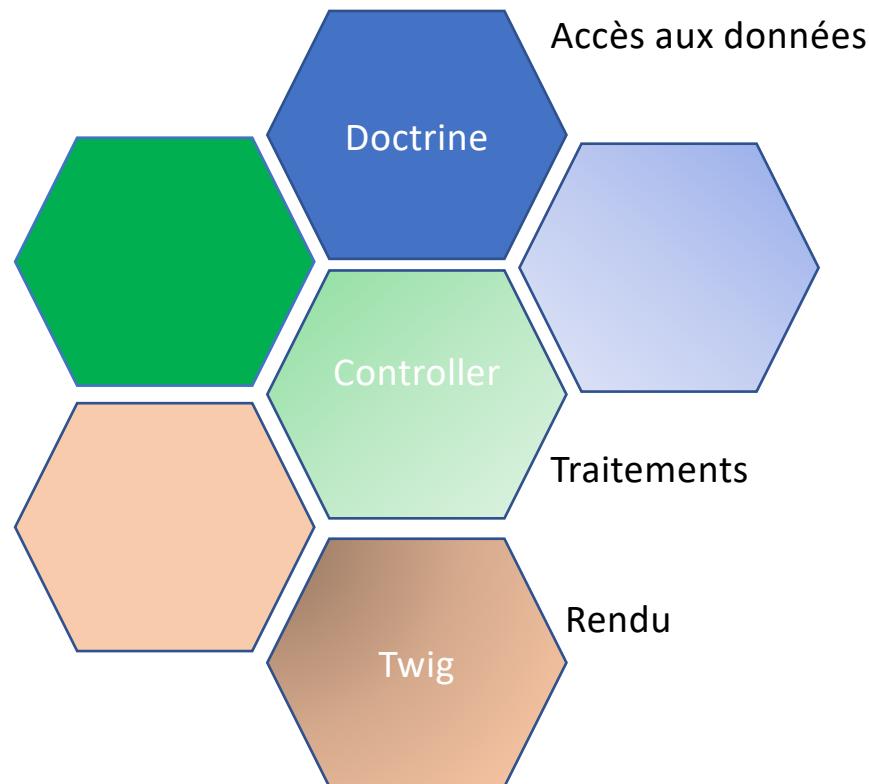
2

3  {% block title %}Acceuil{% endblock %}

4

5  {% block body %}
6      <h1>Bonjour ! </h1>
7      <p> Il est {{ 'now'|date('H:i') }}</p>
8      <footer>&copy; Copyright {{ 'now'|date('Y') }}</footer>
9
10 
```

3. Un exemple : un blog



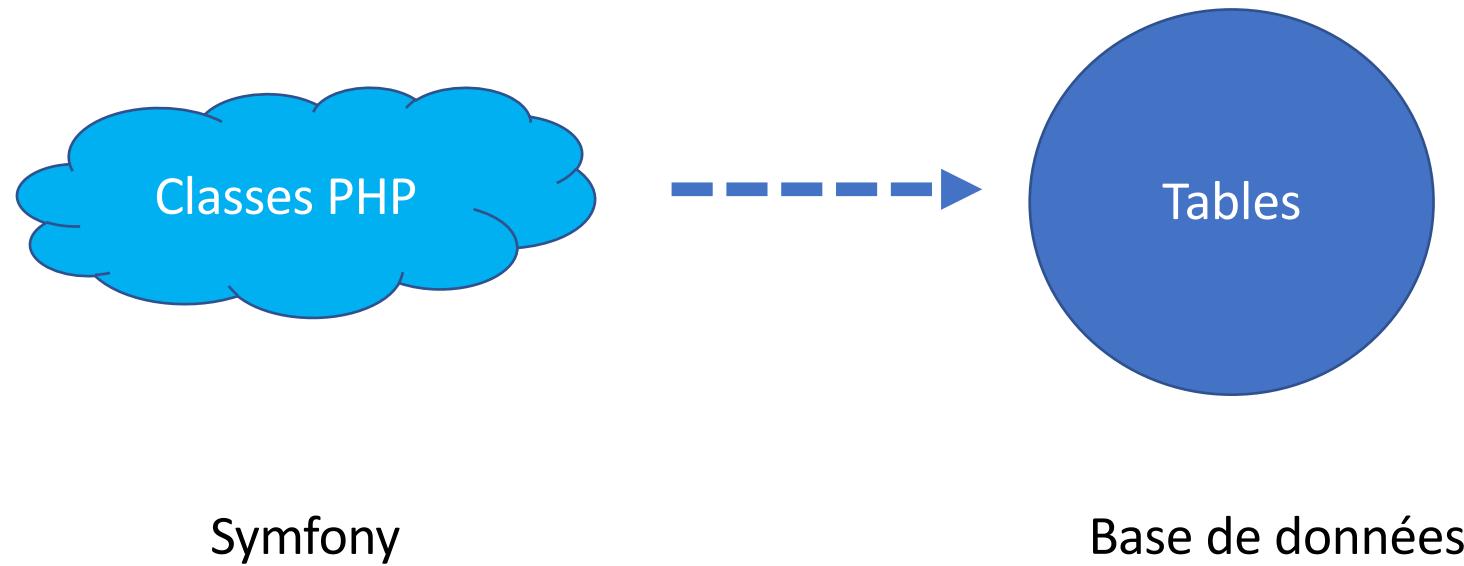
ORM Doctrine

Symfony et les bases de données

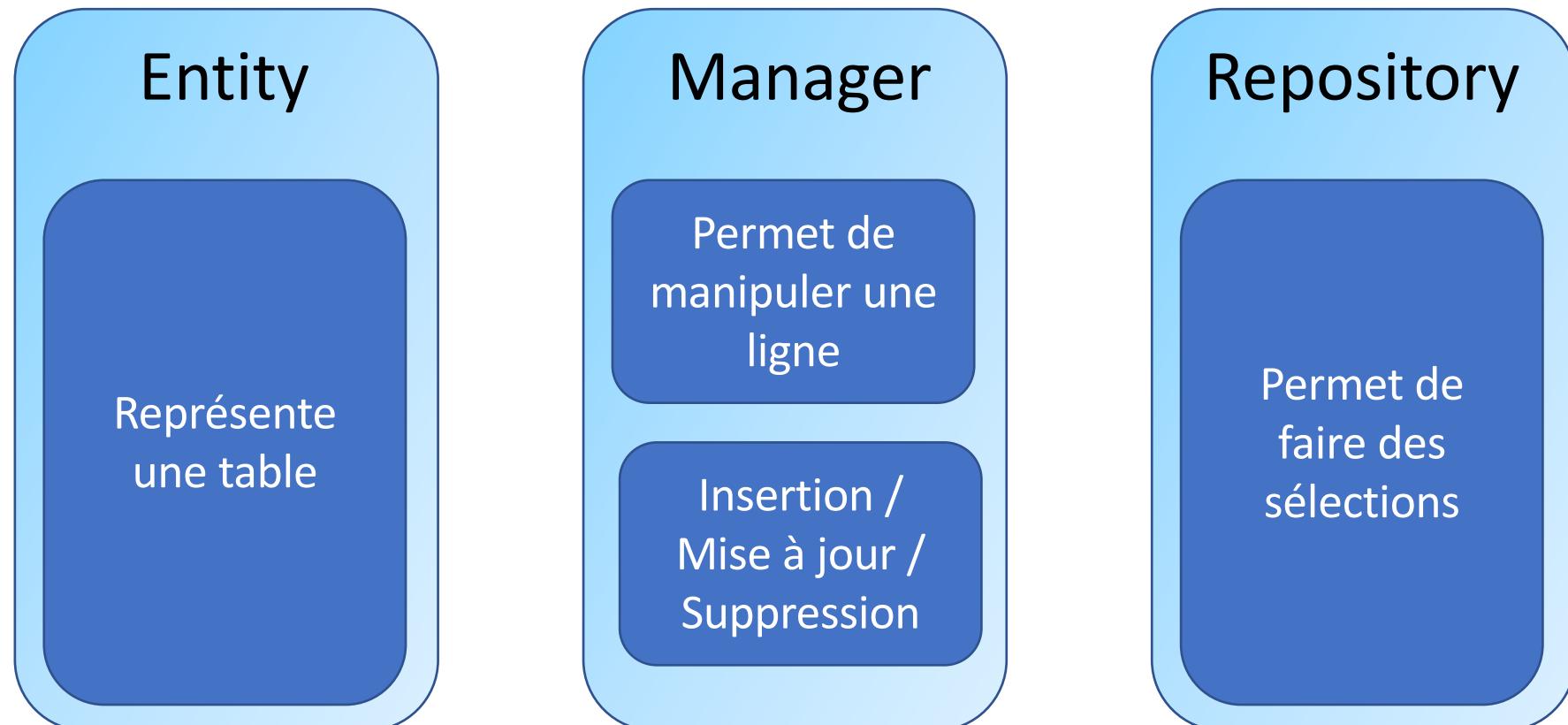
The screenshot shows a web browser displaying the Symfony Documentation website at symfony.com/doc/current/doctrine.html. The page title is "Databases and the Doctrine ORM (Symfony Docs)". The header includes the Symfony logo, a "Connect" button, and a search icon. Below the header, there's a navigation bar with links for "About", "Documentation" (which is highlighted in blue), "Screencasts", "Cloud", "Certification", "Community", "Businesses", "News", and a "DOWNLOAD" button. A banner for "Sponsored by SensioLabs" is visible. The main content area features two sections: "SymfonyCon" (Disneyland Paris, Dec. 3-4, 2020) and "SymfonyLive" (LILLE (France), February 28, 2020; PARIS (France), March 26-27, 2020; TUNIS (Tunisia), April 18, 2020; WARSZAWA (Poland), June 11-12, 2020; BERLIN (Germany), Sep. 24-25, 2020). On the left sidebar, there are links for "Getting Started" (Setup, Creating Pages, Routing, Controllers, Templates, Configuration), "Guides" (with a right arrow), "Components" (with a right arrow), "Training", "Certification", and "Master Symfony fundamentals" (Be trained by SensioLabs experts (2 to ...)). The main content on the right lists topics under "Databases and the Doctrine ORM": Installing Doctrine (Configuring the Database, Creating an Entity Class), Migrations: Creating the Database Tables/Schema (Migrations & Adding more Fields, Persisting Objects to the Database, Validating Objects, Fetching Objects from the Database, Automatically Fetching Objects (ParamConverter), Updating an Object, Deleting an Object, Querying for Objects: The Repository). A "5.0 version" dropdown and an "edit this page" button are located on the right side of the main content area.

L'ORM de Symfony : Doctrine

Object Relational Mapping



L'ORM de Symfony : Doctrine



Ensemble de classes PHP

Les migrations dans Symfony

The screenshot shows a web browser displaying the Symfony documentation for the DoctrineMigrationsBundle. The URL in the address bar is `symfony.com/doc/master/bundles/DoctrineMigrationsBundle/index.html`. The page header includes the Symfony logo, navigation links for About, Documentation (which is currently selected), Screencasts, Cloud, Certification, Community, Businesses, News, and a Download button. A sidebar on the left provides links to Getting Started (Setup, Creating Pages, Routing, Controllers, Templates, Configuration), Guides, Components, Training, and Certification. A section titled "Master Symfony fundamentals" offers training sessions. The main content area is titled "DoctrineMigrationsBundle" and explains database migrations. It includes a "2.0 version" dropdown and an "edit this page" button. Below the text, there's an "Installation" section with a terminal command: `$ composer require doctrine/doctrine-migrations-bundle "^2.0"`.

Les migrations dans Symfony

Migration #1

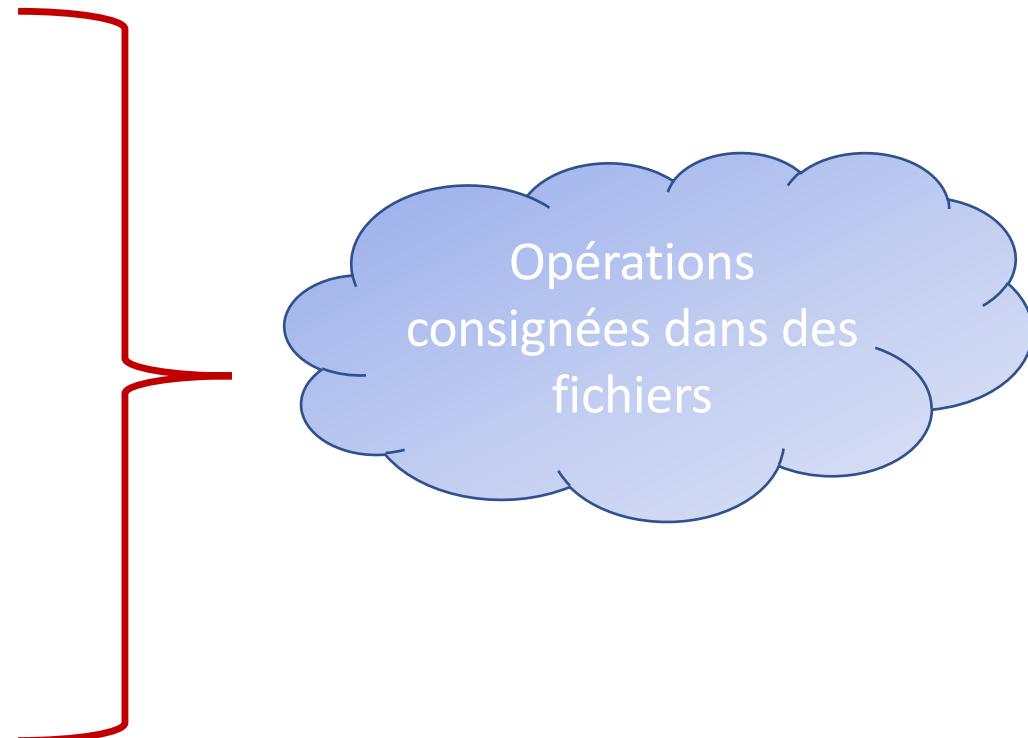
- Je crée deux tables

Migration #2

- Je crée une autre table
- J'en supprime une

Migration #3

- Je supprime un champ d'une table
- J'ajoute une relation entre deux tables

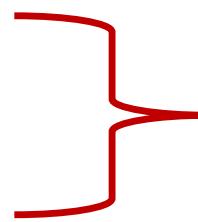


Les Fixtures

The screenshot shows a web browser displaying the Symfony Documentation page for the DoctrineMigrationsBundle. The URL in the address bar is `symfony.com/doc/master/bundles/DoctrineMigrationsBundle/index.html`. The page header includes the Symfony logo, a navigation bar with links for About, Documentation (which is currently selected), Screencasts, Cloud, Certification, Community, Businesses, News, and a Download button. A sidebar on the left provides links to Getting Started, Guides, Components, Training, and Certification. Below the sidebar is a section titled "Master Symfony fundamentals" with a link to `training.sensiolabs.com`. The main content area is titled "DoctrineMigrationsBundle" and discusses database migrations. It includes a "2.0 version" dropdown menu and a "edit this page" button. A code snippet at the bottom shows the command `$ composer require doctrine/doctrine-migrations-bundle "^2.0"`.

Les Fixtures

- Créer des jeux de fausses données



- Exécutable à souhait
- Réutilisable par les autres

Création de la base de données

(1) Modifier le fichier « .env »

```
###> doctrine/doctrine-bundle ###
# Format described at https://www.doctrine-project.org/projects/doctrine-bundle/1.x/en/configuring.html#configuration-environment-variables
# IMPORTANT: You MUST configure your server version, either here or in config/packages/doctrine.yaml
#
DATABASE_URL="sqlite:///%kernel.project_dir%/var/data.db"
# DATABASE_URL="mysql://db_user:db_password@127.0.0.1:3306/db"
# DATABASE_URL="postgresql://db_user:db_password@127.0.0.1:5432/db"
###< doctrine/doctrine-bundle ###
```

(2) Lancer la commande

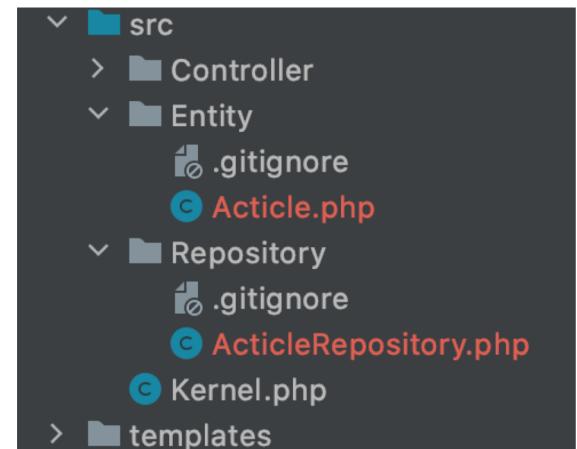
```
bash-5.0# symfony console doctrine:database:create
Created database /var/www/html/monprojet/var/data.db for connection named default
bash-5.0#
```

Création d'une entité

```
bash-5.0# symfony console make:entity Article
```

- Crédation de 2 fichiers :
 - Article.php
 - ArticleRepository

Analysons les fichiers !!!



Création de table dans la base de données

(1) Création d'un fichier migration

```
bash-5.0# symfony console make:migration
```

(2) Création de la table

```
bash-5.0# symfony console doctrine:migrations:migrate
```

Analysons les fichiers !!!

data.db 1	
▼	schemas 1
▼	main
▼	acticle
!	id INTEGER (auto increment)
!	title VARCHAR(255)
!	description CLOB
!	key #1 (id)

Création d'une « fixture »

- (1) Installer la package fixture : symfony composer require orm-fixtures --dev
- (2) Créer une fixture : symfony console make:fixture

```
bash-5.0# symfony console make:fixture

The class name of the fixtures to create (e.g. AppFixtures):
> ArticleFixtures

created: src/DataFixtures/ArticleFixtures.php
```



Ajout d'un fichiers php

Création d'une « fixture »

```
namespace App\DataFixtures;

use ...

class ArticleFixtures extends Fixture
{
    public function load(ObjectManager $manager)
    {
        // $product = new Product();
        // $manager->persist($product);

        $manager->flush();
    }
}
```



Modifier le code !

Création d'une « fixture »

```
namespace App\DataFixtures;

use Doctrine\Bundle\FixturesBundle\Fixture;
use Doctrine\Persistence\ObjectManager;
use App\Entity\Article;

class ArticleFixtures extends Fixture
{
    public function load(ObjectManager $manager)
    {
        for($i=1; $i<=10; $i++) {
            $article = new Article();
            $article->setTitle("Titre de l'article n°$i")
            ->setDescription("Contenu de l'article n°$i");
            $manager->persist($article);
        }

        $manager->flush();
    }
}
```

Création d'une « fixture »

Lancer la commande : `symfony console doctrine:fixtures:load`

	id	title	description
1	1	Titre de l'article n°1	23B of 24B obtenu de l'article n°1
2	10	Titre de l'article n°10	24B of 25B obtenu de l'article n°10
3	2	Titre de l'article n°2	23B of 24B obtenu de l'article n°2
4	3	Titre de l'article n°3	23B of 24B obtenu de l'article n°3
5	4	Titre de l'article n°4	23B of 24B obtenu de l'article n°4
6	5	Titre de l'article n°5	23B of 24B obtenu de l'article n°5
7	6	Titre de l'article n°6	23B of 24B obtenu de l'article n°6
8	7	Titre de l'article n°7	23B of 24B obtenu de l'article n°7
9	8	Titre de l'article n°8	23B of 24B obtenu de l'article n°8
10	9	Titre de l'article n°9	23B of 24B obtenu de l'article n°9

Accédons aux données dans Symfony

- (1)Modifier le contrôleur : BlogController.php
- (2)Modifier les fichiers twig

Fonction index de PagesController.php

```
class BlogController extends AbstractController
{
    /**
     * @Route("/blog", name="blog")
     * @Route("/", name="home")
     */
    public function index(ArticleRepository $repo): Response
    {
        $articles=$repo->findAll();
        return $this->render( view: 'blog/index.html.twig', [
            'articles' =>$articles
        ]);
    }
}
```

Index.html.twig

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

{% block body %}
    {% for article in articles %}
        <article>
            <h2>{{ article.title }}</h2>
            {{ article.description | raw }}
        </article>
    {% endfor %}
{% endblock %}
```

Résultat

Titre de l'article n°1

Contenu de l'article n°1

Titre de l'article n°2

Contenu de l'article n°2

Titre de l'article n°3

Contenu de l'article n°3

Titre de l'article n°4

Contenu de l'article n°4

Titre de l'article n°5

Ajout d'un lien sur chaque article

(1) Fonction show de PagesController.php

```
/**
 * @Route("/blog/show/{id}", name="blog_show")
 */
public function show(Article $article): Response
{
    return $this->render( view: 'blog/show.html.twig', [
        'article' =>$article
    ]);
}
```

Ajout d'un lien sur chaque article

(2) Fichier show.html.twig

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

{% block body %}
<article>
    <h2>{{ article.title }}</h2>
    {{ article.description | raw }}
</article>
{% endblock %}
```

index.html.twig (avec bouton revu)

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

{% block body %}
    {% for article in articles %}
        <article>
            <h2>{{ article.title }}</h2>
            {{ article.description |raw }}
            <a href="{{ path('blog_show', {id:article.id}) }}"  

                class="btn btn-primary">lien</a>
        </article>
    {% endfor %}
{% endblock %}
```

Ajout du paramètre !

Ajout d'un article : le formulaire

Version 0 du contrôleur

```
/**
 * @Route("/blog/create", name="blog_create")
 */
public function create(): Response
{
    $article = new Article();
    $form = $this->createFormBuilder($article)
        ->add('title')
        ->add('description')
        ->getForm();
    return $this->render('blog/create.html.twig', [
        'formArticle' =>$form->createView()
    ]);
}
```

Ajout d'un article : le formulaire (suite)

Version 0 du twig

```
{% extends 'base.html.twig' %} ① 1 ▲ 4

{% block body %}
    <h1> Création d'un article</h1>
    {{ form_start(formArticle) }}
    {{ form_widget(formArticle) }}
    <button type="submit" class="btn btn-success"> Enregistrer </button>
    {{ form_end(formArticle) }}
{% endblock %}
```

Version 1 du contrôleur

```
#[Route('/blog/create', name: 'blog_create')]
public function create(Request $request,
                      EntityManagerInterface $entityManager) : Response
{
    $article = new Article();

    $form = $this->createFormBuilder($article)
        ->add('title')
        ->add('description')
        ->getForm();
    $form->handleRequest($request);

    if ($form->isSubmitted() && $form->isValid()) {
        $entityManager->persist($article);
        $entityManager->flush();
        return $this->redirectToRoute(route: 'blog_show', ['id' => $article->getId()]);
    }

    return $this->render(view: 'blog/create.html.twig', [
        'formArticle' => $form->createView()
    ]);
}
```

Modification d'un article (suite)

Faire de petits ajustements dans « create.html.twig » et dans le contrôleur !!!
Penser à ajouter dans la barre de navigation « ajout »

```
#[Route('/blog/create', name: 'blog_create')]
#[Route('/blog/{id}/edit', name: 'blog_edit')]
public function create(Article $article = null, Request $request,
    EntityManagerInterface $entityManager) : Response
{
    $create = false;
    if ($article ==null) {
        $article = new Article();
        $create =true;
    }

    return $this->render( view: 'blog/create.html.twig', [
        'formArticle'=>$form->createView(),
        'create'=> $create
    ]);
}
```

Suppression d'un article (suite)

Ajouter une fonction remove dans le contrôleur et un bouton dans « show.html.twig »

```
#[Route('/blog/{id}/delete', name: 'blog_delete')]
public function delete(Article $article,
                      EntityManagerInterface $entityManager) : Response
{
    $entityManager->remove($article);
    $entityManager->flush();
    return $this->redirectToRoute(route: 'blog');
}
```

Injection de dépendance (bonus)

```
class BlogController extends AbstractController
{
    /**
     * @Route("/blog", name="blog")
     */
    public function index(ArticleRepository $repo)
    {
        $articles=$repo->findAll();
        return $this->render( view: 'blog/index.html.twig', [
            'controller_name' => 'BlogController',
            'articles' =>$articles
        ]);
    }

    /**
     * @Route("/", name="home"); ...
     */
    public function home(){...}

    /**
     * @Route("/blog/{id}", name="blog_show");
     */
    public function show(Article $article)
    {
        return $this->render( view: 'blog/show.html.twig', [
            'article'=>$article
        ]);
    }
}
```

Quoi faire !

(1) Voir la vidéo de Lior CHAMLA

<https://www.youtube.com/watch?v=UTusmVpwJXo>

Et celle qui suit !

Ou regarder « créer et déployer une application Symfony

https://youtube.com/playlist?list=PLIxQJeQRaKDTxU_kTI0Ltn4WGZFIB1xt

T

(2) Vous pouvez aussi lire

<https://symfony.com/doc/current/the-fast-track/fr/index.html>

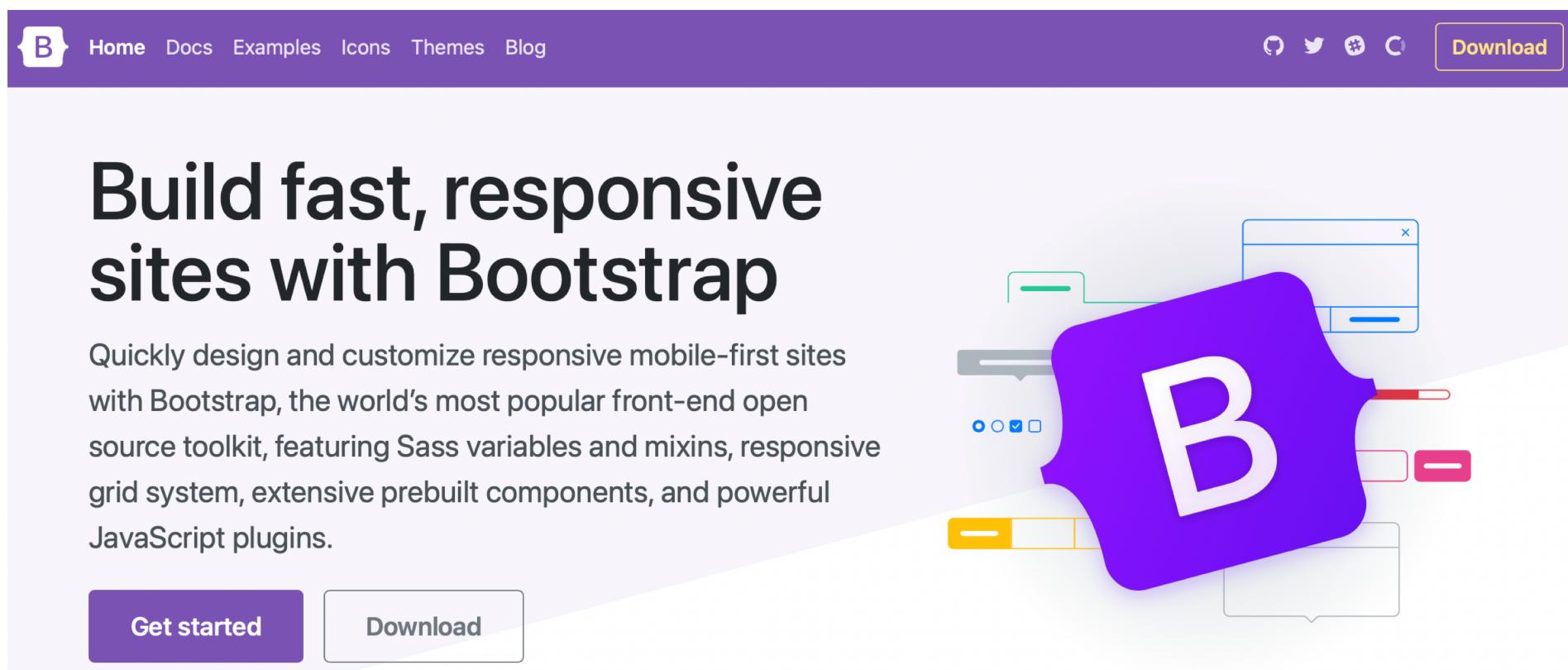
(3) Refaire les expériences du cours et s'assurer d'avoir un environnement fonctionnel

(4) Et recommencer jusqu'à être capable de le faire rapidement ...



Expérimetons avec Bootstrap

<https://getbootstrap.com>



The screenshot shows the official Bootstrap website. At the top is a purple header bar with the Bootstrap logo (a white 'B' icon), navigation links for Home, Docs, Examples, Icons, Themes, and Blog, and social media icons for GitHub, Twitter, and others. A prominent 'Download' button is also in the header. Below the header, the main title 'Build fast, responsive sites with Bootstrap' is displayed in large, bold, black font. A descriptive paragraph follows, explaining that Bootstrap is a mobile-first toolkit with a responsive grid system, Sass variables, mixins, and JavaScript plugins. At the bottom left are two buttons: 'Get started' in a purple box and 'Download' in a white box. To the right of the text is a graphic featuring a large purple hexagon with a white 'B' inside, overlaid on a background of several smaller, semi-transparent mobile device icons.

Build fast, responsive sites with Bootstrap

Quickly design and customize responsive mobile-first sites with Bootstrap, the world's most popular front-end open source toolkit, featuring Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful JavaScript plugins.

Get started Download

(1) Mise en place simple

<https://getbootstrap.com/docs/5.1/getting-started/introduction/>

Ajouter dans base.html.twig les lignes suivantes

CSS

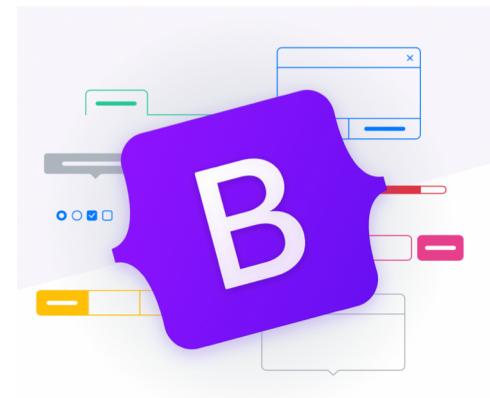
Copy-paste the stylesheet <link> into your <head> before all other stylesheets to load our CSS.

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" Copy
```

Bundle

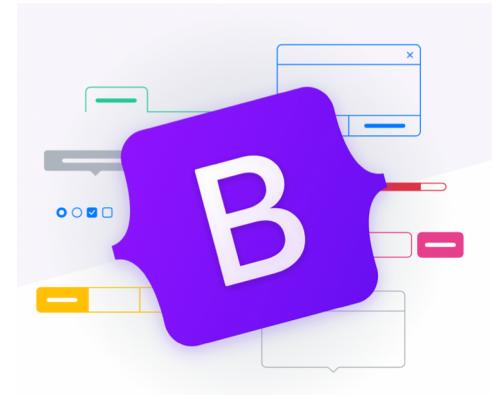
Include every Bootstrap JavaScript plugin and dependency with one of our two bundles. Both `bootstrap.bundle.js` and `bootstrap.bundle.min.js` include [Popper](#) for our tooltips and popovers. For more information about what's included in Bootstrap, please see our [contents](#) section.

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" type="text/javascript" Copy
```



(2) Avec Webpack Encore

<https://symfony.com/doc/current/frontend/encore/bootstrap.html#importing-bootstrap-styles>



Voir la documentation de Symfony !

ou bien



Bootstrap 5 avec Symfony 5 et Webpack Encore
YoanDev
12 k vues • il y a 1 an

<https://www.youtube.com/watch?v=obv0t92plQg>