

# Web 2.0

## Les étapes de création d'un projet Symfony :

- `composer create-project symfony/website-skeleton nom_de_projet "4.4.*"`

## Exécutez le projet :

- `composer require symfony/web-server-bundle`
- `php bin/console server:run`

ps : ay haja faha «require » a3rf ray bech ta3ml update fel fichier console eli fi west el dossier bin

## Génération automatique d'un contrôleur :

`php bin/console make:controller Esm_el_controller`

## Manipulation du Contrôleur et Routing :

El routing howa kifh el controleur bech ihez k lel interface mta3k eli hya tabda page TWIG exemple index.html.twig :

bech ikoun 3andk methode wala zada najmou n9ouloulha action fel controleur hya eli bech ta5ou link w ta3tik interface

Fama zouz tore9 lel routing :

1ère façon: YML : temchi lel routes.yaml : ta3tih el path ya3ni el lien eli bech tektbtou ba3ed 127.0.0.1 :8000

W chnia l'action eli bech test3mlha fel controleur

## Exemple :

Routes.yaml :

```
blog_show:
  path:          /blog
  controller:    App\Controller\BlogController::show
```

BlogController.php :

```
public function show()  
{  
    //  
}
```

## 2-temchi lel controuleur fou9 l'action mta3k tektb :

```
/**  
 * @Route("/blog", name="blog_show")  
 */
```

**Donc @route /blog howa el lien eli bech tktbou ba3ed 127.0.0.1 :8000 exemple : 127.0.0.1 :8000/blog w el show hya l'action eli bech tsir ki bech tktb el lien a4aka**

## TWIG :

### Qu'est ce qu'un moteur de templates :

**PHP peut être considéré comme un moteur de template**

- Il est possible de mélanger du PHP avec du code HTML mais il reste très verbeux et peu pratique pour certaines tâches.

#### **Pourquoi utiliser un moteur de templates ?**

- Séparer le traitement de l'affichage
- Permettre aux designers de développer rapidement des gabarits sans spécialement connaître le langage utilisé
- Minimiser le code et le rendre plus clair

#### **Les avantages de Twig:**

- Permet de séparer la présentation des données du traitement.
- Permet la personnalisation de la page web.
- Permet de rendre les pages web plus lisibles, plus claires.
- Apporte de nouvelles fonctionnalités (comme l'héritage des templates, les filtres...).
- Apporte plus de sécurité.

## Retourner un Template Twig :

Depuis le contrôleur, on utilise la méthode `render()` pour retourner une interface

- la méthode `render()` prend en paramètre:

1. Le chemin vers le template:

/Template/vue

2. Un tableau des paramètres à afficher dans le twig

**Exemple :**

```
public function index(): Response
{
    // ...

    // the `render()` method returns a `Response` object with the
    // contents created by the template
    return $this->render('product/index.html.twig', [
        'esm_el_variable' => 'valeur_el_variable',
    ]);
}
```

**Emplacement des templates :**

3la kol controleur(bel commande automatique ) tas3nou ya3ml dossier b esm el controleur a4aka Ta7tou les templates mta3ou par default yabda fih index.html.twig fi kol dossier mta3 kol controleur

## La Syntaxe de base :

### Trois types de syntaxes :

- Syntaxe de base pour afficher des variables
  - `{{ ... }}` affiche quelque chose
- Syntaxe de base pour les structures de contrôle et les expressions
  - `{% ... %}` exécute une action
- Syntaxe pour les commentaires
  - `{# ... #}` définit un commentaire

`{{ taffichi baha variables /formulaire ect .. }}`

`{% thot faha ay opération wala boucle ect .. %}`

### Déclaration d'une variable :

`{% set pi = 3.14 %}`

## **Concaténation :**

```
{{ var1 ~ var2 }}
```

## **La structure conditionnelle: {% if ... %} ... {% endif %}**

- Une condition avec empty

```
{% if produits is empty %}
```

il n'y a plus de produit

```
{% endif %}
```

- Une condition avec and, or ,defined

```
{% if ((a==1 and b>0) or not c==0) and d is defined %}
```

```
{% set resultat = (d + a * b) / c %}
```

```
{{ resultat }}
```

```
{% endif %}
```

- Une condition avec start , ends

```
{% if 'Fabien' starts with 'F' %}
```

commence par F

```
{% endif %}
```

## **Une condition avec matches: permet de déterminer si une variable**

### **respecte un motif donné par une expression régulière**

```
{% if phone matches '/^[\\d\\.]+$/ ' %}
```

format telephone ok

```
{% endif %}
```

- **Une condition avec not in**

```
{% if 5 not in [1, 2, 3] %}
```

5 non présent

```
{% endif %}
```

- **Une condition avec else if**

```
{% if var is odd %} ou
```

```
yes
```

```
{%else%}
```

```
no
```

```
{% endif %}
```

```
{{ var is odd? 'yes': 'no' }}
```

## **La structure itérative: for**

Parcourir un tableau associatif

```
{% for produit in produits %}
```

```
{{ produit.nom }}
```

```
{% endfor %}
```

- Parcourir un tableau indexé: (on peut utiliser l'opérateur ' .. pour définir un intervalle )

```
{% for i in 0..9 %} // pareil que {% for i in range(0, 9) %}
```

```
{{ i }}
```

```
{% endfor %}
```

Parcourir un tableau associatif avec une condition:

```
{% for produit in produits if produit.etat==1 %}
```

```
{{ produit.nom }}
```

```
{% endfor %}
```

Parcourir un tableau associatif avec une condition vide:

```
{% for article in articles %}
```

```
{{ article.nom }}  
{% else %} pas d'article trouvé  
{% endfor %}
```

clés et valeurs:

```
{% for key, value in table %}  
{{ key }} {{ value }}  
{% endfor %} //table 2d
```

### **Les filtres :**

Permettant de formater et modifier l'affichage d'une donnée´

- Pouvant prendre un ou plusieurs paramètres `
- Syntaxe : {{ variable | fonction filtre[paramètres] }}
- On peut appliquer des filtres sur une variable à afficher, sur une variable d'une

condition IF ou d'une boucle FOR...

filtre nest3mlohom :

upper : Affiche la variable en majuscules

lower : Affiche la variable en minuscules

trim : Supprime les caractères spéciaux indiqués du début et de la fin d'une chaîne de caractères

```
{{"hello world ."}|trim('.')}} =>hello world,
```

Slice(start,length) ; Extrait les éléments de la position start et de nombre length

lenght calcule le nombre d'élèments d'un tableau ou le nombre de caractères d'une chaîne

## Les liens :

Path() et url ()

o Elles permettent de référencer une route.

- Path() génère une URL relative.
- url() génère une URL absolue.

Exemple :

```
<a href="{{ path('produit_route') }}">Produit</a>
```

```
<a href="{{ url('produit_route') }}">Produit</a>
```

o On peut également définir une route paramétrée

- ```
<a href="{{ path('produit_route',{'id': 'value'}) }}">Produit détails</a>
```

Ajout de fichier

Asset(): Permet d'appeler les fichiers ressources css, js, images définis dans le dossier public

exemple :

- CSS

```
<link rel="stylesheet" href="{{ asset('css/style.css') }}" type =  
"text/css">
```

- JS

```
<script src="{{ asset('js/script.js') }}"></script>
```

- Image

```

```

## Doctrine :

### Configuration de la base de données

Configurer la base de données de l'application  
dans le fichier .env

fi west .env fama star lazmk tbadlou :

#

**DATABASE\_URL="mysql://db\_user:db\_password@127.0.0.1:num\_  
port/esm\_el\_base?serverVersion=5.7"**

Ba3ed mat3ml el config fel fichier .env tnajm test3ml el commande ha4i  
bech tasn3 el base mat3k w hya test3ml les donnés eli hatithom enti fel  
.env bech tnjm tas3nha :

**php bin/console doctrine:database:create**

### Les entités :

L'entités hya rahi image mta3 classe w mta3 table fel base  
fard wa9t

Ajouter une entité en lançant la commande suivante :

→ **php bin/console make:entity esm\_entité**

Ajouter les attributs et les paramètres

Configuration de l'entité:

Ha4i el configuration eli bech na3mlouha fel entité bech ki  
nhezouha lel base bech tab9ha :

**@ORM\Id():** spécifie la clé primaire

- **@ORM\GeneratedValue():** auto-incrémentée l'ID
- **@ORM\Column:** s'applique sur un attribut



## Exemple :

```
/**
 * @ORM\Id()
 * @ORM\GeneratedValue()
 * @ORM\Column(type="integer")
 */
private $id;
```

**La Migration : el migration tsir bech ntab9 el hajat eli tkbthom fel les fichiers mta3 les entités fel base mta3i el migration tsir 3la zouz mara7l**

### **1. php bin/console make:migration**

tans3 fichier thot fih les commandes SQL el lazmin bech tkoun el base mta3k kifha kima les entités eli hatithom ya3ni 3la kol entité ta3mmlk table w kan 3andk des relations zada thothom bech yabda fi west el fichier deux fonctions wa7da esmha up() w wahda down()

el up() ta3ml l creation down() tafas5 eli 3amltou el up()

### **2. php bin/console doctrine:migrations:migrate**

ha4i tji ba3d el make :migration w bech tmchi t5adm eli maktoub fel up() w wa9tha tetsama 3malt migration

## Des commandes o5rin mta3 migration fahom akther options :

|                                                   |                                                                                       |
|---------------------------------------------------|---------------------------------------------------------------------------------------|
| doctrine:migrations:current                       | Afficher la version actuelle                                                          |
| doctrine:migrations:execute version               | Exécuter une seule version de migration                                               |
| doctrine:migrations:generate                      | Créer un fichier de migration vide                                                    |
| doctrine:migrations:latest                        | Afficher la dernière version migration                                                |
| doctrine:migrations:migrate                       | Exécuter plusieurs version de migrations non exécutées                                |
| doctrine:migrations:status                        | Afficher l'état d'un ensemble de migrations                                           |
| doctrine:migrations:up-to-date                    | Nous indiquer si le schéma est à jour                                                 |
| doctrine:migrations:version version -- add/delete | Ajouter ou supprimer manuellement les versions de migration de la table des versions. |
| doctrine:migrations:sync-metadata-storage         | Garantit que le stockage des métadonnées est à la dernière version.                   |
| doctrine:migrations:list                          | Afficher la liste de toutes les migrations disponibles et leur état                   |

## Entity Manager :

EM est un gestionnaire d'entités.

3andk hajtin fel EM :

→ `$em= $this->getDoctrine()->getManager()` ; ha4i bech t5dm baha l'ajout el modifier el supprimer

`$em->persist($object)` ajouter

`$em->flush()` mise a jour

`$em->remove(object)` supprimer

➔ Benesba lel find/findAll/findOneBy el kol tal9ahom fel

Repository eli howa fichier yetsn3 ki tasn3 l'entité donc bech ta3ml recherche w lazmk test3ml

`$em=$this->getDoctrine()`

`->getRepository(esmL'entité::class)`

`$em->find(thot l clé primaire )` , ect..

**Les relations entre les entités / Les Formulaires lazmk tchouf el ppt mafahom maytla5s**

## DQL

**Kl Ta3ml entity el commande wa7dha tasn3lk el** Repository eli howa fih les fonctions mta3 l recherche par défaut ama a7na sa3t 7achta b hajt m3a9da akther meli mawjoudin donc nwaliw nektbou codes wa7dna fel fichier Repository ha4aka

«

Il ne peut faire que des SELECT, UPDATE, DELETE

l'insertion se fait par la persistance des entités.

»

Exemple :

-CreateQuery tektb faha query bel SQL w ta3taha el lien mta3 el Entity mta3k eli howa dima App\Entity\esml'entity

Fi west el Repository

```
public function findAllGreaterThanPrice(int $price): array
{
    $entityManager = $this->getEntityManager();

    $query = $entityManager->createQuery(
        'SELECT p
        FROM App\Entity\Product p
        WHERE p.price > :price
        ORDER BY p.price ASC'
    )->setParameter('price', $price);

    // returns an array of Product objects
    return $query->getResult();
}
```

Fi west el controller :

```
// from inside a controller
$minPrice = 1000;

$products = $this->getDoctrine()
    ->getRepository(Product::class)
    ->findAllGreaterThanPrice($minPrice);

// ..
```

Wala fama methode o5ra zada kifkif fi west Repository ama test3ml createQueryBuilder :

```
public function showAllClubByTitle(){  
    return $this->createQueryBuilder( alias: 'b')  
        ->where( predicates: 'b.name LIKE :name')  
        ->setParameter( key: 'name', value: '%T')  
        ->getQuery()  
        ->getResult()  
    ;  
}
```

w t3aytlha kifkif fi west el controller kima lo5ra