**Symfony**

Pour créer un dossier avec la structure Symfony

* Composer create-project symfony/skeleton my-project (preciser version si besoin)

OU si on souhaite avoir une version spécifique

Exemple : composer create-project symfony/skeleton:4.4.\* DB\_Symfony

* composer create-project symfony/skeleton Projet\_Symfony "4.4.\*"
* composer require server –dev (création du serveur)
* php bin/console server:run

Composer install => permet de reprendre les librairies.

**packagist** = Site internet ou se trouve des dependances utiles (exemple carbon qui est utilisé pour gérer les dates).

Symfony/flex, cherche des librairies et les pré configure.

**Installation des annotations pour les routes**

* composer require annotations

**Installation du debugger**

C’est pour le web profiler

* composer require debug

- php bin/console debug-router (+ nom route, cela nous permet d’avoir des infos precises

**Installation TWIG**

* composer require twig ("twig/twig:^2.0" si version precise)

Twig peut générer n’importe quel format avec du texte.

**Installation flex :**

* composer require asset

Permet d’avoir la commande et de faire le lien plus simple d’utilisation pour twig.  
Plus propre pour les images

Exemple dans twig :

<img src="{{ asset('images/smashbros.jpg') }}" alt="" />

**Installation var dumper** :  
Permet d’utiliser (dump) ou encore (print\_r)

* composer require var-dumper

**Doctrine installation** : Est un composant, indépendant de Symfony et nous permettra de travailler avec les bases de données. Il va nous permettre de communiquer et travailler avec les bases de données sans ouvrir php myadmin.

* php bin/console list doctrine (pour avoir toutes les commandes)

composer require symfony/orm-pack

composer require --dev symfony/maker-bundle

Les ORM, permet de faire l’abstraction des BD, fait un mapping des bases de données. Une surcouche, qui permet de nous faciliter la vie avec les requetes sur les bases de données.

**Créer base de donnée en ligne de commande**  
Ne pas oublier de modifier le fichier **ENV** et mettre root (user) et root(mdp) ainsi que le nom de la bd(db name)

Lancer MAMP avant de créer la DB

* php bin/console doctrine:database:create (lancer mamp avant !!)
* php bin/console make:entity Annonces (creation table Annonces. Attention des mots clés existent déjà : comme User, group.Peut également servir a ajouter une colonne si on rappelle une table déjà créer)
* php bin/console make:migration (a utiliser si on veut modifier donnée ex : varchar 100 a 110)
* php bin/console doctrine:migrations:migrate (pour modifier aussi avec l’autre commande)
* php bin/console make:entity --regenerate (permet de mettre a jour les données modifier exemple : ajout d’une colonne en plus. Ne pas oublier de faire les deux commande de migration pour mettre à jour !)

Si jamais on se perd dans les commandes on peut supprimer les fichiers migrations puis ensuite refaire un regenerate.

Des qu’on utilise EntityManager :

use Doctrine\ORM\EntityManagerInterface;

Pour vérifier les données dans notre table :  
- php bin/console doctrine:query:sql 'Select \* from Users'