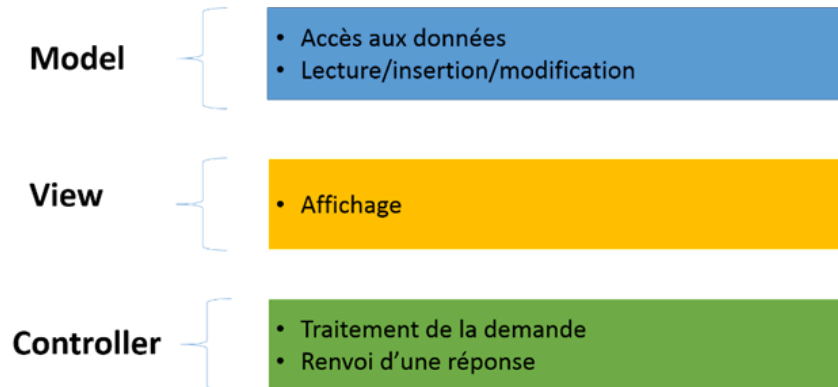


## TP 2 – Symfony : Le concept MVC.

### I. Le MVC

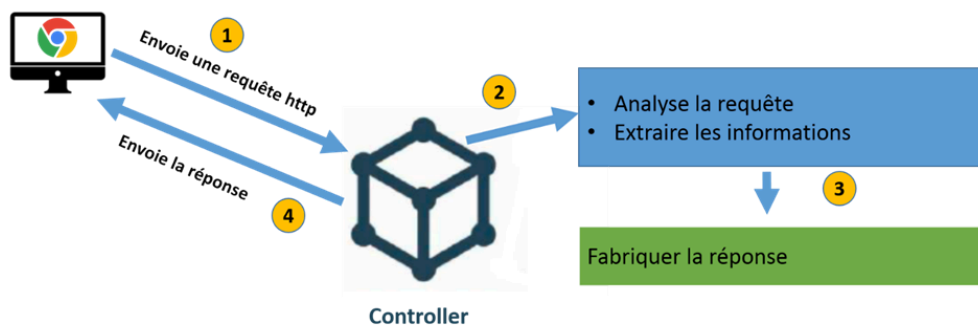
Le design pattern MVC utilisé par Symfony consiste à diviser notre application en 3 couches :



On va s'attarder dans cet atelier sur le concept de **contrôleur**, vu qu'il représente le cœur d'une application symfony.

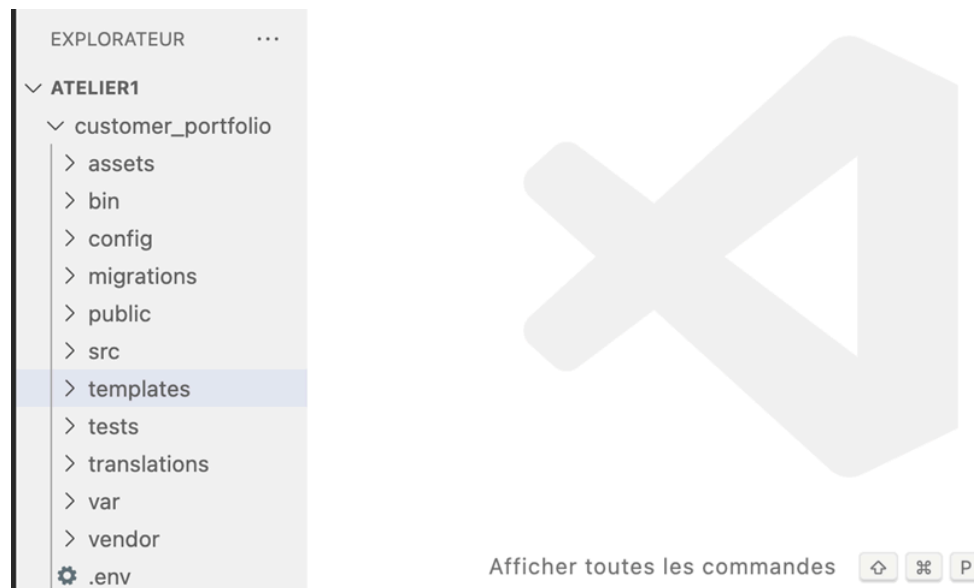
#### 1. Le concept du Controller

Le rôle du Controller va être de nous retourner une réponse par rapport à une route ou une requête donnée.



### II. Arborescence des fichiers d'une application symfony

En examinant attentivement le contenu du projet cutomer\_portfolio. Les dossiers et sous-dossiers sont organisés de sorte à ne pas mélanger les choses qui n'ont rien à voir ensemble.



**bin** : Contient les commandes php dont nous allons nous servir pendant le développement : c'est les commande qu'on lance depuis le terminal ex : php bin/console ...

**config** : Contient les fichier de configuration des routes, et les librairie installer avec l'application comme doctrine,swiftmailer,...

**public** : ce répertoire contient les fichiers destinés aux visiteurs

**src** : Ce dossier doit contenir le code source MVC de l'application, il est composé de sous dossiers: Controller ,entity, migration er repository

**templates** : les fichiers de twig

**tests** : contient les fichiers test unitaire

**translation** : les fichiers de traduction pour une application multi-langues

**var** : les caches et les logs

**vendor** : librairies installées

## Travail 1:

### 1. Créer un controller.

**php bin/console make:controller Customer**

**Le controller est créé, on peut le vérifier dans l'éditeur :**

- Création d'un nouveau fichier CustomerController dans src/Controller
- Création d'un nouveau dossier dans Template nommé customer

Examinons notre Controller CustomerController, au sein de cette classe les trois piliers permettant de construire une page sont présents.

```

class CustomerController extends AbstractController
{
    #[Route('/customer', name: 'app_customer')]
    public function index(): Response
    {
        return $this->render('customer/index.html.twig', [
            'controller_name' => 'CustomerController',
        ]);
    }
}
    
```

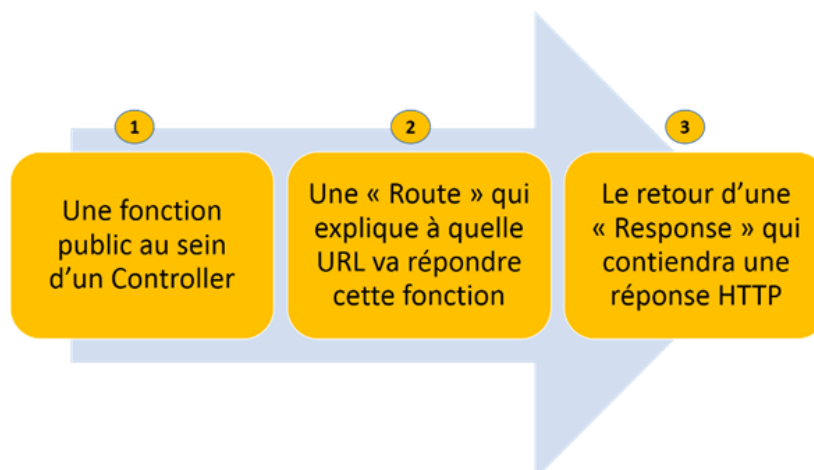
Le 2<sup>ème</sup> pilier

Le 1<sup>er</sup> pilier

Le 3<sup>ème</sup> pilier

### 1. Les piliers de création d'une page symfony

Pour afficher un rendement sur une page il nous faut les 3 piliers encadrés dans la figure précédente :



### Travail 2:

1. Revenez au fichier CustomerController.php

2. Modifier la ligne 15 par la ligne suivante :

```
return $this->render('customer/index.html.twig', ['prenom' => 'Yassmina',]);
```

3. Supprimer le contenu du fichier templates >customer >index.html.twig, et remplacer son contenu par l'instruction suivante :

```
<h1>hello {{ prenom }}</h1>
```

4. Tester votre travail

- Lancer le serveur en tapant la commande : **symfony server :start**
- Accéder à l'url puis ajouter la route définie dans CustomerController.php