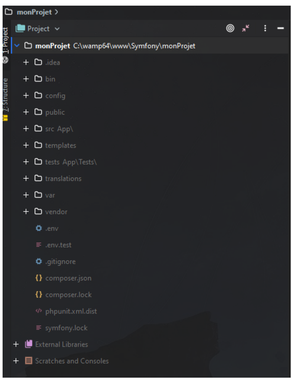
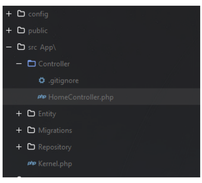
Création du contrôleur

Reprenons l'arborescence de notre projet :



Dans le dossier App/Controller, nous allons créer un nouveau fichier appelé 'HomeController.php' :



La convention de nommage veut que le nom du contrôleur commence par une majuscule.

À l'intérieur, nous allons commencer par indiquer le namespace :

```
<?php
namespace App\Controller;
?>
```

Le namespace indique l'endroit dans l'application Symfony où nous nous trouvons. Ici, nous sommes dans le namespace 'Controller'.

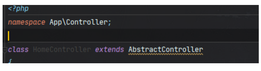
Nous allons ensuite pouvoir déclarer notre première classe :

```
<?php
namespace App\Controller;
class HomeController {

}
?>
```

Le nom de la classe est identique à celui du contrôleur. Cette classe, nous allons l'étendre à 'AbstractController' ce qui va nous permettre d'utiliser directement ses services sans rien faire de plus. On dit que 'HomeController' hérite de 'AbstractController' :

```
<?php
namespace App\Controller;
class HomeController extends AbstractController {

}
?>
```

Si votre IDE prend en charge un plugin Symfony, Vous devez avoir ce genre d'affichage :



L'IDE indique une erreur, car il ne reconnait pas AbstractController. Nous devons importer cette classe :

```
<?php
namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;

class HomeController extends AbstractController {

}
?>
```

Ensuite, nous allons indiquer le chemin vers cette classe, **la route**. Pour cela, nous allons utiliser les annotations.

Les annotations s'apparentent aux commentaires, mais contiennent des informations utiles au fonctionnement de notre application Symfony :

```
<?php
namespace App\Controller;

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

/**
 * @Route("/")
 * @method render(string $string, array $array)
 */
class HomeController extends AbstractController {

}
?>
```

On notera l'import du composant 'Annotation' de Symfony pour le fonctionnement des annotations.

Les annotations commencent par '@'. Pour déclarer une route sur notre classe, nous utilisons :

```
@Route("/")
```

Ce qui veut dire que notre contrôleur (classe) se trouve à la racine de notre projet Symfony.

'@method render ()' indique que cette classe va générer des vues.

La base de notre classe étant posée, nous allons pouvoir construire notre première méthode qui affichera notre vue.

On va déclarer une fonction en la rendant public, c'est-à-dire qu'elle sera accessible de n'importe où :

```
<?php
namespace App\Controller;

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

/**
 * @Route("/")
 * @method render(string $string, array $array)
 */
class HomeController extends AbstractController {

    /**
    * @Route("/", name="home")
    */
    pubic function index() :Response
    {

    }
}
?>
```

Nous lui avons directement indiquer une route pour la retrouver facilement :

- '/' indique que cette méthode est à la racine du contrôleur
- 'name="accueil' est le nom que l'on donne à la route. C'est ce qui permettra de cibler la méthode lorsqu'on en aura besoin.

Maintenant il faut donner une action à notre fonction, on va lui demander d'afficher un template (vue) :

```
<?php
namespace App\Controller;

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

/**
 * @Route("/")
 * @method render(string $string, array $array)
 */
class HomeController extends AbstractController {

    /**
    * @Route("/", name="home")
    */
    pubic function index() :Response
  {
   // affichage de la page d'accueil
        return $this->render('home/index.html.twig');
  }
}
?>
```

Nous pouvons maintenant voir comme nous allons construire notre vue dan la section suivante.