Comment injecter un service dans Symfony ?

A noter qu'un service peut être appelé dans plusieurs controllers, son but est de centraliser et réutiliser une méthode, une tâche.

Afin d'injecter un service il faut tout d'abord créer ou avoir à disposition le Controller ou les Controllers nécessaire ainsi qu' identifier le besoin. Ici nous allons essayer un service de calculs :
La commande pour créer un Controller : php bin/console make:controller

```
lea@mac ProjetDev % php bin/console make:controller

 Choose a name for your controller class (e.g. GentlePuppyController):
 >
```

Dans ce cas, on peut donner le nom de "PrixController"

```
lea@mac ProjetDev % php bin/console make:controller

 Choose a name for your controller class (e.g. GentlePuppyController):
 > Prix

 Do you want to generate PHPUnit tests? [Experimental] (yes/no) [no]:
 >
```

Les tests ne sont pas obligatoires ("no" par défaut si l'on clique sur Entrée)

Un Controller est créé avec sa vue "Twig":

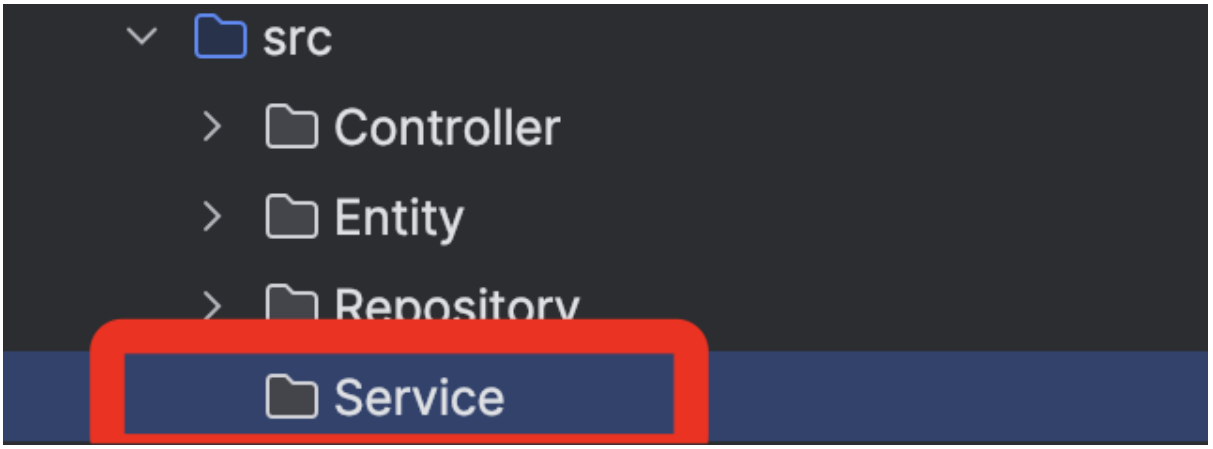
```
 created: src/Controller/PrixController.php
 created: templates/prix/index.html.twig


 Success!
```

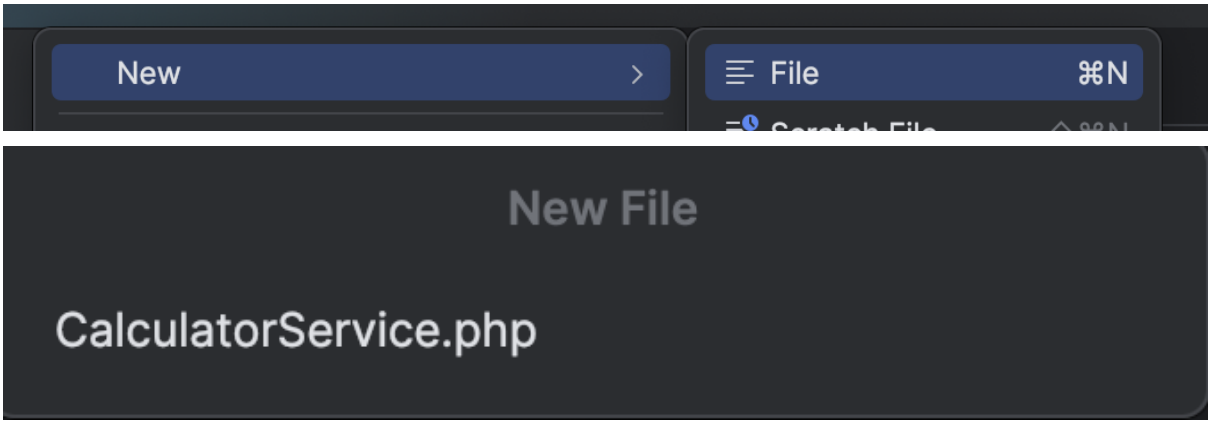
```php
final class PrixController extends AbstractController
{
    #[Route('/prix', name: 'app_prix')]
    public function index(): Response
    {
        return $this->render( view: 'prix/index.html.twig', [
            'controller_name' => 'PrixController',
        ]);
    }
}
```

Lorsque le Controller ou les Controllers sont prêts, nous pouvons créer le Service.

Tout d'abord, si ce n'est pas déjà fait, il faut créer un dossier dans "src" au nom de "Service" :



Ensuite on crée un fichier avec le nom de notre service et se terminant par ".php" (exemple : CaculatorService.php)



Lorsque le fichier est créé, à vide, il est tout d'abord obligatoire de lui ajouter l'intitulé "namespace App\Service;" afin de permettre à Symfony de comprendre que c'est un Service, ainsi qu'ajouter la class avec son nom (ici "Calculator")

```php
<?php
namespace App\Service;

no usages
class CalculatorService {

}
```

Il faut ensuite créer la fonction qui sera appelé dans les Controllers, ici une fonction de calcul :
La fonction permet de multiplier l'objet $prix par deux

```php
class CalculatorService {
    no usages
    public function multiplierpardeux(float $prix): float
    {
        return $prix * 2;
    }
}
```

On ajoute alors le service dans le Controller grâce à un use ainsi qu'un appel dans les parenthèses de la fonction de la route

```php
use App\Service\CalculatorService;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Attribute\Route;

no usages
final class PrixController extends AbstractController
{
    #[Route('/prix', name: 'app_prix')]
    public function index(CalculatorService $prix): Response
    {
```

On utilise ensuite le service :
(Dans cet exemple on initialise une variable $prix qu'on multiple par notre service de calculatrice)

```php
final class PrixController extends AbstractController
{
    #[Route('/prix', name: 'app_prix')]
    public function index(CalculatorService $prix): Response
    {
        $prixDebut = 100;
        $prixDouble = $prix->multiplierpardeux($prixDebut);

        return $this->render( view: 'prix/index.html.twig', [
            'PrixDebut' => $prixDouble,
            'PrixDouble' => $prixDouble,
        ]);
    }
}
```

Le service est désormais appelé dans le Controller, il est nécessaire d'en faire le rendu dans le twig (via le render)
On peut donc appeler le résultat du Service dans le Twig en appelant via "{{ }}" les objets rendus dans le twig

```twig
{% block body %}
    <h1>Produit</h1>

    <p>Prix initial : {{ PrixDebut }} €</p>
    <p>Prix doublé : {{ PrixDouble }} €</p>
{% endblock %}
```

Ce qui permet de visualiser sur la page :

Produit

Prix initial : 200 €

Prix doublé : 200 €