esprit Se former autrement	Année Universitaire: 2020-2021 **Workshops Framework Symfony4** **UP -Web**

Workshop : Création et manipulation d'un service

Objectif

Le but de ce workshop est la création et l'utilisation d'un service dans un projet symfony4

Définition

- Un service est une classe PHP qui permet de centraliser un code qui se répète
- Un service réalise une seule fonctionnalité (exemple : envoi d'email, pagination.)
- Un service est injectable dans toutes les classes.
- Chaque service a un alias
- Un projet peut contenir un ou plusieurs services
- Le framework Symfony propose plusieurs services prédéfinis (router,mailer,request…)
- La commande suivante permet d'afficher la liste des services :
  php bin/console debug:autowiring

Création d'un service

La création des services se fait sous le dossier **src/Service,** Ci-dessous un exemple de service qui permet de supprimer les caractères spéciaux d'un texte

```php
<?php

namespace App\Services;

class TextFormat
{
    public static function removeSpecialChar(string $text) : string
    {
        return preg_replace( pattern: '/[^A-Za-z0-9\-]/', replacement: '', $text);
    }
}
```

Manipulation

1ère méthode

Ajouter la configuration du service dans le fichier **config/services.yml**

```yaml
app.text_format:
    class: App\Services\TextFormat
    public: true
App\Services\TextFormat:
    alias: app.text_format
```

Dans le contrôleur, il suffit de mettre à jour le conteneur de service comme montre le code suivant

```php
/public static function getSubscribedServices(): array
{
    return array_merge(parent::getSubscribedServices(), [
        'app.text_format' => 'App\Services\TextFormat',
    ]);
}
```

Cette méthode retourne un tableau de services de la forme ‘Nom du service’ => ‘Classe du service’. Dans notre cas :

- Nom du service : app.text_format.

- Classe du service : App\Services\TextFormat

L'appel du service dans le contrôleur se fait comme suit :

```php
/**
 * @Route("/service", name="app_service_text_format")
 */
public function indexAction() {
    $textFormat = $this->get('app.text_format');
    return new Response($textFormat->removeSpecialChar( text: '**esprit $^$ è ; ?'));
}
```

2ème méthode (injection de dépendance) la plus recommandée

La 2ème méthode consiste à utiliser l'injection de dépendance de Symfony. Il suffit d'injecter le service dans la fonction comme montre le code suivant

```php
//2eme méthode
public function indexAction(TextFormat $service) {

    return new Response($service->removeSpecialChar( text: '*esprit $ *^$ è ; ?'));
}
```

Références

https://symfony.com/doc/4.4/service_container/alias_private.html

https://symfony.com/doc/4.4/service_container/autowiring.html