

Services et arguments

Passer manuellement des arguments

Il est possible de passer manuellement des arguments aux services pour les configurer.

Pour ce faire, il faut modifier le fichier `config/services.yaml` :

```
parameters:

services:
  _defaults:
    autowire: true
    autoconfigure: true

App\:
  resource: '../src/'
  exclude:
    - '../src/DependencyInjection/'
    - '../src/Entity/'
    - '../src/Kernel.php'
    - '../src/Tests/'

App\Service\MyLog:
  arguments:
    $logFile: 'logs/logs.txt'
```

Il suffit d'indiquer le nom du service entier (avec son `namespace`) et de configurer un ou plusieurs arguments.

Ici, supposons que nous voulons pouvoir modifier le fichier de logs à cet endroit uniquement.

Nous définissons donc un argument `$logFile` qui sera ensuite automatiquement passé au service.

Ainsi, dans notre service `MyLog.php`, nous pouvons récupérer l'argument dans le constructeur de la classe et l'utiliser directement :

```
<?php
```

```
namespace App\Service;

use Symfony\Component\Filesystem\Filesystem;

class MyLog
{
    function __construct(private Filesystem $fs, private string $logFile)
    {
    }

    public function writeLog(string $message)
    {
        $this->fs->appendToFile($this->logFile, $message . PHP_EOL);
    }
}
```

Utiliser un service comme argument d'un autre service

Vous pouvez passer en argument au constructeur d'un service, un autre service.

Pour ce faire, il faut le préfixer par @ et ajouter des guillemets :

parameters:

services:

 _defaults:

 autowire: true

 autoconfigure: true

App\:

 resource: '../src/'

 exclude:

- '../src/DependencyInjection/'
- '../src/Entity/'
- '../src/Kernel.php'
- '../src/Tests/'

App\Service\MyLog:

 arguments:

```
$logFile: 'logs/logs.txt'
```

```
$logger: '@logger'
```

Il suffit ensuite de le déclarer en argument sur le constructeur :

```
<?php
```

```
namespace App\Service;
```

```
use Symfony\Component\Filesystem\Filesystem;
```

```
class MyLog
```

```
{  
    function __construct(private Filesystem $fs, private string $logFile,  
private $logger)
```

```
{  
}
```

```
public function writeLog(string $message)
```

```
{  
    $this->logger->info('Test');  
    $this->fs->appendToFile($this->logFile, $message . PHP_EOL);  
}
```

```
}
```