

## Workshop :

### Analyse statique avec PHPStan dans Symfony

## Introduction :

PhpStan est un outil d'analyse statique pour PHP qui détecte les erreurs **sans exécuter le code**. Il améliore la qualité du code en imposant un typage strict et des bonnes pratiques.



## Objectifs :

- Comprendre l'importance de l'analyse statique en PHP
- Intégrer et configurer PHPStan dans un projet Symfony
- Corriger les erreurs détectées par PHPStan
- Améliorer la qualité du code grâce aux bonnes pratiques

## Étape 1 : Installation et configuration de PHPStan

Exécute la commande suivante pour ajouter le bundle à ton projet Symfony :

```
composer require --dev phpstan/phpstan
```

Ajouter l'extension Symfony

```
composer require --dev phpstan/phpstan-symfony
```

## Workshop :

### Analyse statique avec PHPStan dans Symfony

## Étape 2 : Création du fichier de configuration

Ajouter un fichier **phpstan.neon.dist** à la racine du projet :

```
includes:  
  - vendor/phpstan/phpstan-symfony/extension.neon  
  
parameters:  
  level: 5 # Niveau d'analyse (de 0 à 9, où 9 est le plus strict)  
  paths:  
    - src # Répertoire analysé  
  excludes_analyse:  
    - tests/* # Exclure les fichiers de test
```

## Étape 3 : Exécution de PHPStan

Lancer l'analyse du projet avec la commande suivante :

```
vendor/bin/phpstan analyse
```

La commande suivante nous permet de spécifier le dossier à analyser

```
vendor/bin/phpstan analyse src
```

PHPStan propose **10 niveaux (0 à 9)** :

- level: 0 → Vérifications basiques.
- level: 5 → Bon compromis entre flexibilité et rigueur.
- level: 9 → Analyse ultra-stricte.

**Workshop :**

**Analyse statique avec PHPStan dans Symfony**

```
vendor/bin/phpstan analyse --level=0
```

```
vendor/bin/phpstan analyse --level=9
```

**Références :**

<https://phpstan.org/>