

Le projet Symfony

Découverte et mise en place

Le projet "Demo Symfony"

Pour la suite de la formation, nous avons besoin d'un projet Symfony pour nos exemples, nous allons utiliser le projet de **démonstration** fournit par les équipes Symfony.

- Implémentes toutes les bonnes pratiques Symfony
- Fournit un jeu de données
- Fournit des tests

Notre espace de travail

Je vous propose de centraliser nos travaux dans un repertoire `formation-ci-cd`

A terme, il contiendra 3 sous-repertoires :

- `symfony-local`
- `symfony-gitlab`
- `symfony-github`

Créons donc le repertoire **racine** : `mkdir formation-ci-cd`

Installation du projet Demo

Installons et testons le projet **demo** sur notre poste.

```
cd formation-ci-cd  
symfony new --demo symfony-local  
cd symfony-local  
symfony serve -d  
symfony open:local
```

Les tests du projet

Nous pouvons lancer également les tests du projet avec **PHPunit**:

- `./bin/phpunit`

Les tests du projet sont consultable dans le repertoire

- `./tests`

Ajouter des Linteurs

Avant de nous lancer dans la création d'un pipeline CI/CD, nous allons découvrir et tester le fonctionnement de quelques linters :

- PHP-CS-FIXER
- PHP STAN
- Linteurs *Symfony*

PHP-CS-FIXER :

```
mkdir --parents tools/php-cs-fixer  
composer require --working-dir=tools/php-cs-fixer friendsofphp/php-cs-fixer  
./tools/php-cs-fixer/vendor/bin/php-cs-fixer fix --dry-run
```

PHP STAN :

On supprime les fichiers **phpstan** et on en créer un nouveau.

```
##On supprime les fichiers phpstan##  
composer require --dev phpstan/phpstan  
composer require --dev phpstan/extension-installer  
composer require --dev phpstan/phpstan-symfony  
vendor/bin/phpstan analyse src
```



```
# ./phpstan.neon
parameters:
    level: 6

    symfony:
        container_xml_path: var/cache/dev/App_KernelDevDebugContainer.xml

    checkGenericClassInNonGenericObjectType: false
    checkMissingIterableValueType: false

    ignoreErrors:
        - '#no type specified#'
        - '#Component#'
        - '#never written, only read#'
        - '#expects int, string given#'

    paths:
        - src
```

Linteurs "Symfony"

Fournit avec Symfony

```
# Twig
```

```
./bin/console lint:twig templates --env=prod
```

```
# YAML
```

```
./bin/console lint:yaml config --parse-tags
```

```
# Parameters and Services
```

```
./bin/console lint:container --no-debug
```

```
# Doctrine entities
```

```
./bin/console doctrine:schema:validate --skip-sync -vvv --no-interaction
```