

# Qualité de développement

## Linters

*Philippe Dosch*

[philippe.dosch@univ-lorraine.fr](mailto:philippe.dosch@univ-lorraine.fr)

*Florian Ferbach*

[florian.ferbach@univ-lorraine.fr](mailto:florian.ferbach@univ-lorraine.fr)

*Laurent Pierron*

[laurent.pierron@inria.fr](mailto:laurent.pierron@inria.fr)

**BUT S4.02 parcours DACS**





**C'est quoi un linter ?**

# PHP Lint

Vérifie les erreurs de syntaxe PHP.

## Installation

Rien à faire, inclus à PHP.

## Usage

- Fichier seul

```
$ php -l cfg/conf.sample.php
PHP Parse error:  Unterminated comment starting line 1 in cfg/conf.sample.php on line 1
Errors parsing cfg/conf.sample.php
```

- Tous les .php

```
$ find . -type f -name '*.php' -exec php -l {} \;
```

# PHP Code Sniffer

Vérifie la bonne application des standards PSR.

## Installation

L'installation s'effectue sur le projet directement.

```
# prérequis : extension PHP xml writer  
$ sudo apt install php-xmlwriter  
$ composer require --dev "squizlabs/php_codesniffer=3.*"
```

## Configuration

Configurable via CLI uniquement.

[Documentation](#)

# Usage

```
$ ./vendor/bin/phpcs --extensions=php ./lib/  
FILE: ./lib/Configuration.php
```

```
-----  
FOUND 42 ERRORS AND 15 WARNINGS AFFECTING 47 LINES  
-----
```

```
 5 | ERROR    | [ ] Doc comment long description must start with a capital letter  
 7 | ERROR    | [ ] The tag in position 1 should be the @copyright tag  
 9 | WARNING  | [ ] Line exceeds 85 characters; contains 90 characters  
10 | WARNING  | [ ] Invalid version "1.5.1" in file comment; consider "CVS: <cv<id>" or "SVN:  
   |          |     <svn_id>" or "GIT: <git_id>" or "HG: <hg_id>" instead  
11 | WARNING  | [ ] PHP version not specified  
11 | ERROR    | [ ] Missing @category tag in file comment  
11 | ERROR    | [ ] Missing @package tag in file comment  
11 | ERROR    | [ ] Missing @author tag in file comment  
21 | ERROR    | [ ] Doc comment long description must start with a capital letter  
22 | ERROR    | [ ] Missing @category tag in class comment  
22 | ERROR    | [ ] Missing @package tag in class comment  
22 | ERROR    | [ ] Missing @author tag in class comment  
22 | ERROR    | [ ] Missing @license tag in class comment  
...
```

# PHP Mess Detector

Vérifie la présence de problèmes potentiels dans le code : potentiels bugs, code non optimisé, expressions trop compliquées, code non utilisé.

## Installation

L'installation s'effectue sur le projet directement.

```
$ composer require --dev "phpmd/phpmd=@stable"
```

## Configuration

Configurable via un fichier `ruleset.xml`.

[Documentation](#)

# Usage

```
$ ./vendor/bin/phpmd ./lib ansi codesize,unusedcode,naming
```

```
FILE: ./lib/Configuration.php
```

```
-----  
107 | VIOLATION | The method __construct() has a Cyclomatic Complexity of 35. The configured cyclomatic complexity threshold is 10.  
107 | VIOLATION | The method __construct() has an NPath complexity of 22860. The configured NPath complexity threshold is 200.  
107 | VIOLATION | The method __construct() has 141 lines of code. Current threshold is set to 100. Avoid really long methods.
```

```
FILE: ./lib/Controller.php
```

```
-----  
24  | VIOLATION | The class Controller has an overall complexity of 50 which is very high. The configured complexity threshold is 50.  
111 | VIOLATION | The method __construct() has a Cyclomatic Complexity of 10. The configured cyclomatic complexity threshold is 10.  
199 | VIOLATION | The method _create() has a Cyclomatic Complexity of 11. The configured cyclomatic complexity threshold is 10.
```

```
FILE: ./lib/Data/Database.php
```

```
-----  
26  | VIOLATION | The class Database has an overall complexity of 122 which is very high. The configured complexity threshold is 50.  
60  | VIOLATION | The method __construct() has a Cyclomatic Complexity of 13. The configured cyclomatic complexity threshold is 10.  
143 | VIOLATION | The method create() has a Cyclomatic Complexity of 10. The configured cyclomatic complexity threshold is 10.
```

```
...
```

# #1



- Installez et testez ces 3 linters sur le projet
- Configurer les différents linters avec les règles qui vous paraissent les plus intéressantes
- Créez une nouvelle target dans le makefile de sorte qu'on puisse appeler ces 3 linters avec la commande `make lint`

```
# ...  
lint:  
    # commandes de lint ici  
    # sur plusieurs lignes éventuellement
```

- Corriger 5+ erreurs différentes remontées par ces linters
- N'oubliez pas de commit vos modifications sur GitHub



# **Automatisation**

# PHP CS Fixer

Comme PHP Code Sniffer mais propose de fix automatiquement tout ce qui peut être fixé automatiquement.

## Installation

L'installation s'effectue sur le projet directement.

```
$ composer require --dev "friendsofphp/php-cs-fixer"
```

## Configuration

Configurable via un fichier `.php-cs-fixer.dist.php`.

[Documentation](#)

## Usage

```
$ ./vendor/friendsofphp/php-cs-fixer/php-cs-fixer fix lib
Loaded config default.
 1) lib/Configuration.php
 2) lib/Controller.php
 3) lib/Request.php
 4) lib/Data/FileSystem.php
 5) lib/Data/Database.php
 6) lib/YourIsProxy.php
 7) lib/I18n.php

Fixed 7 of 23 files in 0.838 seconds, 16.000 MB memory used
```

Aussi disponible sous forme d'extension d'éditeur: <https://github.com/PHP-CS-Fixer/PHP-CS-Fixer#editor-integration>

# Pre-commit hook

Un pre-commit hook permet d'effectuer une action avant chaque commit.

## Configuration

Configurable via un fichier bash `.git/hooks/`.

Exemple:

```
#!/usr/bin/env bash

# If any command fails, exit immediately with that command's exit status
set -eo pipefail

# Find all changed files for this commit
# Compute the diff only once to save a small amount of time.
CHANGED_FILES=$(git diff --name-only --cached --diff-filter=ACMR)
# ...
```

```
#!/usr/bin/env bash

# If any command fails, exit immediately with that command's exit status
set -eo pipefail

# Find all changed files for this commit
# Compute the diff only once to save a small amount of time.
CHANGED_FILES=$(git diff --name-only --cached --diff-filter=ACMR)

if [[ -n "$CHANGED_FILES" ]]
then
    # Run black against changed python files for this commit
    black --check $CHANGED_FILES
    echo "black passes all altered python sources."
    # Run flake8 against changed python files for this commit
    flake8 $CHANGED_FILES
    echo "flake8 passed!"
fi
```

# #2



- Installez un pre-commit hook qui :
  - utilise PHP CS Fixer ou prettier pour fixer automatiquement le maximum d'erreurs
  - ajoute ce que PHP CS Fixer a fixé au staging area
  - utilise PHP Mess detector pour interdire le commit si il y a des erreurs
- N'oubliez pas de commit vos modifications sur GitHub : adaptez la configuration de vos linters ou corrigez le code si nécessaire
- Est-il possible de commit en ignorant le pre-commit hook ?



# Infos complémentaires

- PHP CS Fixer peut être appelé sur une liste de fichiers

```
./vendor/friendsofphp/php-cs-fixer/php-cs-fixer fix --config=.php-cs-fixer.php fichier1.php lib/autreFichier.php ...
```

Mais a besoin d'un fichier de configuration `.php-cs-fixer`. [Documentation](#)

- PHPMD peut être appelé sur une liste de fichiers

```
./vendor/bin/phpmd fichier1.php,lib/autreFichier.php ansi codesize,unusedcode,naming
```

Mais les fichiers doivent cette fois être séparés par des virgules.



Pour transformer `fichier1.php lib/autreFichier.php` en `fichier1.php,lib/autreFichier.php`, on peut déclarer cette fonction

```
function join_by {  
    local d=${1-} f=${2-}  
    if shift 2; then  
        printf %s "$f" "${@/#/$d}"  
    fi  
}
```

Utilisée ainsi :

```
echo $(join_by , ${CHANGED_FILES})
```



# Intégration Continue

La CI (Continuous Integration) est un principe qui lance des outils de qualité de code sur chaque PR, et qui permet d'interdire de la merger dans `main` si les attentes ne sont pas respectées.

Chez GitHub, ces tests de qualités sont généralement disponibles sous forme de **GitHub Actions**

 center

# #3



- Installez [une action GitHub de lint](#) sur votre repository : l'action devra empêcher de merger des pull request s'il y a des erreurs de lint
- Puisqu'on veut systématiquement passer par le système de pull request pour profiter de l'action GitHub de lint, protéger la branche `main` pour que personne ne puisse y envoyer des commits directement. Trouvez l'option dans les settings de votre projet GitHub.

# Pour aller plus loin...

- [prettier](#) : un code formatter multilangage aux idées arrêtées
- [pre-commit](#) : un framework de pre-commit hooks