

PHP CodeSniffer (a.k.a phpcs)

Usage

- Syntax:

```
$ phpcs /path/to/file /path/to/dir ...
```

- Set pre-commit hook

```
$ cp /home/dev/joseph_chiang/muchiii/.git/hooks/pre-commit /home/dev/$USER/muchiii/.git/hooks/pre-commit
$ chmod +x /home/dev/$USER/muchiii/.git/hooks/pre-commit
```

- After setting this hook, you can't commit without passing all sniffs checks. However, you can still use --no-verify option to bypass pre-commit hook.

```
$ git commit -a --no-verify
```

Installation

- Use pear install:

```
sudo pear install PHP_CodeSniffer
```

- Folder Structure:

```
/usr/share/pear/PHP/CodeSniffer
|-- CLI.php
|-- CommentParser
|-- DocGenerators
|-- Exception.php
|-- File.php
|-- MultiFileSniff.php
|-- Report.php
|-- Reporting.php
|-- Reports
|-- Sniff.php
|-- Standards
|-- Tokenizers
`-- Tokens.php
```

General Configuration

1. Add Code Standard:

Table of Contents

[Usage](#)[Installation](#)[General Configuration](#)[JSLint Configuration](#)[Git client pre-commit hook](#)

```
$ scp -r -P 443 josephj@josephj.com:/home/josephj/yahoo-backup-20090830
$ sudo mv /usr/share/pear/PHP/CodeSniffer/Standard /usr/share/pear/PHP/
$ sudo mv ~/Standard /usr/share/pear/PHP/CodeSniffer/.
$ phpcs -i #check how many standards have been installed
```

2. Change Code Standard

```
$ phpcs --config-set default_standard YahooStrict
```

3. Modify Code Standard

```
$ sudo vim /usr/share/pear/PHP/CodeSniffer/Standards/Yahoo/Sniffs
```

4. Check configuration file manually

```
$ sudo vim /usr/share/pear/data/PHP_CodeSniffer/CodeSniffer.conf
```

JSLint Configuration

1. Rhino path setting:

```
$ sudo phpcs --config-set rhino_path /usr/share/rhino
```

2. JSLint path setting:

```
$ sudo phpcs --config-set jslint_path /home/m/lib/jslint/jslint.js
```

- `sudo phpcs --config-set jslint_path /home/m/lib/jslint/jslint.js`

Git client pre-commit hook

```
#!/usr/bin/php
<?php $output = array(); $rc = 0;
exec("git rev-parse --verify HEAD 2> /dev/null", $output, $rc);
if ($rc == 0)
{
    $against = "HEAD";
}
else
{
    $against = "4b825dc642cb6eb9a060e54bf8d69288fbee4904";
}
exec("git diff-index --cached --name-only ". $against, $output);
$pattern = "/\.php$/";
$exit_status = 0;
// Check Code Standard Error
$files = array();
foreach ($output as $file)
{
    if (!preg_match($pattern, $file))
    {
        continue;
    }
    $files[] = escapeshellarg($file);
}
$output = array();
$return = 0;
$list = implode(" ", $files);
exec("phpcs -n --report=summary " . (implode(" ", $files)), $output, $return);
if ($return == 1)
{
    $exit_status = 1;
    echo "COMMIT FAILED!\n";
    echo "Code standard reports: \n";
    $output = array_slice($output, 2);
    array_splice($output, count($output) - 4, 2);
    echo implode("\n", $output);
    echo "Please fix your code and commit again.\n";
}
exit($exit_status);
?>
```