

PHP Code Sniffer with Symfony2 Coding Standards Integration in PhpStorm - Symfony Development using PhpStorm



Redirection Notice

This page will redirect to <https://www.jetbrains.com/help/phpstorm/using-php-code-sniffer-tool.html> in about 2 seconds.

[Previous](#) | [Symfony2 Command Line Tool Integration](#)

[Top](#) | [Symfony Development using PhpStorm](#)

[Next](#) | [Debugging, Profiling and Testing Symfony2](#)

[Tweet](#)

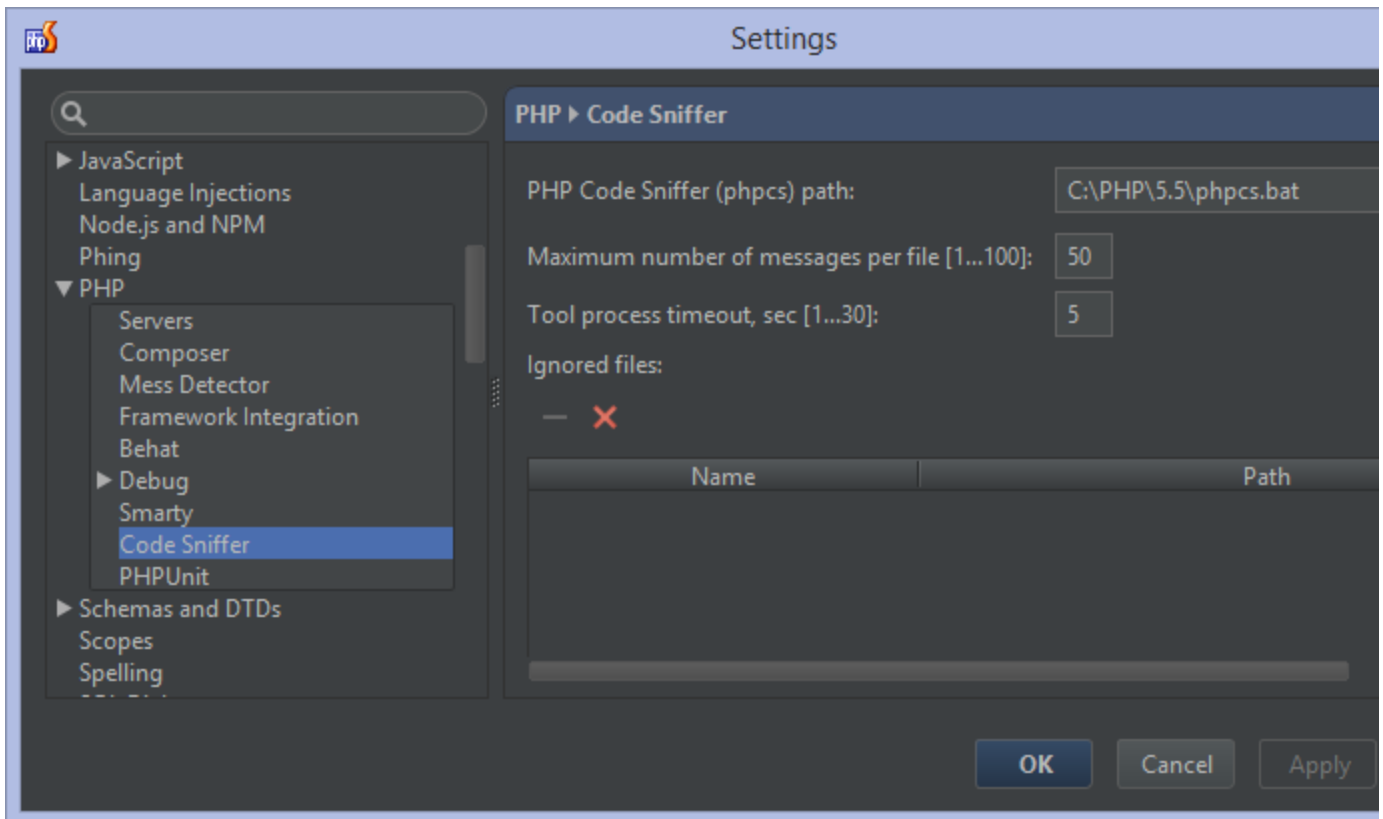
PHP Code Sniffer with Symfony2 Coding Standards Integration in PhpStorm

[PHP Code Sniffer](#) is a tool designed to detect violations of a defined coding standard. It is an essential development tool that helps ensure your code remains clean and consistent. It can also help prevent some common semantic errors made by developers.

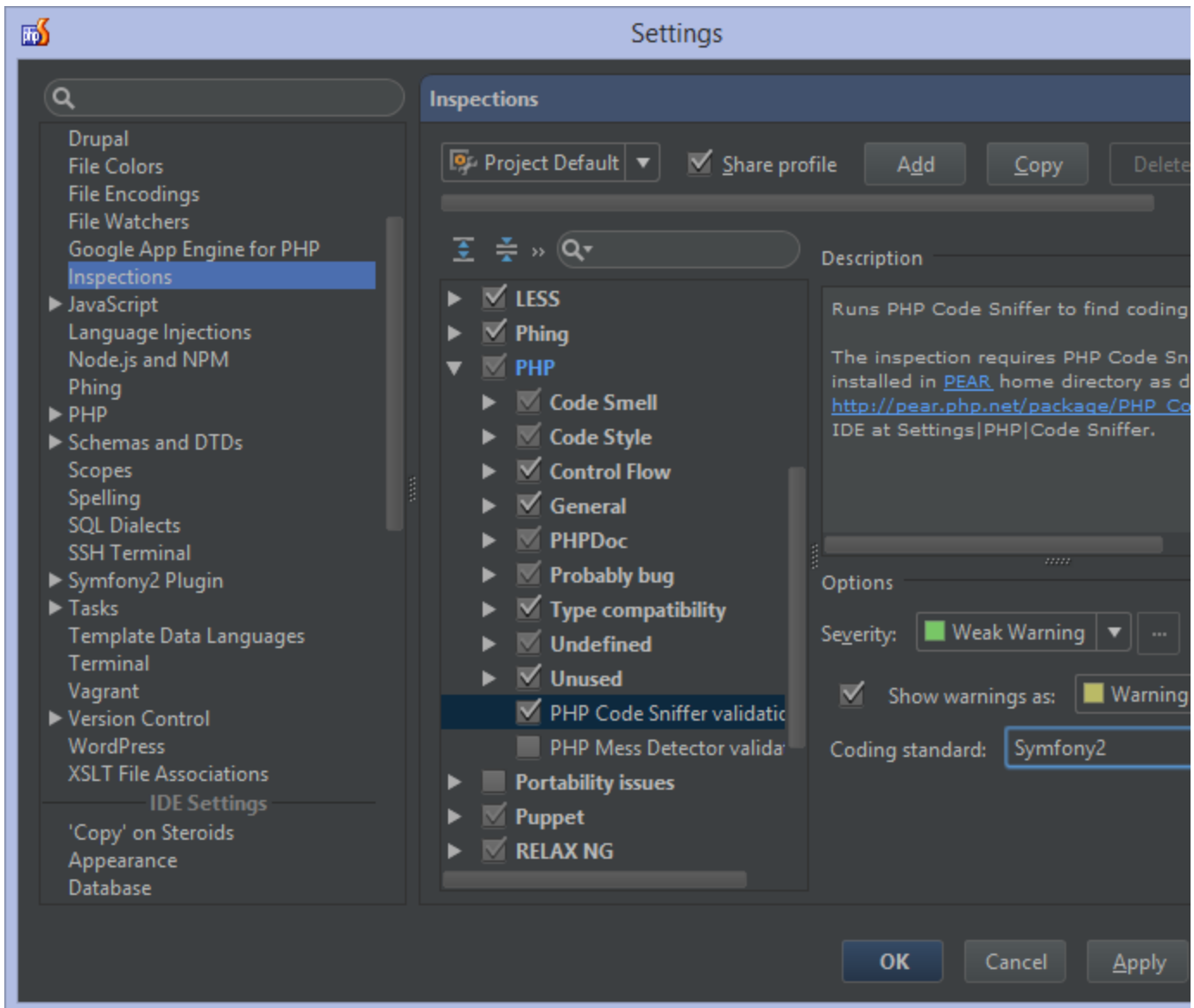
[Symfony2 Coding Standards](#) rules for PHP Code Sniffer are maintained in [this GitHub repository](#) by the community.

PHP Code Sniffer is supported in PhpStorm, which means you can get its notifications inside the IDE editor. Symfony2-specific coding standards are easy to add in PHP Code Sniffer, so that Symfony Coding Standards violations will be shown in the IDE editor along with standard PHP Code Sniffer notifications.

1. Install PHP Code Sniffer tool as a PEAR package or by using the Composer tool. Installation instructions are available in the [PhpStorm web help](#) and on the [PHP Code Sniffer webpage](#).
2. Download [Symfony2 Coding Standards rules for PHPCode Sniffer](#) following installation guide described in the GitHub repository. The entire package can be downloaded as an archive or checked out from the git repository.
3. Unpack the downloaded archive (if needed) and navigate to the directory containing files (the one with Sniffs folder and ruleset.xml file inside). You need to move all the directory contents to `/usr/share/php/PHP/CodeSniffer/Standards/Symfony2` (the path depends on your configuration). A file named `ruleset.xml` should be in this directory as well as some other subdirectories.
4. Configure PHP Code Sniffer by providing the path in Project Settings | PHP | Code Sniffer. Click the Validate button to check if the Code Sniffer tool can be reached.



5. Turn on PHP Code Sniffer inspections in Project Settings | Inspections | PHP | PHP Code Sniffer validation. Select the Symfony2 coding standard in the list (use the Refresh button if needed). Click OK to enable these new settings.



6. Now your code is inspected by PHP Code Sniffer with Symfony2 Coding Standards, in addition to PhpStorm's inspections. You will see notifications marked with phpcs in the editor.

```

AssetCollection.php x
\AssetCollection

<?php
/**
 * An asset collection.
 */
interface AssetCollection extends AssetInterface, \Traversable
{
    /**
     * Returns all child assets.
     *
     * @return array An array of AssetInterface objects
     */
    public function all();

    /**
     * Adds an asset to the current collection.
     *
     * @param AssetInterface $asset An asset
     */
    public function add(AssetInterface $asset);
}

```

phpcs: Interface name is not suffixed with "Interface"

7. If we run batch inspections with Code | Inspect Code, the results of PHP Code Sniffer inspections will be listed in the section PHP | PHP Code Sniffer validation.

Inspection Results for Inspection Profile 'Project Default'

- SymfonyBlog (10,440 items)
 - Code Style (3 items)
 - Code style issues (1 item)
 - CSS (51 items)
 - General (16 items)
 - HTML (8 items)
 - Invalid elements (664 items)
 - PHP (109 items)
 - PHP Code Sniffer validation (109 items)
 - src (68 items)
 - phpcs: Interface name is not suffixed with "Interface"
 - phpcs: Missing class doc comment
 - phpcs: Missing class doc comment
 - phpcs: Missing class doc comment

Name: AssetCollection.php

Location: file [SymfonyBlog] - ... \src \Blogger \BlogBundle \A

Problem synopsis: phpcs: Interface name is not suffixed with "Interface" (at line 5)

Check PHP Code Sniffer in PhpStorm for more information on using this tool.

[Previous](#) | [Symfony2 Command Line Tool Integration](#)

[Top](#) | [Symfony Development using PhpStorm](#)

[Next](#) | [Debugging, Profiling and Testing Symfony2](#)
[Tweet](#)