

PHP_CodeSniffer

Avi Miller, Product Evangelist

Squiz.net

Agenda

- ✦ What is PHP_CodeSniffer?
 - ✦ What is a coding standard?
- ✦ Why should I use it?
 - ✦ What doesn't it do?
- ✦ How do I use it?
 - ✦ Advanced usage options
- ✦ How do I make it better?
 - ✦ Creating your own standard/sniffs

Installing PHP_CodeSniffer

- ✦ Installing via PEAR:
 - ✦ # pear install PHP_CodeSniffer
- ✦ Packages exist for:
 - ✦ Debian

What is PHP_CodeSniffer?

- ✦ PHP_CodeSniffer is a PHP5 script that tokenises and "sniffs" PHP code to detect violations of a defined set of coding standards
- ✦ It is an essential development tool that ensures that your code remains clean and consistent
- ✦ It can even help prevent some common semantic errors made by developers

What is a coding standard?

- ✦ A coding standard, in PHP_CodeSniffer, is a collection of *sniff* files
- ✦ Each sniff file checks one part of the coding standard only
- ✦ Multiple coding standards can be used within PHP_CodeSniffer
- ✦ The default coding standard used by PHP_CodeSniffer is the PEAR coding standard

Example

```
$ phpcs /path/to/code/myfile.php
```

```
FILE: /path/to/code/myfile.php
```

```
-----  
FOUND 5 ERROR(S) AFFECTING 2 LINE(S)  
-----
```

```
 2 | ERROR | Missing file doc comment  
20 | ERROR | PHP keywords must be lowercase; expected "false" but found "FALSE"  
47 | ERROR | Line not indented correctly; expected 4 spaces but found 1  
51 | ERROR | Missing function doc comment  
88 | ERROR | Line not indented correctly; expected 9 spaces but found 6  
-----
```

Why should I use it?

- ✦ If you want to ensure you adhere to a set of coding standards, PHP_CodeSniffer is a quick and easy way to do that
- ✦ PHP_CodeSniffer is a replacement for the more manual task of checking coding standards in code reviews
- ✦ With PHP_CodeSniffer, you can reserve code reviews for the checking of code execution/functionality
- ✦ PHP_CodeSniffer uses the in-built PHP tokenizer

What doesn't it do?

- ✦ PHP_CodeSniffer does not perform any code coverage or unit testing
- ✦ PHP_CodeSniffer does not parse your code, and cannot tell if your code contains parse errors
- ✦ PHP_CodeSniffer does not fix your code for you

More examples

- ✦ Checking files and folders
- ✦ Printing a summary report
- ✦ Printing verbose output
- ✦ Specifying a coding standard
- ✦ Printing a list of installed coding standards

Advanced usage

- ✦ Specifying valid file extensions
- ✦ Ignoring files and folders
- ✦ Replacing tabs with spaces
- ✦ Printing an XML/Checkstyle Report
- ✦ Printing a CSV report
- ✦ Analysing verbose/debugging output

Creating a coding standard

- ✦ Directory structure
- ✦ Coding standard class
 - ✦ `getIncludedSniffs()` and `getExcludedSniffs()`
- ✦ Creating a sniff
 - ✦ `register()` and `process()`
 - ✦ the Token stack

Documentation

- ✦ http://pear.php.net/package/PHP_CodeSniffer

Questions?