

Magento Coding Standard

 [View license](#)

☆ 212 stars  82 forks

☆ Star

👁 Watch ▼

Code

Issues 23

Pull requests 2

Actions

Projects 2

Wiki

Security

Insights

 develop ▼

...



sivaschenko ...

✓ 3 days ago



[View code](#)

README.md

Magento Coding Standard

A set of Magento rules for [PHP_CodeSniffer](#) tool.

Installation within a Magento 2 site

To use within your Magento 2 project you can use:

```
composer require --dev magento/magento-coding-standard
```

Due to security, when installed this way the Magento standard for phpcs cannot be added automatically. You can achieve this by adding the following to your project's

`composer.json` :

```
"scripts": {
    "post-install-cmd": [
        "([ $COMPOSER_DEV_MODE -eq 0 ] || vendor/bin/phpcs --config-set installed_paths",
    ],
    "post-update-cmd": [
        "([ $COMPOSER_DEV_MODE -eq 0 ] || vendor/bin/phpcs --config-set installed_paths",
    ]
}
```

Installation for development

You can install Magento Coding Standard by cloning this GitHub repo:

```
git clone git@github.com:magento/magento-coding-standard.git
cd magento-coding-standard
composer install
```

It is possible also to install a standalone application via [Composer](#)

```
composer create-project magento/magento-coding-standard --stability=dev magento-codin
```

Verify installation

Command should return the list of installed coding standards including Magento2.

```
vendor/bin/phpcs -i
```

Usage

Once installed, you can run `phpcs` from the command-line to analyze your code

`MyAwesomeExtension`

```
vendor/bin/phpcs --standard=Magento2 app/code/MyAwesomeExtension
```

Fixing issues automatically

Also, you can run `phpcbf` from the command-line to fix your code `MyAwesomeExtension` for warnings like "PHPCBF CAN FIX THE [0-9]+ MARKED SNIFF VIOLATIONS AUTOMATICALLY"

```
vendor/bin/phpcbf --standard=Magento2 app/code/MyAwesomeExtension
```

Contribution

See the community [contribution model](#).

Where to contribute

- Documentation of existing rules. See [ExtDN PHP CodeSniffer rules for Magento 2](#) as a good example.
- Bug fixes and improvements of existing rules.
- Creation of new PHP CodeSniffer rules.
- Discussions on new rules (through periodic hangouts or discussions per GitHub issue).

How to contribute

1. Start with looking into [Community Dashboard](#). Any ticket in `Up for grabs` is a good candidate to start.
2. Didn't satisfy your requirements? [Create one of three types of issues](#):
 - **Bug report** - Found a bug in the code? Let us know!
 - **Existing rule enhancement** - Know how to improve existing rules? Open an issue describe how to enhance Magento Coding Standard.
 - **New rule proposal** - Know how to improve Magento ecosystem code quality? Do not hesitate to open a proposal.
3. The issue will appear in the `Backlog` column of the [Community Dashboard](#). Once it will be discussed and get `accepted` label the issue will appear in the `Up for grabs` column.

Testing

All rules should be covered by unit tests. Each `Test.php` class should be accompanied by a `Test.inc` file to allow for unit testing based upon the `PHP_CodeSniffer` parent class `AbstractSniffUnitTest`. You can verify your code by running

```
vendor/bin/phpunit
```

Also, verify that the sniffer code itself is written according to the Magento Coding Standard:

```
vendor/bin/phpcs --standard=Magento2 Magento2/ --extensions=php
```

License

Each Magento source file included in this distribution is licensed under the OSL-3.0 license.

Please see [LICENSE.txt](#) for the full text of the [Open Software License v. 3.0 \(OSL-3.0\)](#).

 **Version 6** (Latest)

3 days ago

[+ 8 releases](#)

Packages

No packages published

Used by 1.9k



Contributors 31



[+ 20 contributors](#)

Languages

● PHP 100.0%