Magento Coding Standard

The **Magento Coding Standard** is a set of rules and guidelines for writing clean, consistent, and high-quality code in Magento. These standards are designed to ensure code readability, maintainability, and compatibility with the Magento ecosystem.

Magento leverages **PHP_CodeSniffer (PHPCS)** to enforce coding standards. The Magento Coding Standard is provided as a custom set of rules for PHP_CodeSniffer and helps developers detect and fix issues in their code.

Key Features

1. **Consistency**: Ensures a uniform coding style across the codebase.
2. **Error Detection**: Identifies potential bugs or anti-patterns.
3. **Compliance**: Helps developers write code that adheres to Magento's best practices.

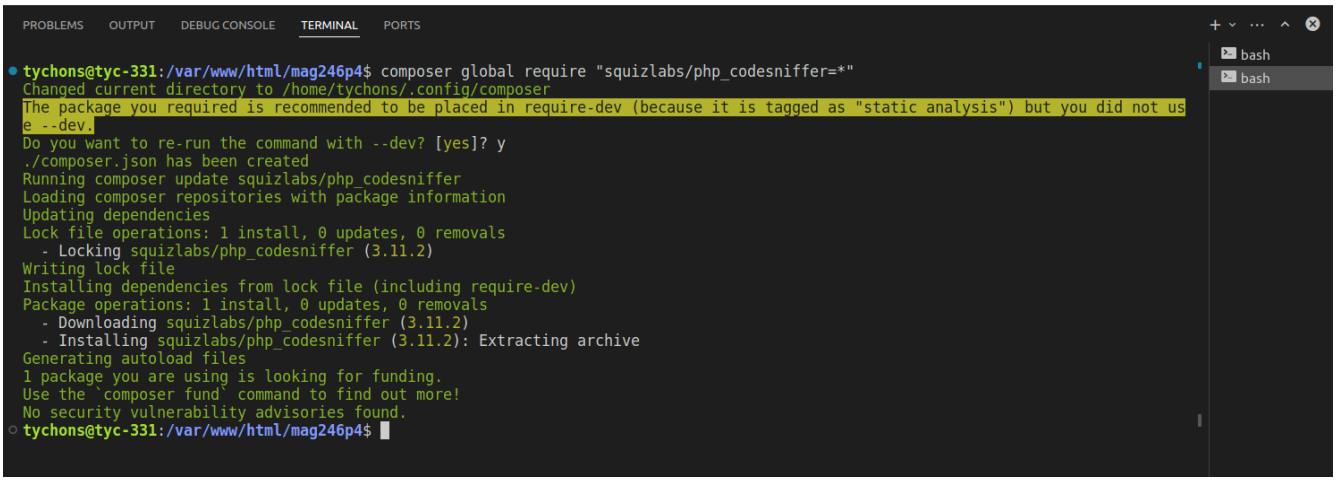
How to Use Magento Coding Standard

Step 1: Install PHP_CodeSniffer

You need PHP_CodeSniffer installed on your system. Use Composer to install it globally or locally:

Bash

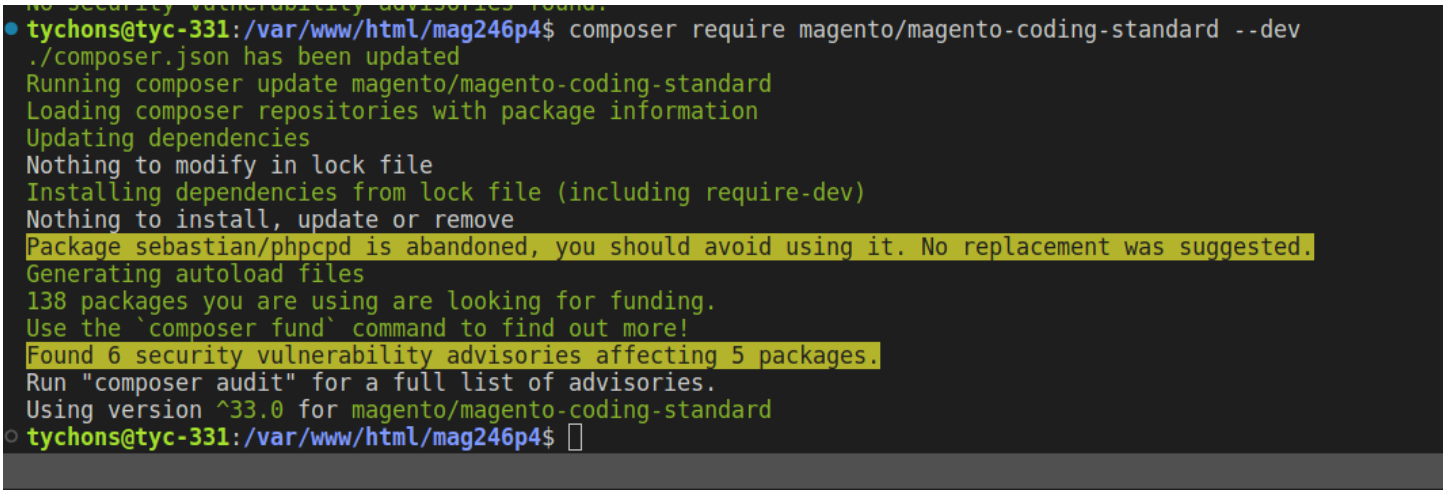
composer global require "squizlabs/php_codesniffer=*"



Step 2: Install Magento Coding Standard

Magento provides a custom set of coding standards that you can include in your project. Use Composer to install the Magento Coding Standard package:

bash
composer require magento/magento-coding-standard –dev



Step 3: Configure PHP_CodeSniffer

Specify Magento's coding standard in PHP_CodeSniffer:

bash
phpcs --config-set installed_paths vendor/magento/magento-coding-standard

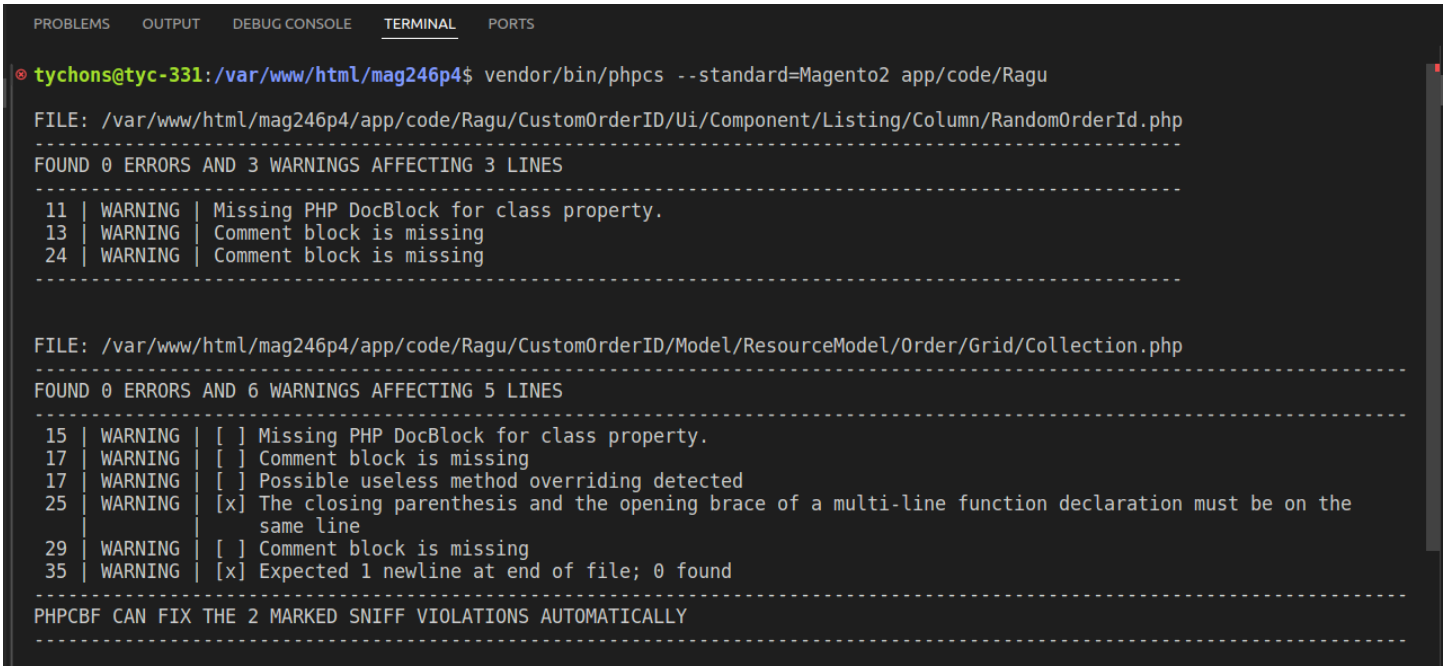
This command sets up Magento's coding standard rules for PHP_CodeSniffer.

Step 4: Validate Code Against Magento Standards

Run the following command to analyze your code:

bash
phpcs --standard=Magento2 /path/to/your/code

- Replace **/path/to/your/code** with the directory or file you want to analyze.
- It will display warnings and errors that violate Magento's standards.



Step 5: Fix Issues (Optional)

You can use the phpcbf command to **automatically fix** some coding standard issues:

bash

```
phpcbf --standard=Magento2 /path/to/your/code
```