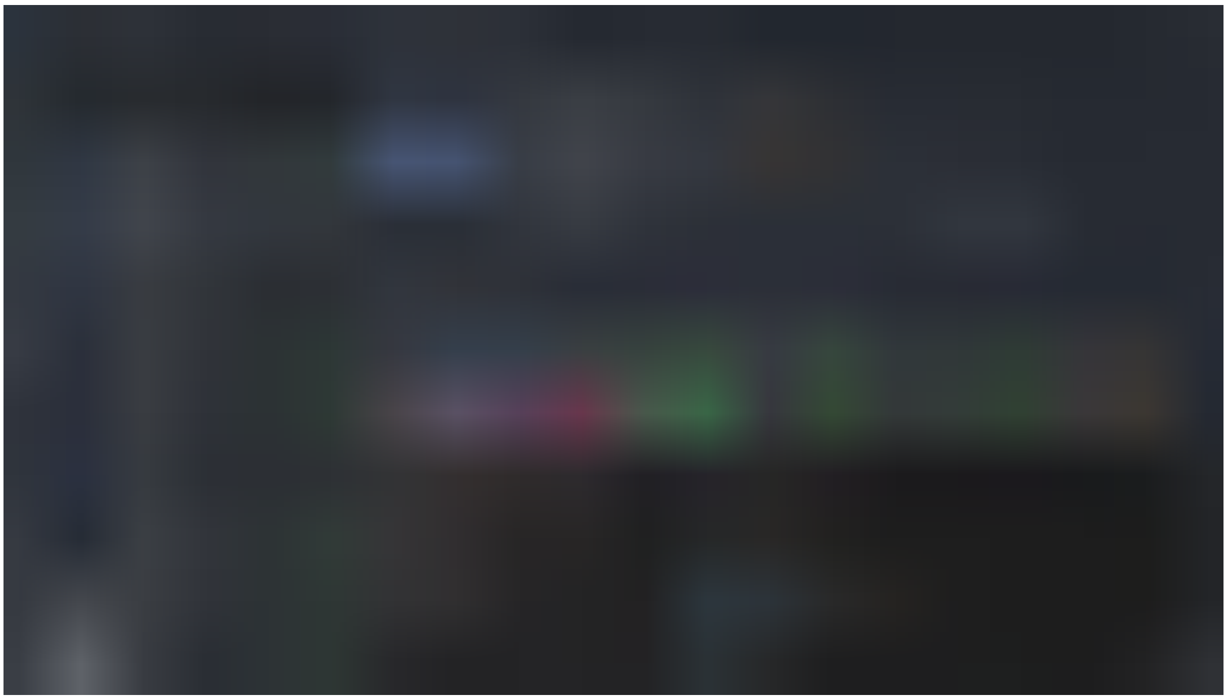
# Setup VS Code for Efficient PHP development 🚀

July 19, 2019 Louis-Marie Michelin 11 min read



I recently started programming in PHP using the well-known Symfony framework. I wanted to keep my VS Code habits and proficiency I had from my previous projects in Node.js and Vue.js, so I tried to configure VS Code for PHP development as well. Moreover, I didn't want to invest €199 in the famous PHPStorm IDE... 😕

## TL;DR

### VS Code extensions you should install and configure for PHP development:

- **PHP debug**
- **PHP CS Fixer**
- **Twig Language 2**
- **SQLTools**

In this article, I will explain how I managed to make my development environment comparable to PhpStorm's. The commands will be detailed for Ubuntu 18.04, but they can be adapted for Mac OS and possibly for Windows.

I will take an empty Symfony application as an example for this article. Let's start!

## Prerequisites

### Install PHP and Composer

To be able to run PHP code you obviously need to install PHP. You will also probably need Composer, a commonly used dependency manager for PHP. You can install them by running the following command:

```
sudo apt install php-cli composer
```

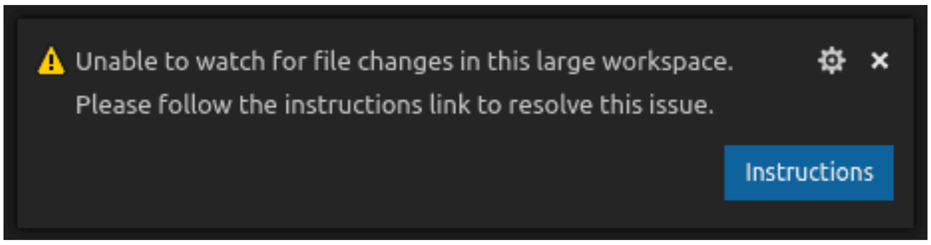
You can install `php-all-dev` instead of `php-cli` if you want to install some useful libraries as well.

Code and launch the development server:

```
composer create-project symfony/website-skeleton m
cd my-symfony-website
code .
bin/console server:run
```

## Increase the number of system watchers if needed

VS Code needs to watch for file changes to work correctly. That's why the following warning may appear because our Symfony project contains a lot of files:



Unable to watch for file changes in this large workspace. Please follow the instructions link to resolve this issue.

To fix it, you can run the following commands in your terminal, and then restart VS Code:

```
echo fs.inotify.max_user_watches=524288 | sudo te
sudo sysctl -p
```

Go to **http://127.0.0.1:8000** from your favorite web browser and check that your Symfony website works.

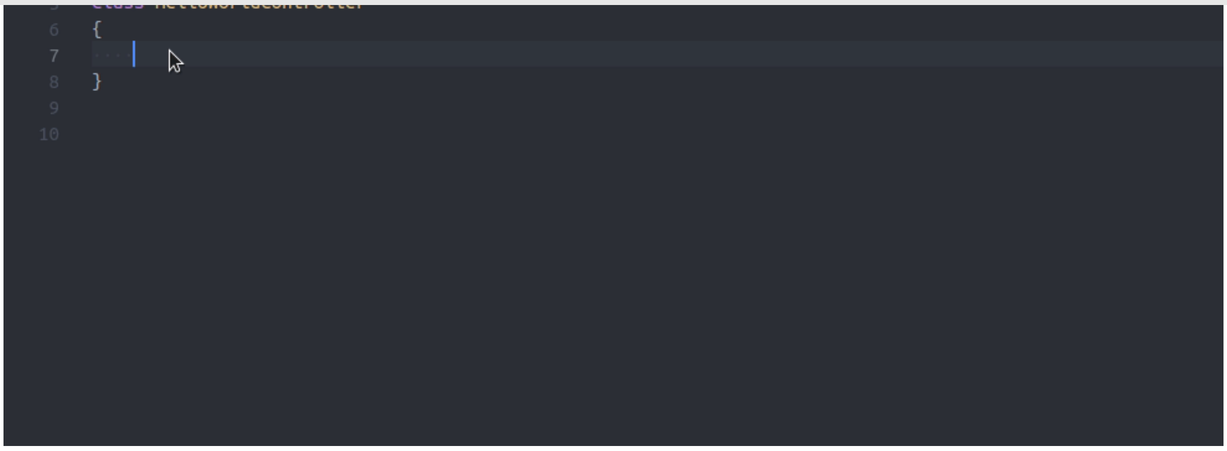
## Autocomplete & Intellisense & Go to definition

Let's create a new route in our application. Just add an empty file `HelloWorldController.php` in the `src/Controller` folder.

### Install PHP Intelephense

By default, VS Code provides some basic code suggestions but they are generic and quite useless. **To get useful code suggestions as you type, I recommend installing the PHP Intelephense extension for VS Code.** I also tried the **PHP Intellisense extension** but it's slower on big projects and provides fewer features (it doesn't gray out unused imports for example).

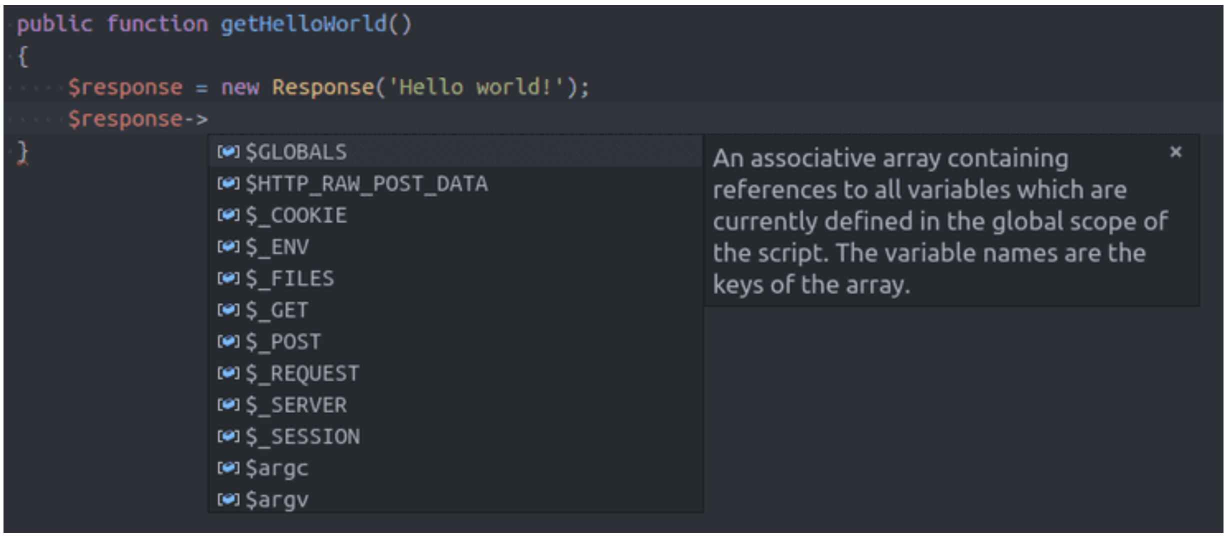
Now that the **PHP Intelephense extension** is installed, we can start writing our HelloWorldController!


```
{

}
```

As you can see, code suggestions are very relevant and VS Code auto imports the corresponding namespace when you validate a suggestion!

## Disable basic suggestions

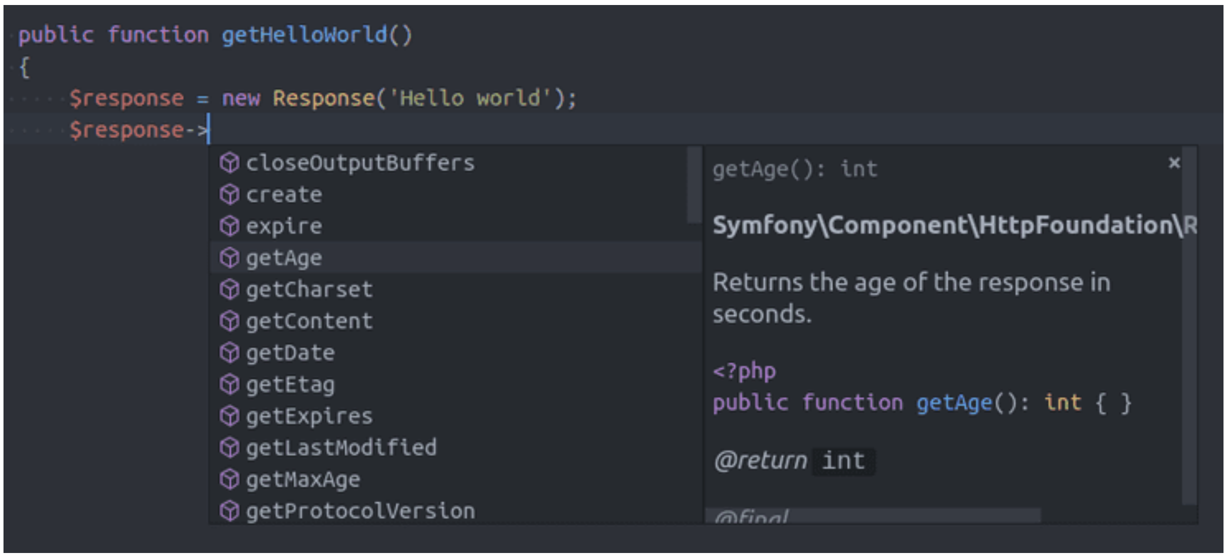
PHP Intelephense can also auto-suggest relevant methods when typing `$myVar->`. The problem is that these suggestions are polluted by the default PHP suggestions.


```
public function getHelloWorld()
{
    $response = new Response('Hello world!');
    $response->
}
```
```
$GLOBALS
$HTTP_RAW_POST_DATA
$_COOKIE
$_ENV
$_FILES
$_GET
$_POST
$_REQUEST
$_SERVER
$_SESSION
$argc
$argv
```
An associative array containing references to all variables which are currently defined in the global scope of the script. The variable names are the keys of the array.

Basic PHP code suggestions we want to disable the following line:

```
"php.suggest.basic": false,
```

When it's done, you can see useful suggestions!



```
public function getHelloWorld()
{
    $response = new Response('Hello world');
    $response->
```

getAge(): int

Symfony\Component\HttpFoundation\R

Returns the age of the response in seconds.

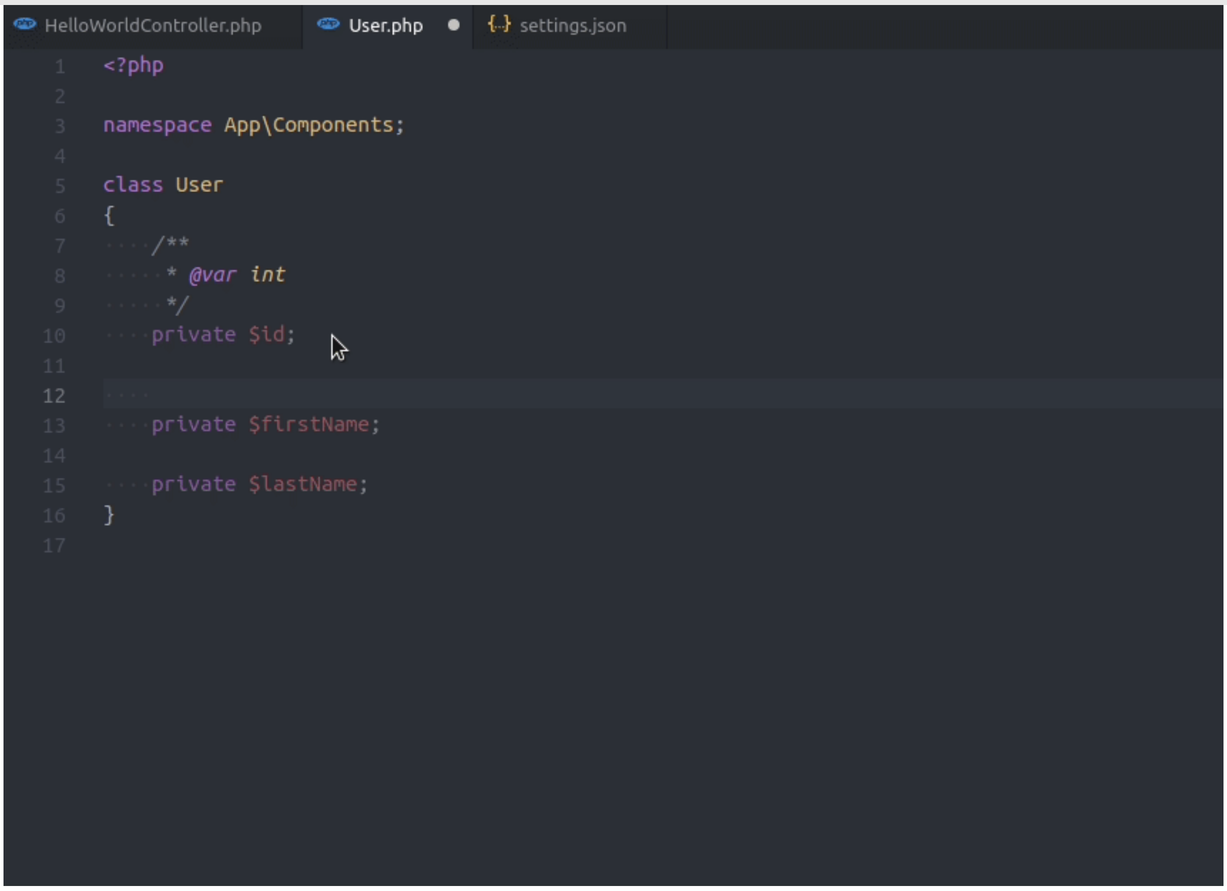
```
<?php
public function getAge(): int { }
```

@return int

Relevant suggestions when basic suggestions are disabled

## Enable autocomplete in comments / annotations

When you write PHP code, a good practice is to add PHPDoc annotations to make your code more understandable and to help your IDE provide you with relevant code suggestions. By default, VS Code doesn't suggest anything while writing annotations. To activate code suggestions in comments, open the `settings.json` file and add the following line:



```php
<?php

namespace App\Components;

class User
{
    /**
     * @var int
     */
    private $id;

    

    private $firstName;

    private $lastName;
}
```

Code suggestions in annotations

## Allow autocomplete in tests

By default, PHP Intelephense excludes Symfony test folders from indexing because the default exclude settings contains a too generic pattern:

```
"intelephense.files.exclude": [
  "**/.git/**",
  "**/.svn/**",
  "**/.hg/**",
  "**/bower_components/**",
  "**/vendor/**/{Test,test,Tests,tests}/**"
],
```

To enable suggestions for Symfony test classes, all you need to do is edit your `settings.json` file and add:

```
  "intelephense.files.exclude": [
    "**/.git/**",
    "**/.svn/**",
    "**/.hg/**",
    "**/CVS/**",
    "**/.DS_Store/**",
    "**/node_modules/**",
    "**/bower_components/**",
    "**/vendor/**/{Test,test,Tests,tests}/**/*Test.
  ],
```

As you can see, the last line is more specific than in the default settings.

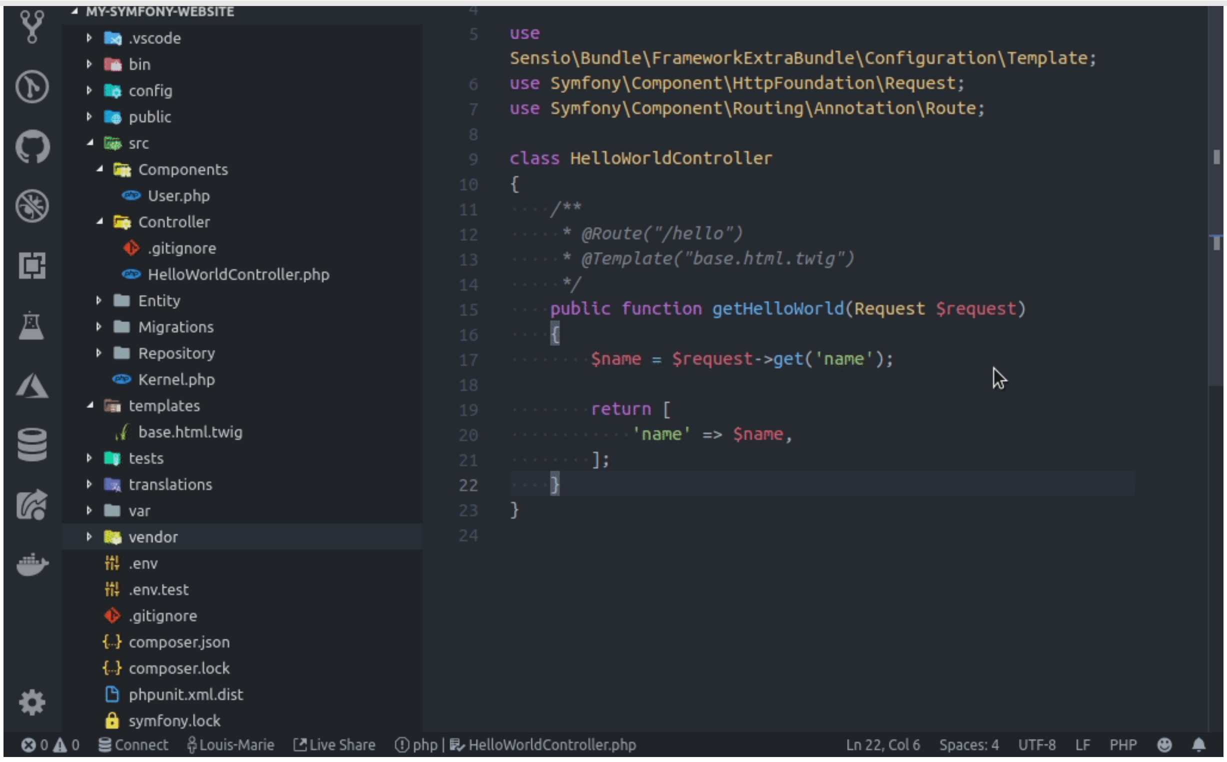
The result 😉:

```
{
}
```

Code completion enabled for tests

## Go to definition

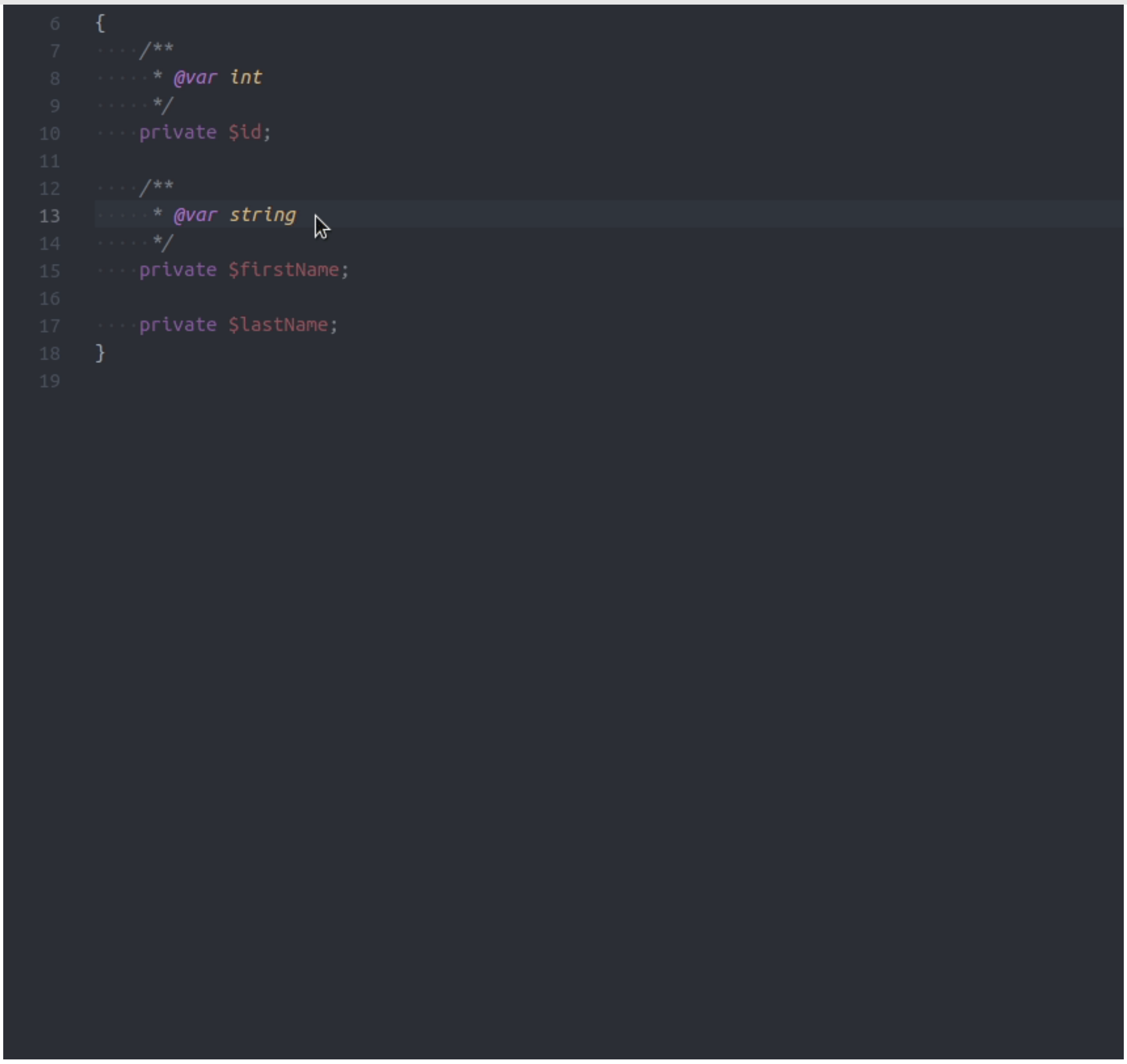
When you want to get more information about a function, you can Ctrl+Click on it and VS Code will open the line where the function is declared!



```
use
Sensio\Bundle\FrameworkExtraBundle\Configuration\Template;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\Routing\Annotation\Route;

class HelloWorldController
{
    /**
     * @Route("/hello")
     * @Template("base.html.twig")
     */
    public function getHelloWorld(Request $request)
    {
        $name = $request->get('name');

        return [
            'name' => $name,
        ];
    }
}
```

## Generate getters and setters for class properties

If you want VS Code to generate getters and setters for you, you can install **this extension**. Then, right-click on a class property and select *Insert PHP Getter & Setter*.



```
{
    /**
     * @var int
     */
    private $id;

    /**
     * @var string
     */
    private $firstName;

    private $lastName;
}
```

Generate getter and setter

## Debugging

Debugging your PHP code can be painful without a debugger. You can use `dump($myVar); die();` (😕) but it's not very convenient...

## Install Xdebug

Xdebug is not shipped with PHP, you need to install it on your development environment:

```
sudo apt install php-xdebug
```

Then run `sudo nano /etc/php/7.2/mods-available/xdebug.ini` and add the following lines:

```
xdebug.remote_enable = 1
xdebug.remote_autostart = 1
```

*Tip!* If your PHP project runs in a Docker container, you need also to add the following line to the `xdebug.ini` file, where `172.17.0.1` is the IP address of the `docker0` interface on your computer (on Mac OS you have to put `host.docker.internal` instead):

```
xdebug.remote_host = 172.17.0.1
```

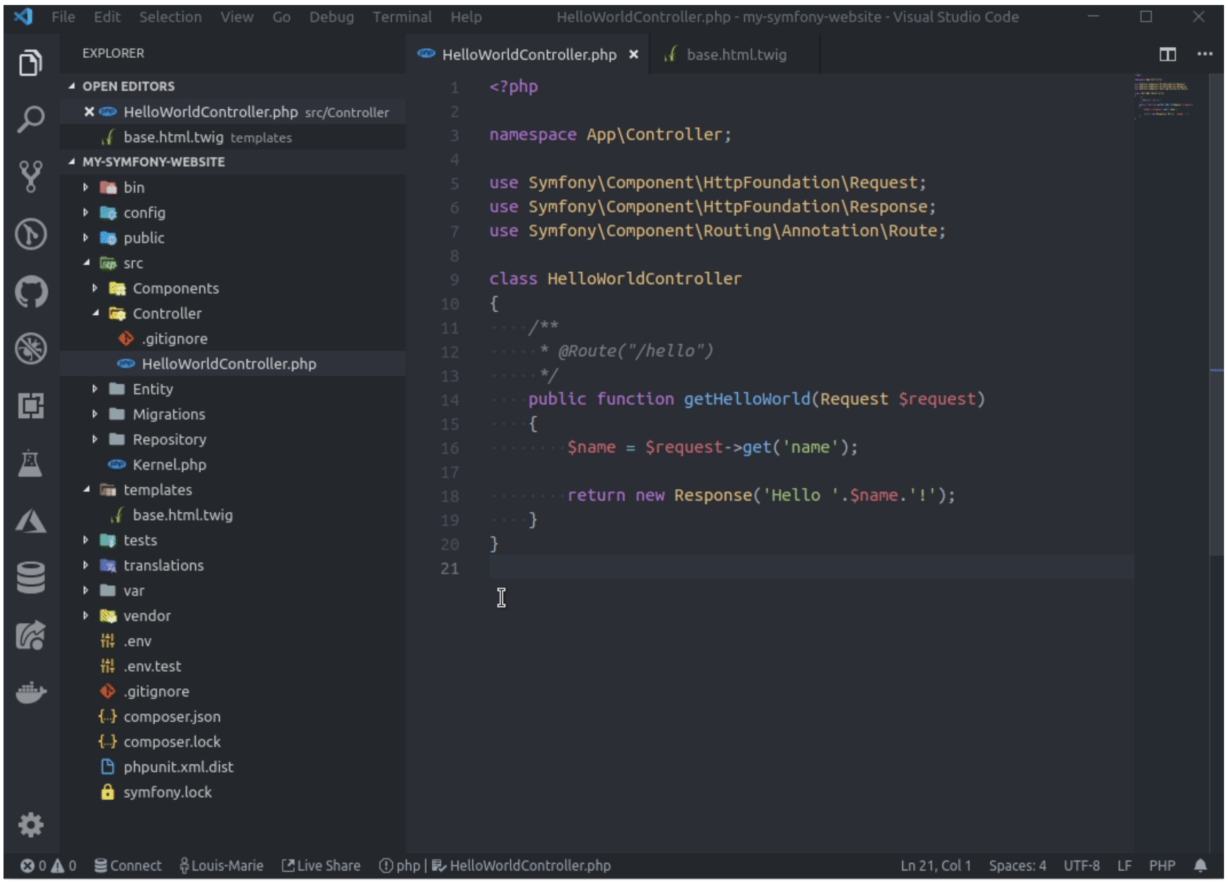
Check that Xdebug is successfully installed by running `php -v`. You should get something like:

```
PHP 7.2.19-0ubuntu0.18.04.1 (cli) (built: Jun  4
with Xdebug v2.6.0, Copyright (c) 2002-2018, I
```

## Configure VS Code debugger to listen for Xdebug

**To be able to use Xdebug in VS Code, you need to install the PHP debug extension.**

Then go to the debug tab in VS Code (accessible in the side menu) and click on *Add configuration.* Select *PHP*, and close the newly created `launch.json` file:



```php
<?php

namespace App\Controller;

use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;

class HelloWorldController
{
    /**
     * @Route("/hello")
     */
    public function getHelloWorld(Request $request)
    {
        $name = $request->get('name');

        return new Response('Hello '.$name.'!');
    }
}

```

Configure VS Code debugger to listen for Xdebug

```
"name": "Listen for XDebug",
...
"port": 9000,
"pathMappings": {
  "/path/to/app/in/docker": "${workspaceFolder}"
},
```

Your debugging environment is now ready! 🕷🚀

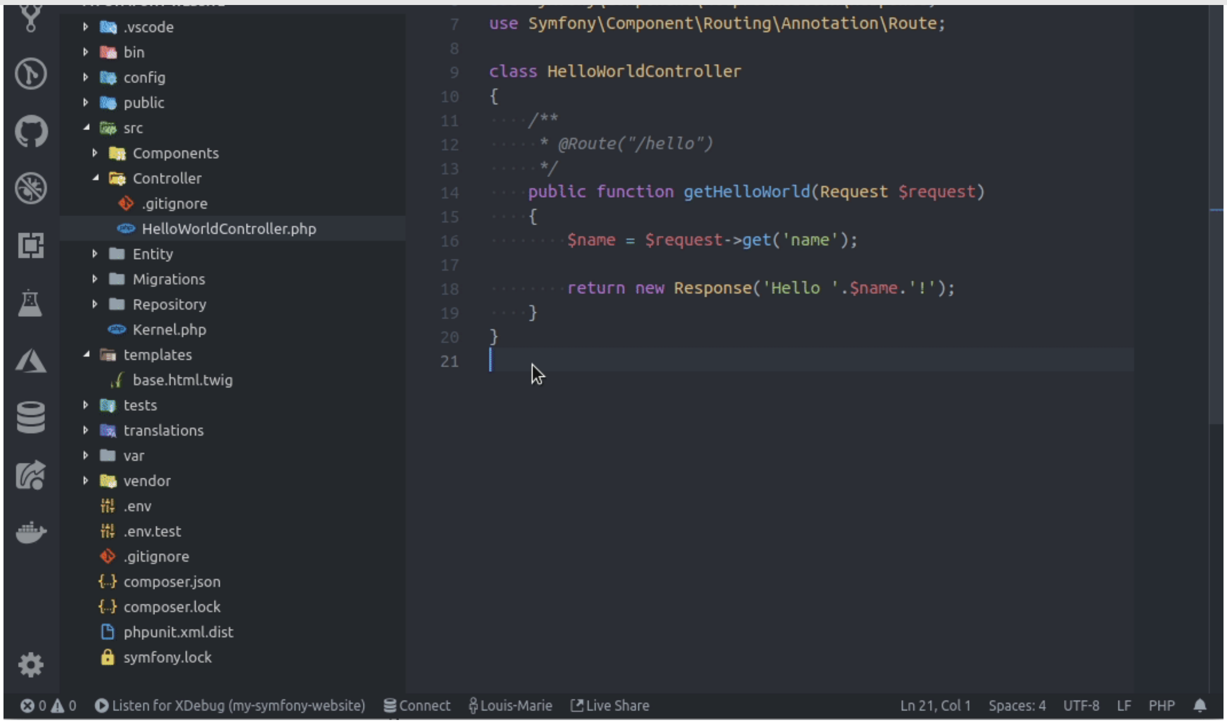
## Debug your code with VS Code

Let's debug our `HelloWorldController.php` to see which are the query parameters given by the user. You can use the following code for example:

```
<?php

namespace App\Controller;

use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;

class HelloWorldController
{
    /**
    {
        $name = $request->get('name');

        return new Response('Hello '.$name.'!');
    }
}
```

Press *F5* in VS Code to start the debugging session.

Uncheck the _Everything _option in the breakpoints list and put a breakpoint at line 18 by clicking left of the line number. Then go to **http://127.0.0.1:8000/hello?name=Louis**. You can now see the value of local variables by hovering the mouse pointer over them:



```
.vscode
bin
config
public
src
  Components
  Controller
    .gitignore
    HelloWorldController.php
  Entity
  Migrations
  Repository
  Kernel.php
templates
  base.html.twig
tests
translations
var
vendor
.env
.env.test
.gitignore
composer.json
composer.lock
phpunit.xml.dist
symfony.lock
```

```php
use Symfony\Component\Routing\Annotation\Route;

class HelloWorldController
{
    /**
     * @Route("/hello")
     */
    public function getHelloWorld(Request $request)
    {
        $name = $request->get('name');

        return new Response('Hello '.$name.'!');
    }
}

```

Request to **http://127.0.0.1:8000/hello?name=Louis** paused in debugger

You can do a lot of amazing things with the debugger 😃, like:

- adding or removing breakpoints
- stepping over, into or out
- watching variables
- evaluating expressions in the debug console

For more information about the power of VS Code debugger, you can visit **the official website**.

When you encounter an error, or when you use the Symfony debug toolbar or the profiler, you may want to open the desired file directly in your IDE with the cursor at the corresponding line.

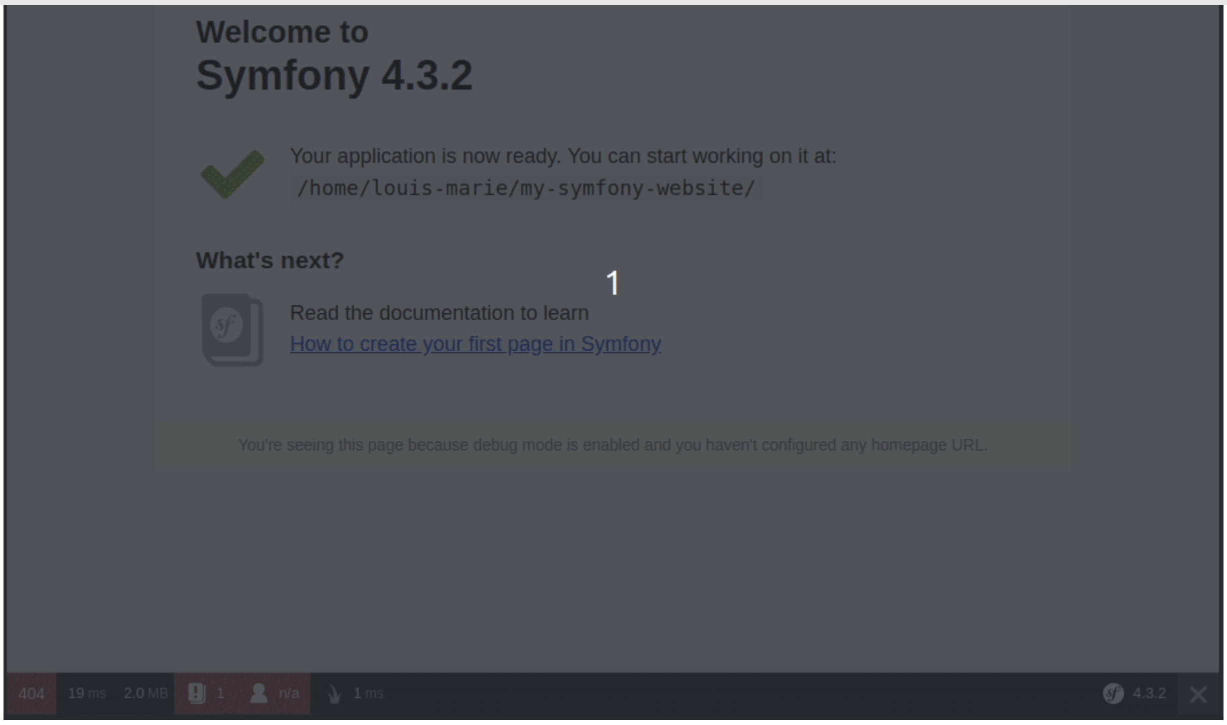
To be able to do that, you need to add the following line to your `/etc/php/7.2/mods-available/xdebug.ini`:

```
xdebug.file_link_format = vscode://file/%f:%l
```

*Tip!* If you run PHP in a Docker container, you can specify a path mapping between your host and your docker like this:

```
xdebug.file_link_format = 'vscode://file/%f:%l&/pa
```

Now, when an error occurs, you can click on it and go directly to the corresponding line in VS Code:


Welcome to
Symfony 4.3.2

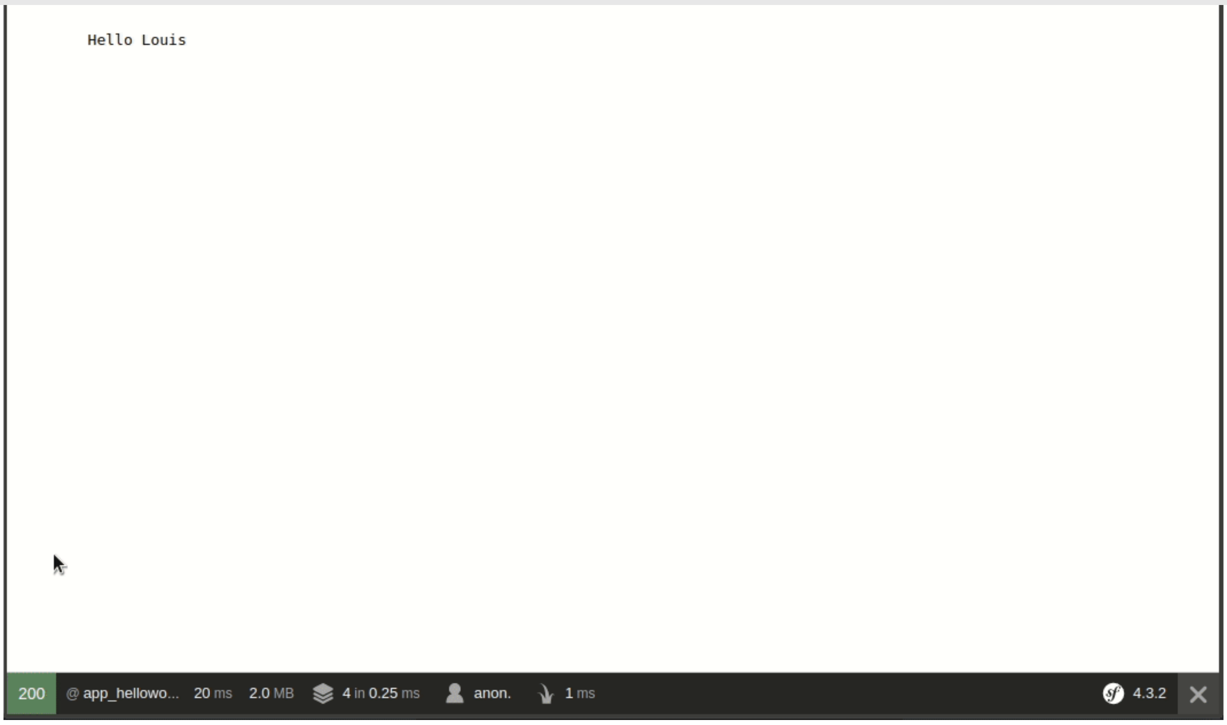
Your application is now ready. You can start working on it at:
`/home/louis-marie/my-symfony-website/`

What's next?

Read the documentation to learn
How to create your first page in Symfony

You're seeing this page because debug mode is enabled and you haven't configured any homepage URL.

When you visit a route, you can also jump to the corresponding controller by clicking on the debugging toolbar:



## Formatting & linting

Formatting and linting your code is a very good practice to keep it clean automatically, and to inform you about some errors.

### PHP CS Fixer

**The best and commonly used linter for PHP is PHP CS Fixer.** It provides a large number of rules to keep your code clean. You can visit **this website** for more information about the rules and the categories they belong to.

### Integrate PHP CS Fixer into VS Code

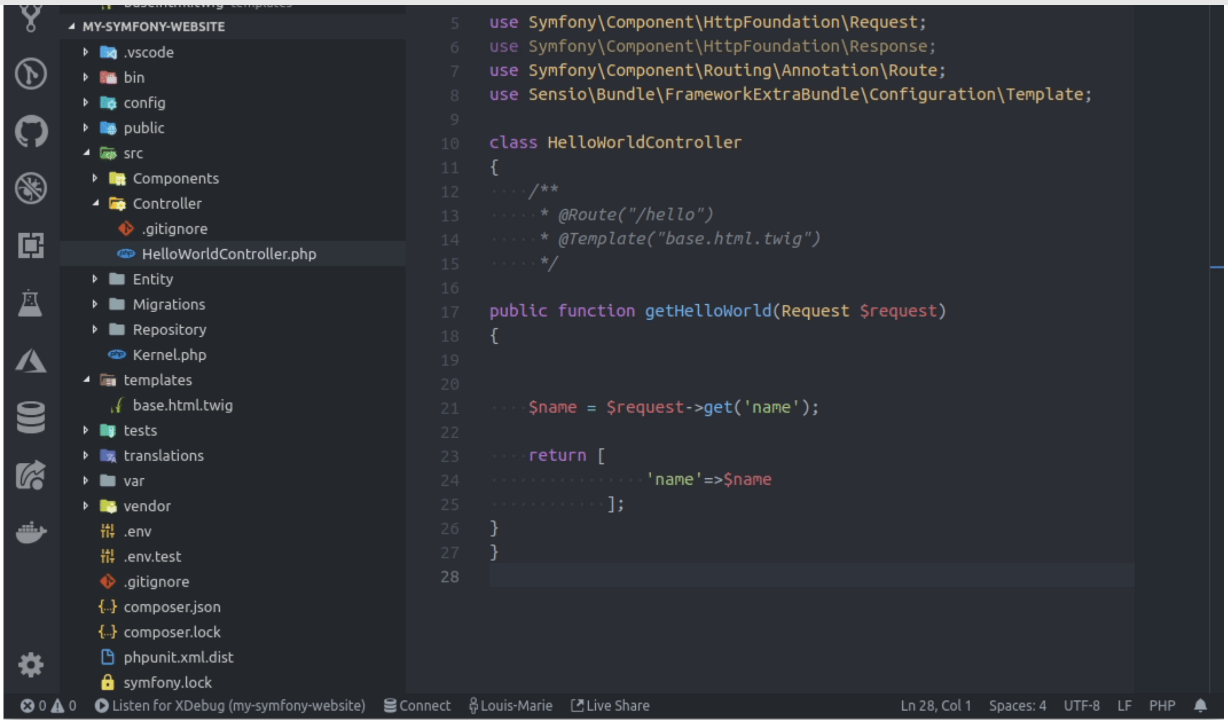
**To be able to auto-format your PHP files in VS Code,**

is *@PhpCsFixer*. It's a good point to start. To enable this set of rules in VS Code, open `settings.json` and add the following line:

```
"php-cs-fixer.rules": "@PhpCsFixer",
```

To auto-format your PHP files on save, you also need to add the following lines to your `settings.json`:

```
"[php]": {
  "editor.defaultFormatter": "junstyle.php-cs-fix
  "editor.formatOnSave": true
},
```



```php
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Template;

class HelloWorldController
{
    /**
     * @Route("/hello")
     * @Template("base.html.twig")
     */

public function getHelloWorld(Request $request)
{


    $name = $request->get('name');

    return [
                'name'=>$name
            ];
}
}

```

Auto-format on save with PHP CS Fixer

## PHP CS Fixer config file

If you want to disable or enable some specific linting rules, you can do it in a `.php_cs` file at the root folder of your project. If this configuration file is present, VS Code takes it into account and overrides the configuration found in `settings.json`. You can find more information about the `.php_cs` file in the **GitHub repository**. A simple config file to use *@PhpCsFixer* category and disable some rules could be:

```php
<?php
```

```
            '@PhpCsFixer' => true,
            'php_unit_internal_class' => false,
            'php_unit_test_class_requires_covers' =>
        ])
    ;
```

## Bonus: add Prettier to PHP CS Fixer

**Prettier for PHP** is a code formatter which makes some improvements that PHP CS Fixer does not do. It allows you, among other things, to configure a max line length and makes your code even cleaner.

To add Prettier for PHP to your PHP CS Fixer configuration, you can follow **these instructions**.

## In case of issues with format on save

Depending on your computer's speed, the length of your files and the number of rules you activate, linting a file on save can be slow, causing VS Code to refuse it. To fix this behavior, change the *formatOnSaveTimeout* in your `settings.json`:

```
"editor.formatOnSaveTimeout": 5000,
```

## Twig support

To enable emmet suggestions like in HTML files, add the following line to your `settings.json`:

```
"emmet.includeLanguages": { "twig": "html" },
```

## Database management

A good VS Code extension to manage your databases is **SQLTools**. Feel free to install it and manage your databases directly from your IDE!

Enjoy coding in PHP with VS Code 🚀 and feel free to give me your feedback! 😉

*Want to learn more about how Theodo can help bring your Symfony project to the next level?* ***Learn more about our team of Symfony experts.***

**Louis-Marie Michelin**

**Web Developer at Theodo**

**← Store your data in AWS Serverless architecture**

**Build a Real-Time Serverless Web Application with AWS →**

Theodo

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

**Mauricio Martins da Cunha** • 6 days ago

The only thing that I missed here is a good plugin to implement interfaces skeleton ...

1 | • Reply • Share ›

**Zecka** • 6 months ago

NIce guide thank you !

Any way to auto generate methods when implement interface ?

1 | • Reply • Share ›

**Andrea Zaccheroni** • 3 months ago

I'm Working with a php docker and both Intellisense and PHP CS Fixer tell me: executablePath not found, please check your settings.

| • Reply • Share ›

**Amir Gharajedaghi** • 6 months ago

But go to definition not work ?

| • Reply • Share ›

**Amir Gharajedaghi** • 6 months ago

very good.

| • Reply • Share ›

**Tetuaoro Lenoir** • 6 months ago

Hehe working with plugins is so fun now

| • Reply • Share ›

**Paolo Minervino** • 6 months ago

Nice guide. Thank you. Bravo.

| • Reply • Share ›

**Gaurav Singh** • 7 months ago

Theodo