Debug PHP In VSCode With XDebug

Posted on [**July 20, 2018**](https://www.codewall.co.uk/debug-php-in-vscode-with-xdebug/) by [**Dan Englishby**](https://www.codewall.co.uk/author/heyksj1/)[**1 Commenton Debug PHP In VSCode With XDebug**](https://www.codewall.co.uk/debug-php-in-vscode-with-xdebug/#comments)

VSCode is great, the fact that it can work with a vast amount of languages help’s it cater for everyone’s taste. One special tool that comes with VSCode is the Debug feature and when configured correctly, is a dream to use. If you’ve previously worked with Visual Studio 2017 for example, you will know what I mean. This guide will go through how to set-up the PHP xDebug module to work in tandem with VSCode, completely enabling full-on functional debugging across your application.

Getting the basics ready

To complete this tutorial you will need an installation of **[VSCode](https://code.visualstudio.com/" \t "_blank)** and as a bare minimum [**PHP 7.0**](http://php.net/downloads.php). Of course, you will need a web server to run the PHP too, so make sure this is in place also. Once these are ready, make note of your PHP module directory, for example, mine is set to ‘**C:\xampp\php\ext**‘. This is were the XDebug module will reside.

**Configuring PHP to use XDebug**

**Step 1**

Create an index.php file in the root directory of your web server, and add the following code.

<?php

phpinfo();

?>

Copy

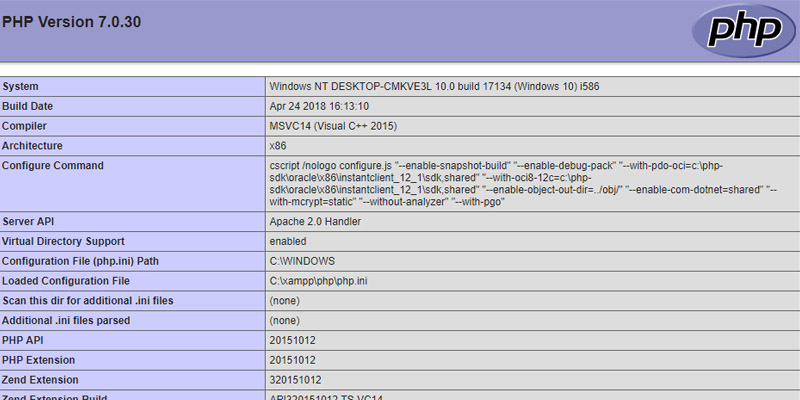
Run the index.php file in a browser, preferably, Google Chrome. This will render the following information as shown in the image below –

**[[](https://bit.ly/2WBgXm1)](https://bit.ly/2WBgXm1" \t "_blank)**

**[Angular 7 - The Complete Guide](https://bit.ly/2WBgXm1" \t "_blank)**

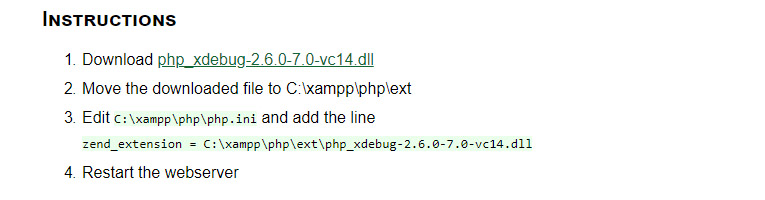
[Master Angular (Angular 2+, incl. Angular 7) and build awesome, reactive web apps with the successor of Angular.js](https://bit.ly/2WBgXm1" \t "_blank)

[Average Rating 4.6/5  from over 77,000 students!](https://bit.ly/2WBgXm1" \t "_blank)



Once this is loaded up, follow these sub-steps.

1. Highlight all of the information on the page and copy it.
2. Go to the **[XDebug Wizard tool](https://xdebug.org/wizard.php" \t "_blank)**.
3. Paste the information into the Text Box input on the wizard tool page.
4. Locate and click the ‘Analyse my phpinfo() output’ button.
5. Download the provided module under Instructions, number 1. (See image below)



Follow the remaining instructions in the image above but remember, in this example, the directory path may be different to your own. Do not forget to **restart the web server** after adding the module to the ext directory.

**Step 2**

Let’s go ahead and configure the php.ini file to use the new xDebug module and set-up the extra settings to work with VSCode.

* Locate and open the php.ini configuration file, this will be inside the PHP root directory, for example, ‘C:\xampp\php\’.
* Add the settings below to the very bottom of the php.ini file.
* **Important:**Make sure that the zend\_extension matches the one you have downloaded and residing in your php/ext folder.
* **Also Important:**Enabling remote\_enable & remote\_autostart must not be missed, otherwise, xDebug will not execute if either the browser or as singular script launch.
* Save it
* Restart the web server again.

[xDebug]

zend\_extension="php\_xdebug-2.6.0-7.0-vc14.dll"

xdebug.remote\_enable = 1

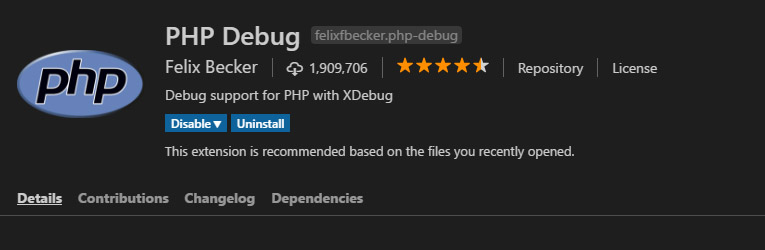
xdebug.remote\_autostart=on

Copy

**Configuring VSCode to use PHP XDebug**

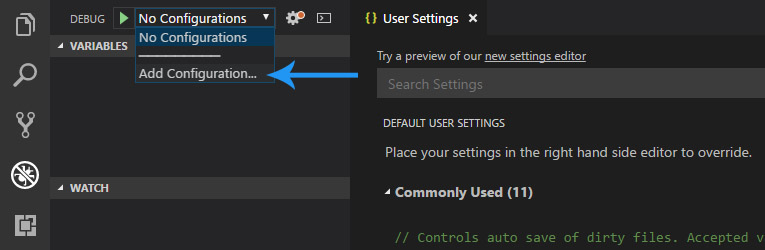
**Step 1**

Install the PHP Debug extension by Felix Becker. This can be done by clicking the extensions tab in VSCode and searching for ‘PHP Debug’.

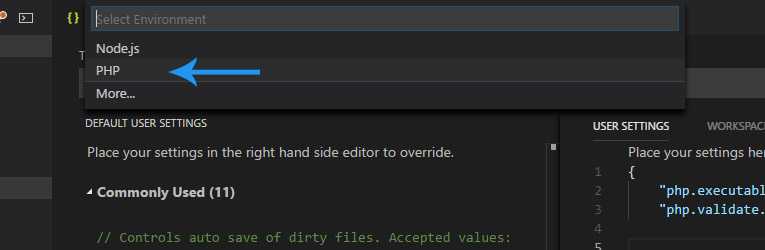


**Step 2**

Then, click on the Debug tab on the left-hand pane, click the debug drop-down and select Add Configuration.

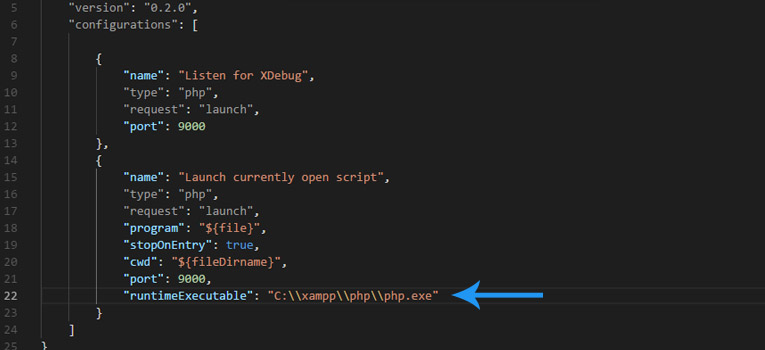


Then, click the ‘PHP’ option from the drop-down that pops out.



After this point, VSCode will of created a launch.json file within your root directory inside the VSCode folder. There is one more step to take to make sure that the script execution works correctly.

Specify the **runtimeExecutable**which is the path to your PHP.exe interpreter. For example. mine looks like the following.



Now go ahead and **save** the launch.json file.

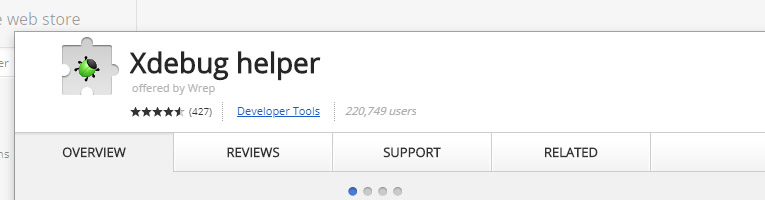
**Step 3**

After following steps 1 and 2, you will be able to use the ‘Launch currently open script’ option, which will do exactly what it says on the tin. You can debug a single script without having to worry about a browser.

Now, if we want to be able to debug a script as a proper user, for example, debugging using a browser. This is always worth setting up because it’s extremely handy when debugging Get & Post to name a few.

What we need is an xDebug Helper, which is a Google Chrome extension available to use for free. Find the extension by going to the Google Chrome Web Store. See the following steps –

1. Goto the following URL [**https://chrome.google.com/webstore/category/extensions**](https://chrome.google.com/webstore/category/extensions)
2. Search for ‘xDebug Helper’
3. Install the extension (It has a green ladybug in the logo) – See image below.



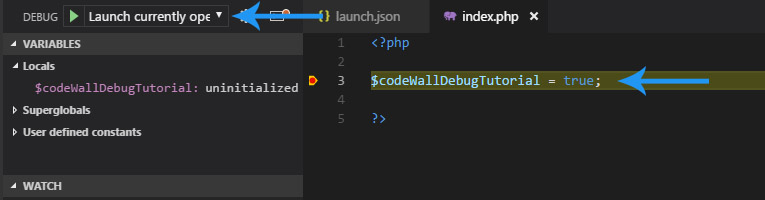
That is it, your VSCode is now prepared to debug your code.

**How to use**

**Launch currently open script**

Ok, so now both debug methods are set-up, how do we use them? It’s pretty simple. First of all, let us go through using the ‘Launch currently open script’ method.

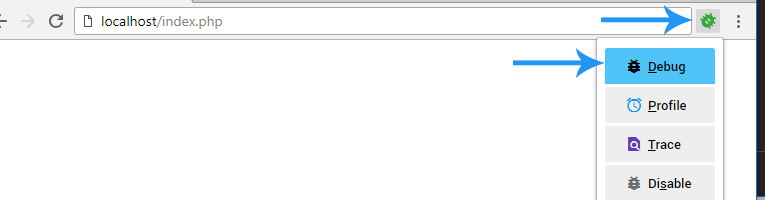
1. Go to any .php file, and add some breakpoints to your code (Shortcut is F9 on the keyboard).
2. Make sure that ‘Launch currently open script’ is already selected in the Debug drop down in the top left of the debug window.
3. Press F5 to start the debug process.
4. You will see the debugger break at the first breakpoint, see the image below.



**Listen for xDebug**

Now, let’s go through the ‘Listen for xDebug method which takes the browser into account.

1. Open a .php file, for this example the index.php file in the root directory.
2. Add some code and add some breakpoints. Use F9 shortcut key to add breakpoints.
3. Change the Debug select option to ‘Listen for xDebug’.
4. Press F5 to start the debugger.
5. Open Chrome and navigate to your index.php file, in my case its http://localhost/index.php
6. Click the new XDebug Helper extension and click the Debug option.
7. You will notice the helper icon will now turn green (See image below)
8. Finally, refresh the page in the browser, and VSCode will react and start the debugging process.
9. Go back into VSCode and you will see that it has stopped at the breakpoint.



**Summary**

Although this tutorial is pretty long-winded, you only have to do it once. I really advise the installation of XDebug Helper as it transforms debugging into another level. It’s free to use and invaluable.

If you struggle with this tutorial, or you get a different result to what has been stated in this article, please feel free to leave a comment.

SHARE ON



**The Author: Dan Englishby**

[**View all posts by Dan Englishby**](https://www.codewall.co.uk/author/heyksj1/)

Posted in [**PHP**](https://www.codewall.co.uk/category/php/)

Post navigation

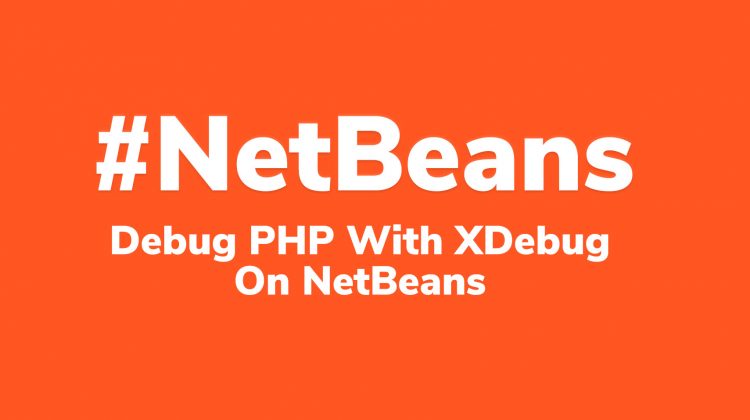
[The Best JavaScript Data Visualization & Charting Libraries 2018](https://www.codewall.co.uk/the-best-javascript-data-visualization-charting-libraries/)

[44 Best Accounts For Developers To Follow On Twitter in 2018/2019](https://www.codewall.co.uk/accounts-for-developers-to-follow-on-twitter/)

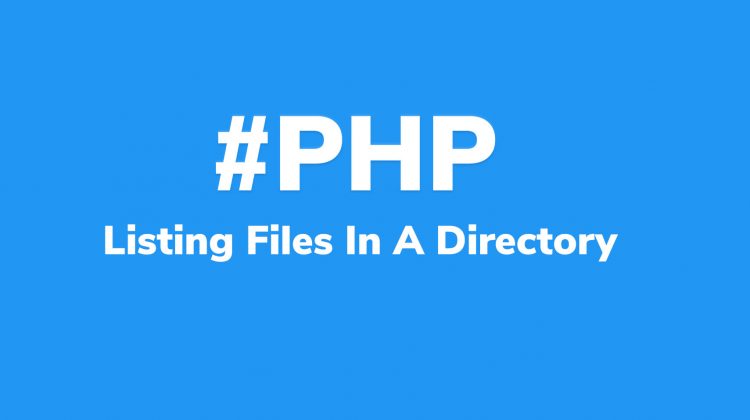
**You May Also Like**

* [](https://www.codewall.co.uk/free-laravel-artisan-guis/)

[Free Laravel Artisan GUI’s](https://www.codewall.co.uk/free-laravel-artisan-guis/)

* [](https://www.codewall.co.uk/debug-php-with-xdebug-on-netbeans/)

[Debug PHP With XDebug On NetBeans](https://www.codewall.co.uk/debug-php-with-xdebug-on-netbeans/)

* [](https://www.codewall.co.uk/list-files-in-directory-with-php/)

[List Files In Directory With PHP](https://www.codewall.co.uk/list-files-in-directory-with-php/)

One thought on “Debug PHP In VSCode With XDebug”

1. https://secure.gravatar.com/avatar/?s=45&d=identicon&r=gJohnno says:

[*July 29, 2018 at 7:53 pm*](https://www.codewall.co.uk/debug-php-in-vscode-with-xdebug/#comment-57)

Thanks for this great tutorial, seriously, been trying to work how to get this set up for days and just came across this article! I previously used Netbeans and that was pretty easy to get working. I actually Hadn’t heard of the xdebug helper either so that’s a bonus!

[Reply](https://www.codewall.co.uk/debug-php-in-vscode-with-xdebug/#comment-57)