Server-side
Web Development

Unit 03. Debugging PHP

Fidel Oltra, Ricardo Sánchez

IES Jaume II El Just
Tavernes de la Valldigna
Departament d'Informàtica
Curs 2024-25

Debugging is a very important part of developing. In this tutorial we will learn how to debug PHP applications.

1 Install Xdebug on Xubuntu

The first step is to install the **Xdebug** debugger in Xubuntu. Open a terminal and write:

```
sudo apt update
sudo apt install php-xdebug
sudo systemctl daemon-reload
```

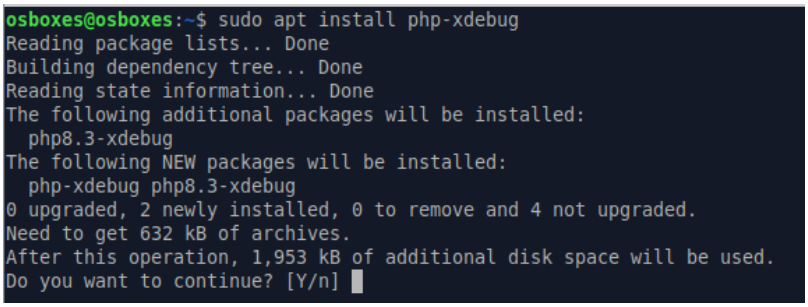


Figure 1: Xdebug installation

Once installed, add the following lines to the end of your `php.ini`:

```
[xdebug]
xdebug.mode=develop,debug
xdebug.discover_client_host=1
xdebug.client_port = 9003
xdebug.start_with_request=yes
```

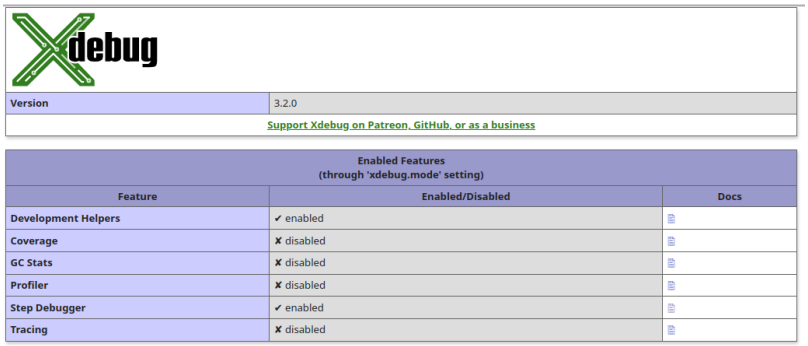
And restart Apache:

```
sudo systemctl restart apache2
```

To verify the installation, write a simple program to print the output of **phpinfo()** function as shown below:

```
# info.php
<?php
echo phpinfo();
```

Load the test page in a web browser and scroll down until you find the Xdebug configuration:



Version	3.2.0
Support Xdebug on Patreon, GitHub, or as a business	

Enabled Features (through 'xdebug.mode' setting)		
Feature	Enabled/Disabled	Docs
Development Helpers	✔ enabled	
Coverage	✘ disabled	
GC Stats	✘ disabled	
Profiler	✘ disabled	
Step Debugger	✔ enabled	
Tracing	✘ disabled	

Figure 2: Xdebug configuration in phpinfo()

Make sure that the “Step Debugger” section is enabled. If not, check the step in which you have modified the php.ini file.

2 Install Xdebug on Windows

First, show the output of **phpinfo()** in Xampp:

Figure 3: phpinfo() in Xampp

The output should be similar to:

PHP Version 8.2.12

System	Windows NT ARGOSWIN 10.0 build 22631 (Windows 11) AMD64
Build Date	Oct 24 2023 21:10:40
Build System	Microsoft Windows Server 2019 Datacenter [10.0.17763]
Compiler	Visual C++ 2019
Architecture	x64
Configure Command	cscript /nologo /e:jscript configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=..\..\..\..\instantclient\sdk,shared" "--with-oci8-19=..\..\..\..\instantclient\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	*no value*
Loaded Configuration File	C:\xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)

Figure 4: phpinfo() output

Now, copy the content and paste it within the input box of Xdebug Installation Wizard:

Installation Wizard

This page helps you finding which file to download, and how to configure PHP to get Xdebug running. Please paste the **full** output of phpinfo() (either a copy & paste of the HTML version, the HTML source or `php -i` output) and submit the form to receive tailored download and installation instructions.

```
Other Contributors      Previously active authors, editors and other
contributors are listed in the manual.
PHP Quality Assurance Team
Ilia Alshanetsky, Joerg Behrens, Antony Dovgal, Stefan Esser, Moriyoshi Koizumi,
Magnus Maatta, Sebastian Nohn, Derick Rethans, Melvyn Sopacua, Pierre-Alain
Joye, Dmitry Stogov, Felipe Pena, David Soria Parra, Stanislav Malyshev, Julien
Pauli, Stephen Zarkos, Anatol Belski, Remi Collet, Ferenc Kovacs
Websites and Infrastructure team
PHP Websites Team       Rasmus Lerdorf, Hannes Magnusson, Philip Olson, Lukas
Kahwe Smith, Pierre-Alain Joye, Kalle Sommer Nielsen, Peter Cowburn, Adam
Harvey, Ferenc Kovacs, Levi Morrison
Event Maintainers       Damien Seguy, Daniel P. Brown
Network Infrastructure  Daniel P. Brown
Windows Infrastructure  Alex Schoenmaker
PHP License
This program is free software; you can redistribute it and/or modify it under
the terms of the PHP License as published by the PHP Group and included in the
distribution in the file: LICENSE

This program is distributed in the hope that it will be useful, but WITHOUT ANY
WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.

If you did not receive a copy of the PHP license, or have any questions about
PHP licensing, please contact license@php.net.
```

The information that you upload will not be stored. The script will only use a few regular expressions to analyse the output and provide you with instructions. You can see the code **here**.

Analyse my phpinfo() output

Figure 5: Xdebug installation wizard

Click on the **Analyse my phpinfo() output** button to start analyzing it. It will show whether Xdebug is installed and also shows the steps to install or update the most recent version of Xdebug for PHP:

Installation Wizard

Summary

- **Xdebug installed:** no
- **Server API:** Apache 2.0 Handler
- **Windows:** yes
- **Compiler:** MS VS16
- **Architecture:** x64
- **Zend Server:** no
- **PHP Version:** 8.2.12
- **Zend API nr:** 420220829
- **PHP API nr:** 20220829
- **Debug Build:** no
- **Thread Safe Build:** yes
- **OPcache Loaded:** no
- **Configuration File Path:** no value
- **Configuration File:** C:\xampp\php\php.ini
- **Extensions directory:** C:\xampp\php\ext

Instructions

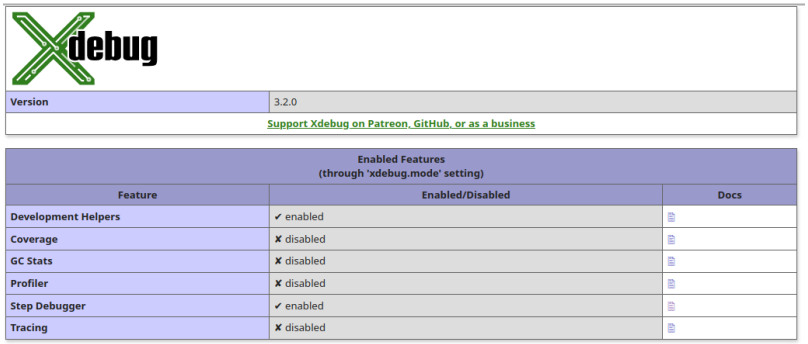
1. Download php_xdebug-3.3.2-8.2-vs16-x86_64.dll
2. Move the downloaded file to C:\xampp\php\ext, and rename it to `php_xdebug.dll`
3. Update `C:\xampp\php\php.ini` and add the line: `zend_extension = xdebug`
4. Restart the Apache Webserver

Figure 6: Output of the Xdebug installation wizard

Scroll down to the instructions section and follow the steps. In our case, the steps are:

1. Download the required `dll`.
2. Move the downloaded file to `C:\xampp\php\ext`, and rename it to `php_xdebug.dll`
3. Update `C:\xampp\php\php.ini` and add the line: `zend_extension = xdebug`
4. To enable step debugging, add to the `php.ini` file the line: `xdebug.mode = debug`
5. Restart the Apache server

Reload the test page and scroll down until you find the Xdebug configuration:



Version	3.2.0
Support Xdebug on Patreon, GitHub, or as a business	

Enabled Features (through 'xdebug.mode' setting)		
Feature	Enabled/Disabled	Docs
Development Helpers	✔ enabled	
Coverage	✘ disabled	
GC Stats	✘ disabled	
Profiler	✘ disabled	
Step Debugger	✔ enabled	
Tracing	✘ disabled	

Figure 7: Xdebug configuration in phpinfo()

3 Debugging in VS Code

If you have already installed the **PHP Tools** extension, it comes with a debugging tool. Otherwise, you can install the **PHP Debug** extension of **Xdebug**.

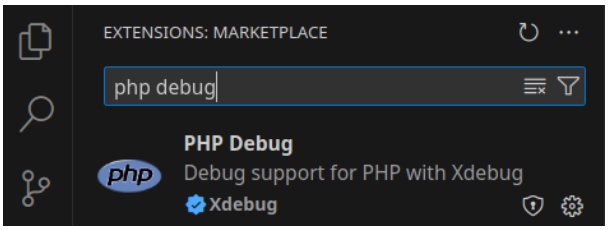


Figure 8: PHP Debug

Create a script to check the debugging process, something like:

```
<?php
$array = [0,5,10,15,20];
```

```
foreach ($array as $value) {
    $v = $value;
    echo "$v <br>";
}
```

Save it as `testdebug.php`.

We need to configure the Visual Studio Code to debug our code. Click the Run Icon on the Activity Bar on the left side to start configuring for xdebug. Now, click the create a **launch.json** file to configure the project for debugging. It will create the next file:

```
.vscode > {} launch.json > Launch Targets > {} Listen for Xdebug
1  {
2      // Use IntelliSense to learn about possible attributes.
3      // Hover to view descriptions of existing attributes.
4      // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
5      "version": "0.2.0",
6      "configurations": [
7          {
8              "name": "Launch built-in server and debug",
9              "type": "php",
10             "request": "launch",
11             "runtimeArgs": [
12                 "-S",
13                 "localhost:9003",
14                 "-t",
15                 "."
16             ],
17             "port": 9003,
18             "serverReadyAction": {
19                 "action": "openExternally"
20             }
21         },
22         {
23             "name": "Debug current script in console",
24             "type": "php",
25             "request": "launch",
26             "program": "${file}",
27             "cwd": "${fileDirname}",
28             "externalConsole": false,
29             "port": 9003
30         },
```

Figure 9: launch.json for PHP Debug

Go to the element with the name “Listen for Xdebug” and add the lines:

```
"pathMappings": {
    "/var/www/html": "${workspaceRoot}"
}
```

> If you are using Windows or XAMMP, adjust the *pathMappings* value to your server root.

Don’t forget the comma before `pathMappings`!:

Now, add a breakpoint in the line 5 (`$v = $value`) of your `testdebug.php` file (by clicking on the left of the number) and start the debugger with the configuration “Listen for Xdebug”:

```
{
    "name": "Listen for Xdebug",
    "type": "php",
    "request": "launch",
    "port": 9003,
    "pathMappings": {
        "/var/www/html": "${workspaceRoot}"
    }
}
```

Figure 10: launch.json for PHP Debug, changes

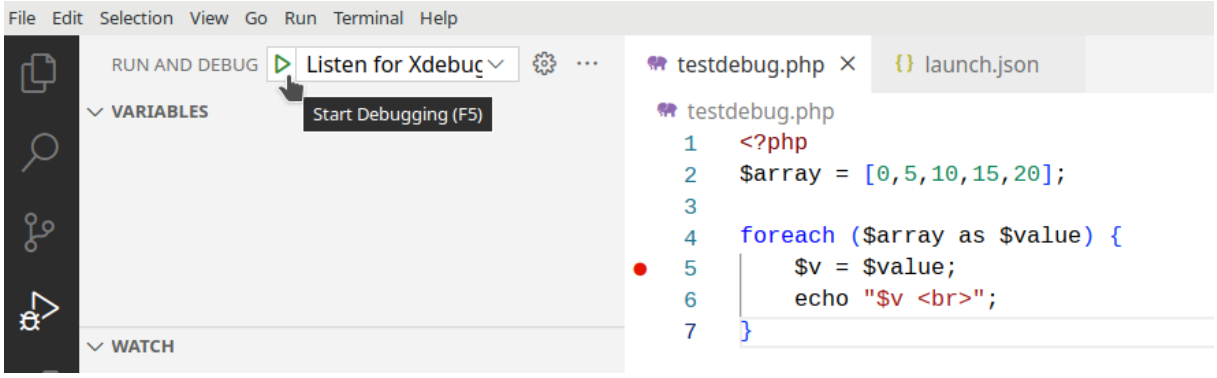


Figure 11: Start debugging in VSCode

Go to your browser and type `localhost/testdebug.php`. The debugger bar on VSCode should now show the debugger options (continue, step over, etc). You could also see the variables values on the left panel:

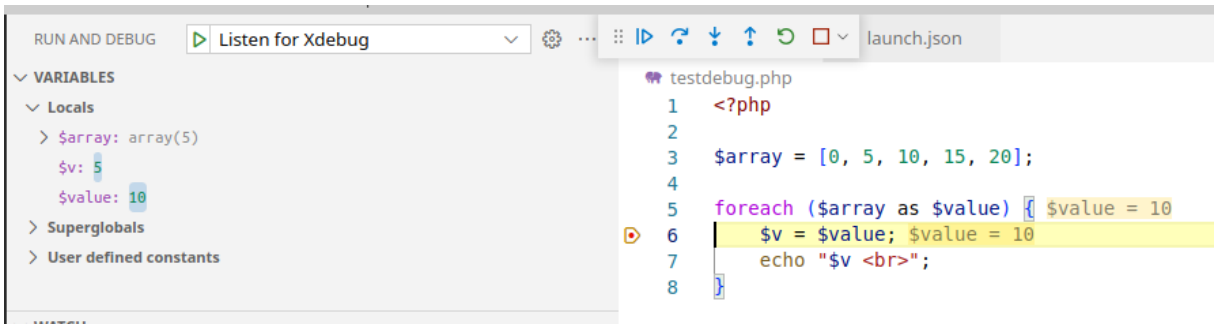


Figure 12: Debugging in VSCode

You can also start the debugger with the “Debug current script in console” configuration. It will show the output step by step in the VSCode console.

4 Debugging in PhpStorm

If we have installed Xdebug already, make a new project and copy or write the content of the previous `testdebug.php` script.

Create a new breakpoint in the line 5.

To start debugging, click on the Debug button. It will open the debug console where you can run the program step by step from the breakpoint you've created.

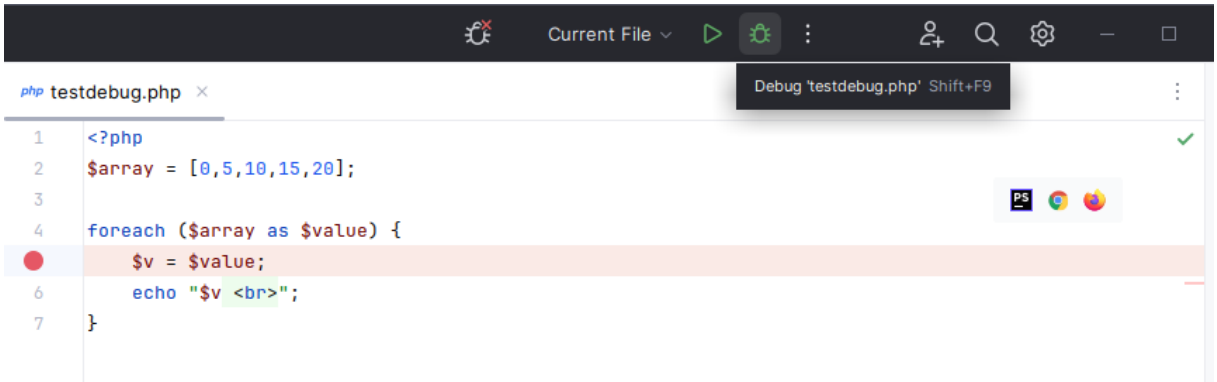


Figure 13: Debugging in PhpStorm

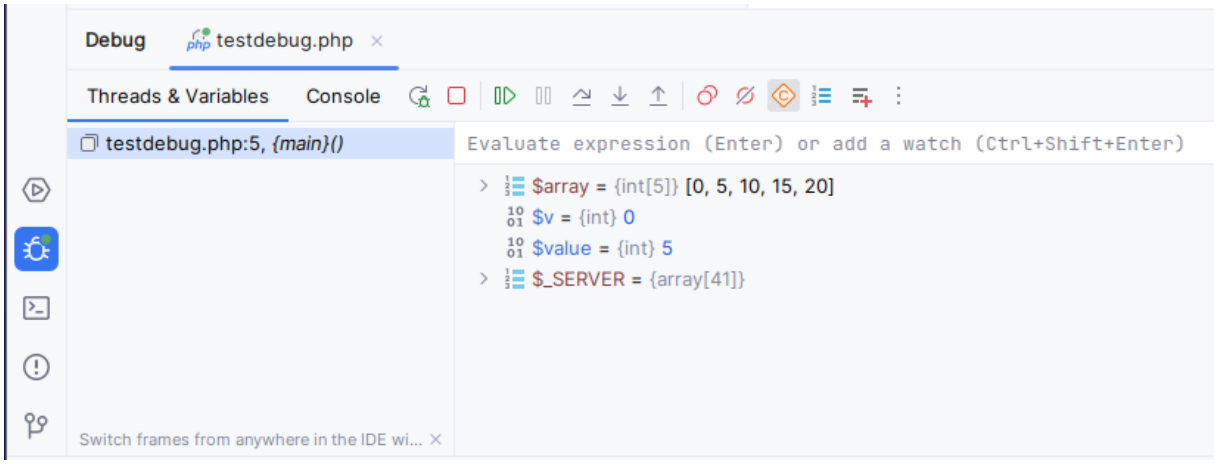


Figure 14: Debug console

> PhpStorm automates the process of getting Xdebug up and running. On your first attempt to run a debugging session without a debugger installed, the IDE will prompt you to download and install the relevant version of Xdebug.

If you want to debug interactively a web app, toggle the `Start Listening for PHP Debug Connections` button or choose the same option in the Run menu. You need to have the browser's add-on installed on your browser.

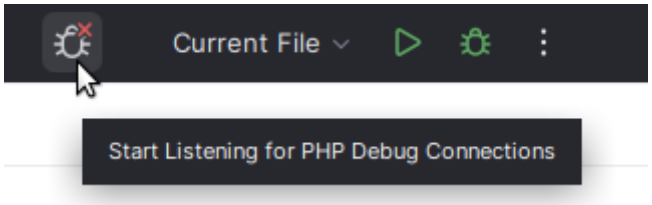


Figure 15: Start Listening for PHP Debug Connections

Then, open your app in the browser. It will appear the incoming connection from Xdebug dialog in PhpStorm. Then you can debug your app and see the result at the same time in the web browser.

5 Install the browser's add-on

We can install an add-on to our browser in order to enhance the debugging process. In Firefox or Google Chrome search for the Xdebug Helper add-on and install it:

Figure 16: Xdebug Helper for Firefox

To activate it, click on the bug in the address bar and change it to green:

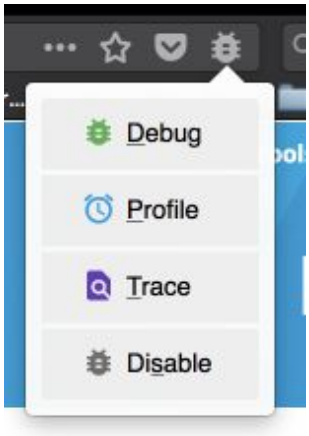


Figure 17: Xdebug Helper