Introduction
○○○○○

Installation
○○○
○○

Profiler
○
○

Tracer
○

PHP functions

References
○

Debug PHP with Xdebug
with examples for Vagrant with Debian / Ubuntu

Christian Pojoni

`www.secret-source.eu`

Mar 2019

Introduction
○○○○○

Installation
○○○
○○

Profiler
○

Tracer

PHP functions
○

References
○

Contents

Why use Xdebug

Common Approach

Listing 1: Manual Debugging

```php
var_dump($thing);
// or
print_r($data);
// or
error_log(print_r($someArray, true));
// or
echo __METHOD__ . ' ' . __LINE__;
```

Why use Xdebug

Common Approach

Listing 2: Manual Debugging

```
var_dump($thing);
// or
print_r($data);
// or
error_log(print_r($someArray, true));
// or
echo __METHOD__ . ' ' . __LINE__;
```

Remove that again

don't commit these lines into VCS

Introduction
○●○○○

Installation
○○○
○○

Profiler
○
○

Tracer

PHP functions

References
○

Traditional Way

1. insert outputs
2. reload page
3. look for output
4. change some code

Traditional Way

1 insert outputs
2 reload page
3 look for output
4 change some code
5 repeat 1. until error is fixed

Introduction
○●○○○

Installation
○○○
○○

Profiler
○

Tracer
○

PHP functions

References
○

Traditional Way

1. insert outputs
2. reload page
3. look for output
4. change some code
5. repeat 1. until error is fixed
6. clean up outputs
7. load final version of the page

Introduction
○○●○○

Installation
○○○
○○

Profiler
○

Tracer
○

PHP functions

References
○

Xdebug Way

1. add breakpoints

Introduction
○○●○○

Installation
○○○
○○

Profiler
○

Tracer
○

PHP functions

References
○

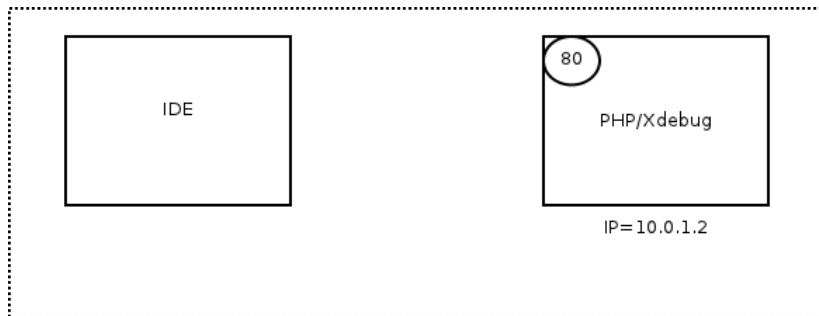# Xdebug Way

1. add breakpoints
2. reload page
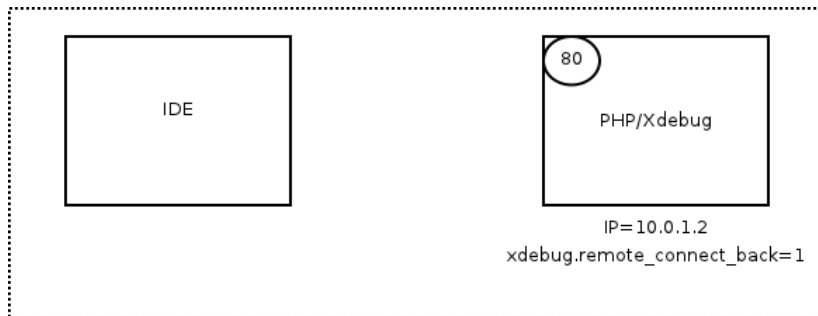
Xdebug Way

1. add breakpoints
2. reload page
3. step into the code

Advantages

- Faster development time
- Better understanding of what the code is doing
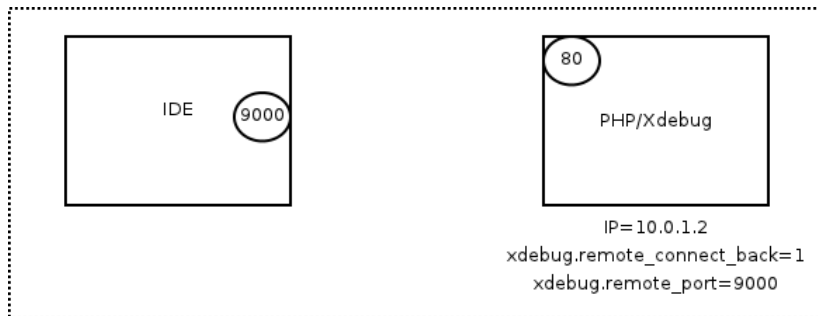- Does not require code changes
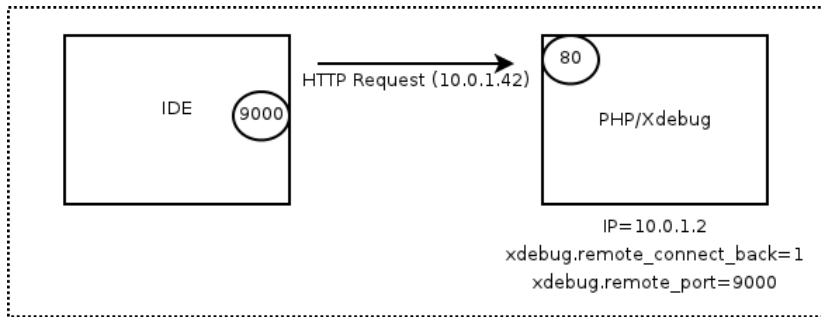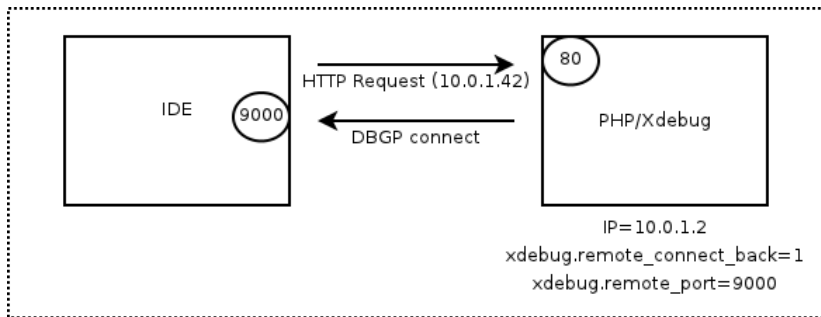
Introduction
○○○○●

Installation
○○○
○○

Profiler
○

Tracer
○

PHP functions

References
○

How it works

Introduction
○○○○●

Installation
○○○
○○

Profiler
○

Tracer
○

PHP functions

References
○

How it works

Introduction
○○○○●

Installation
○○○
○○

Profiler
○

Tracer
○

PHP functions

References
○

How it works

Introduction
○○○○●

Installation
○○○
○○

Profiler
○

Tracer
○

PHP functions
○

References
○

How it works

Introduction
○○○○●

Installation
○○○
○○

Profiler
○

Tracer

PHP functions

References
○

How it works

Introduction
○○○○●

Installation
○○○
○○

Profiler
○

Tracer
○

PHP functions

References
○

How it works

Introduction
ooooo

Installation
●oo
oo

Profiler
o
o

Tracer

PHP functions

References
o

Client

Browser extensions

- Xdebug Helper for Firefox
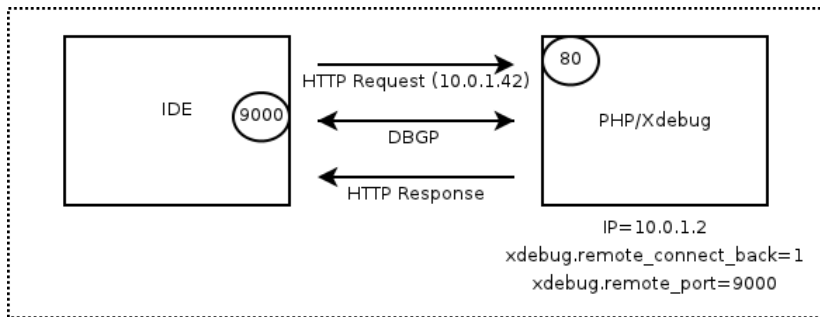- Xdebug Helper for Chrome
- Xdebug Toggler for Safari
- Xdebug launcher for Opera

Browser extensions

- Xdebug Helper for Firefox
- Xdebug Helper for Chrome
- Xdebug Toggler for Safari
- Xdebug launcher for Opera

or

- HTTP GET/POST variable XDEBUG_SESSION_START
- Set a Cookie in the header XDEBUG_SESSION

VS Code Configuration Part 1

Install Extension: PHP Debug

Introduction
00000

Installation
○●○
○○

Profiler
○

Tracer
○

PHP functions

References
○

Client

VS Code Configuration Part 1

Install Extension: PHP Debug

Introduction
○○○○○

Installation
○●○
○○

Profiler
○

Tracer
○

PHP functions

References
○

Client

VS Code Configuration Part 1

Install Extension: PHP Debug

Listing 5: launch.json

```
{
    "name": "Listen for XDebug",
    "type": "php",
    "request": "launch",
    "port": 9000,
    "pathMappings": {
        "/var/www/html": "${workspaceRoot}"
    }
}
```

VS Code Configuration Part 2

Listing 6: launch.json continuation

```
{
    "name": "Launch currently open script",
    "type": "php",
    "request": "launch",
    "program": "${file}",
    "cwd": "${fileDirname}",
    "port": 9000
}
```

Installation on Debian with PHP-FPM and Nginx

1 `apt-get install php-xdebug`

Installation on Debian with PHP-FPM and Nginx

1. `apt-get install php-xdebug`
2. `phpenmod xdebug`

Installation on Debian with PHP-FPM and Nginx

1 `apt-get install php-xdebug`

2 `phpenmod xdebug`

3 `systemctl restart php7.3-fpm`

Introduction
00000

Installation
000
●○

Profiler
○
○

Tracer
○
○

PHP functions

References
○

Server

Installation on Debian with PHP-FPM and Nginx

1. `apt-get install php-xdebug`
2. `phpenmod xdebug`
3. `systemctl restart php7.3-fpm`

Check your Target

Install on your target machine, e.g. Vagrant or Docker container.
Check the installation with `php -i | grep xdebug`.

Introduction
ooooo

Installation
ooo
●o

Profiler
o
o

Tracer
o

PHP functions
o

References
o

Server

Installation on Debian with PHP-FPM and Nginx

1. `apt-get install php-xdebug`
2. `phpenmod xdebug`
3. `systemctl restart php7.3-fpm`

Check your Target

Install on your target machine, e.g. Vagrant or Docker container.
Check the installation with `php -i | grep xdebug`.

Common Pitfall

Check your PHP version with `php -v` or with `<?php phpinfo();`

PHP settings

Listing 7: Remote Debugging

```
zend_extension = xdebug.so
xdebug.remote_enable = On
xdebug.remote_connect_back = On

; xdebug.remote_host = 10.0.49.1
```

/etc/php/7.3/mods-available/xdebug.ini

These values are already present in Debian / Ubuntu. No need to edit! Make sure you enabled the plugin with `phpenmod`.

Introduction
00000

Installation
000
00

Profiler
●
○

Tracer

PHP functions

References
○

Profiler

What does it do?

Xdebugs Profiler is a powerful tool that gives you the ability to analyze your PHP code and determine bottlenecks or generally see which parts of your code are slow and could use a speed boost.

Introduction
ooooo

Installation
ooo
oo

Profiler
●
○

Tracer

PHP functions

References
○

Profiler

What does it do?

Xdebugs Profiler is a powerful tool that gives you the ability to analyze your PHP code and determine bottlenecks or generally see which parts of your code are slow and could use a speed boost.

- Output Files: cachegrind.out

Profiler

What does it do?

Xdebugs Profiler is a powerful tool that gives you the ability to analyze your PHP code and determine bottlenecks or generally see which parts of your code are slow and could use a speed boost.

- Output Files: cachegrind.out
- Analysis Tools
  - KCacheGrind (Linux)
  - QCacheGrind (Linux, Mac)
  - WinCacheGrind (Windows)
  - Webgrind (web-based)

Introduction
ooooo

Installation
ooo
oo

Profiler
o
●

Tracer
o

PHP functions
o

References
o

Configuration

Configuration

```
systemctl restart php7.3-fpm or
```

- Set HTTP GET/POST variable XDEBUG_PROFILE
- Set a Cookie in the header XDEBUG_PROFILE

Introduction
00000

Installation
000
00

Profiler
0
0

Tracer

PHP functions

References
●

Resources & Further Readings

- Official Xdebug Documentation https://xdebug.org/docs