MODERN DEBUGGING IN PHP

OR: WHAT TO DO WHEN YOUR CODE PLAYS TRICKS ON YOU

DAN HOLMES @ APRIL 1ST, 2015

KANSAS CITY PHP USER'S GROUP

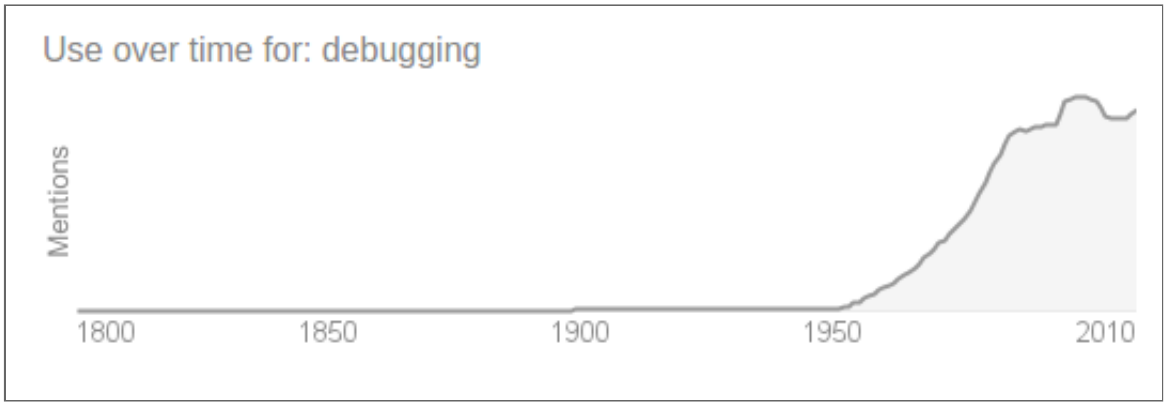
WHO IS THIS TALK FOR?

YOU

Assuming you do anything with PHP

DEBUGGING

... a methodical process of finding and reducing the number of bugs, or defects, in a computer program...thus making it behave as expected.



Source: wikipedia, Google

WHEN I SAY BUGS...

- Things your users find that just "aren't possible"
- Error messages you see in your logs
- Problems during development
- Curiosities

PHP BUILT-INS

- Error Reporting (to screen)
- Error logging
- print_r
- var_dump
- debug_backtrace

ERROR REPORTING (TO SCREEN)

Reload

php-error-report.php

```
<?php
// Enable Error Reporting on Specific levels
    error_reporting(E_ERROR | E_WARNING | E_PARSE | E_NOTICE);
// Report all errors except E_NOTICE
    error_reporting(E_ALL & ~E_NOTICE);
// or, everything except Deprecated Notices
    error_reporting(E_ALL & ~E_DEPRECATED);
// Turn on or Off writing errors to browser:
    ini_set('display_errors','on');
    ini_set('display_errors','off');
```

ERROR REPORTING (TO LOG)

Reload

php-error-report-log.php

```
<?php
// Report all errors except E_NOTICE
error_reporting(E_ALL & ~E_NOTICE);
ini_set('display_errors','off');
ini_set("log_errors",1);
ini_set("error_log","../logs/error.log");

error_log("Hey!  Here I am!");
```

Server's Error Log:

```
[18-Apr-2015 13:33:21 America/Chicago] Hey!  Here I am!
```

refresh (/show-error-log) clear (/clear-error-log)

ERROR REPORTING (TO LOG WITH INFO)

Reload

php-error-report-log-2.php

```
<?php
error_log("Hey!  Here I am! @" . __FILE__  . " line:" . __LINE__);
```

Server's Error Log:

```
[18-Apr-2015 13:33:24 America/Chicago] Hey!  Here I am! @/home/dholmes/Projects/talks/kcpu
```

refresh (/show-error-log) clear (/clear-error-log)

BACKTRACE

Reload

php-error-report-log-3.php

```
<?php
debug_print_backtrace();

$backtrace = debug_backtrace();
print_r($backtrace);
```

Response

```
#0  require() called at [/home/dholmes/Projects/talks/kcpug-debugging-2015/php-example/vendor
/View.php:278]
#1  Slim\View->render(views/php-error-report-log-3.php) called at [/home/dholmes/Projects/tal
ing-2015/php-example/lib/SimpleView.php:14]
#2  SimpleView->render(views/php-error-report-log-3.php, ) called at [/home/dholmes/Projects/
ugging-2015/php-example/vendor/slim/slim/Slim/View.php:255]
#3  Slim\View->fetch(views/php-error-report-log-3.php, ) called at [/home/dholmes/Projects/ta
ging-2015/php-example/vendor/slim/slim/Slim/View.php:243]
#4  Slim\View->display(views/php-error-report-log-3.php) called at [/home/dholmes/Projects/ta
ging-2015/php-example/vendor/slim/slim/Slim/Slim.php:757]
#5  Slim\Slim->render(views/php-error-report-log-3.php) called at [/home/dholmes/Projects/tal
ing-2015/php-example/public/index.php:69]
#6  {closure}(php-error-report-log-3)
#7  call_user_func_array(Closure Object (), Array ([0] => php-error-report-log-3)) called at
```

ERROR REPORTING (WRITING TRACE TO THE ERROR LOG)

Reload

php-error-report-log-4.php

```
<?php
$error = new Exception( "Something weird happened" );
error_log( $error->getTraceAsString() );
```

Server's Error Log:

```
lmes/Projects/talks/kcpug-debugging-2015/php-example/vendor/slim/slim/Slim/View.php(278):
5/php-example/lib/SimpleView.php(14): Slim\View->render('views/php-error...')
5/php-example/vendor/slim/slim/Slim/View.php(255): SimpleView->render('views/php-error...
5/php-example/vendor/slim/slim/Slim/View.php(243): Slim\View->fetch('views/php-error...',
5/php-example/vendor/slim/slim/Slim/Slim.php(757): Slim\View->display('views/php-error...
5/php-example/public/index.php(69): Slim\Slim->render('views/php-error...')
..')
5/php-example/vendor/slim/slim/Slim/Route.php(468): call_user_func_array(Object(Closure),
```

refresh (/show-error-log) clear (/clear-error-log)

ERROR REPORTING (WRITING BETTER TRACE TO THE ERROR LOG)

Reload

php-error-report-log-5.php

```
<?php
function err ($message)
{
    $error = new Exception("Something weird happened");
    error_log($message . "\n" . $error);
}
err("Hey! Something weird just happened");
```

Server's Error Log:

```
go] Hey! Something weird just happened
'Something weird happened' in /home/dholmes/Projects/talks/kcpug-debugging-2015/php-exam

ug-debugging-2015/php-example/templates/views/php-error-report-log-5.php(7): err('Hey! S
ug-debugging-2015/php-example/vendor/slim/slim/Slim/View.php(278): require('/home/dholme
ug-debugging-2015/php-example/lib/SimpleView.php(14): Slim\View->render('views/php-erro
ug-debugging-2015/php-example/vendor/slim/slim/Slim/View.php(255): SimpleView->render('v
ug-debugging-2015/php-example/vendor/slim/slim/Slim/View.php(243): Slim\View->fetch('vie
```

refresh (/show-error-log) clear (/clear-error-log)

INSPECTING VARIABLES WITH PRINT_R()

Reload

php-print_r.php

```
<?php /* Example from xDebug docs */
class test {
    public $pub = false;
    private $priv = true;
    protected $prot = 42;
}
$t = new test;
$t->pub = $t;
$data = array(
    'one' => 'a somewhat long string!',
    'two' => array(
        'two.one' => array(
            'two.one.zero' => 210,
            'two.one.one' => array(
                'two.one.one.zer
o' => 3.1415925,
                'two.one.one.on
e'  => 2.7,
            ),
        ),
    ),
    'three' => $t,
    'four' => range(0, 5),
);
print_r( $data );
```

Response

```
Array
(
    [one] => a somewhat long string!
    [two] => Array
        (
            [two.one] => Array
                (
                    [two.one.zero] => 21
0
                    [two.one.one] => Arr
ay
                        (
                            [two.one.one
.zero] => 3.1415925
                            [two.one.one
.one] => 2.7
                        )

                )

        )

    [three] => test Object
        (
            [pub] => test Object
 *RECURSION*
            [priv:test:private] => 1
            [prot:protected] => 42
```

INSPECTING VARIABLES WITH PRINT_R() RETURNING STRING

Reload

php-print_r-log.php

```
<?php
function err ($message,$data=null){

    $error = new Exception("Something weird happened");
    $error .= "\n" . print_r( $data ,true);

    error_log($message . "\n" . $error);
}

$data = array(
    'one' => 'a somewhat long string!',
    'two' => array(),
    'three' => 'x',
    'four' => range(0, 5),
);

err("Dump my info to the log",array($data,'Server'=>$_SERVER,'Post'=>$_POST));
```

INSPECTING VARIABLES WITH VAR_DUMP()

Reload

php-var-dump.php

```
<?php /* Example from xDebug docs */
class test {
    public $pub = false;
    private $priv = true;
    protected $prot = 42;
}
$t = new test;
$t->pub = $t;
$data = array(
    'one' => 'a somewhat long string!',
    'two' => array(
        'two.one' => array(
            'two.one.zero' => 210,
            'two.one.one' => array(
                'two.one.one.zer
o' => 3.1415925,
                'two.one.one.on
e'  => 2.7,
            ),
        ),
    ),
    'three' => $t,
    'four' => range(0, 5),
);
var_dump( $data );
```

Response

```
array(4) {
  ["one"]=>
  string(23) "a somewhat long string!"
  ["two"]=>
  array(1) {
    ["two.one"]=>
    array(2) {
      ["two.one.zero"]=>
      int(210)
      ["two.one.one"]=>
      array(2) {
        ["two.one.one.zero"]=>
        float(3.1415925)
        ["two.one.one.one"]=>
        float(2.7)
      }
    }
  }
  ["three"]=>
  object(test)#37 (3) {
    ["pub"]=>
    *RECURSION*
    ["priv":"test":private]=>
    bool(true)
    ["prot":protected]=>
    int(42)
  }
  ["four"]=>
```

"LIGHTLY" INSPECTING VARIABLES WITH JSON_ENCODE()

Reload

php-json_encode.php

```
<?php /* Example from xDebug docs */
class test {
    public $pub = false;
    private $priv = true;
    protected $prot = 42;
}
$t = new test;

$data = array(
    'one' => 'a somewhat long string!',
    'two' => array(
        'two.one' => array(
            'two.one.zero' => 210,
            'two.one.one' => array(
                'two.one.one.zer
o' => 3.141,
                'two.one.one.on
e'  => 2.7,
            ),
        ),
    ),
    'three' => $t,
    'four' => range(0, 5),
);

echo json_encode( $data , JSON_PRETTY_PR
INT);
```

Response


```
{
    "one": "a somewhat long string!",
    "two": {
        "two.one": {
            "two.one.zero": 210,
            "two.one.one": {
                "two.one.one.zero": 3.14
1,
                "two.one.one.one": 2.7
            }
        }
    },
    "three": {
        "pub": false
    },
    "four": [
        0,
        1,
        2,
        3,
        4,
        5
    ]
}
```

PHP BUILT-INS - QUICK REVIEW

- Error Handling
  - error_reporting
  - error_log
  - ini settings like display_errors, log_errors, etc
- Backtrace
  - debug_backtrace()
  - Exception->getTraceAsString()
- Inspecting Variables
  - print_r
  - print_r return as string
  - var_dump
  - "serialize" functions like json_encode()

for debugging only, you can set "breakpoints" with die() or exit()

MODERN?

MODERN:

- Shouldn't need to update code
- Shouldn't need to remember to update code again when you are done
- Hope you didn't break something / leave something in
- Out of the way until we need it
- Or, maybe just LOOKS nice!

CATCH EXCEPTIONS WITH WHOOPS!

https://github.com/filp/whoops

0.
…/templates/views/-whoops-run.php:9

Whoops \ Exception \ ErrorException (E_ERROR)

Class 'SomeClassThatDoesntExist' not found

/home/dholmes/Projects/talks/kcpug-debugging-2015/php-example/templates/views/whoops-run.php

```
2.  if ("debug") {
3.      $whoops = new \Whoops\Run;
4.      $whoops->pushHandler( new
    \Whoops\Handler\PrettyPageHandler );
5.      $whoops->register();
6.  }
7.
8.  // Try to instantiate some class
9.  $foo = new SomeClassThatDoesntExist();
```

No comments for this stack frame.

Server/Request Data

/home/dholmes/Projects/talks/kcpug

CATCH EXCEPTIONS WITH WHOOPS!

https://github.com/filp/whoops

```
composer require flip/whoops
```

Reload

whoops.php

```
<?php

if ("debug") {
    $whoops = new \Whoops\Run;
    $whoops->pushHandler( new \Whoops\Handler\PrettyPageHandler );
    $whoops->register();
}
```

MONITOR THE THINGS WITH A DEBUG BAR!

http://phpdebugbar.com

```
composer require maximebf/debugbar
```

PHP Debug Bar

Fork me on GitHub

Do enta

Debugging in PHP has never been easier

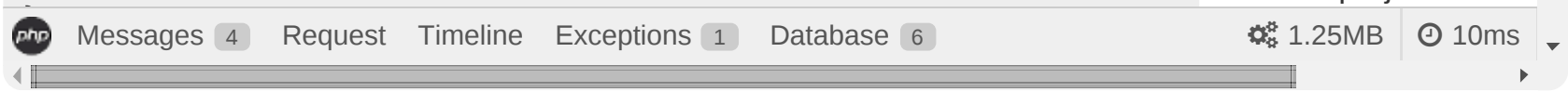
The DebugBar integrates easily in any projects and can display profiling data from any part application. It comes built-in with data collectors for standard PHP features and popular proj

Install the debug bar using Composer:

```
{
  "require": {
    "maximebf/debugbar": "1.*"
  }
```

Features:

- Generic debug dependencies
- Easy to integra project



VAR_DUMP MADE NICER

With Symfony VarDumper and XDebug

symfony-var-dumper-xdebug.php

```
<?php /* Example from xDebug docs */
class test {
    public $pub = false;
    private $priv = true;
    protected $prot = 42;
}
$t = new test;
$t->pub = $t;
$data = array(
    'one' => 'a somewhat long string!',
    'two' => array(
        'two.one' => array(
            'two.one.zero' => 210,
            'two.one.one' => array(
                'two.one.one.zer
o' => 3.141592564,
                'two.one.one.on
e'  => 2.7,
            ),
        ),
    ),
```

Response

Reload

Symfony's VarDumper

```
array:4 [▼
  "one" => "a somewhat long string!"
  "two" => array:1 [▶]
  "three" => test {#37 ▶}
  "four" => array:6 [▶]
]
```

Xdebug's var_dump()

```
array (size=4)
  'one' => string 'a somewhat long str
ing!' (length=23)
  'two' =>
    array (size=1)
      'two.one' =>
        array (size=2)
          'two.one.zero' => int 210
          'two.one.one' =>
            array (size=2)
              ...
  'three' =>
    object(test)[37]
      public 'pub' =>
```

XDEBUG!

- NO need to update code
- NO need to remember to update code again when done
- Out of the way until we need it
- Depending on IDE: just LOOKS nice!

XDEBUG DEMO!

- Inspecting (changing) Variables
- Backtrace (Frames)
- Breakpoints
- Step over, step into, step out
- Break on specific values

CONFIGURE XDEBUG

YOUR MILEAGE WILL VARY

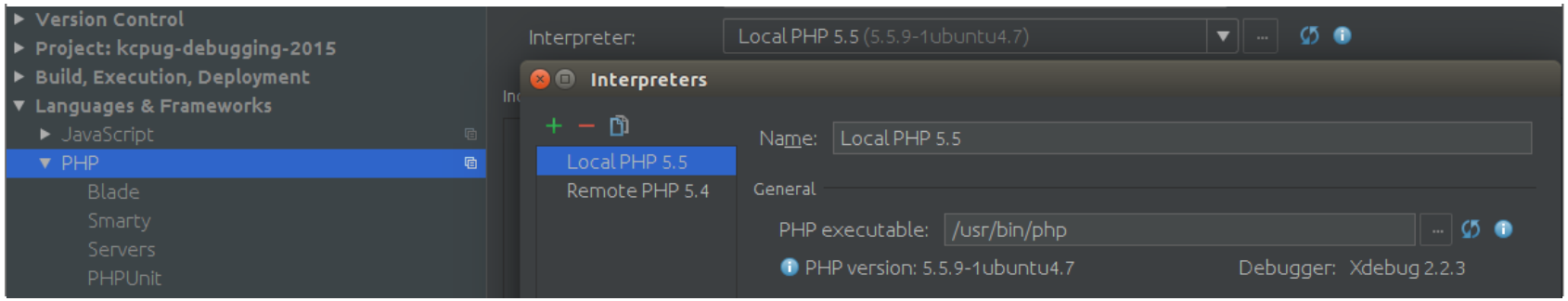
- OS
- PHP Version
- IDE
- Local, Local VM or Remote?

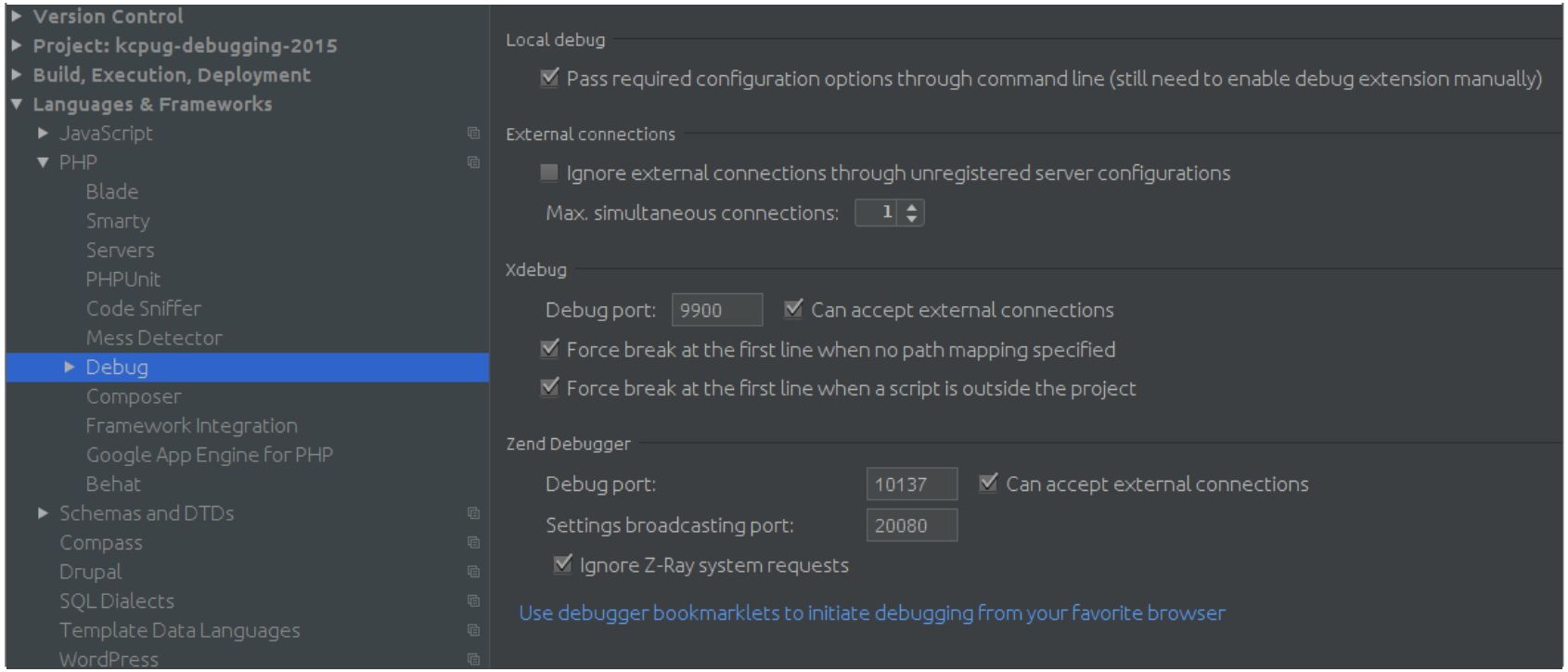
Step 1. Register xDebug and settings with PHP

```
apt-get install php5-xdebug; # download xdebug
vim /etc/php/cli/conf.d/20-xdebug.ini # configure xdebug
```

```
zend_extension=xdebug.so
xdebug.remote_enable=1
xdebug.remote_port=9900
xdebug.profiler_enable=1
```

Step 2. Setup your IDE (phpStorm shown)





Step 3: Make some Bookmarklets

https://www.jetbrains.com/phpstorm/marklets/

Start Debug

```
javascript:(
/** @version 0.5.2 */
    function() {
        document.cookie='XDEBUG_SESSION='+'PHPSTORM'+';path=/;';
    }
)()
```

Stop Debug

```
javascript:(
/** @version 0.5.2 */
    function() {
        document.cookie='XDEBUG_SESSION='+''
        +';expires=Mon, 05 Jul 2000 00:00:00 GMT;path=/;';
    }
)()
```

WALK AWAY

Sometimes it's best to get out of your head and out of your code.

REPLS

Read-Eval-Print-Loop

- php -a ***well, maybe***
- **Boris**
- **PsySh** *My Fav! Thx Eric!*
- **phpsh**

ONLINE "REPLS"

3V4L.ORG

PHPEPL

AND SOMETIMES YOU NEED...

PHP LIVE REGEX

JSFIDDLE

SQL FIDDLE

PHPUNIT!

- NO need to update code
- NO need to remember to update code again when done
- Out of the way until we need it
- Depending on reports: just LOOKS nice!

BEST OF ALL!

once debugged, always debugged!

But that's a whole other talk

THANK YOU!

QUESTIONS? COMMENTS?

Leave Feedback on Joind.in

@dan_holmes

github/dholmes

THESE SLIDES ARE IN+ GITHUB!

github/kcphpug