





Still not dead yet: PHP 8, Xdebug 3 and Composer 2

Carsten Dietrich | 10.11.2020

Agenda

Still not dead yet: PHP 8, Xdebug 3 and Composer 2

1. Who am I
2. Composer 2
3. Xdebug 3
4. PHP8
 - JIT
 - Union Types
 - Attributes
 - Named Arguments
 - Minor changes
 - Breaking changes
5. Conclusion

Who am I

Still not dead yet: PHP 8, Xdebug 3 and Composer 2



Carsten Dietrich

Living in Mainz
Software Developer @ AOE

5+ years PHP / 2+ years Go

Twitter: @carstencodes
GitHub: @carstendietrich

Composer

Still not dead yet: PHP 8, Xdebug 3 and Composer 2

- Has revolutionized the PHP world
- Previously dependencies were checked into SCM or added via Git Submodules or similar concepts
- Maintained by Nils Adermann and Jordi Boggiano
- Development started in the beginning of 2011
- First stable release April 2016 (1.0.0)
- Default repository: Packagist



Whats new in Composer 2

Still not dead yet: PHP 8, Xdebug 3 and Composer 2

- Finally faster download speeds out of the box (2x faster than Composer v1)
 - Parallel downloads / HTTP2 / relies partly on cURL
 - No need for [hirak/prestissimo](#) any more
 - Benchmark with blank Laravel 8.13 (using PHP 7.4.12)
 - Composer 2.0.6: 1m13.072s
 - Composer 1.10.17: 2m29.985s
- New repository metadata format
 - Smaller metadata JSONs due to separation by package
- Repository filtering / priorities / canonical

```
{
  "repositories": [
    {
      "type": "composer",
      "url": "https://packages.drupal.org/8",
      "only": ["drupal/*"]
    },
    {
      "type": "composer",
      "url": "https://composer.typo3.org",
      "exclude": ["typo3/*", "foo/bar"]
    }
  ]
}
```



Whats new in Composer 2

Still not dead yet: PHP 8, Xdebug 3 and Composer 2

Fatal error: Composer detected issues in your platform: Your Composer dependencies require a PHP version "`>= 7.3.0`". You are running 7.1.33 in `/var/www/html/vendor/composer/platform_check.php` on line 24

- New Platform Check during runtime
 - Before autoloading checks requirements like PHP version etc.
 - Execute `composer check-platform-reqs` on target system in your pipeline
- Dry-run support for `require` and `remove` commands
- PEAR support dropped
- `--ignore-platform-req` option to ignore platform requirements selectively
 - `composer install --ignore-platform-req php`
- `composer-plugin-api` is now 2.0
 - If you maintain a plugin go ahead and update
- Try Composer v2 today!
 - `composer self-update --snapshot`



Xdebug

Still not dead yet: PHP 8, Xdebug 3 and Composer 2

- PHP Extension for Debugging/Profiling PHP
- More then 18 years old code (release 1.0.0 end of 2002)
- Maintained by Derick Rethans
- Current stable version 2.9.8
- Xdebug 3.0.0beta1 was released mid October, containing support for PHP8 (2020-10-14)
- In depth talk by Derick about upcoming Xdebug 3:
<https://www.youtube.com/watch?v=114XVdpV4ow>
- **Problems when using Xdebug**
 - Poor Performance (Xdebug was not build for modern PHP apps containing 60k+ functions)
 - Port collision with PHP-FPM (both using port 9000 by default)
 - Massive configuration option amount (currently 65 configuration settings)
 - Lagging behind (Xdebug for PHP 7.3 was 4 months late)



Quick peek at Xdebug 3

Still not dead yet: PHP 8, Xdebug 3 and Composer 2



- Current configuration options 2.9.8:

```
x.auto_trace
x.cli_color
x.collect_assignments
x.collect_includes
x.collect_params
x.collect_return
x.collect_vars
x.coverage_enable
x.default_enable
x.dump.COOKIE
x.dump.ENV
x.dump.FILES
x.dump.GET
x.dump.POST
x.dump.REQUEST
x.dump.SERVER
x.dump.SESSION
x.dump_globals
x.dump_once
x.dump_undefined
x.extended_info
x.file_link_format
```

```
x.filename_format
x.force_display_errors
x.force_error_reporting
x.gc_stats_enable
x.gc_stats_output_dir
x.gc_stats_output_name
x.halt_level
x.idekey
x.max_nesting_level
x.max_stack_frames
x.overload_var_dump
x.profiler_aggregate
x.profiler_append
x.profiler_enable
x.profiler_enable_trigger
x.profiler_enable_trigger_value
x.profiler_output_dir
x.profiler_output_name
x.remote_addr_header
x.remote_autostart
x.remote_connect_back
x.remote_cookie_expire_time
```

```
x.remote_enable
x.remote_handler
x.remote_host
x.remote_log
x.remote_mode
x.remote_port
x.remote_timeout
x.scream
x.show_error_trace
x.show_exception_trace
x.show_local_vars
x.show_mem_delta
x.trace_enable_trigger
x.trace_enable_trigger_value
x.trace_format
x.trace_options
x.trace_output_dir
x.trace_output_name
x.var_display_max_children
x.var_display_max_data
x.var_display_max_depth
```

Quick peek at Xdebug 3

Still not dead yet: PHP 8, Xdebug 3 and Composer 2



- Introducing modes (xdebug.mode)

xdebug.mode	description
off	Nothing is enabled. Xdebug does no work besides checking whether functionality is enabled. Use this setting if you want close to 0 overhead.
develop	Enables Development Aids including the overloaded var_dump() .
coverage	Enables Code Coverage Analysis to generate code coverage reports, mainly in combination with PHPUnit .
debug	Enables Step Debugging . This can be used to step through your code while it is running, and analyse values of variables.
gcstats	Enables Garbage Collection Statistics to collect statistics about PHP's Garbage Collection Mechanism.
profile	Enables Profiling , with which you can analyse performance bottlenecks with tools like KCacheGrind .
trace	Enables the Function Trace feature, which allows you record every function call, including arguments, variable assignment, and return value that is made during a request to a file.



Quick peek at Xdebug 3

Still not dead yet: PHP 8, Xdebug 3 and Composer 2

- Config changes for step debugging in detail:

2	- xdebug.remote_enable=1
3	- xdebug.remote_host=host.docker.internal
4	- xdebug.remote_autostart=1
5	- xdebug.remote_port=9000
6	- xdebug.max_nesting_level=512
2	+ xdebug.client_host=host.docker.internal
3	+ xdebug.client_port=9000
4	+ xdebug.mode=debug
5	+ xdebug.start_with_request=yes

- Performance will be improved over time
- Default port will be switched to 9003 (PHPStorm support for the new Port with v2020.3)
- CLI debugging:
 - export XDEBUG_MODE=debug
 - export XDEBUG_SESSION=previously_the_idekey_thingy
- Full upgrade guide: https://3.xdebug.org/docs/upgrade_guide



Shiny new PHP 8.0

Still not dead yet: PHP 8, Xdebug 3 and Composer 2

- Current PHP Version 7.4
(Active Support until: 28 Nov 2021 Security Support until: 28 Nov 2022)
- PHP 8.0 will be released on 26 Nov 2020
- Features we'll have a look today:
 - JIT
 - Union Types
 - Attributes
 - Named Arguments
 - Minor changes
 - Breaking changes

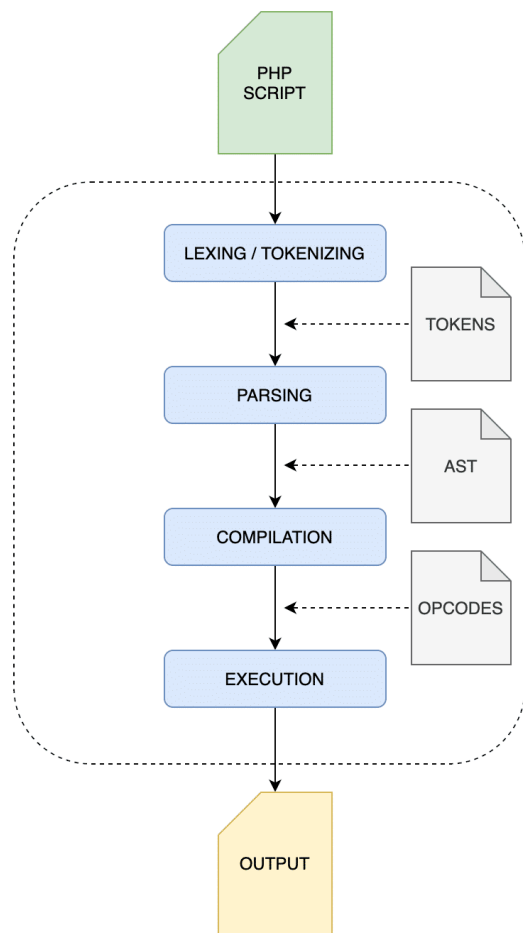
Oct 01 2020	RC 1
Oct 15 2020	RC 2
Oct 29 2020	RC 3
Nov 12 2020	RC 4
Nov 26 2020	GA



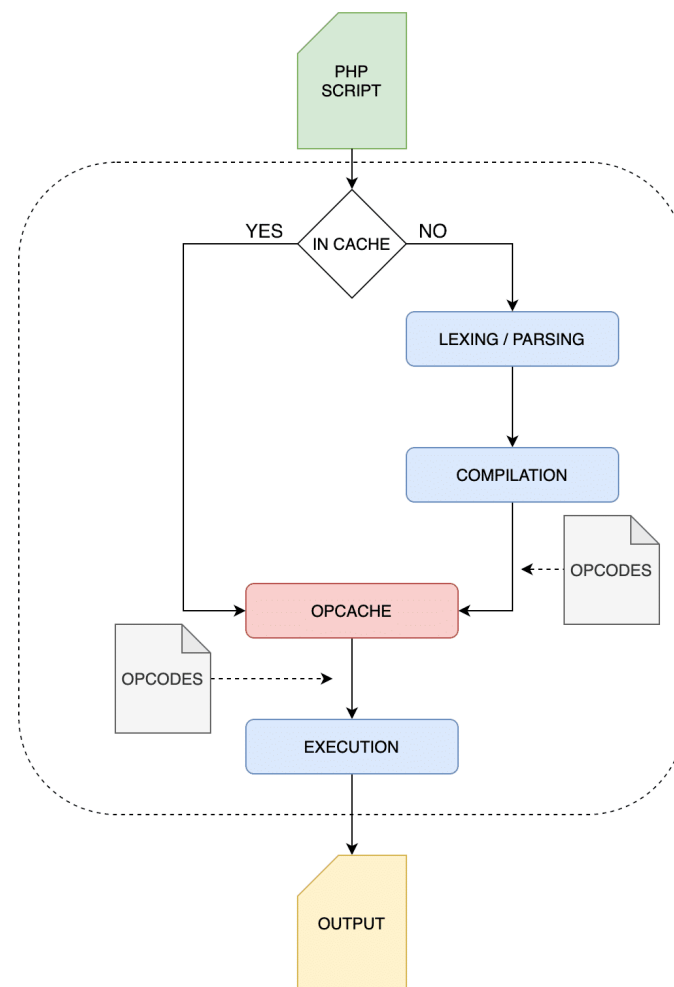
Just in Time Compiler

Still not dead yet: PHP 8, Xdebug 3 and Composer 2

Normal PHP Execution



PHP 5.5+ with OPcache Extension enabled



PHP 7.4+ OPcache preloading



Just in Time Compiler

Still not dead yet: PHP 8, Xdebug 3 and Composer 2

- “The benefits of the JIT compiler are roughly (and as already outlined in the RFC):
 - Significantly better performance for numerical code.
 - Slightly better performance for “typical” PHP web application code.
 - The potential to move more code from C to PHP, because PHP will now be sufficiently fast.”

```
function iterate($x, $y)
{
    $scr = $y - 0.5;
    $ci = $x;
    $zr = 0.0;
    $zi = 0.0;
    $i = 0;
    while (true) {
        $i++;
        $temp = $zr * $zi;
        $zr2 = $zr * $zr;
        $zi2 = $zi * $zi;
        $zr = $zr2 - $zi2 + $scr;
        $zi = $temp + $temp + $ci;
        if ($zi2 + $zr2 > BAILOUT) {
            return $i;
        }
        if ($i > MAX_ITERATIONS) {
            return 0;
        }
    }
}
```

```
JIT$Mandelbrot::iterate: ; (/home/dmitry/php/bench/b.php)
    sub $0x10, %esp
    cmp $0x1, 0x1c(%esi)
    jb .L14
    jmp .L1

.ENTRY1:
    sub $0x10, %esp

.L1:
    cmp $0x2, 0x1c(%esi)
    jb .L15
    mov $0xec3800f0, %edi
    jmp .L2

.ENTRY2:
    sub $0x10, %esp

.L2:
    cmp $0x5, 0x48(%esi)
    jnz .L16
    vmovsd 0x40(%esi), %xmm1
    vsubsd 0xec380068, %xmm1, %
    ...
```



Union Types

Still not dead yet: PHP 8, Xdebug 3 and Composer 2

- Since PHP 7.0 you can declare single types for properties, parameters and return types.
- Since PHP 7.1 you can have nullable types (?string) and void
- Starting with PHP 8.0 you can declare multiple types.

```
class Example {  
    private int|float $foo;  
    private ?string $same;  
    private string|null $samesame;  
  
    public function squareAndAdd(float|int $bar): int|float {  
        return $bar * 2 + $this->foo;  
    }  
  
    public function legacy(): Example|false {  
        if $this->same === null {  
            return false;  
        }  
        return $this;  
    }  
}
```



Attributes

Still not dead yet: PHP 8, Xdebug 3 and Composer 2

- Before PHP 8.0 we commonly used Annotations within DocBlock comments to store metadata (most rely on doctrine/annotations)

```
/**
 * @Route("/home")
 */

/**
 * @ORM\Id
 * @ORM\Column(type="string")
 * @ORM\GeneratedValue
 */
```

- PHP 8.0 brings Attributes

```
<<ORM\Id>>
<<ORM\Column("string")>>
<<ORM\GeneratedValue>>
<<FooBar>>
```



Named Arguments

Still not dead yet: PHP 8, Xdebug 3 and Composer 2

- Allows passing arguments to a function based on the parameter name, rather than the parameter position.
- Makes the meaning of the argument self-documenting
- Makes the arguments order-independent, and allows skipping default values arbitrarily.

```
function test(string $start = "default-start", string $end = "default-end")
{
    echo $start . $end;
}

test(end: "foo");
test(start: "foo");
test();
test("foo", "bar");
test(end: "foo", start: "bar");
```

- Renaming Parameters in the future can be potentially breaking
- `call_user_func_array()` potentially breaks when using associative array



Minor Changes

Still not dead yet: PHP 8, Xdebug 3 and Composer 2

- The built-in server supports dynamic port selection
 - `php -S localhost:0`
- New `mixed` pseudo type (used for legacy core functions)
 - Please use union types in your functions
- JSON extension is always available
 - You can remove `ext-json` from your `composer.json`
- New `str_contains` function
 - No more `strpos() !== false`
- Class constructor property promotion

```
class PHP {  
    private string $name;  
    public function __construct(string $name) {  
        $this->name = $name;  
    }  
}  
  
class PHP8 {  
    public function __construct(private string $name) {}  
}
```



Minor Changes

Still not dead yet: PHP 8, Xdebug 3 and Composer 2

- An optional trailing comma is now allowed in parameter lists

```
function method_with_many_arguments(  
    $a,  
    $b,  
) {}  
  
method_with_many_arguments(  
    1,  
    2,  
);
```

- Added support for nullsafe operator (?->)

```
$user = new User();  
  
$country = $user?->getAddress()?->country?->iso_code;
```



Breaking Changes

Still not dead yet: PHP 8, Xdebug 3 and Composer 2

- Internal functions now also throw `TypeError`'s. (Previously only warnings)
- Large number of warnings are now error exceptions in PHP 8.
- Before PHP 8, it was possible to apply arithmetic or bitwise operators on arrays, resources or objects. This isn't possible anymore, and will throw a `TypeError`.
- Methods with the same name as the class are no longer interpreted as constructors. The `__construct()` method should be used instead.
- Magic Methods will now have their arguments and return types checked if they have them declared.
- Using "parent" inside a class that has no parent will now result in a fatal compile-time error.
- Refer to <https://github.com/php/php-src/blob/PHP-8.0/UPGRADING>



Further useful Resources

Still not dead yet: PHP 8, Xdebug 3 and Composer 2

- Upgrade guide from PHP8: <https://github.com/php/php-src/blob/PHP-8.0/UPGRADING>
- Try out all new features in Browser: <https://pociot.dev/32-php-8-try-out-all-new-features>
- In detail explanation of most of the changes: <https://php.watch/versions/8.0>
- Keep an eye on [phpcompatibility/php-compatibility](https://github.com/phpcompatibility/php-compatibility) for upcoming PHP_CodeSniffer rule set helping to automatically check your code for PHP version support.
- Use [phpstan/phpstan](https://github.com/phpstan/phpstan) with `parameters.phpVersion: 80000`
- [rector/rector](https://github.com/rectorphp/rector) can help you with PHP8 stuff and also help you with PHPUnit 9
- Read about how the WordPress core / eco system will struggle (PHP 5.6+): <https://developer.yoast.com/blog/the-2020-wordpress-and-php-8-compatibility-report/>



Further useful Resources

Still not dead yet: PHP 8, Xdebug 3 and Composer 2

- Try it out yourself locally (of course with DDEV 🥰):
 - DDEV Edge 1.16.0-rc3 contains: PHP 8.0, Xdebug 3 and Composer 2
 - <https://github.com/drud/ddev/releases/tag/v1.16.0-rc3>
- PHPStorm 2020.2
 - currently supports only PHP 8.0 union types, Xdebug 3 works as usual
- PHPStorm EAP 2020.3
 - supports most of the PHP 8.0 changes, adds Xdebug new default port 9003





Conclusion

