PHP 7.2

PHP Forum – Paris – October, 26th 2017

Talk by:

Remi Collet
Senior Software Engineer, Red Hat Inc.
PHP 7.2 Release Manager.

Licensed under Creative Commons Attribution Share Alike – CC-BY-SA
Summary

1. Introduction
2. Roadmap
3. Breaking changes
4. New features
5. Benefits
6. Questions
Introduction
Remi Collet

- 1998 : PHP 3.0 user
- 2005 : Remi's RPM repository / LAMP
- 2006 : Fedora contributor (PHP stack)
- 2007 : Fedora PHP co-maintainer
- 2011 : PECL developer
- 2012 : Fedora / Red Hat PHP maintainer
- 2012 : PHP developer
- 2017 : PHP 7.2 Release Manager

=> http://fr.linkedin.com/in/remicollet
Remi's RPM Repository

- Cible : Fedora, RHEL, CentOS
- PHP 5.4, 5.5, 5.6, 7.0, 7.1, 7.2
- Paquets de base
  - Remplacement (php-*), 1 dépôt par version
- Software Collections
  - Installation en parallèle (php##-*)
  - ~150 extensions
- Upstream de Fedora / RHEL / RHSCL

=> https://rpms.remirepo.net/
Roadmap
PHP 7.2 roadmap

- October, 26<sup>th</sup> 2017: 7.2.0RC5 (**TODAY!!**)  
- November, 7<sup>th</sup> 2017: PHP-7.2.0 branch  
- November, 9<sup>th</sup> 2017: 7.2.0RC6  
- November, 30<sup>th</sup> 2017: 7.2.0 GA

=> https://wiki.php.net/todo/php72
Breaking changes
Fedora QA

- >500 libraries monitored
- build raised by change in dependency tree
- Koschei
  - F28: PHP 7.2.x
  - F27: PHP 7.1.x (including RC)

https://apps.fedoraproject.org/koschei/
Breaking changes

- **mcrypt** is removed
  - libmcrypt is unmaintained for ~8 years
  - was deprecated in 7.1

https://wiki.php.net/rfc/mcrypt-viking-funeral
https://blog.remirepo.net/post/2015/07/07/About-libmcrypt-and-php-mcrypt
Deprecations #1

- __autoload
- $php_errormsg
- create_function()
- mbstring.func_overload
- (unset) cast
- parse_str() without second argument
- gmp_random()
- each()
- assert() with string argument
- $errcontext argument of error handler

https://wiki.php.net/rfc/deprecations_php_7_2
Deprecations #2

- Deprecate png2wbmp() and jpeg2wbmp()
  https://wiki.php.net/rfc/deprecate-png-jpeg-2wbmp

- Deprecate and remove INTL_IDNA_VARIANT_2003
  - 7.2: deprecated
  - 7.4: switch default to INTL_IDNA_VARIANT_UTS46
  - 8.0: drop

Deprecations #3

- Deprecate and Remove Bareword (Unquoted) Strings
  https://wiki.php.net/rfc/deprecate-bareword-strings

```bash
$ php -r 'var_dump(PHP_VERSOIN);'
PHP Notice: Use of undefined constant PHP_VERSOIN - assumed 'PHP_VERSOIN' in Command line code on line 1
string(11) "PHP_VERSOIN"
```
Deprecations #4

- Counting of non-countable objects
- Counts all elements in an array, or **something** in an object.

```php
class Foo implements Countable {
    public function count() {
        return 42;
    }
    ...
}
```
Deprecations #5

Counting of non-countable objects

```php
$ php -r 'var_dump(count(NULL));'
PHP Warning: count(): Parameter must be an array or an object that implements Countable in Command line code on line 1
int(0)

Function doSomething(Array foo=NULL);
```

https://wiki.php.net/rfc/counting_non_countables
Stricter prototype check

- Prototype in extended class must really be compatible

$ PHP Fatal error: Declaration of Zend\Log\Writer\FingersCrossed::setFormatter($formatter) must be compatible with Zend\Log\Writer\AbstractWriter::setFormatter($formatter, ?array $options = NULL) in ...
In object to array cast

```php
$ php -r 'var_dump((object)[0=>1]);'
object(stdClass)#1 (1) {
    ["0"]=>
        int(1)
}
```

- `$obj→{'0'}` and `$obj→{0}` allowed

https://wiki.php.net/rfc/convert_numeric_keys_in_object_array_casts
**Numeric keys #2**

- In array to object cast

```php
$ php -r '$obj = new class {
    public function __construct()
    {
        $this->{0} = 1;
    }
};
var_dump((array)$obj);'
array(1) {
    [0]=>
        int(1)
}
```

- `$arr['0']` and `$arr[0]` allowed
Reserved keyword

- object
- type hinting
Session

- Session is made to manage session status correctly and prevents invalid operations. Only inappropriate codes are affected by this change.

- This new corrected behavior may affect command line mode CLI scripts that manage sessions. Use output buffer just like web applications to resolve problems on CLI scripts.

PHP Warning: ini_set(): Headers already sent. You cannot change the session module's ini settings at this time in ...
New features
Extension loading from ini file:

- **PHP <= 7.1**
  - Linux
    - `extension=foo.so`
  - Windows
    - `extension=php_foo.dll`
- **PHP >= 7.2**
  - `extension=foo`
For security (openssl)

- Improved SSL / TLS constants
- Make tls:// default to TLSv1.0 + TLSv1.1 + TLSv1.2
- Make ssl:// an alias of tls://
- Make STREAM_CRYPTO_METHOD_TLS_* constants default to TLSv1.0 + TLSv1.1 + TLSv1.2 instead of TLSv1.0 only

https://wiki.php.net/rfc/improved-tls-constants
For security (sodium)

- Sodium
- New cryptographic API

```php
$pass = 'secret';
$hash = sodium_crypto_pwhash_str($pass, SODIUM_CRYPTO_PWHASH_OPSLIMIT_SENSITIVE, SODIUM_CRYPTO_PWHASH_MEMLIMIT_SENSITIVE);
$test = sodium_crypto_pwhash_str_verify($hash, $pass);
```
For security (password)

- Argon2 password hash available (optional)
  - PASSWORD_DEFAULT = PASSWORD_BCRYPT

```php
$options = [
    'memory_cost' => PASSWORD_ARGON2_DEFAULT_MEMORY_COST,
    'time_cost' => PASSWORD_ARGON2_DEFAULT_TIME_COST,
    'threads' => PASSWORD_ARGON2_DEFAULT_THREADS,
];

$password_hash = password_hash('password', PASSWORD_ARGON2I, $options);
```
Type Widening

Parameter Type Widening

class ArrayClass {
    public function foo(array $foo) {
        ...
    }
}

class EverythingClass extends ArrayClass {
    public function foo($foo) {
        ...
    }
}
Object type hinting

- allowed for argument and return type hinting

```php
function foo(object $obj): object {
    return $obj;
}
```
Resource to Object

- Migration Hash Context from Resource to Object
- HashContext is reentrant

```php
$ php -r 'var_dump(hash_init("sha256"));'
resource(4) of type (Hash Context)

$ php -r 'var_dump(hash_init("sha256"));'
object(HashContext)#1 (0) {
}
```

https://wiki.php.net/rfc/hash-context.as-resource
Migration Guide

- Migrating from PHP 7.1.x to PHP 7.2.x

Internals (for ext) #1

- ZEND_ACC_CLONE is removed, but was not used in previous versions

- IS_TYPE_IMMUTABLE is removed, IS_TYPE_COPYABLE can be used instead
  Z_IMMUTABLE() check function is still available

- zend_arg_info class_name member is removed, use ZEND_TYPE_NAME instead
Internals #2

- ZEND_BEGIN_ARG_WITH_RETURN_TYPE_INFO_EX
  classname option is removed use:
  - ZEND_BEGIN_ARG_WITH_RETURN_TYPE_INFO_EX for scalar
  - ZEND_BEGIN_ARG_WITH_RETURN_OBJ_INFO_EX for object

- valid_symbol_table is removed from executorGlobals. Use EG(active) instead of removed EG(valid_symbol_table)

- More in UPGRADING.INTERNALS
Benefits
Supported version

- 5.6: security support until Dec 2018
- 7.0: end of active support in Dec 2017
  - Security support until Dec 2018
- 7.1: end of active support in Dec 2018
  - Security support until Dec 2019

=> http://php.net/supported-versions.php
Performance

- Time to run Zend/benchmark.php
  - 7.0: 0.484"
  - 7.1: 0.463"
  - 7.2: 0.443"

- 7.0 => 7.1: ~4-5% faster
- 7.1 => 7.2: ~4-5% faster
Small or big steps?

- Decade upgrade: Big Step
  - Upgrade from 5.4 to 7.2
- Yearly upgrade: Small steps
  - Upgrade from 5.4 to 5.5
  - Upgrade from 5.5 to 5.6
  - Upgrade from 5.6 to 7.0
  - ...
- Fix all deprecated messages ASAP
  - Ready for next minor version
Feedback
https://joind.in/talk/c0e66
Questions?

Contact: remi@php.net