Migrando para o PHP7

Prof. Er Galvão Abbott

Licença de Uso

Resumo:

Você é livre para:

- Compartilhar – Copiar e redistribuir este material em qualquer meio ou formato;
- Adaptar – Remixar, transformar e usar este material como base.

para qualquer propósito, até mesmo Comercial.

Contanto que você o faça sob as seguintes condições:

Atribuição – Você precisa dar o devido crédito, fornecer um link para a licença e indicar explicitamente que mudanças foram feitas. Você pode fazer isto de qualquer maneira razoável, mas isso não implica, de forma alguma, que o licenciador apóia você ou o seu uso deste material.

Compartilhar da mesma forma – Se você remixar, transformar ou usar este material como base, você precisa distribuir o seu material sob a mesma licença do original.

Link para o texto completo da licença:

https://creativecommons.org/licenses/by-sa/4.0/legalcode

Índice

O Fim do PHP5: É hora de migrar!

O dia 19 de Janeiro de 2017 ficou marcado na história da linguagem como o dia em que foi lançada a última atualização regular do PHP5. A partir desta data, e até no mais tardar no dia 31 de Dezembro de 2018, a versão 5 terá apenas atualizações relacionadas a segurança.

Isto significa que, a não ser que se encontre alguma vulnerabilidade de segurança ou que algum bug impacte diretamente a segurança da linguagem, o PHP5 não sofrerá mais modificações. Extraimos disto, portanto, que o PHP5.6 é, de fato, a última versão da linguagem relacionada a *major* 5.

Razões para migrar

- O fim do suporte regular a versão 5;
- Performance: Com uma nova *engine* o PHP7 é, pelo menos, duas vezes mais rápido em seu processamento do que a versão anterior;
- Além disso, o uso de memória caiu significativamente, quase pela metade quando comparado com a versão anterior;
- Novidades: Diversas novas *features*, como Scalar Type Hinting, Return Type Declarations, Group Use Declarations, entre tantas outras que serão abordadas neste curso só estão disponíveis na versão 7. Como veremos no decorrer do curso, estas novas *features* levam a linguagem a um patamar completamente novo.

Novidades no PHP7

Operadores

Dois novos operadores foram introduzidos na versão 7:

- NULL Coalesce (RFC)
- Spaceship (RFC)

Null Coalesce: ??

Retorna operando a esquerda se este não é nulo, senão o operando a direita.

O operador de Null Coalesce traz uma forma mais tranquila de se trabalhar com estruturas de armazenamento que possivelmente não estejam definidas ou inicializadas (ou seja, tem seu valor valendo NULL).

PHP5:

```
<?php
echo (isset($foo) ? $foo : 'Sem valor'); // Output: Sem valor
```

PHP7:

```
<?php
echo ($foo ?? 'Sem valor'); // Output: Sem valor
```

As principais vantagens no uso do NULL Coalesce se resumem a um código mais limpo e não precisar repetir o nome da estrutura mais de uma vez, reduzindo a possibilidade de erros. Além disso ainda é possível encadear múltiplos operadores:

```
<?php
$z = 1;
echo ($x ?? $y ?? $z ?? 'Sem valor'); // Output: 1
```

Spaceship: <=>

Retorna 1 se operando a esquerda é maior, 0 se iguais, -1 se o da direita é maior.

O operador spaceship (“espaçonave”) ou operador de comparação combinada realiza uma comparação entre dados e retorna 3 valores possíveis de acordo com essa comparação: 1, 0 ou -1:

PHP5:

```
<?php
$x = 5;
$y = 2;
echo ($x > $y ? 1 : ($x == $y ? 0 : -1));
```

PHP7:

```
<?php
$x = 5;
$y = 2;

echo $x <=> $y;
```

É possível usar o operador spaceship com diversos tipos diferentes de estruturas, gerando exemplos muito interessantes.

Tipos

Arrays como valores de Constantes

Um problema comum de repetição em PHP5 era causado ao se definir constantes (tipicamente aquelas usadas como “configuração”). O PHP7 agora traz a possibilidade de usarmos arrays como valores de constantes, indo além dos simples tipos escalares.

PHP5:

```
<?php
define(‘DB_HOST’, ‘localhost’);
define(‘DB_USER’, ‘foo’);
define(‘DB_PASS’, ‘bar’);
define(‘DB_PORT’, 3306);
define(‘DB_NAME’, ‘my_database’);
```

PHP7:

```
<?php
define(‘DB’, [
    ‘HOST’ => ‘localhost’,
    ‘USER’ => ‘foo’,
    ‘PASS’ => ‘bar’,
    ‘PORT’ => 3306,
    ‘NAME’ => ‘my_database’,
]);
```

Ganha-se com um código mais legível, organizado e menos repetitivo.

Geração de Dados Aleatórios

A versão 7 traz duas novas funções para trabalhar com geração de dados aleatórios:

- random_int
- random_bytes

RFC

random_int($min, $max)


Retorna un número inteiro >= $min e <= $max

Exemplo:
<?php
echo random_int(1, 10); // Output: Um número aleatório ente 1 e 10

random_bytes($len)


Retorna um dado binário de comprimento $len

Como dados binários são de difícil representação/exibição, é comum utilizarmos uma função específica para armazenar a “versão legível” da informação, como por exemplo a função bin2hex, que gera uma representação hexadecimal de um dado binário.

PHP7:
<?php
$foo = random_bytes(32);
echo bin2hex($foo); // Output: Um dado aleatório representado em hexadecimal

Genericamente falando, a representação hexadecimal gera uma string com o dobro do comprimento do dado original, ou seja, para um dado binário onde $len = 16 sua representação hexadecimal terá 32 caracteres como comprimento.

Sessões

Com o PHP7 é possível definir as configurações de uma sessão programaticamente, ou seja, sobrescrevendo as configurações originais do php.ini. Esta modificação é extremamente interessante por três motivos:

1. Para quem ainda utiliza hospedagens compartilhadas tradicionais, onde mais de um cliente é hospedado no mesmo servidor físico, o php.ini é normalmente inacessível de propósito. Isto evita que alterações na configuração do interpretador da linguagem causem problemas a outras aplicações hospedadas no mesmo servidor. Com a introdução da configuração programática isso deixa de se tornar um problema, permitindo total liberdade a cada desenvolvimento de aplicação;

2. É possível manter uma base de configurações desejáveis no php.ini e alterar apenas algumas outras dependendo do caso;

3. É possível criar, programaticamente, sessões diferentes com configurações diferentes.

Ajustaremos, no curso, nossa sessão para que ela seja executada com settings específicos.

Tipagem

STH – Scalar Type Hints

Tipar parâmetros de funções/métodos usando tipos escalares (int, float, string, bool).

```
<?php
function foo(int $bar)
{
   return $bar + 1;
}
```

RTD – Return Type Declarations

Tipar retorno de funções/métodos usando tipos escalares (int, float, string, bool) e compostos.

```
<?php
function foo(int $bar): int
{
   return $bar + 1;
}
```

Tipagem Estrita

Gerar uma Exception/Erro ao violar STH e RTD.

```
<?php
declare(strict_types = 1);

function foo(int $bar)
{
   return $bar + 1;
}

foo('1');
```

Nullable Types

Possibilidade de usar NULL em STH/RTD, independente da tipagem.

```
<?php
function foo(?int $bar)
{
   return $bar + 1;
}
```

Void [7.1]

Tipar retorno sem valor.

```
<?php
function foo(int &$bar): void
{
   $bar++;
}
```

Quebras de Compatibilidade Reversa

Estruturas de Controle

O fim dos *defaults* múltiplos

No PHP5 era possível a definição de vários casos *default*, o que configura um óbvio erro lógico:

PHP5

```
<?php
$a = 2;

switch ($a) {
   case 0:
      echo 'zero';
   break;

   case 1:
      echo 'um';
   break;

   default:
      echo 'Nem zero, nem um';
   break;

   default:
      echo 'Valor desconhecido';
   break;
}
```

Resultado no PHP5 (independente do uso de *break* nos *defaults*):

Nem zero, nem um

Resultado no PHP7:

PHP Fatal error: Switch statements may only contain one default clause in php shell code on line 16

O fim das funções mysql

O PHP7 trouxe, finalmente, o fim das funções da família mysql. A partir da versão 7 deve-se utilizar mysqli ou PDO (preferido). No curso veremos como utilizar a PDO no lugar das funções mysql.

Próximas Versões: O que esperar

PHP 7.2

Já existem algumas questões interessantes definidas para a próxima minor da versão 7, entre elas:

- Introdução do algoritmo Argon2 para as funções password_hash;
- Introdução da biblioteca criptográfica libsodium como nativa na linguagem (anteriormente estava disponível apenas como extensão PECL);

Referências

- PHP Changelog: http://php.net/manual/en/doc.changelog.php
- Suporte Oficial a versões da linguagem: http://php.net/supported-versions.php
- RFCs: https://wiki.php.net/rfc

Bibliografia Recomendada

- Upgrading to PHP 7
  Davey Shafik
  Ed. O'Reilly
- Learning PHP 7
  Antonio Lopez
  Ed. Packt