PHP 5.6: highlights en wijzigingen

PHP 5.6. levert een mooie performance winst op voor zowel geheugen als de processor. De wijzigingen zijn dan ook voornamelijk gericht op performance en niet zo zeer functionaliteit. Daarnaast zijn er diverse nieuwe functies beschikbaar. Toch moet je ook rekening houden met enkel zaken waarmee PHP 5.6. niet langer backwards compatible is.

Onderstaande vindt u een overzicht met de belangrijkste wijzigingen in PHP 5.6. ten opzichte van PHP 5.5.

Array keys worden niet langer overschreven wanneer men een array definieerd als een property van een class via een array literal.

Voorbeeld;

```
class C {
    const ONE = 1;
    public $array = [
        self::ONE => 'foo',
        'bar',
        'quux',
    ];
}var_dump((new C)->array);
```

Output PHP 5.5

```
array(2) {
 [0]=&gt;
 string(3) &quot;bar&quot;
 [1]=&gt;
 string(4) &quot;<span data-scayt_word="quux" data-scaytid="11">quux</span>&quot;
}
```

Output PHP 5.6.

```
array(3) {
 [1]=&gt;
 string(3) &quot;<span data-scayt_word="foo" data-scaytid="13">foo</span>&quot;
 [2]=&gt;
 string(3) &quot;bar&quot;
 [3]=&gt;
 string(4) &quot;<span data-scayt_word="quux" data-scaytid="14">quux</span>&quot;
}
```

- json_decode() strictheid; json_decode() geeft voortaan altijd in kleine letters true, false, of null terug. Voorheen was het resultaat gedeeltelijk of geheel in hoofdletters.
-

- GMP resources zijn voortaan objecten. De funtionele API implementatie in de GMP extensie is niet gewijzigd, waardoor code ongewijzigd zou moeten werken tenzij expliciet is_resource() wordt gebruikt.
- Mcrypt_encrypt(), mcrypt_decrypt(), mcrypt_cbc(), mcrypt_cfb(), mcrypt_ecb(), mcrypt_generic() en mcrypt_ofb() accepteren niet langer keys of IVs van een incorrecte grootte en block cipher modes die IV's vereisen geven een foutmelding wanneer de IV niet wordt opgegeven.
- CURLFile gebruikt moeten worden.

Alle backwards incompatible commando's terug lezen.

Nieuwe functies zijn uiteraard ook aanwezig, maar deze hebben geen impact op uw bestaande code. Wilt u alle nieuwe functies doornemen? Kijk dan op PHP.net.

Functies welke binnenkort gaan vervallen (deprecated functies in PHP 5.6.), maar nog wel funcioneren in deze versie;

Calls vanaf incompatible context zijn vanaf PHP 5.6. deprecated. Dit betekend dat je niet langer E_STRICT als foutmelding krijgt, maar E_DEPRECATED. Support voor deze functies worden niet langer ondersteund in de volgende versie van PHP.

Voorbeeld:

```
class A {
    function f() { echo
get_class($this); }
 }

 class B {
    function f() { A::f
(); }
 }

 (new B)->f();
```

Output:

```
Deprecated: Non-static method A::f() should not be called statically,
assuming $this from incompatible context in - on line 7
B
```

- $HTTP_RAW_POST_DATA and always_populate_raw_data genereren beiden een E_DEPRECATED foutmelding. Het vervangende commando voor http_raw_post_data is php://input
- Iconv en mbstring encoding instellingen. Onderstaande iconv and mbstring configuratie opties gerelateerd aan encoding worden in de eerst volgende versie van PHP niet langer ondersteund:
  iconv.input_encoding
  iconv.output_encoding

iconv.internal_encoding
mbstring.http_input
mbstring.http_output
mbstring.internal_encoding

Gewijzigde functies;

- crypt() genereerd een **E_NOTICE** error als de salt parameter wordt gebruikt.
- substr_compare() accepteerd nu 0 als lenght parameter.
- unserialize()
- Diverse andere wijzigingen welke minder gebruikt worden zijn te vinden via PHP.net