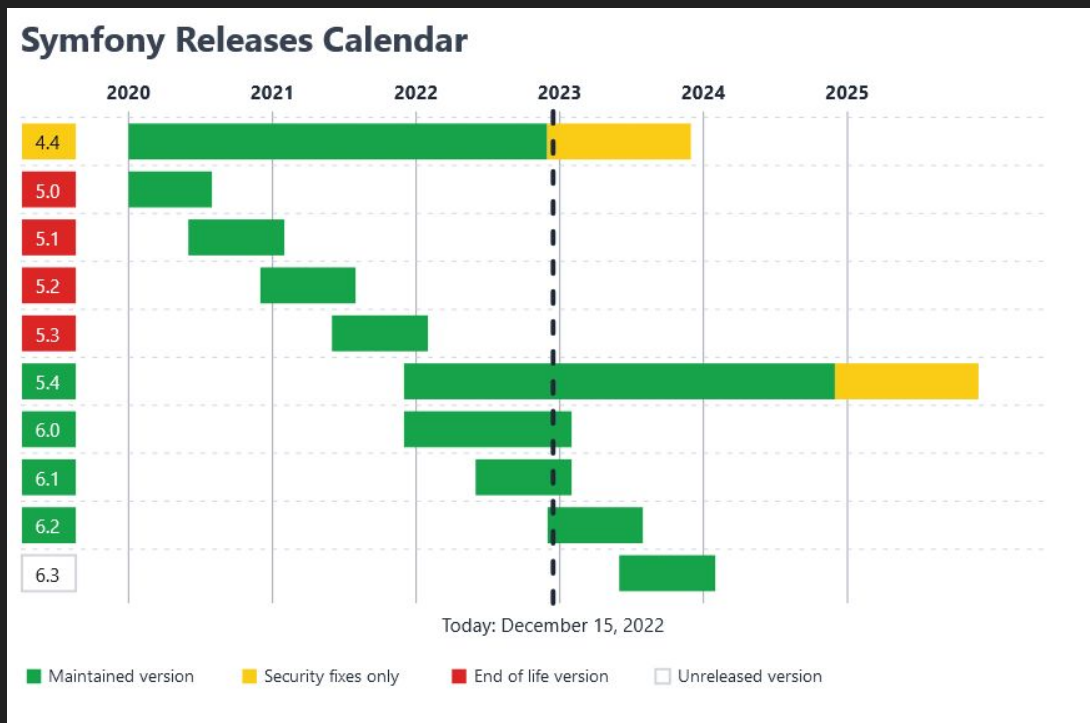


Actu dans l'univers PHP

Sondage

Version Symphony



Version PHP

Currently Supported Versions

Branch	Initial Release		Active Support Until		Security Support Until	
8.0	26 Nov 2020	<i>2 years ago</i>	26 Nov 2022	<i>19 days ago</i>	26 Nov 2023	<i>in 11 months</i>
8.1	25 Nov 2021	<i>1 year ago</i>	25 Nov 2023	<i>in 11 months</i>	25 Nov 2024	<i>in 1 year, 11 months</i>
8.2	8 Dec 2022	<i>7 days ago</i>	8 Dec 2024	<i>in 1 year, 11 months</i>	8 Dec 2025	<i>in 2 years, 11 months</i>

Or, visualised as a calendar:



PHP 8.2

- Nouvelles classes readonly
- Autoriser true, false, et null comme types autonomes
- Types de forme normale disjonctive (DNF)
- Expurgez les paramètres sensibles dans les Back Traces
- Nouvelle fonction mysqli_execute_query et méthode mysqli::execute_query
- Récupérer les propriétés de enum dans les expressions const
- Autoriser les constantes dans les Traits
- Dépréciation des propriétés dynamiques (et nouvel attribut #[AllowDynamicProperties])

- Dépréciation des appelants partiellement supportés
- Dépréciation des fonctions #utf8_encode() et utf8_decode()
- Dépréciation de \${} Interpolation de chaînes de caractères
- Dépréciation des fonctions mbstring pour les entités Base64/QPrint/Uencode/HTML
- Suppression du support de libmysql de mysqli
- Conversion des cas indépendants de la localisation
- Amélioration de l'extension aléatoire

class readonly

```
<?php

readonly class Point
{
    public function __construct(
        public int $x,
        public int $y
    ) {}
}
```

true, false et null comme types d'union

```
public function getUser(int $id): User|null
{
    return null;
}

public function racine(int $nb): float|false
{
    if ($nb<0) {
        return false;
    }
    return sqrt($nb);
}
```

Dépréciation : type dynamique

```
class Post {
```

```
    private int $pid;
```

```
}
```

```
$post = new Post();
```

```
$post->name = 'Kinsta';
```

Dépréciation : callable

"self::method"

"parent::method"

"static::method"

["self", "method"]

["parent", "method"]

["static", "method"]

["Foo", "Bar::method"]

[new Foo, "Bar::method"]

Dépréciation : autre

Deprecated: Function utf8_encode()

mbstring, iconv, et intl

Deprecated: Function utf8_decode()

Deprecated: Using \${}

SF 6.2

- [New in Symfony 6.2: Profiler Redesign](#)
- [New in Symfony 6.2: Console improvements](#)
- [New in Symfony 6.2: Security Improvements \(Part 1\)](#)
- [New in Symfony 6.2: More Extensible Mailer](#)
- [New in Symfony 6.2: DX Improvements](#)
- [New in Symfony 6.2: New Uid Features](#)
- [New in Symfony 6.2: Improved Enum Support](#)
- [New in Symfony 6.2: Better Debugging Commands](#)
- [New in Symfony 6.2: Improved File Validator](#)

- [New in Symfony 6.2: Notifier and Mailer Integrations](#)
- [New in Symfony 6.2: Better PHP Translation Extractor](#)
- [New in Symfony 6.2: Conditional Constraints](#)
- [New in Symfony 6.2: PSR-4 Route Loader](#)
- [New in Symfony 6.2: Finder Improvements](#)
- [New in Symfony 6.2: Clock Component](#)
- [New in Symfony 6.2: Access Token Authenticator](#)
- [New in Symfony 6.2: Better Emoji Support 🎉](#)
- [New in Symfony 6.2: Built-in Cache, Security, Template and Doctrine Attributes](#)

```
class Discount
{
    #[Assert\GreaterThan(0)]

    #[Assert\When(
        expression: 'this.type == "percent"',
        constraints: [
            new Assert\LessThanOrEqual(100, message: 'The value should be between 1 and 100!')
        ],
    )]
    private ?int $value;

    // ...
}
```

Site utile

<https://wiki.php.net/rfc>

<https://kinsta.com/fr/blog/php-8-2/>

<https://symfony.com/blog/category/living-on-the-edge/6.2>