PHP attributes

The end of PHP DocBlocks?

PHP Bergen - Stein Magne Bjørklund

DocBlock and annotation

A piece of documentation in your source code that informs you what the function of a certain class, method or other Structural Element is.

```php
/**
 * Get an example of a beer style.
 *
 * @param string $style
 *   Beer style to query for.
 *
 * @return string
 *   Returns beer style or if nothing found, a empty string.
 *
 * @throws \UnexpectedValueException
 */
function get_beer_style($style) {}
```

#[Attribute]

Attributes

https://www.php.net/manual/en/language.attributes.php

- Short style introduced in PHP 8.0
- [Attr]
  - C#: [Attr] - C++: [[attr]] - Rust: #[attr]
- @Attr
  - Swift: @attr - TypeScript/JS: @Attr - Python: @attr - Java: @Attr - Kotlin: @Attr

Attributes

Built in PHP Core

- #[\Deprecated] - PHP 8.4
- #[\Overriden] - PHP 8.3
- #[\SensitiveParameter] - PHP 8.2
- #[AllowDynamicProperties] PHP 8.2
- #[ReturnTypeWillChange] - PHP 8.1

Laravel

Attributes provided 10.x/11.x

- Observed By
- Scoped By
- Contextual Attributes
- Delete When Missing Models
- Without Relations
- Custom Eloquent Collections

Drupal version 10.2

Example: The humble block

```
#[Block(
    id: "custom_block",
    admin_label: new TranslatableMarkup("Custom Block"),
    category: new TranslatableMarkup("Custom Category"),
)]
```

Doctrine

database storage and object mapping

```
#[ORM\Entity]

#[ORM\Table(name: 'products')]

class Product

{

    #[ORM\Id]

    #[ORM\Column(type: 'integer')]

   private int|null $id = null;

    #[ORM\Column(type: 'string')]

    private string $name;
```

PHPUnit

https://docs.phpunit.de/en/11.5/attributes.html

- #[Test] - That method is a test to be run
- #[TestDox('Hold my beer, PHP Bergen tests is running')]

Drush - annotation vs attributes

```
/**
 * @command xkcd:fetch
 * @param $search Optional argument to retrieve the cartoons matching an index number, keyword, or "random".
 * @option image-viewer Command to use to view images (e.g. xv, firefox).
 * @option google-custom-search-api-key Google Custom Search API Key.
 * @usage drush xkcd
 *   Retrieve and display the latest cartoon.
 * @usage drush xkcd sandwich
 *   Retrieve and display cartoons about sandwiches.
 * @aliases xkcd
 */
public function fetch($search = null, $options = ['image-viewer' ⇒ 'open', 'google-custom-search-api-key' ⇒ 'AIza']) {
    $this→doFetch($search, $options);
}
```

```
/**
 * Retrieve and display xkcd cartoons (attribute variant).
 */
#[
CLI\Command(name: 'xkcd:fetch-attributes', aliases: ['xkcd-attributes'])]
#[CLI\Argument(
  name: 'search',
  description: 'Optional argument to retrieve the cartoons matching an index, keyword, or "random".',
)]
#[CLI\Option(
  name: 'image-viewer',
  description: 'Command to use to view images (e.g. xv, firefox).',
  suggestedValues: ['open', 'xv', 'firefox'],
)]
#[CLI\Option(
  name: 'google-custom-search-api-key',
  description: 'Google Custom Search API Key',
)]
#[CLI\Usage(
  name: 'drush xkcd',
  description: 'Retrieve and display the latest cartoon',
)]
#[CLI\Usage(
  name: 'drush xkcd sandwich',
  description: 'Retrieve and display cartoons about sandwiches.',
)]
public function fetch(
  $search = null,
  $options = [
    'image-viewer' => 'open',
    'google-custom-search-api-key' => 'AIza',
  ],
) {
    $this->doFetch($search, $options);
}
```