

Générer de la Documentation en PHP

Générer de la documentation pour du code PHP peut être très bénéfique pour la maintenabilité et la lisibilité de votre projet. Il existe plusieurs outils pour générer automatiquement de la documentation à partir de commentaires dans le code. Voici quelques-uns des outils les plus populaires :

1. PHPDoc



PHPDoc est le standard de facto pour documenter le code PHP. Voici comment l'utiliser :

Installation

Vous pouvez installer PHPDoc via Composer :

```
composer require --dev phpdocumentor/phpdocumentor
```

Néanmoins, Composer avec PHPDocumentor peut s'avérer problématique

Il vous suffit d'opter pour une des solutions décrites ici :

<https://docs.phpdoc.org/guide/getting-started/installing.html>

Ou de suivre les instructions suivantes dans un environnement Linux:

```
sudo apt install php-pear
apt install graphviz
wget https://github.com/phpDocumentor/phpDocumentor2
/releases/download/v2.9.0/phpDocumentor.phar
sudo mv phpDocumentor.phar /usr/local/bin/phpdoc
```

```
sudo chown -aG root:root /usr/local/bin/phpdoc
sudo chmod +x /usr/local/bin/phpdoc
```

Il ne reste plus qu'à créer un dossier « docs » par exemple :

```
mkdir docs
```

Et d'exécuter avec:

```
phpdoc -d . -t docs/
```

Exemple de Commentaires PHPDoc

Voici un exemple de la façon dont vous pouvez commenter votre code en utilisant PHPDoc :

```
/**
 * Class Car
 *
 * This class represents a car.
 */
class Car {
    /**
     * @var string The make of the car.
     */
    private $make;

    /**
     * @var string The model of the car.
     */
    private $model;

    /**
     * Car constructor.
     *
     * @param string $make The make of the car.
     * @param string $model The model of the car.
     */
    public function __construct(string $make,
        string $model) {
        $this->make = $make;
    }
}
```

```

        $this->model = $model;
    }

    /**
     * Get the make of the car.
     *
     * @return string The make of the car.
     */
    public function getMake(): string {
        return $this->make;
    }

    /**
     * Get the model of the car.
     *
     * @return string The model of the car.
     */
    public function getModel(): string {
        return $this->model;
    }
}

```

2. Doxygen



Doxygen est un autre outil populaire pour générer de la documentation. Bien qu'il soit souvent utilisé pour C++, il supporte également PHP.

Installation

Vous pouvez installer Doxygen en utilisant un gestionnaire de paquets comme apt sur Ubuntu :

```
sudo apt-get install doxygen
```

Utilisation

Pour initialiser un fichier de configuration Doxygen, exécutez :

```
doxygen -g
```

Cela créera un fichier Doxyfile que vous pouvez configurer. Ensuite, pour générer la documentation :

```
doxygen Doxyfile
```

Exemple de Commentaires Doxygen

Les commentaires Doxygen sont similaires à ceux de PHPDoc :

```
/**
 * @brief This class represents a car.
 */
class Car {
    /**
     * @brief The make of the car.
     * @var string
     */
    private $make;

    /**
     * @brief The model of the car.
     * @var string
     */
    private $model;

    /**
     * @brief Constructor.
     *
     * @param string $make The make of the car.
     * @param string $model The model of the car.
     */
    public function __construct(string $make,
        string $model) {
        $this->make = $make;
        $this->model = $model;
    }
}
```

```

    * @brief Get the make of the car.
    *
    * @return string The make of the car.
    */
    public function getMake(): string {
        return $this->make;
    }

    /**
    * @brief Get the model of the car.
    *
    * @return string The model of the car.
    */
    public function getModel(): string {
        return $this->model;
    }
}

```

3. ApiGen



ApiGen est un autre générateur de documentation pour PHP qui est très facile à utiliser.

Installation

Vous pouvez installer ApiGen via Composer :

```
composer require --dev apigen/apigen
```

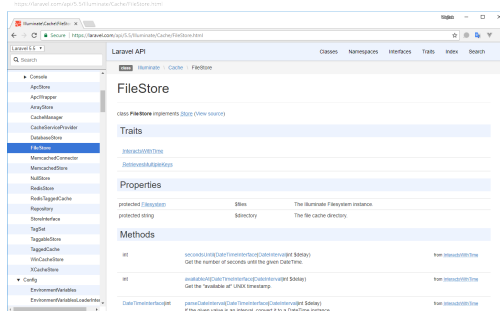
Utilisation

Pour générer la documentation, exécutez :

```
vendor/bin/apigen generate src --destination docs
```

4. Sami (abandonné)

Sami:



Sami était un générateur de documentation PHP créé par Symfony. Bien que le projet soit maintenant abandonné, il est encore utilisé dans certains projets.

Installation

Pour installer Sami, ajoutez-le à vos dépendances via Composer :

```
composer require --dev sami/sami
```

Utilisation

Créez un fichier de configuration `sami.php` :

```
<?php
```

```
use Sami\Sami;
use Sami\Parser\Filter\TrueFilter;
use Sami\Parser\Filter\FALSEFilter;
use Sami\Version\GitVersionCollection;
use Sami\RemoteRepository\
GitHubRemoteRepository;
use Symfony\Component\Finder\Finder;

$iterator = Finder::create()
    ->files()
    ->name('*.php')
```

```

->in(__DIR__ . '/src');

$versions = GitVersionCollection::create(__DIR__)
->add('master', 'master branch');

return new Sami($iterator, [
    'versions' => $versions,
    'title' => 'My Project API',
    'build_dir' => __DIR__ . '/docs/api/%version%',
    'cache_dir' => __DIR__ . '/cache/%version%',
    'remote_repository' =>
        new GitHubRemoteRepository('username/repository',
            __DIR__),
    'default_opened_level' => 2,
]);

```

Ensuite, générez la documentation avec la commande suivante :

```
vendor/bin/sami.php update sami.php
```

Conclusion

Ces outils peuvent automatiser la génération de documentation pour votre code PHP, facilitant ainsi la compréhension et la maintenabilité de vos projets. N'oubliez pas de bien commenter votre code en utilisant les annotations appropriées pour maximiser l'efficacité de ces outils.