

phpDocumentor

Jesús Rodríguez González | 2º DAW
Despliegue de Aplicaciones Web

Índice

Requisitos previos	3
Instalación	3
DocBlocks	4
Corrección de errores	5
Estructura de directorios y ficheros	6
Visualización de documentación generada	6
Creación de repositorio en GitHub	7

En esta práctica vamos a utilizar la herramienta **phpDocumentor** para documentar el código de una aplicación realizada en PHP. Para ello primero vamos a instalar la herramienta.

Requisitos previos

Si vamos a [la documentación del proyecto](#), vemos que los requisitos del sistema incluyen el propio PHP (desde la versión 5.3.3), la **extensión intl** de PHP y Graphviz. Comprobamos que la versión de PHP es correcta, que está habilitada la extensión e instalamos el paquete Graphviz.

```
Jesús sábado 7 marzo 2020 09:51
[jesu@croqueta Escritorio]$ php -v
PHP 7.4.3 (cli) (built: Feb 18 2020 15:35:13) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
;extension=imap
;extension=intl
;extension=ldap

Jesús sábado 7 marzo 2020 09:50
[jesu@croqueta Escritorio]$ sudo pacman -S graphviz
[sudo] password for jesu:
Sorry, try again.
[sudo] password for jesu:
resolviendo dependencias...
buscando conflictos entre paquetes...

Paquetes (3) gts-0.7.6.121130-1 netpbm-10.73.30-1 graphviz-2.42.3-2

Tamaño total de la descarga: 9,77 MiB
Tamaño total de la instalación: 25,00 MiB

:: ¿Continuar con la instalación? [S/n] S
:: Obteniendo los paquetes...
netpbm-10.73.30-... 1858,7 KiB 692 KiB/s 00:03 [#####] 100%
gts-0.7.6.121130-... 194,9 KiB 303 KiB/s 00:01 [#####] 100%
graphviz-2.42.3-... 7,8 MiB 978 KiB/s 00:08 [#####] 100%
(3/3) comprobando las claves del depósito [#####] 100%
(3/3) verificando la integridad de los paquetes [#####] 100%
(3/3) cargando los archivos de los paquetes [#####] 100%
(3/3) comprobando conflictos entre archivos [#####] 100%
(3/3) comprobando el espacio disponible en el ... [#####] 100%
:: Procesando los cambios de los paquetes...
(1/3) instalando netpbm [#####] 100%
(2/3) instalando gts [#####] 100%
(3/3) instalando graphviz [#####] 100%
```

Instalación

Dado que en [la documentación del proyecto](#) se recomienda no usar **composer**, vamos a utilizarlo con PHAR, ya que con este método no tenemos que instalarlo y, podemos ejecutarlo directamente.

Primero nos descargamos el paquete .phar que encontramos en [la página de versiones](#), y luego lo ejecutamos con el siguiente comando, en el que le indicamos el fichero (-f) o

directorio (-d) sobre el que queremos que actúe, así como el target o destino donde se va a guardar la documentación generada tras la ejecución.

```
$ php phpDocumentor.phar -f php/funcion.php -t docs/api
```

DocBlocks

Sin embargo, antes debemos realizar una serie de anotaciones en nuestro código para que la documentación se genere adecuadamente. [Como vemos en la guía de phpDocumentor](#), nuestro código debe contar con lo que los desarrolladores llaman **DocBlock** y definen como “un fragmento de documentación insertada entre las líneas de tu código que informa [a phpDocumentor] de cuál es la función de una clase, método u otro elemento estructural”.

```
1  <?php
2  /**
3   * A summary informing the user what the associated element does.
4   *
5   * A *description*, that can span multiple lines, to go _in-depth_ into the details of this element
6   * and to provide some background information or textual references.
7   *
8   * @param string $myArgument With a *description* of this argument, these may also
9   *   span multiple lines.
10  *
11  * @return void
12  */
13  function myFunction($myArgument)
14  {
15  }
```

Vamos a documentar una sencilla función que guarda en dos variables **\$a** y **\$b** sendos números entre 1 y 10 generados aleatoriamente con la función **rand()**. Dichas variables se pasan por parámetros a la función **suma()**, que devuelve el resultado de dicha suma. Luego guardamos el valor en la variable **\$resultado**, que imprimimos junto con el valor de **\$a** y **\$b**. Junto al código incluimos el **DocBlock** para documentar el funcionamiento de nuestra función.

```
<?php
/**
 * suma() function: brief and simple function to sum two int params
 *
 * This function receives as arguments two integer numbers, $a and $b, previously randomly and automatically
 * generated by the script; then the function adds both arguments and returns the result of the operation;
 * finally, the result is assigned to a variable $resultado, and it is printed with echo alongside $a and $b values
 *
 * @param int $a random number from 1 to 10
 * @param int $b random number from 1 to 10
 *
 * @return int $a+$b result of adding two params
 */
$a=rand(1,10);
$b=rand(1,10);

function suma($a,$b){
    return $a+$b;
}

$resultado=suma();
echo "Suma de ".$a."+".$b."=".$resultado;
?>
```

Corrección de errores

Siendo ésta la disposición original del *DocBlock*, recibimos, sin embargo, varios errores.

```
Parsing /home/jesu/Escritorio/php/funcion.php
Argument $a is missing from the Docblock of suma
Argument $b is missing from the Docblock of suma
No summary for function \suma()
```

Realizamos, pues, algunas modificaciones siguiendo la documentación de varias páginas y los comentarios de un par de issues abiertos por usuarios en el repositorio de GitHub de phpDocumentor. De este modo, el *DocBlock* queda finalmente así.

```
<?php
/**
 * Brief and simple script to sum two int params
 */

$a=rand(1,10);
$b=rand(1,10);

/**
 * Brief and simple script to sum two int params
 *
 * This function receives as arguments two integer numbers, $a and $b, previously randomly and automatically
 * generated by the script; then the function adds both arguments and returns the result of the operation;
 * finally, the result is assigned to a variable $resultado, and it is printed with echo alongside $a and $b values
 *
 * @param int $a
 * @param int $b
 *
 * @return int
 */

function suma($a,$b){
    return $a+$b;
}

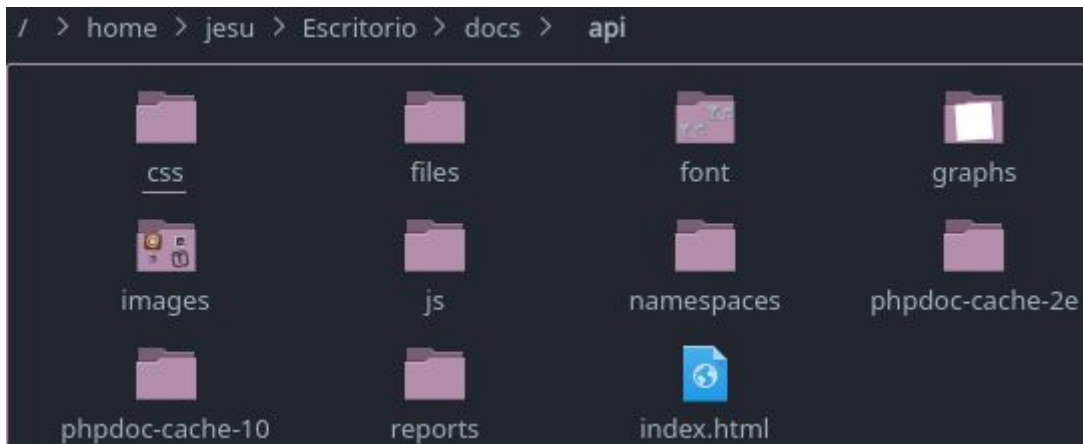
$resultado=suma();
echo "Suma de ".$a."+".$b."=".$resultado;
?>
```

Cuando ejecutamos, vemos que ya no hay errores.

```
Collecting files .. OK
Initializing parser .. OK
Parsing files
Parsing /home/jesu/Escritorio/php/funcion.php
Storing cache in "/home/jesu/Escritorio/docs/api" .. OK
Load cache .. 0.000s
Preparing template "clean" .. 0.003s
Preparing 17 transformations .. 0.000s
Build "elements" index .. 0.000s
Replace textual FQCNs with object aliases .. 0.000s
Resolve @link and @see tags in descriptions .. 0.000s
Enriches inline example tags with their sources .. 0.000s
Build "packages" index .. 0.000s
Build "namespaces" index and add namespaces to "elements" .. 0.000s
Collect all markers embedded in tags .. 0.000s
Transform analyzed project into artifacts ..
Applying 17 transformations
  Initialize writer "phpDocumentor\Plugin\Core\Transformer\Writer\FileIo"
  Initialize writer "phpDocumentor\Plugin\Twig\Writer\Twig"
  Initialize writer "phpDocumentor\Plugin\Graphs\Writer\Graph"
  Execute transformation using writer "FileIo"
  Execute transformation using writer "FileIo"
  Execute transformation using writer "FileIo"
  Execute transformation using writer "FileIo"
  Execute transformation using writer "FileIo"
  Execute transformation using writer "twig"
```

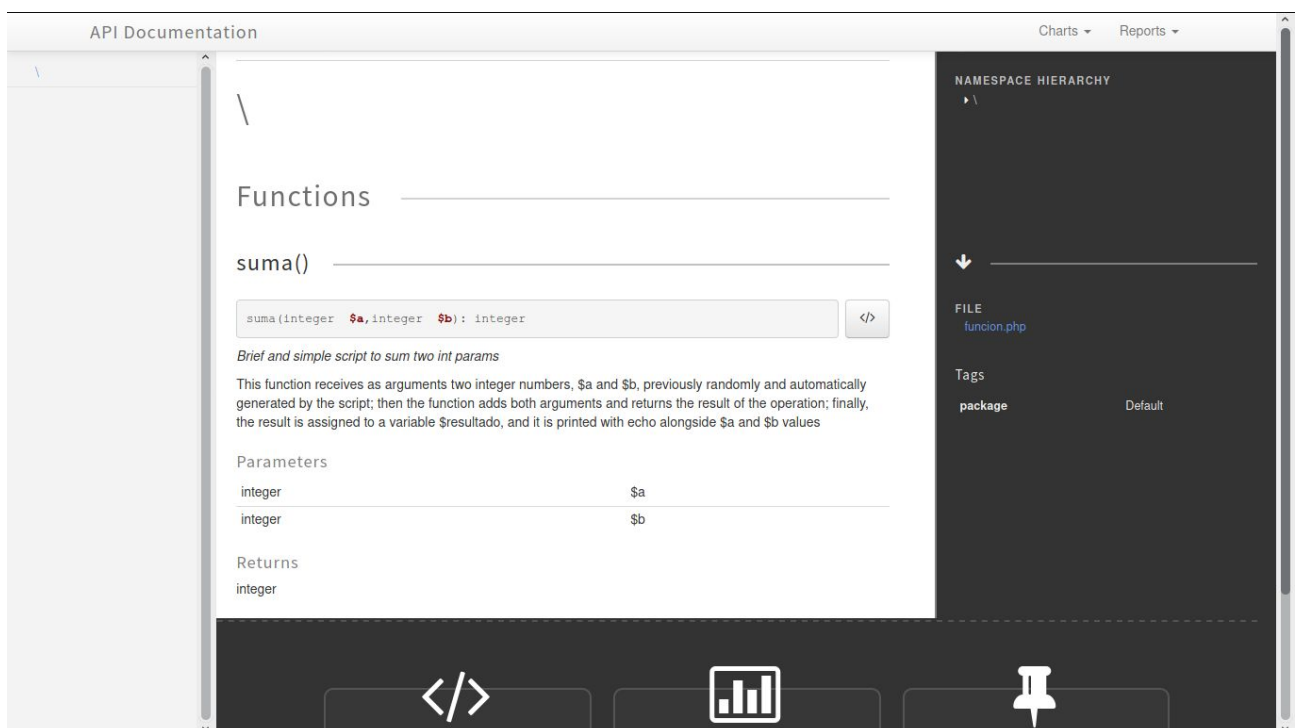
Estructura de directorios y ficheros

Ahora, si nos vamos al directorio `docs/api`, nos encontramos estos archivos y directorios.



Visualización de documentación generada

Si ejecutamos el `index.html` en nuestro navegador, veremos lo siguiente: una web autogenerada con el gestor de plantillas Twig en la que se nos muestra la documentación de nuestro código generada con phpDocumentor.



Creación de repositorio en GitHub

Por último, vamos a crear un repositorio en GitHub para subir nuestra documentación. El repositorio está disponible en la siguiente URL.

<https://github.com/jesusrodriguezg/phpDocumentor-sample>

The screenshot shows the GitHub interface for the repository 'jesusrodriguezg / phpDocumentor-sample'. At the top, there's a navigation bar with links for Pull requests, Issues, Marketplace, and Explore. Below this, the repository name is displayed along with buttons for Unwatch (1), Star (0), and Fork (0). A secondary navigation bar includes links for Code, Issues (0), Pull requests (0), Actions, Projects (0), Wiki, Security, Insights, and Settings. The main content area features a title 'Sample PHP function used to try phpDocumentor' with an 'Edit' button. Below the title, there's a section for repository statistics: 3 commits, 1 branch, 0 packages, 0 releases, and 1 contributor. A 'Branch: master' dropdown and a 'New pull request' button are also present. A table lists recent commits, showing the latest commit 'jesusrodriguezg Update README.md' from 1 minute ago. Below the commit list, there's a section for the 'README.md' file, which contains the title 'phpDocumentor-sample' and a description: 'Sample PHP function used to try phpDocumentor'. The description text reads: 'This is a simple project created as classwork documentation for the Web Applications Deployment module of Web Application Development degree, taught by Diego Fernández Raposo in the IES Polígono Sur high school in Seville,'.