



Variables de entorno en PHP con archivos .env

PHP

 Lectura: 14 min[🏠](#) > [Manuales](#) > [Taller de PHP](#) Última actualización: 26 de febrero de 2020

Trabajar con variables de entorno en aplicaciones PHP. Acceso sencillo a los datos de archivos .env usando la librería PHP dotenv.

En este artículo vamos a realizar una de las prácticas habituales y necesarias en las aplicaciones PHP, que es el almacenamiento de las variables de entorno y su posterior recuperación en el código fuente de las páginas.

Para poder incorporar de una manera ágil las variables de entorno en aplicaciones PHP usaremos una librería que ya viene preparada con las funcionalidades básicas que podremos llegar a necesitar en el día a día. La librería se llama "PHP dotenv" y básicamente permite la lectura sencilla de las variables de entorno en archivos de extensión ".env". Mediante estos archivos podemos almacenar las variables en una notación bastante habitual y usada en diversos frameworks profesionales. En nuestro caso no usaremos un framework, sino una librería independiente, que nos proporcionará esta funcionalidad concreta en cualquier aplicación PHP que podamos construir.



Por qué son tan útiles las variables de entorno

El uso de variables de entorno, o environment, en las aplicaciones web es importante para la separación del código por responsabilidades: no debemos mezclar el código de la aplicación con los valores de configuración.

Las variables de configuración cambiarán generalmente según el entorno de ejecución, es decir, cuando una aplicación está siendo ejecutada en distintos servidores. Por ejemplo, el servidor local para desarrollo y el servidor remoto donde se va a colocar el sitio web accesible a los usuarios de Internet. Por tanto, separarlas en archivos independientes permitirá que el mismo código funcione en cualquier lugar, sin necesidad de modificaciones, independientemente de los valores de configuración que se tengan en cada entorno.

Seguramente veamos la utilidad de las variables de entorno si recurrimos a un ejemplo típico de uso. Pensemos en los valores de conexión con la base de datos (host, user, password...). Esos valores seguramente serán distintos en el servidor de desarrollo y en el servidor de producción. En la práctica esos valores los almacenaremos en variables de entorno, y cuando sea necesario se usarán esos valores para realizar las correspondientes conexiones con la base de datos.

Otros elementos que sería importante tener separados en variables de entorno son las credenciales de acceso a diversos servicios, llaves APIs (API Keys), el nombre del host donde se está ejecutando la aplicación, etc.

Archivos .env

Los archivos .env se pueden considerar un estándar para el almacenamiento de variables de entorno. Estos archivos tienen un formato muy sencillo y fácil de escribir y de leer.

La sintaxis de los .env contiene pares clave (nombre de variable) y valor, separados por un carácter "=". Cada variable en una línea.

```
MODE="development"
```

Lo estás imaginando... usamos cookies!
y por ello nos obligan a avisarte y
ofrecerte estas herramientas para la
configuración.

Aceptar todas

configurar

Nota: Como ves, los nombres de las variables de entorno se suelen poner con mayúsculas, por convención, ya que son valores constantes.

El archivo .env generalmente lo colocarás en la raíz de tu proyecto, en un archivo sin nombre. La ruta del .env no debe estar accesible por Apache, sino que debe de ser un directorio por encima, para que los usuarios no puedan acceder a la configuración de nuestras variables de entorno. Fíjate también que el nombre del archivo comenzará con un punto y luego las tres letras "env". En sistemas como Linux o Mac, el archivo que comienza por "." se considera oculto.

Lo que es importante es que estos archivos no se encuentren al alcance de los usuarios, por lo que nunca deberían estar dentro de la carpeta de publicación, sino algún directorio por encima en el servidor. Es decir, se colocará en la raíz del proyecto, pero nunca dentro de la carpeta raíz de publicación, pues si estuvieran allí los usuarios podrían acceder a esos valores componiendo la ruta como "example.com/.env". Obviamente, los valores de configuración deseamos que estén seguros, por lo que no deben ser accesibles por el público en general.

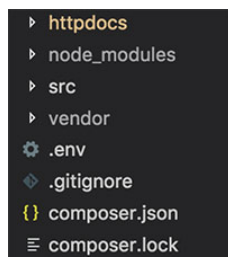
Nota: Este punto es de vital importancia. Cuando decimos la raíz del proyecto, el lugar para colocar los .env, no nos referimos a la carpeta raíz de publicación. Generalmente en un dominio tienes una serie de carpetas en tu host.

- data
- httpdocs
- stats
- ...

Estos nombres de carpetas pueden variar, pero siempre habrá una que será la carpeta raíz de publicación. Algo como "httpdocs", "www", "html", "httpsdocs"... el nombre es lo de menos. Para aclararnos, la raíz de publicación será el lugar donde tengas el index.php de tu portada. Es importante que el .env no lo coloques en la carpeta raíz de publicación, sino en la carpeta superior. Así es como lo tendrás fuera del alcance de los usuarios que te visiten, algo muy importante ya que en los .env sueles colocar información muy sensible.

Cuando decimos que debe estar en la raíz del proyecto, nos referimos fuera del directorio de publicación!!!

Para que quede claro coloco aquí una imagen con una estructura típica de un proyecto PHP, en la que encuentras el .env en la raíz del proyecto.



En la anterior imagen, el directorio de publicación, donde está el index.php se llama "httpdocs". Como puedes comprobar, el .env no está ahí dentro, pues si no, quedaría accesible a tus usuarios.

Nota: otra cosa habitual es que los archivos .env se encuentren fuera del repositorio de la aplicación. Por ejemplo, si usas Git tendrías que poner los archivos .env dentro del [gitignore](#).

Librería PHP dotenv

Esta es una de muchas alternativas para la lectura de variables de entorno en archivos .env. Es nuestra elección para este artículo por ser compatible con la mayoría de versiones de PHP, desde la 5.4.

Nota: Seguramente el framework de turno tendrá su propia librería, o incluso puede que use algún formato diferente donde almacenar estos datos. Si no usamos un framework podremos contar con esta librería para facilitarnos las cosas.

Lo estás imaginando... usamos cookies!
y por ello nos obligan a avisarte y
ofrecerte estas herramientas para la
configuración.

Aceptar todas

configurar

tarea de abrir el archivo donde las variables de entorno se almacenan
irlas cómodamente dentro de las aplicaciones.

Usando Composer puedes instalar fácilmente esta librería en tu proyecto por medio del comando:

```
php composer.phar require vlucas/phpdotenv
```

Nota: En el [Manual de Composer](#) puedes encontrar más información sobre esta librería, instalación, usos, etc. Si ya conoces Composer también sabrás que existe una alternativa a este comando: "composer require vlucas/phpdotenv", que te funcionará si has colocado el ejecutable de Composer en el path.

Una vez instalada la librería podrás usarla donde lo necesites de tu código. Para ello simplemente tienes que usar el autoloader de clases que viene incorporado con Composer.

```
<?php
require('../vendor/autoload.php');
?>
```

Carga de archivos con variables de entorno

PHP dotenv nos permite tener varios archivos de configuración, aunque lo común es tener un único archivo que contenga todas las variables de entorno. Para acceder al contenido de los archivos .env tenemos que crear un nuevo objeto de la clase "Dotenv", enviando la ruta del archivo al que queremos acceder (pero sin la extensión ".env") y luego ejecutar el método load().

Actualizado: en la versión 4 de la librería Dotenv se usa este código para cargar las variables de entorno:

```
$dotenv = Dotenv\Dotenv::createImmutable('../');
$dotenv->load();
```

En versiones anteriores se usaba este otro código:

```
$dotenv = new Dotenv\Dotenv('../');
$dotenv->load();
```

En ambos ejemplos se abriría un archivo llamado ".env" que cuelga del directorio padre del actual, ruta '../'.

Acceso a las variables de entorno

Una vez que hemos cargado los archivos de variables de entorno, estarán disponibles sus variables y valores en el código de las aplicaciones por medio de varios métodos diferentes.

La función getenv(), en la que pasamos la cadena de la variable de entorno que queremos acceder.

- A través del array superglobal \$_ENV
- A través del array superglobal \$_SERVER

Por ejemplo, podríamos acceder al valor de la variable de entorno llamada "MODE", de la siguiente manera:

```
$mode = getenv('MODE');
```

Otra alternativa posible sería:

```
$clave = $_ENV['DDBB_PASSWORD'];
```

Sobreescritura de variables de entorno

Generalmente las variables de entorno son de "solo lectura" y además inmutables. Es por ello que, si ya se ha definido una variable de entorno, no se puede sobreescribir.

No obstante, si deseas que unas variables de entorno cargadas puedan machacar valores de variables de entorno ya existentes, entonces

Esto abrirá un archivo llamado ".env" en el directorio actual y luego cargará dichos valores, machacando otros que ya pudiera haber.

Conclusión

Hemos aprendido qué son las variables de entorno. Es algo muy importante que debes de conocer y no tiene que ver específicamente con PHP, sino que se usa en prácticamente cualquier lenguaje backend.

Además hemos conocido una librería llamada PHP dotenv que resulta muy útil y sencilla para consumir variables de entorno en PHP, almacenadas en archivos .env, el formato más común para el almacenamiento de este tipo de datos del environment.

Miguel Angel Alvarez

Fundador de DesarrolloWeb.com y la
plataforma de formación online EscuelaIT.
Com...



Manual

Taller de PHP

Intervention Image, manipulación de imágenes con
PHP

¿Te ha resultado útil este artículo?

(2)

(0)

¿Alguna duda?

Pregunta y ayuda en la comunidad con tus respuestas en la [sección de FAQ](#)

HACER UNA PREGUNTA

Aprender

Comunidad

Lo estás imaginando... usamos cookies!
y por ello nos obligan a avisarte y
ofrecerte estas herramientas para la
configuración.

Aceptar todas

configurar

Preguntas y respuestas

Hacer una pregunta

Elecciones

Experiencia