

## Configuración

Todos los archivos de configuración del framework Laravel se almacenan en el directorio **config**. Cada opción está documentada, así que siéntase libre de mirar a través de los archivos y familiarizarse con las opciones disponibles.

Estos archivos de configuración permiten configurar cosas tales como la conexión a la base de datos, el servidor de correo, así como otros valores de configuración básicos, como la zona horaria de la aplicación y la clave de cifrado.

## Descripción general de la aplicación

Puede obtener una visión general rápida de la configuración, los controladores y el entorno de la aplicación a través del comando about de Artisan

```
php artisan about
```

Si solo se está interesado en una sección en particular de la salida de la descripción general de la aplicación, se puede filtrar por esa sección usando la opción: **--only**

```
php artisan about --only=environment
```

## Configuración del entorno

A menudo es útil tener diferentes valores de configuración según el entorno donde se ejecuta la aplicación. Por ejemplo, es posible que se desee utilizar un controlador de caché diferente localmente al que se utiliza en un servidor de producción.

Para hacer esto fácil, Laravel utiliza la biblioteca PHP [DotEnv](#) de Vance Lucas. Esta librería básicamente permite la lectura sencilla de las variables de entorno en archivos de extensión ".env". En una instalación nueva de Laravel, el directorio raíz de la aplicación contiene un archivo **.env.example**. Al realizar la instalación de Laravel a través de Composer, este archivo se copia automáticamente a **.env**. Si no se realiza la instalación con Composer, se debe copiar el archivo manualmente.

El archivo **.env** no debe estar bajo el control de versiones del código fuente de la aplicación, ya que cada desarrollador/servidor que use la aplicación puede requerir una configuración de entorno diferente. Además, esto es un riesgo de seguridad en el caso de que un intruso obtenga acceso a su repositorio de control de versiones del código fuente, ya que cualquier credencial confidencial quedaría expuesta.

Si estás desarrollando con un equipo de trabajo, es posible que se debas incluir un archivo **env.example** en la aplicación. Al colocar valores en el archivo de configuración de ejemplo, otros desarrolladores de un equipo pueden ver claramente qué variables de entorno son necesarias para ejecutar la aplicación. También se puede crear un archivo

**.env.testing.** Este archivo anula al archivo **.env** al ejecutar pruebas **PHPUnit** o ejecutar comandos Artisan con la opción **--env=testing**.

**NOTA:** Cualquier variable del archivo **.env** puede ser anulada por variables de entorno externas.

Los archivos con variables de entornos siguen la estructura CLAVE=VALOR, por ejemplo, para Laravel 9 el archivo **.env.example** predeterminado es el siguiente:

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel
DB_USERNAME=root
DB_PASSWORD=

BROADCAST_DRIVER=log
CACHE_DRIVER=file
FILESYSTEM_DISK=local
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120

MEMCACHED_HOST=127.0.0.1

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_MAILER=smtp
MAIL_HOST=mailhog
MAIL_PORT=1025
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
MAIL_FROM_ADDRESS="hello@example.com"
MAIL_FROM_NAME="${APP_NAME}"

AWS_ACCESS_KEY_ID=
AWS_SECRET_ACCESS_KEY=
AWS_DEFAULT_REGION=us-east-1
AWS_BUCKET=
AWS_USE_PATH_STYLE_ENDPOINT=false
```

```
PUSHER_APP_ID=  
PUSHER_APP_KEY=  
PUSHER_APP_SECRET=  
PUSHER_HOST=  
PUSHER_PORT=443  
PUSHER_SCHEME=https  
PUSHER_APP_CLUSTER=mt1  
  
VITE_PUSHER_APP_KEY="${PUSHER_APP_KEY}"  
VITE_PUSHER_HOST="${PUSHER_HOST}"  
VITE_PUSHER_PORT="${PUSHER_PORT}"  
VITE_PUSHER_SCHEME="${PUSHER_SCHEME}"  
VITE_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"
```

Por ejemplo, para definir que el proyecto está instalado en el entorno local dejamos la clave `APP_ENV` con el valor **local**, en cambio, si es el entorno de producción colocamos **production** y para el entorno de pruebas ponemos **testing**.

Todas las variables en los archivos **.env** se analizan como cadenas. Si necesitamos definir una variable de entorno con un valor que contenga espacios, puede hacerse encerrando el valor entre comillas dobles.

```
APP_NAME="My Application"
```

## Recuperando la configuración del entorno

Todas las variables enumeradas en los archivos **.env** se cargan en **\$\_ENV** cuando la aplicación reciba una solicitud. Sin embargo, puede usar el helper (una función ayudante) **env**, para recuperar valores de estas variables en los archivos de configuración. Este helper acepta dos parámetros: la variable a buscar y un valor por defecto en caso que dicha variable no exista:

```
'default' => env('DB_CONNECTION', 'mysql')
```

El segundo valor pasado a la función **env** es el valor predeterminado. Este valor se utilizará si no existe una variable de entorno para la clave dada.

## Acceso a los valores de configuración

Puedes acceder fácilmente a los valores de configuración utilizando la función auxiliar global **config** desde cualquier lugar de una aplicación. Se puede acceder a los valores de configuración utilizando la sintaxis " punto ", que incluye el nombre del archivo y la opción a la que desea acceder. También se puede especificar un valor predeterminado que se devolverá si la opción de configuración no existe:

```
$value=config('app.timezone');
```

```
// Devuelve un valor por defecto si el valor de configuración no  
existe  
$value=config('app.timezone', 'Europe/Madrid');
```

Para establecer valores de configuración en tiempo de ejecución, pase un array al helper (función) config:

```
config(['app.timezone' => 'Europe/Madrid']);
```

## Configuración de almacenamiento en caché

Para aumentar la velocidad de una aplicación, debes almacenar en caché todos tus archivos de configuración en un solo archivo usando el comando **config:cache** de Artisan. Esto combina todas las opciones de configuración para la aplicación en un solo archivo que se carga más rápidamente.

Normalmente se debe ejecutar el comando **php artisan config:cache** como parte de su rutina de implementación de producción. El comando no debe ejecutarse durante el desarrollo local, ya que las opciones de configuración con frecuencia se cambian durante el desarrollo de una aplicación.

## Modo depuración

La opción **debug** del archivo de configuración **config/app.php** determina cuánta información sobre un error se muestra realmente al usuario. De forma predeterminada, esta opción está configurada para respetar el valor de la variable de entorno, que se almacena en el archivo **.env**.

Para el desarrollo local, debe establecer la variable de entorno **APP\_DEBUG** a **true**. En el entorno de producción, este valor siempre debe ser **false**. Si la variable se establece en producción a **true**, corre el riesgo de exponer valores de configuración confidenciales a los usuarios finales de la aplicación.

## Configuración adicional

Laravel no necesita casi ninguna otra configuración. Sin embargo, puede que quieras revisar el archivo **config/app.php** y su documentación. Contiene varias opciones como **timezone** y **locale** que es posible que desees ajustar en tu aplicación.

También puede que quieras configurar algunos componentes adicionales de Laravel, como:

- [Cache](#)
- [Base de Datos](#)
- [Sesiones](#)