

[\\$_GET »](#)
[« \\$GLOBALS](#)

- [Manual de PHP](#)
- [Referencia del lenguaje](#)
- [Variables predefinidas](#)

Change language: Spanish ▼

[Submit a Pull Request](#) [Report a Bug](#)

\$_SERVER

\$HTTP_SERVER_VARS [eliminado]

(PHP 4 >= 4.1.0, PHP 5, PHP 7, PHP 8)

\$_SERVER -- \$HTTP_SERVER_VARS [eliminado] — Información del entorno del servidor y de ejecución

Descripción ¶

\$_SERVER es un array que contiene información, tales como cabeceras, rutas y ubicaciones de script. Las entradas de este array son creadas por el servidor web. No hay garantía que cada servidor web proporcione alguna de estas entradas, existen servidores que pueden omitir algunas o proporcionar otras no recogidas aquí. Un gran número de estas variables se encuentran recogidas en [» especificación CGI 1.1](#), así que al menos debe esperar encontrar estas entradas.

Nota: Antes de PHP 5.4.0, \$HTTP_SERVER_VARS contenía la misma información inicial, aunque no era una [superglobal](#). (Observe que \$HTTP_SERVER_VARS y \$_SERVER eran variables diferentes, por lo que PHP las trata de forma distinta).

Índices ¶

Puede encontrar o no los siguientes elementos en \$_SERVER. Tenga en cuenta que si ejecuta PHP desde [línea de comando](#) pocos o ninguno de los siguientes elementos estarán disponibles (o tendrán algún significado).

'PHP_SELF'

El nombre del archivo de script ejecutándose actualmente, relativa al directorio raíz de documentos del servidor. Por ejemplo, el valor de \$_SERVER['PHP_SELF'] en un script ejecutado en la dirección `http://example.com/foo/bar.php` será `/foo/bar.php`. La constante [__FILE__](#) contiene la ruta completa del fichero actual, incluyendo el nombre del archivo. Si PHP se está ejecutando como un proceso de línea de comando, esta variable es el nombre del script desde PHP 4.3.0. En anteriores versiones no estaba disponible.

'argv'

Array de los argumentos enviados al script. Cuando se ejecuta el script en línea de comando se obtiene acceso a los parámetros de línea de comando con un estilo parecido a como sería en C. Cuando se ejecuta el script mediante el método GET, contendrá la cadena de la consulta.

'argc'

Contiene el número de parámetros de línea de comando enviados al script (si se ejecuta en línea de comando).

'GATEWAY_INTERFACE'

Número de revisión de la especificación CGI que está empleando el servidor, por ejemplo 'CGI/1.1'.

'SERVER_ADDR'

La dirección IP del servidor donde se está ejecutando actualmente el script.

'SERVER_NAME'

El nombre del host del servidor donde se está ejecutando actualmente el script. Si el script se ejecuta en un host virtual se obtendrá el valor del nombre definido para dicho host virtual.

Nota: Bajo Apache 2, se debe establecer `UseCanonicalName = On` y `ServerName`. De lo contrario, este valor refleja el nombre del host proporcionado por el cliente, el cual se puede burlar. No es seguro depender de este valor en contextos que necesiten seguridad.

`'SERVER_SOFTWARE'`

Cadena de identificación del servidor dada en las cabeceras de respuesta a las peticiones.

`'SERVER_PROTOCOL'`

Nombre y número de revisión del protocolo de información a través del cual la página es solicitada, por ejemplo `'HTTP/1.0'`.

`'REQUEST_METHOD'`

Método de petición empleado para acceder a la página, por ejemplo `'GET'`, `'HEAD'`, `'POST'`, `'PUT'`.

Nota:

El script de PHP se considera terminado después de enviar las cabeceras (es decir después de producir cualquier resultado sin emplear buffers para el resultado) si el método de la petición empleado era HEAD.

`'REQUEST_TIME'`

Fecha Unix de inicio de la petición. Disponible desde PHP 5.1.0.

`'REQUEST_TIME_FLOAT'`

El timestamp del inicio de la solicitud, con precisión microsegundo. Disponible desde PHP 5.4.0.

`'QUERY_STRING'`

Si existe, la cadena de la consulta de la petición de la página.

`'DOCUMENT_ROOT'`

El directorio raíz de documentos del servidor en el cual se está ejecutando el script actual, según está definida en el archivo de configuración del servidor.

`'HTTP_ACCEPT'`

Contenido de la cabecera `Accept`: de la petición actual, si existe.

`'HTTP_ACCEPT_CHARSET'`

Contenido de la cabecera `Accept-Charset`: de la petición actual, si existe. Por ejemplo: `'iso-8859-1,*;utf-8'`.

`'HTTP_ACCEPT_ENCODING'`

Contenido de la cabecera `Accept-Encoding`: de la petición actual, si existe. Por ejemplo: `'gzip'`.

`'HTTP_ACCEPT_LANGUAGE'`

Contenido de la cabecera `Accept-Language`: de la petición actual, si existe. Por ejemplo: `'en'`.

`'HTTP_CONNECTION'`

Contenido de la cabecera `Connection`: de la petición actual, si existe. Por ejemplo: `'keep-alive'`.

`'HTTP_HOST'`

Contenido de la cabecera `Host`: de la petición actual, si existe.

`'HTTP_REFERER'`

Dirección de la página (si la hay) que emplea el agente de usuario para la página actual. Es definido por el agente de usuario. No todos los agentes de usuarios lo definen y algunos permiten modificar `HTTP_REFERER` como parte de su funcionalidad. En resumen, es un valor del que no se puede confiar realmente.

`'HTTP_USER_AGENT'`

Contenido de la cabecera `User-Agent`: de la petición actual, si existe. Consiste en una cadena que indica el agente de usuario empleado para acceder a la página. Un ejemplo típico es: `Mozilla/4.5 [en] (X11; U; Linux 2.2.9 i586)`. Entre otras opciones, puede emplear dicho valor con [get_browser\(\)](#) para personalizar el resultado de la salida de la página en función de las capacidades del agente de usuario empleado.

`'HTTPS'`

Ofrece un valor no vacío si el script es pedido mediante el protocolo HTTPS.

Nota: Tenga en cuenta que si se emplea ISAPI con IIS el valor será off si la petición no se ha realizado a través del protocolo HTTPS.

'REMOTE_ADDR'

La dirección IP desde la cual está viendo la página actual el usuario.

'REMOTE_HOST'

El nombre del host desde el cual está viendo la página actual el usuario. La obtención inversa del dns está basada en la *REMOTE_ADDR* del usuario.

Nota: Su servidor web debe estar configurado para crear esta variable. Por ejemplo en Apache necesita que exista `HostnameLookups On` dentro de *httpd.conf*. Consulte también [gethostbyaddr\(\)](#).

'REMOTE_PORT'

El puerto empleado por la máquina del usuario para comunicarse con el servidor web.

'REMOTE_USER'

El usuario autenticado.

'REDIRECT_REMOTE_USER'

El usuario autenticado si la petición es redirigida internamente.

'SCRIPT_FILENAME'

La ruta del script ejecutándose actualmente en forma absoluta.

Nota:

Si un script se ejecuta mediante CLI como ruta relativa, como por ejemplo *file.php* o *../file.php*, entonces `$_SERVER['SCRIPT_FILENAME']` contendrá la ruta relativa especificada por el usuario.

'SERVER_ADMIN'

El valor dado a la directiva `SERVER_ADMIN` (de Apache) en el archivo de configuración del servidor web. Si el script se está ejecutando en un host virtual, el valor dado será el definido para dicho host virtual.

'SERVER_PORT'

El puerto de la máquina del servidor usado por el servidor web para la comunicación. Para las configuraciones por omisión, el valor será '80'; el empleo de SSL, por ejemplo, cambiará dicho valor al valor definido para el puerto HTTP seguro.

Nota: Bajo Apache 2, se debe establecer `UseCanonicalName = On`, así como `UseCanonicalPhysicalPort = On` para poder obtener el puerto físico (real), de otro modo, este valor podría ser burlado y podría o no devolver el valor del puerto físico. No es seguro confiar en este valor en contextos que requieran seguridad.

'SERVER_SIGNATURE'

Cadena que contiene la versión del servidor y el nombre del host virtual que son añadidas a las páginas generadas por el servidor, si esta habilitada esta funcionalidad.

'PATH_TRANSLATED'

Ruta de acceso basada en el sistema (no en el directorio raíz de documentos del servidor) del script actual, después de cualquier mapeo de virtual a real realizada por el servidor.

Nota: A partir de PHP 4.3.2, *PATH_TRANSLATED* no está definida de forma implícita en el SAPI de Apache 2, en comparación a la situación de Apache 1, donde era necesario establecer el mismo valor que la variable del servidor *SCRIPT_FILENAME* cuando no era proporcionada por Apache. Este cambio ha sido realizado para cumplir la especificación CGI donde *PATH_TRANSLATED* sólo debe existir si *PATH_INFO* está definida. Los usuarios de Apache 2 pueden emplear `AcceptPathInfo = On` dentro de *httpd.conf* para definir *PATH_INFO*.

'SCRIPT_NAME'

Contiene la ruta del script actual. Esto es de utilidad para las páginas que necesiten apuntarse a si mismas. La constante [FILE](#) contiene la ruta absoluta y el nombre del archivo actual incluido.

'REQUEST_URI'

La URI que se empleó para acceder a la página. Por ejemplo: '/index.html'.

'PHP_AUTH_DIGEST'

Cuando se hace autenticación Digest HTTP, esta variable se establece para el encabezado 'Authorization' enviado por el cliente (el cual se debe entonces usar para hacer la validación apropiada).

'PHP_AUTH_USER'

Cuando se hace autenticación HTTP, esta variable se establece para el nombre de usuario provisto por el usuario.

'PHP_AUTH_PW'

Cuando se hace autenticación HTTP, esta variable se establece para la clave provista por el usuario.

'AUTH_TYPE'

Cuando se realiza la autenticación HTTP, esta variable se establece para el tipo de autenticación.

'PATH_INFO'

Contiene cualquier información sobre la ruta proporcionada por el cliente a continuación del nombre del fichero del script actual pero antecediendo a la cadena de la petición, si existe. Por ejemplo, si el script actual se accede a través de la URL `http://www.example.com/php/path_info.php/some/stuff?foo=bar`, entonces `$_SERVER['PATH_INFO']` contendrá `/some/stuff`.

'ORIG_PATH_INFO'

Versión original de `'PATH_INFO'` antes de ser procesado por PHP.

Historial de cambios ¶

Versión	Descripción
5.4.0	<code>\$HTTP_SERVER_VARS</code> ya no está disponible debido a la eliminación de arrays grandes de registro.
5.3.0	La directiva <code>register_long_arrays</code> , la cual hacía que estuviera disponible <code>\$HTTP_SERVER_VARS</code> está obsoleta.

Ejemplos ¶

Ejemplo #1 *Ejemplo de \$_SERVER*

```
<?php
echo $_SERVER['SERVER_NAME'];
?>
```

El resultado del ejemplo sería algo similar a:

`www.example.com`

Notas ¶

Nota:

Esta es una 'superglobal' o una variable automatic global. Significa simplemente que es una variable que está disponible en cualquier parte del script. No hace falta hacer **global \$variable;** para acceder a la misma desde funciones o métodos.

Ver también ¶

- [Filtrado de datos](#)

[+ add a note](#)