



## PHP con register\_globals=off

Fuentes: <http://www.zend.com> y [http://mx.php.net/register\\_globals](http://mx.php.net/register_globals)

### Introducción

Muchos desarrolladores se han acostumbrado a escribir código PHP arrastrando un ya conocido agujero de seguridad, las controvertidas register\_globals que en muchos servidores están a **ON**, esto significa que se pueden pasar parámetros por ejemplo en la url lo que supone un error de seguridad.

Ahora hay que cambiar el código y añadirle mayor seguridad desactivando esta opción en el servidor. (**OFF**)

Con la llegada de PHP 4.2.0, el valor por defecto para register global paso a off, un problema que hay que resolver de inmediato para no comprometer el servidor.

¿Porque se ha hecho esto? Hace falta una única palabra para explicarlo **Seguridad**. Tu código es mucho más seguro y estable cuando sabes exactamente de donde y hacia donde va cada una de las variables que utilizas, ten cuidado declarar de manera arbitraria las variables en tu código no es una práctica muy recomendable.

La configuración de las register\_globals se encuentra en el archivo llamado php.ini.

### Como recoger variables de manera segura

Los más experimentados que ya saben como hacerlo deberán saltarse este punto y pasar directamente a la sección de superglobals.

Las variables pueden venir de diferentes fuentes , \$var='valor'; tiene diferentes formas de ser recogidas en tu script, incluyendola como parte de la URL, en un formulario, en una cookie o como parte del servidor en el que está funcionando nuestro código.

Este ejemplo está explicado para un servidor con el parámetro register\_globals On,y m´as tarde veras como conseguir la misma variable pero con el parámetro register\_globals off

### Desde la URL

Una de las maneras más comunes de recoger la información de una variable es pasando las variables por la URL.

<http://www.tusitio.com/index.php?variable=valor&variable2=valor2>

Con las register\_globals On podríamos recoger en nuestro script estas variables de manera automática.

Cuando un parametro es especificado en la URL, PHP creara un array global llamado \$HTTP\_GET\_VARS. Para el ejemplo de arriba, PHP automáticamente creara \$HTTP\_GET\_VARS = array ('variable1' => 'valor1', 'variable2' => 'valor2');

Desde PHP 4.1.0, la variable global \$\_GET contendrán lo mismo que \$HTTP\_GET\_VARS. Este array es un superglobal y será disctuido más adelante en este artículo.



### **Desde un formulario**

Es la segunda manera más utilizada para recoger las variables:

Partimos de la base de que has creado tu formulario explicaremos cual es el funcionamiento, cuando enviamos el formulario a través del botón introducido en nuestro formulario el navegador enviará el formulario al script con una variable con el nombre de cada uno de los campos que hayas definido en el formulario, siempre y cuando tengamos `register_globals` On el script tendrá la variable disponible como una variable global por defecto.

Igualmente que lo explicado anteriormente , PHP automáticamente creará `$HTTP_POST_VARS` un array asociativo que tendrá como resultado la creación de algo así:  
`$HTTP_POST_VARS['variable'] = 'valor';`

A partir de PHP 4.1.0 y superiores, la variable `$_POST` contendrá el mismo array asociativo.

### **Desde una cookie**

Esta es la manera de fijar una cookie en PHP:

*/\* Fijar cookie para un dia \*/*

```
setcookie('variable', 'valor', time()+86400, "", $HTTP_HOST);
```

Nota: Con `register_globals = 'off'`, necesitaríamos `$_SERVER['HTTP_HOST']` en vez de `$HTTP_HOST`.

Igual que de las otras maneras señaladas pasará la variable y la definirá de una manera muy parecida.

### **Desde el Servidor**

PHP crea adicionalmente arrays asociativos como

`$HTTP_ENV_VARS` y `$HTTP_SERVER_VARS`.

A partir de PHP 4.1.0 se llaman `$_ENV` y `$_SERVER`.

### **Usa Superglobales**

Ahora que entiendes como recoger variables en PHP, y que ellas no son creadas automaticamente por tu PHP cuando `register_globals` estan desactivadas, ahora tratare de mostraros como ajustar vuestro código a la nueva configuración.

Tu primera eleccion debe ser usar las nuevos arrays superglobales, despues de todo esto es para lo que lo añadieron los desarrolladores de PHP.

### **¿Por qué son llamadas superglobales?**

Superglobales son una excepción con respecto a otras variables. Tu puedes usar las variables `$_GET`, `$_POST`, `$_COOKIE`, `$_ENV`, `$_SERVER` y `$_SESSION` sin tener que declararlas como globales primero.



Nota: Tu no puedes usar una variable para acceder a arrays superglobales en funciones, por ejemplo el siguiente código no funcionará:

```
function foo()  
{  
  $sg = '_GET';  
  return ${$sg}[$var];  
}  
?>
```

La función foo() no devolvera los valores de \$\_GET.

### Otras técnicas

Estas son unas funciones que nos ayudarán a realizar la transición:

La primera es register()

```
/**  
 * devuelve un valor de los arrays globales  
 *  
 * @param string $varname  
 * el nombre de la variable a registrar  
 *  
 * @param string $defval opcional  
 * el valor a devolver si no se encuentra  
 *  
 * @return string el valor de la variable si  
 * esta registrada  
 */  
function register($varname, $defval=NULL)  
{  
  if (array_key_exists($varname, $_SERVER)) {  
    $retval = $_SERVER[$varname];  
  } elseif (array_key_exists($varname, $_COOKIE)) {  
    $retval = $_COOKIE[$varname];  
  } elseif (array_key_exists($varname, $_POST)) {  
    $retval = $_POST[$varname];  
  } elseif (array_key_exists($varname, $_GET)) {  
    $retval = $_GET[$varname];  
  } elseif (array_key_exists($varname, $_ENV)) {  
    $retval = $_ENV[$varname];  
  } else {  
    $retval = $defval;  
  }  
  
  return $retval;  
}  
?>
```

Esta función nos permite registrar variables que tu esperabas que fueran pasadas por tu código. Normalmente se usa esto utilizando \$mode = register('mode');.La función esta definida con los siguientes parámetros en el php.ini, y además devolverá un resultado idéntico que con



register\_globals on. Esta función permite especificar el valor por defecto con el que te gustaría inicializar las variables si el valor no se encuentra en los arrays superglobales.

Esta función tiene un problema que siempre devuelve un valor y además siempre inicializa las variables con algo. Si solemos utilizar isset() para determinar que un valor ha sido pasado podemos utilizar esta otra función para registrar los valores.

```
/**
 * fija una variable global si la
 * especificada get o post existe
 *
 * @param string $test_vars
 * el array de las variables
 * a registrar, aceptar una cadena como
 * nombre para una unica variable
 *
 * @global la variable, si esta está fijada
 */
function getpost_ifset($test_vars)
{
    if (is_array($test_vars)) {
        $test_vars = array($test_vars);
    }

    foreach($test_vars as $test_var) {
        if (isset($_POST[$test_var])) {
            global $$test_var;
            $$test_var = $_POST[$test_var];
        } elseif (isset($_GET[$test_var])) {
            global $$test_var;
            $$test_var = $_GET[$test_var];
        }
    }
}
?>
```

Esta función nos permitira pasar un array de variables para registrar. Si alguna de las variables fueron pasadas por el método GET o POST, serán fijadas como globales también podrás chequearlas usando isset() para comprobar si fueron pasadas.