

- [Manual de PHP](#)
- [Referencia de funciones](#)
- [Extensiones de bases de datos](#)
- [Capas abstractas](#)
- [PDO](#)

Change language: Spanish ▼

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

Constantes predefinidas ¶

Estas constantes están definidas por esta extensión y estarán disponibles sólo cuando la extensión haya sido compilada con PHP, o bien sea cargada dinámicamente en ejecución.

Advertencia

PDO utiliza constantes de clase desde PHP 5.1. Las versiones anteriores emplean constantes globales bajo la forma `PDO_PARAM_BOOL`.

PDO::PARAM_BOOL (integer)

Representa un tipo de dato booleano.

PDO::PARAM_NULL (integer)

Representa el tipo de dato NULL de SQL.

PDO::PARAM_INT (integer)

Representa el tipo de dato INTEGER de SQL .

PDO::PARAM_STR (integer)

Representa el tipo de dato CHAR, VARCHAR de SQL, u otro tipo de datos de cadena.

PDO::PARAM_STR_NATL (integer)

Disponible desde PHP 7.2.0

PDO::PARAM_STR_CHAR (integer)

Disponible desde PHP 7.2.0

PDO::PARAM_LOB (integer)

Representa el tipo de dato de objeto grande (LOB) de SQL.

PDO::PARAM_STMT (integer)

Representa un tipo de conjunto de registros. No está soportado actualmente por ningún controlador.

PDO::PARAM_INPUT_OUTPUT (integer)

Especifica que el parámetro es de entrada/salida (INOUT) para un procedimiento almacenado. Se debe realizar un OR a nivel de bit con un tipo de datos PDO::PARAM_* explícito.

PDO::FETCH_LAZY (integer)

Especifica que el método de obtención debe devolver cada fila como un objeto con los nombres de las variables equivalentes a los nombres de las columnas devueltos en el conjunto de resultados. **PDO::FETCH_LAZY** crea los nombres de las variables del objeto a medida que se acceden a ellas. No es válida dentro de [PDOStatement::fetchAll\(\)](#).

PDO::FETCH_ASSOC (integer)

Especifica que el método de obtención debe devolver cada fila como un array indexado por los nombres de las columnas devueltos en el correspondiente conjunto de resultados. Si éste contiene varias columnas con el mismo nombre, **PDO::FETCH_ASSOC** devuelve un único valor por nombre de columna.

PDO::FETCH_NAMED (integer)

Especifica que el método de obtención debe devolver cada fila como un array indexado por los nombres de las columnas devueltos en el correspondiente conjunto de resultados. Si éste contiene varias columnas con el mismo nombre, **PDO::FETCH_NAMED** devuelve un array de valores por nombre de columna.

PDO::FETCH_NUM (integer)

Especifica que el método de obtención debe devolver cada fila como un array indexado por los números de columna devueltos en el correspondiente conjunto de resultados, comenzando por la columna 0.

PDO::FETCH_BOTH (integer)

Especifica que el método de obtención debe devolver cada fila como un array indexado tanto por los nombres como por los números de las columnas devueltos en el correspondiente conjunto de resultados, comenzando por la columna 0.

PDO::FETCH_OBJ (integer)

Especifica que el método de obtención debe devolver cada fila como un objeto con los nombres de sus propiedades equivalentes a los nombres de las columnas devueltos en el conjunto de resultados.

PDO::FETCH_BOUND (integer)

Especifica que el método de obtención debe devolver TRUE y asignar los valores de las columnas del conjunto de resultados a las variables de PHP a las cuales están vinculadas con los métodos [PDOStatement::bindParam\(\)](#) o [PDOStatement::bindColumn\(\)](#).

PDO::FETCH_COLUMN (integer)

Especifica que el método de obtención debe devolver una única columna solicitada de la siguiente fila del conjunto de resultados.

PDO::FETCH_CLASS (integer)

Especifica que el método de obtención debe devolver una nueva instancia de la clase solicitada, haciendo corresponder las columnas con los nombres de las propiedades de la clase.

Nota: El método mágico [__set\(\)](#) se invoca si la propiedad no existe en la clase solicitada.

PDO::FETCH_INTO (integer)

Especifica que el método de obtención debe actualizar una instancia existente de la clase solicitada, haciendo corresponder las columnas con los nombres de las propiedades de la clase.

PDO::FETCH_FUNC (integer)

Permite personalizar completamente la forma en que los datos son tratados sobre la marcha (únicamente válida dentro de [PDOStatement::fetchAll\(\)](#)).

PDO::FETCH_GROUP (integer)

Grupo devuelto según valores. Usualmente combinada con **PDO::FETCH_COLUMN** o **PDO::FETCH_KEY_PAIR**.

PDO::FETCH_UNIQUE (integer)

Obtener solamente los valores únicos.

PDO::FETCH_KEY_PAIR (integer)

Obtener un resultado de dos columnas como un array donde la primera columna es la clave y la segunda el valor. Disponible desde PHP 5.2.3.

PDO::FETCH_CLASSTYPE (integer)

Determinar el nombre de la clase desde el valor de la primera columna.

PDO::FETCH_SERIALIZE (integer)

Igual que **PDO::FETCH_INTO** salvo que el objeto es proporcionado como una cadena serializada. Disponible desde PHP 5.1.0. Desde PHP 5.3.0, nunca se llama al constructor de la clase si está establecido este indicador.

PDO::FETCH_PROPS_LATE (integer)

Llamar al constructor antes de establecer propiedades. Disponible desde PHP 5.2.0

PDO::ATTR_AUTOCOMMIT (integer)

Si este valor es **false**, PDO intenta desactivar la autoconsigna para que la conexión comience una transacción.

PDO::ATTR_PREFETCH (integer)

Establecer el tamaño de la precarga permite obtener un equilibrio entre rapidez y utilización de memoria para la aplicación. No todas las combinaciones de base de datos y controlador permiten establecer el tamaño de la precarga. Un valor más grande de precarga da como resultado un aumento del rendimiento a costa de un mayor uso de memoria.

PDO::ATTR_TIMEOUT (integer)

Establece el valor del tiempo de espera, en segundos, para las comunicaciones con la base de datos.

PDO::ATTR_ERRMODE (integer)

Véase la sección [Errores y su manejo](#) para obtener más información sobre este atributo.

PDO::ATTR_SERVER_VERSION (integer)

Este es un atributo de sólo lectura; devuelve información sobre la versión del servidor de bases de datos al que está conectado PDO.

PDO::ATTR_CLIENT_VERSION (integer)

Este es un atributo de sólo lectura; devuelve información sobre la versión de las bibliotecas cliente que está usando el controlador PDO.

PDO::ATTR_SERVER_INFO (integer)

Este es un atributo de sólo lectura; devuelve alguna metainformación sobre el servidor de bases de datos al que está conectado PDO.

PDO::ATTR_CONNECTION_STATUS (integer)

PDO::ATTR_CASE (integer)

Forzar a los nombres de las columnas a emplear las mayúsculas/minúsculas especificadas por las constantes **PDO::CASE_***.

PDO::ATTR_CURSOR_NAME (integer)

Obtener o establecer el nombre del cursor a utilizar. Es más útil cuando se emplean cursores desplazables y actualizaciones posicionadas.

PDO::ATTR_CURSOR (integer)

Selecciona el tipo de cursor. PDO actualmente admite **PDO::CURSOR_FWDONLY** y **PDO::CURSOR_SCROLL**. Cíñase a **PDO::CURSOR_FWDONLY** a menos que necesite un cursor desplazable.

PDO::ATTR_DRIVER_NAME (string)

Devuelve el nombre del controlador.

Ejemplo #1 Uso de **PDO::ATTR_DRIVER_NAME**

```
<?php
if ($bd->getAttribute(PDO::ATTR_DRIVER_NAME) == 'mysql') {
    echo "Ejecutándose sobre mysql; hacer algo específico de mysql aquí\n";
}
```

```
}  
?>
```

PDO::ATTR_ORACLE_NULLS (integer)

Convertir string vacíos a valores NULL de SQL en la obtención de datos.

PDO::ATTR_PERSISTENT (integer)

Solicitar una conexión persistente, en vez de crear una nueva conexión. Véase [Conexiones y su administración](#) para obtener más información sobre este atributo.

PDO::ATTR_STATEMENT_CLASS (integer)

PDO::ATTR_FETCH_CATALOG_NAMES (integer)

Anteponer el nombre del catálogo contenedor a cada nombre de columna devuelto en el conjunto de resultados. El nombre del catálogo y el nombre de columna están separados por un carácter punto (.). El soporte de este atributo es a nivel del controlador; podría no estar admitido por el controlador que se esté utilizando.

PDO::ATTR_FETCH_TABLE_NAMES (integer)

Anteponer el nombre de la tabla contenedora a cada nombre de columna devuelto en el conjunto de resultados. El nombre de la tabla y el nombre de columna están separados por un carácter punto (.). El soporte de este atributo es a nivel del controlador; podría no estar admitido por el controlador que se esté utilizando.

PDO::ATTR_STRINGIFY_FETCHES (integer)

PDO::ATTR_MAX_COLUMN_LEN (integer)

PDO::ATTR_DEFAULT_FETCH_MODE (integer)

Disponible desde PHP 5.2.0.

PDO::ATTR_EMULATE_PREPARES (integer)

Disponible desde PHP 5.1.3.

PDO::ATTR_DEFAULT_STR_PARAM (integer)

Disponible desde PHP 7.2.0.

PDO::ERRMODE_SILENT (integer)

No emitir ningún error o excepción si ocurrieran. El desarrollador es responsable de comprobar explícitamente los errores. Este es el modo predeterminado. Véase [Errores y su manejo](#) para obtener más información sobre este atributo.

PDO::ERRMODE_WARNING (integer)

Emitir un mensaje **E_WARNING** de PHP si ocurre algún error. Véase [Errores y su manejo](#) para obtener más información sobre este atributo.

PDO::ERRMODE_EXCEPTION (integer)

Lanzar una [PDOException](#) si ocurre algún error. Véase [Errores y su manejo](#) para obtener más información sobre este atributo.

PDO::CASE_NATURAL (integer)

Dejar los nombres de las columnas tal como son devueltos por el controlador de la base de datos.

PDO::CASE_LOWER (integer)

Forzar los nombres de las columnas a minúsculas.

PDO::CASE_UPPER (integer)

Forzar los nombres de las columnas a mayúsculas.

PDO::NULL_NATURAL (integer)

PDO::NULL_EMPTY_STRING (integer)

PDO::NULL_TO_STRING (integer)

PDO::FETCH_ORI_NEXT (integer)

Obtener la siguiente fila del conjunto de resultados. Únicamente válida para cursores desplazables.

PDO::FETCH_ORI_PRIOR (integer)

Obtener la fila anterior del conjunto de resultados. Únicamente válida para cursores desplazables.

PDO::FETCH_ORI_FIRST (integer)

Obtener la primera fila del conjunto de resultados. Únicamente válida para cursores desplazables.

PDO::FETCH_ORI_LAST (integer)

Obtener la última fila del conjunto de resultados. Únicamente válida para cursores desplazables.

PDO::FETCH_ORI_ABS (integer)

Obtener la fila solicitada del conjunto de resultados mediante el número de fila. Únicamente válida para cursores desplazables.

PDO::FETCH_ORI_REL (integer)

Obtener la fila solicitada del conjunto de resultados mediante la posición relativa de la posición actual del cursor. Únicamente válida para cursores desplazables.

PDO::CURSOR_FWDONLY (integer)

Crear un objeto [PDOStatement](#) con un cursor de sólo avance. Este es la elección del cursor predeterminada, ya que es el patrón de acceso a datos más rápido y común en PHP.

PDO::CURSOR_SCROLL (integer)

Crear un objeto [PDOStatement](#) con un cursor desplazable. De deben pasar las constantes `PDO::FETCH_ORI_*` para controlar las filas obtenidas del conjunto de resultados.

PDO::ERR_NONE (string)

Corresponde al SQLSTATE '00000', el cual significa que la sentencia SQL fue ejecutada con éxito, sin errores ni advertencias. Esta constante sirve para la comprobación de [PDO::errorCode\(\)](#) o [PDOStatement::errorCode\(\)](#) para determinar si ha ocurrido algún error. En cualquier caso, normalmente se sabrá si este es el caso examinando el código devuelto por el método que provocó la condición de error.

PDO::PARAM_EVT_ALLOC (integer)

Evento de asignación

PDO::PARAM_EVT_FREE (integer)

Evento de desasignación

PDO::PARAM_EVT_EXEC_PRE (integer)

Evento provocado antes de la ejecución de una sentencia preparada.

PDO::PARAM_EVT_EXEC_POST (integer)

Evento provocado después de la ejecución de una sentencia preparada.

PDO::PARAM_EVT_FETCH_PRE (integer)

Evento provocado antes de la obtención de un resultado de un conjunto de resultados.

PDO::PARAM_EVT_FETCH_POST (integer)

Evento provocado después de la obtención de un resultado de un conjunto de resultados.

PDO::PARAM_EVT_NORMALIZE (integer)

Evento provocado durante el registro de un parámetro vinculado, permitiendo al controlador normalizar el nombre del parámetro.

[+ add a note](#)

User Contributed Notes 6 notes

[up](#)[down](#)

23

[sam at xnet dot tk ¶](#)**9 years ago**

It should be noted that PDO::FETCH_CLASS will call the constructor *after* setting the values (or calling __set).

[up](#)[down](#)

5

[todd at toddwiggins dot com dot au ¶](#)**2 years ago**

Reference to all the PDO::MYSQL_* constants is available in MySQL's documentation for the driver here:

<https://dev.mysql.com/doc/connectors/en/apis-php-pdo-mysql.html>

Such as the ones I was looking for PDO::MYSQL_ATTR_SSL_CA and PDO::MYSQL_ATTR_SSL_CAPATH which are not listed on this page.

[up](#)[down](#)

4

[kevin at kevinlocke dot name ¶](#)**3 years ago**

PDO::PARAM_STR_CHAR and PDO::PARAM_STR_NATL must be combined with PDO::PARAM_STR using bitwise-OR for parameter binding.

These flags control value quoting (e.g. PDO::quote) and, in some situations (see below), parameter binding (e.g. PDO::bindParam, PDO::bindValue) to prefix string literals with N'' as defined in SQL-92. As of PHP 7.3, only dblib and mysql support these flags. For the mysql driver, the flags only affect parameter binding when PDO::ATTR_EMULATE_PREPARES is true (the default).

MySQL and MariaDB interpret string literals prefixed with N as being utf8 (not utf8mb4) regardless of `SET NAMES` or the charset parameter. This can cause problems if the database/table/column charset is not utf8. For example, in a database using utf8mb4, the query "SELECT * FROM table WHERE col = :param" and bindValue(":param", "\u{1F600}", PDO::PARAM_STR | PDO::PARAM_STR_NATL) will cause "PDOException: SQLSTATE[HY000]: General error: 1267 Illegal mix of collations (utf8mb4_general_ci,IMPLICIT) and (utf8_general_ci,COERCIBLE) for operation '='". Using PDO::PARAM_STR without PDO::PARAM_STR_NATL and ensuring the charset DSN parameter is set correctly can avoid this issue.

See:

<https://wiki.php.net/rfc/extended-string-types-for-pdo>

<https://mariadb.com/kb/en/library/string-literals/>

<https://dev.mysql.com/doc/refman/8.0/en/string-literals.html>

[up](#)[down](#)

2

[Sbastien ¶](#)**2 years ago**

PDO::FETCH_UNIQUE not only fetches the unique values, it also uses the first SQL column as array key result, what is very useful for create quickly an index, eg :

<?php

```
$sql = <<<SQL
    SELECT ALL
        c1, -- For result indexing
        c1, c2
    FROM (
        VALUES
            ROW('ID-1', 'Value 1'),
            ROW('ID-2', 'Value 2a'),
            ROW('ID-2', 'Value 2b'),
            ROW('ID-3', 'Value 3')
    ) AS t (c1, c2);
SQL;
$result = $pdo->query($sql);
print_r($result->fetchAll(PDO::FETCH_UNIQUE));
```

/*

Gives :

ID-1 => [c1 => ID-1, c2 => Value 1]

ID-2 => [c1 => ID-2b, c2 => Value 2b]

ID-3 => [c1 => ID-3, c2 => Value 3]

*/

?>

[up](#)[down](#)

3

[Oleg Andreyev ¶](#)**6 years ago**

Default value for \PDO::ATTR_TIMEOUT is 30 seconds.

Ref.: https://github.com/php/php-src/blob/PHP-7.1.0/ext/pdo_mysql/mysql_driver.c#L600

[up](#)

[down](#)

-6

[ko dot izawa at gmail dot com ¶](#)**6 years ago**

When you use REAL type column on SQLite, your setting of pdo parameter of bindValue() is PDO::PARAM_STR.

'''

```
$stmt->bindValue(':elapsed_time', $e_time, PDO::PARAM_STR);
```

'''

[+ add a note](#)

- [PDO](#)
 - [Introducción](#)
 - [Instalación/Configuración](#)
 - [Constantes predefinidas](#)
 - [Conexiones y su administración](#)
 - [Transacciones y autoconsigna \("auto-commit"\)](#)
 - [Sentencias preparadas y procedimientos almacenados](#)
 - [Errores y su manejo](#)
 - [Objetos grandes \(LOB\)](#)
 - [PDO](#)
 - [PDOStatement](#)
 - [PDOException](#)
 - [Controladores de PDO](#)
- [Copyright © 2001-2023 The PHP Group](#)
- [My PHP.net](#)
- [Contact](#)
- [Other PHP.net sites](#)
- [Privacy policy](#)

