

PDOSTATEMENT

Funcionalidad y utilidad

- Sirve para el manejo de registros de una base de datos en un programa escrito en php
- Representa un conjunto de resultados (result set) de una consulta realizada a una base de datos
- Se genera al usar los métodos prepare(), query() o exec() de la clase PDO

Propiedades

- `$queryString`
 - ✓ Propiedad: publica
 - ✓ Tipo: String

Métodos

- `PDOStatement::fetch`
 - ✓ Obtiene una fila de un conjunto de resultados con un array y avanza el puntero por el conjunto de registros
 - ✓ Parámetros:
 - `$mode` (opcional)
 - Controla cómo se devolverá la siguiente fila a la persona que llama
 - Tiene que tener el valor de alguna de estas constantes:
 - ✓ `PDO::FETCH_OBJ`(funciona igual que `fetchObject()`)
 - ✓ `PDO::FETCH_BOTH`(predeterminado)
 - ✓ `PDO::FETCH_ASSOC`
 - ✓ `PDO::FETCH_NAMED`
 - ✓ `PDO::FETCH_NUM`
 - `$cursorOrientation` (opcional)
 - Determina qué fila se devolverá del *result set*
 - Tipo: int
 - `$cursorOffset` (opcional)
 - Especifica el número absoluto de la fila en el conjunto de resultados
 - Tipo: int
 - ✓ Returns: depende del valor de `$mode`, en el caso de ausencia, devuelve un array asociativo con los valores y los campos de la consulta. Devuelve *null* cuando termina de recorrer el conjunto de resultados.
 - ✓ Lanza una `PDOException` si el atributo `PDO::ATTR_ERRMODE` está establecido en `PDO::ERRMODE_EXCEPTION`
 - ✓ Ejemplos:
 - Código con `PDO::FETCH_OBJ`

```
<?php
$gsent = $gbd->prepare("SELECT name, colour FROM fruit");
$gsent->execute();
```

```

print("PDO::FETCH_OBJ: ");
print("Devolver la siguiente fila como un objeto anónimo con nombres de columna como
propiedades\n");
$result = $gsent->fetch(PDO::FETCH_OBJ);
print $result->name;
print("\n");
?>

```

→ Resultado con PDO::FETCH_OBJ

```

PDO::FETCH_OBJ: Devolver la siguiente fila como un objeto anónimo con nombres de columna
como propiedades
kiwi

```

→ Código con PDO::FETCH_BOTH

```

<?php
$gsent = $gbd->prepare("SELECT name, colour FROM fruit");
$gsent->execute();

print("PDO::FETCH_BOTH: ");
print("Devolver la siguiente fila como un array indexado por nombre y número de columna\n");
$result = $gsent->fetch(PDO::FETCH_BOTH);
print_r($result);
print("\n");

```

→ Resultado con PDO::FETCH_BOTH

```

PDO::FETCH_BOTH: Devolver la siguiente fila como un array indexado por nombre y número de
columna
Array
(
    [name] => banana
    [0] => banana
    [colour] => yellow
    [1] => yellow
)

```

- **PDOStatement::fetchObject**

- ✓ Instancia un objeto PDOStatement que almacena un registro y avanza el puntero por el conjunto de registros

- ✓ Parámetros:

→ \$class (opcional)

- Nombre de la clase creada
- Tipo: String

→ \$constructorArgs (opcional)

- Los elementos de esta matriz se pasan al constructor
- Tipo: array

- ✓ Returns: Devuelve un objeto PDOStatement con propiedades que corresponden a los nombres de las columnas (Ejemplo: *\$Usuario->T01_Nombre*). Devuelve *false* en caso de error.

- ✓ Emite un error con el nivel *E_WARNING* si el atributo *PDO::ATTR_ERRMODE* está establecido en *PDO::ERRMODE_WARNING*.

- ✓ **Ejemplos:**

- **PDOStatement::bindParam**

- ✓ Bindea un parámetro al nombre de variable especificado, es decir, asigna un valor a un parámetro que aparece en la consulta SQL con una ? o con 2 puntos seguido de una cadena de caracteres
- ✓ Parámetros:
 - \$parameter (obligatorio)
 - Si la consulta SQL está construida con ?, recibe el numero relativo a la posicion de la interrogación a la que queremos asignarle el valor (\$variable)
 - Si la consulta SQL está construida con : seguido de una cadena, recibe la cadena con los 2 puntos a la que queremos asociarle el valor (\$variable)
 - Tipo: String
 - \$variable (obligatorio)
 - Nombre de la variable que contiene el valor que le queremos asignar al parámetro (\$parameter)
 - Tipo: variable php (cualquier tipo)
 - \$data_type (opcional)
 - El tipo de datos explícito para el parámetro usando las constantes *PDO::PARAM_** (parámetro por defecto: *PDO::PARAM_STR*)
 - Tipo: int
 - \$length (opcional)
 - Hace referencia a la longitud del tipo de datos
 - Tipo: int
- ✓ Returns: Devuelve *true* en caso de éxito o *false* en caso de error
- ✓ **Ejemplos:**

→ Consulta con interrogaciones

```
<?php
/* Ejecutar una sentencia preparada vinculando variables de PHP */
$calorías = 150;
$color = 'red';
$gsent = $gbd->prepare('SELECT name, colour, calories
    FROM fruit
    WHERE calories < ? AND colour = ?');
$gsent->bindParam(1, $calorías, PDO::PARAM_INT);
$gsent->bindParam(2, $color, PDO::PARAM_STR, 12);
$gsent->execute();
?>
```

→ Consulta con :cadena

```
<?php
/* Ejecutar una sentencia preparada vinculando variables de PHP */
$calorias = 150;
$color = 'red';
$gsent = $gbd->prepare('SELECT name, colour, calories
    FROM fruit
    WHERE calories < :calories AND colour = :colour');
$gsent->bindParam(':calories', $calorias, PDO::PARAM_INT);
$gsent->bindParam(':colour', $color, PDO::PARAM_STR, 12);
$gsent->execute();
?>
```

- **PDOStatement::execute**

- ✓ Ejecuta una consulta preparada. Debe utilizarse después de **PDO::prepare** y de **PDOStatement::bindParam** para ejecutar la consulta con los parámetros bindeados

- ✓ **Parámetros:**

→ \$input parameters (opcional)

- Un array de valores con tantos elementos como parámetros vinculados en la sentencia SQL que va a ser ejecutada.
- Tipo: array

- ✓ Returns: Devuelve *true* en caso de éxito o *false* en caso de error

- ✓ **Ejemplos:**

→ Con parámetro

```
<?php
/* Ejecutar una sentencia preparada proporcionando un array de valores de inserción */
$calorias = 150;
$color = 'red';
$gsent = $gbd->prepare('SELECT name, colour, calories
    FROM fruit
    WHERE calories < :calories AND colour = :colour');
$gsent->execute(array(':calories' => $calorias, ':colour' => $color));
?>
```

→ Sin parámetro

```
<?php
/* Ejecutar una sentencia preparada vinculando una variable y un valor */
$calorias = 150;
$color = 'gre';
$gsent = $gbd->prepare('SELECT name, colour, calories
    FROM fruit
    WHERE calories < :calorias AND colour LIKE :color');
$gsent->bindParam(':calorias', $calorias, PDO::PARAM_INT);
$sth->bindValue(':colour', "%{$color}%");
$gsent->execute();
?>
```

- **PDOStatement::rowCount**

- ✓ Almacena en una variable el número de filas afectadas por la última declaración SQL cuando ya ha sido preparada y ejecutada
- ✓ **Parámetros:** No recibe parámetros

- ✓ Returns: Devuelve un número entero asociado al número de filas afectadas por la consulta

- ✓ **Ejemplo:**

→ Código

```
<?php
/* Borrar todas las filas de la tabla FRUIT */
$del = $gbd->prepare('DELETE FROM fruit');
$del->execute();

/* Devolver el número de filas que fueron eliminadas */
print("Devolver el número de filas que fueron eliminadas:\n");
$cuenta = $del->rowCount();
print("Eliminadas $cuenta filas.\n");
?>
```

→ Resultado

Devolver el número de filas que fueron eliminadas:
Eliminadas 9 filas.

- **PDOStatement::fetchAll**

- ✓ Recupera las filas restantes de un conjunto de resultados

- ✓ Parámetros:

→ \$fetch_style (opcional)

- Controla el contenido del array devuelto al igual que \$mode de **PDOStatement::fetch** (por defecto es *PDO::FETCH_BOTH*)
- Tipo: int

→ \$fetch_argument (opcional)

- Este argumento tiene un significado diferente dependiendo del valor del parámetro \$fetch_style
- Tipo: depende de \$fetch_style

→ \$ctor_args (opcional)

- Los argumentos del constructor de la clase cuando el parámetro \$fetch_style es *PDO::FETCH_CLASS*
- Tipo: array

- ✓ Returns: Devuelve un array que contiene todas las filas del conjunto de resultados

- ✓ **Ejemplo:**

→ Código

```
<?php
$gsent = $gbd->prepare("SELECT name, colour FROM fruit");
$gsent->execute();

/* Obtener todas las filas restantes del conjunto de resultados */
print("Obtener todas las filas restantes del conjunto de resultados:\n");
$resultado = $gsent->fetchAll();
print_r($resultado);
?>
```

→ Resultado

Obtener todas las filas restantes del conjunto de resultados:

```
Array
(
    [0] => Array
        (
            [name] => pear
            [0] => pear
            [colour] => green
            [1] => green
        )

    [1] => Array
        (
            [name] => watermelon
            [0] => watermelon
            [colour] => pink
            [1] => pink
        )

)
```
