php

□

- Downloads
- Documentation
- Get Involved
- Help

Search

PHPKonf: Istanbul PHP Conference 2017

Getting Started
    Introduction
    A simple tutorial
Language Reference
    Basic syntax
    Types
    Variables
    Constants
    Expressions
    Operators
    Control Structures
    Functions
    Classes and Objects
    Namespaces
    Errors
    Exceptions
    Generators
    References Explained
    Predefined Variables
    Predefined Exceptions
    Predefined Interfaces and Classes
    Context options and parameters
    Supported Protocols and Wrappers

Security
    Introduction
    General considerations
    Installed as CGI binary
    Installed as an Apache module
    Session Security
    Filesystem Security
    Database Security
    Error Reporting
    Using Register Globals
    User Submitted Data
    Magic Quotes
    Hiding PHP
    Keeping Current
Features
    HTTP authentication with PHP
    Cookies
    Sessions

- Dealing with XForms
- Handling file uploads
- Using remote files
- Connection handling
- Persistent Database Connections
- Safe Mode
- Command line usage
- Garbage Collection
- DTrace Dynamic Tracing

Function Reference

- Affecting PHP's Behaviour
- Audio Formats Manipulation
- Authentication Services
- Command Line Specific Extensions
- Compression and Archive Extensions
- Credit Card Processing
- Cryptography Extensions
- Database Extensions
- Date and Time Related Extensions
- File System Related Extensions
- Human Language and Character Encoding Support
- Image Processing and Generation
- Mail Related Extensions
- Mathematical Extensions
- Non-Text MIME Output
- Process Control Extensions
- Other Basic Extensions
- Other Services
- Search Engine Extensions
- Server Specific Extensions
- Session Extensions
- Text Processing
- Variable and Type Related Extensions
- Web Services
- Windows Only Extensions
- XML Manipulation
- GUI Extensions

Keyboard Shortcuts

?
    This help

j
    Next menu item

k
    Previous menu item

g p
    Previous man page

g n
    Next man page

G
    Scroll to bottom

g g
    Scroll to top

g h
Goto homepage
g s
Goto search
(current page)
/
Focus search box

PDOStatement::bindColumn »
« PDO::setAttribute

- PHP Manual
- Function Reference
- Database Extensions
- Abstraction Layers
- PDO

Change language: English ▼

Edit Report a Bug

The PDOStatement class ¶

(PHP 5 >= 5.1.0, PHP 7, PECL pdo >= 1.0.0)

Introduction ¶

Represents a prepared statement and, after the statement is executed, an associated result set.

Class synopsis ¶

PDOStatement implements Traversable {
/* Properties */
readonly string *$queryString*;
/* Methods */
public bool bindColumn (mixed `$column` , mixed `&$param` [, int `$type` [, int `$maxlen` [, mixed `$driverdata`]]])
public bool bindParam (mixed `$parameter` , mixed `&$variable` [, int `$data_type` = PDO::PARAM_STR [, int `$length` [, mixed `$driver_options`]]])
public bool bindValue (mixed `$parameter` , mixed `$value` [, int `$data_type` = PDO::PARAM_STR])
public bool closeCursor (void)
public int columnCount (void)
public void debugDumpParams (void)
public string errorCode (void)
public array errorInfo (void)
public bool execute ([array `$input_parameters`])
public mixed fetch ([int `$fetch_style` [, int `$cursor_orientation` = PDO::FETCH_ORI_NEXT [, int `$cursor_offset` = 0]]])
public array fetchAll ([int `$fetch_style` [, mixed `$fetch_argument` [, array `$ctor_args` = array()]]])
public mixed fetchColumn ([int `$column_number` = 0])
public mixed fetchObject ([string `$class_name` = "stdClass" [, array `$ctor_args`]])
public mixed getAttribute (int `$attribute`)
public array getColumnMeta (int `$column`)

public bool nextRowset (void)
public int rowCount (void)
public bool setAttribute (int `$attribute` , mixed `$value`)
public bool setFetchMode (int `$mode`)
}

Properties ¶

queryString

Used query string.

Table of Contents ¶

⊞ add a note

User Contributed Notes 2 notes

up
down
0
luis-m-cardoso at ext dot ptinovacao dot pt ¶
3 years ago

```
Solved ;)

<?php

$host = "yourHost";
$user = "yourUser";
$pass = "yourPass";
```

```
$db = "yourDB";

$cursor = "cr_123456";

try
{
  $dbh = new PDO("pgsql:host=$host;port=5432;dbname=$db;user=$user;password=$pass");
  echo "Connected<p>";
}
catch (Exception $e)
{
  echo "Unable to connect: " . $e->getMessage() ."<p>";
}

$dbh->beginTransaction();

$query = "SELECT yourFunction(0::smallint,'2013-08-01 00:00','2013-09-01
00:00',1::smallint,'$cursor')";

$dbh->query($query);

$query = "FETCH ALL IN \"$cursor\"";

echo "begin data<p>";

foreach ($dbh->query($query) as $row)
{
    echo "$row[0] $row[1] $row[2] <br>";
}

echo "end data";

?>
```

up
down
-43

rosko at zeta dot org dot au ¶

7 years ago

There are many references around for returning a refcursor from a pgSQL function using pg_query. All essentially boil down to executing the following single statement (or some variation of it):

```
begin; select yourFunction(params...); fetch all in cursorname; commit;
```

In PDO, this doesn't work because PDO won't allow multiple statements submitted as a single statement (due to SQL injection detection). Instead, try this or similar:

```
<?php
$sql = 'select yourFunction(params...)';
$db = new PDO('pgsql:dbname=yourDBname');
$db->beginTransaction();
$cmd = $db->prepare($sql);
if ($cmd->execute()) {
  if ($query = $db->query('fetch all in cursorname')) {
    ...processing...
    $query->closeCursor();
    $cmd->closeCursor();
  }
}
$db->commit();
?>
```

add a note

- PDO
  - Introduction
  - Installing/Configuring
  - Predefined Constants
  - Connections and Connection management
  - Transactions and auto-commit
  - Prepared statements and stored procedures
  - Errors and error handling
  - Large Objects (LOBs)
  - PDO
  - PDOStatement
  - PDOException
  - PDO Drivers

- My PHP.net
- Contact
- Other PHP.net sites
- Mirror sites
- Privacy policy