Fetching Data With Zend_Db

By Quentin Zervaas, 7 April 2010, tagged in PHP, Zend Framework

In this PhpRiot Snippet I will show you a couple of different ways to fetch data when using the Zend Framework's Zend_Db component.

Zend_Db is a useful database abstraction class that comes with the Zend Framework. This article assumes you have already established a database connection and you have tables to select data from. For more information about Zend_Db, visit the manual page at http://framework.zend.com/manual/en/zend.db.html.

I'll assume your database connection is stored in a variable called `$db`.

Fetching a Single Value

If you want to retrieve a single value from the database, use the `fetchOne` method. This is often useful when counting the number of matching rows, as shown in the following listing.

This listing defines a function that checks if a given value exists.

Listing 1. Fetching a single value with fetchOne() (listing-1.php)

```
<?php
    $db = Zend_Db::Factory();

    $query = "select count(*) from myTable where foo = 'bar'";

    if ($db->fetchOne($query) > 0) {
        echo "Value exists";
    }
    else {
        echo "Value does not exist";
    }
?>
```

Fetching a Single Column

Sometimes you want to retrieve only a single column of data. The `fetchCol` method does this for you, returning all values in an array.

The following listing demonstrates this.

Listing 2. Fetching a single column with fetchCol() (listing-2.php)

```
<?php
    $db = Zend_Db::Factory();

    $query = "select distinct tag from tags";

    $tags = $db->fetchCol($query);

    var_dump($tags);
?>
```

Fetching a Pair of Columns

You can fetch two columns of data and turn them into an associative array using the `fetchPairs` method. The first column is used as the array key, while the second column is the array value.

Listing 3. Fetching two columns with fetchPairs() (listing-3.php)

```
<?php
    $db = Zend_Db::Factory();

    $query = "select some_id, url from links";

    $links = $db->fetchPairs($query);
?>
```

Fetching All Data

Finally, if you want to turn all data returned from a query into a single array, you can use the `fetchAll` method. Be careful though - if you return a large number of rows from your query your array may use a lot of memory.

Each array entry corresponds to a single database row. Each entry is an associative array with the column name as the key to the corresponding value.

Copyright 2010 Quentin Zervaas, All Rights Reserved

Fetching Data With Zend_Db

By Quentin Zervaas, 7 April 2010, tagged in PHP, Zend Framework

Listing 4. Fetching all data with fetchAll() (listing-4.php)

```
<?php
    $db = Zend_Db::Factory();

    $query = "select * from myTable where foo = 'bar'";

    $data = $db->fetchAll($query);
?>
```

Copyright 2010 Quentin Zervaas, All Rights Reserved