Using Return Codes in PHP

By Quentin Zervaas, 23 March 2010, tagged in PHP

In this PhpRiot Snippet I will show you how to execute a command on your server using `system()`. I will show you how to check if the command was successful using its return value.

To demonstrate this functionality we'll write a short script restarts your web server. Before restarting the server it will verify the configuration file is valid.

To execute a command we use the `system()` function. This function accepts as its first argument the command to execute, while the second argument is a variable you pass by reference that the return value is written to.
Note: You can also use functions such as `exec()` or `passthru()` if you need to access output from the program, but to keep things simple we'll ignore these functions.

The following listing shows how you to call `system()`.

Listing 1. Accessing and outputting the return value from system() (listing-1.php)

```
<?php
    system('/path/to/someCommand', $retval);
    echo "Return value: " . $retval;
?>
```

If a command-line program has been written correctly, it will exit with a status code of 0 when the program executes correctly, or it will exit with a non-zero value when something does not work correctly.

The following listing shows a basic check of the return value. The specific error that has occurred will depend on the command being run.

Listing 2. Checking the return value and acting accordingly (listing-2.php)

```
<?php
    system('/path/to/someCommand', $retval);

    if ($retval == 0) {
        echo "Executed correctly\n";
    }
    else {
        echo "Some error occurred: " . $retval;
    }
?>
```

To demonstrate this, we'll write a short script that calls makes use of the `apachectl` program. This is used to control an Apache installation (start / stop / restart the server). If you run this command with an argument of `configtest` it will tell you if the configuration is valid. If you run it with an argument of `graceful` it will restart the server (gracefully - that is, it will complete any active requests).

When you run `apachectl configtest` it will tell you if an error occurs, but additionally, it will exit with a code of 8 (you can see this documented inside the `apachectl` script).

Therefore, you can run this command and check if the configuration file is valid. If it is, we'll restart the server, if not, we won't. The following listing demonstrates this.

Listing 3. Restarting your Apache server if the configuration is valid (listing-3.php)

```
<?php
    $ctl = '/usr/local/apache/bin/apachectl';

    system($ctl . ' configtest', $retval);

    if ($retval == 0) {
        system($ctl . ' graceful');
    }
    else {
        echo "Configuration error, unable to restart\n";
        exit($retval);
    }
?>
```

In this script we chained the return value. That is, when an error occurred we also exited with that return code. That way, if yet another script uses our PHP script they can also check the error code if required.

That's all there is to it! You could extend this script by checking the return value of the second command also, since this call can also fail.

Copyright 2010 Quentin Zervaas, All Rights Reserved