CallbackFilterHandler Book

Laurent Laville

CallbackFilterHandler Book

Laurent Laville

Table of Contents

This complete guide documents CallbackFilterHandler 1.0.0, published on 2015-04-21.

Part I. Getting Started

Chapter 1. Install via Composer

This handler what is not yet part of standard Monolog distribution, is available on Packagist bartlett/monolog-callbackfilterhandler [http://packagist.org/packages/bartlett/monolog-callbackfilterhandler] and as such installable via Composer [http://getcomposer.org/].

```
$ php composer.phar require bartlett/monolog-callbackfilterhandler
```

Chapter 2. Your first filter

We will create an anonymous function to filter only record on `fake error` messages with `exception` as contextual data.

CallbackFilterHandler class constructor accept an array of callback functions. Here is an example of the definition corresponding to our goal.

```
<?php
$filters = array(
    function ($record) {
        if (!array_key_exists('exception', $record['context'])) {
            return false;
        }
        return (preg_match('/fake error/', $record['message']) === 1);
    }
);
```

Chapter 3. Full example

This script used the first filter described previously, on a important notification demo system, while all events are logged to a simple file.

```
<?php

use Bartlett\Monolog\Handler\CallbackFilterHandler;

use Monolog\Logger;
use Monolog\Handler\RotatingFileHandler;
use Monolog\Handler\StreamHandler;

// Create the logger
$logger = new Logger('my_logger');

// Create filter rules
$filters = array(
    function ($record) {
        if (!array_key_exists('exception', $record['context'])) {
            return false;
        }
        return (preg_match('/fake error/', $record['message']) === 1);
    }
);

// Create some handlers
$stream = new RotatingFileHandler(__DIR__ . DIRECTORY_SEPARATOR . 'my_logger.log');
$stream->setFilenameFormat('{filename}-{date}', 'Ymd');

$mailer = new StreamHandler(__DIR__ . DIRECTORY_SEPARATOR . 'notifications.log', Logger:

// add handlers to the logger
$logger->pushHandler($stream);
$logger->pushHandler(new CallbackFilterHandler($mailer, $filters));

// You can now use your logger
$logger->addInfo('My logger is now ready');

$logger->addError('A fake error has occured. Will be logged to file BUT NOT notified by m

try {
    throw new \RuntimeException();

} catch (\Exception $e) {
    $logger->addCritical(
        'A fake error has occured. Will be logged to file AND notified by mail.',
        array('exception' => (string) $e)
    );
}
```

Chapter 4. Summary

Let’s review what we’ve done :

- installed the latest stable version using Composer.
- built your first filter.
- used it with a concrete example.