



链滴

# MongoDB 处理日志

作者: [henryspace](#)

原文链接: <https://ld246.com/article/1590482087090>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

Monolog是php下比较全又容易扩展的记录日志组件。目前有包括Symfony、Laravel、CakePHP 诸多知名php框架都内置了Monolog。Monolog可以把你的日志发送到文件，sockets，收件箱，数据库和各种web服务器上。一些特殊的组件可以给你带来特殊的日志策略。

## PHP框架中引入扩展包

```
# 安装 monolog
composer require monolog/monolog
# 安装 mongodb
composer require mongodb/mongodb
```

## 封装MongoDB和Monolog代码：

```
<?php
/**
 * Monolog + MongoDB 配置日志记录
 */

namespace App\Core\Lib;

use Monolog\Logger;
use MongoDB\Client;
use Monolog\Handler\MongoDBHandler;

class Monolog {

    private static $loggers;

    /**
     * 日志等级
     * @var int
     */
    private static $level = Logger::DEBUG;

    public static function __callStatic($name, $arguments)
    {
        $logger = self::createLogger($name);

        $message      = empty($arguments[0]) ? '' : $arguments[0];
        $context      = empty($arguments[1]) ? [] : $arguments[1];
        $levelName    = empty($arguments[2]) ? $name : $arguments[2];
        $backtraceOffset = empty($arguments[3]) ? 0 : intval($arguments[3]);

        $level = Logger::toMonologLevel($levelName);
        if (!is_int($level)) $level = Logger::INFO;

        // $backtrace数组第$idx元素是当前行，第$idx+1元素表示上一层，另外function、class需再
        上取一个层次
        // PHP7 不会包含'call_user_func'与'call_user_func_array'，需减少一层
        if (version_compare(PCRE_VERSION, '7.0.0', '>=')) {
```

```

        $backtrace = debug_backtrace(DEBUG_BACKTRACE_IGNORE_ARGS);
        $idx      = 0 + $backtraceOffset;
    } else {
        $backtrace = debug_backtrace();
        $idx      = 1 + $backtraceOffset;
    }

    $trace = basename($backtrace[$idx]['file']) . ":" . $backtrace[$idx]['line'];
    if (!empty($backtrace[$idx + 1]['function'])) {
        $trace .= '##';
        $trace .= $backtrace[$idx + 1]['function'];
    }

    $message = sprintf('==> LOG: %s -- %s', $message, $trace);

    return $logger->addRecord($level, $message, $context);
}

/**
 * 创建日志
 * @param $name
 * @return mixed
 */
private static function createLogger($name)
{
    if (empty(self::$loggers[$name])) {
        // 根据业务域名与方法名进行日志名称的确定
        $category = $_SERVER['SERVER_NAME'];
        // 创建日志
        $logger = new Logger($category);
        // 日志相关操作 (mongodb配置信息可配置到配置文件)
        $client = new Client('mongodb://localhost:27017');
        $handler = new MongoDBHandler($client, 'logs', 'prod');
        $logger->pushHandler($handler);

        self::$loggers[$name] = $logger;
    }
    return self::$loggers[$name];
}
}

```

## 使用示例

```

*
* 使用示例，首先在配置文件中配置mongodb的连接URI
* 示例: 'mongodb://localhost:27017'
* 代码逻辑中直接引入命名空间
* use App\Core\Lib\Monolog;
*
* // @param string $channel 一个日志实例通道
* // @param array $arguments 日志内容
* Monolog::debug('test', ["名称:{$storagecount}"]);
* Monolog::info('test', ["名称:{$storagecount}"]);
* Monolog::notice('test', ["名称:{$storagecount}"]);

```

```
* Monolog::warning('test', ["名称:{$storagecount}"]);
* Monolog::error('test', ["名称:{$storagecount}"]);
* Monolog::criticle('test', ["名称:{$storagecount}"]);
* Monolog::alert('test', ["名称:{$storagecount}"]);
* Monolog::emergency('test', ["名称:{$storagecount}"]);
*
```

了解更多Monolog请参见: <https://www.jianshu.com/p/b99dc5c3b760>