

# Let's Log!

**Elton Minetto**

@eminetto

<http://eltonminetto.net>

# About

Graduação em Ciência de Computação pela Unochapecó e especialização em Ciência da Computação pela UFSC/UNOESC.

Autor do livro **Frameworks para Desenvolvimento em PHP**, co-autor do livro **Grid Computing in Research and Education** e autor dos e-books **Zend Framework na prática**, **Zend Framework 2 na prática**, **Iniciando com o Zend Framework 2** e **Doctrine na Prática**.

É co-fundador da **Coderockr** e do **Planrockr**

[eminetto@gmail.com](mailto:eminetto@gmail.com)

- O que gera logs?
  - aplicações
  - infra
  - algumas ferramentas externas : stats, CI, etc
- Qual o propósito?
  - acompanhar ocorrência de eventos
  - debug
  - segurança
  - ações ligadas ao negócio
- Quem usa os logs e para que?
- Ter um grande log ou múltiplos?

# PSR-3

Padrão que descreve uma interface comum para log  
Aumenta a reusabilidade

# PSR-3

A *LoggerInterface* define 8 métodos para escrevermos logs

**Emergency** – o sistema não está operante

**Alert** – ação imediata é requerida

**Critical** – condições críticas

**Error** – erros que não requerem ação imediata mas que devem ser monitorados

**Warning** – ocorrências fora do normal mas que não são erros

**Notice** – eventos normais mais significantes

**Info** – eventos interessantes

**Debug** – informações detalhadas para debug

# Monolog

O Monolog é uma biblioteca compatível com o PSR-3 criada pelo Jordi Boggiano (também criador do Composer)

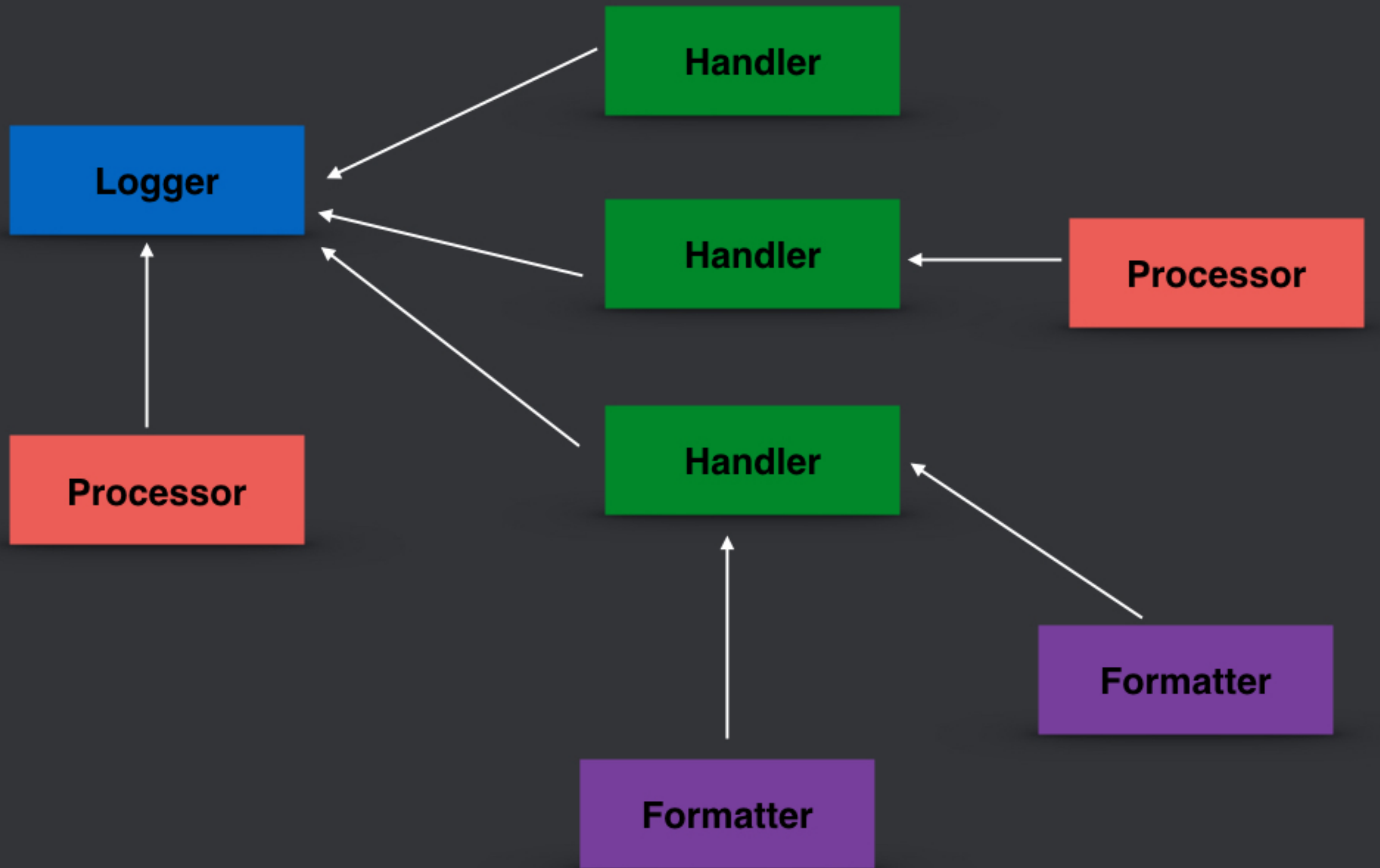
<https://github.com/seldaek/monolog>

Permite enviar logs para arquivos, sockets, e-mail, bases de dados e outros serviços.

"Handlers" especiais permitem construir estratégias avançadas de log

# Arquitetura

- Handler
  - Transporta a informação para o destino
- Formatter
  - Transforma o evento de log em algo que pode ser compreendido pelo handler
- Processor
  - Adiciona informações extras ao evento de log (Memory usage, Request information, Environment details)





# Log Levels

DEBUG (100)

INFO (200)

NOTICE (250)

WARNING (300)

ERROR (400)

CRITICAL (500)

ALERT (550)

EMERGENCY (600)

# Demo

```
composer require monolog/monolog
```

```
<?php

include __DIR__.' /vendor/autoload.php';

// log channel
$log = new Monolog\Logger('SimpleDemoLetsLog');

//handler
$log->pushHandler(new Monolog\Handler\StreamHandler('/tmp/monolog.log', Monolog\Logger::WARNING));
$browserConsoleHandler = new Monolog\Handler\BrowserConsoleHandler();
$browserConsoleHandler->setFormatter(new Monolog\Formatter\JsonFormatter());
$log->pushHandler($browserConsoleHandler);

//processor
$log->pushProcessor(new Monolog\Processor\IntrospectionProcessor());
$log->pushProcessor(new Monolog\Processor\MemoryUsageProcessor());
$log->pushProcessor(new Monolog\Processor\ProcessIdProcessor());
$log->pushProcessor(new Monolog\Processor\WebProcessor());

// add records to the log
$log->warning('Aviso');
$log->error('Erro');

echo "Hello, Monolog!";
```

# Handlers

Salvam log para arquivos e *syslog*

- **StreamHandler:** Salva para streams de PHP, geralmente arquivos
- RotatingFileHandler: Salva em arquivo e automaticamente cria um por dia, apagando os antigos após `$maxFiles`
- SyslogHandler: Envia os registros para o *syslog*.
- ErrorLogHandler: Usa a função nativa do PHP `error_log()`

# Handlers

Enviam alertas e e-mails

- **NativeMailerHandler:** Usa a função nativa *mail()*
- **SwiftMailerHandler:** Envia e-mail usando o *Swift\_Mailer*
- **HipChatHandler:** Envia os registros para o *HipChat*
- **SlackHandler:** Envia os registros para o *Slack*

# Handlers

Enviam registros para servidores de rede

- SocketHandler: Envia usando sockets *UNIX* ou *TCP*
- ZendMonitorHandler: Envia para o *Zend Server*
- AmqpHandler: Envia para o *RabbitMQ* usando a extensão *php-amqp*
- GelfHandler, CubeHandler, RavenHandler, NewRelicHandler, LogglyHandler, RollbarHandler, SyslogUdpHandler, LogEntriesHandler.

# Handlers

Log em ambiente de desenvolvimento

- FirePHPHandler: Envia registros para a extensão *FirePHP* do Firefox
- ChromePHPHandler: Envia registros para o *Chrome*
- **BrowserConsoleHandler**: Envia os dados para o console JavaScript de qualquer navegador, sem a necessidade de extensões

# Handlers

Log em base de dados

- RedisHandler, MongoDBHandler, CouchDBHandler, DoctrineCouchDBHandler, **ElasticSearchHandler**, DynamoDbHandler



# Wrappers / Special Handlers

FingersCrossedHandler, NullHandler, BufferHandler, GroupHandler, FilterHandler, TestHandler, WhatFailureGroupHandler

# Formatters

LineFormatter, HtmlFormatter, NormalizerFormatter, ScalarFormatter,  
JsonFormatter, WildfireFormatter, ChromePHPFormatter,  
GelfMessageFormatter

# Processors

IntrospectionProcessor, WebProcessor, MemoryUsageProcessor, MemoryPeakUsageProcessor, ProcessIdProcessor, UidProcessor, GitProcessor, TagProcessor.

# Demo

<https://github.com/Coderockr/silex-application-log>

```
<?php

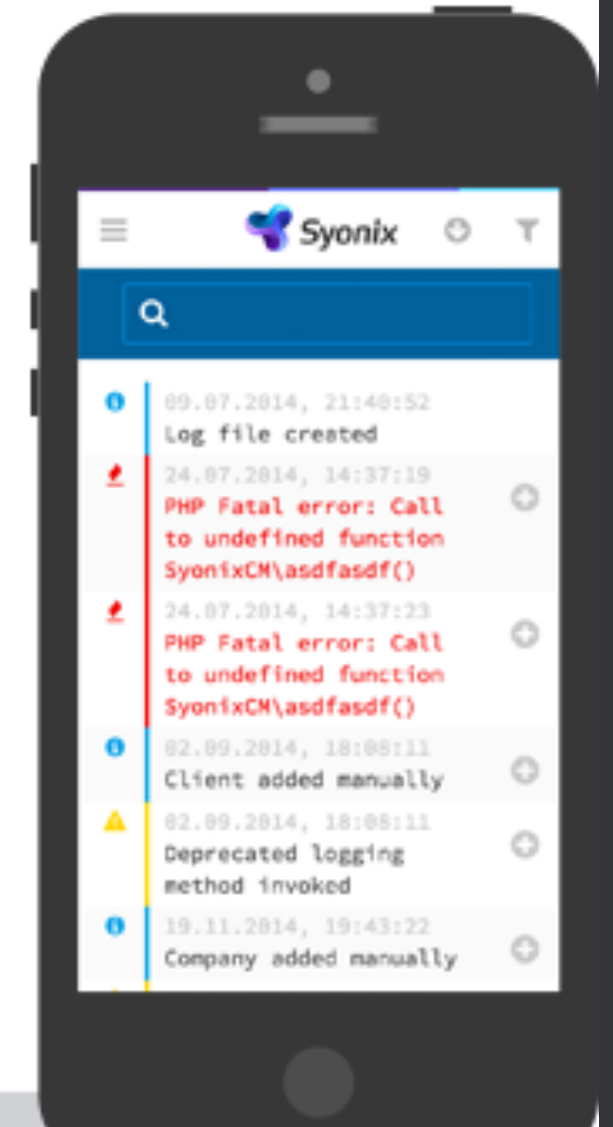
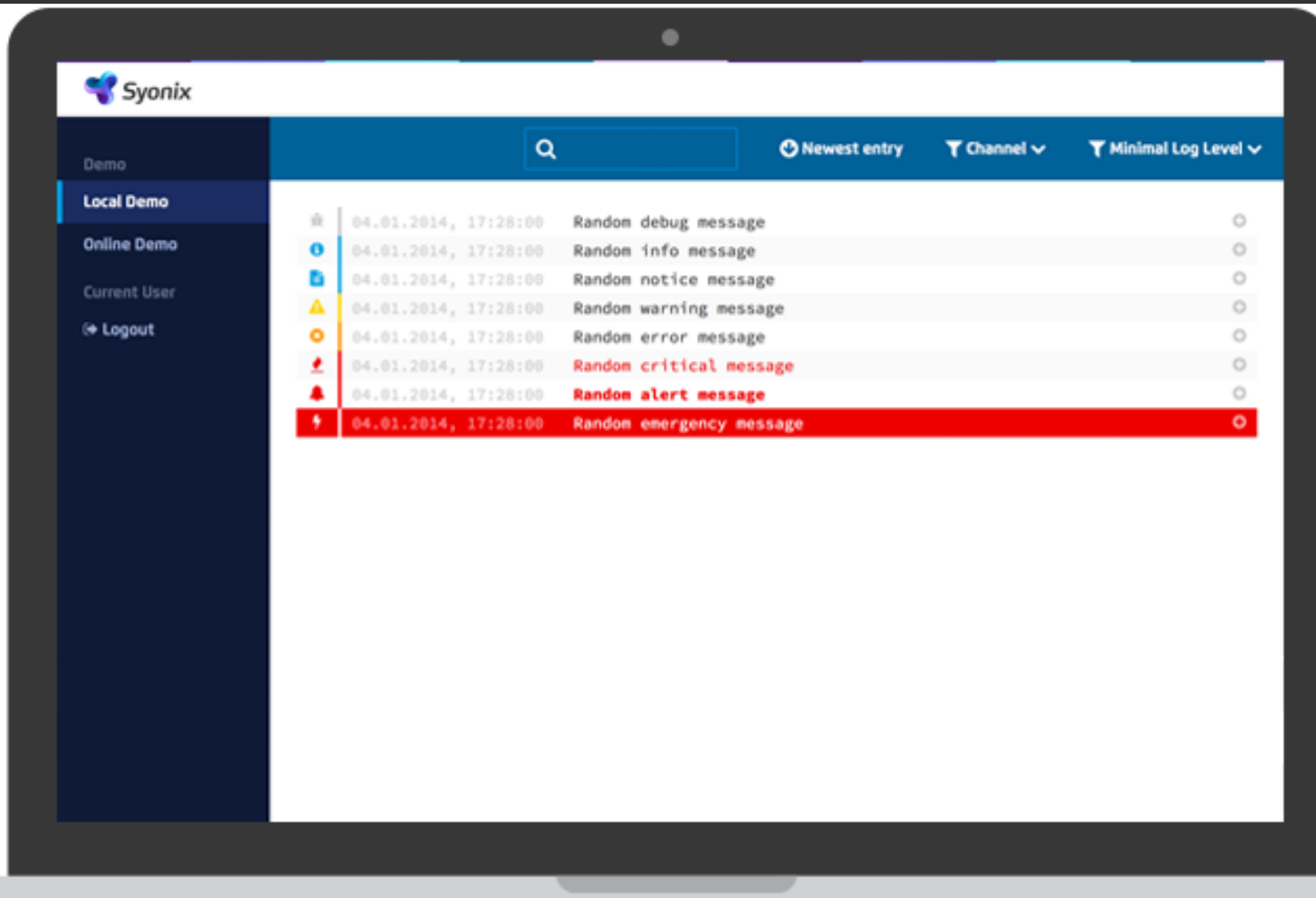
require_once __DIR__.'/vendor/autoload.php';

$app = new Silex\Application();

$config = [
    'config' => [
        'applicationLog' => [
            'name' => 'Channel',
            'processor' => [
                'Monolog\Processor\IntrospectionProcessor',
                'Monolog\Processor\MemoryUsageProcessor',
                'Monolog\Processor\ProcessIdProcessor',
                'Monolog\Processor\WebProcessor',
            ],
            'streamHandler' => [
                'stream' => '/tmp/monolog.log',
                'level' => 'DEBUG',
            ],
        ],
    ],
];

$app->register(new \ApplicationLog\Provider\ApplicationLog(), $config);

$app->run();
```



<https://github.com/Syonix/monolog-viewer>


# Demo

Docker, Silex, coderockr/silex-application-log, monolog-viewer



Look! I've been working on the logs. I set up a new system

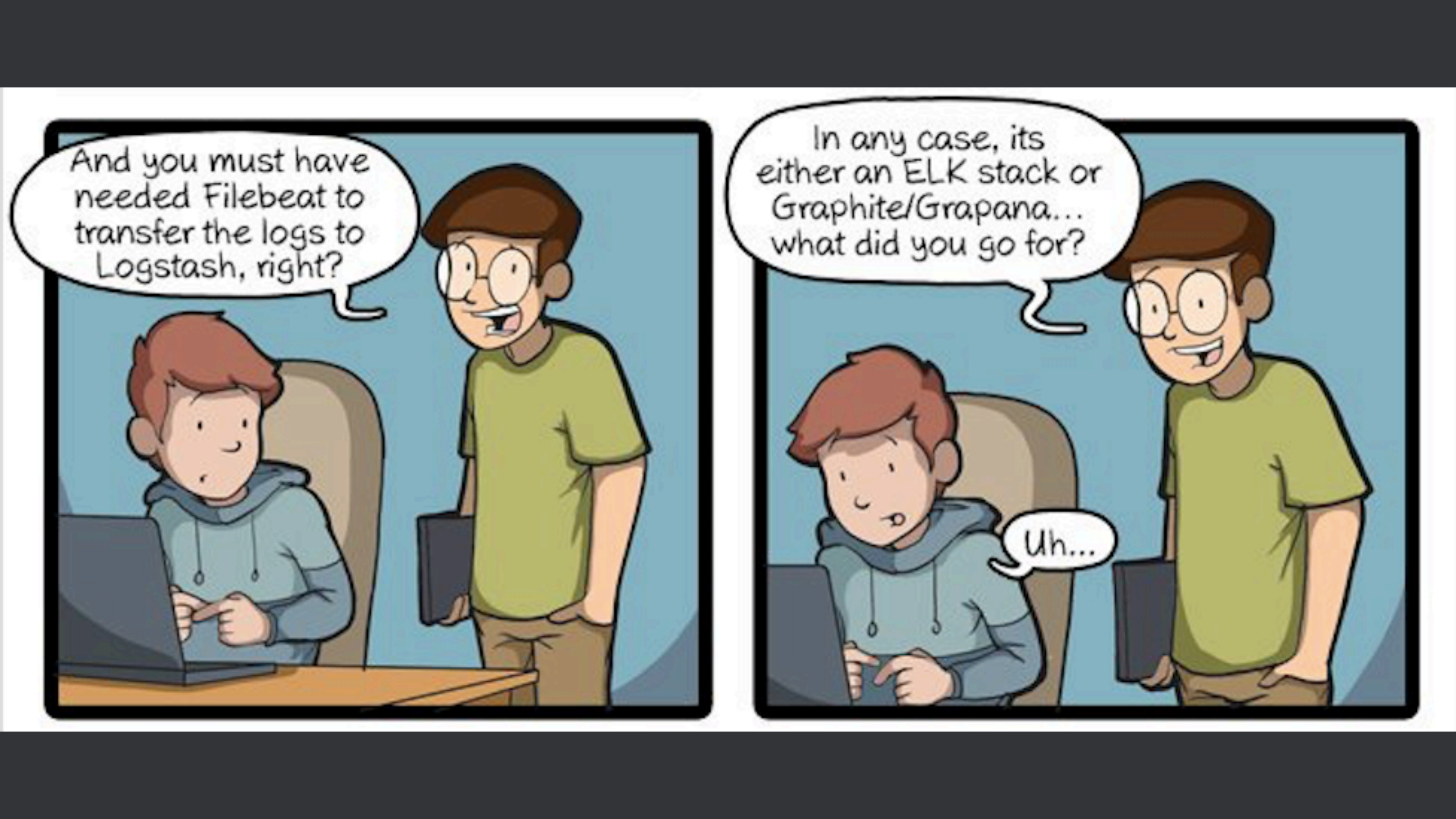
Oh yeah? Let me have a look...



What did you use? Monolog to get the logs? And ElasticSearch, Logstash and Kibana to look through them?

Um...





And you must have needed Filebeat to transfer the logs to Logstash, right?

In any case, its either an ELK stack or Graphite/Grapana... what did you go for?

Uh...



# Referências

<http://slides.com/pvmchau/psr3-monolog#/>

<http://slides.com/grummfy/go-further-with-you-logs-php-monolog#/>

<https://speakerdeck.com/odolbeau/when-monolog-meet-elk>

<https://github.com/Seldaek/monolog/>

<http://www.php-fig.org/psr/psr-3/>

<https://jeremycook.ca/2012/10/02/turbocharging-your-logs/>

<https://github.com/eminetto/palestra-lets-log>