



# THE DEVELOPER'S CONFERENCE

**Logging: Você sabe tudo o que precisa?**

**Charles Willian Coco**



## Charles Willian Coco

- Engenheiro da Computação
- Arquiteto que curte codar
- 13 anos em desenvolvimento Back/Front
- Atualmente interessado/trabalhando com
  - Microsserviços
  - Clean Architecture
  - Pirâmide de Testes
  - Observability

<PEOPLE>  
<COMMUNITIES>  
<KNOWLEDGE>

All together\_



# Porque falar de Logging?



- Não é incomum ver erros de uso dos frameworks de log
- Features/recursos não tão populares
  - AsyncLogger, métodos entry/exit, throwing
  - MDC
- Observability (SRE)
  - Logs, métricas e rastreamento
  - Essencial devido a complexidade dos sistemas atuais
- Analytics



THE  
DEVELOPER'S  
CONFERENCE



# SLF4J

# SLF4J



THE  
DEVELOPER'S  
CONFERENCE

- Simple Logging Facade For Java
  - “API”/Abstração para frameworks de logging
  - Framework muda, mas o código não!
  - Fácil configuração
    - api principal => `org.slf4j:slf4j-api`
    - fmk logging => `org.apache.logging.log4j:log4j-slf4j-impl`

# SLF4J



THE  
DEVELOPER'S  
CONFERENCE

```
private static org.slf4j.Logger LOGGER =  
    org.slf4j.LoggerFactory.getLogger(SearchPersonUseCase.class);  
...  
  
public void metodo() {  
    ...  
    LOGGER.info("...")  
}
```



THE  
DEVELOPER'S  
CONFERENCE



Log4j2

# Log4j2 - Features



## ➤ Substituição de parâmetros

```
LOG.debug("A requisição da URL {} demorou {} segundos", url, duracao);
```

## ➤ Em resposta as deficiências anteriores

```
LOG.debug("A requisição da URL" + url + " demorou " + duracao + " segundos");
```

```
if (LOG.isDebugEnabled()) {  
    LOG.debug("A requisição da URL" + url + " demorou " + duracao + " segundos");  
}
```



# Log4j2 - Features



THE  
DEVELOPER'S  
CONFERENCE

- Suporte a lambda para lazy logging!

```
LOG.debug("{} ", () -> createRequestMessage(request));
```

- Executar? Somente o necessário!

# Log4j2 - Features



THE  
DEVELOPER'S  
CONFERENCE

- Métodos para tracking de fluxo
  - entry / traceEntry
  - exit / traceExit
  - catching
  - throwing
- Suporte slf4j via slf4j-ext

# Log4j2 - Features



THE  
DEVELOPER'S  
CONFERENCE

```
@GetMapping(value = "/person/{person_id}")
public ResponseEntity<PersonModelResponse> searchPersonById(
    @PathVariable(name = "person_id") Long personId) {

    log.entry(personId);

    try {
        ...

        log.exit(result);
        return result;
    } catch (IllegalArgumentException exception) {
        ResponseStatusException responseStatusException
            = new ResponseStatusException(HttpStatus.UNPROCESSABLE_ENTITY, exception.getMessage(), exception);

        log.throwing(responseStatusException);
        throw responseStatusException;
    }
}
```



# Log4j2 - Features



THE  
DEVELOPER'S  
CONFERENCE

## ➤ Loggers assincronos!

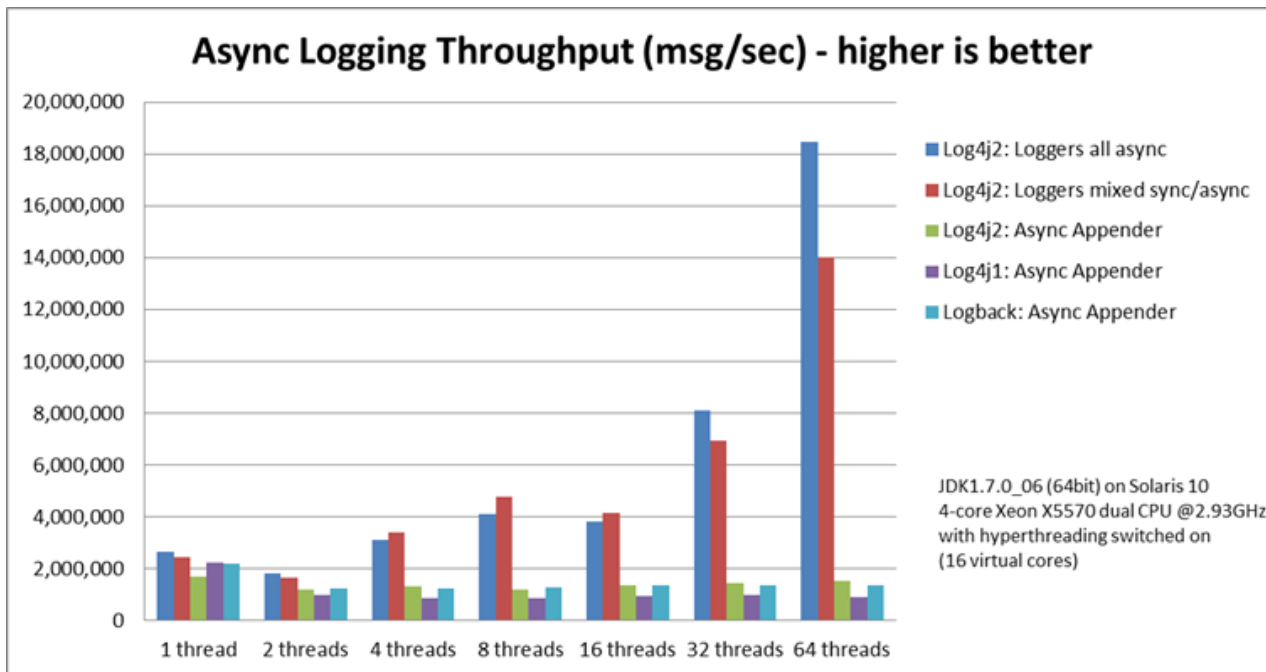
```
<Loggers>
  <AsyncLogger name="com.chawico" level="TRACE"
    additivity="false">
    <AppenderRef ref="stdout" />
    <AppenderRef ref="graylog" />
  </AsyncLogger>
  <Root level="ERROR" includeLocation="true">
    <AppenderRef ref="RandomAccessFile" />
  </Root>
</Loggers>
```

# Log4j2 - Features



THE  
DEVELOPER'S  
CONFERENCE

Alta performance!



Fonte: <https://logging.apache.org/log4j/2.x/performance.html>

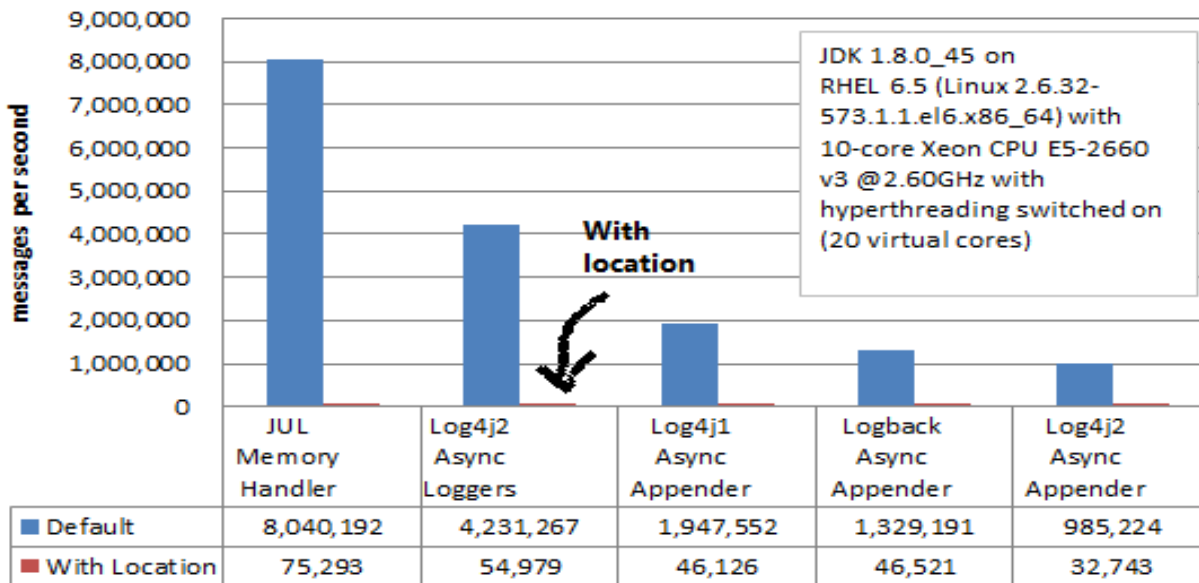
# Log4j2 - Features



THE  
DEVELOPER'S  
CONFERENCE

Mas cuidado na hora de configurar...

## Performance Impact of Capturing Caller Location (higher is better)



Fonte: <https://logging.apache.org/log4j/2.x/performance.html>

# Logging e exceções



THE  
DEVELOPER'S  
CONFERENCE

## Server Error

### **500 - Internal server error.**

There is a problem with the resource you are looking for, and it cannot be displayed.

```
16114 [http-nio-8080-exec-1] ERROR SearchPersonController - Ocorreu o erro null
```

```
try {  
    ...  
} catch (Exception exception) {  
    log.error("Ocorreu o erro " + exception.getMessage());  
}
```

➤ Stacktrace não adicionada ao log

# Logging e exceções



THE  
DEVELOPER'S  
CONFERENCE

```
try {  
    ...  
} catch (Exception exception) {  
    log.error("Ocorreu um erro na operação", exception);  
}
```

```
7354 [http-nio-8080-exec-1] ERROR SearchPersonController - Ocorreu um erro na operação - correlationID= 34432sdfdsf  
java.lang.NullPointerException: null  
    at com.chawico.showcase.bootlog4j2.entrypoint.SearchPersonController.searchPersonById(SearchPersonController.java:42) [classes/:?]  
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) ~[?:1.8.0_212]  
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62) ~[?:1.8.0_212]  
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) ~[?:1.8.0_212]  
    at java.lang.reflect.Method.invoke(Method.java:498) ~[?:1.8.0_212]
```



# Logging e exceções



THE  
DEVELOPER'S  
CONFERENCE

```
} catch (IllegalArgumentException exception) {  
    ResponseStatusException responseStatusException = new ResponseStatusException(HttpStatus.UNPROCESSABLE_ENTITY);  
  
    log.throwing(responseStatusException);  
    throw responseStatusException;  
}
```

```
org.springframework.web.server.ResponseStatusException: 422 UNPROCESSABLE_ENTITY  
    at com.chawico.showcase.bootlog4j2.entrypoint.SearchPersonController.searchPersonById(SearchPersonController.java:51) ~[classes/:?]  
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) ~[?:1.8.0_212]  
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62) ~[?:1.8.0_212]  
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) ~[?:1.8.0_212]
```

➤ Stacktrace original perdida

# Logging e exceções



THE  
DEVELOPER'S  
CONFERENCE

```
} catch (IllegalArgumentException exception) {  
    ResponseStatusException responseStatusException  
        = new ResponseStatusException(HttpStatus.UNPROCESSABLE_ENTITY, exception.getMessage(), exception);  
  
    log.throwing(responseStatusException);  
    throw responseStatusException;  
}
```

Caused by: java.lang.IllegalArgumentException: No null or negative Person identifier value is acceptable!  
at com.chawico.showcase.bootlog4j2.usecase.SearchPersonUseCase.findPersonById(SearchPersonUseCase.java:34) ~[classes/:?]  
at com.chawico.showcase.bootlog4j2.entrypoint.SearchPersonController.searchPersonById(SearchPersonController.java:43) ~[classes/:?]  
... 67 more

➤ Causa original do problema preservada

Até agora está tudo bom... tranquilo mas...  
como ficam microsserviços e sistemas distribuídos?





THE  
DEVELOPER'S  
CONFERENCE



# Graylog

# Graylog



- Gerenciamento de logs centralizado
  - Envio de logs via appender log4j (gelf) ou sidecar
  - Interface gráfica com buscas via elasticsearch
  - Escalável
  - Possui versão open source!

\* gelf = graylog extended logging format



Search in the last 2 days ▾



Not updating ▾

Saved searches ▾



http\_response\_status=422



## Search result

Found **3 messages** in 50 ms, searched in **1 index**.

Results retrieved at 2019-07-16 08:47:19.

Add count to dashboard ▾

Save search criteria

More actions ▾

Fields

Decorators

Default

All

None

Filter fields

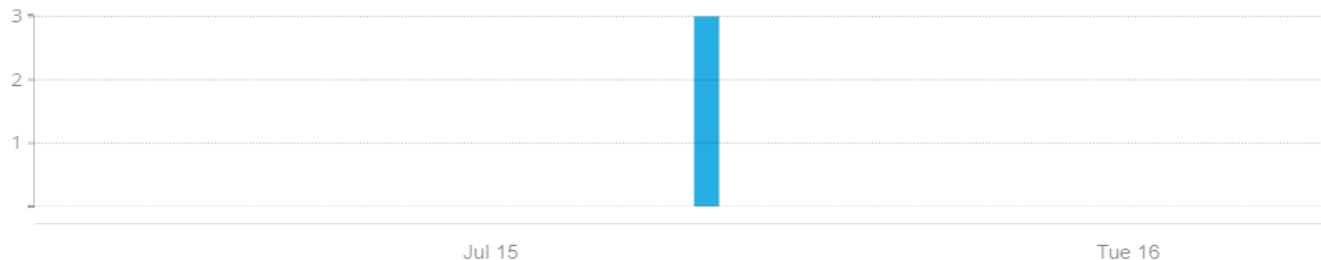
- className
- correlation\_id
- facility
- flow\_id

List fields of **current page** or **all fields**.

## Histogram

Add to dashboard ▾

Year, Quarter, Month, Week, Day, **Hour**, Minute



## Messages

Previous

1

Next



Timestamp 1

source

2019-07-15 09:09:04.632

DESKTOP-HVHBS68

[GET uri=/bootlog4j2/person/-1 completed with STATUS 422 in 9ms](#)

2019-07-15 09:08:51.736

DESKTOP-HVHBS68

[GET uri=/bootlog4j2/person/-1 completed with STATUS 422 in 6ms](#)



THE  
DEVELOPER'S  
CONFERENCE

Legal... mas em um sistema com múltiplos usuários e milhares de linhas de log, como eu sei quem é quem?



# MDC



THE  
DEVELOPER'S  
CONFERENCE

## ➤ Mapped Diagnostic Context

- Contexto por thread
- Comumente utilizado para tracking de requisições
- Configurável para aparecer nos logs
- É fácil de usar!

```
org.slf4j.MDC.put("correlationId", UUID.randomUUID().toString());
```

- Mas não esqueça de limpar!

```
org.slf4j.MDC.remove("correlationId");
```





THE  
DEVELOPER'S  
CONFERENCE



# Zipkin

# Zipkin



- Sistema distribuído para tracking
  - Visão timeline desde origem até fim da requisição
  - Buscas pelo trace ID (correlation ID), nome do serviço, etc
  - Integra-se com spring-cloud-sleuth
    - Adiciona tracking automaticamente a aplicação

# Zipkin



THE  
DEVELOPER'S  
CONFERENCE

Discover

Trace ID:

Please select the criteria for your trace lookup

10  1 HOUR

ef86c83c0a05a6d6 Duration: 38.793ms Services: 3 Depth: 3 Total Spans: 6

Service Name	Span Name	0ms	76.504s	153.009s	229.513s	306.017s
mobile-gateway	get	38.793ms				
auth-service	http:/parent/auth-service/v1/authenticati...	14.077ms				
auth-service	extend-token-life	1.291ms				
content-service	get /content/personal	20.333ms				
content-service	hystrix	863μs				
content-service	get	8.965ms				

# Zipkin



THE  
DEVELOPER'S  
CONFERENCE



# Referências



THE  
DEVELOPER'S  
CONFERENCE

Repo - <https://gitlab.com/chawico>

Observability - <https://www.oreilly.com/library/view/distributed-systems-observability/9781492033431/ch04.html>

SLF4J - <https://www.slf4j.org/>

Log4j2 - <https://logging.apache.org/log4j/2.x/manual/configuration.html>

Graylog - <https://www.graylog.org/>

Gelf - <https://logging.paluch.biz/examples/log4j-2.x.html>

Zipkin - <https://zipkin.io/>

# Resumindo...



THE  
DEVELOPER'S  
CONFERENCE

- SLF4J -> independência de código sobre frameworks de logging
- Log4J2 -> melhor performance, novos métodos para tracking
- MDC -> mapeamento por thread para logging
- Graylog -> centralização de logs
- Zipkin -> sistema distribuído p/ tracking





# THE DEVELOPER'S CONFERENCE

Obrigado!!!



[/in/charlescoco](#)



[/chawico](#)



[/chawico](#)