

21.1. Introdução ao Logback e SLF4J

sábado, 29 de abril de 2023 00:03

O Logback é uma das estruturas de log mais amplamente usadas na Comunidade Java. É um substituto para seu antecessor, Log4j. O Logback oferece uma implementação mais rápida, oferece mais opções de configuração e mais flexibilidade no arquivamento de arquivos de log antigos.

```
@Override
@GetMapping
public PagedModel<CozinhaDTO> listar(@PageableDefault(size = 5) Pageable pageable) {

    logger.info("Quantidade de cozinhas {}", pageable.getPageSize());
}
```

```
1 usage
@RestController
@RequestMapping(value = "/v1/cozinhas", produces = MediaType.APPLICATION_JSON_VALUE)
@Api(tags = "Cozinhas")
public class CozinhaController implements CozinhaControllerOpenApi {

    1 usage
    public static final Logger logger = LoggerFactory.getLogger(CozinhaController.class);

    @Autowired
```

ou utilizar a anotação @Slf4j do lombok

```
4 usages
@Slf4j
@RestController
@RequestMapping(value = "/v1/cozinhas", produces = MediaType.APPLICATION_JSON_VALUE)
@Api(tags = "Cozinhas")
public class CozinhaController implements CozinhaControllerOpenApi {

    @Autowired
```

<https://www.baeldung.com/java-logging-intro>

<https://www.baeldung.com/logback>

21.2. Desafio: registrando logs de exceptions não tratadas

sábado, 29 de abril de 2023

00:26

Resolução do desafio

1 - Alterando ApiExceptionHandler

Vamos adicionar Slf4j na nossa classe, utilizando Lombok

```
@Slf4j
```

```
@ControllerAdvice
```

```
public class ApiExceptionHandler extends ResponseEntityExceptionHandler {
```

```
    //...
```

```
}
```

Agora vamos utilizar o log, onde antes usávamos o "printStackTrace()"

```
log.error(ex.getMessage(), ex);
```

2 - Alterando CozinhaController

No método "listar", vamos adicionar um if apenas para podermos lançar uma Exception no início do método.

```
if (true) {
```

```
    throw new RuntimeException("Teste de exception");
```

```
}
```

21.3. Criando uma conta no Loggly: serviço de gerenciamento de logs na nuvem

sábado, 29 de abril de 2023 00:26

<https://www.loggly.com/>

21.4. Configurando o appender do Loggly no Logback

sábado, 29 de abril de 2023 10:50

Um appender é um componente que tem a função de escrever logs em algum lugar, em arquivos, sites, servidores externos.

```
<?xml version="1.0" encoding="UTF-8" ?>

<configuration>
  <!-- Adicionando Logs padrão do Spring Boot-->
  <include resource="org/springframework/boot/logging/logback/base.xml"/>

  <springProperty name="logglyToken" source="logging.loggly.customer-token"/>
  <!-- Registrando Logs do Loggly.com -->
  <appender name="loggly" class="ch.qos.logback.ext.loggly.LogglyBatchAppender">
    <endpointUrl>https://logs-01.loggly.com/bulk/${logglyToken}/tag/logback</endpointUrl>
    <pattern>%d{"ISO8601", UTC} %p %t %c %M - %m%n</pattern>
  </appender>

  <!-- Appender para registro assíncrono de Logs no Loggly-->
  <appender name="logglyAsync" class="ch.qos.logback.classic.AsyncAppender">
    <appender-ref ref="loggly"/>
  </appender>

  <!-- Registrando logs de nível info para cima-->
  <root level="info">
    <appender-ref ref="logglyAsync"/>
  </root>
</configuration>
```

```
#21.4. Configurando o appender do Loggly no Logback
#logging.loggly.customer-token=seu_token
```

em application.properties

pom.xml

```
<!-- Loggly > Logback > Slf4j -->
<dependency>
  <groupId>org.logback-extensions</groupId>
  <artifactId>logback-ext-loggly</artifactId>
  <version>0.1.5</version>
</dependency>
```

21.5. Configurando o Logback para alternar as configurações por Spring Profiles

sábado, 29 de abril de 2023 17:28

```
<configuration>
  <!-- Adicionando logs padrão do Spring Boot-->
  <include resource="org/springframework/boot/logging/logback/base.xml"/>

  <!--Só é ativado caso o profile "production" estiver ativado-->
  <springProfile name="production">
    <springProperty name="logglyToken" source="logging.loggly.customer-token"/>
    <!-- Registrando logs do Loggly.com -->
    <appender name="loggly" class="ch.qos.logback.ext.loggly.LogglyBatchAppender">
      <endpointUrl>https://logs-01.loggly.com/bulk/${logglyToken}/tag/logback</endpointUrl>
      <pattern>%d{"ISO8601", UTC} %p %t %c %M - %m%n</pattern>
    </appender>

    <!-- Appender para registro assíncrono de logs no Loggly-->
    <appender name="logglyAsync" class="ch.qos.logback.classic.AsyncAppender">
      <appender-ref ref="loggly"/>
    </appender>

    <!-- Registrando logs de nível info para cima-->
    <root level="info">
      <appender-ref ref="logglyAsync"/>
    </root>
  </springProfile>
</configuration>
```