

Logging e Debug

DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

AGENDA

- Revisão do projeto
- Appenders e Layout
- Debug
- Loggly
- Material Complementar

O que é Logging

- Utilizado para mostrar mensagens e rastrear problemas
- Usamos sem perceber, o mais comum é com `System.out` e `System.err`

- **Log4J**
 - API muito difundida
 - Permite bastante flexibilidade
- **SL4J**
 - Utiliza os mesmos conceitos de Log4J
 - É preciso usar alguma API de logging integrada com o SL4J

Funcionamento do Logback

- Para usar o Logback, basta adicionar os JARs necessários no classpath da aplicação
- O Logback pode ser configurado através do arquivo logback.xml
- No spring boot crie o arquivo logback-spring.xml

Funcionamento do Logback

- Logback é um framework de log para aplicações Java, um dos mais usados
- Desenvolvido para ser o sucessor do Log4J
- Logback melhora o desempenho, tem mais opções de configuração
- Em uma aplicação Spring Boot como o projeto agenda-clamed, Logback já é o framework de log padrão, já foi adicionado e configurado automaticamente no projeto
- Com a configuração padrão não é necessário fazer nada para funcionar
- O que vemos no console quando iniciamos a aplicação é graças a configuração padrão do Logback

Loggers

- São objetos utilizados quando há a necessidade de gerar uma informação de log
- Um logger é uma fonte de mensagens de log
- O LOGGER do pacote “org.slf4j” é uma biblioteca que abstrai o código de diversos frameworks de logging
- Funciona como uma fachada, nosso código conversa com slf4j que repassa as informações para framework de log que estamos usando

```
@Service
public class CrudFuncionarioService {

    private static final Logger LOGGER = LoggerFactory.getLogger(CrudFuncionarioService.class);
```

Loggers

- Podemos registrar os seguintes níveis de log com objeto Logger

```
5 usages
private static final Logger logger = LoggerFactory.getLogger(CadastroContatoService.class);

1 usage
public Contato salvar(Contato contato){

    logger.trace("Uma mensagem de TRACE");
    logger.debug("Uma mensagem de DEBUG ");
    logger.info("Uma mensagem de INFO ");
    logger.warn("Uma mensagem de AVISO");
    logger.error("Uma mensagem de ERRO");

    return contatoRepository.save(contato);
}
```


Loggers

- Também podemos declarar o loggers com anotação @Slf4j do Lombok, a diferença é que o Lombok cria o objeto com o nome de log.

```
@Slf4j
@Service
public class CadastroContatoService {
    3 usages
    @Autowired
    private ContatoRepository contatoRepository;

    // private static final Logger logger = LoggerFactory.getLogger(CadastroContatoService.class);
    1 usage
    public Contato salvar(Contato contato){

        log.trace("Uma mensagem de TRACE");
        log.debug("Uma mensagem de DEBUG ");
        log.info("Uma mensagem de INFO ");
        log.warn("Uma mensagem de AVISO");
        log.error("Uma mensagem de ERRO");

        return contatoRepository.save(contato);
    }
}
```

Appendors

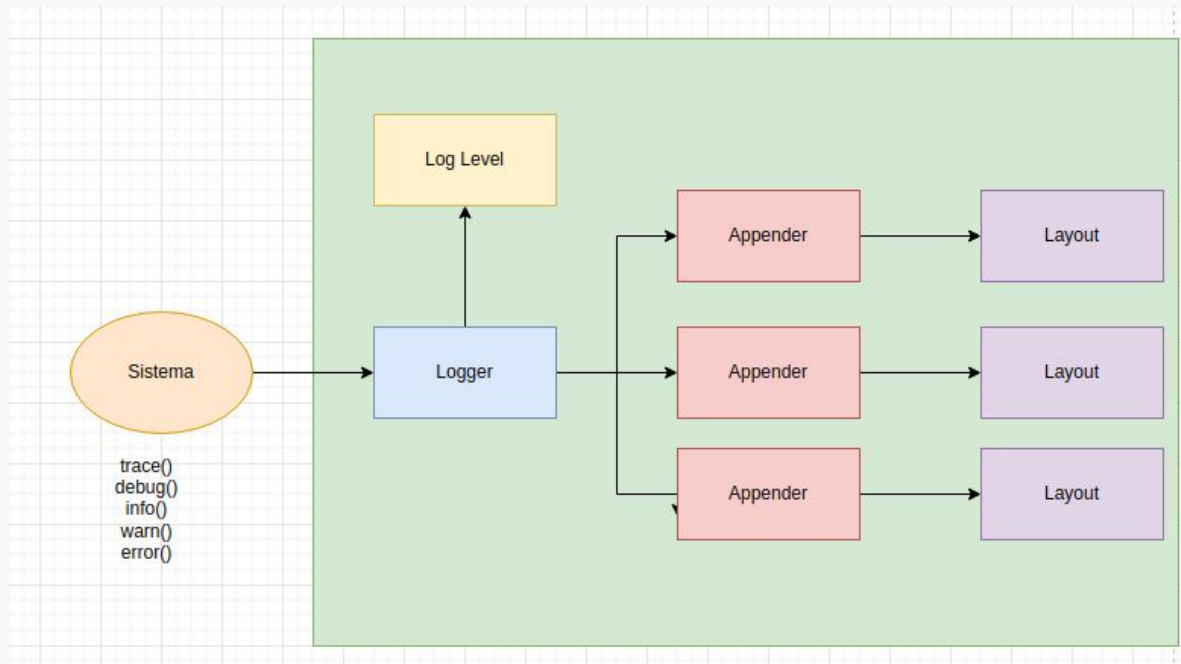
- Representam o destino das mensagens de log
- Um logger pode ter um ou mais appenders associados
- Alguns appenders importantes:

Appender	Descrição
ConsoleAppender	Direciona os logs para o console
FileAppender	Direciona os logs para um arquivo
RollingFileAppender	Direciona os logs para um arquivo e permite definir políticas sobre tamanho máximo do arquivo, número de arquivos de backup e etc.

- Formatam os dados dos logs
- Um appender deve ter um layout associado
- Alguns layouts importantes:

Layouts	Descrição
PatternLayout	Formata a saída com base em um padrão de conversão
HTMLayout	Formata a saída em tabela HTML
XMLLayout	Formata a saída em XML
JsonLayout	Formata a saída em JSON

Juntando as partes



- É um serviço de gerenciamento de logs na nuvem
- Uma solução que tem plano gratuito e podemos usar para testar logs de nossas aplicações
- Tradicionalmente colocamos os logs da aplicação em um arquivo na máquina onde instalamos a aplicação
- Quando estamos desenvolvendo uma aplicação para rodar em ambiente de cloud, o ideal é não deixar os logs na própria máquina
- Fazer o cadastro em <https://www.loggly.com/>

Configurando o appender do Loggly no Logback

- Adicionar a dependência do Loggly no pom.xml

```
<dependency>
```

```
  <groupId>org.logback-extensions</groupId>
```

```
  <artifactId>logback-ext-loggly</artifactId>
```

```
  <version>0.1.5</version>
```

```
</dependency>
```

- No diretório resources criar o arquivo logback-spring.xml
- Configurar o appender no arquivo logback-spring.xml
- No site do Loggly obter o token do usuário em:
 - Logs->Source Setup->Customer Token

Debug

O que é debug

- É o processo de encontrar e reduzir defeitos em um aplicativo de software ou mesmo em hardware. Erros de software incluem aqueles que impossibilitam o programa de ser executado e/ou que produzem um resultado inesperado.
- Quais os principais ganhos em utilizar essa técnica no desenvolvimento de software?
 - Aprender mais sobre o código
 - Entender regras de negócio da aplicação
 - Ajudar na refatoração de código
 - Testar algo antes de alterar o código
 - Verificar se existe algo errado no código, ou se foi a regra de negócio implementada erroneamente

Breakpoint

- O breakpoint serve para indicar a IDE onde a aplicação deve parar para que o desenvolvedor possa observar o que está acontecendo naquele determinado ponto
- Não há limites para a quantidade de breakpoints a serem colocados no código, é possível colocar o quanto julgar necessário

- Cada botão destes possui uma função para navegar entre as linhas do código de sua aplicação durante o processo de debug:
 - **Step Into (F7):** entra no método em que o debug está parado
 - **Step Over (F8):** vai para a próxima linha de código
 - **Step Return (Shift+F8):** retorna ao método anterior e só funciona quando o Step Into é acionado antes

Material Complementar

- [Help sobre debug no IntelliJ](#)
- [Java Logback](#)

INTERVALO DE AULA

DEV!

Finalizamos o nosso primeiro período de hoje. Que tal descansar um pouco?!

Nos vemos em 20 minutos.

Início: 20:20

Retorno: 20:40



AVALIAÇÃO DOCENTE

O que você está achando das minhas aulas neste conteúdo?

[Clique aqui](#) ou escaneie o QRCode ao lado para avaliar minha aula.

Sinta-se à vontade para fornecer uma avaliação sempre que achar necessário.





DEVinHouse

Parcerias para desenvolver a sua carreira

OBRIGADO!



<LAB365>