



# Trace distribuído com Spring Sleuth e Zipkin

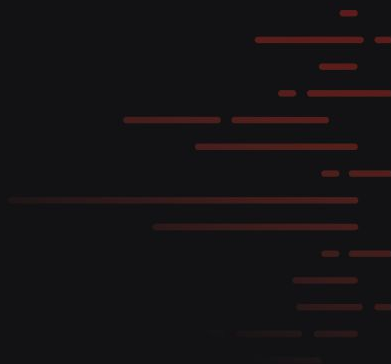


16/11/2021



# Agenda

- Sobre mim e a minha relação com o código;
- O que teremos ao término;
- Slueth.



# Sobre mim e a minha relação com o código



- **Nome: Gustavo Figueiredo**
- **Formação: Ciência da Computação e Mestrado na área**
- **Tempo trabalhando como dev: 9 anos**
- **Posição / empresa: Engenheiro de Software Sênior / Dock**
- **Redes sociais:**
  - Plataforma da Rocketseat: <https://app.rocketseat.com.br/me/gustavo-figueiredo-07013>
  - LinkedIn: <https://www.linkedin.com/in/gustavo-figueiredo-8602966b/>
  - Github: <https://github.com/gustavodsf>

# Sobre a aula e o que será entregue no final



- **Ao término teremos uma aplicação onde todas as requisições são monitoradas, facilitando o acompanhamento do gargalos da aplicação.**
- **O que teremos de entrega no final do conteúdo :**
  - <https://github.com/rocketseat-experts-club/spring-slueth-tracing-2021-11-16>
- **Ambiente necessário:**
  - Java (JDK  $\geq$  11)
  - IDE de sua preferência
  - Docker

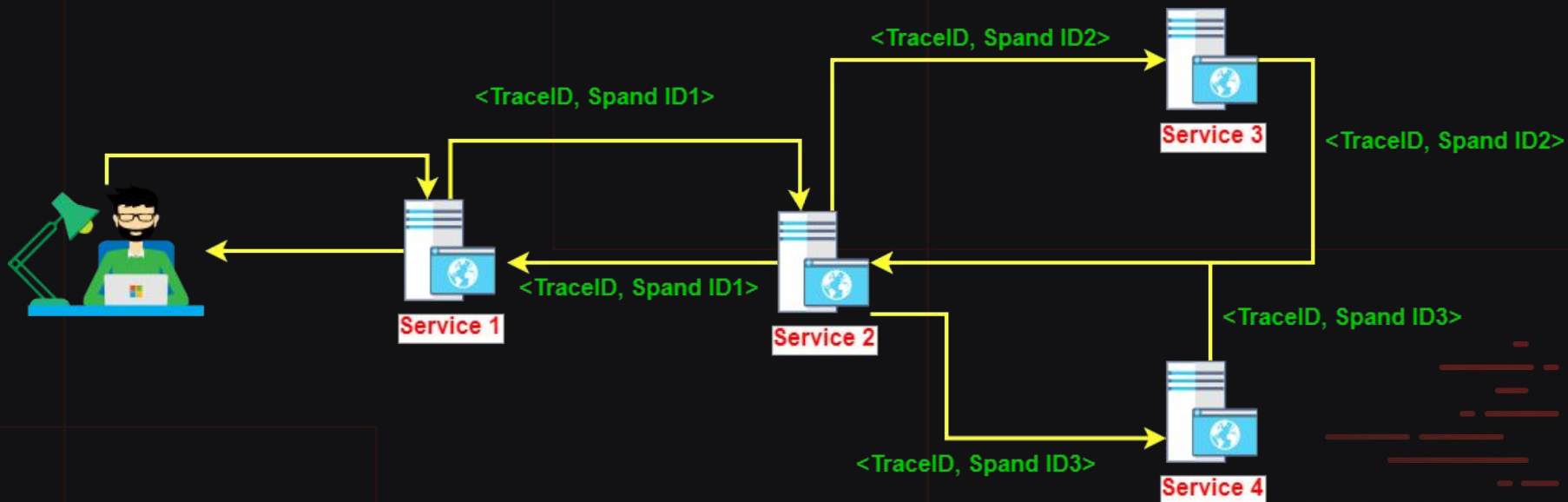


- Slueth implementa uma solução de rastreamento (Tracing);
- Sendo invisível ao usuário;
- Adiciona id de Trace e de span nos logs gerados de forma que possam ser verificados em um agregador de logs;
- Provê uma abstração acima dos objetos de rastreo: trace e span(formando um grafo acíclico e direcional);
- Instrumenta os componentes de entrada;
- Se o zipkin estiver disponível na aplicação o sleuth irá gerar e coletar dados compatíveis via HTTP para o zipkin.



- INFO [{nome da aplicação},{traceID},{spanID},{logPropagado}] 9579 --- [nio-7070-exec-1]  
b.c.l.pocsluethfeign.Application : Handling home
  - O nome da aplicação pode ser definido no application.properties spring.application.name;
  - spanID é gerado pelo próprio sleuth;
  - traceID é gerado pelo sleuth e propagado pelas aplicações no caso de chamadas externas;
  - logPropagado é um identificador para mostrar se o log deve ou não ser propagado as outras aplicações.

# Funcionamento





experts club

# Code Time!





- Trocar a forma de envio do zipkin para do tipo stream;
- Adicionar um novo span para verificar pontos importantes do sistema;
- Realizar o envio de log para um agregador.



- Colocamos o Zipkin para Rodar (Usando Docker);
- Criação da Primeira Aplicação;
- Criação da Segunda Aplicação;
- Criação da Terceira Aplicação;
- Criação da Quarta Aplicação;
- Customização da Primeira Aplicação;
- Visualização dos Resultados no Zipkin.





experts club

# Obrigado!

**Gustavo Figueiredo**

gustavodsf1@gmail.com

Redes sociais: <https://github.com/gustavodsf>

Linkedin: <https://github.com/gustavodsf>

<https://app.rocketseat.com.br/me/gustavo-figueiredo-07013>

