Logging

Logging in Spring Boot & Spring Cloud

Spring Boot logging

- Начиная со Spring Boot 2.X.X Logback используется по дефолту а все стартеры наследуются от **spring-boot-starter-logging** и уже имеют предустановленные конфиги логирования с шаблонами форматирования, цветами вывода.

Spring Boot logging

- Логирование имеет уровни (TRACE, DEBUG, INFO, WARN, ERROR) и с помощью конфига **application.yaml** можно легко задать уровень логирования. Например можно легко логировать все request приходящие на веб контроллеры приложения всего одной строчкой в конфиге `logging.level.org.springframework.web.servlet.DispatcherServlet: DEBUG`

Spring Boot logging

- Дефолтные настройки можно переопределить (форматирование, каналы вывода, цвета вывода, ротация файла логирования) поместив в src/main/resources **logback-spring.xml**

Spring Boot logging

- Можно вовсе заменить дефолтный Logstash на что то более модное, содержащее слово async, например на **Log4j2**.

Spring Boot logging

- Когда недостаточно output(ов) или нужно централизованно собирать логи со всех сервисов,то можно легко натравить логирование на **ELK**(ElasticSearch, Logstash, Kibana).

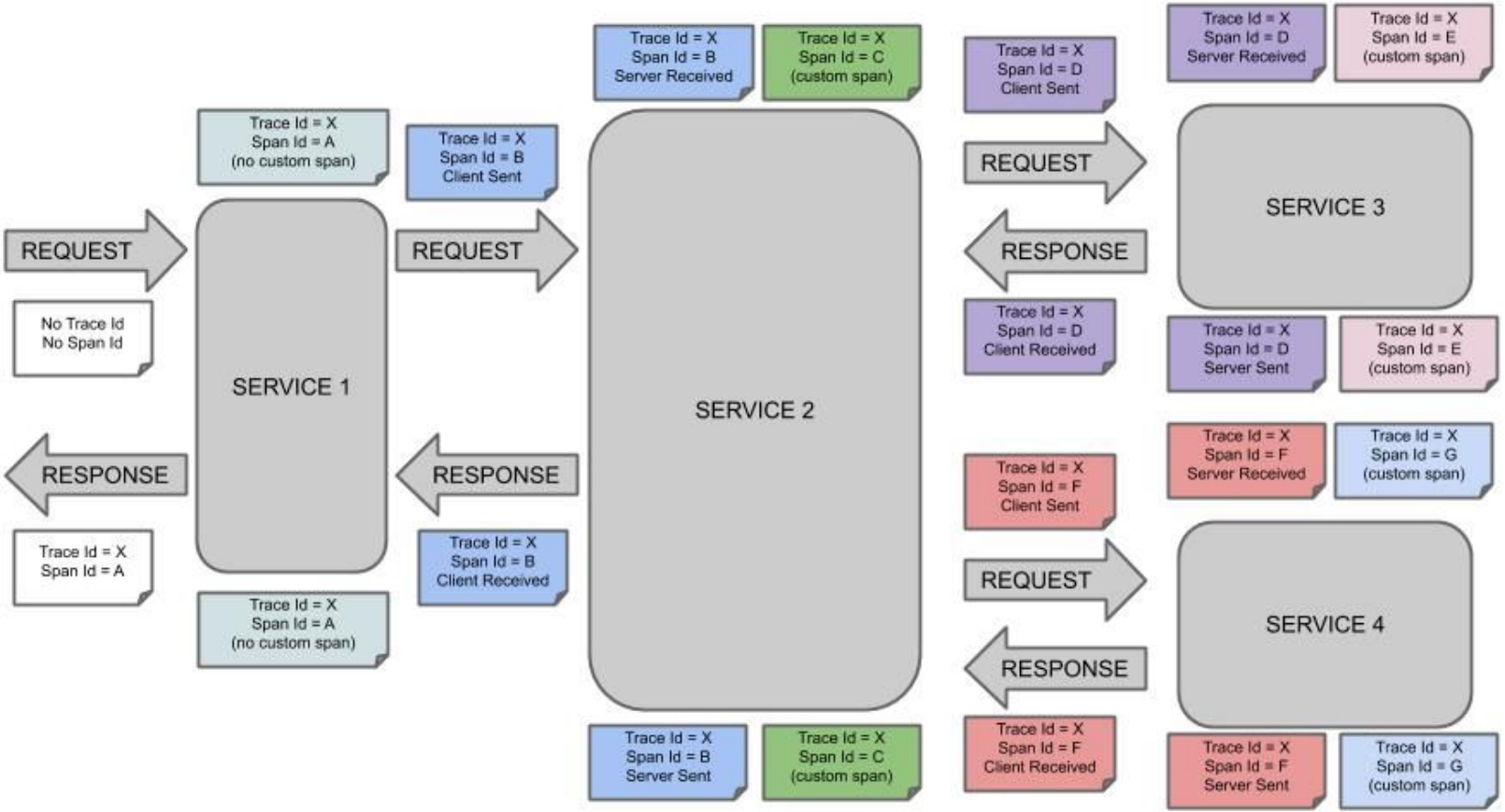
Spring Cloud logging

В случае когда сервисы начинают разрастаться (в своем разнообразии, по числу инстансов) то транзакция скорее всего становится распределенной и отследить всю цепочку вызовов в при необходимости становится крайне сложно. Тут логично бы пробрасывать некий ID уникальный в скопе одной транзакции.

Spring Cloud logging

Чем же хороша библиотека **Spring Cloud Sleuth**?

- Не нужны самописные фильтры/интерсепторы
- Есть интеграция с **Open Zipkin**
- Простота детализации измерения работы


REQUEST
No Trace Id
No Span Id
SERVICE 1
RESPONSE
Trace Id = X
Span Id = A
Trace Id = X
Span Id = A
(no custom span)
Trace Id = X
Span Id = B
Client Sent
REQUEST
SERVICE 2
RESPONSE
Trace Id = X
Span Id = B
Client Received
Trace Id = X
Span Id = B
Server Received
Trace Id = X
Span Id = C
(custom span)
Trace Id = X
Span Id = B
Server Sent
Trace Id = X
Span Id = C
(custom span)
Trace Id = X
Span Id = D
Client Sent
REQUEST
SERVICE 3
RESPONSE
Trace Id = X
Span Id = D
Client Received
Trace Id = X
Span Id = D
Server Received
Trace Id = X
Span Id = E
(custom span)
Trace Id = X
Span Id = D
Server Sent
Trace Id = X
Span Id = E
(custom span)
Trace Id = X
Span Id = F
Client Sent
REQUEST
SERVICE 4
RESPONSE
Trace Id = X
Span Id = F
Client Received
Trace Id = X
Span Id = F
Server Received
Trace Id = X
Span Id = G
(custom span)
Trace Id = X
Span Id = F
Server Sent
Trace Id = X
Span Id = G
(custom span)

Spring Cloud logging

- Работа **Spring Cloud Sleuth** в связке с **Open Zipkin** дают нам из коробки мощный инструмент агрегирования логов
- Из приятного,**Zipkin** так же предоставляет графический интерфейс