# **Distributed tracing using Sleuth**

Slueth will assign a unique request id for each and every request, so that even if same request travels through all the micro services the trace id will print in each logger , so if u want to see what happened to that request in each microservice , u can see search with that traceid in each microservice and then u will get all the logs of that request in svc-1 , and with the same trace id if u search in microservice -2 then u can see that request related logs which was originated in ms-1

I think it will assign that to some thread, that’s why event if the request goes to some other appn also that request id will be printed

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-sleuth</artifactId>

<version>3.1.5</version>

</dependency>

# How this sleuth works

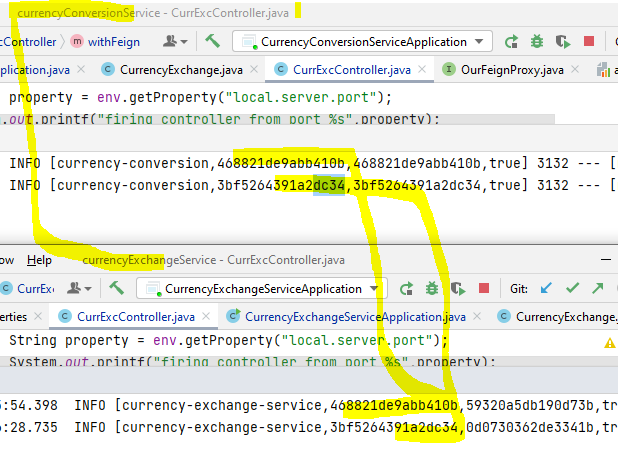
This works just by creating and adding the bean to the context and it will speak with log 4j and prints the trace id while logging each line

Create a bean and all this bean to all the appn where u want to print the traceid to be printed in logs

@Bean  
public Sampler defaSampler(){  
 return Sampler.*ALWAYS\_SAMPLE*;  
}

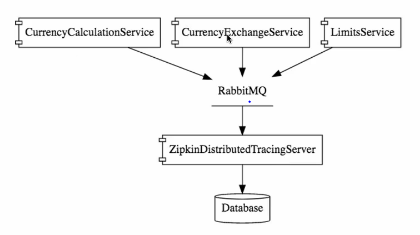
I think sleuth will interact will lof4j and prints the trace id/ span id across each and every log

Sleuth is only responsible to add the trace id to the sleuth , Zipkin is just like a splunk



In above picture u can see same request travelled from conversion service to cur exchange service and each request have same trace id

# Zipkin



The approach is to push all the logs to rabbit mq , so that Zipkin can read and consolidate them using the trace id.

Here is the URL you can use to download Zipkin jar

<https://search.maven.org/remote_content?g=io.zipkin&a=zipkin-server&v=LATEST&c=exec>

If you get a 404 while downloading the jar, use the curl command to download :

1. curl -sSL https://zipkin.io/quickstart.sh | bash -s
2. java -jar zipkin.jar

For more information, please go through the below link:

<https://zipkin.io/pages/quickstart>

#### **ONLY FOR WINDOWS USERS**

If you are on Windows, this is important for you:

After you watch the next video, You can use the below commands to run Zipkin Server.

1. set RABBIT\_URI=amqp://localhost
2. java -jar zipkin-server-2.7.0-exec.jar

## Debugging Zipkin

[https://github.com/in28minutes/in28minutes-initiatives/tree/master/The-in28Minutes-TroubleshootingGuide-And-FAQ#debugging-problems-with-zipkin](https://github.com/in28minutes/in28minutes-initiatives/tree/master/The-in28Minutes-TroubleshootingGuide-And-FAQ" \l "debugging-problems-with-zipkin" \t "_blank)