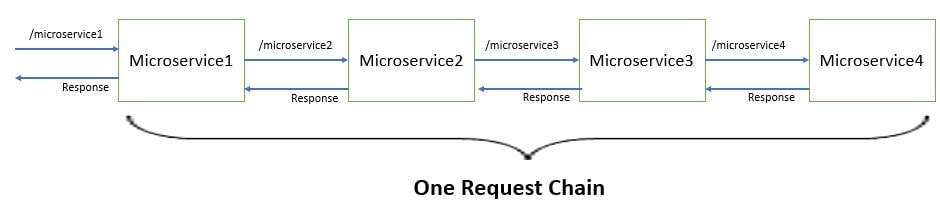
Lets Begin-

We will be dividing this tutorial into 3 parts-

* 1. Develop four Spring Boot Microservices modules which interact with each other.
* 2. Implement distributed tracing using Spring Cloud Sleuth
* 3. View distributed tracing using Zipkin

Develop four Spring Boot Microservices modules which interact with each other

We will be developing the spring boot microservices as follows-

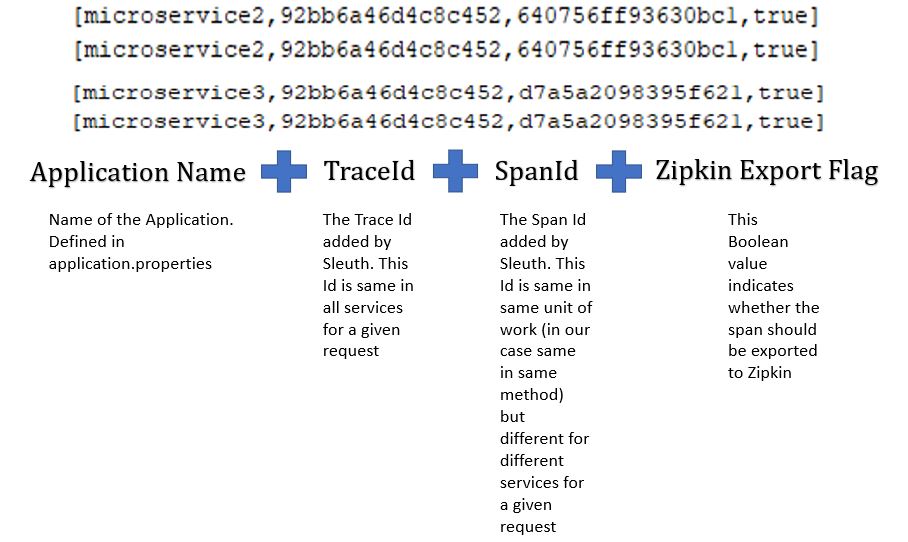


Application.properties

server.port=8080

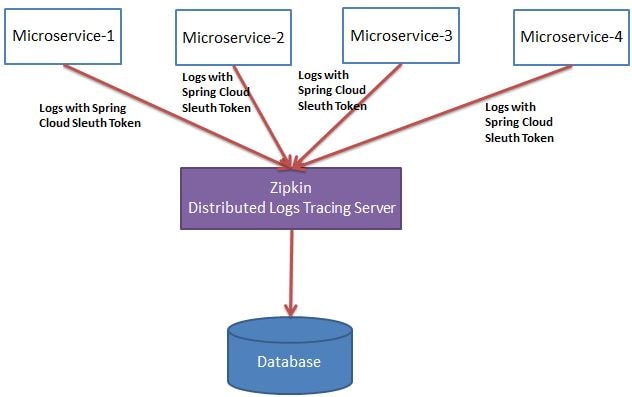
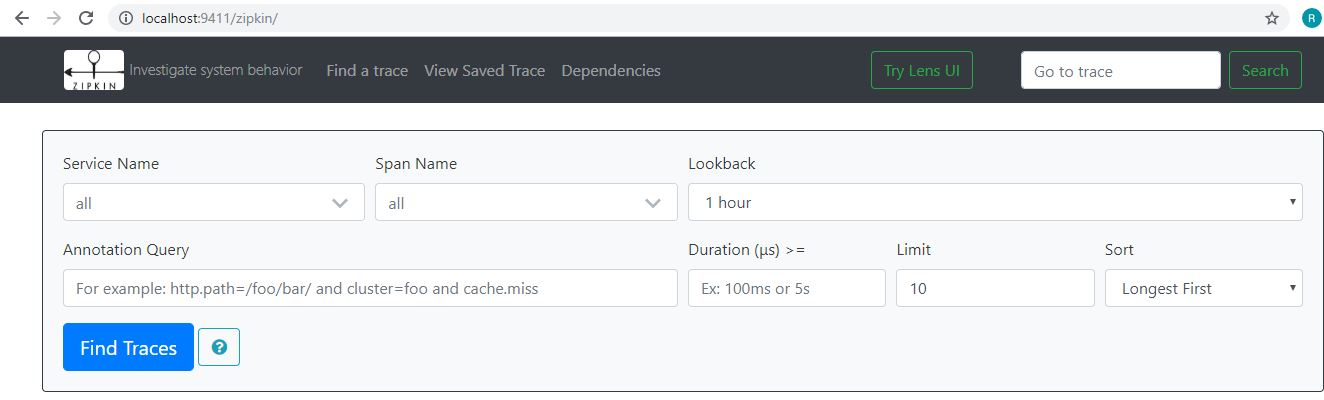
spring.application.name=microservice1

**spring.zipkin.enabled=false**



* Microservice-1 logs -  
  Spring Cloud Sleuth Module1 output
* Microservice-2 logs -  
  Spring Cloud Sleuth Module2 output
* Microservice-3 logs -  
  Spring Cloud Sleuth Module3 output
* Microservice-4 logs -  
  Spring Cloud Sleuth Module4 output

Use Zipkin for distributed log analysis-

* Zipkin is a distributed tracing system. It helps gather timing data needed to troubleshoot latency problems in service architectures. Features include both the collection and lookup of this data.  
    
  From [maven repository get the latest zipkin jar.](https://search.maven.org/remote_content?g=io.zipkin.java&a=zipkin-server&v=LATEST&c=exec) Once you have downloaded this jar using the command prompt run the execute jar command as follows-  
    
  If we now go to the url - **localhost:9411** we see the zipkin dashboard as follows-  
  
* In all the pom.xml add the spring cloud zipkin dependency as follows-
* <dependency>
* <groupId>org.springframework.cloud</groupId>
* <artifactId>spring-cloud-starter-zipkin</artifactId>
* </dependency>

Also from the application.properties remove the property we had defined

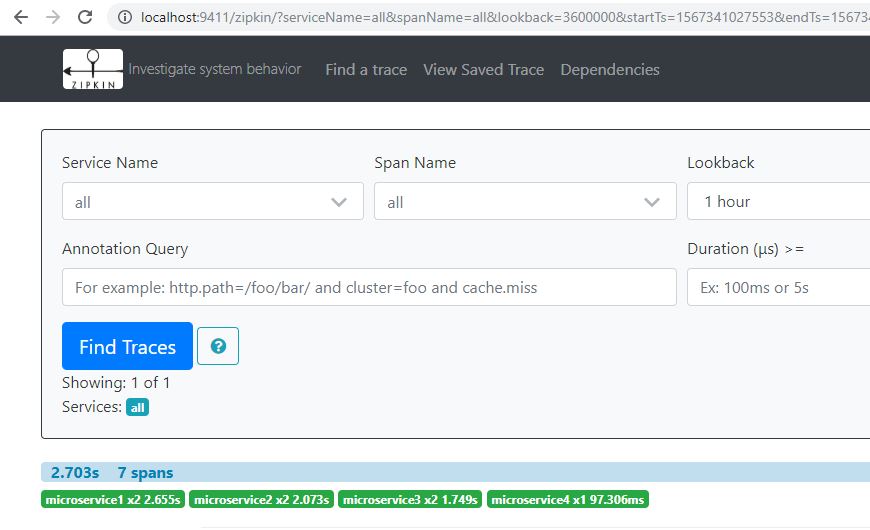
earlier - **spring.zipkin.enabled=false**

Start the microservices.

And again hit the url localhost:8080/microservice1

Java –jar xxxx.jar

Open port 9411

* Now go the zipkin dashboard and click on search logs-  
  
* Here we can see the log stacktrace for the particular request chain. We can further analyze this request chain by selecting it in the dashboard.  
  