

# URLs

## Ports

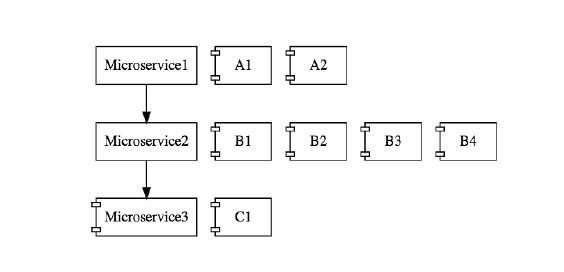
| **Application** | **Port** |
| --- | --- |
| Limits Service | 8080, 8081, ... |
| Spring Cloud Config Server | 8888 |
|  |  |
| Currency Exchange Service | 8000, 8001, 8002, .. |
| Currency Conversion Service | 8100, 8101, 8102, ... |
| Netflix Eureka Naming Server | 8761 |
| Netflix Zuul API Gateway Server | 8765 |
| Zipkin Distributed Tracing Server | 9411 |

## URLs

| **Application** | **URL** |
| --- | --- |
| Limits Service | <http://localhost:8080/limits> POST -> <http://localhost:8080/actuator/refresh> |
| Spring Cloud Config Server | <http://localhost:8888/limits-service/default> <http://localhost:8888/limits-service/dev> |
| Currency Converter Service - Direct Call | <http://localhost:8100/currency-converter/from/USD/to/INR/quantity/10> |
| Currency Converter Service - Feign | <http://localhost:8100/currency-converter-feign/from/EUR/to/INR/quantity/10000> |
| Currency Exchange Service | <http://localhost:8000/currency-exchange/from/EUR/to/INR>  <http://localhost:8001/currency-exchange/from/USD/to/INR> |
| Eureka | <http://localhost:8761/> |
| Zuul - Currency Exchange & Exchange Services | <http://localhost:8765/currency-exchange-service/currency-exchange/from/EUR/to/INR>  <http://localhost:8765/currency-conversion-service/currency-converter-feign/from/USD/to/INR/quantity/10> |
| Zipkin | <http://localhost:9411/zipkin/> |
| Spring Cloud Bus Refresh | <http://localhost:8080/bus/refresh> |

# Challenges with Micro Service <https://www.udemy.com/course/microservices-with-spring-boot-and-spring-cloud/learn/lecture/8005708#overview>

## Configuration Management



On a 3 MicroService, on peut avoir 50, chaque microservices à plusieurs instances dans chaque environnement

## Dynamic Scale Up and Scale Down

Des moments j’aurai besoin de 2 instances d’un microService

Et d’autres moment j’aurai besoin de plus ou de moins d’instance, c’est cela la Dynamic Scale Up and Scale Down.

## Visibility :

Limit services

Currency service

Exchange service

# Eureka

# Ribbon

# Bus

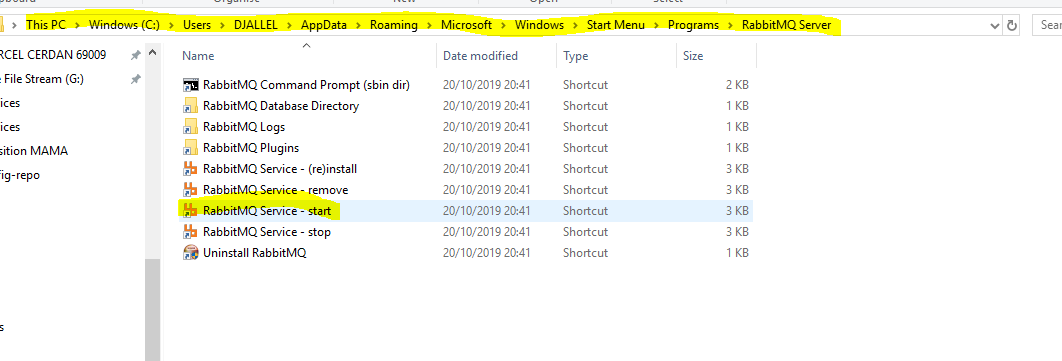
# RabbitMQ

## Run RabbitMQ

|  |
| --- |
| The RabbitMQ service starts automatically. You can stop/reinstall/start the RabbitMQ service from the Start Menu. |

|  |
| --- |
| To launch RabbitMQ il faut faire  /usr/local/sbin/rabbitmq-server |

Tu peux aussi runner rabbitMQ depuis ici ( voiri image ci-dessous )



# Zuul

# Spring cloud sleuth

Spring cloud sleuth ajoutera un identifiant unique à une demande afin que vous puissiez la placer dans plusieurs composants.

Il se fait par 2 étapes simples :

## Ajouter une dépendance dans le pom <spring cloud starer sleuth>

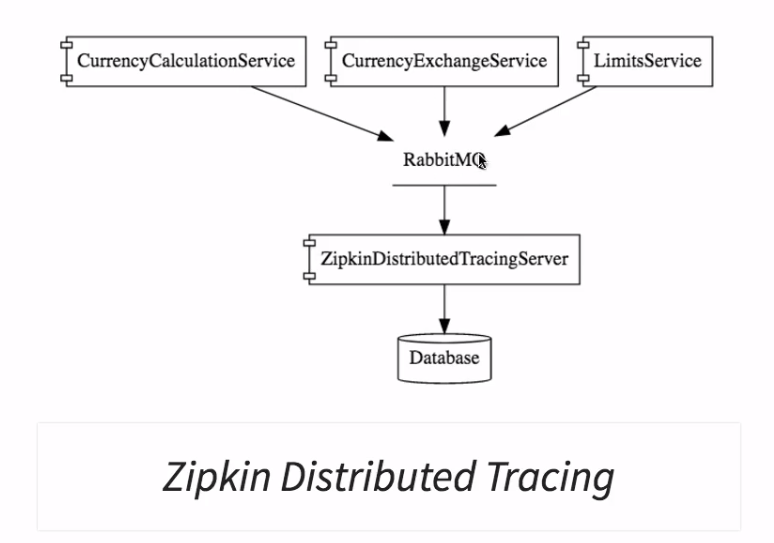
<dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-starter-sleuth</artifactId>  
</dependency>

## Ajouter un bean

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

# Zipkin distributed Tracing

<dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-sleuth-zipkin</artifactId>  
</dependency>



Updates to Step 39 - Running Zipkin on Windows

In the next step, we set up our Zipkin Server by downloading a jar.

Move to the next step if you are not on Windows!

**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

## Pour runner Zipking

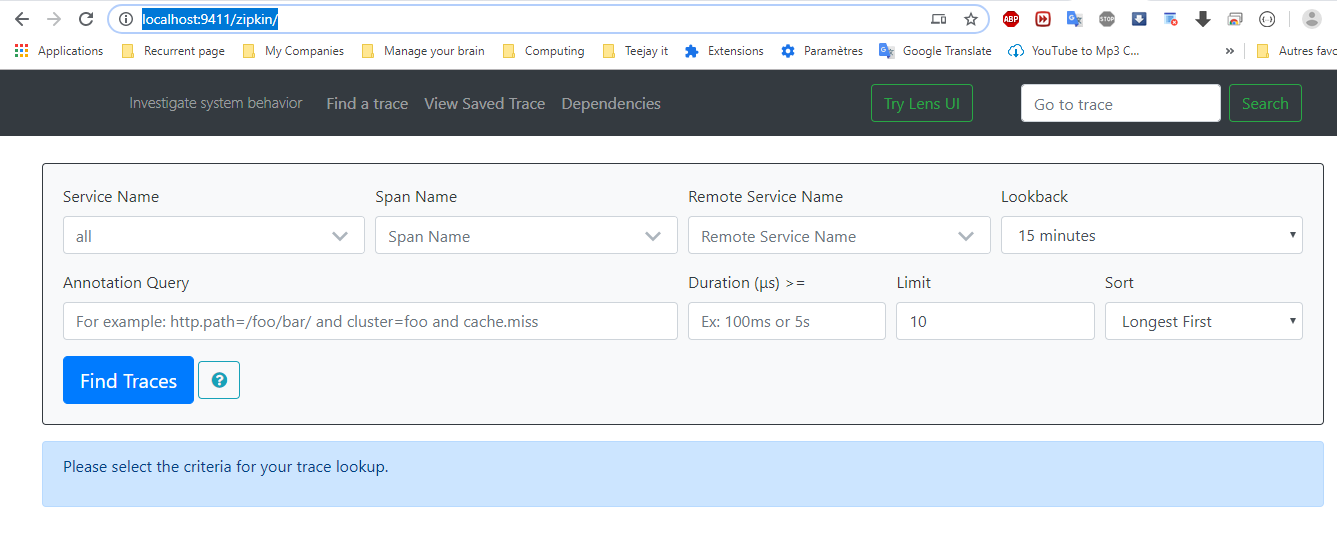
cd J:\5-FORMATION\TABET Djalal\MicroService with Spring Cloud\MicroService with Spring Cloud\03.microservices

java -jar zipkin.jar

set RABBIT\_URI=amqp://localhost

java -jar zipkin.jar

puis aller à <http://localhost:9411/zipkin/>



Tous les MicrosService vont mettre leurs log message sur la file d’attente RabbitMQ et je serai en mesure de suivre sur Zipkin Distributed Tracing

Zipkin ecoute RabbitMQ.

## Pour runner Zipking avec RabbitMQ

|  |
| --- |
| cd J:\5-FORMATION\TABET Djalal\MicroService with Spring Cloud\MicroService with Spring Cloud\03.microservices  java -jar zipkin.jar  set RABBIT\_URI=amqp://localhost  java -jar zipkin.jar |

Dans cette etape, nous avons installer rabbitMQ, lancer zipkin

Puis dis à zipking de écouter rabbit mq