<https://www.interviewgrid.com/interview_questions/spring_cloud/spring_cloud_netflix>

## **Spring Cloud Netflix – Eureka**

* We have a service registry (Eureka Server)
* We have a REST service, which registers itself at the registry (Eureka Client)

To Implement a ***Eureka Server*** for service registry,

* We have to add [spring-cloud-starter-netflix-eureka-server](https://search.maven.org/search?q=spring-cloud-starter-netflix-eureka-server) dependencies to pom.xml
* To enable the Eureka Server in a [@SpringBootApplication](https://www.baeldung.com/spring-boot-application-configuration) by annotating it with @EnableEurekaServer
* We have to configure some properties: **Here we're configuring an application port; the default one for***Eureka***servers is***8761***.** We're telling the built-in *Eureka Client* not to register with itself because our application should be acting as a server.

server:

port: 8761

eureka:

client:

registerWithEureka: false

fetchRegistry: false

* Now we'll point our browser to [http://localhost:8761](http://localhost:8761/) to view the *Eureka* dashboard, where we'll later inspect the registered instances.
* At the moment, we can see basic indicators, such as status and health indicators:
* Instances currently registered with eureka

For ***Eureka Client***,

* We have to add [spring-cloud-starter-netflix-eureka-client](https://search.maven.org/search?q=spring-cloud-starter-netflix-eureka-server) dependencies to pom.xml
* To enable the Eureka Client in a [@SpringBootApplication](https://www.baeldung.com/spring-boot-application-configuration) by annotating it with @EnableEurekaClient
* We have to configure some properties,

spring:

application:

name: spring-cloud-eureka-client

server:

port: 0

eureka:

client:

serviceUrl:

defaultZone: ${EUREKA\_URI:http://localhost:8761/eureka}

instance:

preferIpAddress: true

* Now we'll run the client, and point our browser to [*http://localhost:8761*](https://localhost:8761/) again to see its registration status on the Eureka Dashboard