

Elaboro	Documento	Versión	Descripción	Fecha
Luis Sánchez Martínez	Práctica Eureka.	1.0	Creación de documento	14/04/2024

Práctica Eureka

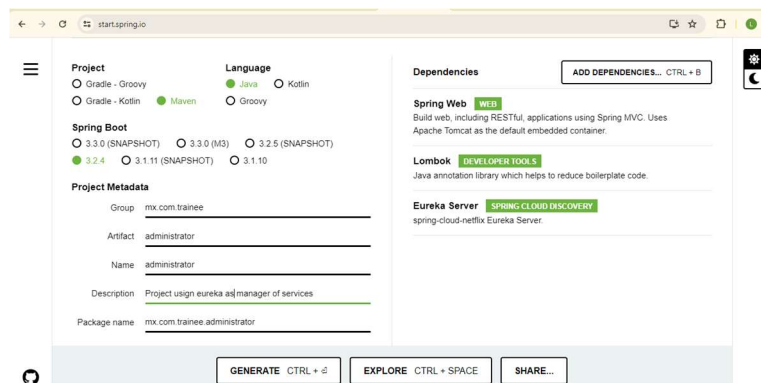
Tiempo estimado: 2 horas.

Uso de:

- JDK 17
- Maven
- IntelliJ
- Postman
- Git
- Spring Cloud
 - Resilience4J
 - Actuator
 - FeignClient
 - Eureka

Crear Servidor Eureka

1. Crear un proyecto SpringBoot para utilizarlo como servidor Eureka incluyendo la dependencia spring-cloud-starter-netflix-eureka-server.



```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>
</dependency>
```

```

17 <java.version>17</java.version>
18 <spring-cloud.version>2023.0.1</spring-cloud.version>
19 </properties>
20 <dependencies>
21 <dependency>
22 <groupId>org.springframework.boot</groupId>
23 <artifactId>spring-boot-starter-web</artifactId>
24 </dependency>
25 <dependency>
26 <groupId>org.springframework.cloud</groupId>
27 <artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>
28 </dependency>
29 <dependency>
30 <groupId>org.springframework.boot</groupId>
31 <artifactId>spring-boot-starter-actuator</artifactId>
32 </dependency>
33
34 <dependency>
35 <groupId>org.projectlombok</groupId>
36 <artifactId>lombok</artifactId>
37 <optional>true</optional>
38 </dependency>

```

2. Configurar el archivo de propiedades, en este caso el yaml, las propiedades para indicar un puerto y evitar que el servidor Eureka intente registrarse el mismo.

```

#Application Name
spring:
  application:
    name: administrator

server:
  port: 8089

eureka:
  client:
    #define that the server not to register himself in the service registry
    registerWithEureka: false
    fetchRegistry: false
    serviceUrl:
      defaultZone: http://localhost:8089/eureka
  server:
    #wait time for subsequent sync
    waitTimeInMsWhenSyncEmpty: 0

```

```

1 #Application Name
2 spring:
3   application:
4     name: administrator
5
6 server:
7   port: 8089
8
9 eureka:
10  client:
11    #define that the server not to register himself in the service registry
12    registerWithEureka: false
13    fetchRegistry: false
14    serviceUrl:
15      defaultZone: http://localhost:8089/eureka
16  server:
17    #wait time for subsequent sync
18    waitTimeInMsWhenSyncEmpty: 0

```

3. Agregar la notación `@EnableEurekaServer` en la clase `Application` para indicar que se crea el servicio Eureka.

```
@EnableEurekaServer
@SpringBootApplication
public class AdministratorApplication {

    public static void main(String[] args) {
        SpringApplication.run(AdministratorApplication.class, args);
    }

}
```

```
2
3 > import ...
7
8 @EnableEurekaServer
9 @SpringBootApplication
10 public class AdministratorApplication {
11
12 >     public static void main(String[] args) { SpringApplication.run(AdministratorApplication.class, args);
15
16     }
17
```

4. Inicializar el servicio para realizar una prueba ejecutando una consulta a la URL: <http://localhost:8089>.

```
Run AdministratorApplication x
INFO 1576 --- [administrator] [ Thread-9] c.n.e.r.PeerAwareInstanceRegistryImpl : Got 1 instances from neighboring DS node
INFO 1576 --- [administrator] [ Thread-9] c.n.e.r.PeerAwareInstanceRegistryImpl : Renew threshold is: 1
INFO 1576 --- [administrator] [ Thread-9] c.n.e.r.PeerAwareInstanceRegistryImpl : Changing status to UP
INFO 1576 --- [administrator] [ Thread-9] e.s.EurekaServerInitializerConfiguration : Started Eureka Server
INFO 1576 --- [administrator] [ main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8089 (http with context path '')
INFO 1576 --- [administrator] [ main] .s.c.n.e.s.EurekaAutoServiceRegistration : Updating port to 8089
INFO 1576 --- [administrator] [ main] m.c.t.a.AdministratorApplication : Started AdministratorApplication in 19.873 seconds (process running ...)
```

The screenshot shows the Spring Eureka web interface at localhost:8089. The page has a dark header with the Spring Eureka logo and navigation links for HOME and LAST 1000 SINCE STARTUP. The main content area is divided into several sections:

- System Status:** A table showing environment details.

Environment	test	Current time	2024-04-14T18:46:35 -0600
Data center	default	Uptime	00:05
		Lease expiration enabled	false
		Renews threshold	1
		Renews (last min)	0
- DS Replicas:** A section showing the local data center as 'localhost'.
- Instances currently registered with Eureka:** A table with columns for Application, AMIs, Availability Zones, and Status. It currently shows 'No instances available'.
- General Info:** A section at the bottom of the page.

Registrar el microservicio permiso-service

5. Agregar la dependencia de Eureka spring-cloud-starter-netflix-eureka-client.

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
</dependency>
```

```
78  <dependency>
79    <groupId>org.springframework.cloud</groupId>
80    <artifactId>spring-cloud-starter-openfeign</artifactId>
81  </dependency>
82
83  <dependency>
84    <groupId>org.springframework.cloud</groupId>
85    <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
86  </dependency>
87
88  <dependency>
89    <groupId>com.fasterxml.jackson.core</groupId>
90    <artifactId>jackson-databind</artifactId>
91  </dependency>
92
93  <dependency>
```

6. Configurar el archivo de propiedades, agregando las propiedades para que el servicio se registre en Eureka y funcione correctamente.

```
spring.application.name=permiso-service
server.port=8090
eureka.client.register-with-eureka= true
eureka.client.service-url.defaultZone=http://localhost:8089/eureka
```

```
1  spring.application.name=permiso-service
2  server.port=8090
3
4  springdoc.api-docs.path=/permiso/api-docs
5  springdoc.packages-to-scan=mx.com.tcs.permiso.controller
6  springdoc.swagger-ui.tryItOutEnabled=false
7  springdoc.swagger-ui.operationsSorter=method
8  springdoc.swagger-ui.tagsSorter=alpha
9  springdoc.swagger-ui.filter=false
10
36 spring.cloud.openfeign.circuitbreaker.enabled=true
37 spring.cloud.openfeign.circuitbreaker.alphanumeric-ids.enabled=true
38 resilience4j.circuitbreaker.instances.UserTypeFeignClient.getId.minimumNumberOfCalls=50
39 resilience4j.timelimiter.instances.UserTypeFeignClient.getId.timeoutDuration=2s
40
41 eureka.client.register-with-eureka= true
42 eureka.client.service-url.defaultZone=http://localhost:8089/eureka
```

7. Inicializar el servicio permiso-service para realizar una prueba ejecutando una consulta a la URL:
<http://localhost:8089> y validar que el servicio se registre.
 - a. Log del microservicio permiso-service.

```

Run PermisoServiceApplication
o.s.web.client.RestTemplate : HTTP GET http://localhost:8089/eureka/apps/
o.s.web.client.RestTemplate : Accept=[application/json, application/*+json]
o.s.web.client.RestTemplate : Response 200 OK
o.s.web.client.RestTemplate : Reading to [org.springframework.cloud.netflix.eureka.http.EurekaApplications]
o.s.web.client.RestTemplate : HTTP POST http://localhost:8089/eureka/apps/PERMISO-SERVICE
o.s.web.client.RestTemplate : Accept=[application/json, application/*+json]
o.s.web.client.RestTemplate : Writing [InstanceInfo [instanceId = host.docker.internal:permiso-service:8090, appName = PERMISO-SERVICE, host
o.s.web.client.RestTemplate : Response 204 NO_CONTENT
o.s.web.client.RestTemplate : HTTP GET http://localhost:8089/eureka/apps/
o.s.web.client.RestTemplate : Accept=[application/json, application/*+json]
o.s.web.client.RestTemplate : HTTP PUT http://localhost:8089/eureka/apps/PERMISO-SERVICE/host.docker.internal:permiso-service:8090/status=UP
o.s.web.client.RestTemplate : Response 200 OK
o.s.web.client.RestTemplate : Reading to [org.springframework.cloud.netflix.eureka.http.EurekaApplications]
o.s.web.client.RestTemplate : Accept=[application/json, application/*+json]
o.s.web.client.RestTemplate : Response 200 OK

```

b. Log del microservicio administrator.

```

Run AdministratorApplication
c.n.e.registry.AbstractInstanceRegistry : Running the evict task with compensationTime 14ms
c.n.e.registry.AbstractInstanceRegistry : Running the evict task with compensationTime 0ms
c.n.e.registry.AbstractInstanceRegistry : Running the evict task with compensationTime 0ms
c.n.e.r.PeerAwareInstanceRegistryImpl : Current renewal threshold is : 1
c.n.e.registry.AbstractInstanceRegistry : Running the evict task with compensationTime 0ms
c.n.e.registry.AbstractInstanceRegistry : Running the evict task with compensationTime 11ms
c.n.e.registry.AbstractInstanceRegistry : Registered instance PERMISO-SERVICE/host.docker.internal:permiso-service:8090 with status UP (replication=false)
c.n.e.registry.AbstractInstanceRegistry : Registered instance PERMISO-SERVICE/host.docker.internal:permiso-service:8090 with status UP (replication=true)
c.n.e.registry.AbstractInstanceRegistry : Running the evict task with compensationTime 13ms
c.n.e.registry.AbstractInstanceRegistry : Running the evict task with compensationTime 2ms

```

c. Consulta de los servicios registrados en eureka.

System Status			
Environment	test	Current time	2024-04-14T19:31:48 -0600
Data center	default	Uptime	00:50
		Lease expiration enabled	true
		Renews threshold	3
		Renews (last min)	4
DS Replicas			
localhost			
Instances currently registered with Eureka			
Application	AMIs	Availability Zones	Status
PERMISO-SERVICE	n/a (1)	(1)	UP (1) - host.docker.internal:permiso-service:8090
General Info			
Name	Value		
total-avail-memory	83mb		
num-of-cpus	8		

Registrar el microservicio tipousuario-service

8. Agregar la dependencia de Eureka spring-cloud-starter-netflix-eureka-client.

```

<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
</dependency>

```

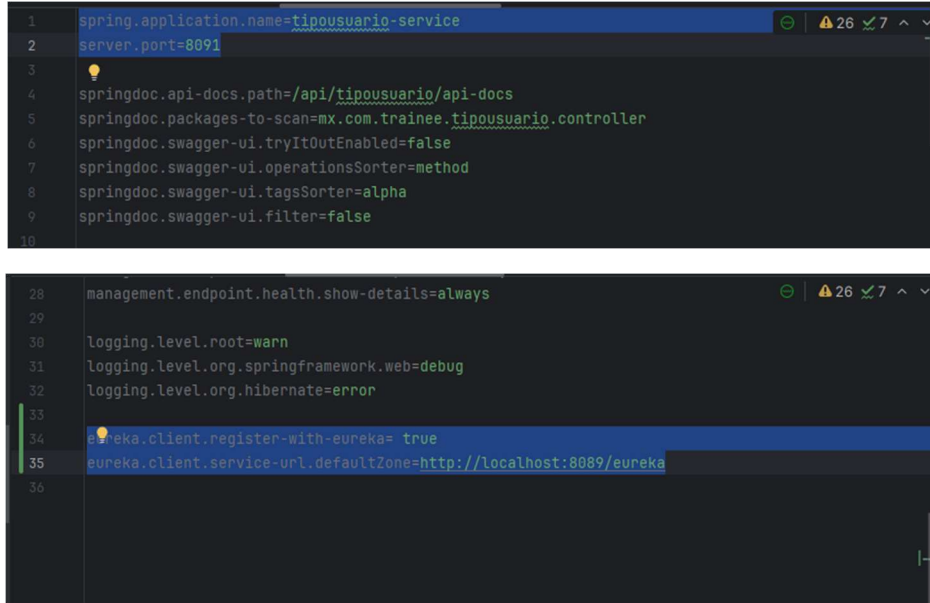
```

35
36
37
38
39
40
41
42
43
44
45
46
47
project dependencies dependency
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-circuitbreaker-resilience4j</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
</dependency>
<dependency>
  <groupId>org.springdoc</groupId>
  <artifactId>springdoc-openapi-starter-webmvc-ui</artifactId>
  <version>2.2.0</version>
</dependency>

```

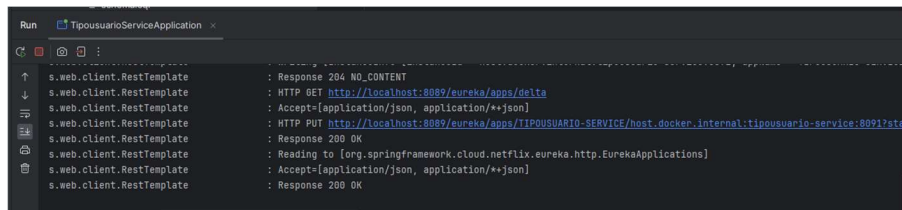
9. Configurar el archivo de propiedades, agregando las propiedades para que el servicio se registre en Eureka y funcione correctamente.

```
spring.application.name=tipousuario-service
server.port=8091
eureka.client.register-with-eureka= true
eureka.client.service-url.defaultZone=http://localhost:8089/eureka
```

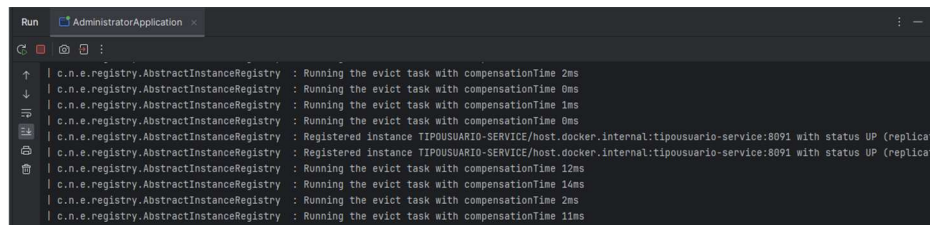


10. Inicializar el servicio tipousuario-service para realizar una prueba ejecutando una consulta a la URL: <http://localhost:8089> y validar que el servicio se registre.

- a. Log del microservicio tipousuario-service.



- b. Log del microservicio administrator.



- c. Consulta de los servicios registrados en eureka.

The screenshot shows the Eureka Server UI at localhost:8089. It displays system status, DS Replicas, and instances currently registered with Eureka.

System Status	
Environment	test
Data center	default
Current time	2024-04-14T19:51:14 -0600
Uptime	01:09
Lease expiration enabled	true
Renews threshold	5
Renews (last min)	8

DS Replicas

localhost

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
PERMISO-SERVICE	n/a (1)	(1)	UP (1) - host.docker.internal:permiso-service:8090
TIPOUSUARIO-SERVICE	n/a (1)	(1)	UP (1) - host.docker.internal:tipousuario-service:8091

General Info

11. Validar que quitando el host de la definición del FeignClient en el servicio permiso-service se puede aceptar la solicitud.

```
@FeignClient(name = "tipousuario-service", path = "/api/tipousuario", url =
"localhost:8091",
configuration = FeignConfiguration.class, fallback = UserTypeFallback.class)
```

```
@FeignClient(name = "tipousuario-service", path = "/api/tipousuario",
configuration = FeignConfiguration.class, fallback = UserTypeFallback.class)
```

```
12 * @author luis
13 * @since 1.0
14 * Interface to create feign client to request tipousuario-service and get the description of the UserType objec.
15 */
16 @FeignClient(name = "tipousuario-service", path = "/api/tipousuario",
17 configuration = FeignConfiguration.class, fallback = UserTypeFallback.class)
18 public interface UserTypeFeignClient {
19
20 /**
21 * Method used to get UserType record find by Id.
```

Referencias

<https://www.arquitecturajava.com/spring-cloud-eureka-server-y-discovery/>