



Eureka Server

Your Name :- Lokesh Khadse

Your Company :- Eidiko Systems Integrators

Employee ID - 1177

What is Eureka Server?

1. Eureka Server is part of the Spring Cloud Netflix suite.
2. Eureka server is also one of microservices.
3. It is used for service registration.
4. It allows different microservices to register here, so by using Eureka Server, they can communicate with each other easily.
5. It also helps in load balancing and resolves hardcoding issue
6. One important thing is that in Eureka Server whatever Microservices are connected they don't need to know the exact addresses of each other.

Dependency

```
<spring-cloud.version>2023.0.3</spring-cloud.version>
```

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

```
<dependency>

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

  <artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>

</dependency>
```

```
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-dependencies</artifactId>
      <version>${spring-cloud.version}</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>
</dependencyManagement>
```

Application Properties

```
spring.application.name=ServicerRegistry
server.port =8761
eureka.instance.hostname=localhost
eureka.client.register-with-eureka=false // this prevent eureka server from
register itself to another Eureka Server
eureka.client.fetch-registry=false //means client unable to fetch Eureka
Registry Information form Eureka Server
```

Annotations

1. @EnableEurekaServer

How To Register Microservices on Eureka Server

Add Dependencies

```
<spring-cloud.version>2023.0.3</spring-cloud.version>
```

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

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

```
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-dependencies</artifactId>
      <version>${spring-cloud.version}</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>
</dependencyManagement>
```

Application Properties

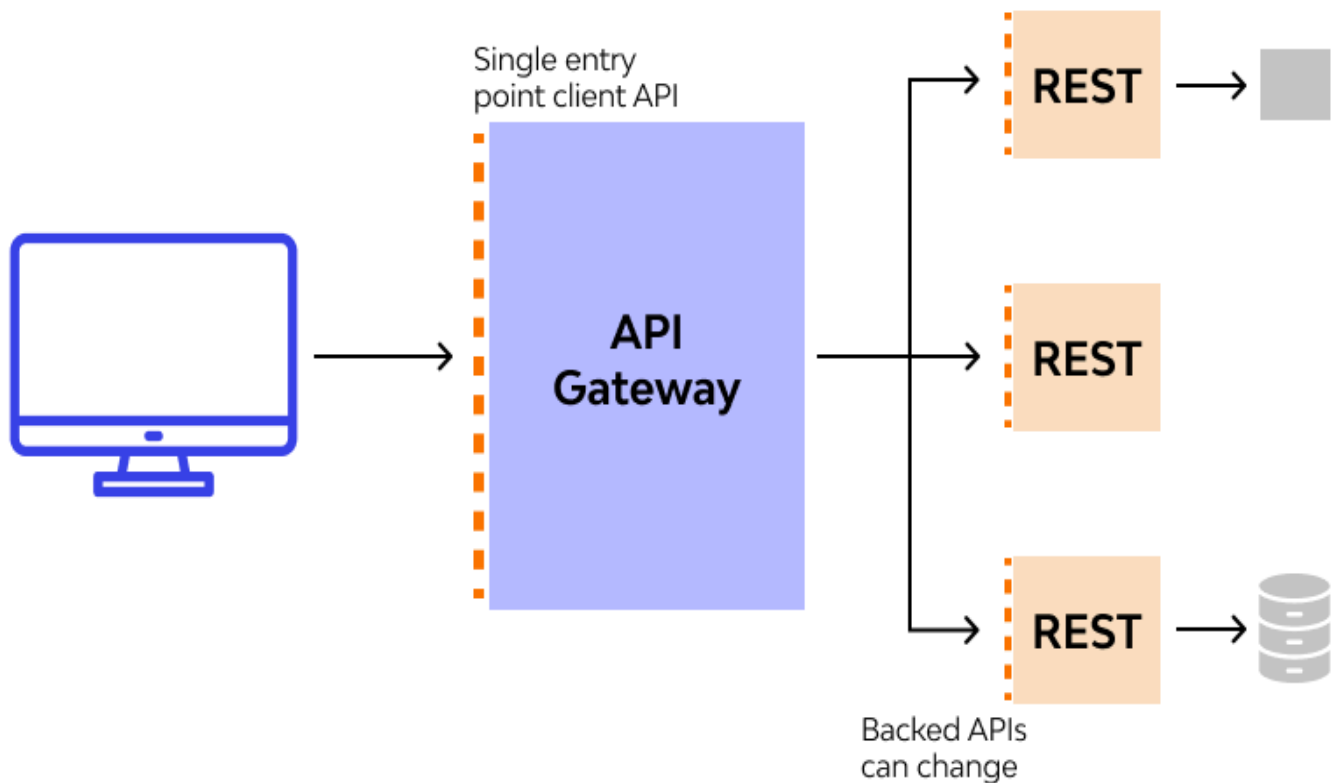
```
#eureka.instance.prefer-ip-address=true //it is use for communication instead  
of hostname  
  
#eureka.client.register-with-eureka=true  
  
#eureka.client.fetch-registry=true  
  
#eureka.client.service-url.defaultZone=http://localhost:8761/eureka
```

Annotations

1. EnableDiscoveryClient (use above this on your main method) //EnableEurekaClient

API GATEWAY

What is an API Gateway?



1. It is Microservices and
2. You don't need to send manually requests for different microservices , because all microservices are connected to Api Gateway so clients can easily retrieve data from API Gateway.
3. You just send a request to Api Gateway other internal implementation or work take care by API GATEWAY.
4. This is like your middleware between client and microservices.
5. Through API Gateway clients can easily interact with multiple microservices.

Example:-

<http://localhost:9003/ratings>

```
-> [
  {
    "ratingId": "1b05f496-3410-472a-aa80-a7fe798fba2f",
    "userId": "ff6e1376-b1af-40db-a9a2-11521217cb39",
    "hotelId": "e9f3e864-db6f-4a70-8395-e2679206318d",
    "rating": 8,
    "feedback": "this hotel is very good"
  },
  {
    "ratingId": "ea04e962-dbc2-4d34-b00c-a41dd8e208b5",
    "userId": "7517f3b7-f2e1-410d-9836-a45e2e7450c2",
    "hotelId": "478de6ff-bbda-4496-828c-6a90b0873a50",
    "rating": 9,
    "feedback": "this hotel is osm"
  },
  {
    "ratingId": "4dcf029b-4b7d-4940-b40c-e782cc93af47",
    "userId": "7517f3b7-f2e1-410d-9836-a45e2e7450c2",
    "hotelId": "d50e732f-b607-41c9-83b9-779d31da2c42",
    "rating": 5,
    "feedback": "this hotel is not good"
  }
]
```

CREATE API GATEWAY

```
<spring-cloud.version>2023.0.3</spring-cloud.version>
```

```
dependency>    //include the core components of Spring Cloud in your project
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter</artifactId>
</dependency>
<dependency>  //used to include Spring Cloud Gateway in your project
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-gateway</artifactId>
</dependency>
<dependency> //this is used to act like Eureka client
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
</dependency>
```

Register API Gateway on Eureka Server + @EnableDiscoveryClient (ON MAIN METHOD)

```
#eureka.instance.prefer-ip-address=true
#eureka.client.register-with-eureka=true
#eureka.client.fetch-registry=true
#eureka.client.service-url.defaultZone=http://localhost:8761/eureka
```

Configuration

```
@Configuration
public class ApiGatewayConfig {

    @Bean
    public RouteLocator routeLocator(RouteLocatorBuilder builder) {

        return builder

            .routes()

            .route(predicateSpec -> predicateSpec.path("/users/**") //or
("//**")

                .uri("lb://USERSERVICE"))

            .route(predicateSpec -> predicateSpec.path("/ratings/**")

                .uri("lb://RATINGSERVICE"))

            .route(predicateSpec ->
predicateSpec.path("/hotels/**", "/staffs/**")

                .uri("lb://HOTELSERVICE"))

            .build();

    }

}
```




System Status

| | | | |
|-------------|---------|--------------------------|---------------------------|
| Environment | test | Current time | 2024-09-10T00:26:05 -0700 |
| Data center | default | Uptime | 10:27 |
| | | Lease expiration enabled | false |
| | | Renews threshold | 10 |
| | | Renews (last min) | 10 |

EMERGENCY! EUREKA MAY BE INCORRECTLY CLAIMING INSTANCES ARE UP WHEN THEY'RE NOT. RENEWALS ARE LESSER THAN THRESHOLD AND HENCE THE INSTANCES ARE NOT BEING EXPIRED JUST TO BE SAFE.

DS Replicas

Instances currently registered with Eureka

| Application | AMIs | Availability Zones | Status |
|---------------|---------|--------------------|---|
| APIGATEWAY | n/a (1) | (1) | UP (1) - Win10-Dev01:ApiGateway:9003 |
| CONFIGSERVER | n/a (1) | (1) | UP (1) - Win10-Dev01:ConfigServer:9004 |
| HOTELSERVICE | n/a (1) | (1) | UP (1) - Win10-Dev01:HotelService:9001 |
| RATINGSERVICE | n/a (1) | (1) | UP (1) - Win10-Dev01:RatingService:9002 |
| USERSERVICE | n/a (1) | (1) | UP (1) - Win10-Dev01:UserService:9000 |