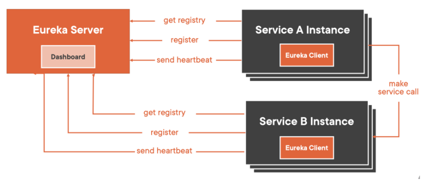
SpringBoot - Microservices – SpringCloud Eureka Registering and Discovering Server

✓ **Aufgaben TrackZilla**



Registering and Discovering services with SpringCloud - Eureka
Exercises TrackZilla with SpringCloud - Eureka

Description
We will create an Eureka Server, then create Services and register them, we than discover the services, we will load balance the services on the clients. Finally we will create som healthchecks. What we saw on the Demo-App now we will implement on our TrackZilla Applikation and Services!

Zeitbedarf:	4 Lektionen
Hilfsmittel:	SpringBoot, SpringCloud
Methode/Sozialform:	, ,
Lernziele:	✓ **SpringCloud Eureka – Server** ✓ **Aufgaben TrackZilla** ✓ **Registering Services:Application-Catalog-Service, Ticket-Management-Service and User-Management-Service** ✓ **Discovering Services: dito** ✓ **LoadBalancing the RESTful Methods** ✓ **Healthcheck**

Legende: Einzelarbeit, Partnerarbeit, Dokumentation, Code, Präsentation

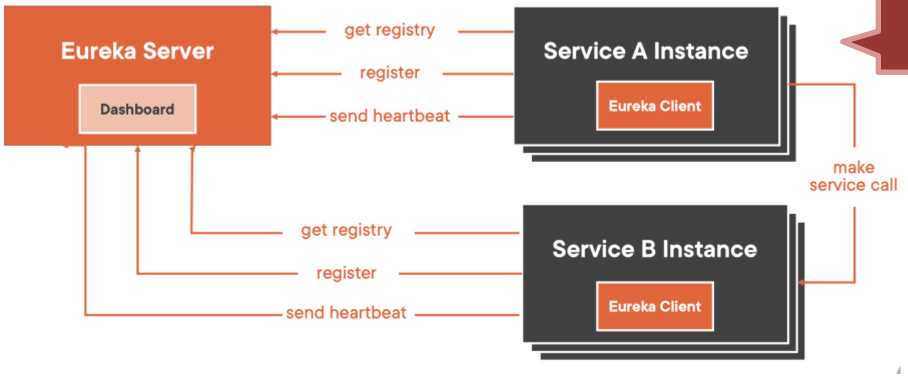
Inhaltsverzeichnis

1 Exercises (hands on)

1.1 Exercise: Creating an Eureka Server

Spring Cloud Microservices Setup:

1.2 Components of Eureka Environment



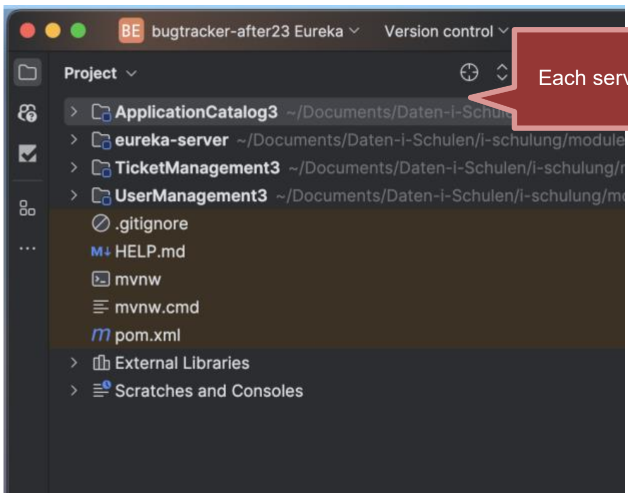
Architecture of Eureka

First, let's set up the basic Spring Cloud infrastructure:

1.3 Eureka Server:

1.4 Exercise: Preparing the project

We know now how to work with modules in Intellij, we will do that here and prepare the project with the services and the eureka server:

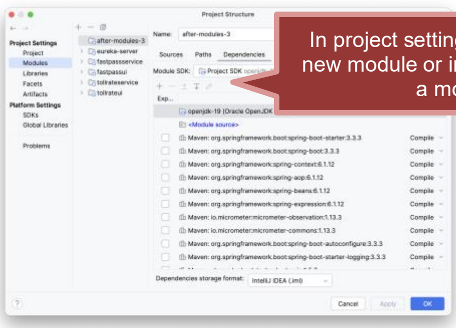


bugtracker-after23 Eureka   Version control

Project

- ApplicationCatalog3 ~/Documents/Daten-i-Schul
- eureka-server ~/Documents/Daten-i-Schulen/i-schulung/module
- TicketManagement3 ~/Documents/Daten-i-Schulen/i-schulung/
- UserManagement3 ~/Documents/Daten-i-Schulen/i-schulung/m
- .gitignore
- HELP.md
- mvnw
- mvnw.cmd
- pom.xml
- External Libraries
- Scratches and Consoles

Each service is a module in intellij!



Project Structure

Project Settings: Project, Modules, Libraries, Facets, Artifacts
Platform Settings: SDKs, Global Libraries
Problems

Modules: after-modules-3, eureka-server, fastpassservice, fastpassui, tollrateservice, tollrateui

Name: after-modules-3

Sources | Paths | Dependencies

Module SDK: Project SDK

Exp...		
	openjdk-19 (Oracle OpenJDK	
	<Module source>	
☐	Maven: org.springframework.boot:spring-boot-starter:3.3.3	Compile
☐	Maven: org.springframework.boot:spring-boot:3.3.3	Compile
☐	Maven: org.springframework:spring-context:6.1.12	Compile
☐	Maven: org.springframework:spring-aop:6.1.12	Compile
☐	Maven: org.springframework:spring-beans:6.1.12	Compile
☐	Maven: org.springframework:spring-expression:6.1.12	Compile
☐	Maven: io.micrometer:micrometer-observation:1.13.3	Compile
☐	Maven: io.micrometer:micrometer-commons:1.13.3	Compile
☐	Maven: org.springframework.boot:spring-boot-autoconfigure:3.3.3	Compile
☐	Maven: org.springframework.boot:spring-boot-starter-logging:3.3.3	Compile

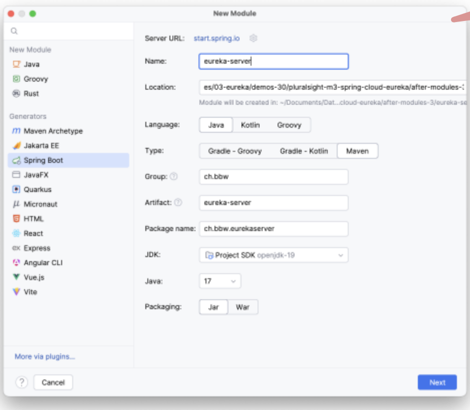
Dependencies storage format: IntelliJ IDEA (.iml)

Cancel | Apply | OK

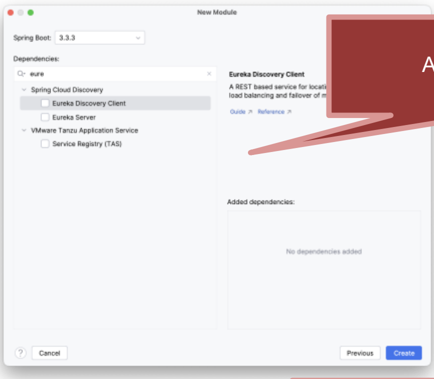
In project settings you can add e new module or import a project as a module!

1.5 Create an Eureka Server with spring initializr in intellij

Add a new module and use the initializr directly !



Server URL: start.spring.io
Name: eureka-server
Location: es/03-eureka/demos-30/pluralsight-m3-spring-cloud-eureka/after-modules-
Module will be created in: ~/Documents/Dat...cloud-eureka/after-modules-3/eureka-se
Language: [✓ Java] Kotlin Groovy
Type: Gradle - Groovy Gradle - Kotlin [✓ Maven]
Group: ch.bbw
Artifact: eureka-server
Package name: ch.bbw.eurekaserver
JDK: Project SDK openjdk-19
Java: 17
Packaging: [✓ Jar] War



Spring Boot: 3.3.3
Dependencies:
eure
Spring Cloud Discovery
Eureka Discovery Client
Eureka Server
VMware Tanzu Application Service
Service Registry (TAS)

Eureka Discovery Client
A REST based service for locati
load balancing and failover of m
Guide ↗ Reference ↗

Added dependencies:
No dependencies added

Add Eureka Server!

Maven dep.

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

In your main application class:

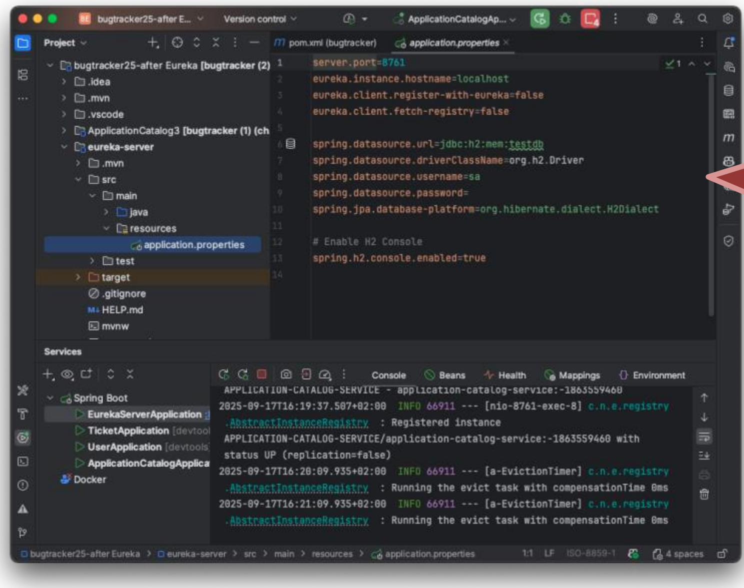
```
@EnableEurekaServer
@SpringBootApplication
public class EurekaServerApplication {

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

}
```

@EnableEurekaServer

Selecting the properties:



```
server.port=8761
eureka.instance.hostname=localhost
eureka.client.register-with-eureka=false
eureka.client.fetch-registry=false

spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

# Enable H2 Console
spring.h2.console.enabled=true
```

```
APPLICATION-CATALOG-SERVICE - application-catalog-service:-1863559460
2025-09-17T16:19:37.507+02:00  INFO 66911 --- [nio-8761-exec-8] c.n.e.registry
.AbstractInstanceRegistry  : Registered instance
APPLICATION-CATALOG-SERVICE/application-catalog-service:-1863559460 with
status UP (replication=false)
2025-09-17T16:20:09.935+02:00  INFO 66911 --- [a-EvictionTimer] c.n.e.registry
.AbstractInstanceRegistry  : Running the evict task with compensationTime 0ms
2025-09-17T16:21:09.935+02:00  INFO 66911 --- [a-EvictionTimer] c.n.e.registry
.AbstractInstanceRegistry  : Running the evict task with compensationTime 0ms
```

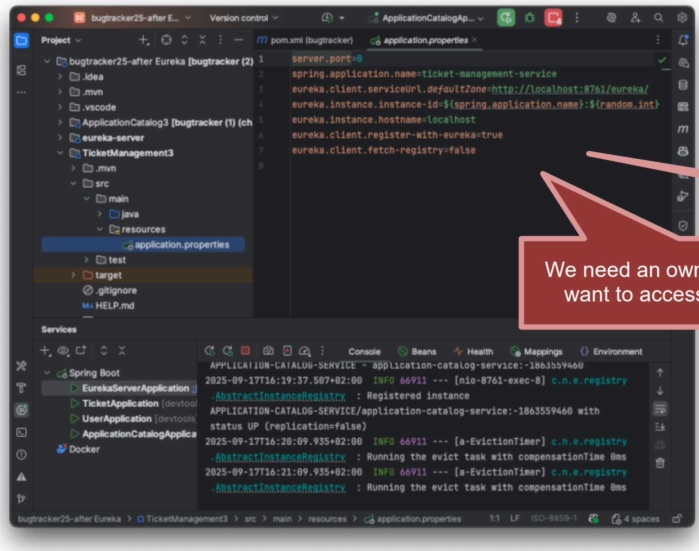
Default port for Eureka Server
hostname: localhost
register with eureka: false (we are eureka)
Fetch-registry: also false

1.6 Configure and register each service

Add to class path:

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

The fastpass-service and the tollrate-service are configured in the same way.



```
server.port=0
spring.application.name=ticket-management-service
eureka.client.serviceUrl.defaultZone=http://localhost:8761/eureka/
eureka.instance.instance-id=${spring.application.name}:${random.int}
eureka.instance.hostname=localhost
eureka.client.register-with-eureka=true
eureka.client.fetch-registry=false
```

```
 APPLICATION-CATALOG-SERVICE - application-catalog-service:-1863559460
2025-09-17T16:19:37.507+02:00  INFO 66911 --- [nio-8761-exec-8] c.n.e.registry
 .AbstractInstanceRegistry  : Registered instance
 APPLICATION-CATALOG-SERVICE/application-catalog-service:-1863559460 with
 status UP (replication=false)
2025-09-17T16:20:09.935+02:00  INFO 66911 --- [a-EvictionTimer] c.n.e.registry
 .AbstractInstanceRegistry  : Running the evict task with compensationTime 0ms
2025-09-17T16:21:09.935+02:00  INFO 66911 --- [a-EvictionTimer] c.n.e.registry
 .AbstractInstanceRegistry  : Running the evict task with compensationTime 0ms
```

bugtracker25-after Eureka
Version control
ApplicationCatalogAp...
pom.xml (bugtracker)
application.properties
Project
bugtracker25-after Eureka [bugtracker (2)]
.idea
.mvn
.vscode
ApplicationCatalog3 [bugtracker (1) (ch
eureka-server
TicketManagement3
.mvn
src
main
java
resources
application.properties
test
target
.gitignore
HELP.md
Services
Spring Boot
EurekaServerApplication
TicketApplication [devtools]
UserApplication [devtools]
ApplicationCatalogApplica
Docker
Console
Beans
Health
Mappings
Environment
bugtracker25-after Eureka > TicketManagement3 > src > main > resources > application.properties
1:1 LF ISO-8859-1 4 spaces

Port=0 defines a random Port

Name and url

Instance id

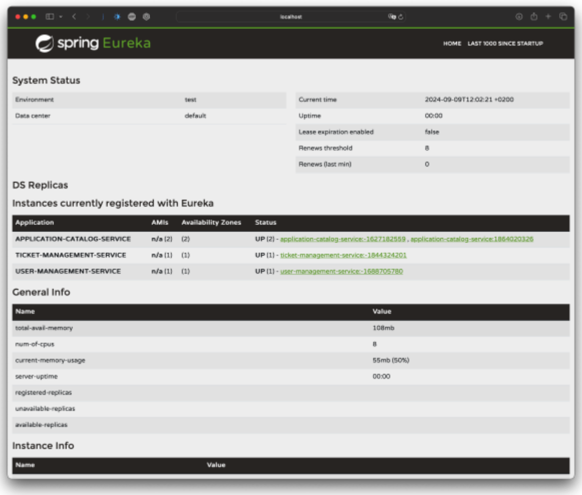
Registering on Eureka

We need an own registry only if we want to access another service

In each microservice's main class:

```
@SpringBootApplication
@EnableDiscoveryClient
public class MicroserviceApplication {
    public static void main(String[] args) {
        SpringApplication.run(MicroserviceApplication.class, args);
    }
}
```

2 Using the Eureka Dashboard



System Status

Environment	test
Data center	default

Current time	2024-09-09T12:02:21 +0200
Uptime	00:00
Lease expiration enabled	false
Renews threshold	8
Renews (last min)	0

DS Replicas

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
APPLICATION-CATALOG-SERVICE	n/a (2)	(2)	UP (2) - application-catalog-service:-1627182559 , application-catalog-service:1864020326
TICKET-MANAGEMENT-SERVICE	n/a (1)	(1)	UP (1) - ticket-management-service:-1844324201
USER-MANAGEMENT-SERVICE	n/a (1)	(1)	UP (1) - user-management-service:-1688705780

General Info

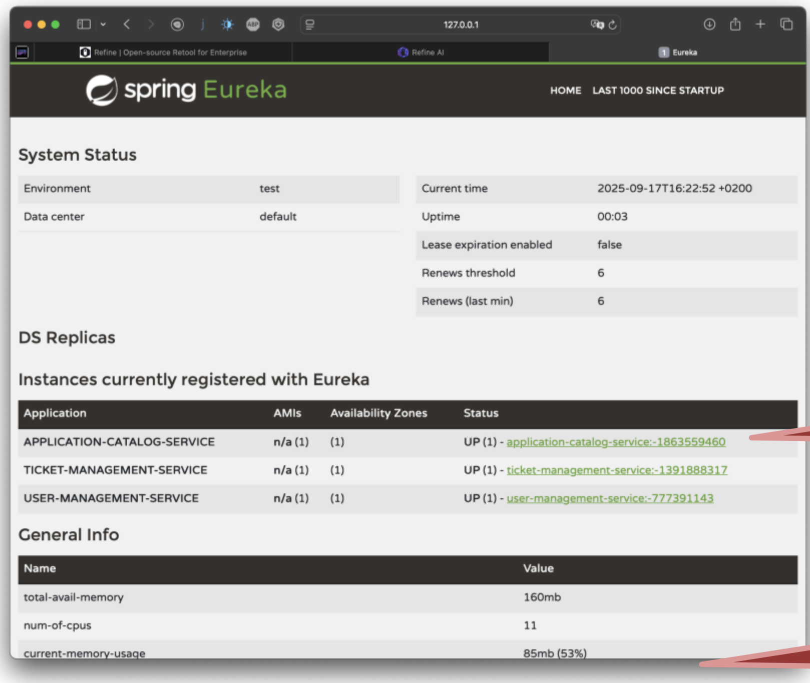
Name	Value
total-avail-memory	108mb
num-of-cpus	8
current-memory-usage	55mb (50%)
server-uptime	00:00
registered-replicas	
unavailable-replicas	
available-replicas	

Instance Info

Name	Value

- Enabled by default
- Shows environment info
- Lists registered services and instances
- View service health

Whe you run the Eureka-Server you can see the dashboard!



System Status

Environment	test
Data center	default

Current time	2025-09-17T16:22:52 +0200
Uptime	00:03
Lease expiration enabled	false
Renews threshold	6
Renews (last min)	6

DS Replicas

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
APPLICATION-CATALOG-SERVICE	n/a (1)	(1)	UP (1) - application-catalog-service:-1863559460
TICKET-MANAGEMENT-SERVICE	n/a (1)	(1)	UP (1) - ticket-management-service:-1391888317
USER-MANAGEMENT-SERVICE	n/a (1)	(1)	UP (1) - user-management-service:-777391143

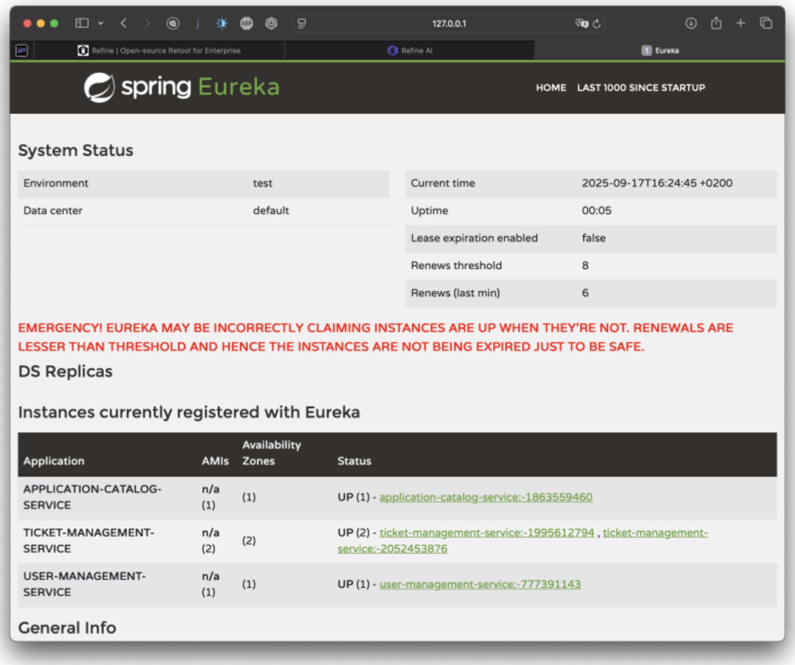
General Info

Name	Value
total-avail-memory	160mb
num-of-cpus	11
current-memory-usage	85mb (53%)

You do not see the services yet!

Start more instances

mvn spring-boot:run



System Status

Environment	test
Data center	default

Current time	2025-09-17T16:24:45 +0200
Uptime	00:05
Lease expiration enabled	false
Renews threshold	8
Renews (last min)	6

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
APPLICATION-CATALOG-SERVICE	n/a (1)	(1)	UP (1) - application-catalog-service:-1863559460
TICKET-MANAGEMENT-SERVICE	n/a (2)	(2)	UP (2) - ticket-management-service:-1995612794 , ticket-management-service:-2052453876
USER-MANAGEMENT-SERVICE	n/a (1)	(1)	UP (1) - user-management-service:-777391143

General Info

Start more instances

That's it :)