### 02-eureka-server

1. START 02 EUREKA SERVER
2. SHOW CONSOLE EUREKA
3. START 06 7111
4. SHOW REGISTRATION
5. START 06 SERVICE CONSUMER
6. SHOW /LIST SERVICES
7. SHOW /

Overview of the project

File yml etc

Configurations of Eureka

Run project from command line .bat

Show console



#### Detail the message “THE SELF PRESERVATION MODE IS TURNED OFF.THIS MAY NOT PROTECT INSTANCE EXPIRY IN CASE OF NETWORK/OTHER PROBLEMS”

2016-09-08 14:52:35.431 WARN 8232 --- [a-EvictionTimer] c.n.eureka.PeerAwareInstanceRegistry : The self preservation mode is disabled!. Hence allowing the

instances to expire.

<https://github.com/ExampleDriven/spring-cloud-eureka-example/blob/master/eureka-server/src/main/resources/application.yml>

http://stackoverflow.com/questions/33921557/understanding-spring-cloud-eureka-server-self-preservation-and-renew-threshold

### 06\_bookABatterySERVICE4EUREKA

Overview of the project

Eureka directives on application .class

File yml etc

Run project from command line .bat

Check registration



#### 06\_bookABatterySERVICE4EUREKA 7113-7115

Overview of the project

Eureka directives on application .class

File yml etc

Run project from command line .bat

Check registration



### Discovery

### 06\_bookABatteryCLIENT\_DISCOVERY\_SERVICE

Code description

Run from .bat

Call:

## Load balancing with ribbon

### 06\_bookABatteryCLIENT\_FEIGN\_SERVICE

Show implementation

<http://sdpsvrsa094:7112/>

show requested on the tree instance



Stop two and how requested on the one running





Suppress warn and info on services display

## PWS Load Balancing

**[PWS DISCOVERY AND BALANCING]**

Predeployed application only one service

Pivotal ws disco and balance

requirements satisfaction about

simple Scale up/down

No downtime during scale up

No load balancing configuration

Resolution of instance made by pws mechanisms

Describe eureka console

1. Service
2. Consumer

Launch test of balancing showing that only one instance is replying

LIVE: Scale instance of service then wait enough time (**take time**)

Launch test of available instances until discovered

Launch test of balancing

http://06-bookabatteryclient\_discovery\_service.cfapps.io/listDiscovery

Launch test of balancing

**[EUREKA ALSO DEFINITION]**

server:

port: 8761

eureka:

numberRegistrySyncRetries: 1

instance:

hostname: localhost

client:

registerWithEureka: false

fetchRegistry: false

serviceUrl:

defaultZone: http://${eureka.instance.hostname}:${server.port}/eureka/

server:

enable-self-preservation: true

Explaining the most important lines:

server:

port: 8761

Here, we are configuring the Eureka Server to run on port 8761. This is the default port and you can change, but you need to give this port to the clients later on.

eureka:

numberRegistrySyncRetries: 1

If you are running locally, there is a 2 to 3 minutes wait until fulling boot up. This happens because Eureka will be looking for peers. To disable this, set to 0 (although you should never do this in production)

client:

registerWithEureka: false

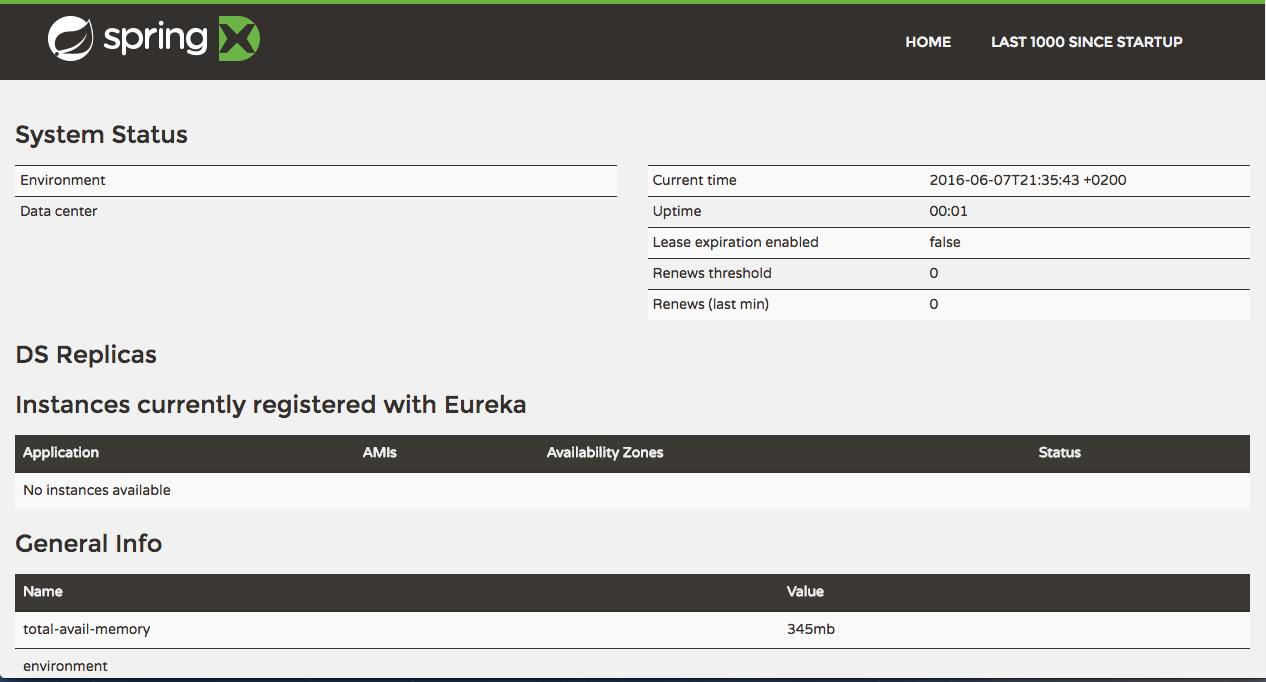
As this is the Eureka Server, we do not want it to register itself. Will always be set to false on server and true on the clients.

serviceUrl:

defaultZone: http://${eureka.instance.hostname}:${server.port}/eureka/

the defaultZone is the fallback URL for every client that doesn't specify a preference for a server.

Now, the only thing we should do is enable the Eureka Server.   
To do so, go to the main class (at this point, you should only have one class in the project, though) and annotate it with @EnableEurekaServer, as follow



# Backup

The historic district of Rome  
(Prima sfide)  
Storia  
Romans  
Middle age  
900

So it is

A possible solution that realize both mobility and respect of environment could be led by smart electrical vehicle. Small that they can easily move among the small Street without no emission

The hystorical center of rrome is full of business activity, besides the seat of govermant institution so goods delivery is a critical   
To be much more

These vehicle colud lead an ideal  24/7 service without charging stops

Mobility sould be conducted in respect of ancient monuments and fragile buildings

Pit stop

Rome is the city where I live since I wos born. It counts about 3.5 official inhabitants to witch we have to add about 1 million of non resident people (students, tourists, workers). To these figures we have also to add about 700 vehicles each 1000 people, a rate that makes mobility a challenge.

Everyone could easily realize that with there figures besides mobility there are also pollution issues.

~~A partial solution of these issues could be addressed reducing the vehicles that daily deliver goods.~~

For a more sustainable for the environment AND RESPECTING THIS ARCHITECTONICAL VALUES and efficient mobility in such a context the daily delivery of goods could be done by means of (through) “smart” electrical vehicles.

~~To overcome the limited life of the batteries these vehicles has been enginnered with special battery pack that could be easily and rapidly changed in dedicated service station.~~

~~In this way these vehicle will not need to stop for charging their batteries and so could guarantee much more short breaks~~

~~Could be much more efficient in comparison to the other electric vehicle (that do non support this kind of batteries – not engineered in such a way)~~

I do not know if this is a realizable scenario but if so it is necessary a software solution by witch manage this scenario.

Each driver of this ecological delivery company according to his delivery plan each morning program the expected battery changes that will reasonably occur during a day. By means of a mobile application (web application) he will book one or more fresh batteries supplied by the stations distributed in the city.

In case of emergency it will be the vehicle itself that will notify to the driver the necessity to come to the nearest station with available fresh batteries, find by a software running in the vehicle.