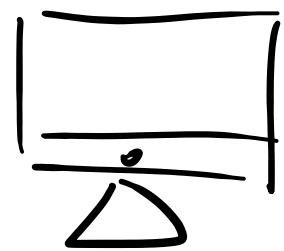
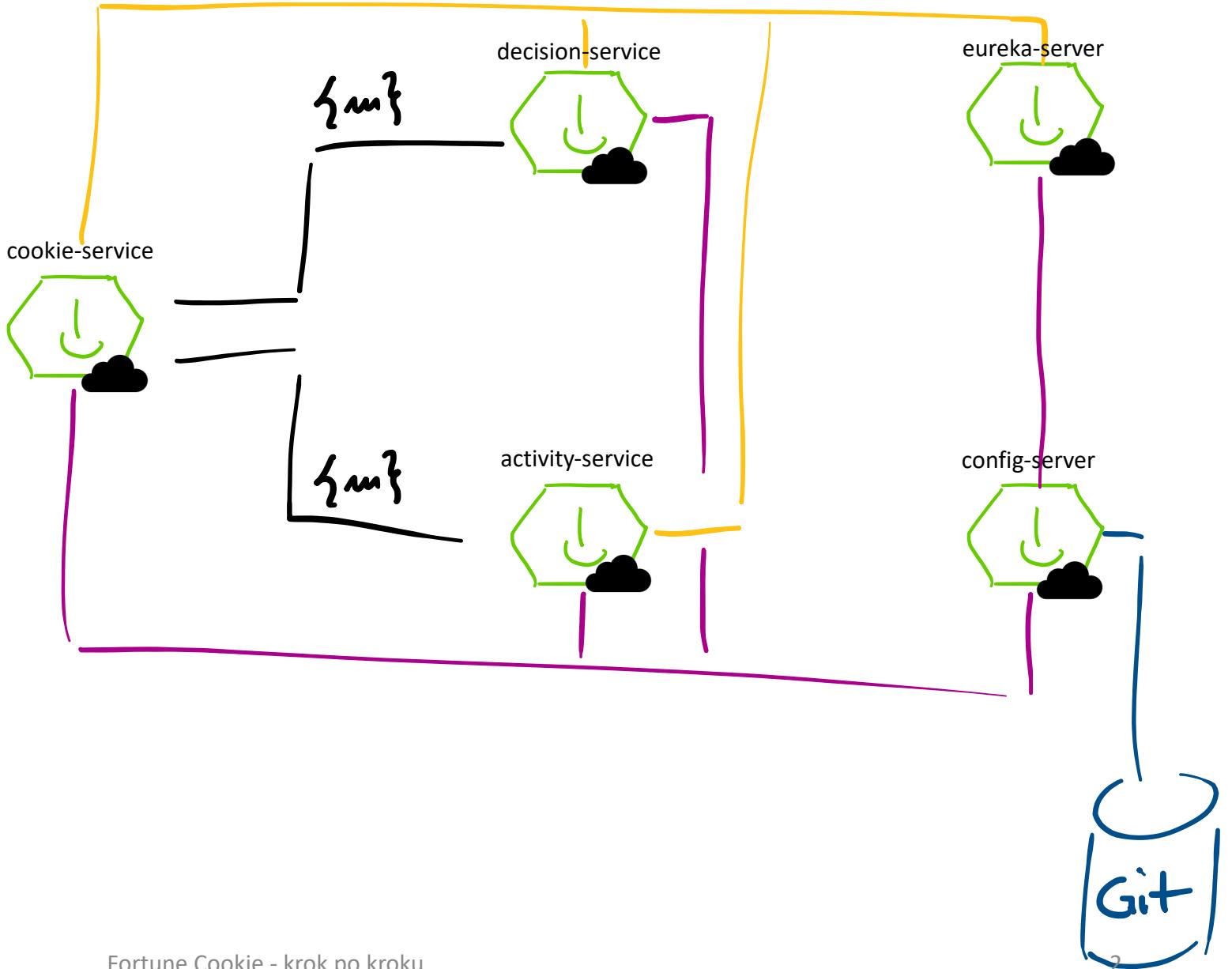


# Fortune Cookie



{mu}



# Hosts – dodanie wpisów

<https://www.howtogeek.com/howto/27350/beginner-geek-how-to-edit-your-hosts-file/>

```
#START fortune-cookie
127.0.0.1      eureka-peer1
127.0.0.1      eureka-peer2
127.0.0.1      config-server
#END fortune-cookie
```

# Hosts - test

```
uki@iMac ~ % ping -c1 eureka-peer1
PING eureka-peer1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.040 ms
```

# InteliJ - Inicjalizacja projektu

- ▶  activity-service 4 files ~/Desktop/noble/fortune-cookie/activity-service
- ▶  config-server 4 files ~/Desktop/noble/fortune-cookie/config-server
- ▶  cookie-service 4 files ~/Desktop/noble/fortune-cookie/cookie-service
- ▶  decision-service 4 files ~/Desktop/noble/fortune-cookie/decision-service
- ▶  eureka-server 4 files ~/Desktop/noble/fortune-cookie/eureka-server
- ▶  message 1 file ~/Desktop/noble/fortune-cookie/message

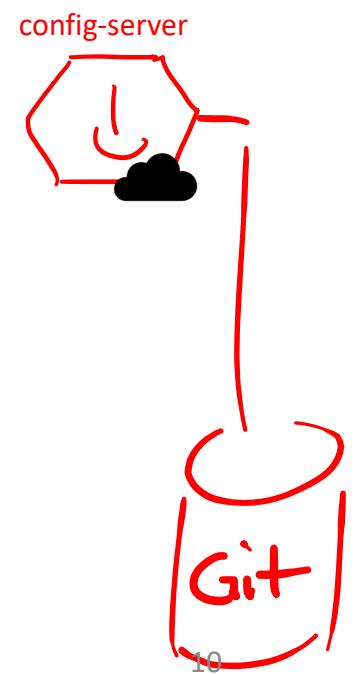
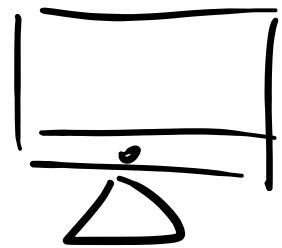
Spring Boot

MAVEN

① Inicjalizacja projektu (EMPTY) , DODANIE MODUŁÓW :

- ② - spring boot : eureka-server, config-server, decision-service, activity-service, cookie-service
- ③ - maven: message

# config-server





## Config Server

- ① - dodanie zależności do pom.xml
- ② - konfiguracja repozytorium w application.yaml
- ③ - @EnableConfigServer w klasie głównej

```

<name>config-server</name>
<description>Fortune Cookie - config server</description>
<properties>
    <java.version>11</java.version>
</properties>
<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter</artifactId>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>

<project>

```

**CONFIG SERVER**

```

14 15          <description>Fortune Cookie - config server</description>
15 16          <properties>
16 17              <java.version>11</java.version>
17 18          <spring-cloud.version>2020.0.2</spring-cloud.version>
18 19      </properties>
19 20      <dependencies>
20 21          <dependency>
21 22              <groupId>org.springframework.cloud</groupId>
22 23              <artifactId>spring-cloud-config-server</artifactId>
23 24      </dependency>
24 25      <dependency>
25 26          <groupId>org.springframework.boot</groupId>
26 27          <artifactId>spring-boot-starter-test</artifactId>
27 28          <scope>test</scope>
28 29      </dependency>
29 30  </dependencies>
30 31  <dependencyManagement>
31 32      <dependencies>
32 33          <dependency>
33 34              <groupId>org.springframework.cloud</groupId>
34 35              <artifactId>spring-cloud-dependencies</artifactId>
35 36              <version>${spring-cloud.version}</version>
36 37              <type>pom</type>
37 38              <scope>import</scope>
38 39          </dependency>
39 40      </dependencies>
40 41  </dependencyManagement>
41 42

```

```
1 ②spring:  
2   cloud:  
3     config:  
4       server:      REPOZYTORIUM (MOŻNA LOKALNE LUB PLIKI)  
5         git:  
6           uri: https://github.com/lukasze/fortune-cookie.git  
7     application:  
8       name: config-server  
9   server:  
10    port: 8888      DEFAULT
```

ConfigServerApplication.java (/Users/uki/Desktop/noble/fortune-cookie/config-server/src/main/java/com/example/configserver)

d2c4f30165484a3766ed6a2f86feaf71e4412b10

857b2e3ad78a3654d99aad7313c169fdf7669fab

2 differences

```
package com.example.configserver;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication
public class ConfigServerApplication {

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

1 1 package com.example.configserver;

2 2

3 3 import org.springframework.boot.SpringApplication;

4 4 import org.springframework.boot.autoconfigure.SpringBootApplication;

5 5 import org.springframework.cloud.config.server.EnableConfigServer;

6 6

7 7 @SpringBootApplication

8 8 @EnableConfigServer (3)

9 9 public class ConfigServerApplication {

10 10

11 11 public static void main(String[] args) {

12 12 SpringApplication.run(ConfigServerApplication.class, args);

13 13 }

14 14 }

# config-server - test

# Services



▼ Spring Boot

▼ Running

**ConfigServerApplication :8888/**



localhost:8888/eureka-server/

localhost:8888/eureka-server/peer1

```
{  
  "name": "eureka-server",  
  "profiles": [  
    "peer1"  
  ],  
  "label": null,  
  "version": "c8febb1c16816e7563ceb937f8bf3f2dd634b041",  
  "state": null,  
  "propertySources": [  
    {  
      "name": "https://github.com/lukasze/fortune-cookie.git/Config resource 'file [/var/folders/vb/td8mqmcn1656371bwn8w8phm0015484258505888150041/eureka-server.yml' via location '' (document #1)",  
      "source": {  
        "spring.profiles": "peer1",  
        "eureka.instance.hostname": "eureka-peer1",  
        "eureka.instance.appname": "${my-eureka-cluster.name}",  
        "eureka.client.registerWithEureka": true,  
        "eureka.client.fetchRegistry": true,  
        "eureka.client.serviceUrl.defaultZone": "http://eureka-peer2:8762/eureka",  
        "server.port": "${my-eureka-cluster.port.peer1}"  
      }  
    },  
    {  
      "name": "https://github.com/lukasze/fortune-cookie.git/Config resource 'file [/var/folders/vb/td8mqmcn1656371bwn8w8phm0015484258505888150041/eureka-server.yml' via location '' (document #0)",  
      "source": {  
        "my-eureka-cluster.port.peer1": 8761,  
        "my-eureka-cluster.port.peer2": 8762,  
        "my-eureka-cluster.name": "eureka-cluster"  
      }  
    }  
  ]  
}
```

github.com/lukasze/fortune-cookie/blob/master/eureka-server.yml

Why GitHub? Team Enterprise Explore Marketplace Pricing

lukasze / fortune-cookie

Code Issues Pull requests Actions Projects Wiki Security Insights

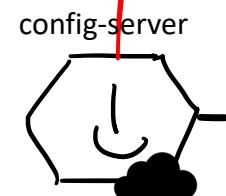
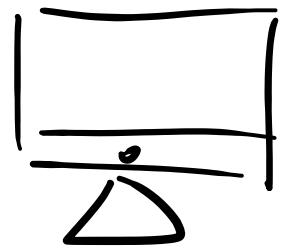
master fortune-cookie/eureka-server.yml

lukasze clean up

2 contributors

```
33 lines (33 sloc) | 670 Bytes  
1 my-eureka-cluster:  
2   port:  
3     peer1: 8761  
4     peer2: 8762  
5   name: eureka-cluster  
6 ---  
7   spring:  
8     profiles: peer1  
9   eureka:  
10    instance:  
11      hostname: eureka-peer1  
12      appname: ${my-eureka-cluster.name}  
13    client:  
14      registerWithEureka: true  
15      fetchRegistry: true  
16      serviceUrl:  
17        defaultZone: http://eureka-peer2:8762/eureka  
18    server:  
19      port: ${my-eureka-cluster.port.peer1}  
20 ---  
21   spring:  
22     profiles: peer2  
23   eureka:  
24    instance:  
25      hostname: eureka-peer2  
26      appname: ${my-eureka-cluster.name}  
27    client:  
28      registerWithEureka: true  
29      fetchRegistry: true  
30      serviceUrl:  
31        defaultZone: http://eureka-peer1:8761/eureka  
32    server:  
33      port: ${my-eureka-cluster.port.peer2}
```

# eureka-server



pom.xml (/Users/uki/Desktop/noble/fortune-cookie/eureka-server)

857b2e3ad78a3654d99aad7313c169fdf7669fab

<version>0.0.1-SNAPSHOT</version>

<name>eureka-server</name>

<description>Fortune Cookie - eureka server</description>

<properties>

    <java.version>11</java.version>

</properties>

<dependencies>

    <dependency>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter</artifactId>

    </dependency>

    <dependency>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-test</artifactId>

        <scope>test</scope>

    </dependency>

</dependencies>

<build>

    <plugins>

        <plugin>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-maven-plugin</artifactId>

        </plugin>

    </plugins>

</build>

</project>

ef2249db501abc966760b751661c177a7ece59a7

<spring-cloud.version>2020.0.2</spring-cloud.version>

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

EUREKA  
CONFIG  
SERVER  
(CLIENT)

Fortune Cookie - krok po kroku

4 differences

1

</properties>

<dependencies>

    <dependency>

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

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

    </dependency>

    <dependency>

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

        <artifactId>spring-cloud-starter-config</artifactId>

    </dependency>

    <dependency>

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

        <artifactId>spring-cloud-starter-bootstrap</artifactId>

    </dependency>

    <dependency>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-test</artifactId>

        <scope>test</scope>

    </dependency>

</dependencies>

<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>



↑ ↓ ⌚ | ← → ⏪

Side-by-side viewer ▾

Do not ignore ▾

Highlight words ▾



ef2249db501abc966760b751661c177a7ece59a7

1 ② **spring:**

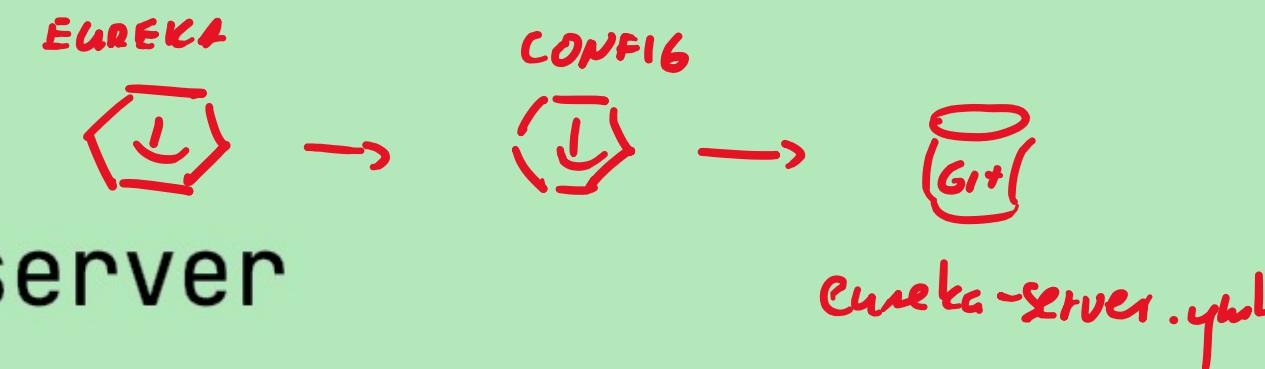
2     **application:**

3         **name:** eureka-server

4     **cloud:**

5     **config:**

6         **uri:** http://config-server:8888



DODANE W HOSTS

```
import org.springframework.boot.SpringApplication;
import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;

@SpringBootApplication
public class EurekaServerApplication {

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

import org.springframework.boot.SpringApplication;
import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;

@EnableEurekaServer
@SpringBootApplication
public class EurekaServerApplication {

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

The code shows two identical Spring Boot applications. The first application is annotated with `@SpringBootApplication`. The second application is annotated with `@EnableEurekaServer` and `@SpringBootApplication`. A red circle with the number '3' has been drawn around the `@EnableEurekaServer` annotation in the second application's code.

# eureka-server - test

Services

The Services tool window shows two running Spring Boot applications:

- ConfigServerApplication :8888/
- EurekaServerApplicationPeer1 :8761/

Configuration Tab (selected):

- Name: EurekaServerApplicationPeer1
- Main class: com.example.eurekasher.EurekaServerApplication
- Environment (expanded):
- Spring Boot (expanded):
  - Enable debug output:
  - Hide banner:
  - Enable launch optimization:
  - Enable JMX agent:
- Running Application Update Policies:
  - On 'Update' action: Do nothing
  - On frame deactivation: Do nothing
- Active profiles: peer1

## Services

The screenshot shows the Spring Boot Actuator 'Endpoints' page. On the left, there's a sidebar with various icons and a tree view showing a 'Spring Boot' node with a 'Running' child, and two applications listed: 'ConfigServerApplication :8888/' and 'EurekaServerApplicationPeer1 :8761/'. The main area has tabs for 'Console' and 'Endpoints'. The 'Console' tab displays application logs. A red circle highlights the log entry: 'INFO 55619 --- [main] c.c.c.ConfigServicePropertySourceLocator: Located environment: name=eureka-server, profiles=[peer1], ...'. Another red circle highlights the log entry: 'INFO 55619 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer: Tomcat initialized with port(s): 8761 (http) ...'. The logs also mention Jersey and Jackson JSON libraries.

```

    INFO 55619 --- [main] c.c.c.ConfigServicePropertySourceLocator: Fetching config from server at : http://config-server:8888
    INFO 55619 --- [main] c.c.c.ConfigServicePropertySourceLocator: Located environment: name=eureka-server, profiles=[peer1], ...
    INFO 55619 --- [main] b.c.PropertySourceBootstrapConfiguration: Located property source: [BootstrapPropertySource {name='boot...
    INFO 55619 --- [main] c.e.e.EurekaServerApplication: The following profiles are active: peer1
    INFO 55619 --- [main] o.s.cloud.context.scope.GenericScope: BeanFactory id=ddd47b32-9e86-3b27-84c6-492215acaec6
    INFO 55619 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer: Tomcat initialized with port(s): 8761 (http)
    INFO 55619 --- [main] o.apache.catalina.core.StandardService: Starting service [Tomcat]
    INFO 55619 --- [main] org.apache.catalina.core.StandardEngine: Starting Servlet engine: [Apache Tomcat/9.0.45]
    INFO 55619 --- [main] o.a.c.c.C.[Tomcat].[localhost].[]: Initializing Spring embedded WebApplicationContext
    INFO 55619 --- [main] w.s.c.ServletWebServerApplicationContext: Root WebApplicationContext: initialization completed in 894 m...
    INFO 55619 --- [main] c.s.j.s.i.a.WebApplicationImpl: Initiating Jersey application, version 'Jersey: 1.19.4 05/24/2018 ...
    INFO 55619 --- [main] c.n.d.provider.DiscoveryJerseyProvider: Using JSON encoding codec LegacyJacksonJson
    INFO 55619 --- [main] c.n.d.provider.DiscoveryJerseyProvider: Using JSON decoding codec LegacyJacksonJson
  
```

- ① URUCHOMIĘ 2 PROFILE ( PROFILE ZDEFINIONOWANE W eureka-server.yaml )
- ② CZY SERVIS POBIERA WEZWIĄZANIA CONFIGURACJE ?

STATELESS

```
com.netflix.discovery.shared.transport.TransportException Create breakpoint : Cannot execute request on any known server
at com.netflix.discovery.shared.transport.decorator.RetryableEurekaHttpClient.execute(RetryableEurekaHttpClient.java)
at com.netflix.discovery.shared.transport.decorator.EurekaHttpClientDecorator.register(EurekaHttpClientDecorator.java)
at com.netflix.discovery.shared.transport.decorator.EurekaHttpClientDecorator$1.execute(EurekaHttpClientDecorator.java)
at com.netflix.discovery.shared.transport.decorator.SessionedEurekaHttpClient.execute(SessionedEurekaHttpClient.java)
at com.netflix.discovery.shared.transport.decorator.EurekaHttpClientDecorator.register(EurekaHttpClientDecorator.java)
at com.netflix.discovery.DiscoveryClient.register(DiscoveryClient.java:876) ~[eureka-client-1.10.11.jar:1.10.11]
at com.netflix.discovery.InstanceInfoReplicator.run(InstanceInfoReplicator.java:121) ~[eureka-client-1.10.11.jar:1.10.11]
at com.netflix.discovery.InstanceInfoReplicator$1.run(InstanceInfoReplicator.java:101) ~[eureka-client-1.10.11.jar:1.10.11]
```

## Services

The screenshot shows the Spring Boot Actuator Services UI. On the left, there's a sidebar with various icons. The main area lists services under 'Spring Boot'.

- Running:**
  - ConfigServerApplication :8888/
  - EurekaServerApplicationPeer1 :8761/
  - EurekaServerApplicationPeer2 :8762/ (highlighted with a blue selection bar)
- Not Started:**
  - ActivityServiceApplicationLazy
  - CookieServiceApplication
  - DecisionServiceApplication

The screenshot shows the Spring Boot Actuator Console tab. It displays logs related to the Eureka client initialization process.

```
.netflix.discovery.DISCOVERYCLIENT
.discovery.InstanceInfoReplicator
.netflix.discovery.DiscoveryClient
.eureka.DefaultEurekaServerContext
.eureka.cluster.PeerEurekaNodes
.d.provider.DiscoveryJerseyProvider
.d.provider.DiscoveryJerseyProvider
.d.provider.DiscoveryJerseyProvider
.eureka.cluster.PeerEurekaNodes
.e.registry.AbstractInstanceRegistry
.eureka.DefaultEurekaServerContext

Starting heartbeat executor. Renew interval is: 30
: InstanceInfoReplicator onDemand update allowed rate per mi
: Discovery Client initialized at timestamp 1620044315753 wi
: Initializing ...
: Adding new peer nodes [http://eureka-peer1:8761/eureka/]
: Using JSON encoding codec LegacyJacksonJson
: Using JSON decoding codec LegacyJacksonJson
: Using XML encoding codec XStreamXml
: Using XML decoding codec XStreamXml
: Replica node URL: http://eureka-peer1:8761/eureka/
: Finished initializing remote region registries. All known
: Initialized
```

**System Status**

Environment	N/A	2021-05-03T14:20:24 +0200
Data center	N/A	00:02
		true
	Renews threshold	3
	Renews (last min)	5

(2)

**DS Replicas**

eureka-peer2

**Instances currently registered with Eureka**

Application	AMIs	Availability Zones	Status
EUREKA-CLUSTER	n/a (2)	(2)	UP (2) - 192.168.1.107:eureka-server:8761 , 192.168.1.107:eureka-server:8762

(3)

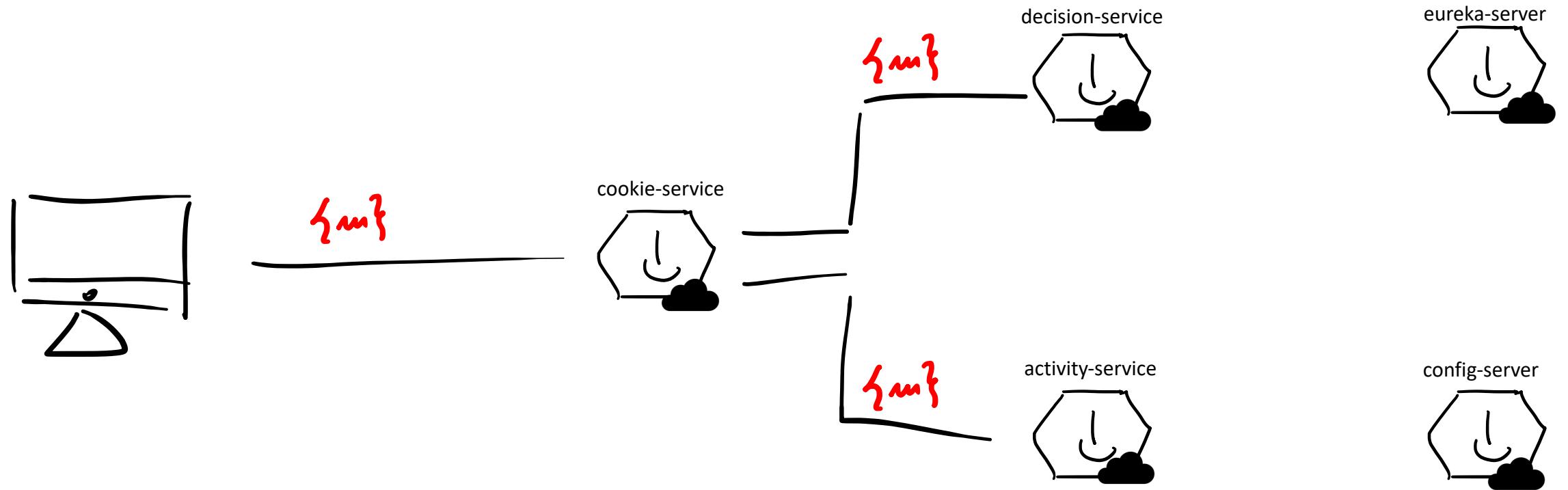
**General Info**

Name	Value
total-avail-memory	364mb
num-of-cpus	6
current-memory-usage	152mb (41%)
server-upptime	00:02
registered-replicas	<a href="http://eureka-peer2:8762/eureka/">http://eureka-peer2:8762/eureka/</a>
unavailable-replicas	
available-replicas	<a href="http://eureka-peer2:8762/eureka/">http://eureka-peer2:8762/eureka/</a> ,

**Instance Info**

Name	Value
ipAddr	192.168.1.107
status	Fortune Cookie - krok po kroku

# Message



▼ message 2 files ~/Desktop/noble/fortune-cookie/message

▼ src/main/java/com/example 1 file

J Message.java 2

pom.xml 1

## Message

- ① - wersja Javy w pom.xml (sprawdź wersję u module projektu)
- ② - klasa Message (pole service, message, konstruktory, get/set)

pom.xml (/Users/uki/Desktop/noble/fortune-cookie/message)

Side-by-side viewer | Do not ignore | Highlight words | 1 difference

ef2249db501abc966760b751661c177a7ece59a7

dd3b5709dc6fa41d7719fd6cca5c49e894f2e376

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>org.example</groupId>
    <artifactId>message</artifactId>
    <version>1.0-SNAPSHOT</version>
</project>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>org.example</groupId>
    <artifactId>message</artifactId>
    <version>1.0-SNAPSHOT</version> 1
    <properties>
        <maven.compiler.source>11</maven.compiler.source>
        <maven.compiler.target>11</maven.compiler.target>
    </properties>
</project>
```

Side-by-side viewer ▾ Do not ignore ▾ Highlight words ▾ ?

```
dd3b5709dc6fa41d7719fd6cca5c49e894f2e376
```

1 package com.example;

2

3 public class Message { *(2)*

4 private String service;

5 private String message;

6

7 public Message() {

8 }

9

10 public Message(String service, String message) {

11 this.service = service;

12 this.message = message;

13 }

14

15 public String getService() {

16 return service;

17 }

18

19 public void setService(String service) {

20 this.service = service;

21 }

22

23 public String getMessage() {

24 return message;

25 }

26

27 public void setMessage(String message) {

28 this.message = message;

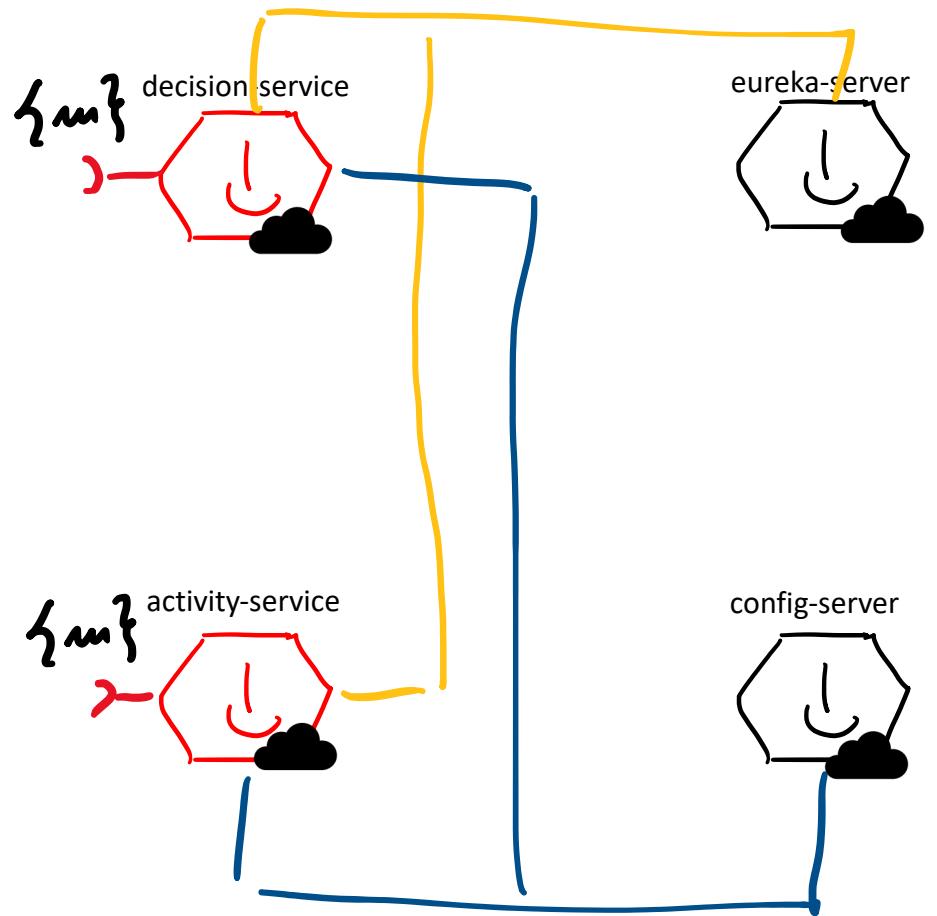
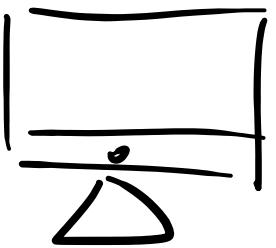
29 } *Fortune Cookie - krok po kroku*

30 }

*mvn clean install*

```
[INFO] -- maven-install-plugin:2.4:install (default-install) @ message ---
[INFO] Installing /Users/uki/Desktop/noble/fortune-cookie/message/target/message-1.0-SNAPSHOT.jar to /Users/uki/.m2/repository/org/example/message/1.0-SNAPSHOT/message-1.0-SNAPSHOT.jar
[INFO] Installing Users/uki/Desktop/noble/fortune-cookie/message/pom.xml to /Users/uki/.m2/repository/org/example/message/1.0-SNAPSHOT/message-1.0-SNAPSHOT.pom
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  1.632 s
[INFO] Finished at: 2021-05-03T14:36:56+02:00
[INFO] -----
```

# activity-service & decision-service





## Activity Service

- 1 dodanie zależności w pom.xml
- 2 bootstrap.yaml z nazwą serwisu i url do config server
- 3 ActivityController z jednym endpointem: /activity, zwraca wartość pobraną z config server

dd3b5709dc6fa41d7719fd6cca5c49e894f2e376

```

<parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.4.5</version>
    <relativePath/> 
</parent>
<groupId>com.example</groupId>
<artifactId>activity-service</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>activity-service</name>
<description>Fortune Cookie - activity service</description>
<properties>
    <java.version>11</java.version>
</properties>
<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter</artifactId>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>
</project>

```

7f7dc79dd83a48eb455ab4e322b14d357ee1bceb

```

<properties>
    <java.version>11</java.version>
    <spring-cloud.version>2020.0.2</spring-cloud.version>
</properties>
<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-starter-config</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-starter-bootstrap</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
    </dependency>
    <dependency>
        <groupId>org.example</groupId>
        <artifactId>message</artifactId>
        <version>1.0-SNAPSHOT</version>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
</dependencies>
<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>

```

**REST**  
**CONFIG (Klient)**  
**EUREKA (CLIENT)**  
**MESSAGE**

1

Side-by-side viewer ▾ Do not ignore ▾ Highlight words ▾

7f7dc79dd83a48eb455ab4e322b14d357ee1bceb

1 **2** spring:  
2 application:  
3     name: activity-service  
4 cloud:  
5 config:  
6     uri: http://config-server:8888

ACTIVITY → LOWFIL → **GIRL** **activity-service.yaml**

```
graph LR; ACTIVITY --> LOWFIL; LOWFIL --> GIRL
```

ActivityController.java (/Users/uki/Desktop/noble/fortune-cookie/activity-service/src/main/java/com/example/activityservice/controller)

Side-by-side viewer ▾ Do not ignore ▾ Highlight words ▾ ?

7f7dc79dd83a48eb455ab4e322b14d357ee1bceb

```
1 package com.example.activityservice.controller;
2
3 import com.example.Message;
4 import org.springframework.beans.factory.annotation.Value;
5 import org.springframework.web.bind.annotation.GetMapping;
6 import org.springframework.web.bind.annotation.RestController;
7
8 import java.util.Random;
9
10 @RestController
11 public class ActivityController {
12
13     @Value("${activities}")
14     private String [] activities;
15     @Value("${spring.application.name}")
16     private String serviceName;
17
18     @GetMapping("/activity")
19     public Message decision() {
20         return getMessageWithRandomActivity();
21     }
22
23
24     @
25     private Message getMessageWithRandomActivity() {
26         String msg = activities[new Random().nextInt(activities.length)];
27         return new Message(serviceName.toUpperCase(), msg);
28     }

```

Fortune Cookie - krok po kroku

A red circle with the number '3' is drawn around the '@Value("\${activities}")' annotation. A red arrow points from this circle down to the 'activities' variable declaration. To the right of the code, the text 'activity-service.yml' is written in red. On the far left, there is a large red bracket spanning multiple lines of code, and above it, a red curly brace is drawn over the opening brace of the ActivityController class definition.

# activity-service - test

## Services

The screenshot shows the Spring Boot Actuator Services interface. On the left, there's a sidebar with various icons. The main area lists running Spring Boot applications under the 'Spring Boot' section:

- ActivityServiceApplicationLazy :8010/ (highlighted in blue)
- ConfigServerApplication :8888/
- EurekaServerApplicationPeer1 :8761/
- EurekaServerApplicationPeer2 :8762/

Below this, there's a 'Not Started' section with two entries:

- CookieServiceApplication
- DecisionServiceApplication

The screenshot shows the Spring Boot Actuator Console tab displaying application startup logs for the selected service. The logs include:

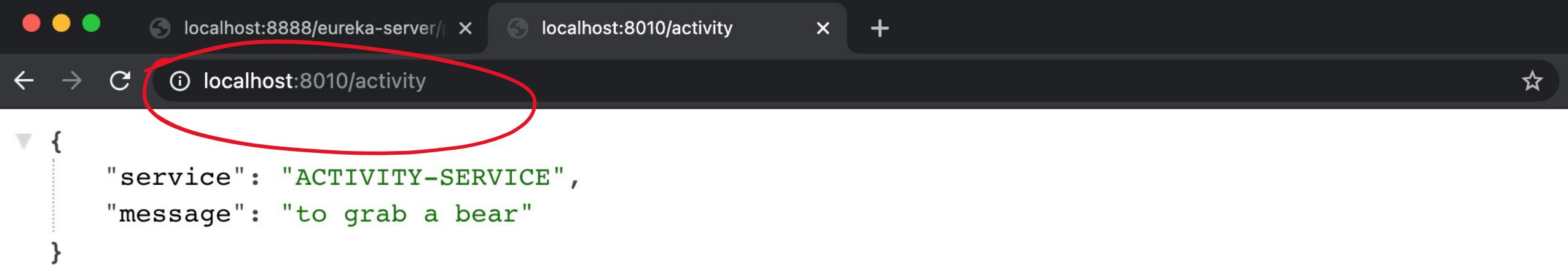
```
2023-09-12 10:42:42.123 [main] INFO  o.s.b.SpringApplication - : Fetching config from server at : http://config-server:8888
2023-09-12 10:42:42.123 [main] INFO  o.s.b.SpringApplication - : Located environment: name=activity-service, profiles=[lazy]
2023-09-12 10:42:42.123 [main] INFO  o.s.b.SpringApplication - : Located property source: [BootstrapPropertySource {name='bootstrap.properties'}]
2023-09-12 10:42:42.123 [main] INFO  o.s.b.SpringApplication - : The following profiles are active: lazy
2023-09-12 10:42:42.123 [main] INFO  o.s.b.SpringApplication - : BeanFactory id=205298a8-1e67-3dda-bc16-fa51fe5cc212
2023-09-12 10:42:42.123 [main] INFO  o.s.b.SpringApplication - : Tomcat initialized with port(s): 8010 (http)
2023-09-12 10:42:42.123 [main] INFO  o.s.b.SpringApplication - : Starting service [Tomcat]
2023-09-12 10:42:42.123 [main] INFO  o.s.b.SpringApplication - : Starting Servlet engine: [Apache Tomcat/9.0.45]
2023-09-12 10:42:42.123 [main] INFO  o.s.b.SpringApplication - : Initializing Spring embedded WebApplicationContext
2023-09-12 10:42:42.123 [main] INFO  o.s.b.SpringApplication - : Root WebApplicationContext: initialization completed in 715 ms
```

Handwritten annotations in red:

- ① URUCHOMIĘ Z PROFILEM (PROFILE ZDEFINIONOWANE W activity-service.yaml)
- ② CZY SERVIS POBIERA WEŚCIWU konfiguracji ?

## Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
ACTIVITY-SERVICE	n/a (1)	(1)	UP (1) - 192.168.1.107:activity-service:8010
EUREKA-CLUSTER	n/a (2)	(2)	UP (2) - 192.168.1.107:eureka-server:8761 , 192.168.1.107:eureka-server:8762



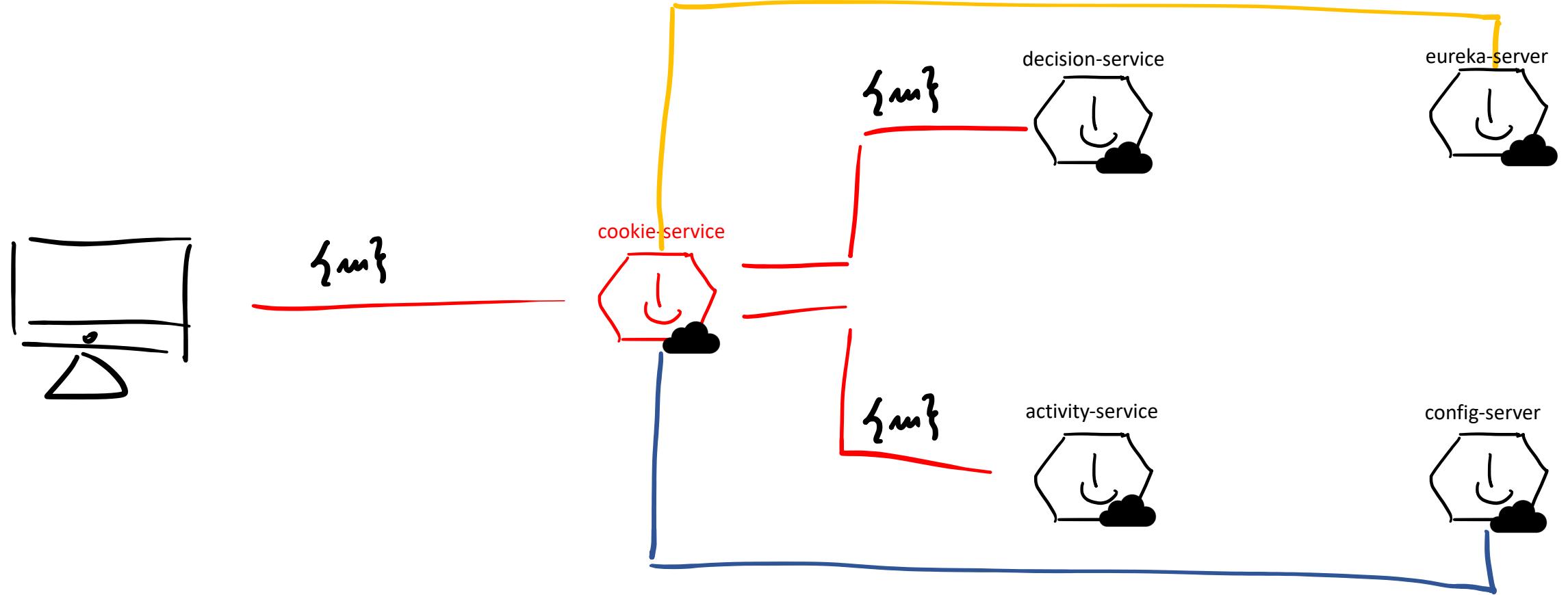
localhost:8888/eureka-server/ X    localhost:8010/activity X +

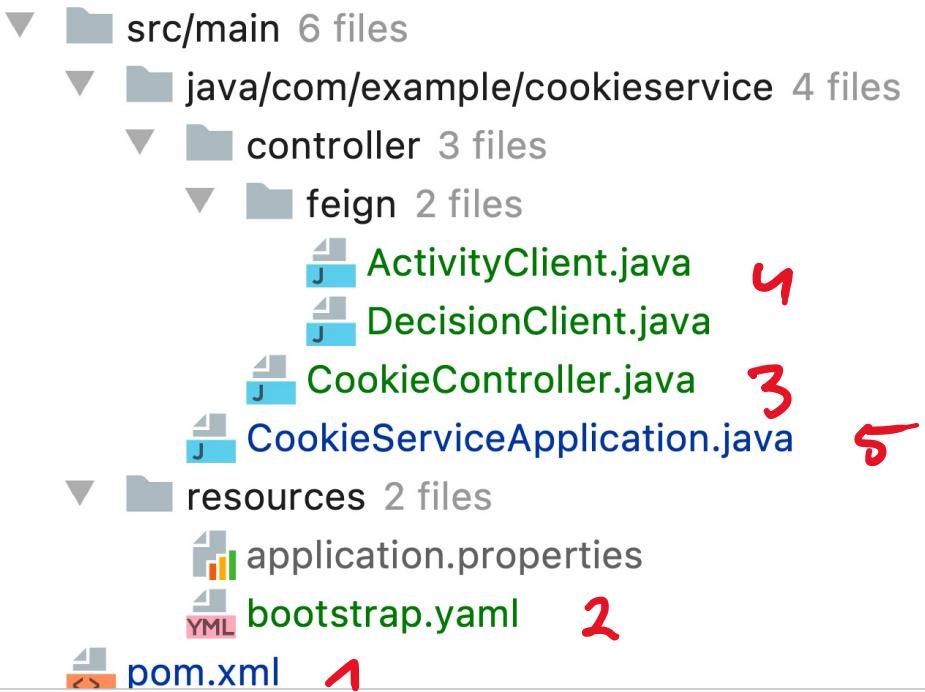
← → C ⓘ localhost:8010/activity ⭐

```
{  
  "service": "ACTIVITY-SERVICE",  
  "message": "to grab a bear"  
}
```

# decision-service - analogicznie

# cookie-service





## Cookie Service

- ① - dodanie zależności w pom.xml
- ② - bootstrap.yaml z nazwą serwisu i url do config server
- ③ - CookieController z jednym endpointem: /fortune, zwraca wartość pobraną z decision i activity service
  - kontroler korzysta z klienta Feign: utworzone interfejsy ActivityClient i DecisionClient, skonfigurowane do komunikacji z serwisami activity / decision  
(@FeignClient używa nazwy serwisu, pod jaką występuje w Eurece, jest ona zdefiniowana w application.name serwisu)
- ④ - @EnableFeignClients w klasie głównej

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.4.5</version>
    <relativePath/> 
  </parent>
  <groupId>com.example</groupId>
  <artifactId>cookie-service</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>cookie-service</name>
  <description>Fortune Cookie - cookie service</description>
  <properties>
    <java.version>11</java.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter</artifactId>
    </dependency>

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
      <scope>test</scope>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
      </plugin>
    </plugins>
  </build>
</project>

```

(1)

```

<properties>
  <java.version>11</java.version>
  <spring-cloud.version>2020.0.2</spring-cloud.version>
</properties>
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-config</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-bootstrap</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-openfeign</artifactId>
  </dependency>
  <dependency>
    <groupId>org.example</groupId>
    <artifactId>message</artifactId>
    <version>1.0-SNAPSHOT</version>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>
<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>

```



Side-by-side viewer ▾ | Do not ignore ▾ | Highlight words ▾ |

1 **2** **Spring:**

2     **application:**

3         **name:** cookie-service

4     **cloud:**

5     **config:**

6         **uri:** http://config-server:8888

7

CookieController.java (/Users/uki/Desktop/noble/fortune-cookie/cookie-service/src/main/java)

Side-by-side viewer ▾ Do not ignore ▾ Highlight words ▾

00f51da2aa6328832533ad66bd217dbda613f98c

3 @RestController  
public class CookieController {  
 private final ActivityClient activity;  
 private final DecisionClient decision; } FEIGN TWORZY IMPLEMENTACJE

private final ActivityClient activity;  
private final DecisionClient decision;

@Value("\${spring.application.name}")  
private String serviceName;

public CookieController(ActivityClient activity, DecisionClient decision) {  
 this.activity = activity;  
 this.decision = decision;  
}

@GetMapping("/fortune")  
public Message fortune() {  
 return getMessage();  
}

private Message getMessage() {  
 return getFortune();  
}

private Message getFortune() {  
 return new Message(serviceName.toUpperCase(),  
 decision.getMsg().getMessage() + " " +  
 activity.getMsg().getMessage());  
}

Fortune Cookie - krok po kroku

Side-by-side viewer ▾   Do not ignore ▾   Highlight words ▾   

00f51da2aa6328832533ad66bd217dbda613f98c

```
1 package com.example.cookieservice.controller.feign;
2
3 import com.example.Message;
4 import org.springframework.cloud.openfeign.FeignClient;
5 import org.springframework.web.bind.annotation.GetMapping;
6
7 @FeignClient(name="ACTIVITY-SERVICE")    ← EUREKA DOŚIARCZY INFO,
8 public interface ActivityClient {          ← GDZIE SERWIS
9     @GetMapping("/activity")
10    Message getMsg();      ← FEIGN WYGENERUJE IMPLEMENTACJĘ
11 }
```

CookieServiceApplication.java (/Users/uki/Desktop/noble/fortune-cookie/cookie-service/src/main/java/com/example/cookieservice)

Side-by-side viewer ▾ Do not ignore ▾ Highlight words ▾

2 differences

🔒 f08b1335204674d1c485f691e4a8cafb5c1d7be5

🔒 00f51da2aa6328832533ad66bd217dbda613f98c

```
package com.example.cookieservice;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
@SpringBootApplication
public class CookieServiceApplication {

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

1 1 package com.example.cookieservice;

2 2

3 3 import org.springframework.boot.SpringApplication;

4 4 import org.springframework.boot.autoconfigure.SpringBootApplication;

5 5 import org.springframework.cloud.openfeign.EnableFeignClients;

6 6

7 7 @EnableFeignClients

8 8 @SpringBootApplication

9 9 public class CookieServiceApplication {

10 10

11 11

12 12

13 13

14 14

15 }



## cookie-service - test:

- czy serwis rejestruje się w Eurece
- czy działa localhost:8080/fortune