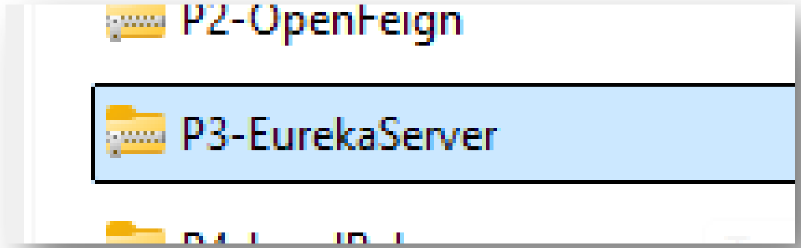
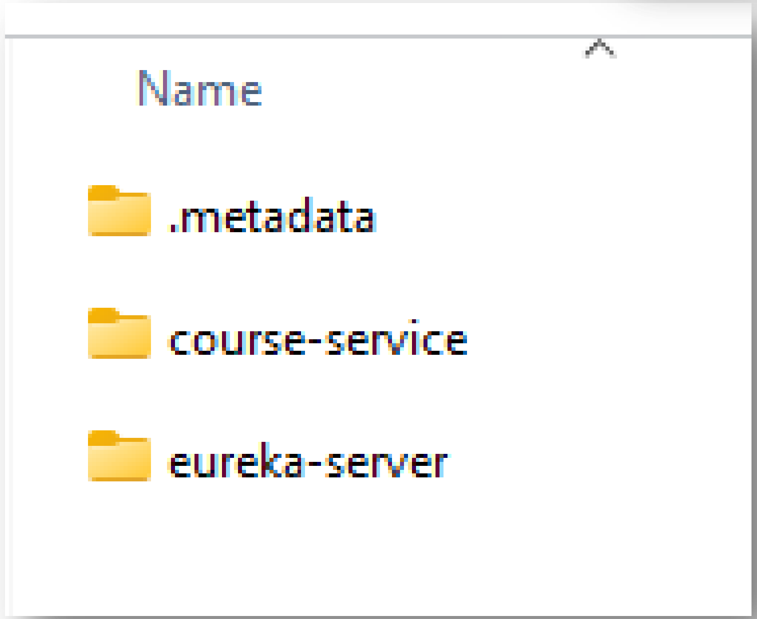
Config Server

- We need eureka server along with a service
- We copy the project of EurekaServer Demo
- In that we can have only one service



P2-OpenFeign

P3-EurekaServer

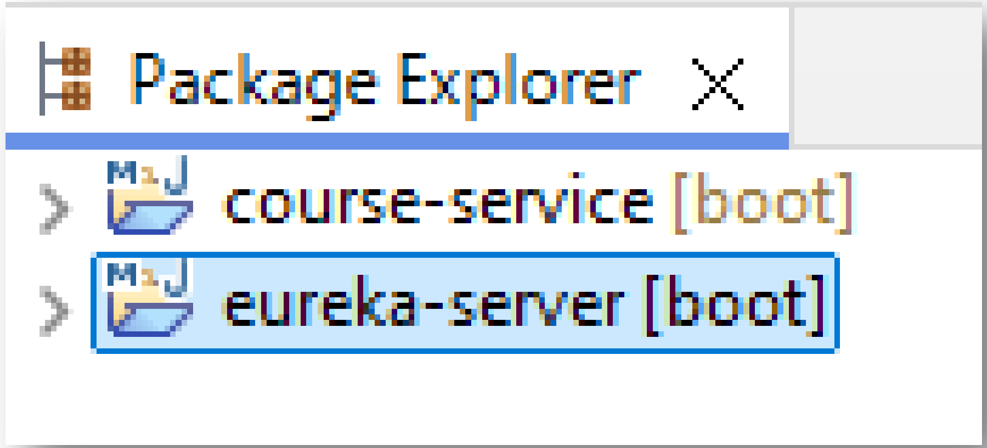


Name

.metadata

course-service

eureka-server

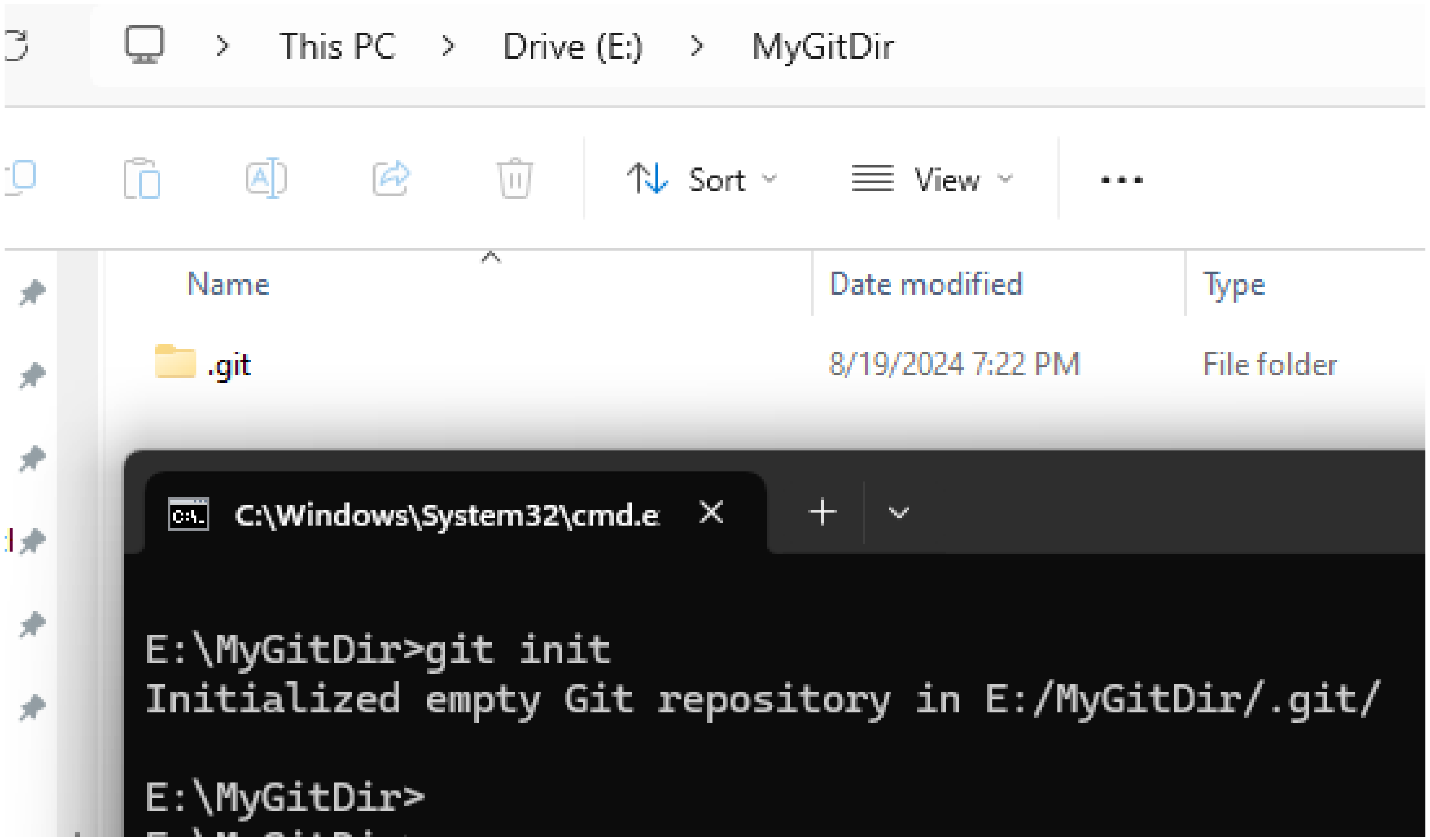


Package Explorer

course-service [boot]

eureka-server [boot]

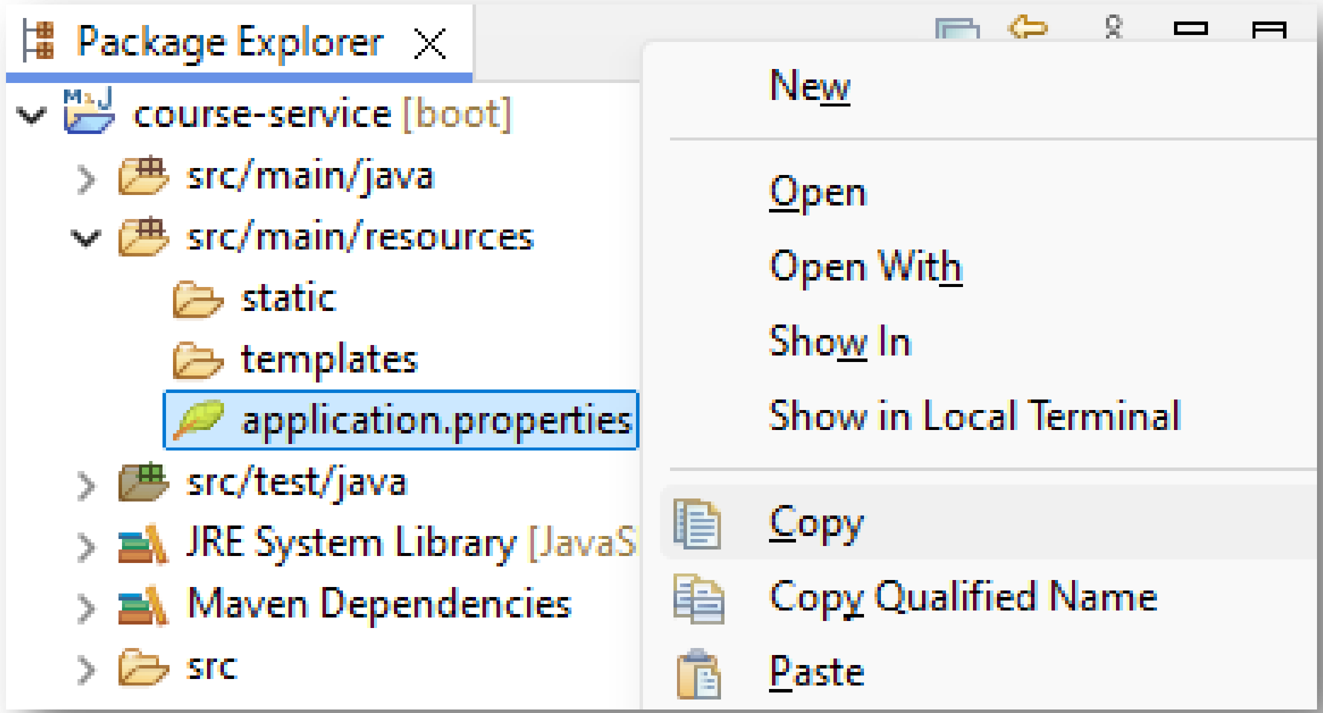
Create a folder, open in cmd & git init

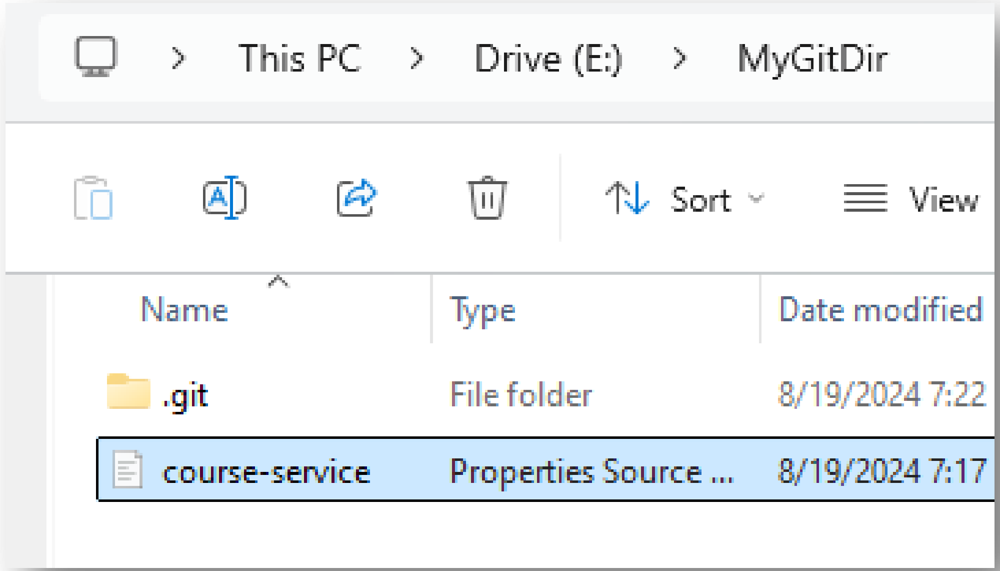


Name	Date modified	Type
.git	8/19/2024 7:22 PM	File folder

```
E:\MyGitDir>git init
Initialized empty Git repository in E:/MyGitDir/.git/

E:\MyGitDir>
```

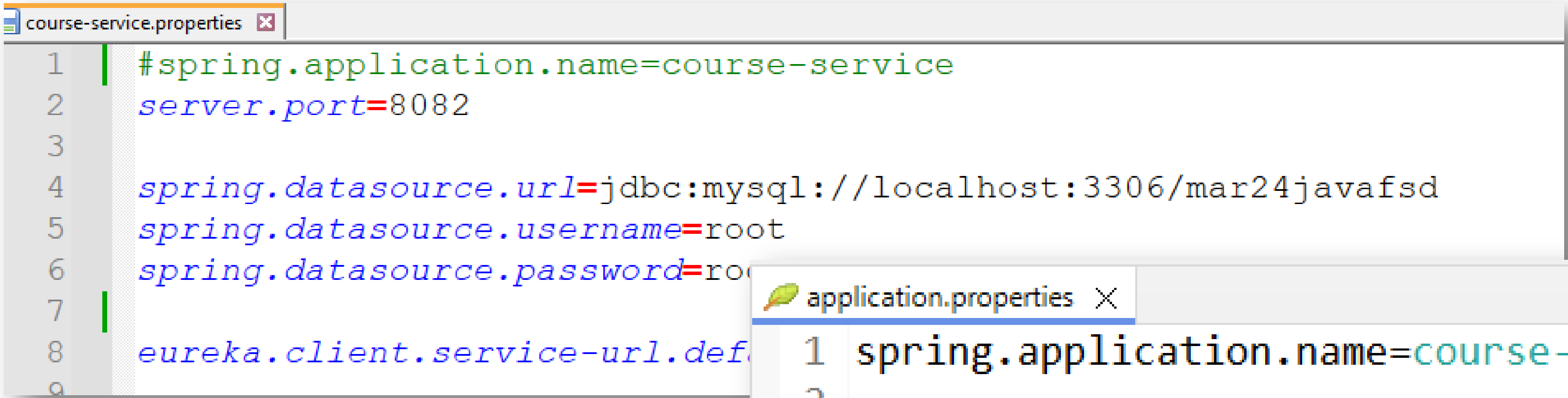
Copy & paste course-service properties file in git folder as course-service.properties





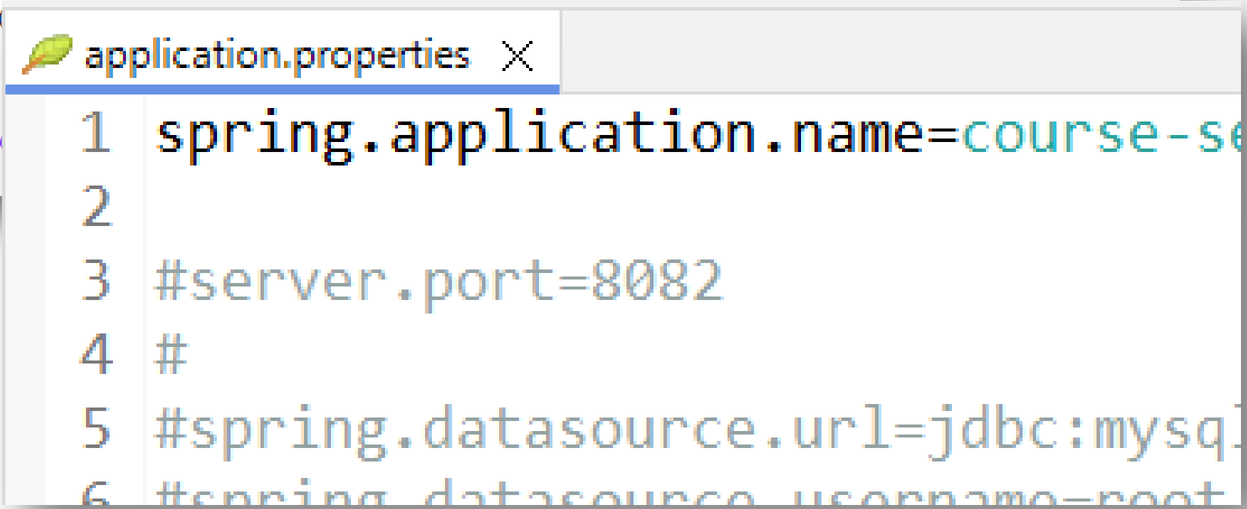
Name	Type	Date modified
.git	File folder	8/19/2024 7:22
course-service	Properties Source ...	8/19/2024 7:17

- Now this file is in default profile, Later dev, uat, prod profiles
- We have to delete the application name in properties for in git folder, Also delete all the properties except application name in course-service application properties



course-service.properties

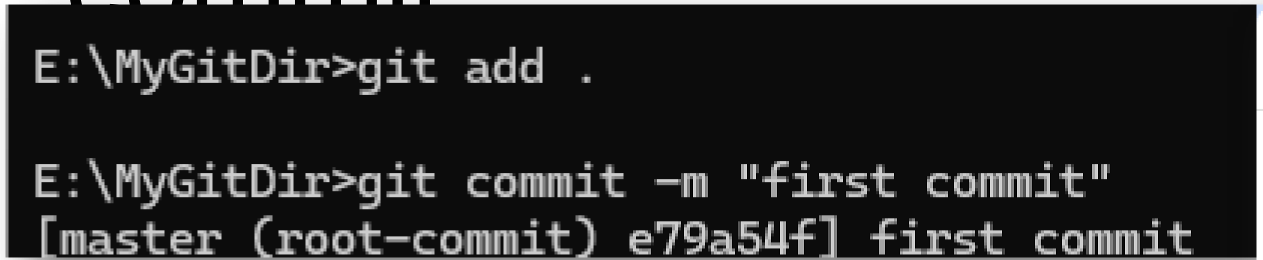
```
#spring.application.name=course-service
server.port=8082

spring.datasource.url=jdbc:mysql://localhost:3306/mar24javafsd
spring.datasource.username=root
spring.datasource.password=ro
 
eureka.client.service-url.def
```



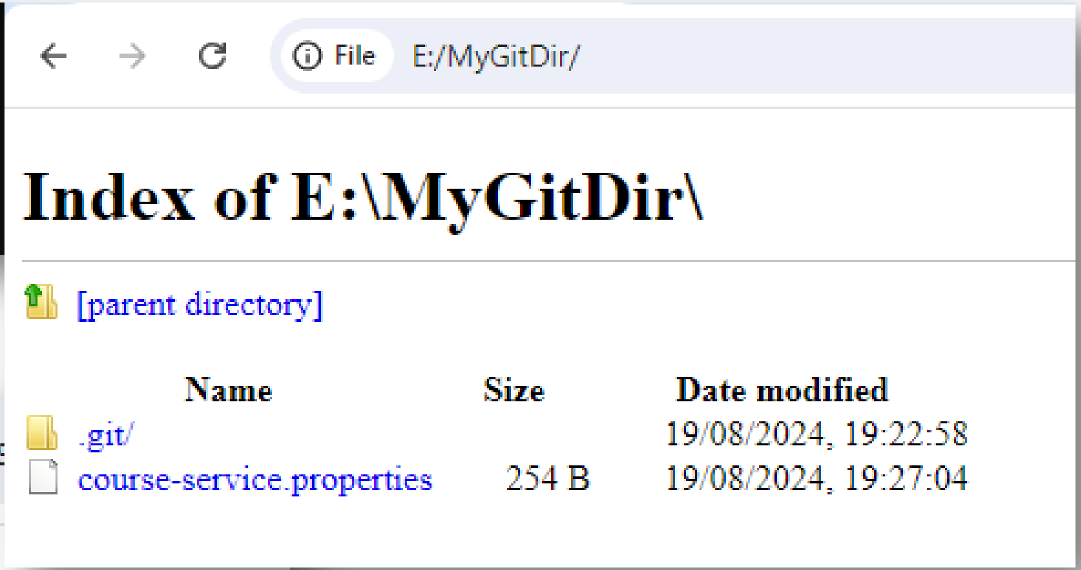
application.properties

```
spring.application.name=course-se

#server.port=8082
#
#spring.datasource.url=jdbc:mysql
#spring.datasource.username=root
```

Check the git url in browser after git commit



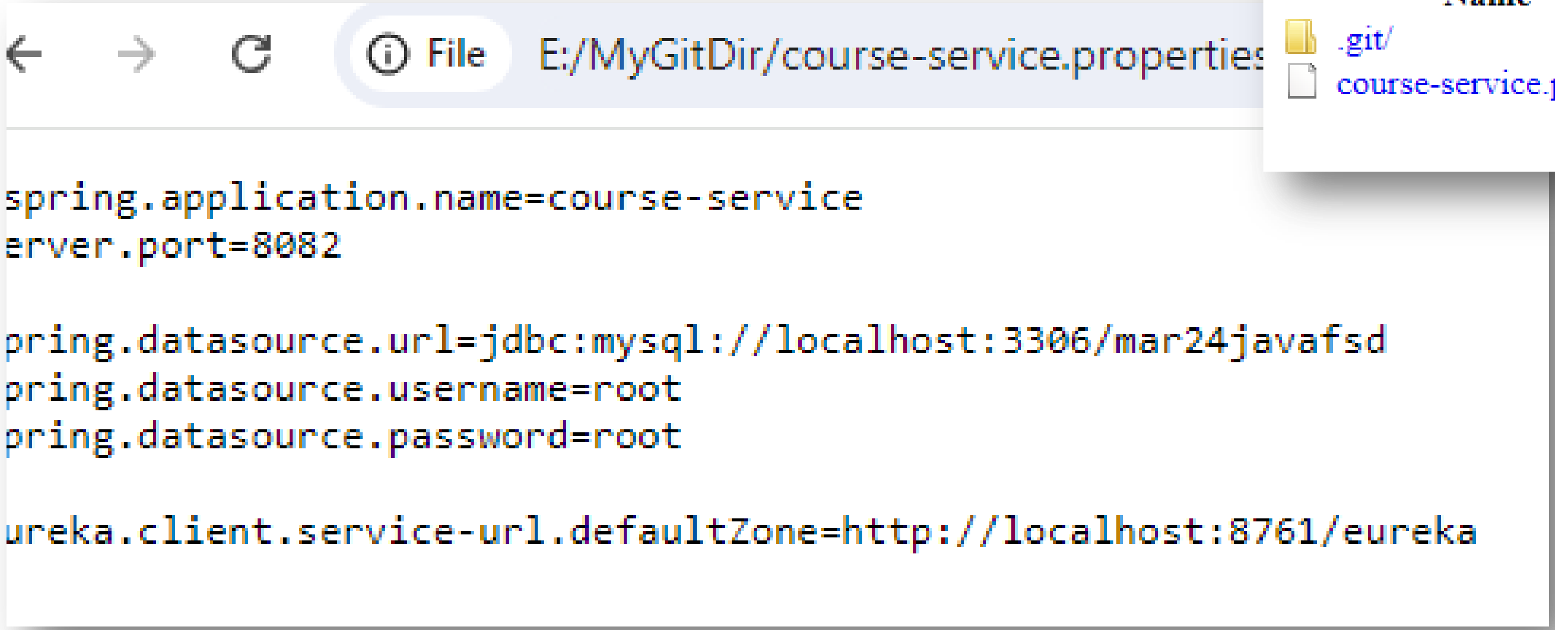
```
E:\MyGitDir>git add .

E:\MyGitDir>git commit -m "first commit"
[master (root-commit) e79a54f] first commit
```



Index of E:\MyGitDir\

[parent directory]

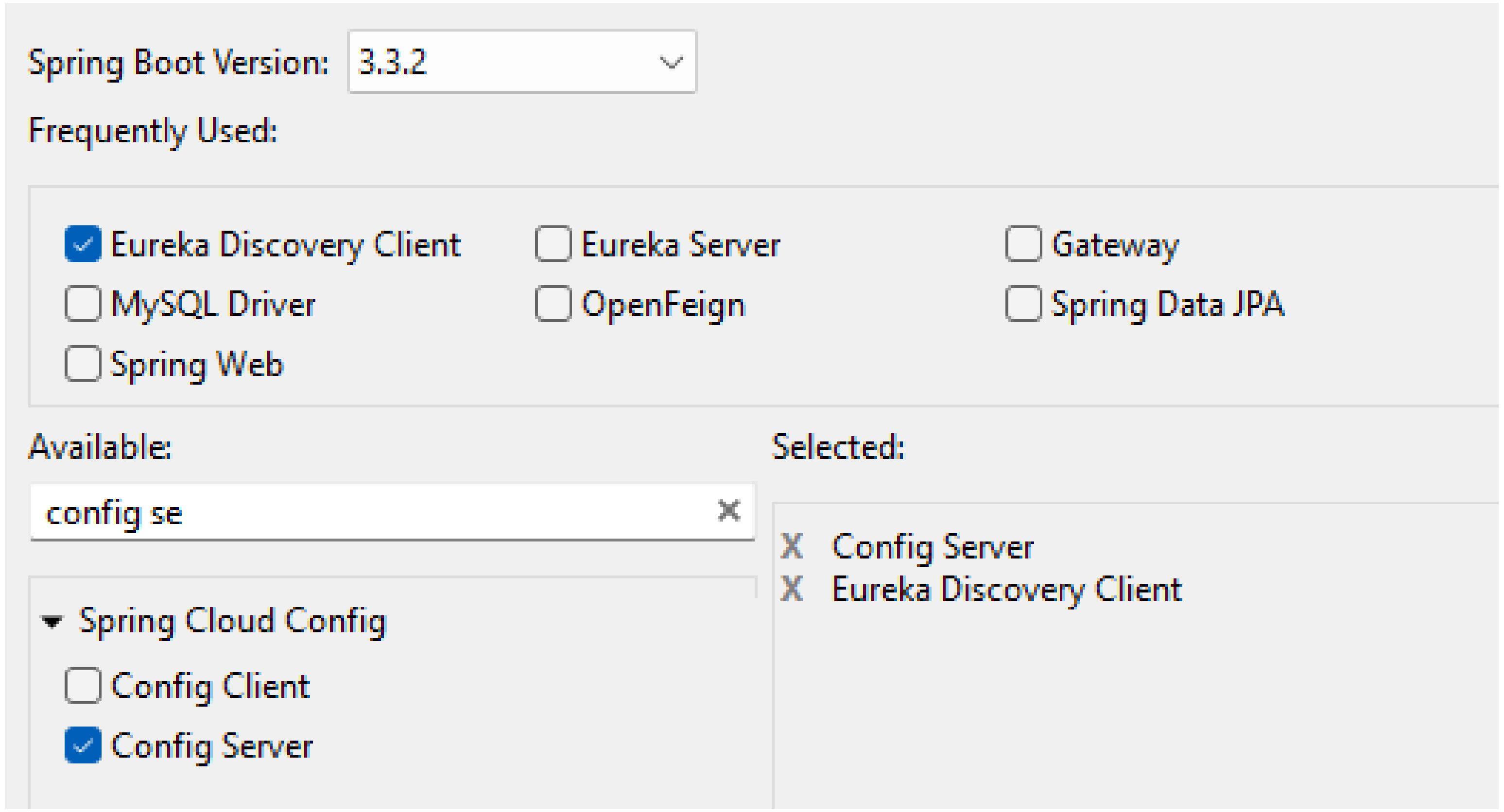
Name	Size	Date modified
.git/		19/08/2024, 19:22:58
course-service.properties	254 B	19/08/2024, 19:27:04



```
spring.application.name=course-service
erver.port=8082

pring.datasource.url=jdbc:mysql://localhost:3306/mar24javafsd
pring.datasource.username=root
pring.datasource.password=root

ureka.client.service-url.defaultZone=http://localhost:8761/eureka
```

Create a new project with 2 dependencies

Config Server & Eureka Discovery Client



Spring Boot Version: 3.3.2

Frequently Used:

- [✓] Eureka Discovery Client
- [ ] Eureka Server
- [ ] Gateway
- [ ] MySQL Driver
- [ ] OpenFeign
- [ ] Spring Data JPA
- [ ] Spring Web

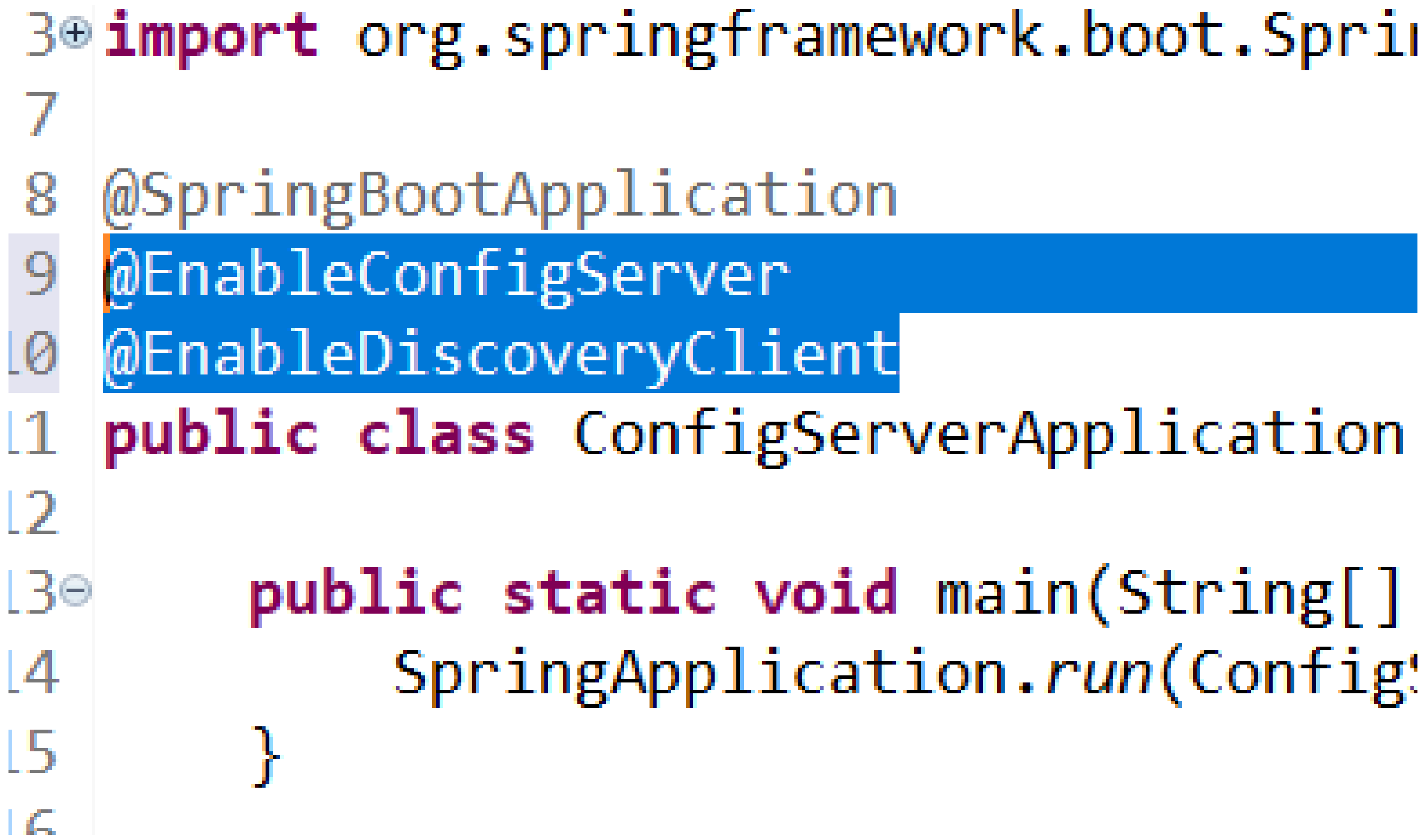
Available: config se

Spring Cloud Config
- [ ] Config Client
- [✓] Config Server

Selected:

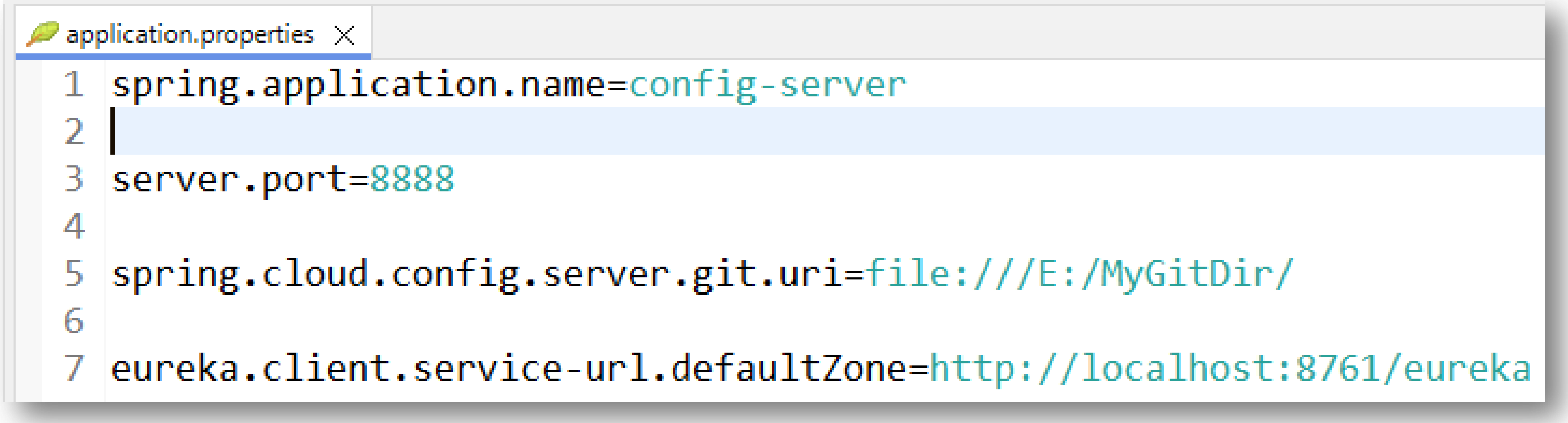
X Config Server
X Eureka Discovery Client

Enable Config Server & Eureka Discovery Client in the main class



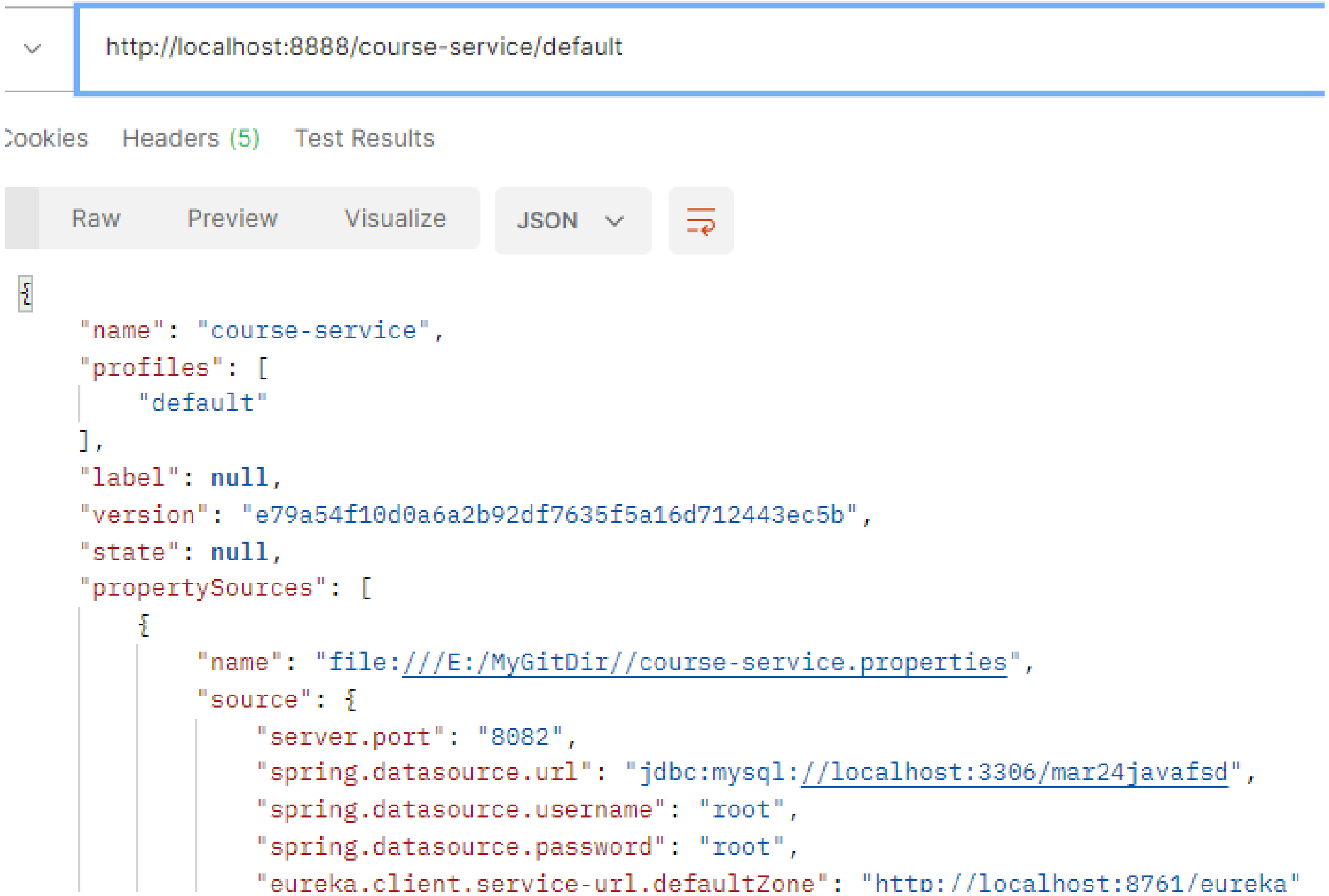
```
import org.springframework.boot.Spri

@SpringBootApplication
@EnableConfigServer
@EnableDiscoveryClient
public class ConfigServerApplication

    public static void main(String[]
        SpringApplication.run(Config
    }
```

In properties file include port, name, git url & eureka client


application.properties

```
spring.application.name=config-server

server.port=8888

spring.cloud.config.server.git.uri=file:///E:/MyGitDir/

eureka.client.service-url.defaultZone=http://localhost:8761/eureka
```

Run Eureka Server & Config Server

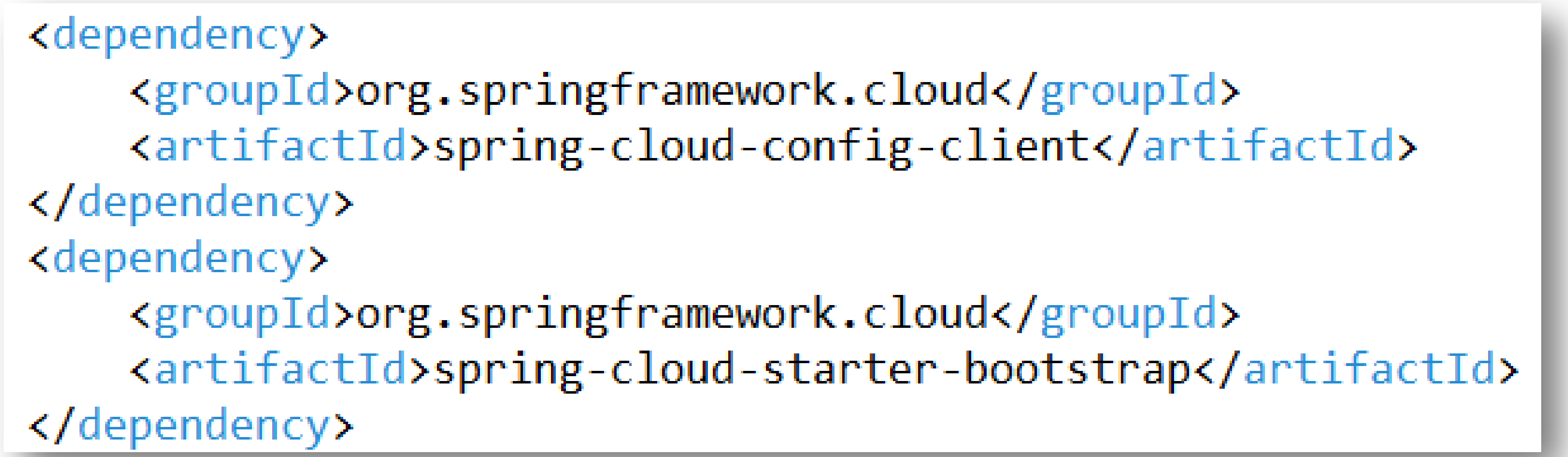

http://localhost:8888/course-service/default

Cookies Headers (5) Test Results

Raw Preview Visualize JSON

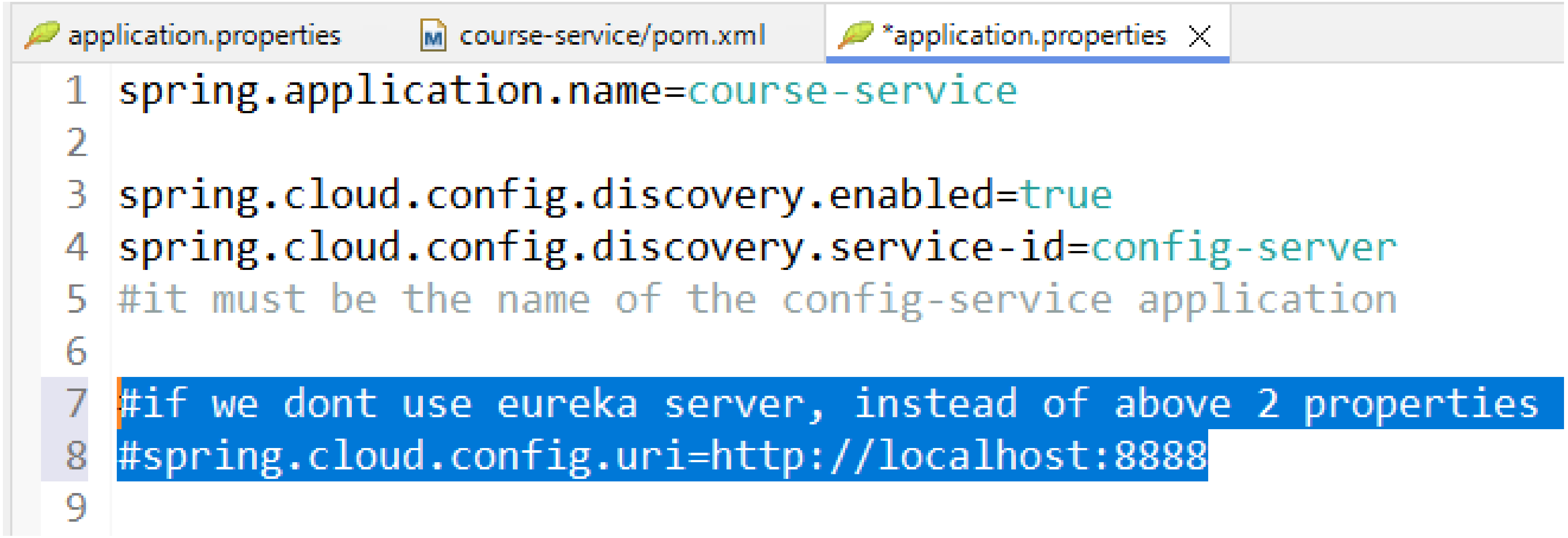
```
{
    "name": "course-service",
    "profiles": [
        "default"
    ],
    "label": null,
    "version": "e79a54f10d0a6a2b92df7635f5a16d712443ec5b",
    "state": null,
    "propertySources": [
        {
            "name": "file:///E:/MyGitDir//course-service.properties",
            "source": {
                "server.port": "8082",
                "spring.datasource.url": "jdbc:mysql://localhost:3306/mar24javafsd",
                "spring.datasource.username": "root",
                "spring.datasource.password": "root",
                "eureka.client.service-url.defaultZone": "http://localhost:8761/eureka"
```

We need to add following 2 dependencies in all our microservices to read config properties from config server



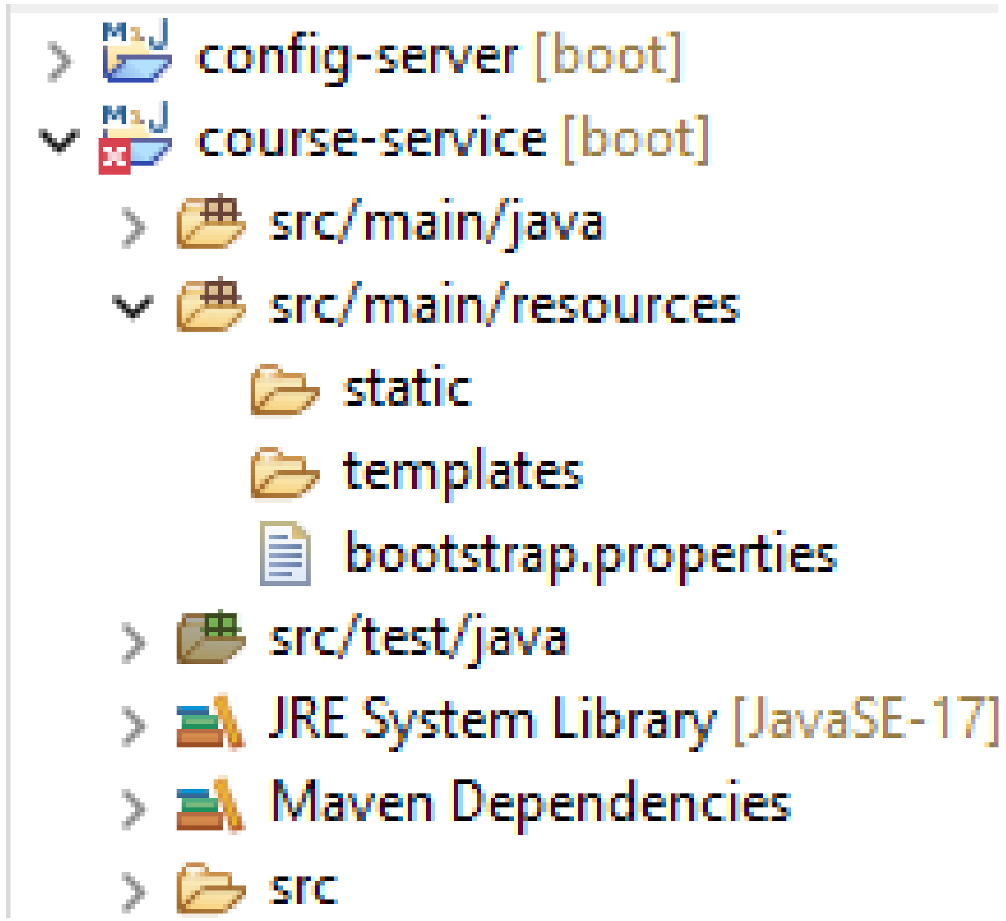
```
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-config-client</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-bootstrap</artifactId>
</dependency>
```

Next we need to 2 properties includes the config server application name



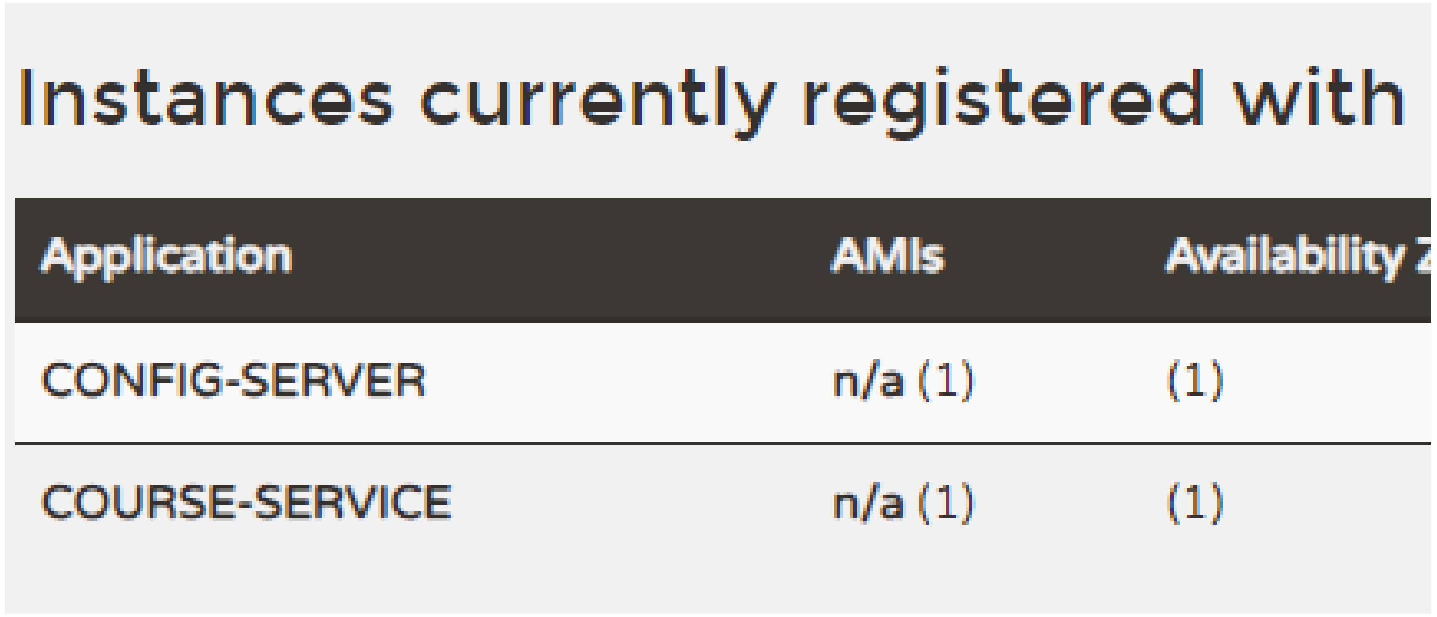
```
spring.application.name=course-service

spring.cloud.config.discovery.enabled=true
spring.cloud.config.discovery.service-id=config-server
#it must be the name of the config-service application

#if we dont use eureka server, instead of above 2 properties
#spring.cloud.config.uri=http://localhost:8888

```

Finally rename properties file as bootstrap.properties



```
> config-server [boot]
v course-service [boot]
    > src/main/java
    v src/main/resources
        static
        templates
        bootstrap.properties
    > src/test/java
    > JRE System Library [JavaSE-17]
    > Maven Dependencies
    > src
```

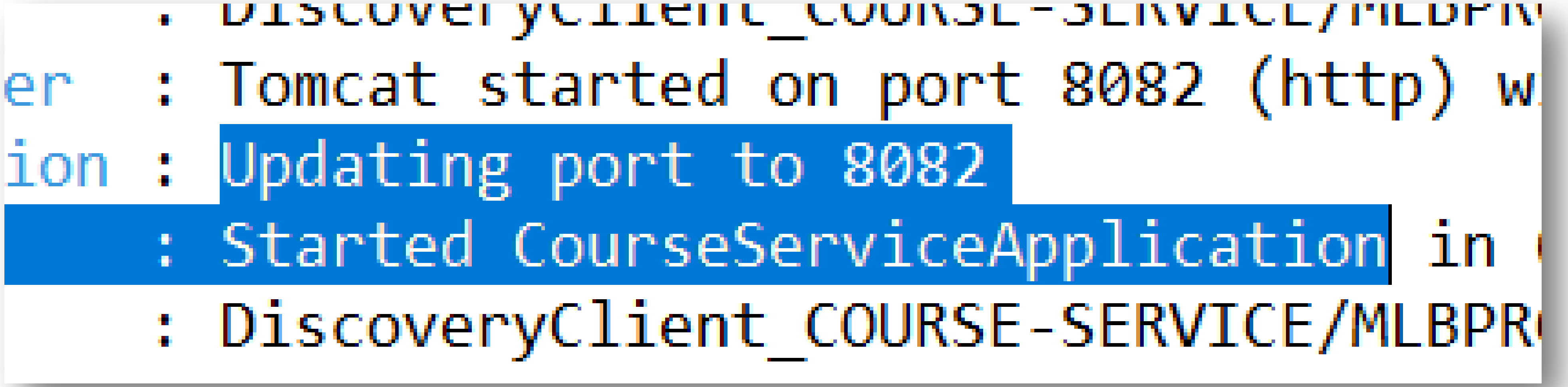
Lets start eureka, config and course services



Instances currently registered with

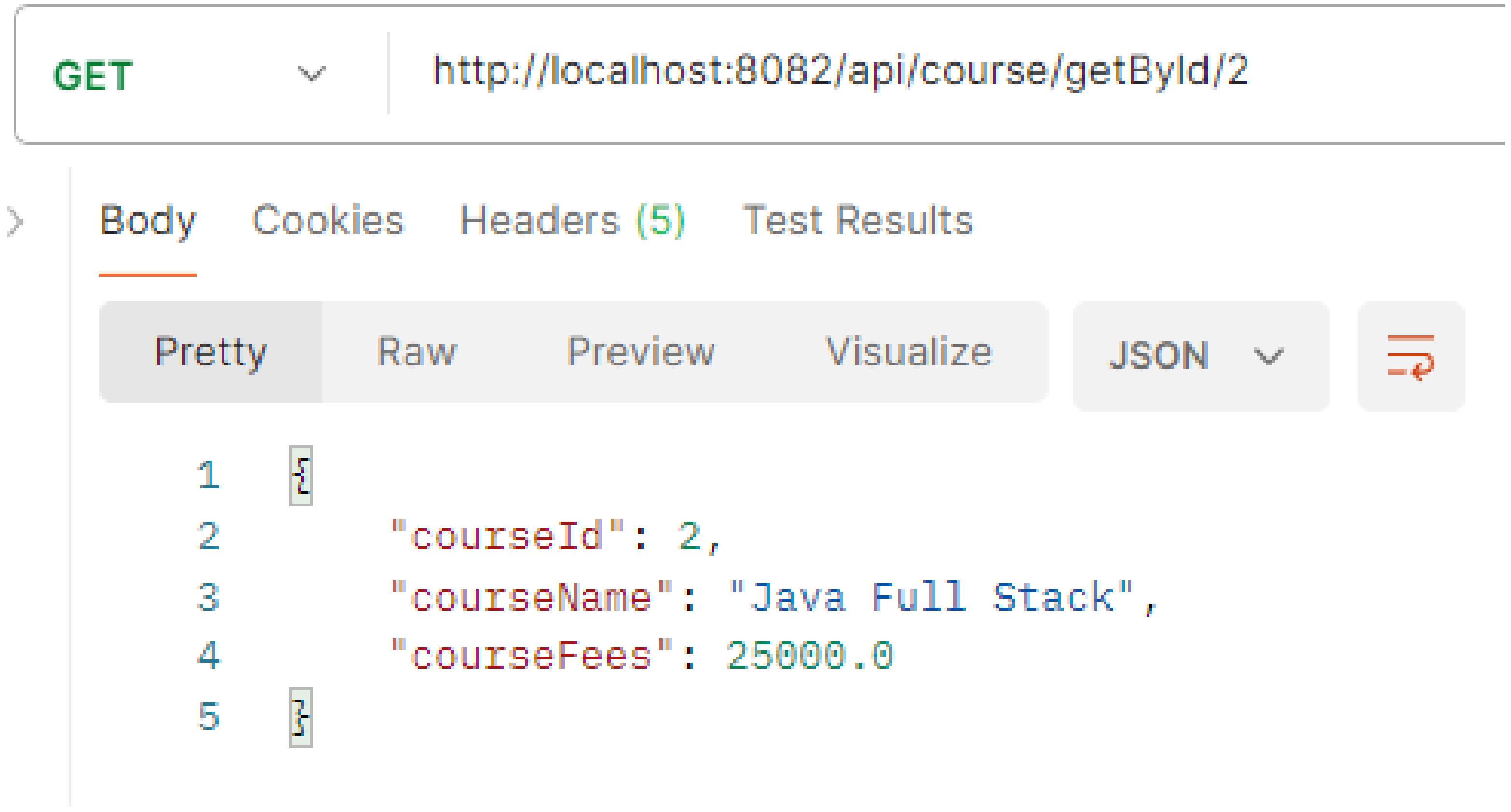
Application	AMIs	Availability Z
CONFIG-SERVER	n/a (1)	(1)
COURSE-SERVICE	n/a (1)	(1)

In Course Service we haven't give port no
it is fetching from git through config server



```
    : DiscoveryClient_COURSE-SERVICE/MLBPR
er  : Tomcat started on port 8082 (http) w
ion : Updating port to 8082
    : Started CourseServiceApplication in
    : DiscoveryClient_COURSE-SERVICE/MLBPR
```

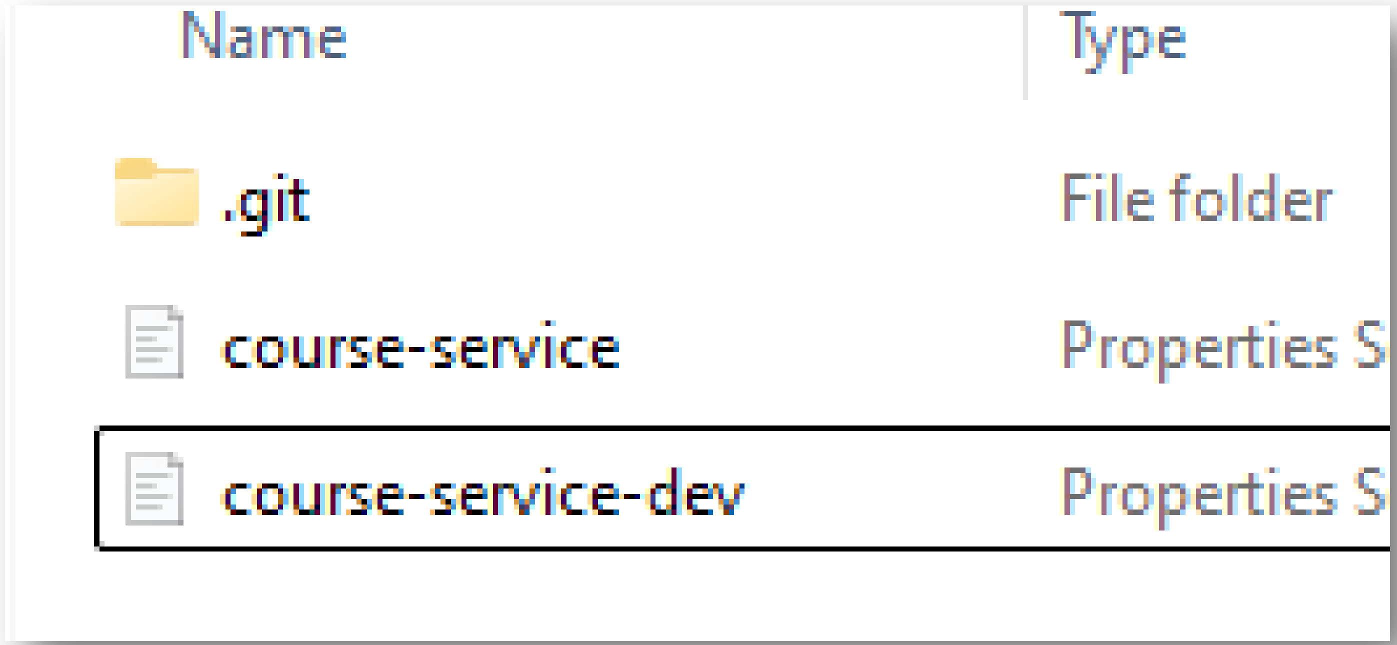
It is only possible if application read db configuration from config server



http://localhost:8082/api/course/getById/2

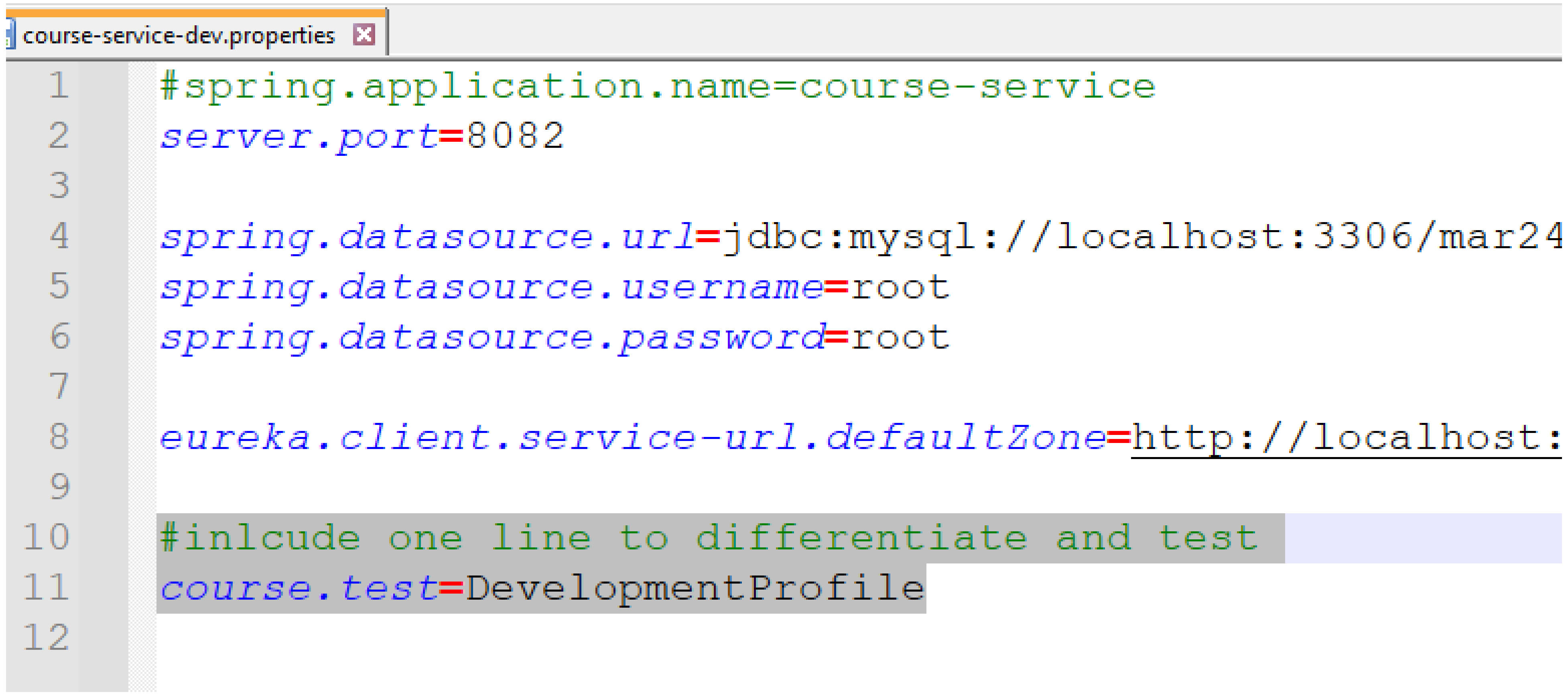
```
{
    "courseId": 2,
    "courseName": "Java Full Stack",
    "courseFees": 25000.0
}
```

Profile specific properties
In Local Repo copy the properties and add your profile



Name	Type
.git	File folder
course-service	Properties S
course-service-dev	Properties S

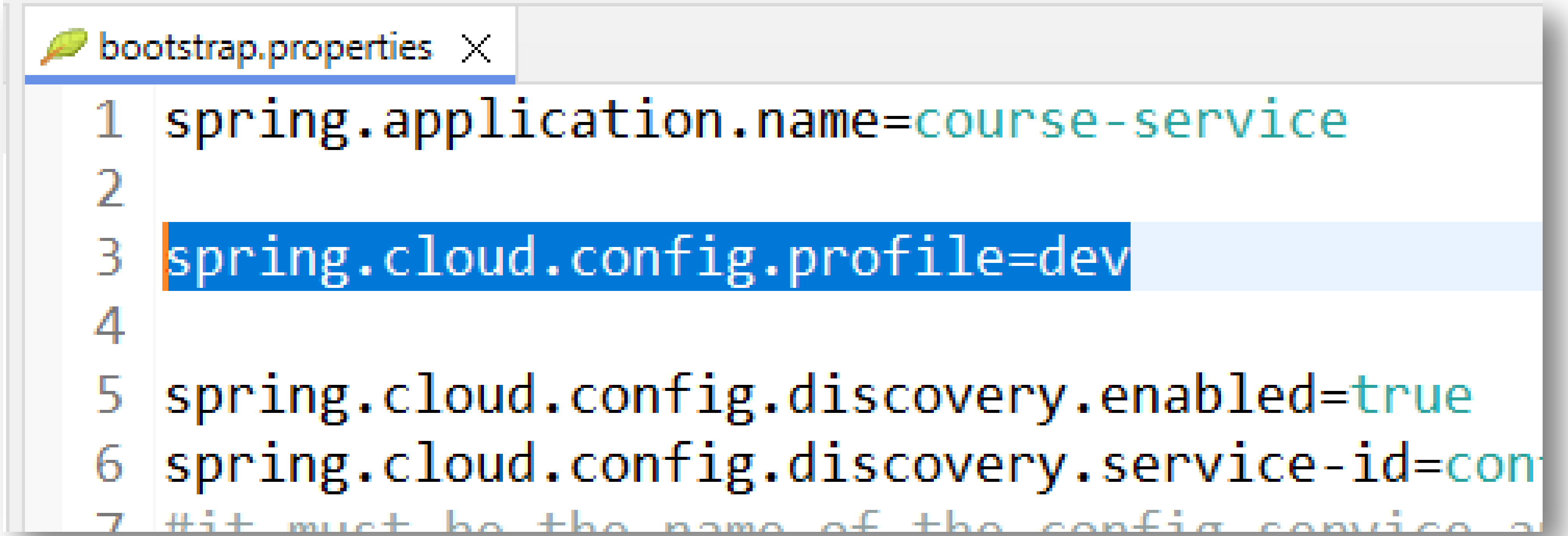
Just to differentiate property files



course-service-dev.properties

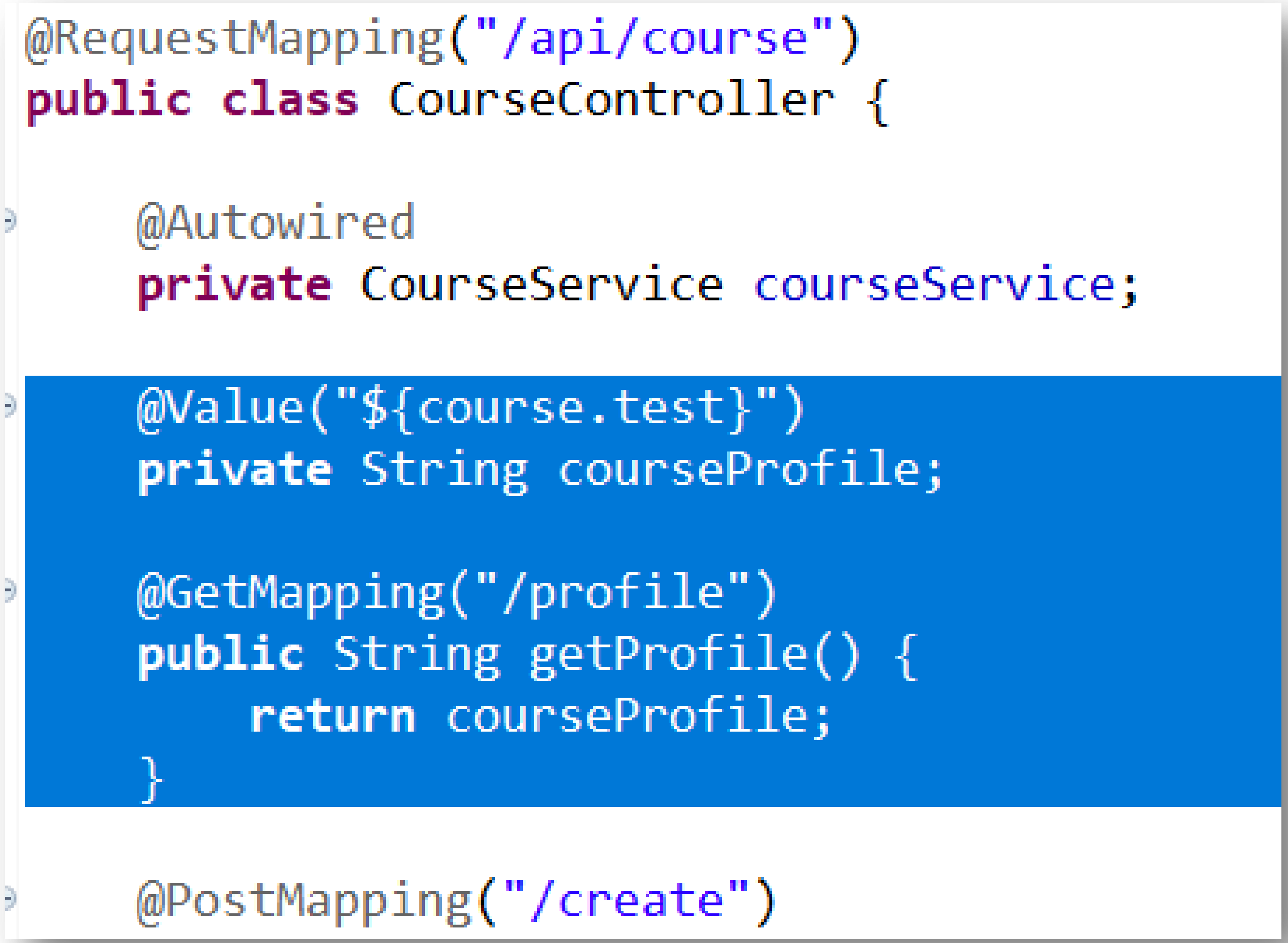
```
#spring.application.name=course-service
server.port=8082

spring.datasource.url=jdbc:mysql://localhost:3306/mar24
spring.datasource.username=root
spring.datasource.password=root

eureka.client.service-url.defaultZone=http://localhost:

#inlcude one line to differentiate and test
course.test=DevelopmentProfile

```

Include the properties filename profile part in bootstrap.properties


bootstrap.properties

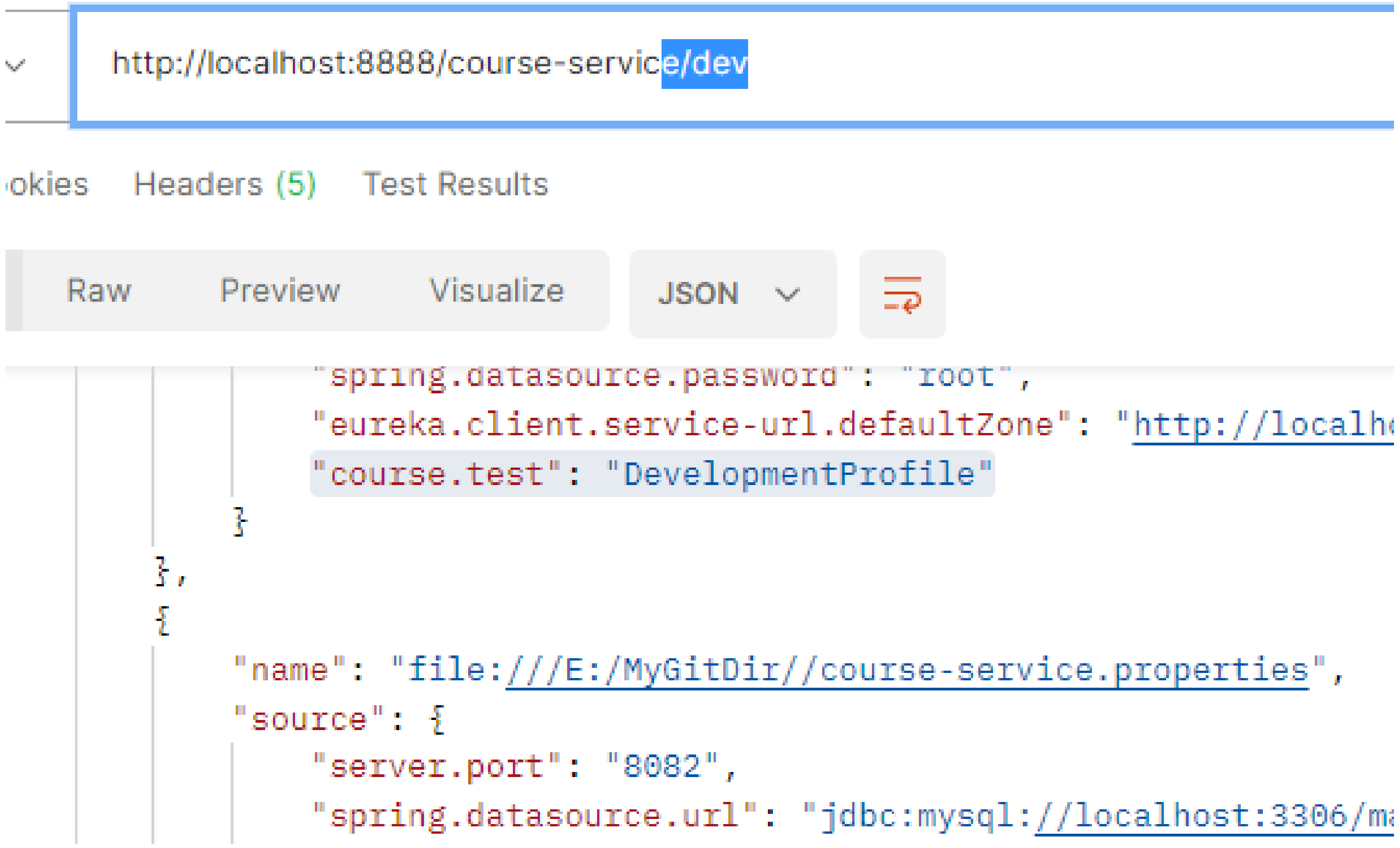
```
spring.application.name=course-service

spring.cloud.config.profile=dev

spring.cloud.config.discovery.enabled=true
spring.cloud.config.discovery.service-id=con
#it must be the name of the config service a
```

Add an end point in controller to test it



```
@RequestMapping("/api/course")
public class CourseController {

    @Autowired
    private CourseService courseService;

    @Value("${course.test}")
    private String courseProfile;

    @GetMapping("/profile")
    public String getProfile() {
        return courseProfile;
    }

    @PostMapping("/create")
```

Test the dev profile



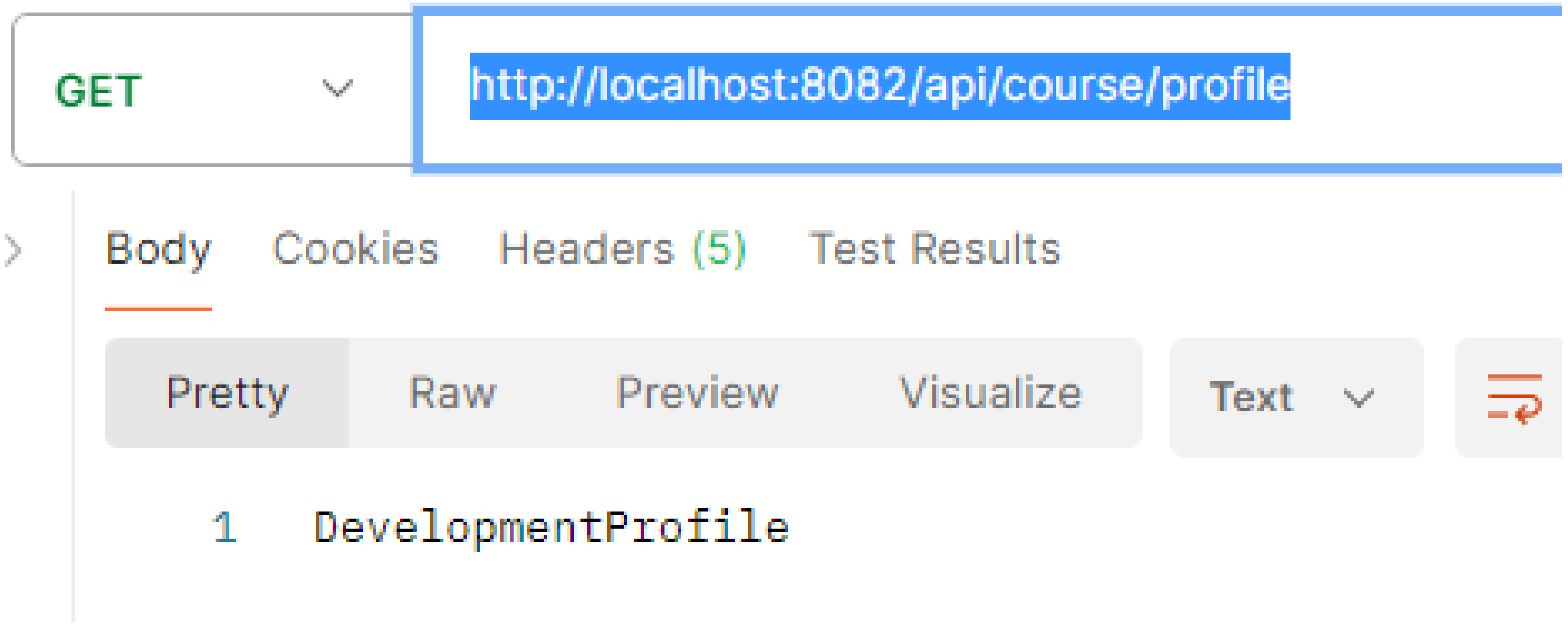
http://localhost:8888/course-service/dev

okies Headers (5) Test Results

Raw Preview Visualize JSON

```
            "spring.datasource.password": "root",
            "eureka.client.service-url.defaultZone": "http://localh
            "course.test": "DevelopmentProfile"
        }
    },
    {
        "name": "file:///E:/MyGitDir//course-service.properties",
        "source": {
            "server.port": "8082",
            "spring.datasource.url": "jdbc:mysql://localhost:3306/ma
```

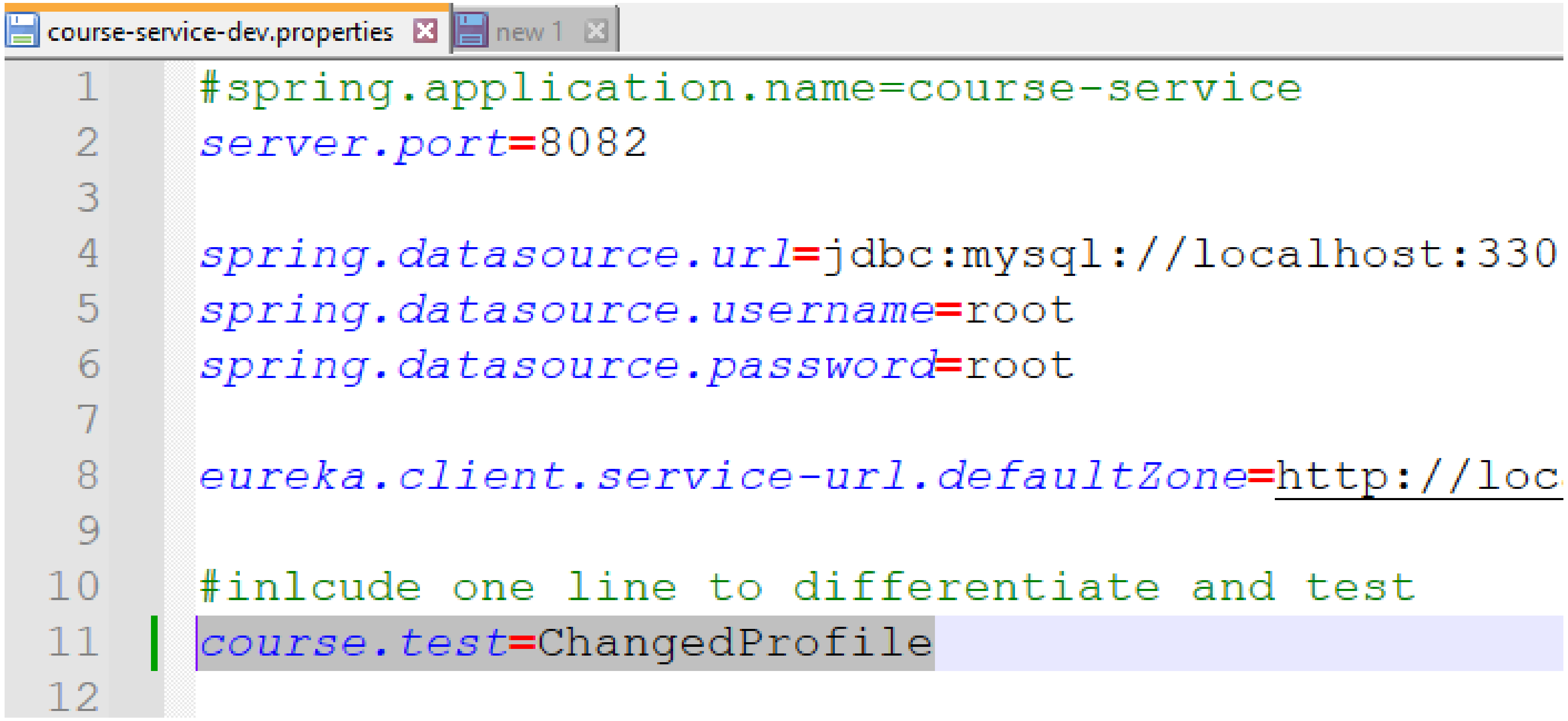
Test it with end point



http://localhost:8082/api/course/profile

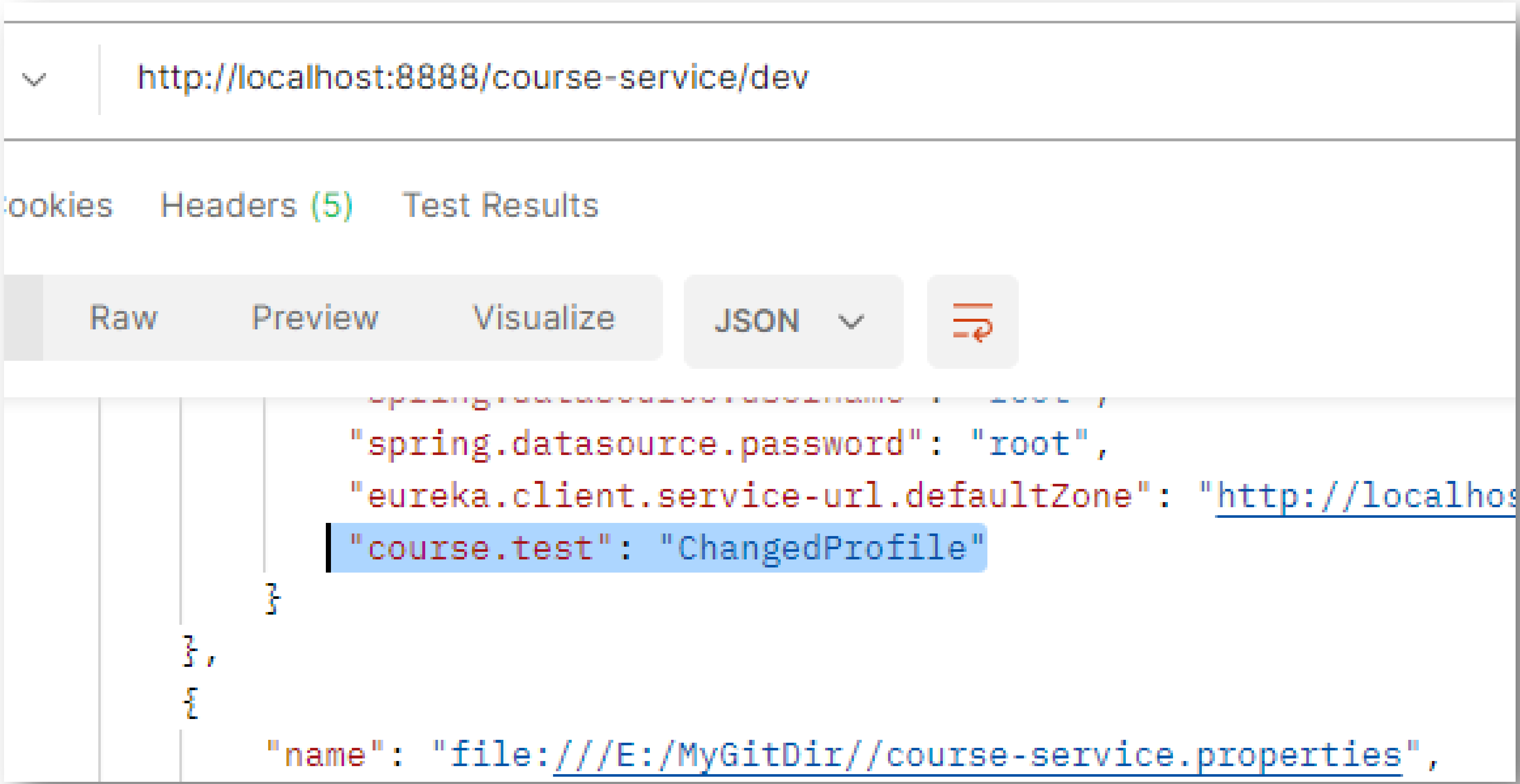
```
DevelopmentProfile
```

Now lets change the git details



```
#spring.application.name=course-service
server.port=8082

spring.datasource.url=jdbc:mysql://localhost:330
spring.datasource.username=root
spring.datasource.password=root

eureka.client.service-url.defaultZone=http://loc

#inlcude one line to differentiate and test
course.test=ChangedProfile

```

Config server fetch the updated value

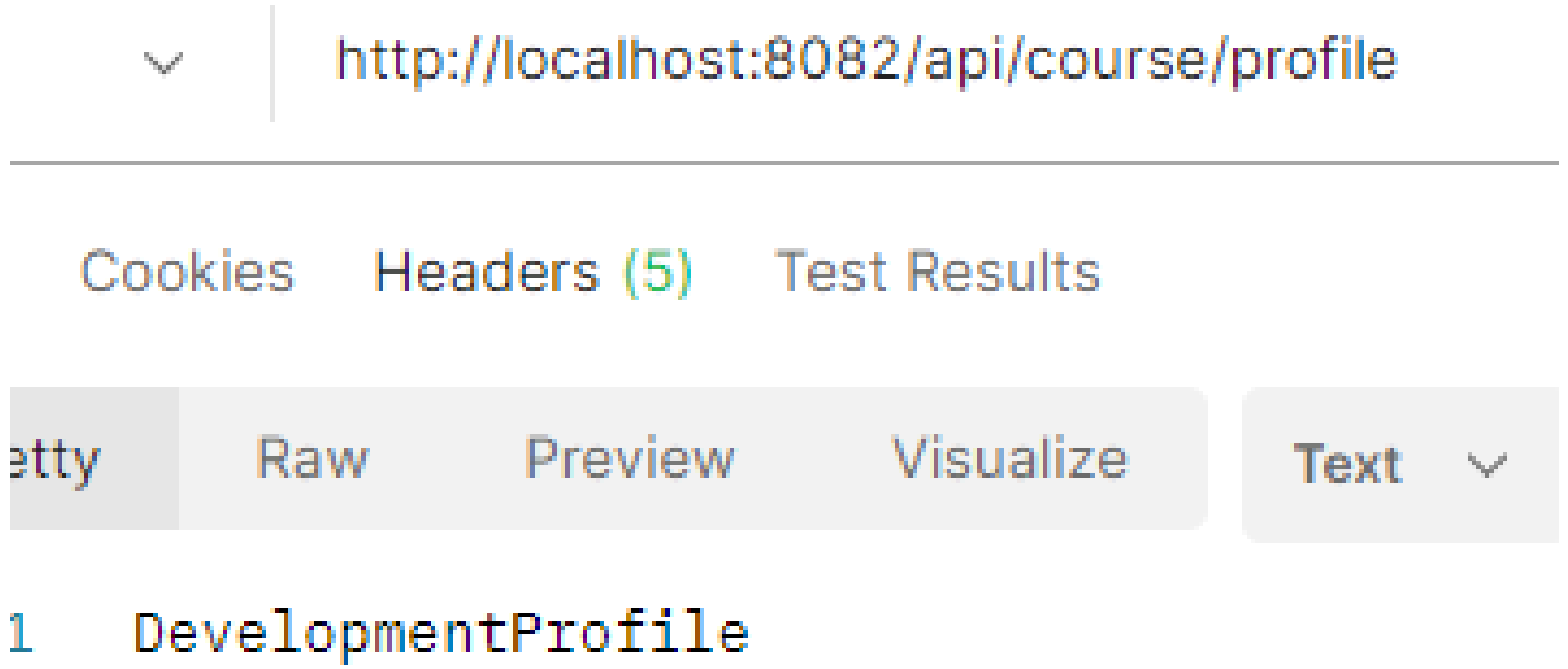


http://localhost:8888/course-service/dev

```
        "spring.datasource.password": "root",
        "eureka.client.service-url.defaultZone": "http://localhos
        "course.test": "ChangedProfile"
    }
},
{
  "name": "file:///E:/MyGitDir//course-service.properties",
```

But not the microservice

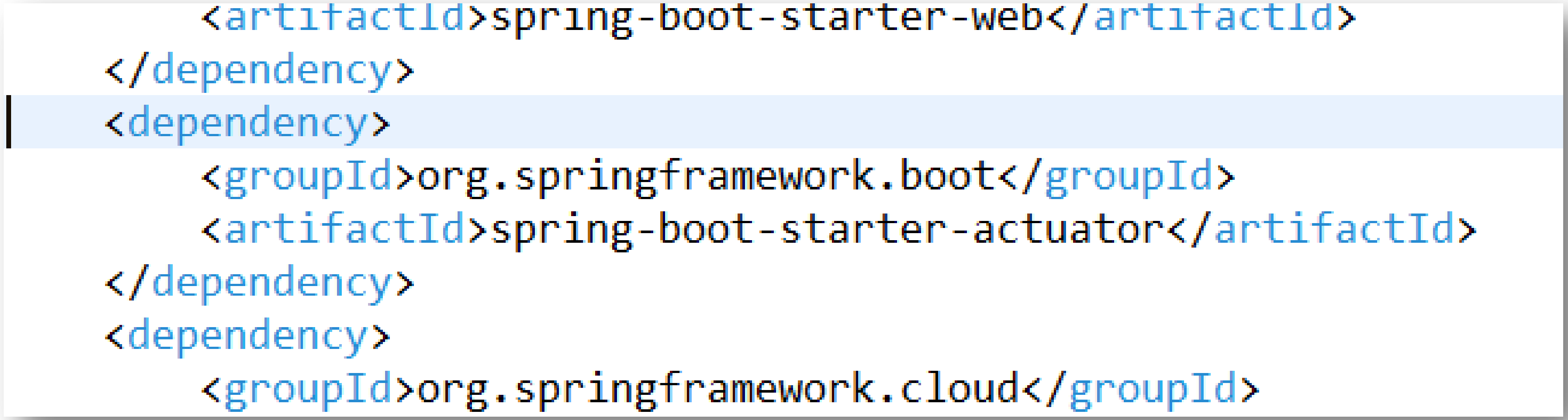
- This is because microservice loads the config details while starting the service, If we restart our microservices it will load updated, but just for small changes we should not restart



```
http://localhost:8082/api/course/profile
```

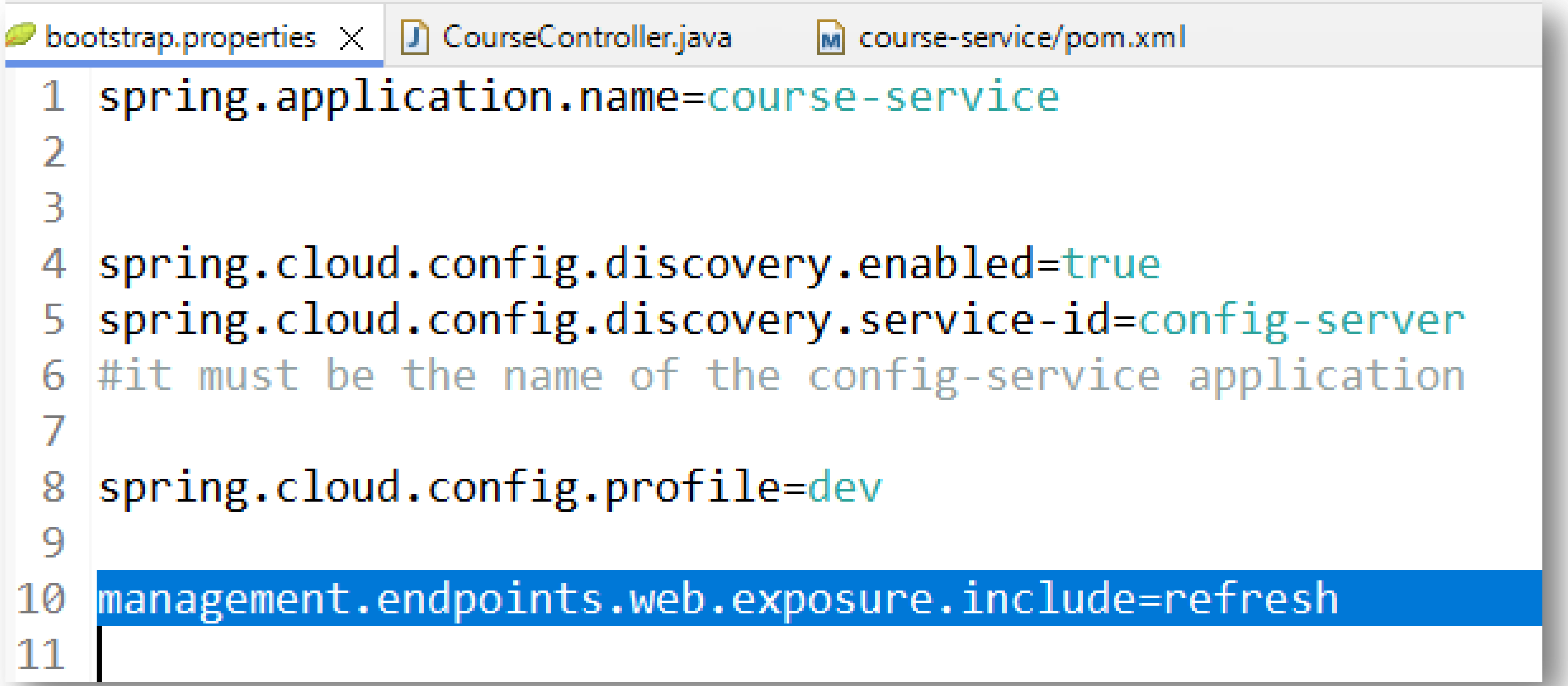
```
1  DevelopmentProfile
```

For reading updated values
Add Actuator dependency in all microservices



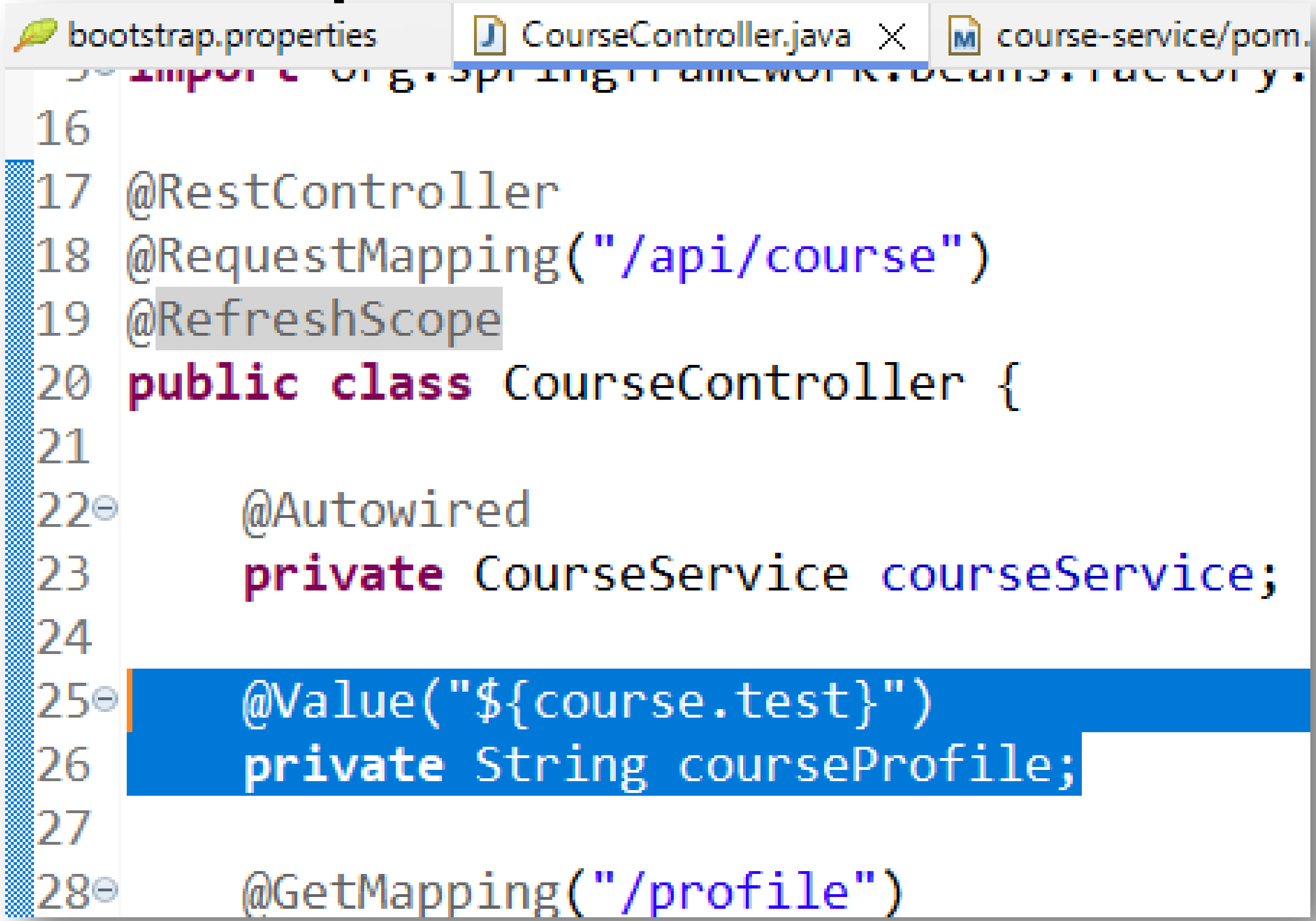
```
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-actuator</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.cloud</groupId>
```

We need to include a property in microservice



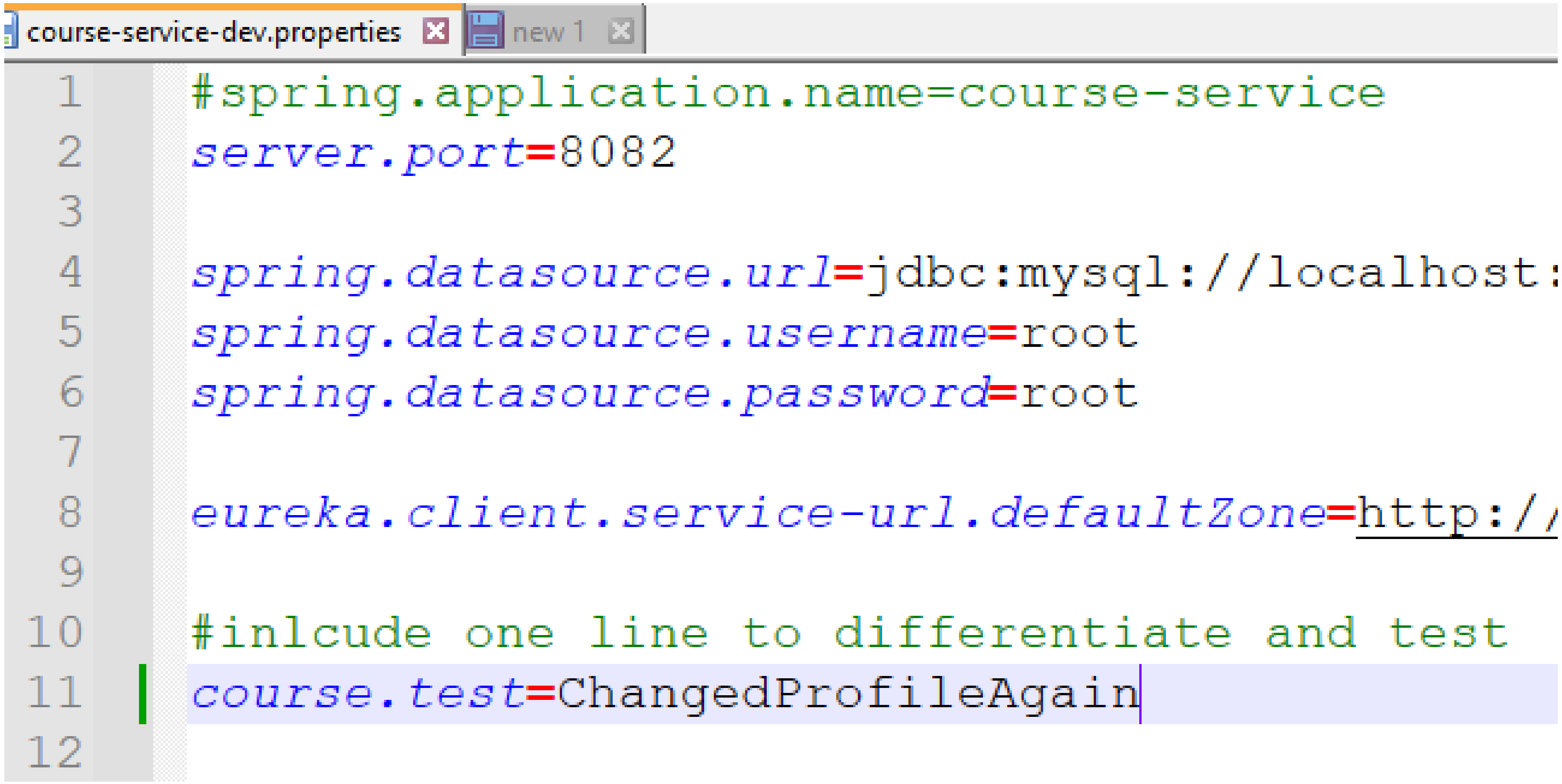
```
spring.application.name=course-service


spring.cloud.config.discovery.enabled=true
spring.cloud.config.discovery.service-id=config-server
#it must be the name of the config-service application

spring.cloud.config.profile=dev

management.endpoints.web.exposure.include=refresh

```

In which classes we use @Value{} we have to include an annotation @RefreshScope



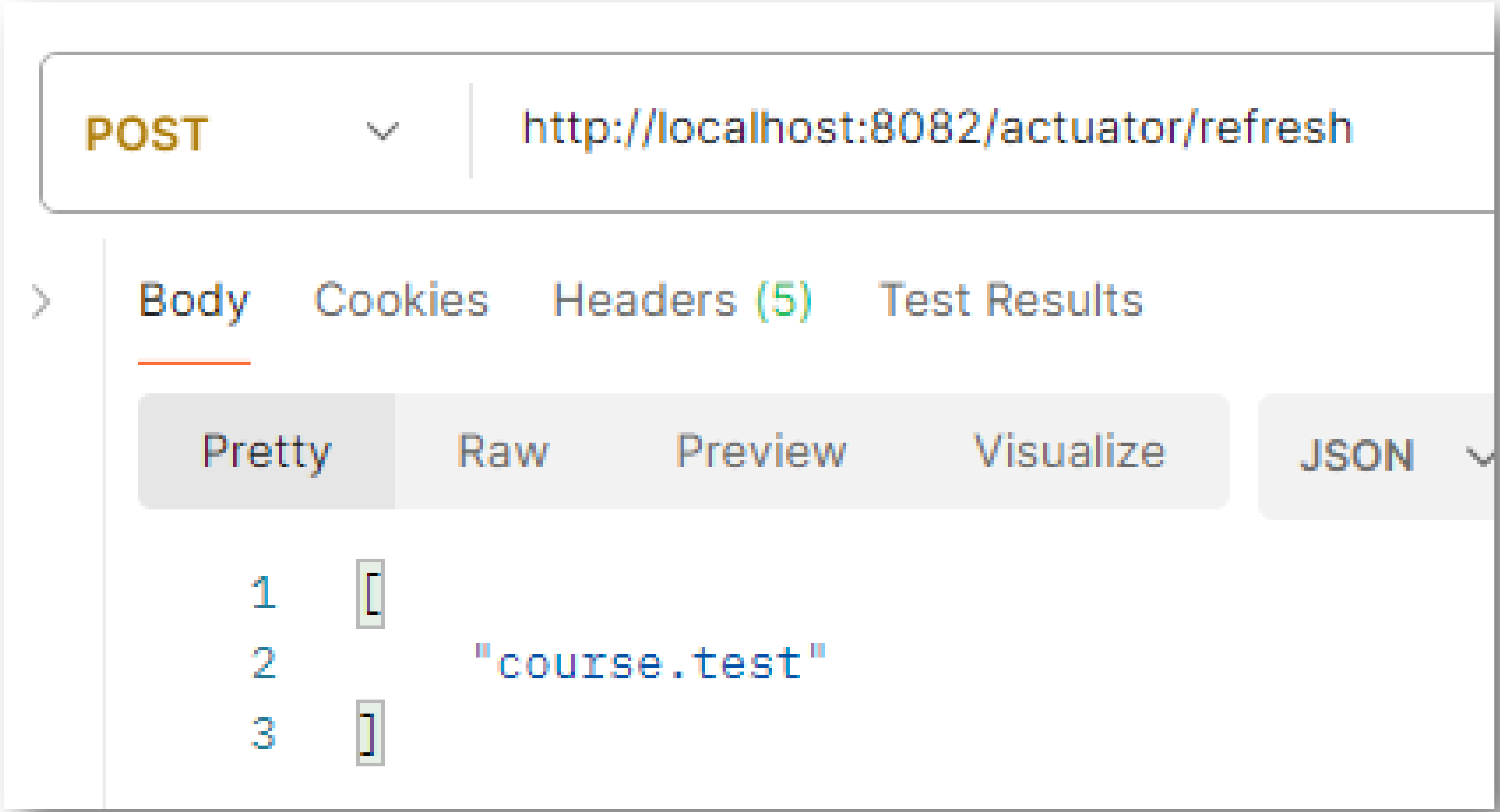
```java

@RestController
@RequestMapping("/api/course")
@RefreshScope
public class CourseController {

    @Autowired
    private CourseService courseService;

    @Value("${course.test}")
    private String courseProfile;

    @GetMapping("/profile")
```

Update the data



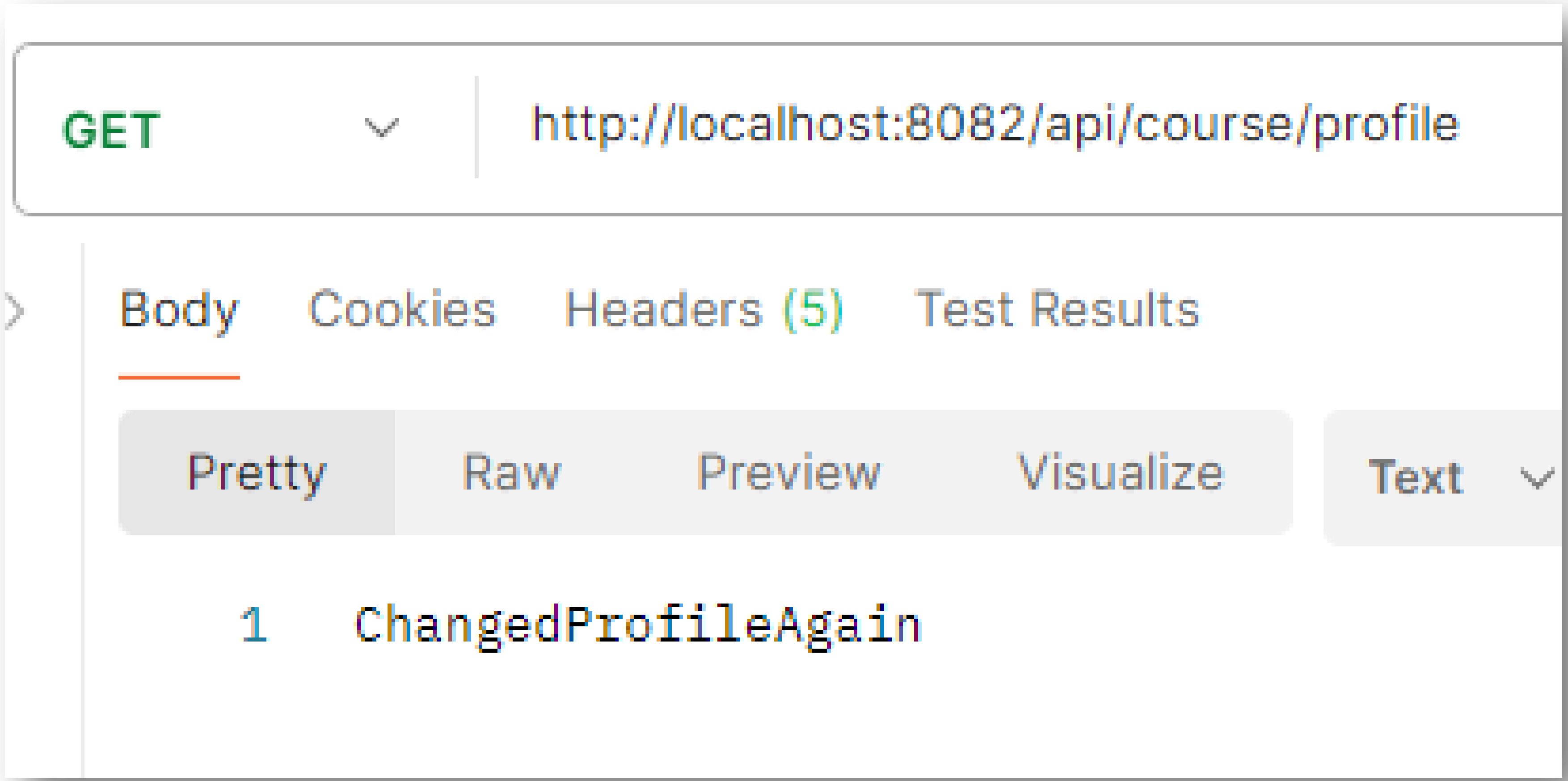
```
#spring.application.name=course-service
server.port=8082

spring.datasource.url=jdbc:mysql://localhost:
spring.datasource.username=root
spring.datasource.password=root

eureka.client.service-url.defaultZone=http://

#inlcude one line to differentiate and test
course.test=ChangedProfileAgain

```

Post actuator refresh


POST http://localhost:8082/actuator/refresh

```
[
    "course.test"
]
```

No need the restart any service



```
GET http://localhost:8082/api/course/profile
```

```
ChangedProfileAgain
```

Thank you