**Creating Microservices for account and loan**

**Account Microservice**

**Controller class**

package com.cognizant.account.controller;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.RestController;

import java.util.Map;

@RestController

public class AccountController {

@GetMapping("/accounts/{number}")

public Map<String, Object> getAccountDetails(@PathVariable String number) {

return Map.of(

"number", number,

"type", "savings",

"balance", 234343

);

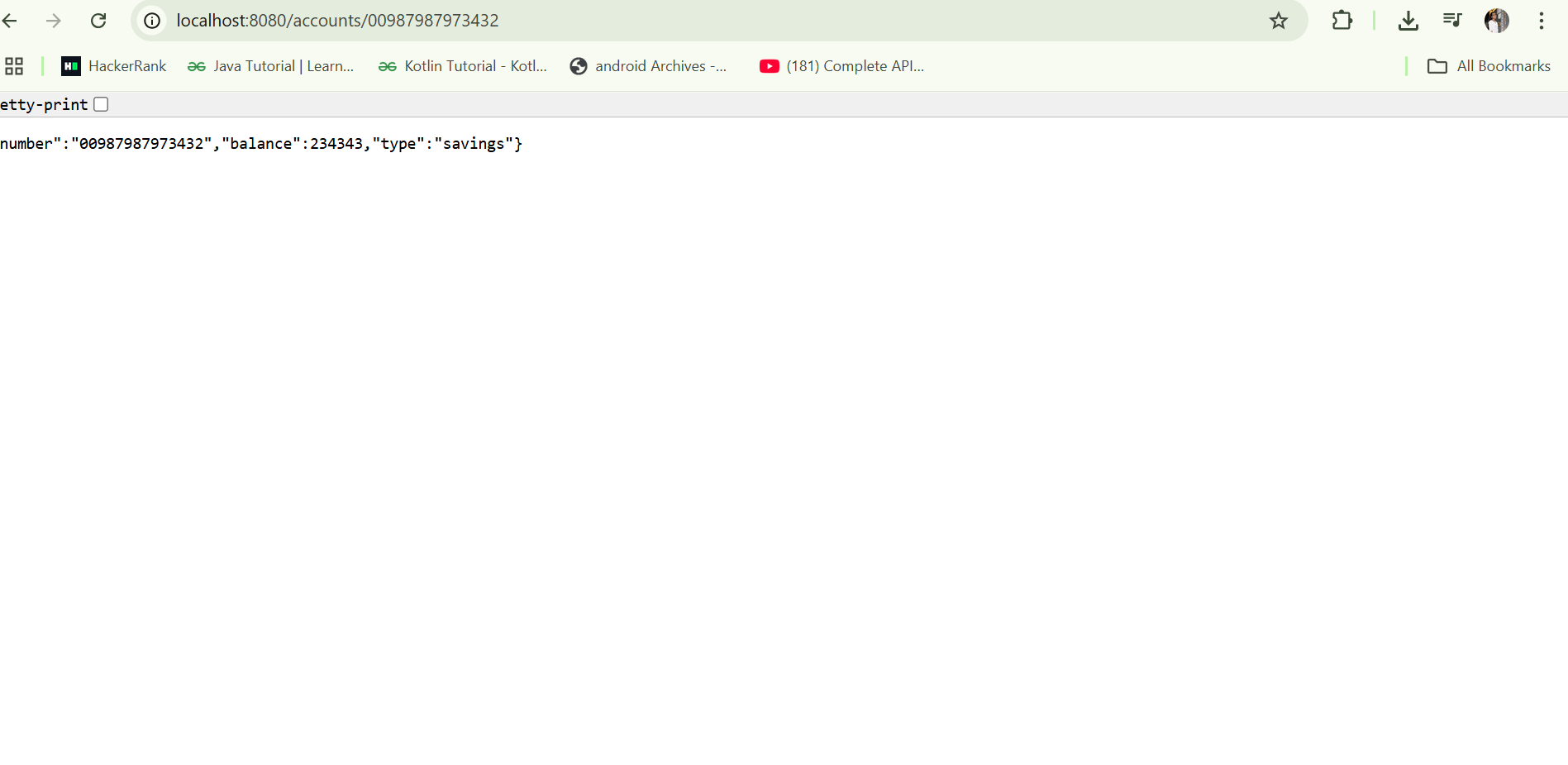
}

}

**URL:**

[**http://localhost:8080/accounts/00987987973432**](http://localhost:8080/accounts/00987987973432)

**Output:**

****

**Loan Microservice**

**Loan Controller Class**

package com.cognizant.loan.controller;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.RestController;

import java.util.Map;

@RestController

public class LoanController {

@GetMapping("/loans/{number}")

public Map<String, Object> getLoanDetails(@PathVariable String number) {

return Map.of(

"number", number,

"type", "car",

"loan", 400000,

"emi", 3258,

"tenure", 18

);

}

}

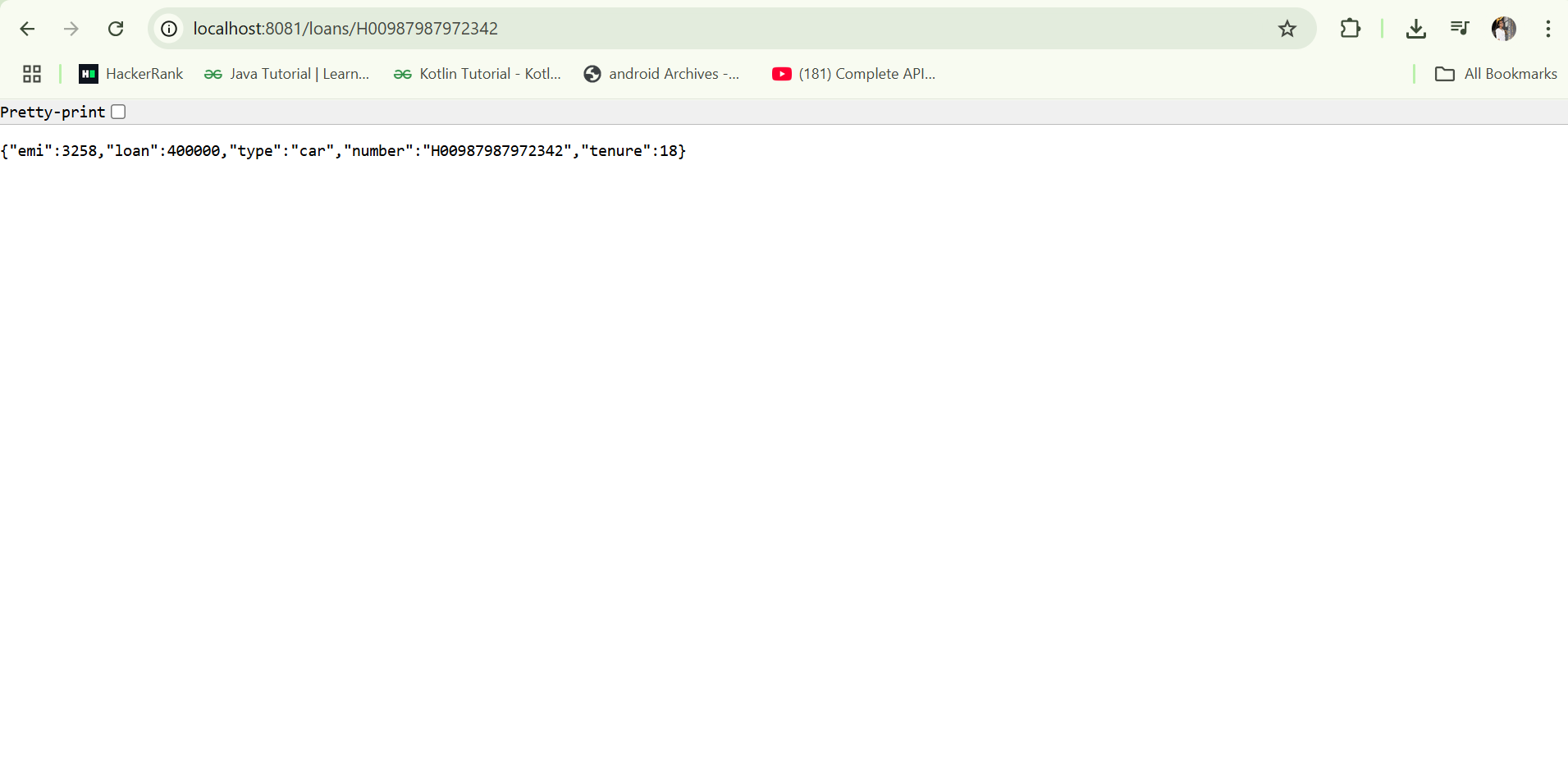
**application.properties**

server.port=8081

**URL:**

[**http://localhost:8081/loans/H00987987972342**](http://localhost:8081/loans/H00987987972342)

**Output:**

****

**Create Eureka Discovery Server and register**

**EurekaDiscoveryServerApplication.java**

package com.cognizant.eureka\_discovery\_server;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

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

@SpringBootApplication

@EnableEurekaServer

public class EurekaDiscoveryServerApplication {

public static void main(String[] args) {

SpringApplication.run(EurekaDiscoveryServerApplication.class, args);

}

}

**application.properties**

server.port=8761

spring.application.name=edureka

# Optional: disable self-registration

eureka.client.register-with-eureka=false

eureka.client.fetch-registry=false

**pom.xml**

<dependencies>

<dependency>

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

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

</dependency>

</dependencies>

<dependencyManagement>

<dependencies>

<dependency>

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

<artifactId>spring-cloud-dependencies</artifactId>

<version>2023.0.1</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

**Register Account and Loan Services as Eureka Clients**

pom.xml → Add Eureka Client Dependency

<dependency>

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

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

</dependency>

**Main Application Class (for both)**

**AccountApplication.java or LoanApplication.java**

package com.cognizant.account; // or loan

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.client.discovery.EnableDiscoveryClient;

@SpringBootApplication

@EnableDiscoveryClient

public class AccountApplication {

public static void main(String[] args) {

SpringApplication.run(AccountApplication.class, args);

}

}

**application.properties (account)**

server.port=8083

spring.application.name=account-service

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

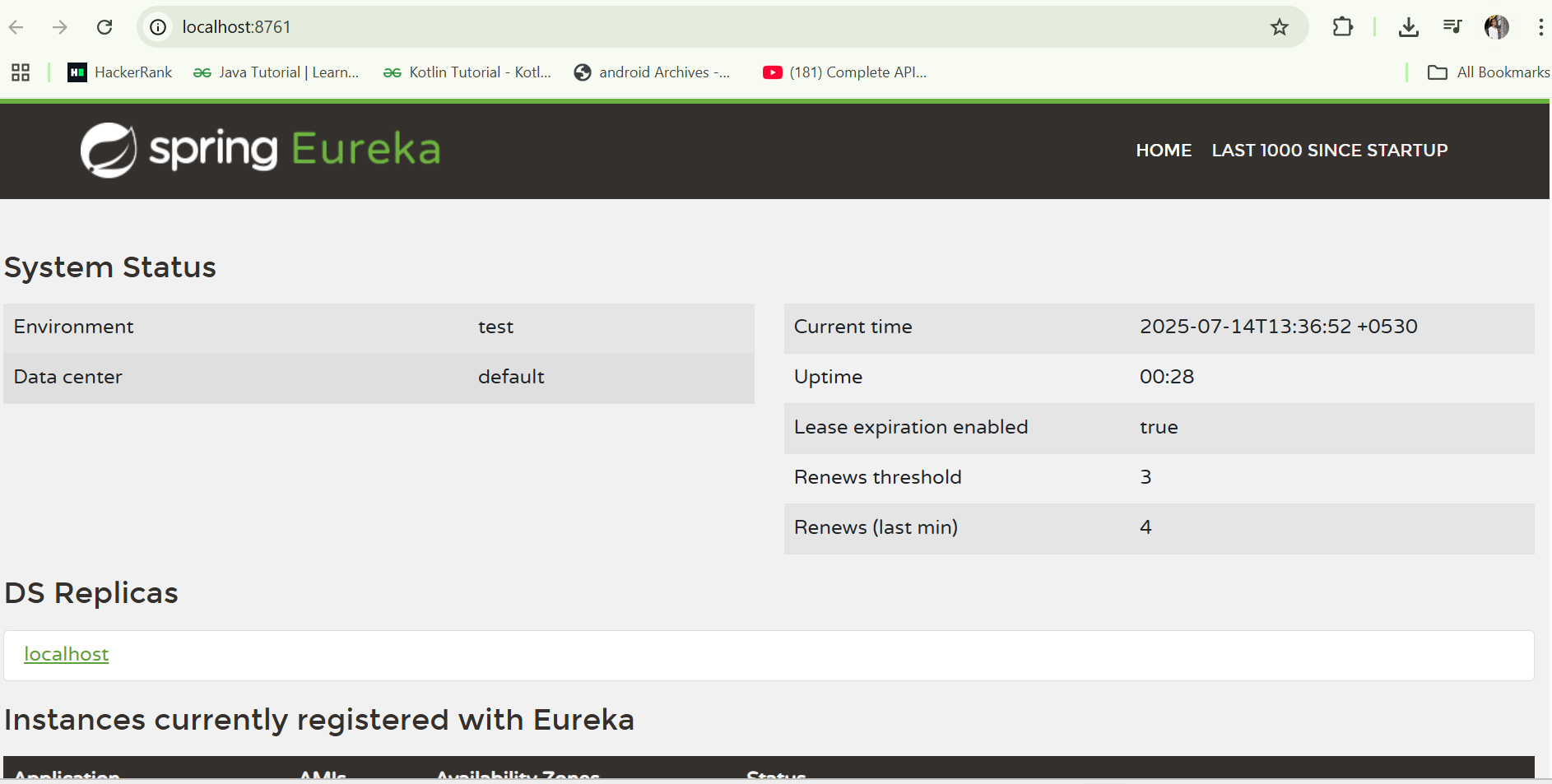
**application.properties(loan)**

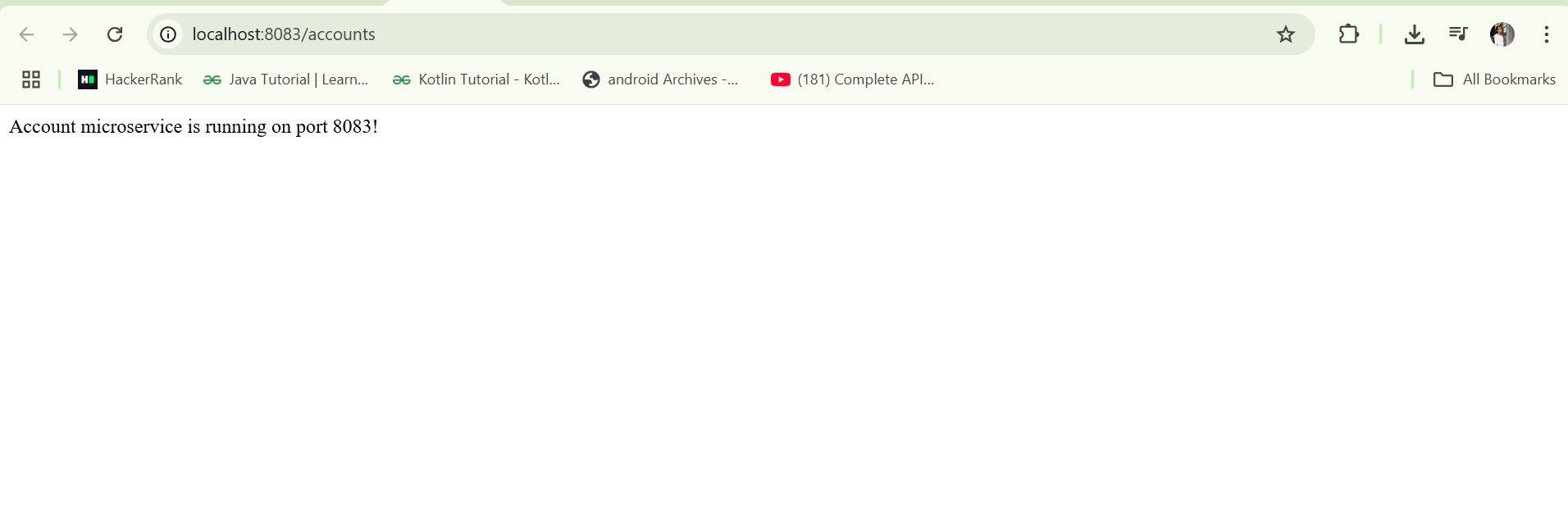
server.port=8084

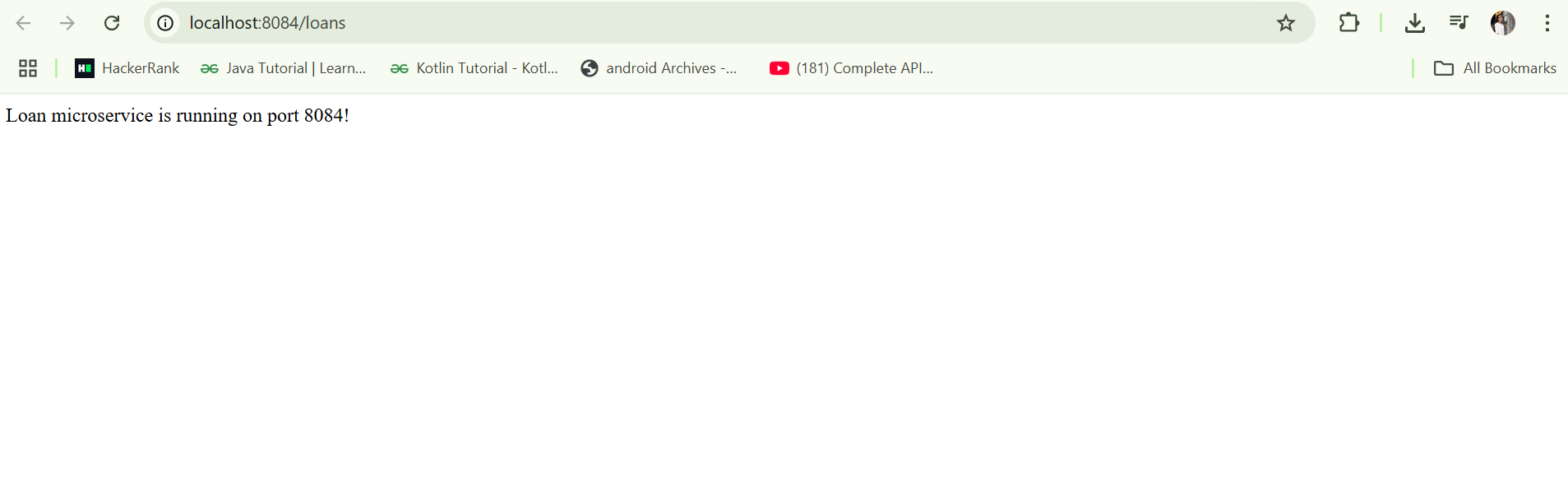
spring.application.name=loan-service

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

**Output:**

****

****

****