**Creating Microservices for account and loan**

**Step 1: Create account Microservice**

**Tasks:**

1. Go to <https://start.spring.io/>
2. Fill in:
   * **Group:** com.cognizant
   * **Artifact:** account
3. Select Dependencies:
   * Spring Boot DevTools
   * Spring Web
4. Click **Generate**, download and extract the ZIP.

**Step 2: Import into Eclipse & Implement Controller**

1. Open Eclipse.
2. Import the Maven project (File > Import > Maven > Existing Maven Projects).

**AccountController.java**

package com.cognizant.account;

import org.springframework.web.bind.annotation.\*;

*@RestController*

*@RequestMapping*("/accounts")

public class AccountController {

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

public Account getAccount(*@PathVariable* String number) {

return new Account(number, "savings", 234343);

}

static class Account {

public String number;

public String type;

public int balance;

public Account(String number, String type, int balance) {

this.number = number;

this.type = type;

this.balance = balance;

}

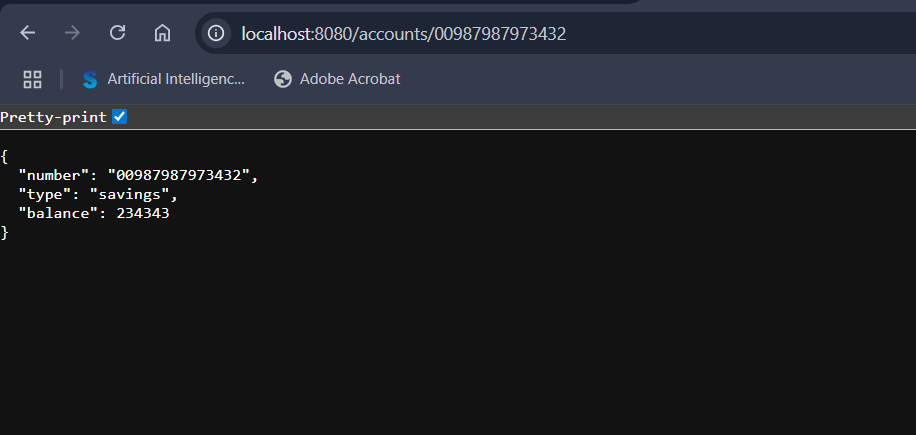
}

}

**3. Run Application**

* Right-click on main class: AccountApplication.java → Run As > Java Application
* Open your browser and go to: http://localhost:8080/accounts/00987987973432

**Output:**



**Step 4: Create loan Microservice**

**Tasks:**

1. Go to <https://start.spring.io>
2. Fill in:
   * **Group:** com.cognizant
   * **Artifact:** loan
3. Add Dependencies:
   * Spring Boot DevTools
   * Spring Web
4. Click **Generate**, download, and extract the ZIP.

**Step 5: Import & Implement the Controller**

1. Import the loan project into Eclipse.
2. In src/main/java, create a controller class named LoanController.java under the package: com.cognizant.loan

**LoanController.java**

package com.cognizant.loan;

import org.springframework.web.bind.annotation.\*;

@RestController

@RequestMapping("/loans")

public class LoanController {

@GetMapping("/{number}")

public Loan getLoan(@PathVariable String number) {

return new Loan(number, "car", 400000, 3258, 18);

}

static class Loan {

public String number;

public String type;

public int loan;

public int emi;

public int tenure;

public Loan(String number, String type, int loan, int emi, int tenure) {

this.number = number;

this.type = type;

this.loan = loan;

this.emi = emi;

this.tenure = tenure;

}

}

}

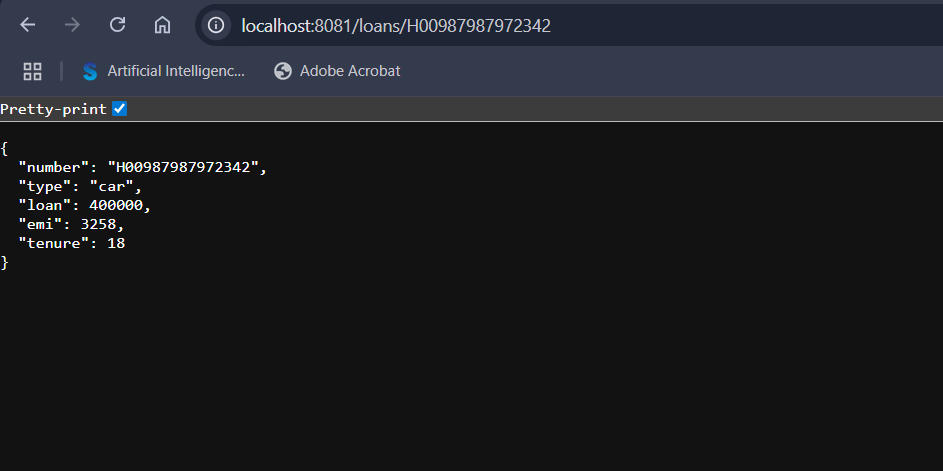
**Step 6: Change Port (To Avoid Conflict with account)**

Edit application.properties of the loan project:

server.port=8081

**Step 7: Run and Test**

1. Start the loan application (LoanApplication.java).
2. Test in browser or Postman: <http://localhost:8081/loans/H00987987972342>



**Create Eureka Discovery Server and register microservices**

**Step 1: Create Eureka Discovery Server**

**Tasks:**

1. Go to <https://start.spring.io>
2. Fill in:
   * **Group:** com.cognizant
   * **Artifact:** eureka-discovery-server
3. Add Dependency:
   * Spring Cloud Discovery > Eureka Server
4. Click **Generate**, download, and extract the ZIP.

**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);

}

}

**Edit application.properties**

server.port=8761

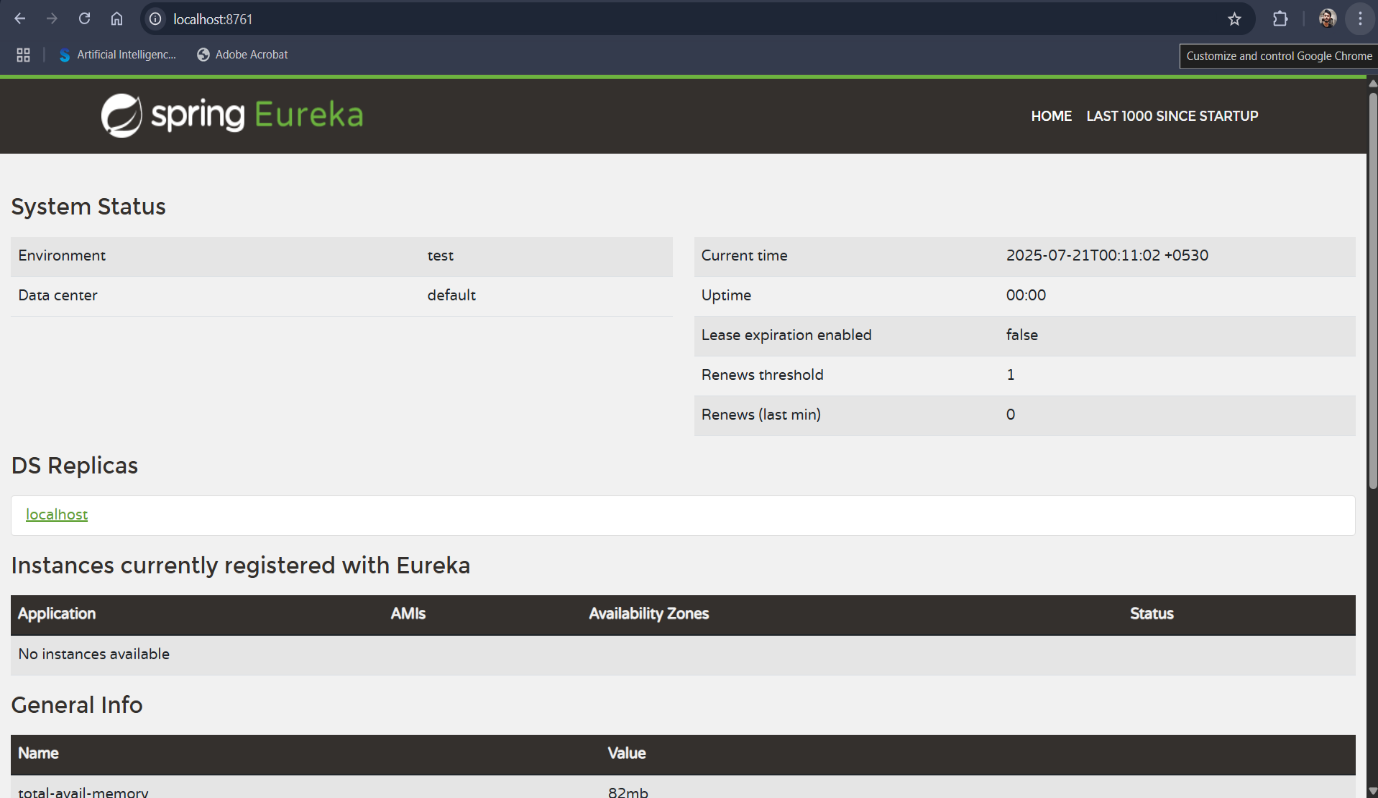
eureka.client.register-with-eureka=false

eureka.client.fetch-registry=false

logging.level.com.netflix.eureka=OFF

logging.level.com.netflix.discovery=OFF

**Run the Discovery Server**

****

**Step 2: Register account Microservice with Eureka**

**Tasks:**

1. Go to <https://start.spring.io>
2. Enter:
   * **Group:** com.cognizant
   * **Artifact:** account
3. Add dependencies:
   * Spring Web
   * Spring Boot DevTools
   * Eureka Discovery Client
4. Click **Explore** and copy the pom.xml contents.
5. Replace the existing pom.xml of your **account** project with this new one.
6. In Eclipse, right-click the project → Maven > Update Project (to sync dependencies).

**AccountApplication.java**

package com.cognizant.account;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

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

*@EnableDiscoveryClient*

*@SpringBootApplication*

public class AccountApplication {

public static void main(String[] args) {

SpringApplication.*run*(AccountApplication.class, args);

}

}

**Update application.properties**

spring.application.name=account-service

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

**Step 3: Register loan Microservice with Eureka**

Repeat the same steps for the **loan** project:

1. Use Spring Initializr to get updated pom.xml with:
   * Spring Web
   * Spring Boot DevTools
   * Eureka Discovery Client
2. Replace your loan project's pom.xml.
3. Add @EnableDiscoveryClient to the main class LoanApplication.java.
4. Update **application.properties:**

server.port=8081

spring.application.name=loan-service

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

**Step 4: Run and Verify**

1. Stop all services in Eclipse.
2. Start:
   * eureka-discovery-server
   * Then account
   * Then loan
3. Go to: <http://localhost:8761>

