

NAME:Vidya CS

TRACK: JAVA FSE

Creating Microservices for account and loan

I) Creating account section

Step1:-Setup Project Folder

Step2:-Create the Account Microservice

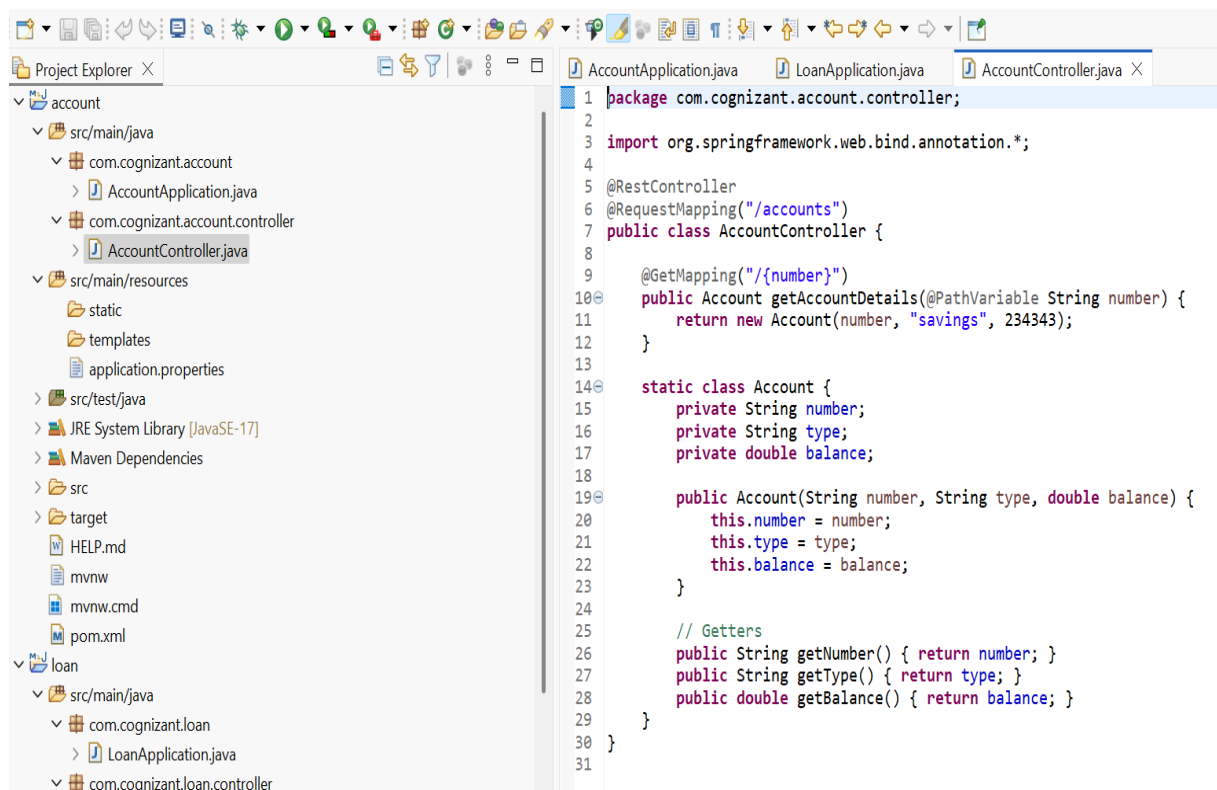
2.1 Generate Spring Boot Project

2.2 Build the Project

2.3 Import in Eclipse

2.4 set the port :- server.port=7080 in application.properties

Step3:- Create Controller for Account



The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer displays the project structure for 'account' and 'loan'. The 'account' project is expanded, showing the package 'com.cognizant.account' with files 'AccountApplication.java' and 'AccountController.java'. The 'loan' project is also expanded, showing 'LoanApplication.java'. The main editor on the right displays the code for 'AccountController.java'. The code includes package declarations, imports, annotations, and a REST controller for account details.

```
1 package com.cognizant.account.controller;
2
3 import org.springframework.web.bind.annotation.*;
4
5 @RestController
6 @RequestMapping("/accounts")
7 public class AccountController {
8
9     @GetMapping("/{number}")
10    public Account getAccountDetails(@PathVariable String number) {
11        return new Account(number, "savings", 234343);
12    }
13
14    static class Account {
15        private String number;
16        private String type;
17        private double balance;
18
19        public Account(String number, String type, double balance) {
20            this.number = number;
21            this.type = type;
22            this.balance = balance;
23        }
24
25        // Getters
26        public String getNumber() { return number; }
27        public String getType() { return type; }
28        public double getBalance() { return balance; }
29    }
30 }
31
```

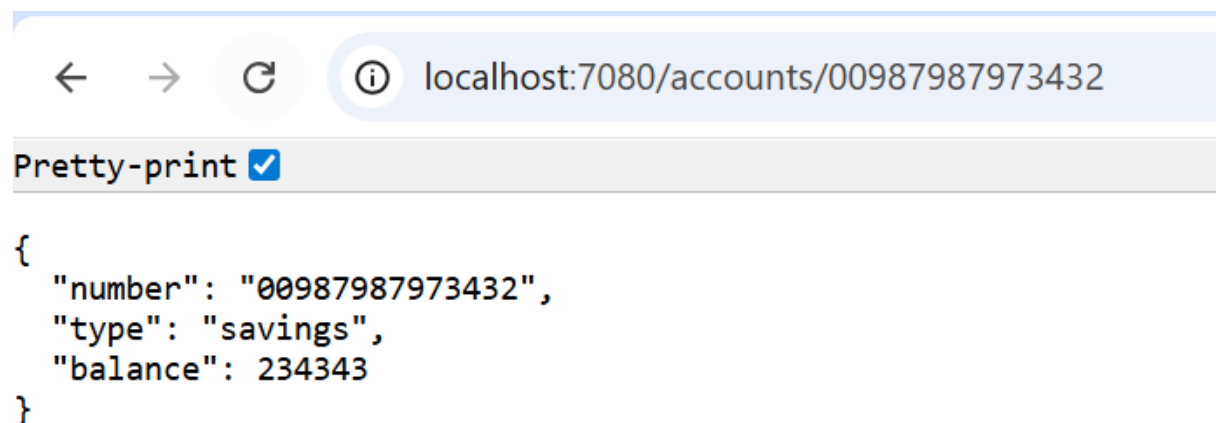
Step 4:- Run & Test Account Microservice

Run :- AccountApplication.java

Output :- At console

```
.account.AccountApplication : Starting AccountApplication using Java 21.0.7 with PID 40072 (C:\Users\Dharan Gowda\OneDrive\Desktop\ec
.account.AccountApplication : No active profile set, falling back to 1 default profile: "default"
PropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.add-properties' to 'false' to disable
PropertyDefaultsPostProcessor : For additional web related logging consider setting the 'logging.level.web' property to 'DEBUG'
aded.tomcat.TomcatWebServer : Tomcat initialized with port 7080 (http)
alina.core.StandardService : Starting service [Tomcat]
alina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.42]
Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
stWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1563 ms
ptionalLiveReloadServer : LiveReload server is running on port 35729
aded.tomcat.TomcatWebServer : Tomcat started on port 7080 (http) with context path '/'
.account.AccountApplication : Started AccountApplication in 3.002 seconds (process running for 3.616)
```

At browser :- running at port 7080



II) Creating Loan section

Step1:-Setup Project Folder

Step2:-Create the Loan Microservice

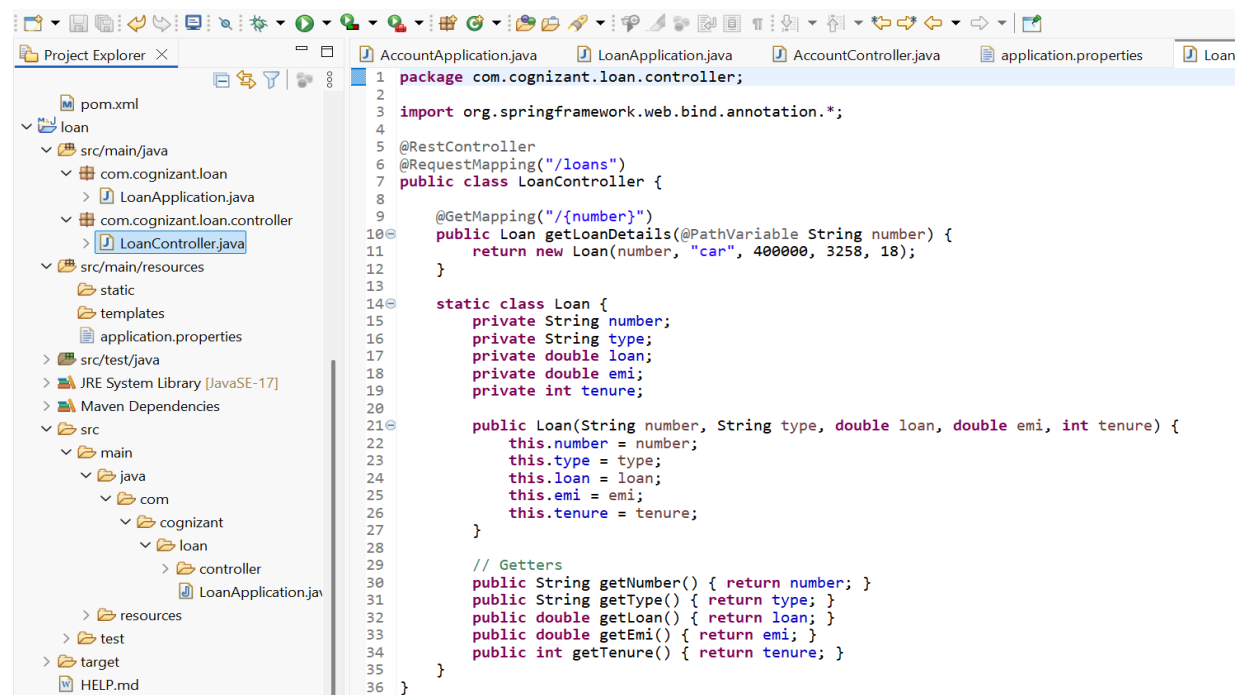
2.1 Generate Spring Boot Project

2.2 Build the Project

2.3 Import in Eclipse

2.4 set the port :- server.port=7081 in application.properties

Step 3: Create Controller for Loan



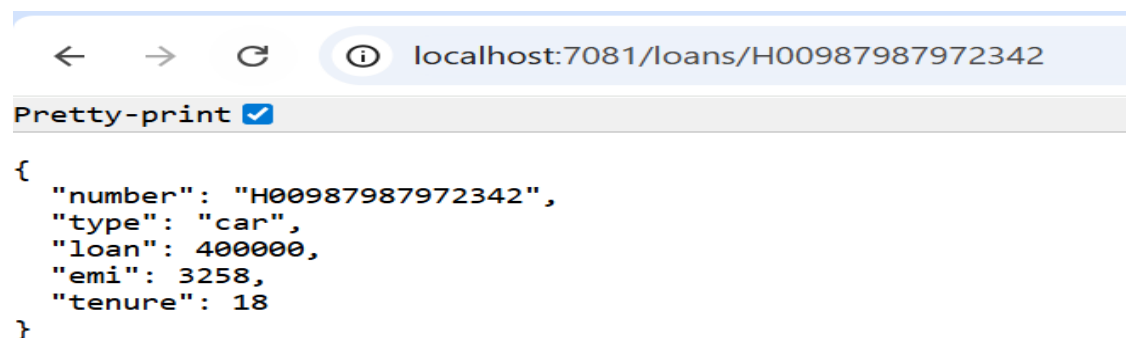
Step 4:- Run & Test Loan Microservice

Run :- LoanApplication.java

Output :- At console

```
loan] [ restartedMain] com.cognizant.loan.LoanApplication : Starting LoanApplication using Java 21.0.7 with PID 10800 (C:\Users
loan] [ restartedMain] com.cognizant.loan.LoanApplication : No active profile set, falling back to 1 default profile: "default"
loan] [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.add-propert
loan] [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting the 'logging.le
loan] [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 7081 (http)
loan] [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
loan] [ restartedMain] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.42]
loan] [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
loan] [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1859 ms
loan] [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : Unable to start LiveReload server
loan] [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 7081 (http) with context path '/'
loan] [ restartedMain] com.cognizant.loan.LoanApplication : Started LoanApplication in 3.394 seconds (process running for 3.968
loan] [nio-7081-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
loan] [nio-7081-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
loan] [nio-7081-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 1 ms
```

At browser :- running at port 7081



Now we have implemented two independent Spring Boot microservices as part of a hands-on banking exercise:

1. Account Microservice:

- Runs on port 7080
- Endpoint: GET /accounts/{number}
- Returns dummy account details in JSON format

2. Loan Microservice:

- Runs on port 7081 (customized from default 8081)
- Endpoint: GET /loans/{number}
- Returns dummy loan details in JSON format