**1.2** **Creating Microservices for Account and Loan Management**

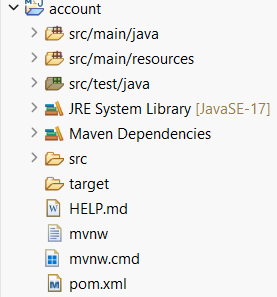
**Objective:**

To understand and implement microservices architecture by building two independent Spring Boot RESTful web services – one for handling account details and the other for handling loan details. i.e.,

* **Account Microservice**
* **Loan Microservice**

Each microservice will be developed as a standalone Spring Boot RESTful service with its own Maven project, running independently on different ports. Both service will return dummy JSON responses.

**Project Structure:**



Building the Account Microservice

**Step1**: Generate Spring Boot Project

1. Open Spring intializr in chrome.
2. Fill the form as follows:

* Group: com.cognizant
* Artifact: account

1. Select the dependencies:

* Spring Boot DevTools
* Spring Web

1. Click generate, and download ZIP file.
2. The, extract the zip file and place it in eclipse workspace.

**Step2:** Build the Project

Initially, run the below command to update the project:

mvn clean package

**Step3**: Import into Eclipse and Implement Controller

1. Open Eclipse Ide and then import the existing maven project.
2. Create a RestController named AccountController:

i.e., AccountController.java

|  |
| --- |
| 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  );  }  } |

Also, the AccountApplication.java is:

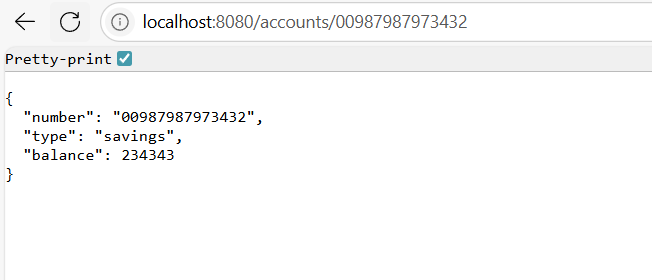
|  |
| --- |
| package com.cognizant.account;  import org.springframework.boot.SpringApplication;  import org.springframework.boot.autoconfigure.SpringBootApplication;  @SpringBootApplication  public class AccountApplication {  public static void main(String[] args) {  SpringApplication.*run* (AccountApplication.class, args);  }  } |

**Step4**: Run and Test the Application

1. Runt he Application i.e., AccountApplication as java application
2. And open the browser and test:

<http://localhost:8080/accounts/00987987973432>

Then the outcome is,



Also, the Outcome in the Console is:

. \_\_\_\_ \_ \_\_ \_ \_

/\\ / \_\_' \_\_ \_ ()\_ \_\_ \_\_ \_ \ \ \ \

( ( )\\_\_\_ | '\_ | '| | ' \/ \_` | \ \ \ \

\\/ \_\_)| |)| | | | | || (\_| | ) ) ) )

' || .|| ||| |\, | / / / /

=========||==============|/=///\_/

:: Spring Boot :: (v3.5.3)

2025-07-20T23:07:30.487+05:30 INFO 2064 --- [account] [ restartedMain] c.cognizant.account.AccountApplication : Starting AccountApplication using Java 21.0.1 with PID 2064 (C:\Users\HP\eclipse-workspace\account\target\classes started by HP in C:\Users\HP\eclipse-workspace\account)

2025-07-20T23:07:30.495+05:30 INFO 2064 --- [account] [ restartedMain] c.cognizant.account.AccountApplication : No active profile set, falling back to 1 default profile: "default"

2025-07-20T23:07:30.617+05:30 INFO 2064 --- [account] [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.add-properties' to 'false' to disable

2025-07-20T23:07:30.617+05:30 INFO 2064 --- [account] [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting the 'logging.level.web' property to 'DEBUG'

2025-07-20T23:07:32.917+05:30 INFO 2064 --- [account] [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)

2025-07-20T23:07:32.956+05:30 INFO 2064 --- [account] [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]

2025-07-20T23:07:32.956+05:30 INFO 2064 --- [account] [ restartedMain] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.42]

2025-07-20T23:07:33.047+05:30 INFO 2064 --- [account] [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext

2025-07-20T23:07:33.050+05:30 INFO 2064 --- [account] [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 2429 ms

2025-07-20T23:07:33.932+05:30 INFO 2064 --- [account] [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729

2025-07-20T23:07:34.009+05:30 INFO 2064 --- [account] [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path '/'

2025-07-20T23:07:34.029+05:30 INFO 2064 --- [account] [ restartedMain] c.cognizant.account.AccountApplication : Started AccountApplication in 4.531 seconds (process running for 5.557)

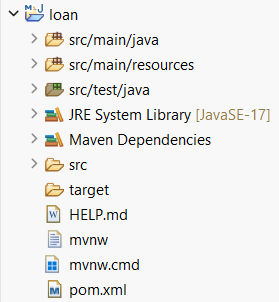
2025-07-20T23:07:37.861+05:30 INFO 2064 --- [account] [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'

2025-07-20T23:07:37.861+05:30 INFO 2064 --- [account] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'

2025-07-20T23:07:37.863+05:30 INFO 2064 --- [account] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 2 ms

Building the Loan Microservice

**Step1:** Generate Spring Boot Project



**Step2**: Build the Project

The project is build using the maven build:

i.e., mvn clean package

**Step3**: Import and Implement Controller

1. Import the loan project into Eclipse.
2. Create a REST controller named LoanController:

i.e., LoanController.java

|  |
| --- |
| 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  );  }  } |

Also, LoanApplication.java is:

|  |
| --- |
| package com.cognizant.loan;  import org.springframework.boot.SpringApplication;  import org.springframework.boot.autoconfigure.SpringBootApplication;  @SpringBootApplication  public class LoanApplication {  public static void main(String[] args) {  SpringApplication.*run* (LoanApplication.class, args);  }  } |

**Step4:** Configure port.

Since port 8080 is already in use update the application.properties.

i.e.,

add **server.port = 8081**

Step5: Run the Application

The expected outcome is,

