# Microservices with Spring Boot 3 and Spring Cloud

# Creating Microservices for account and loan

# 1. Account Microservice

1. Create a folder with your employee ID in D: drive.  
2. Inside it, create a folder named 'microservices'.  
3. Go to https://start.spring.io/ and fill the details:  
 - Group: com.cognizant  
 - Artifact: account  
4. Select dependencies:  
 - Developer Tools > Spring Boot DevTools  
 - Web > Spring Web  
5. Click 'Generate' to download the project zip.  
6. Extract 'account' folder and place it inside 'microservices'.  
7. Open command prompt in the account folder and build with:  
 mvn clean package  
8. Import the project into Eclipse and add the following controller code.

## 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.HashMap;  
import java.util.Map;  
  
@RestController  
public class AccountController {  
  
 @GetMapping("/accounts/{number}")  
 public Map<String, Object> getAccount(@PathVariable String number) {  
 Map<String, Object> account = new HashMap<>();  
 account.put("number", "00987987973432");  
 account.put("type", "savings");  
 account.put("balance", 234343);  
 return account;  
 }  
}

# 2. Loan Microservice

1. Repeat the steps above with Artifact: loan.  
2. Select the same dependencies (Spring Boot DevTools and Spring Web).  
3. Extract 'loan' folder inside 'microservices'.  
4. Import the project into Eclipse and add the following controller code.  
5. Add 'server.port=8081' in application.properties file to avoid port conflict.

## 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.HashMap;  
import java.util.Map;  
  
@RestController  
public class LoanController {  
  
 @GetMapping("/loans/{number}")  
 public Map<String, Object> getLoan(@PathVariable String number) {  
 Map<String, Object> loan = new HashMap<>();  
 loan.put("number", "H00987987972342");  
 loan.put("type", "car");  
 loan.put("loan", 400000);  
 loan.put("emi", 3258);  
 loan.put("tenure", 18);  
 return loan;  
 }  
}

## application.properties

server.port=8081

# 3. Testing

• Start the Account microservice and test it using: http://localhost:8080/accounts/00987987973432  
• Start the Loan microservice and test it using: http://localhost:8081/loans/H00987987972342  
Now both microservices will be running on separate ports.