**Title: Hands-On Lab: Creating Account and Loan Microservices**

**1. Introduction**

This document outlines the step-by-step process to create two standalone microservices, account and loan, using Spring Boot. This exercise follows the initial hands-on section of the provided PDF guide, focusing on creating independent services without a discovery client.

**2. Account Microservice**

This service is responsible for providing bank account details.

**2.1 Process and Code**

1. **Project Setup**: Create a new Spring Boot project using Spring Initializr with the following details:
   * **Group**: com.cognizant
   * **Artifact**: account
   * **Dependencies**: Spring Boot DevTools, Spring Web
2. **Create Model Class**: In the src/main/java directory, create the package com.cognizant.account.model and add the following file.
   * **File**: Account.java

Java

package com.cognizant.account.model;

public class Account {

public String number;

public String type;

public double balance;

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

this.number = number;

this.type = type;

this.balance = balance;

}

}

1. **Create Controller Class**: In src/main/java, create the package com.cognizant.account.controller and add the following file.
   * **File**: AccountController.java

Java

package com.cognizant.account.controller;

import com.cognizant.account.model.Account;

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

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

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

@RestController

public class AccountController {

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

public Account getAccountDetails(@PathVariable String number) {

return new Account("00987987973432", "savings", 234343);

}

}

**2.2 Execution and Output**

1. **Run**: Execute the main AccountApplication.java class. The service will start on port 8080.
2. **Test**: Open a web browser and navigate to http://localhost:8080/accounts/any-number.
3. **Output**: The browser will display the following JSON output.

**(You can take a screenshot of your browser showing the output below and insert it here in your Word document)**

JSON

{

"number": "00987987973432",

"type": "savings",

"balance": 234343

}

**3. Loan Microservice**

This service is responsible for providing loan account details.

**3.1 Process and Code**

1. **Project Setup**: Create a new Spring Boot project with the same dependencies as the account service.
2. **Configure Port**: The application.properties file must be modified to run the service on a different port and avoid conflict.
   * **File**: src/main/resources/application.properties

Properties

server.port=8081

1. **Create Model Class**:
   * **File**: src/main/java/com/cognizant/loan/model/Loan.java

Java

package com.cognizant.loan.model;

public class Loan {

public String number;

public String type;

public long loan;

public int emi;

public int tenure;

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

this.number = number;

this.type = type;

this.loan = loan;

this.emi = emi;

this.tenure = tenure;

}

}

1. **Create Controller Class**:
   * **File**: src/main/java/com/cognizant/loan/controller/LoanController.java

Java

package com.cognizant.loan.controller;

import com.cognizant.loan.model.Loan;

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

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

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

@RestController

public class LoanController {

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

public Loan getLoanDetails(@PathVariable String number) {

return new Loan("H00987987972342", "car", 400000, 3258, 18);

}

}

