Microservices: Account Service, Loan Service, API Gateway

This document contains runnable Spring Boot projects for three microservices:  
1. account-service (port 8081)  
2. loan-service (port 8082)  
3. api-gateway (port 8080) using Spring Cloud Gateway  
  
Each project includes a pom.xml, main application class, models, controllers, simple in-memory repositories, and sample data. Follow the instructions for each project to run them.  
  
Note: These examples use Spring Boot 2.7.x style dependencies. If you use a different Spring Boot version, adjust dependency versions accordingly.

## account-service/pom.xml

<project xmlns="http://maven.apache.org/POM/4.0.0"   
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0   
 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <groupId>com.example</groupId>  
 <artifactId>account-service</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>2.7.12</version>  
 <relativePath/>   
 </parent>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
 </dependencies>  
 <properties>  
 <java.version>11</java.version>  
 </properties>  
</project>

## account-service/src/main/resources/application.properties

server.port=8081  
spring.application.name=account-service

## account-service/src/main/java/com/example/accountservice/AccountServiceApplication.java

package com.example.accountservice;  
  
import org.springframework.boot.CommandLineRunner;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.context.annotation.Bean;  
  
import java.util.Map;  
import java.util.concurrent.ConcurrentHashMap;  
  
@SpringBootApplication  
public class AccountServiceApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.run(AccountServiceApplication.class, args);  
 }  
  
 @Bean  
 public Map<Long, com.example.accountservice.model.Account> accountRepo() {  
 return new ConcurrentHashMap<>();  
 }  
  
 @Bean  
 CommandLineRunner init(Map<Long, com.example.accountservice.model.Account> repo) {  
 return args -> {  
 repo.put(1L, new com.example.accountservice.model.Account(1L, "Alice", "SAVINGS", 5000.0));  
 repo.put(2L, new com.example.accountservice.model.Account(2L, "Bob", "CURRENT", 12000.0));  
 };  
 }  
}

## account-service/src/main/java/com/example/accountservice/model/Account.java

package com.example.accountservice.model;  
  
public class Account {  
 private Long id;  
 private String ownerName;  
 private String type;  
 private Double balance;  
  
 public Account() {}  
  
 public Account(Long id, String ownerName, String type, Double balance) {  
 this.id = id; this.ownerName = ownerName; this.type = type; this.balance = balance;  
 }  
 // getters and setters  
 public Long getId() { return id; }  
 public void setId(Long id) { this.id = id; }  
 public String getOwnerName() { return ownerName; }  
 public void setOwnerName(String ownerName) { this.ownerName = ownerName; }  
 public String getType() { return type; }  
 public void setType(String type) { this.type = type; }  
 public Double getBalance() { return balance; }  
 public void setBalance(Double balance) { this.balance = balance; }  
}

## account-service/src/main/java/com/example/accountservice/controller/AccountController.java

package com.example.accountservice.controller;  
  
import com.example.accountservice.model.Account;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.Map;  
  
@RestController  
@RequestMapping("/accounts")  
public class AccountController {  
  
 private final Map<Long, Account> repo;  
  
 public AccountController(Map<Long, Account> repo) {  
 this.repo = repo;  
 }  
  
 @GetMapping  
 public ResponseEntity<Object> getAll() {  
 return ResponseEntity.ok(repo.values());  
 }  
  
 @GetMapping("/{id}")  
 public ResponseEntity<Account> getById(@PathVariable Long id) {  
 Account a = repo.get(id);  
 if (a == null) return ResponseEntity.notFound().build();  
 return ResponseEntity.ok(a);  
 }  
  
 @PostMapping  
 public ResponseEntity<Account> create(@RequestBody Account account) {  
 repo.put(account.getId(), account);  
 return ResponseEntity.ok(account);  
 }  
}

## loan-service/pom.xml

<project xmlns="http://maven.apache.org/POM/4.0.0"   
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0   
 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <groupId>com.example</groupId>  
 <artifactId>loan-service</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>2.7.12</version>  
 <relativePath/>   
 </parent>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
 </dependencies>  
 <properties>  
 <java.version>11</java.version>  
 </properties>  
</project>

## loan-service/src/main/resources/application.properties

server.port=8082  
spring.application.name=loan-service  
account.service.url=http://localhost:8081/accounts

## loan-service/src/main/java/com/example/loanservice/LoanServiceApplication.java

package com.example.loanservice;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.context.annotation.Bean;  
import org.springframework.web.client.RestTemplate;  
  
@SpringBootApplication  
public class LoanServiceApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.run(LoanServiceApplication.class, args);  
 }  
  
 @Bean  
 public RestTemplate restTemplate() {  
 return new RestTemplate();  
 }  
}

## loan-service/src/main/java/com/example/loanservice/model/Loan.java

package com.example.loanservice.model;  
  
public class Loan {  
 private Long id;  
 private Long accountId;  
 private Double amount;  
 private String status;  
  
 public Loan() {}  
 public Loan(Long id, Long accountId, Double amount, String status) {  
 this.id = id; this.accountId = accountId; this.amount = amount; this.status = status;  
 }  
 // getters and setters  
 public Long getId() { return id; }  
 public void setId(Long id) { this.id = id; }  
 public Long getAccountId() { return accountId; }  
 public void setAccountId(Long accountId) { this.accountId = accountId; }  
 public Double getAmount() { return amount; }  
 public void setAmount(Double amount) { this.amount = amount; }  
 public String getStatus() { return status; }  
 public void setStatus(String status) { this.status = status; }  
}

## loan-service/src/main/java/com/example/loanservice/controller/LoanController.java

package com.example.loanservice.controller;  
  
import com.example.loanservice.model.Loan;  
import org.springframework.beans.factory.annotation.Value;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.\*;  
import org.springframework.web.client.RestTemplate;  
  
import javax.annotation.PostConstruct;  
import java.util.HashMap;  
import java.util.Map;  
  
@RestController  
@RequestMapping("/loans")  
public class LoanController {  
  
 private final Map<Long, Loan> repo = new HashMap<>();  
 private final RestTemplate restTemplate;  
  
 @Value("${account.service.url}")  
 private String accountServiceUrl;  
  
 public LoanController(RestTemplate restTemplate) {  
 this.restTemplate = restTemplate;  
 }  
  
 @PostConstruct  
 public void init() {  
 repo.put(1L, new Loan(1L, 1L, 100000.0, "APPROVED"));  
 repo.put(2L, new Loan(2L, 2L, 250000.0, "PENDING"));  
 }  
  
 @GetMapping  
 public ResponseEntity<Object> getAll() {  
 return ResponseEntity.ok(repo.values());  
 }  
  
 @GetMapping("/{id}")  
 public ResponseEntity<Loan> getById(@PathVariable Long id) {  
 Loan l = repo.get(id);  
 if (l == null) return ResponseEntity.notFound().build();  
 return ResponseEntity.ok(l);  
 }  
  
 @GetMapping("/{id}/account")  
 public ResponseEntity<Object> getLoanAccount(@PathVariable Long id) {  
 Loan loan = repo.get(id);  
 if (loan == null) return ResponseEntity.notFound().build();  
 String url = accountServiceUrl + "/" + loan.getAccountId();  
 Object account = restTemplate.getForObject(url, Object.class);  
 Map<String, Object> resp = new HashMap<>();  
 resp.put("loan", loan);  
 resp.put("account", account);  
 return ResponseEntity.ok(resp);  
 }  
}

## api-gateway/pom.xml

<project xmlns="http://maven.apache.org/POM/4.0.0"   
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0   
 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <groupId>com.example</groupId>  
 <artifactId>api-gateway</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>2.7.12</version>  
 <relativePath/>   
 </parent>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-starter-gateway</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-webflux</artifactId>  
 </dependency>  
 </dependencies>  
 <properties>  
 <java.version>11</java.version>  
 <spring-cloud.version>2021.0.7</spring-cloud.version>  
 </properties>  
 <dependencyManagement>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-dependencies</artifactId>  
 <version>${spring-cloud.version}</version>  
 <type>pom</type>  
 <scope>import</scope>  
 </dependency>  
 </dependencies>  
 </dependencyManagement>  
</project>

## api-gateway/src/main/resources/application.properties

server.port=8080  
spring.application.name=api-gateway

## api-gateway/src/main/java/com/example/gateway/ApiGatewayApplication.java

package com.example.gateway;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class ApiGatewayApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.run(ApiGatewayApplication.class, args);  
 }  
}

## api-gateway/src/main/resources/application.yml

spring:  
 cloud:  
 gateway:  
 routes:  
 - id: account\_route  
 uri: http://localhost:8081  
 predicates:  
 - Path=/accounts/\*\*  
 filters:  
 - StripPrefix=1  
 - id: loan\_route  
 uri: http://localhost:8082  
 predicates:  
 - Path=/loans/\*\*  
 filters:  
 - StripPrefix=1

## How to run each project

1. Create a folder for each project (account-service, loan-service, api-gateway).  
2. Copy the files shown above into the respective paths.  
3. Import each as a Maven project in your IDE or run with command: mvn spring-boot:run  
4. Start account-service (port 8081), then loan-service (port 8082), then api-gateway (port 8080).  
  
Sample requests:  
- GET http://localhost:8081/accounts  
- GET http://localhost:8081/accounts/1  
- GET http://localhost:8082/loans  
- GET http://localhost:8082/loans/1  
- GET http://localhost:8082/loans/1/account (calls account-service)  
- Through gateway: GET http://localhost:8080/accounts/1 or http://localhost:8080/loans/1/account