**Microservices with Spring Boot 3 and Spring Cloud**

**Creating Microservices for account and loan**

**Account project**

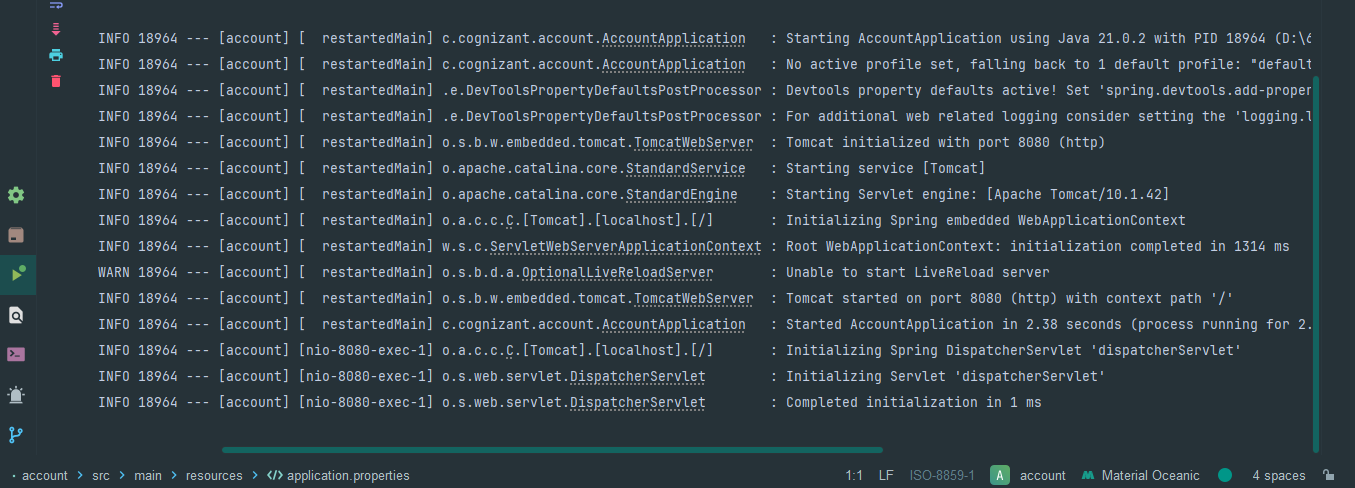
**src/main/java/com/cognizant/account/controller/AccountController.java**

*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;  
  
@RestController  
*public class* AccountController {  
  
 @GetMapping("/accounts/{number}")  
 *public* Account getAccountDetails(@PathVariable String number) {  
 *// Hardcoded response for demo purposes  
 return new* Account(number, "savings", 234343);  
 }  
}  
  
*class* Account {  
 *private* String number;  
 *private* String type;  
 *private double* balance;  
  
 *public* Account(String number, String type, *double* balance) {  
 *this*.number = number;  
 *this*.type = type;  
 *this*.balance = balance;  
 }  
  
 *// Getters and setters  
 public* String getNumber() {  
 *return* number;  
 }  
 *public void* setNumber(String number) {  
 *this*.number = number;  
 }  
 *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;  
 }  
}

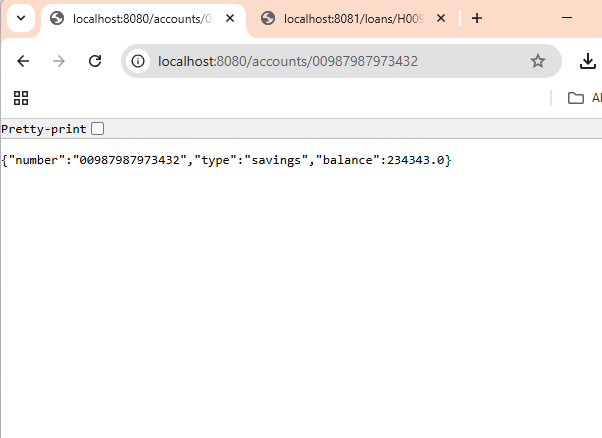
**src/main/java/com/cognizant/account/AccountApplication.java**

*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);  
 }  
  
}

**Output:**

****

**Server Output:**



**Loan Project**

**src/main/java/com/cognizant/loan/controller/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;  
  
@RestController  
*public class* LoanController {  
  
 @GetMapping("/loans/{number}")  
 *public* Loan getLoanDetails(@PathVariable String number) {  
 *// Hardcoded response for demo purposes  
 return new* Loan(number, "car", 400000, 3258, 18);  
 }  
}  
  
*class* Loan {  
 *private* String number;  
 *private* String type;  
 *private double* loan;  
 *private double* emi;  
 *private int* tenure;  
  
 *public* Loan(String number, String type, *double* loan, *double* emi, *int* tenure) {  
 *this*.number = number;  
 *this*.type = type;  
 *this*.loan = loan;  
 *this*.emi = emi;  
 *this*.tenure = tenure;  
 }  
  
 *// Getters and setters  
 public* String getNumber() {  
 *return* number;  
 }  
 *public void* setNumber(String number) {  
 *this*.number = number;  
 }  
 *public* String getType() {  
 *return* type;  
 }  
 *public void* setType(String type) {  
 *this*.type = type;  
 }  
 *public double* getLoan() {  
 *return* loan;  
 }  
 *public void* setLoan(*double* loan) {  
 *this*.loan = loan;  
 }  
 *public double* getEmi() {  
 *return* emi;  
 }  
 *public void* setEmi(*double* emi) {  
 *this*.emi = emi;  
 }  
 *public int* getTenure() {  
 *return* tenure;  
 }  
 *public void* setTenure(*int* tenure) {  
 *this*.tenure = tenure;  
 }  
}

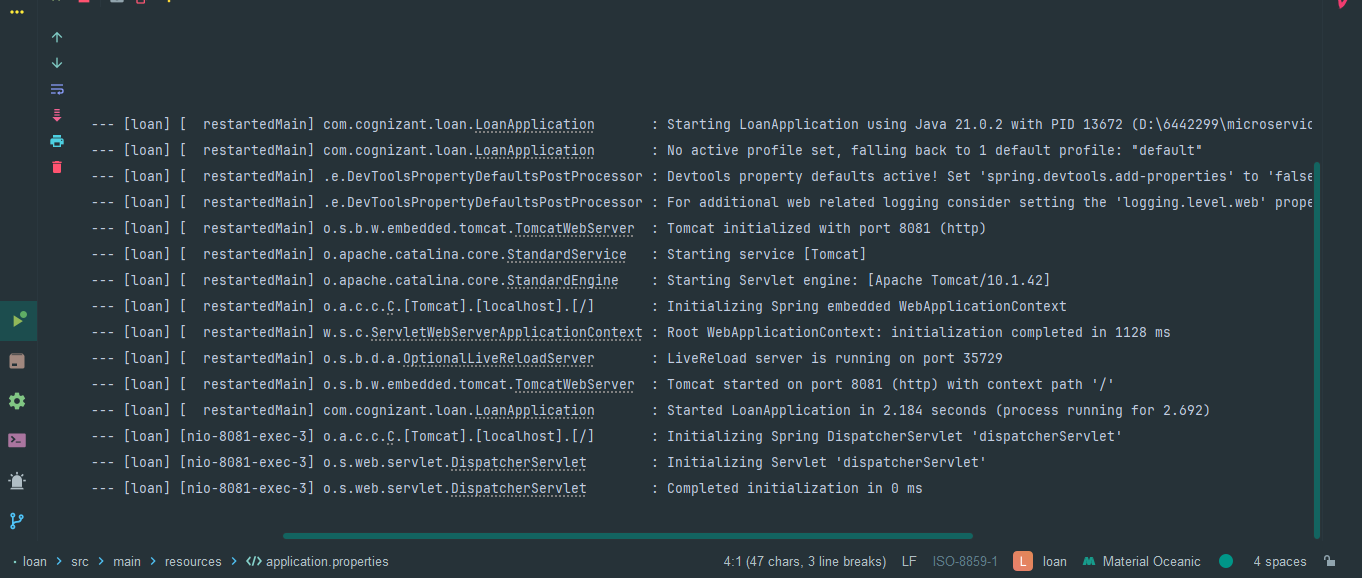
**src/main/java/com/cognizant/loan/LoanApplication.java**

*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);  
 }  
  
}

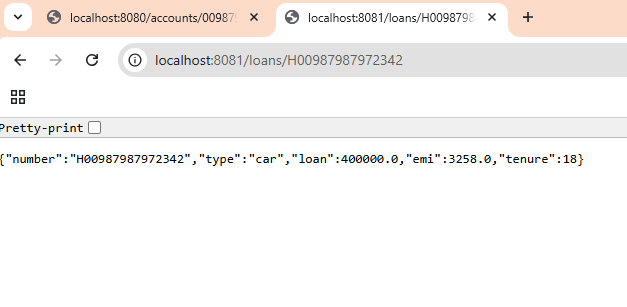
**src/main/resources/application.properties**

*spring.application.name*=loan  
*server.port*=8081

**Terminal Output:**

****

**Server Output:**

****