Automatic evaluation

LoanLender/pom.xml

1 <?xml version="1.0" encoding="UTF-8"?>

2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

3 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

4 <modelVersion>4.0.0</modelVersion>

5 <parent>

6 <groupId>org.springframework.boot</groupId>

7 <artifactId>spring-boot-starter-parent</artifactId>

8 <version>3.0.6</version>

9 <relativePath/> <!-- lookup parent from repository -->

10 </parent>

11 <groupId>com</groupId>

12 <artifactId>LoanLender</artifactId>

13 <version>0.0.1-SNAPSHOT</version>

14 <name>LoanLender</name>

15 <description>LoanLender</description>

16 <properties>

17 <java.version>17</java.version>

18 </properties>

19 <dependencies>

20 <dependency>

21 <groupId>org.springframework.restdocs</groupId>

22 <artifactId>spring-restdocs-mockmvc</artifactId>

23 <scope>test</scope>

24 </dependency>

25 <dependency>

26 <groupId>org.springframework.boot</groupId>

27 <artifactId>spring-boot-starter-web</artifactId>

28 </dependency>

29 <dependency>

30 <groupId>org.springframework.boot</groupId>

31 <artifactId>spring-boot-starter-data-jpa</artifactId>

32 </dependency>

33 <dependency>

34 <groupId>mysql</groupId>

35 <artifactId>mysql-connector-java</artifactId>

36 <version>8.0.26</version>

37 <scope>runtime</scope>

38 </dependency>

39 <dependency>

40 <groupId>org.springframework.boot</groupId>

41 <artifactId>spring-boot-starter-test</artifactId>

42 <scope>test</scope>

43 </dependency>

44 <dependency>

45 <groupId>junit</groupId>

46 <artifactId>junit</artifactId>

47 <version>4.12</version>

48 <scope>test</scope>

49 </dependency>

50 </dependencies>

51 <build>

52 <plugins>

53 <plugin>

54 <groupId>org.springframework.boot</groupId>

55 <artifactId>spring-boot-maven-plugin</artifactId>

56 </plugin>

57 <plugin>

58 <groupId>org.codehaus.mojo</groupId>

59 <artifactId>exec-maven-plugin</artifactId>

60 <version>1.6.0</version>

61 <configuration>

62 <mainClass>com.LoanLenderApplication</mainClass>

63 </configuration>

64 <executions>

65 <execution>

66 <id>run-selenium</id>

67 <phase>integration-test</phase>

68 <goals><goal>java</goal></goals>

69 </execution>

70 </executions>

71 </plugin>

72 </plugins>

73 </build>

74 </project>

75

LoanLender/src/main/java/com/bean/Customer.java

1 *package* com.bean;

2

3 *import* java.util.ArrayList;

4 *import* java.util.List;

5 *import* jakarta.persistence.\*;

6

7

8

9

10 @Entity

11 *public* *class* Customer {

12

13 @Id

14 *private* *long* customerId ;

15 *private* String customerName ;

16 *private* *long* customerContactNumber ;

17

18 @OneToMany(mappedBy = "customer",cascade = CascadeType.ALL)

19 *private* List<Loan> loanList=*new* ArrayList<Loan>();

20

21 *public* *long* getCustomerId() {

22 *return* customerId;

23 }

24 *public* *void* setCustomerId(*long* customerId) {

25 *this*.customerId = customerId;

26 }

27 *public* String getCustomerName() {

28 *return* customerName;

29 }

30 *public* *void* setCustomerName(String customerName) {

31 *this*.customerName = customerName;

32 }

33 *public* *long* getCustomerContactNumber() {

34 *return* customerContactNumber;

35 }

36 *public* *void* setCustomerContactNumber(*long* customerContactNumber) {

37 *this*.customerContactNumber = customerContactNumber;

38 }

39 *public* List<Loan> getLoanList() {

40 *return* loanList;

41 }

42 *public* *void* setLoanList(List<Loan> loanList) {

43 *this*.loanList = loanList;

44 }

45

46

47

48 }

49

LoanLender/src/main/java/com/bean/Loan.java

1 *package* com.bean;

2

3 *import* jakarta.persistence.Entity;

4 *import* jakarta.persistence.Id;

5 *import* jakarta.persistence.ManyToOne;

6 *import* jakarta.persistence.JoinColumn;

7

8 @Entity

9 *public* *class* Loan {

10

11 @Id

12 *private* *long* loanNumber ;

13 *private* String loanType ;

14 *private* *double* loanAmount ;

15 *private* *int* repaymentYears ;

16 *private* *double* monthlyDue ;

17

18 @ManyToOne

19 @JoinColumn(name="customer\_id")

20 *private* Customer customer;

21 *public* *long* getLoanNumber() {

22 *return* loanNumber;

23 }

24 *public* *void* setLoanNumber(*long* loanNumber) {

25 *this*.loanNumber = loanNumber;

26 }

27 *public* String getLoanType() {

28 *return* loanType;

29 }

30 *public* *void* setLoanType(String loanType) {

31 *this*.loanType = loanType;

32 }

33 *public* *double* getLoanAmount() {

34 *return* loanAmount;

35 }

36 *public* *void* setLoanAmount(*double* loanAmount) {

37 *this*.loanAmount = loanAmount;

38 }

39 *public* *int* getRepaymentYears() {

40 *return* repaymentYears;

41 }

42 *public* *void* setRepaymentYears(*int* repaymentYears) {

43 *this*.repaymentYears = repaymentYears;

44 }

45 *public* *double* getMonthlyDue() {

46 *return* monthlyDue;

47 }

48 *public* *void* setMonthlyDue(*double* monthlyDue) {

49 *this*.monthlyDue = monthlyDue;

50 }

51 *public* Customer getCustomer() {

52 *return* customer;

53 }

54 *public* *void* setCustomer(Customer customer) {

55 *this*.customer = customer;

56 }

57

58

59

60

61

62 }

63

LoanLender/src/main/java/com/dao/BankDAO.java

1 *package* com.dao;

2

3

4 *import* org.springframework.stereotype.Repository;

5 *import* java.util.\*;

6 *import* com.bean.Customer;

7 *import* com.bean.Loan;

8 *import* com.repository.CustomerRepository;

9 *import* com.repository.LoanRepository;

10 *import* org.springframework.beans.factory.annotation.Autowired;

11

12 @Repository

13

14 *public* *class* BankDAO {

15

16 @Autowired

17 *private* CustomerRepository customerrepo;

18

19 @Autowired

20 *private* LoanRepository loanrepo;

21

22

23 *public* *void* addCustomer(Customer customer) {

24 customerrepo.save(customer);

25

26

27 }

28

29 *public* *void* allocateLoanToCustomer(*long* customerId, Loan loan) {

30 Customer customer = customerrepo.findById(customerId).orElse(*null*);

31 *if*(customer!=*null*){

32 loan.setCustomer(customer);

33 customer.getLoanList().add(loan);

34 }

35

36

37 }

38

39 *public* List<Loan> loanWithMaxDueMinYears(){

40 *return* loanrepo.loanwithMaxDueMinYears();

41

42 }

43

44 }

45

LoanLender/src/main/java/com/LoanLenderApplication.java

1 *package* com;

2

3 *import* org.springframework.boot.CommandLineRunner;

4 *import* org.springframework.boot.SpringApplication;

5 *import* org.springframework.boot.autoconfigure.SpringBootApplication;

6

7 @SpringBootApplication

8 *public* *class* LoanLenderApplication *implements* CommandLineRunner {

9

10 *public* *static* *void* main(String[] args) {

11

12 SpringApplication app = *new* SpringApplication(LoanLenderApplication.*class*);

13

14 app.run(args);

15 }

16

17 @Override

18 *public* *void* run(String... args) throws Exception {

19 System.out.println("Invoke the methods");

20 }

21 }

LoanLender/src/main/java/com/repository/CustomerRepository.java

1 *package* com.repository;

2

3 *import* com.bean.Customer;

4 *import* org.springframework.data.jpa.repository.JpaRepository;

5

6 *public* *interface* CustomerRepository *extends* JpaRepository<Customer,Long> {

7

8 }

9

LoanLender/src/main/java/com/repository/LoanRepository.java

1 *package* com.repository;

2

3 *import* com.bean.Loan;

4 *import* org.springframework.data.jpa.repository.JpaRepository;

5 *import* org.springframework.data.jpa.repository.Query;

6 *import* java.util.List;

7

8 *public* *interface* LoanRepository *extends* JpaRepository<Loan,Long>{

9

10

11 @Query("select l from Loan l where l.monthlyDue = (select max(l2.monthlyDue) from Loan l2) and l.repaymentYears = (select min(l3.repaymentYears) from Loan l3)")

12

13 List<Loan> loanwithMaxDueMinYears();

14

15 }

16

LoanLender/src/main/resources/application.properties

1 spring.application.name=LoanLender

2 server.port=8090

3 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

4 spring.datasource.url=jdbc:mysql://localhost:3306/test23

5 spring.datasource.username=root

6 spring.datasource.password=@Ibm12345

7

8 spring.jpa.generate-ddl=true

9 spring.jpa.hibernate.ddl-auto=create

10

11 # Show or not log for each SQL query

12 spring.jpa.show-sql=true

13

14 # Allows Hibernate to generate SQL optimized for a particular DBMS

15 # spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

16

Grade

Reviewed on Friday, 23 May 2025, 12:10 AM by Automatic grade  
**Grade** 100 / 100