Xideral

Java Academy

Week 3-Day 4

Java Persistence API

Presented by:

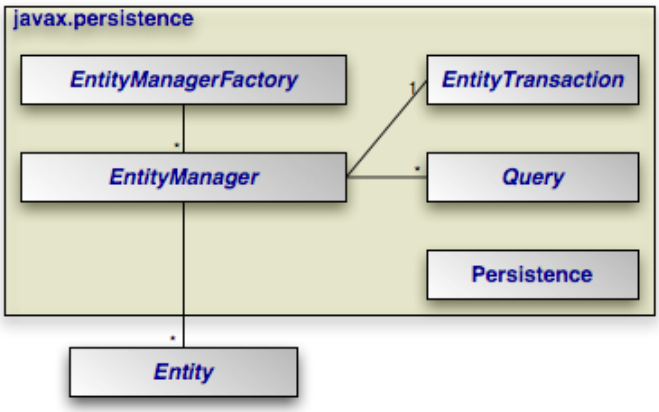
Edgar Itzak Sánchez Rogers

Monterrey Nuevo León México at 30 august 2024

Introduction:

Java Persistence API (JPA) provides a mechanism for managing persistence and relational correlation of objects and functions. The JPA specification defines object-relational mapping internally, rather than relying on vendor-specific mapping implementations. JPA is based on the Java programming model that applies to Java Enterprise Edition (Java EE) environments, but JPA can work within a Java SE environment for testing application functions.

JPA represents a simplification of the persistence programming model. The JPA specification explicitly defines object-relational mapping, rather than relying on vendor-specific mapping implementations.



JPA Demonstration:

The following is an example of using JPA to modify a MYSQL table.

Application.Properties:

```
spring.datasource.url=jdbc:mysql://localhost:3306/restaurant
spring.datasource.username=student
spring.datasource.password=student

#Avoid convert CamelCase to SNAKE_CASE
spring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl
```

MenuItem (Object entity) + constructors, getters and setters:

```java
package com.edgaritzak.JPARestaurant.entity;

import jakarta.persistence.Column;

@Entity
@Table(name="tbl_menu")
public class MenuItem {

    @Id
    @GeneratedValue(strategy= GenerationType.IDENTITY)
    @Column(name="MenuId")
    private int id;

    @Column(name="ItemName")
    private String name;

    @Column(name="Description")
    private String description;

    @Column(name="Category")
    private String category;

    @Column(name="Price")
    private double price;
```

Data Access Object (DAO) interface:

```java
package com.edgaritzak.JPARestaurant.dao;
import java.util.List;

import com.edgaritzak.JPARestaurant.entity.MenuItem;

public interface MenuItemDAO {

    //Select
    List<MenuItem> selectAll();
    List<MenuItem> selectNameLike(String text);
    MenuItem selectById(int id);

    //Insert
    void insertIntoTable(MenuItem item);

    //Update
    void update(MenuItem item);

    //Delete
    void deleteById(int id);

}
```

Data Access Object (DAO) implementation:

```java
package com.edgaritzak.JPARestaurant.dao;

import java.util.List;

@Repository
public class MenuItemDAOImpl implements MenuItemDAO {

    private EntityManager entityMNGR;
    @Autowired
    public MenuItemDAOImpl(EntityManager entityMNGR) {
        this.entityMNGR = entityMNGR;
    }
//
//  SELECT
//
    @Override
    public List<MenuItem> selectAll() {
        TypedQuery<MenuItem> query = entityMNGR.createQuery("FROM MenuItem", MenuItem.class);
        return query.getResultList();
    }

    @Override
    public List<MenuItem> selectNameLike(String text) {
        TypedQuery<MenuItem> query = entityMNGR.createQuery("FROM MenuItem where name like :text",MenuItem.class);
        query.setParameter("text", "%"+text+"%");
        return query.getResultList();
    }
    @Override
    public MenuItem selectById(int id) {
        return entityMNGR.find(MenuItem.class, id);
    }
//
//  INSERT
//
    @Transactional
    @Override
    public void insertIntoTable(MenuItem item) {
        entityMNGR.persist(item);
    }
//
//  UPDATE
//
    @Transactional
    @Override
    public void update(MenuItem item) {
        entityMNGR.merge(item);
    }
//
//  DELETE
//
    @Transactional
    @Override
    public void deleteById(int id) {
        entityMNGR.remove(entityMNGR.find(MenuItem.class, id));
    }

```

Main class:

```java
package com.edgaritzak.JPARestaurant;

import com.edgaritzak.JPARestaurant.dao.MenuItemDAO;
import com.edgaritzak.JPARestaurant.entity.*;

import java.util.List;

import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.Bean;

@SpringBootApplication
public class JPAApplication {

    public static void main(String[] args) {
        SpringApplication.run(JPAApplication.class, args);

    }
    @Bean
    public CommandLineRunner commandLineRunner(MenuItemDAO menuItemDAO){
        return runner -> {
            //Run test methods
            selectById(menuItemDAO);
            selectAll(menuItemDAO);
            selectNameLike(menuItemDAO);
            insertIntoTable(menuItemDAO);
            update(menuItemDAO);
            delete(menuItemDAO);
        };
    }
    //Create test methods
    //SELECT
    static void selectById(MenuItemDAO menuItemDAO) {
        MenuItem item =menuItemDAO.selectById(5);
        System.out.println(item.toString());
    }
    static void selectAll(MenuItemDAO menuItemDAO) {
        List<MenuItem> miList;
        miList =menuItemDAO.selectAll();
        miList.forEach(x-> System.out.println(x.toString()));
    }
    static void selectNameLike(MenuItemDAO menuItemDAO) {
        List<MenuItem> miList;
        miList = menuItemDAO.selectNameLike("Salad");
        System.out.println("------------------------------PRINT SALADS------
        miList.forEach(x-> System.out.println(x.toString()));
    }
    //INSERT
    static void insertIntoTable(MenuItemDAO menuItemDAO) {
        MenuItem newItem = new MenuItem("Lentil Soup", "Hearty and flavorful
        menuItemDAO.insertIntoTable(newItem);
    }
    //UPDATE
    static void update(MenuItemDAO menuItemDAO) {
        MenuItem item = menuItemDAO.selectById(1);
        item.setPrice(9.99);
        menuItemDAO.update(item);
    }
    //DELETE
    static void delete(MenuItemDAO menuItemDAO) {
        menuItemDAO.deleteById(10);
    }
```

Output:

```
MenuItem [id=5, name=Chicken Wings, description=Spicy chicken wings with a side of ranch sauce, category=Appetizer, price=9.99]
MenuItem [id=1, name=Margherita Pizza, description=Classic pizza with tomatoes, mozzarella, and basil, category=Main Course, pr
MenuItem [id=2, name=Spaghetti Carbonara, description=Pasta with eggs, cheese, pancetta, and pepper, category=Main Course, pric
MenuItem [id=3, name=Caesar Salad, description=Fresh Caesar salad with parmesan cheese and croutons, category=Appetizer, price=
MenuItem [id=4, name=Beef Burger, description=Juicy beef burger with lettuce, tomato, and cheese, category=Main Course, price=1
MenuItem [id=5, name=Chicken Wings, description=Spicy chicken wings with a side of ranch sauce, category=Appetizer, price=9.99]
MenuItem [id=6, name=Grilled Salmon, description=Salmon fillet grilled to perfection with lemon and herbs, category=Main Course
MenuItem [id=7, name=Tiramisu, description=Classic Italian dessert with coffee and mascarpone cheese, category=Dessert, price=6
MenuItem [id=8, name=Vegetable Stir-Fry, description=Mixed vegetables stir-fried in a savory sauce, category=Main Course, price=
MenuItem [id=9, name=Greek Salad, description=Crisp salad with feta cheese, olives, and cucumbers, category=Appetizer, price=7.
MenuItem [id=10, name=Chocolate Cake, description=Rich chocolate cake with layers of fudge, category=Dessert, price=5.49]
MenuItem [id=11, name=Pancakes, description=Fluffy pancakes served with maple syrup and butter, category=Breakfast, price=7.49]
MenuItem [id=12, name=Avocado Toast, description=Toast topped with ripe avocado and a sprinkle of salt, category=Breakfast, pri
MenuItem [id=13, name=Margarita Cocktail, description=Refreshing cocktail with tequila, lime, and orange liqueur, category=Drin
MenuItem [id=14, name=Iced Coffee, description=Chilled coffee served with ice and a splash of cream, category=Drink, price=4.99
MenuItem [id=15, name=Cheeseburger, description=Cheeseburger with a beef patty, cheese, pickles, and onions, category=Main Cour
MenuItem [id=16, name=Pumpkin Soup, description=Creamy pumpkin soup with a hint of nutmeg, category=Appetizer, price=7.49]
------------------------------PRINT SALADS------------------------------
MenuItem [id=3, name=Caesar Salad, description=Fresh Caesar salad with parmesan cheese and croutons, category=Appetizer, price=
MenuItem [id=9, name=Greek Salad, description=Crisp salad with feta cheese, olives, and cucumbers, category=Appetizer, price=7.
```

Original table:

MenuId	ItemName	Description	Category	Price
1	Margherita Pizza	Classic pizza with tomatoes, mozzarella, and basil	Main Course	12.99
2	Spaghetti Carbonara	Pasta with eggs, cheese, pancetta, and pepper	Main Course	14.49
3	Caesar Salad	Fresh Caesar salad with parmesan cheese and ...	Appetizer	8.99
4	Beef Burger	Juicy beef burger with lettuce, tomato, and che...	Main Course	11.49
5	Chicken Wings	Spicy chicken wings with a side of ranch sauce	Appetizer	9.99
6	Grilled Salmon	Salmon fillet grilled to perfection with lemon and...	Main Course	16.99
7	Tiramisu	Classic Italian dessert with coffee and mascarp...	Dessert	6.99
8	Vegetable Stir-Fry	Mixed vegetables stir-fried in a savory sauce	Main Course	10.49
9	Greek Salad	Crisp salad with feta cheese, olives, and cucum...	Appetizer	7.99
10	Chocolate Cake	Rich chocolate cake with layers of fudge	Dessert	5.49
11	Pancakes	Fluffy pancakes served with maple syrup and b...	Breakfast	7.49
12	Avocado Toast	Toast topped with ripe avocado and a sprinkle o...	Breakfast	6.99
13	Margarita Cocktail	Refreshing cocktail with tequila, lime, and orang...	Drink	8.49
14	Iced Coffee	Chilled coffee served with ice and a splash of cr...	Drink	4.99
15	Cheeseburger	Cheeseburger with a beef patty, cheese, pickle...	Main Course	12.49
16	Pumpkin Soup	Creamy pumpkin soup with a hint of nutmeg	Appetizer	7.49

Table after operations:

MenuId	ItemName	Description	Category	Price	
1	Margherita Pizza	Classic pizza with tomatoes, mozzarella, and basil	Main Course	9.99	<- price updated
2	Spaghetti Carbonara	Pasta with eggs, cheese, pancetta, and pepper	Main Course	14.49	
3	Caesar Salad	Fresh Caesar salad with parmesan cheese and ...	Appetizer	8.99	
4	Beef Burger	Juicy beef burger with lettuce, tomato, and che...	Main Course	11.49	
5	Chicken Wings	Spicy chicken wings with a side of ranch sauce	Appetizer	9.99	
6	Grilled Salmon	Salmon fillet grilled to perfection with lemon and...	Main Course	16.99	
7	Tiramisu	Classic Italian dessert with coffee and mascarp...	Dessert	6.99	
8	Vegetable Stir-Fry	Mixed vegetables stir-fried in a savory sauce	Main Course	10.49	
9	Greek Salad	Crisp salad with feta cheese, olives, and cucum...	Appetizer	7.99	
11	Pancakes	Fluffy pancakes served with maple syrup and b...	Breakfast	7.49	<- item id=10 deleted
12	Avocado Toast	Toast topped with ripe avocado and a sprinkle o...	Breakfast	6.99	
13	Margarita Cocktail	Refreshing cocktail with tequila, lime, and orang...	Drink	8.49	
14	Iced Coffee	Chilled coffee served with ice and a splash of cr...	Drink	4.99	
15	Cheeseburger	Cheeseburger with a beef patty, cheese, pickle...	Main Course	12.49	
16	Pumpkin Soup	Creamy pumpkin soup with a hint of nutmeg	Appetizer	7.49	
17	Lentil Soup	Hearty and flavorful lentil soup with carrots, cel...	Appetizer	7.99	<- new item

Explanation:

In order to realize this activity, the following steps were followed to carry out this activity:

- Create a database, table and fill it with values
- Add dependencies “spring-boot-starter-data-jpa" and mysql-connector-j in pom.xml file
- Configure connection to database in the application.properties
- Create entity JPA – set object atributes to table colums
- Create a Repository - define a data access component that interacts with a database.
- Create functions to run operations on database
- Write test functions and Run them on main method

Conclusion:

The importance of JPA (Java Persistence API) for manipulating MySQL databases lies in its ability to simplify the interaction between Java applications and relational databases. JPA provides an abstraction layer that allows developers to work with data in the form of Java objects instead of directly handling SQL queries. This not only makes the code more readable and maintainable, but also facilitates the management of CRUD (Create, Read, Update, Delete) operations and the handling of relationships between entities.

References:

[1] Java Persistence API (JPA) - Documentación de IBM

[2] What is Java Persistence API (JPA)? | by Dávid Lakatos | Medium