

1. Creazione del Progetto con Spring Initializer

1. Vai su start.spring.io e configura:

- **Project:** Maven
- **Language:** Java
- **Spring Boot:** 3.4.4

2. Aggiungi le seguenti **Dipendenze**:

- Spring Data JPA (con jakarta persistence)
- PostgreSQL Driver
- Lombok
- Faker (non presente di default: aggiungere la dipendenza)

```
<dependency>
  <groupId>com.github.javafaker</groupId>
  <artifactId>javafaker</artifactId>
  <version>1.0.2</version>
</dependency>
```

3. Scarica ed importa il progetto in IntelliJ.

4. Nel file `application.properties` inserisci le proprietà per il DB usando variabili d'ambiente:

```
spring.datasource.url=${DB_URL}
spring.datasource.username=${DB_USER}
spring.datasource.password=${DB_PWD}
spring.jpa.hibernate.ddl-auto=update
```

Nota: Definisci le variabili d'ambiente (DB_URL, DB_USER, DB_PWD) nelle configurazioni di run di IntelliJ.

2. Struttura ad Albero del Progetto

```
pizza-express/  
└─ src/  
   └─ main/  
      └─ java/  
         └─ it/epicode/pizza_express_2/  
            ├── common/  
            │   ├── config/  
            │   │   └─ FakerConfig.java  
            │   └─ CommonRunner.java  
            ├── menu/  
            │   ├── ElementoMenu.java           // Abstract, SINGLE_TABLE  
            │   ├── Menu.java                  // Entity (non estende ElementoMenu)  
            │   ├── MenuRepository.java  
            │   └─ MenuService.java  
            ├── pizza/  
            │   ├── Pizza.java                 // Estende ElementoMenu  
            │   ├── PizzaRepository.java  
            │   └─ PizzaService.java  
            ├── bevanda/  
            │   ├── Bevanda.java              // Estende ElementoMenu  
            │   ├── BevandaRepository.java  
            │   └─ BevandaService.java  
            ├── topping/  
            │   ├── Topping.java              // Estende ElementoMenu  
            │   ├── ToppingRepository.java  
            │   └─ ToppingService.java  
            ├── tavolo/  
            │   ├── Tavolo.java               // Entity con @Table(name="tavoli")  
            │   ├── TavoloRepository.java  
            │   └─ TavoloService.java  
            └─ ordine/  
                ├── Ordine.java              // Entity con @Table(name="ordini")  
                └─ OrdineRepository.java
```

```
|
└─ resources/
    └─ application.properties
└─ OrdineService.java
```

3. Descrizione dei File

- **Entity:** Rappresentano le tabelle del database; usano le annotazioni di Jakarta Persistence e Lombok (@Data).
 - Le entità *Pizza*, *Bevanda* e *Topping* estendono `ElementoMenu` (con ereditarietà `SINGLE_TABLE`).
 - *Menu*, *Tavolo* e *Ordine* sono entità separate e hanno l'annotazione `@Table` con il nome in forma plurale.
 - **Repository:** Interfacce che estendono `JpaRepository` per gestire le operazioni CRUD.
 - I repository sono usati esclusivamente all'interno dei service.
 - **Service:** Contengono la logica di business e un metodo `createData()` che crea, in un ciclo `for` da 1 a 20, dati falsi con Faker.
 - Il service del menu crea un solo menu che raccoglie tutte le istanze di *ElementiMenu* (cioè pizze, bevande e toppings).
 - Il service dell'ordine usa il menu creato per generare un ordine in cui vengono inseriti, per ogni coperto, gli elementi del menu.
 - **Common:**
 - `FakerConfig` definisce un bean per Faker.
 - `CommonRunner` (`CommandLineRunner`) richiama i metodi `createData()` dei vari service, crea l'ordine e lo stampa invocando il metodo `stampaOrdine()`.
-

4. Codice di Esempio

Di seguito il codice per ciascuna parte.

a) `ElementoMenu` (package `menu`)

```

package it.epicode.pizza_express_2.menu;

import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.Inheritance;
import jakarta.persistence.InheritanceType;
import lombok.Data;

@Data
@Entity
@Inheritance(strategy = InheritanceType.SINGLE_TABLE)
public abstract class ElementoMenu {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String nome;
    private String descrizione;
    private double prezzo;
    private int calorie;

    public String descrizione() {
        return getNome() + " - " + getDescrizione() + " - " + getPrezzo() + "€ - " + getCalorie() + " kcal";
    }
}

```

b) Pizza (package pizza)

```

package it.epicode.pizza_express_2.pizza;

import it.epicode.pizza_express_2.menu.ElementoMenu;
import it.epicode.pizza_express_2.topping.Topping;
import jakarta.persistence.Entity;

```

```

import jakarta.persistence.ManyToMany;
import lombok.Data;
import lombok.NoArgsConstructor;
import java.util.List;

@Data
@NoArgsConstructor
@Entity
public class Pizza extends ElementoMenu {

    @ManyToMany
    private List<Topping> toppings;

    // Il prezzo totale è la somma del prezzo base e dei prezzi dei toppings
    @Override
    public double getPrezzo() {
        double tot = super.getPrezzo();
        if(toppings != null){
            for (Topping t : toppings) {
                tot += t.getPrezzo();
            }
        }
        return tot;
    }

    // Le calorie totali sono la somma delle calorie base e di quelle dei toppings
    @Override
    public int getCalorie() {
        int tot = super.getCalorie();
        if(toppings != null){
            for (Topping t : toppings) {
                tot += t.getCalorie();
            }
        }
        return tot;
    }
}

```

```
}  
}
```

c) Bevanda (package bevanda)

```
package it.epicode.pizza_express_2.bevanda;  
  
import it.epicode.pizza_express_2.menu.ElementoMenu;  
import jakarta.persistence.Entity;  
import lombok.Data;  
import lombok.NoArgsConstructor;  
  
@Data  
@NoArgsConstructor  
@Entity  
public class Bevanda extends ElementoMenu {  
    public Bevanda(String nome, String descrizione, double prezzo, int calorie) {  
        setNome(nome);  
        setDescrizione(descrizione);  
        setPrezzo(prezzo);  
        setCalorie(calorie);  
    }  
}
```

d) Topping (package topping)

```
package it.epicode.pizza_express_2.topping;  
  
import it.epicode.pizza_express_2.menu.ElementoMenu;  
import jakarta.persistence.Entity;  
import lombok.Data;  
import lombok.NoArgsConstructor;
```

```

@Data
@NoArgsConstructor
@Entity
public class Topping extends ElementoMenu {
    public Topping(String nome, String descrizione, double prezzo, int calorie) {
        setNome(nome);
        setDescrizione(descrizione);
        setPrezzo(prezzo);
        setCalorie(calorie);
    }
}

```

e) Menu (package menu)

```

package it.epicode.pizza_express_2.menu;

import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.OneToMany;
import jakarta.persistence.Table;
import lombok.Data;
import java.util.HashSet;
import java.util.Set;

@Data
@Entity
@Table(name = "menus")
public class Menu {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String nome;

```

```

    private String descrizione;

    @OneToMany
    private Set<ElementoMenu> elementi = new HashSet<>();
}

```

f) Tavolo (package tavolo)

```

package it.epicode.pizza_express_2.tavolo;

import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.Table;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@Entity
@Table(name = "tavoli")
public class Tavolo {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private int numeroTavolo;
    private int copertiMax;
    // per semplificare uso string
    private String stato; // Esempio: "LIBERO" o "OCCUPATO"

    public Tavolo(int numeroTavolo, int copertiMax) {
        this.numeroTavolo = numeroTavolo;
        this.copertiMax = copertiMax;
    }
}

```



```

        this.stato = "LIBERO";
    }
}

```

g) Ordine (package ordine)

```

package it.epicode.pizza_express_2.ordine;

import it.epicode.pizza_express_2.menu.ElementoMenu;
import it.epicode.pizza_express_2.tavolo.Tavolo;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.OneToMany;
import jakarta.persistence.ManyToOne;
import jakarta.persistence.Table;
import lombok.Data;
import lombok.NoArgsConstructor;
import java.util.ArrayList;
import java.util.List;

@Data
@NoArgsConstructor
@Entity
@Table(name = "ordini")
public class Ordine {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private int coperti;

    @ManyToOne
    private Tavolo tavolo;
}

```

```

@OneToMany
private List<ElementoMenu> elementi = new ArrayList<>();

public double calcolaTotale() {
    double totale = 0;
    for (ElementoMenu em : elementi) {
        totale += em.getPrezzo();
    }
    return totale;
}

// Stampa l'ordine: tavolo, coperti, lista degli elementi e totale
public void stampaOrdine() {
    System.out.println("Tavolo: " + tavolo.getNumeroTavolo());
    System.out.println("Coperti: " + coperti);
    System.out.println("Elementi Ordinati:");
    for (ElementoMenu em : elementi) {
        System.out.println(em.descrizione());
    }
    System.out.println("Totale Ordine: " + calcolaTotale());
}
}

```

h) Repository

PizzaRepository.java (package pizza)

```

package it.epicode.pizza_express_2.pizza;

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

```

```
public interface PizzaRepository extends JpaRepository<Pizza, Long> {}
```

BevandaRepository.java (package bevanda)

```
package it.epicode.pizza_express_2.bevanda;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface BevandaRepository extends JpaRepository<Bevanda, Long> {}
```

ToppingRepository.java (package topping)

```
package it.epicode.pizza_express_2.topping;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface ToppingRepository extends JpaRepository<Topping, Long> {}
```

MenuRepository.java (package menu)

```
package it.epicode.pizza_express_2.menu;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface MenuRepository extends JpaRepository<Menu, Long> {}
```

TavoloRepository.java (package tavolo)

```
package it.epicode.pizza_express_2.tavolo;

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

public interface TavoloRepository extends JpaRepository<Tavolo, Long> {}
```

OrdineRepository.java (package ordine)

```
package it.epicode.pizza_express_2.ordine;

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

public interface OrdineRepository extends JpaRepository<Ordine, Long> {}
```

i) Service

Esempio di service con metodo `createData()` che usa un ciclo for (da 1 a 20):

PizzaService.java (package pizza)

```
package it.epicode.pizza_express_2.pizza;

import com.github.javafaker.Faker;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import java.util.List;

@Service
public class PizzaService {
    @Autowired
```

```

private PizzaRepository pizzaRepository;
@Autowired
private Faker faker;

public void createData() {
    for (int i = 1; i <= 20; i++) {
        Pizza pizza = new Pizza();
        pizza.setNome("Pizza " + faker.food().dish());
        pizza.setDescrizione("Deliziosa " + faker.lorem().sentence());
        pizza.setPrezzo(faker.number().randomDouble(2, 5, 15));
        pizza.setCalorie(faker.number().numberBetween(700, 1200));
        // Per semplicità non vengono associati toppings
        pizzaRepository.save(pizza);
    }
}

public List<Pizza> findAll() {
    return pizzaRepository.findAll();
}
}

```

BevandaService.java (package bevanda)

```

package it.epicode.pizza_express_2.bevanda;

import com.github.javafaker.Faker;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import java.util.List;

@Service
public class BevandaService {
    @Autowired
    private BevandaRepository bevandaRepository;
    @Autowired

```

```

private Faker faker;

public void createData() {
    for (int i = 1; i <= 20; i++) {
        Bevanda b = new Bevanda();
        b.setNome("Bevanda " + faker.food().ingredient());
        b.setDescrizione("Bibita " + faker.lorem().sentence());
        b.setPrezzo(faker.number().randomDouble(2, 1, 5));
        b.setCalorie(faker.number().numberBetween(50, 200));
        bevandaRepository.save(b);
    }
}

public List<Bevanda> findAll() {
    return bevandaRepository.findAll();
}
}

```

ToppingService.java (package topping)

```

package it.epicode.pizza_express_2.topping;

import com.github.javafaker.Faker;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import java.util.List;

@Service
public class ToppingService {
    @Autowired
    private ToppingRepository toppingRepository;
    @Autowired
    private Faker faker;

    public void createData() {

```

```

        for (int i = 1; i <= 20; i++) {
            Topping t = new Topping();
            t.setNome("Topping " + faker.food().ingredient());
            t.setDescrizione("Extra " + faker.lorem().sentence());
            t.setPrezzo(faker.number().randomDouble(2, 0, 3));
            t.setCalorie(faker.number().numberBetween(10, 100));
            toppingRepository.save(t);
        }
    }

    public List<Topping> findAll() {
        return toppingRepository.findAll();
    }
}

```

MenuService.java (package menu)

```

package it.epicode.pizza_express_2.menu;

import it.epicode.pizza_express_2.bevanda.Bevanda;
import it.epicode.pizza_express_2.bevanda.BevandaService;
import it.epicode.pizza_express_2.pizza.Pizza;
import it.epicode.pizza_express_2.pizza.PizzaService;
import it.epicode.pizza_express_2.topping.Topping;
import it.epicode.pizza_express_2.topping.ToppingService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import java.util.HashSet;
import java.util.Set;

@Service
public class MenuService {
    @Autowired
    private MenuRepository menuRepository;
    @Autowired

```

```

private PizzaService pizzaService;
@Autowired
private BevandaService bevandaService;
@Autowired
private ToppingService toppingService;

public void createData() {
    Set<ElementoMenu> elementi = new HashSet<>();
    for (Pizza p : pizzaService.findAll()) {
        elementi.add(p);
    }
    for (Bevanda b : bevandaService.findAll()) {
        elementi.add(b);
    }
    for (Topping t : toppingService.findAll()) {
        elementi.add(t);
    }
    Menu m = new Menu();
    m.setNome("Menu Speciale");
    m.setDescrizione("Il nostro menu speciale");
    m.setElementi(elementi);
    menuRepository.save(m);
}

public Menu getMenu() {
    return menuRepository.findAll().get(0);
}
}

```

TavoloService.java (package tavolo)

```

package it.epicode.pizza_express_2.tavolo;

import com.github.javafaker.Faker;
import org.springframework.beans.factory.annotation.Autowired;

```



```

import org.springframework.stereotype.Service;
import java.util.List;

@Service
public class TavoloService {
    @Autowired
    private TavoloRepository tavoloRepository;
    @Autowired
    private Faker faker;

    public void createData() {
        for (int i = 1; i <= 20; i++) {
            Tavolo t = new Tavolo();
            t.setNumeroTavolo(i);
            t.setCopertiMax(4);
            t.setStato("LIBERO");
            tavoloRepository.save(t);
        }
    }

    public List<Tavolo> findAll() {
        return tavoloRepository.findAll();
    }
}

```

OrdineService.java (package ordine)

```

package it.epicode.pizza_express_2.ordine;

import it.epicode.pizza_express_2.menu.ElementoMenu;
import it.epicode.pizza_express_2.menu.Menu;
import it.epicode.pizza_express_2.menu.MenuService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import java.util.ArrayList;

```

```

import java.util.List;

@Service
public class OrdineService {
    @Autowired
    private OrdineRepository ordineRepository;
    @Autowired
    private MenuService menuService;

    public void createData() {
        // Crea un ordine utilizzando il menu creato
        Menu menu = menuService getMenu();
        Ordine ordine = new Ordine();
        int coperti = 2;
        ordine.setCoperti(coperti);
        List<ElementoMenu> elementiOrdinati = new ArrayList<>();
        // Per ogni coperto aggiungiamo tutti gli elementi del menu
        for (int i = 0; i < coperti; i++) {
            for (ElementoMenu em : menu.getElementi()) {
                elementiOrdinati.add(em);
            }
        }
        ordine.setElementi(elementiOrdinati);
        // Il tavolo può essere impostato se creato (per semplicità omissso qui)
        ordineRepository.save(ordine);

        // Stampa l'ordine tramite il metodo definito nell'entity
        ordine.stampaOrdine();
    }

    public List<Ordine> findAll() {
        return ordineRepository.findAll();
    }
}

```

j) Common Config e Runner

FakerConfig.java (package common.config)

```
package it.epicode.pizza_express_2.common.config;

import com.github.javafaker.Faker;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

@Configuration
public class FakerConfig {
    @Bean
    public Faker faker() {
        return new Faker();
    }
}
```

CommonRunner.java (package common)

```
package it.epicode.pizza_express_2.common;

import it.epicode.pizza_express_2.bevanda.BevandaService;
import it.epicode.pizza_express_2.menu.MenuService;
import it.epicode.pizza_express_2.ordine.OrdineService;
import it.epicode.pizza_express_2.pizza.PizzaService;
import it.epicode.pizza_express_2.tavolo.TavoloService;
import it.epicode.pizza_express_2.topping.ToppingService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.stereotype.Component;
```

```
@Component
public class CommonRunner implements CommandLineRunner {
    @Autowired
    private PizzaService pizzaService;
    @Autowired
    private BevandaService bevandaService;
    @Autowired
    private ToppingService toppingService;
    @Autowired
    private TavoloService tavoloService;
    @Autowired
    private MenuService menuService;
    @Autowired
    private OrdineService ordineService;

    @Override
    public void run(String... args) throws Exception {
        pizzaService.createData();
        bevandaService.createData();
        toppingService.createData();
        tavoloService.createData();
        menuService.createData();
        ordineService.createData();

        ordineService.stampaOrdine();
    }
}
```