Spring Boot Pizza Order Application - Documentation

Introduction:

This is a Spring Boot application demonstrating the use of dependency injection using the @Autowired annotation for different injection types (Field, Constructor, Setter) in a simple pizza ordering system.

Workflow Explanation:

1. Components:

- VegPizza: A service class marked as a Spring component using the @Component annotation. It provides methods to order and deliver a Veg Pizza.

- PizzaController: A controller class also marked as a component, which has a dependency on VegPizza. The @Autowired annotation is used to inject the VegPizza service into this controller.

2. Application Bootstrapping:

- The Main class is annotated with @SpringBootApplication, which sets up the Spring Boot application context.

- SpringApplication.run(Main.class, args) starts the application and returns the application context.

- Using context.getBean(PizzaController.class), the PizzaController bean is fetched from the application context.

3. Dependency Injection:

- Field Injection: The VegPizza bean is injected directly into the PizzaController class using @Autowired.

- Other commented-out code demonstrates the usage of Constructor Injection and Setter Injection as alternatives.

4. Method Calls:

- The main method invokes OrderPizza() and DeliveredPizza() on the PizzaController bean, which internally calls the corresponding methods of the VegPizza service.

5. Output:

- The output of the application will be:

```
  Ordering pizza: You Ordered Veg Pizza! dude
  Delivering pizza: Your Veg Pizza is Delivered!
```

Conclusion:

This project demonstrates the basic use of Spring Boot's @Autowired annotation for dependency injection. It showcases different injection strategies and how components interact in a simple, modular system.