**Shipment-Item-Scope**

Description

**Objective:**

To work with a Spring Core application using annotations, autowiring, and bean scope concepts.

**Concept Explanation:**

1. **Bean Scope:** Define the lifecycle and visibility of Spring beans, controlling how and when instances are created and destroyed within the application context.

**Concept Implementation:**

1. To ensure that each **Shipment** object has its instance and doesn't interfere with others, the appropriate bean scope needs to be defined.
2. This could be achieved by using the **@Scope**annotation with a value like **"prototype"**, indicating that a new instance should be created for each request for a bean.

**Shipment Details -Scope**

OXI online shopping store wanted to develop a spring application to send the delivery status for their customers. As part of it, every shipment and Item needs to be updated in the system. Partial code for the same is provided. The problem which developers were facing that, every time a Shipment object is updated with the delivery status, it gets reflected for the other Shipment also.

Provide the necessary annotations in the Shipment class so that every time a new Shipment object needs to be created.

**Item** class is already created and provided as a part of code skeleton with the below **private** **attributes** and required **Getter** and **setter** methods.

|  |  |
| --- | --- |
| **Attribute** | **Type** |
| itemName | String |
| price | double |

**Getter and setter** methods for all the above attributes are provided as a part of code skeleton. **Item**class should be registered as a **bean** with appropriate annotations

**Shipment** class is already created and provided as a part of code skeleton with the below **private** **attributes** and required **Getter** and **setter** methods.

|  |  |
| --- | --- |
| **Attribute** | **Type** |
| shipmentId | String |
| Item | Item |
| deliveryStatus | String |

**Getter and setter** methods for all the above attributes are provided as a part of code skeleton. **Shipment**class should be registered as a **bean** with appropriate annotations.

**Driver**class with the below methods are provided as a part of code skeleton.

* **public static void main(String[] args)**-->  Inside the main method get input for Item and Shipment Details with respect to sample input and display the delivery status.

**Design Constraints**

* **Item**class and the **Shipment**class should be present in**com.spring.app** package.
* The class Name/Attribute Name/PackageName should be same as specified in the problem statement. Do not create any new packages.
* Do not create any XML configuration file. You need to use only annotations for configuration.

**Sample Output:**

Shipment Details1

Enter the Item Name

**Headset**

Enter the Item Price

**2000**

Enter the ShipmentId

**CDR112**

Enter the Delivery Status

**pending**

Shipment Details2

Enter the Item Name

**Pendrive**

Enter the Item Price

**500**

Enter the ShipmentId

**CDR322**

Enter the Delivery Status

**Delivered**

Delivery status of shipment ID: CDR112 is pending

Delivery status of shipment ID: CDR322 is Delivered