

Prince Tltiya (202318010)

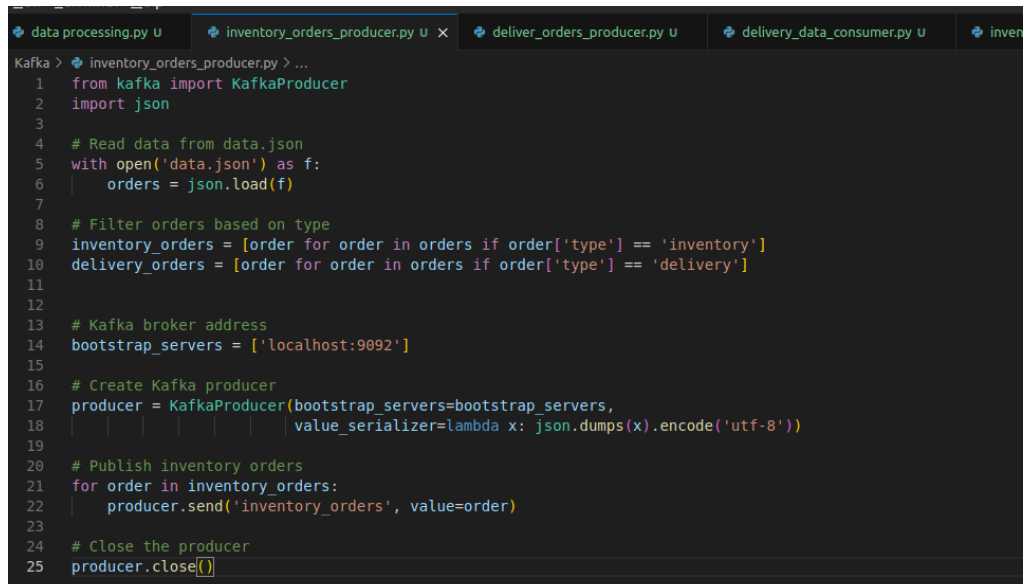
Assignment : Real-Time E-commerce Order Processing System Using Kafka

1. Kafka Producers Implementation:

A client application that transmits records, or messages, to a Kafka cluster is known as a Kafka producer.

Data is published by producers to Kafka topics, which are divided and dispersed among the cluster's Kafka brokers.

Producer 1: Inventory Orders Producer



```
Kafka > inventory_orders_producer.py > ...
1  from kafka import KafkaProducer
2  import json
3
4  # Read data from data.json
5  with open('data.json') as f:
6      orders = json.load(f)
7
8  # Filter orders based on type
9  inventory_orders = [order for order in orders if order['type'] == 'inventory']
10 delivery_orders = [order for order in orders if order['type'] == 'delivery']
11
12
13 # Kafka broker address
14 bootstrap_servers = ['localhost:9092']
15
16 # Create Kafka producer
17 producer = KafkaProducer(bootstrap_servers=bootstrap_servers,
18                           value_serializer=lambda x: json.dumps(x).encode('utf-8'))
19
20 # Publish inventory orders
21 for order in inventory_orders:
22     producer.send('inventory_orders', value=order)
23
24 # Close the producer
25 producer.close()
```

It continuously generates and sends messages related to inventory orders to a Kafka topic (inventory_orders).

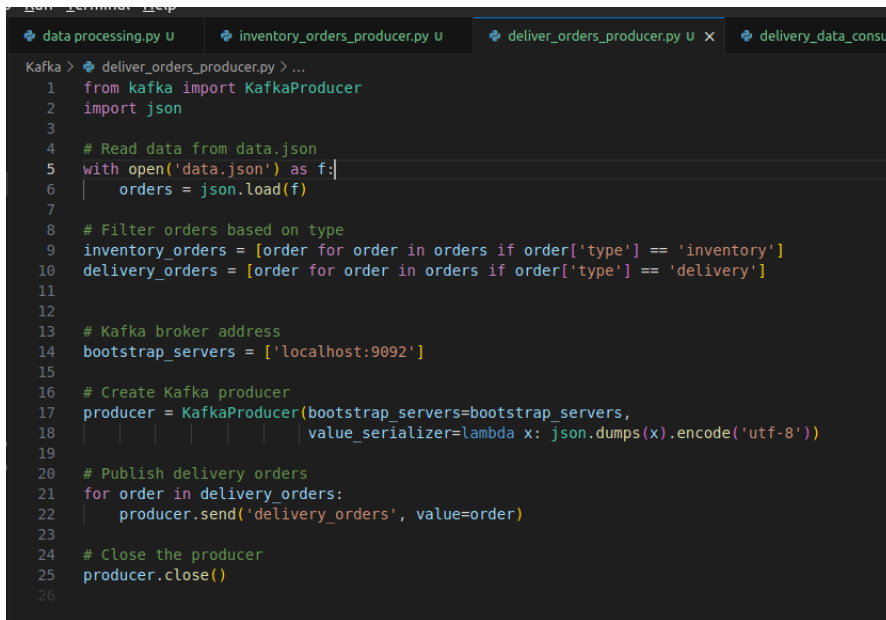
These messages are then consumed by other components or services within the system for further processing, such as updating inventory databases or triggering automated workflows.

It plays an important part in an e-commerce system by allowing events relating to inventory to be sent in real-time via Kafka.

In order to facilitate smooth order processing and fulfillment, it makes inventory data management possible and guarantees timely updates.

Producer 2: Delivery Orders Producer

The "Delivery Orders Producer" is a component responsible for generating and sending messages related to delivery orders to a Kafka topic.



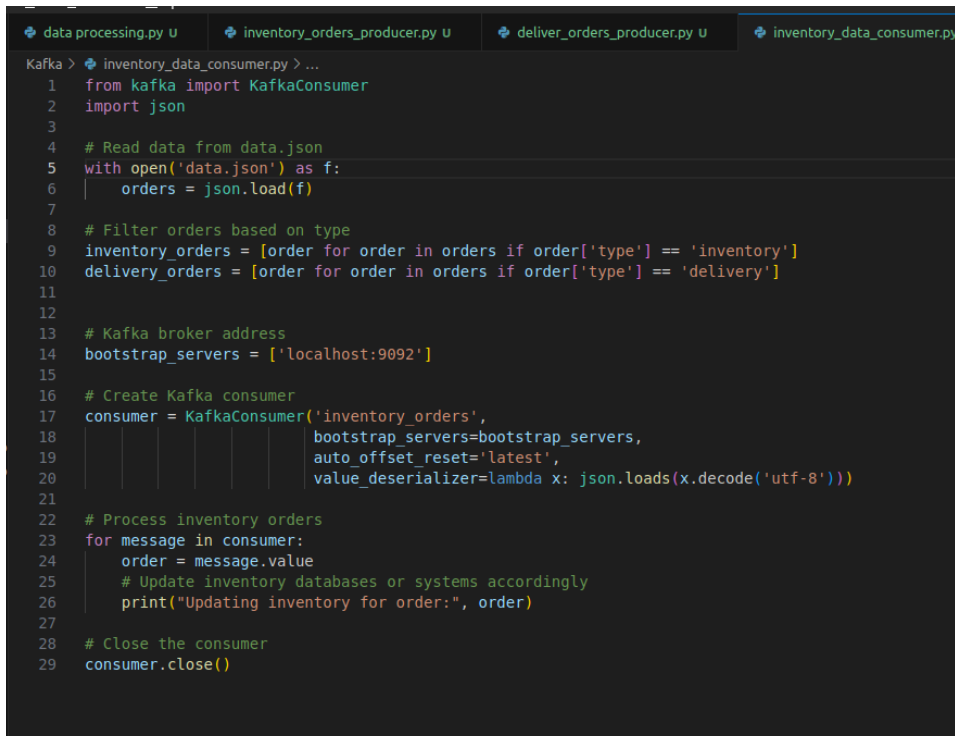
```
Kafka > deliver_orders_producer.py > ...
1  from kafka import KafkaProducer
2  import json
3
4  # Read data from data.json
5  with open('data.json') as f:
6      |   orders = json.load(f)
7
8  # Filter orders based on type
9  inventory_orders = [order for order in orders if order['type'] == 'inventory']
10 delivery_orders = [order for order in orders if order['type'] == 'delivery']
11
12
13 # Kafka broker address
14 bootstrap_servers = ['localhost:9092']
15
16 # Create Kafka producer
17 producer = KafkaProducer(bootstrap_servers=bootstrap_servers,
18 |   |   |   |   |   |   |   value_serializer=lambda x: json.dumps(x).encode('utf-8'))
19
20 # Publish delivery orders
21 for order in delivery_orders:
22     |   producer.send('delivery_orders', value=order)
23
24 # Close the producer
25 producer.close()
26
```

Delivery Orders: These are messages representing orders that require delivery to customers, including details such as order ID, customer information, delivery address, etc. It continuously generates and publishes messages related to delivery orders to a Kafka topic dedicated to delivery order processing. These messages are consumed by downstream systems or services responsible for managing delivery operations, updating order statuses, and coordinating with logistics partners.

An essential part of the architecture of an e-commerce system is the Delivery Orders Producer, which makes sure that events connected to deliveries are promptly communicated via Kafka. Through the effective creation and dissemination of delivery order messages, it facilitates order fulfillment with ease and raises customer satisfaction.

2. Kafka Consumers Implementation

Consumer 1: Inventory Data Consumer



```
Kafka > inventory_data_consumer.py > ...
1 from kafka import KafkaConsumer
2 import json
3
4 # Read data from data.json
5 with open('data.json') as f:
6     orders = json.load(f)
7
8 # Filter orders based on type
9 inventory_orders = [order for order in orders if order['type'] == 'inventory']
10 delivery_orders = [order for order in orders if order['type'] == 'delivery']
11
12
13 # Kafka broker address
14 bootstrap_servers = ['localhost:9092']
15
16 # Create Kafka consumer
17 consumer = KafkaConsumer('inventory_orders',
18                           bootstrap_servers=bootstrap_servers,
19                           auto_offset_reset='latest',
20                           value_deserializer=lambda x: json.loads(x.decode('utf-8')))
21
22 # Process inventory orders
23 for message in consumer:
24     order = message.value
25     # Update inventory databases or systems accordingly
26     print("Updating inventory for order:", order)
27
28 # Close the consumer
29 consumer.close()
```

The Inventory Data Consumer is designed to handle messages related to inventory orders in an e-commerce system.

It listens for inventory order messages from a Kafka topic and processes them to update inventory databases or systems accordingly. It reads orders from a JSON file, filters them based on type, and processes inventory orders received from Kafka. The primary purpose is to integrate Kafka into an e-commerce system for real-time processing of inventory-related events. The code assumes that the consumer will continuously run and process messages as they arrive in the Kafka topic.

The consumer is implemented using the `KafkaConsumer` class from the `kafka-python` library. It specifies the Kafka broker(s) and consumer group, along with other necessary configurations. The consumer subscribes to the 'inventory_orders' topic and continuously polls for messages. Each message is decoded from bytes to a JSON object, and the order details are processed accordingly.

Consumer 2: Delivery Data Consumer

```
data processing.py U  inventory_orders_producer.py U  deliver_orders_producer.py U  inventory_data_consumer.py U  deliver...
Kafka > delivery_data_consumer.py > ...
1 from kafka import KafkaConsumer
2 import json
3
4 # Read data from data.json
5 with open('data.json') as f:
6     orders = json.load(f)
7
8 # Filter orders based on type
9 inventory_orders = [order for order in orders if order['type'] == 'inventory']
10 delivery_orders = [order for order in orders if order['type'] == 'delivery']
11
12
13
14 # Kafka broker address
15 bootstrap_servers = ['localhost:9092']
16
17 # Create Kafka consumer
18 consumer = KafkaConsumer('delivery_orders',
19                           bootstrap_servers=bootstrap_servers,
20                           auto_offset_reset='latest',
21                           value_deserializer=lambda x: json.loads(x.decode('utf-8')))
22
23 # Process delivery orders
24 for message in consumer:
25     order = message.value
26     # Perform actions such as scheduling deliveries, updating delivery status, notifying customers, etc.
27     print(f"Processing delivery order:", order)
28
29 # Close the consumer
30 consumer.close()
31
```

In an e-commerce system, the Delivery Orders Consumer is in charge of managing messages related to delivery orders. It schedules deliveries, notifies customers, and updates delivery statuses while listening for delivery order messages from a Kafka topic.

Similar to the Inventory Data Consumer, this consumer subscribes to a Kafka topic, in this case, 'delivery_orders'.

It continuously polls for new messages from the topic and processes them.

Upon receiving a message, it decodes the JSON-formatted message, extracts the order details, and performs actions such as scheduling deliveries or updating delivery statuses.

An essential part of managing delivery operations in an e-commerce system is the Delivery Orders Consumer.

It aids in real-time updates and task coordination related to deliveries by consuming delivery order messages from the 'delivery_orders' topic. This makes it possible for the e-commerce platform to handle order fulfillment, delivery scheduling, and customer notifications efficiently.

Both consumers play vital roles in the real-time processing and management of inventory and delivery operations within an e-commerce system. They leverage Kafka's messaging capabilities to ensure timely updates and seamless integration with backend systems, contributing to the overall efficiency and reliability of the platform.

Output:

Terminal			
Terminal	Terminal	Terminal	Terminal
Updating inventory for order: {'order_id': '97537', 'product_id': '02758', 'quantity': 35, 'type': 'inventory', 'timestamp': '5/30/2023'}			
Updating inventory for order: {'order_id': '44812', 'product_id': '0', 'quantity': 52, 'type': 'inventory', 'timestamp': '2/15/2024'}			
Updating inventory for order: {'order_id': '781', 'product_id': '0906', 'quantity': 13, 'type': 'inventory', 'timestamp': '10/24/2023'}			
Updating inventory for order: {'order_id': '04', 'product_id': '061', 'quantity': 57, 'type': 'inventory', 'timestamp': '3/28/2024'}			
Updating inventory for order: {'order_id': '8369', 'product_id': '5831', 'quantity': 2, 'type': 'inventory', 'timestamp': '12/13/2023'}			
Updating inventory for order: {'order_id': '76', 'product_id': '59015', 'quantity': 27, 'type': 'inventory', 'timestamp': '10/5/2023'}			
Updating inventory for order: {'order_id': '735', 'product_id': '476', 'quantity': 56, 'type': 'inventory', 'timestamp': '11/6/2023'}			
Updating inventory for order: {'order_id': '2848', 'product_id': '196', 'quantity': 82, 'type': 'inventory', 'timestamp': '12/22/2023'}			
Updating inventory for order: {'order_id': '43', 'product_id': '70662', 'quantity': 65, 'type': 'inventory', 'timestamp': '1/1/2024'}			
Updating inventory for order: {'order_id': '35', 'product_id': '6831', 'quantity': 60, 'type': 'inventory', 'timestamp': '9/25/2023'}			
Updating inventory for order: {'order_id': '1474', 'product_id': '07', 'quantity': 81, 'type': 'inventory', 'timestamp': '1/20/2024'}			
Updating inventory for order: {'order_id': '9', 'product_id': '5', 'quantity': 19, 'type': 'inventory', 'timestamp': '8/31/2023'}			
Updating inventory for order: {'order_id': '42428', 'product_id': '6103', 'quantity': 70, 'type': 'inventory', 'timestamp': '9/25/2023'}			
Updating inventory for order: {'order_id': '5', 'product_id': '28119', 'quantity': 12, 'type': 'inventory', 'timestamp': '8/2/2023'}			
Updating inventory for order: {'order_id': '08665', 'product_id': '0', 'quantity': 14, 'type': 'inventory', 'timestamp': '2/23/2024'}			
Updating inventory for order: {'order_id': '09316', 'product_id': '9', 'quantity': 74, 'type': 'inventory', 'timestamp': '6/26/2023'}			
Updating inventory for order: {'order_id': '2010', 'product_id': '97', 'quantity': 50, 'type': 'inventory', 'timestamp': '8/29/2023'}			
Updating inventory for order: {'order_id': '5132', 'product_id': '3179', 'quantity': 28, 'type': 'inventory', 'timestamp': '8/21/2023'}			
Updating inventory for order: {'order_id': '208', 'product_id': '1', 'quantity': 13, 'type': 'inventory', 'timestamp': '10/21/2023'}			
Updating inventory for order: {'order_id': '182', 'product_id': '5', 'quantity': 91, 'type': 'inventory', 'timestamp': '5/7/2023'}			
Updating inventory for order: {'order_id': '7659', 'product_id': '87250', 'quantity': 65, 'type': 'inventory', 'timestamp': '3/14/2024'}			
Updating inventory for order: {'order_id': '8', 'product_id': '9', 'quantity': 66, 'type': 'inventory', 'timestamp': '12/17/2023'}			
Updating inventory for order: {'order_id': '3', 'product_id': '10', 'quantity': 66, 'type': 'inventory', 'timestamp': '11/27/2023'}			
Updating inventory for order: {'order_id': '8168', 'product_id': '173', 'quantity': 84, 'type': 'inventory', 'timestamp': '1/25/2024'}			
Updating inventory for order: {'order_id': '06', 'product_id': '2', 'quantity': 6, 'type': 'inventory', 'timestamp': '3/3/2024'}			
Updating inventory for order: {'order_id': '12451', 'product_id': '1', 'quantity': 78, 'type': 'inventory', 'timestamp': '4/13/2024'}			
Updating inventory for order: {'order_id': '4856', 'product_id': '9', 'quantity': 60, 'type': 'inventory', 'timestamp': '7/14/2023'}			
Updating inventory for order: {'order_id': '98', 'product_id': '3511', 'quantity': 88, 'type': 'inventory', 'timestamp': '2/21/2024'}			
Updating inventory for order: {'order_id': '49078', 'product_id': '2363', 'quantity': 83, 'type': 'inventory', 'timestamp': '1/13/2024'}			
Updating inventory for order: {'order_id': '1', 'product_id': '34', 'quantity': 35, 'type': 'inventory', 'timestamp': '7/27/2023'}			
Updating inventory for order: {'order_id': '86558', 'product_id': '56093', 'quantity': 68, 'type': 'inventory', 'timestamp': '1/24/2024'}			
Updating inventory for order: {'order_id': '5703', 'product_id': '53941', 'quantity': 20, 'type': 'inventory', 'timestamp': '4/1/2024'}			
Updating inventory for order: {'order_id': '256', 'product_id': '323', 'quantity': 34, 'type': 'inventory', 'timestamp': '8/4/2023'}			
Updating inventory for order: {'order_id': '59955', 'product_id': '40', 'quantity': 3, 'type': 'inventory', 'timestamp': '6/14/2023'}			
Updating inventory for order: {'order_id': '7', 'product_id': '7855', 'quantity': 21, 'type': 'inventory', 'timestamp': '1/27/2024'}			
Updating inventory for order: {'order_id': '1789', 'product_id': '85', 'quantity': 84, 'type': 'inventory', 'timestamp': '12/26/2023'}			
Updating inventory for order: {'order_id': '5', 'product_id': '9', 'quantity': 74, 'type': 'inventory', 'timestamp': '6/14/2023'}			
Updating inventory for order: {'order_id': '858', 'product_id': '336', 'quantity': 76, 'type': 'inventory', 'timestamp': '4/11/2024'}			
Updating inventory for order: {'order_id': '656', 'product_id': '277', 'quantity': 21, 'type': 'inventory', 'timestamp': '1/7/2024'}			
Updating inventory for order: {'order_id': '748', 'product_id': '3271', 'quantity': 82, 'type': 'inventory', 'timestamp': '10/27/2023'}			
Updating inventory for order: {'order_id': '96884', 'product_id': '1', 'quantity': 87, 'type': 'inventory', 'timestamp': '1/15/2024'}			
Updating inventory for order: {'order_id': '74854', 'product_id': '73', 'quantity': 68, 'type': 'inventory', 'timestamp': '2/4/2024'}			
Updating inventory for order: {'order_id': '68102', 'product_id': '2222', 'quantity': 35, 'type': 'inventory', 'timestamp': '3/7/2024'}			
Updating inventory for order: {'order_id': '45190', 'product_id': '4693', 'quantity': 52, 'type': 'inventory', 'timestamp': '12/12/2023'}			
Updating inventory for order: {'order_id': '221', 'product_id': '40769', 'quantity': 42, 'type': 'inventory', 'timestamp': '12/24/2023'}			
Updating inventory for order: {'order_id': '0', 'product_id': '4153', 'quantity': 59, 'type': 'inventory', 'timestamp': '5/30/2023'}			
Updating inventory for order: {'order_id': '3952', 'product_id': '311', 'quantity': 15, 'type': 'inventory', 'timestamp': '2/22/2024'}			
Updating inventory for order: {'order_id': '703', 'product_id': '39', 'quantity': 96, 'type': 'inventory', 'timestamp': '8/24/2023'}			
Updating inventory for order: {'order_id': '9049', 'product_id': '1', 'quantity': 28, 'type': 'inventory', 'timestamp': '5/16/2023'}			
Updating inventory for order: {'order_id': '2351', 'product_id': '951', 'quantity': 20, 'type': 'inventory', 'timestamp': '10/31/2023'}			
Updating inventory for order: {'order_id': '2157', 'product_id': '83', 'quantity': 57, 'type': 'inventory', 'timestamp': '9/3/2023'}			
Updating inventory for order: {'order_id': '797', 'product_id': '02461', 'quantity': 97, 'type': 'inventory', 'timestamp': '8/18/2023'}			

Terminal			
Terminal	Terminal	Terminal	Terminal
Processing delivery order: {'order_id': '7', 'product_id': '6131', 'quantity': 15, 'type': 'delivery', 'timestamp': '3/24/2024'}			
Processing delivery order: {'order_id': '78363', 'product_id': '987', 'quantity': 65, 'type': 'delivery', 'timestamp': '5/20/2023'}			
Processing delivery order: {'order_id': '9113', 'product_id': '1374', 'quantity': 27, 'type': 'delivery', 'timestamp': '1/18/2023'}			
Processing delivery order: {'order_id': '00', 'product_id': '13219', 'quantity': 42, 'type': 'delivery', 'timestamp': '9/22/2023'}			
Processing delivery order: {'order_id': '9', 'product_id': '49466', 'quantity': 42, 'type': 'delivery', 'timestamp': '10/14/2023'}			
Processing delivery order: {'order_id': '51540', 'product_id': '8', 'quantity': 17, 'type': 'delivery', 'timestamp': '11/14/2023'}			
Processing delivery order: {'order_id': '234', 'product_id': '33785', 'quantity': 41, 'type': 'delivery', 'timestamp': '6/14/2023'}			
Processing delivery order: {'order_id': '0030', 'product_id': '2', 'quantity': 99, 'type': 'delivery', 'timestamp': '9/8/2023'}			
Processing delivery order: {'order_id': '6', 'product_id': '855', 'quantity': 52, 'type': 'delivery', 'timestamp': '7/13/2023'}			
Processing delivery order: {'order_id': '7', 'product_id': '87852', 'quantity': 32, 'type': 'delivery', 'timestamp': '2/24/2024'}			
Processing delivery order: {'order_id': '32', 'product_id': '5', 'quantity': 50, 'type': 'delivery', 'timestamp': '9/23/2023'}			
Processing delivery order: {'order_id': '7', 'product_id': '65', 'quantity': 28, 'type': 'delivery', 'timestamp': '11/14/2023'}			
Processing delivery order: {'order_id': '763', 'product_id': '168', 'quantity': 73, 'type': 'delivery', 'timestamp': '3/18/2024'}			
Processing delivery order: {'order_id': '3399', 'product_id': '28596', 'quantity': 92, 'type': 'delivery', 'timestamp': '11/20/2023'}			
Processing delivery order: {'order_id': '6951', 'product_id': '9', 'quantity': 95, 'type': 'delivery', 'timestamp': '8/17/2023'}			
Processing delivery order: {'order_id': '7', 'product_id': '3383', 'quantity': 22, 'type': 'delivery', 'timestamp': '11/5/2023'}			
Processing delivery order: {'order_id': '95538', 'product_id': '6', 'quantity': 85, 'type': 'delivery', 'timestamp': '7/18/2023'}			
Processing delivery order: {'order_id': '1681', 'product_id': '15', 'quantity': 28, 'type': 'delivery', 'timestamp': '6/24/2024'}			
Processing delivery order: {'order_id': '437', 'product_id': '31363', 'quantity': 17, 'type': 'delivery', 'timestamp': '8/16/2023'}			
Processing delivery order: {'order_id': '838', 'product_id': '5', 'quantity': 41, 'type': 'delivery', 'timestamp': '11/12/2023'}			
Processing delivery order: {'order_id': '40', 'product_id': '571', 'quantity': 62, 'type': 'delivery', 'timestamp': '10/12/2023'}			
Processing delivery order: {'order_id': '13977', 'product_id': '8', 'quantity': 13, 'type': 'delivery', 'timestamp': '12/9/2023'}			
Processing delivery order: {'order_id': '28818', 'product_id': '1970', 'quantity': 9, 'type': 'delivery', 'timestamp': '11/15/2023'}			
Processing delivery order: {'order_id': '92', 'product_id': '88688', 'quantity': 90, 'type': 'delivery', 'timestamp': '5/27/2023'}			
Processing delivery order: {'order_id': '36541', 'product_id': '8447', 'quantity': 48, 'type': 'delivery', 'timestamp': '12/31/2023'}			
Processing delivery order: {'order_id': '34784', 'product_id': '0859', 'quantity': 32, 'type': 'delivery', 'timestamp': '6/27/2023'}			
Processing delivery order: {'order_id': '7405', 'product_id': '823', 'quantity': 88, 'type': 'delivery', 'timestamp': '11/22/2023'}			
Processing delivery order: {'order_id': '6817', 'product_id': '2522', 'quantity': 96, 'type': 'delivery', 'timestamp': '12/3/2023'}			
Processing delivery order: {'order_id': '925', 'product_id': '21', 'quantity': 15, 'type': 'delivery', 'timestamp': '8/29/2023'}			
Processing delivery order: {'order_id': '9003', 'product_id': '1', 'quantity': 73, 'type': 'delivery', 'timestamp': '2/3/2024'}			
Processing delivery order: {'order_id': '49', 'product_id': '3697', 'quantity': 55, 'type': 'delivery', 'timestamp': '9/15/2023'}			
Processing delivery order: {'order_id': '76370', 'product_id': '17', 'quantity': 41, 'type': 'delivery', 'timestamp': '11/25/2023'}			
Processing delivery order: {'order_id': '8', 'product_id': '5319', 'quantity': 60, 'type': 'delivery', 'timestamp': '12/20/2023'}			
Processing delivery order: {'order_id': '1', 'product_id': '99569', 'quantity': 18, 'type': 'delivery', 'timestamp': '1/1/2024'}			
Processing delivery order: {'order_id': '79', 'product_id': '90', 'quantity': 52, 'type': 'delivery', 'timestamp': '10/4/2023'}			
Processing delivery order: {'order_id': '9', 'product_id': '4998', 'quantity': 26, 'type': 'delivery', 'timestamp': '12/29/2023'}			
Processing delivery order: {'order_id': '9860', 'product_id': '4', 'quantity': 40, 'type': 'delivery', 'timestamp': '6/28/2023'}			
Processing delivery order: {'order_id': '8692', 'product_id': '3138', 'quantity': 5, 'type': 'delivery', 'timestamp': '2/2/2024'}			
Processing delivery order: {'order_id': '4373', 'product_id': '482', 'quantity': 28, 'type': 'delivery', 'timestamp': '11/23/2023'}			
Processing delivery order: {'order_id': '32', 'product_id': '401', 'quantity': 19, 'type': 'delivery', 'timestamp': '12/5/2023'}			
Processing delivery order: {'order_id': '09614', 'product_id': '72424', 'quantity': 55, 'type': 'delivery', 'timestamp': '9/15/2023'}			
Processing delivery order: {'order_id': '672', 'product_id': '4', 'quantity': 12, 'type': 'delivery', 'timestamp': '8/9/2023'}			
Processing delivery order: {'order_id': '1', 'product_id': '41553', 'quantity': 21, 'type': 'delivery', 'timestamp': '5/31/2023'}			
Processing delivery order: {'order_id': '3848', 'product_id': '9', 'quantity': 25, 'type': 'delivery', 'timestamp': '10/1/2023'}			
Processing delivery order: {'order_id': '093', 'product_id': '75', 'quantity': 40, 'type': 'delivery', 'timestamp': '9/18/2023'}			
Processing delivery order: {'order_id': '9', 'product_id': '481', 'quantity': 89, 'type': 'delivery', 'timestamp': '4/25/2024'}			
Processing delivery order: {'order_id': '9201', 'product_id': '96611', 'quantity': 1, 'type': 'delivery', 'timestamp': '8/31/2023'}			
Processing delivery order: {'order_id': '06', 'product_id': '89462', 'quantity': 66, 'type': 'delivery', 'timestamp': '1/17/2024'}			
Processing delivery order: {'order_id': '86604', 'product_id': '65', 'quantity': 54, 'type': 'delivery', 'timestamp': '10/4/2023'}			
Processing delivery order: {'order_id': '7509', 'product_id': '7253', 'quantity': 3, 'type': 'delivery', 'timestamp': '12/2/2023'}			
Processing delivery order: {'order_id': '87', 'product_id': '05', 'quantity': 85, 'type': 'delivery', 'timestamp': '11/2/2023'}			