



ProductModel*

product_id: str.
name: str
price: float
quantity: int
delivery_date: int

CartModel*

id: PyObjectId
customer_id: str
products: List[ProductModel]

CatalogStore: Add products to Catalog
MyShoppingCart: Add products to the shopping cart of the customer

MsgProcessor: Checkout the cart of the customer, ask if exist stock, if money is enough. If it's ok, update the customer wallet, and update the order.status==success. Or failed if errors. Then use **MessageSender** to send rabbitmq messages as failed_orders or success_orders

Fake classes are for testing purpose

SHIPPING MICROSERVICE

DBConnector<<abstract class>>

db_conn

+insert_one(collection,data)
+get_one(collection,condition)
+update_one(collection,condition,data)

ShippingProcessor

+channel: str
+Notifier: AbstractNotifier
db: DBConnector
+calculate_dates(start:str,end:str)
+process_success_checkout(data:dict,card:AbstractCart)
+process_ordered_orders(data:dict)
+process_warehouse_orders(data:dict)
+process_packaged_orders(data:dict)
+process_carrier_picked_up_orders(data:dict)
+process_out_of_delivery_orders(data:dict)

MongoDBConnector

db_conn

+insert_one(collection,data)
+get_one(collection,condition)
+update_one(collection,condition,data)

FakeMongoDBConnector

db_conn

+insert_one(collection,data)
+get_one(collection,condition)
+update_one(collection,condition,data)

AbstractNotifier

+channel: str
+__init__(channel:str)
+send_to_queue(queue,data_json:dict)

Notifier

+channel: str
+__init__(channel:str)
+send_to_queue(queue,data_json:dict)

FakeNotifier

+channel: str
+__init__(channel:str)
+send_to_queue(queue,data_json:dict)

AbstractCart

+requestCart()

MyShippingCart

+requestCart()

FakeShippingCart

+requestCart()

ShippingProcessor receive the order as "success". It means no problem with stock and order payment. Then send the data message (card_id), between the different stages in the order processing.

Notifier classes send message to rabbitmq service.

Clearly here it's possible a refactor because I'm using many classes to do similar things as "Send a message by rabbitmq"