

A simple Order Management System

Spring boot application with the below requirements :

- Creation of Customers
- Customers can create orders. For simplicity, once an order is created, that's final. There is no order status like created, payment done, completed etc. It is assumed that customer has already made the payment while creating the order.
- Customers are categorized as regular, gold, platinum
- By default, a customer is regular.
 - Customer is promoted to gold when he has placed 10 orders
 - Customer is promoted to platinum when he has placed 20 orders
- Gold = 10% discount, platinum = 20% discount
- When a customer creates an order, if he is a gold customer, automatically 10% discount is applied on the order. 20% for platinum customers.
- Since it is assumed that customer has already made the full payment during creation of the order, this discount information has to be kept safe by the application. We need to keep track of how much discount is given to which customer and for which order, so that customers can claim money back later.
- It is not mandatory to implement any other entities which are not mentioned here, like products or payments etc.

Hints

- Identify the entities
- Think of the REST APIs needed.
- Identify things which can be done asynchronously

Expectations

- Clean, standard REST APIs
- Well defined entities. You can use any database of your choice
- Junit test cases

Nice to have

- Cron job to send emails to customer when they are approaching the gold/platinum barriers. For e.g., email to customer when a customer has 9 orders : "You have placed 9 orders with us. Buy one more stuff and you will be promoted to Gold customer and enjoy 10% discounts!"

You dont need to actually send the emails. It can be a dummy method like this :

```
void sendMail() {  
    System.out.println("Sent mail to customer");  
}
```