B.Sc. (Information Technology), School of Information Technology, KMUTT
INT204 Server side programming II
3rd Examination 2/2023

ภาคปฏิบัติ (2 Hours) 13:00 – 16:00

*** เมื่อ {xxx} คือ เลขท้าย 3 ตัวรหัสนักศึกษาของท่าน ***

ข้อกำหนด

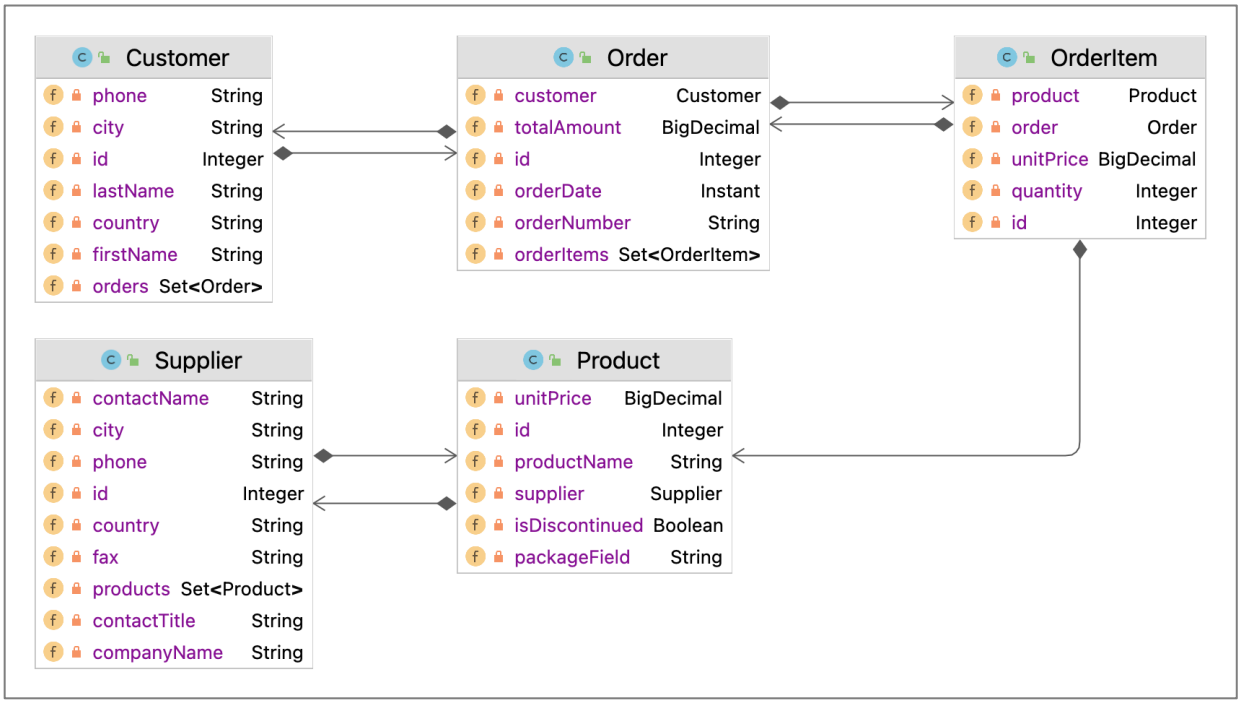
- ให้นักศึกษา สร้างและตั้งชื่อ project ตามรูปแบบดังนี้ **final_{xxx}**
- นักศึกษาต้องใช้ URI ของแต่ข้อตามที่ข้อสอบระบุ ซึ่งต้องเริ่มต้นด้วย รหัสนักศึกษาเสมอ เช่น
  http://localhost:8080/**รหัสนักศึกษาของท่าน**/customers
- บรรทัดแรกของโปรแกรม ต้องมีรูปแบบ ดังนี้ // ชื่อ-สกุล (รหัสนักศึกษา)
- **ผลลัพธ์ที่ capture จาก Postman ต้องเป็นแบบ full windows เท่านั้น**
- **สร้าง Entity, Repository และ DTO เท่าที่จำเป็นต้องใช้**

*** การพิจารณาคะแนน จะดูจาก โปรแกรมทำงานได้ถูกต้อง และ การออกแบบตาม Layer Architectures ***
*** ตรวจและให้คะแนน เฉพาะข้อที่มี **ผลลัพธ์ในไฟล์ PDF** และ ปฏิบัติตาม **ข้อกำหนด** ของข้อสอบ เท่านั้น ***

สิ่งที่ต้องส่ง (ZIP รวมกันตามรูปแบบที่ระบุใน LEB2)

1) Source Program (Controller, Service, Repository, Entity, DTO) ของแต่ละข้อ + ผล Test ส่ง ในรูปแบบไฟล์ PDF
   - ผลการทดสอบของแต่ละข้อเพื่อแสดงให้เห็นว่า น.ศ. สามารถทำโปรแกรมได้ตามความต้องการของโจทย์
   - ส่วนของโปรแกรม ที่เกี่ยวกับโจทย์ข้อนั้นๆ เพื่อแสดงให้เห็นว่า น.ศ. เข้าใจ Flow ของโปรแกรมที่ น.ศ. เขียน
2) Export Project (ในรูปแบบ zip)

>>> การ Hard Coding เพื่อให้ได้ผลลัพธ์ โดยไม่ได้มีการประมวลผลตามที่โจทย์กำหนด ถือว่าเป็นการทุจริต <<<



รูปที่ 1 Class Diagram

การแก้ไข Postman Test Script โดยไม่ได้รับอนุญาต เป็นการทุจริตร้ายแรง

ทุก ๆ end-point ต้องเริ่มต้นด้วย รหัสนักศึกษาเสมอ เช่น

http://localhost:8080/รหัสนักศึกษาของท่าน/customers

จาก Class Diagram ในรูปที่ 1

ให้นักศึกษา สร้าง Spring Boot Project เพื่อให้บริการ API ตามข้อกำหนดดังต่อไปนี้ **(30 คะแนน)**

1) Get order by id

`GET /653000001/orders/{id}`

JSON ของ Response ที่ส่งกลับ ต้องมีรูปแบบ ตาม ตัวอย่าง ด้านล่าง **(10 คะแนน)**

```
{
  "id": 9,
  "orderNumber": "542386",
  "orderDate": "2012-07-15T00:00:00Z",
  "customer": {
    "id": 88,
    "name": "Paula Parente"
  },
  "totalAmount": 517.8,
  "orderItems": [
    {
      "id": 26,
      "productName": "Original Frankfurter grüne Soße",
      "quantity": 12,
      "unitPrice": 10.4
    },
    {
      "id": 25,
      "productName": "Perth Pasties",
      "quantity": 15,
      "unitPrice": 26.2
    }
  ]
}
```

ถ้าไม่เจอข้อมูลของ id ที่ระบุ ให้ return status **404** พร้อม JSON Response ตามตัวอย่าง ด้านล่าง **(5 คะแนน)**

```
{
  "status": 404,
  "message": "Order id '999' does not exist",
  "instance": "uri=/653000001/orders/999"
}
```

---

2) Create new order

`POST /653000001/orders`

- มี JSON input format **ตามรูปด้านล่าง**

```
{
  "orderNumber": "5785277347",
  "customer": {
    "id": 0
  },
  "totalAmount": 0,
  "orderItems": [
    {
      "quantity": 0,
      "unitPrice": 0,
      "product": {
        "id": 0
      }
    }
  ]
}
```

ตัวอย่างข้อมูล

```
{
  "orderNumber": "9991110",
  "customer": {"id" : 5},
  "totalAmount": 505.00,
  "orderItems": [
    {
        "product" : {"id" : 10},
        "unitPrice" : 2.50,
        "quantity" : 2
    },
    {
        "product" : {"id" : 12},
        "unitPrice" :100.00,
        "quantity" : 5
    }
  ]
}
```

2.1) Validation

Constrains: (ถ้าตรวจสอบแล้ว invalided ให้ return http-status **400)**

- orderNumber: Not Null, ต้องเป็นตัวเลขเท่านั้น, ความยาว 6-10 digits
- customer: Not Null, ไม่ตรวจสอบ customer id (ต้องระบุให้ถูกต้อง)
- totalAmount: Not Null, ต้องมีค่า เท่ากับ **ผลรวม** ของ (unitPrice x quantity) ใน orderItems
- orderItems: Not Null, ต้องมีอย่างน้อย 1 รายการ แต่ไม่เกิน 5 รายการ, ไม่ตรวจสอบ product id (ต้องระบุให้ถูกต้อง)

ตัวอย่าง response เมื่อข้อมูล ไม่ถูกต้อง (5 คะแนน)

```
{
    "status": 400,
    "message": "Validation error. Check 'errors' field for details. addOrderDto",
    "instance": "uri=/653000001/orders",
    "errors": [
        {
            "field": "orderNumber",
            "message": "order number must be digits only"
        },
        {
            "field": "orderNumber",
            "message": "size must be between 6 and 10"
        },
        {
            "field": "orderItems",
            "message": "size must be between 1 and 5"
        },
        {
            "field": "customer",
            "message": "must not be null"
        }
    ]
}
```

ตัวอย่าง response กรณี totalAmount มีค่า **ไม่เท่ากับ** ผลรวมของ (unitPrice x quantity) ใน orderItems **(2 คะแนน)**

```
{
  "orderNumber": "0123456789",
  "customer": {"id" : 5},
  "totalAmount": 500,
  "orderItems": [
    {
        "product" : {"id" : 10},
        "unitPrice" : 2.50,
        "quantity" : 2
    },
    {
        "product" : {"id" : 12},
        "unitPrice" :100,
        "quantity" : 5
    }
  ]
}
```

```
{
    "status": 400,
    "message": "Invalid total amount",
    "instance": "uri=/653000001/orders"
}
```

ตัวอย่าง response กรณี orderNumber ซ้ำ (1 คะแนน)

```
{
    "status": 400,
    "message": "Duplicate entry '0123456789' for key 'orders.orderNumber'",
    "instance": "uri=/653000001/orders"
}
```

2.2) **กรณีป้อน ข้อมูล ถูกต้อง และสามารถ บันทึก ข้อมูล Order ได้ เมื่อนำ Order id ไป Get Order by Id ด้วย URI จากข้อที่ 1) จะได้ผลลัพธ์ ดังตัวอย่างด้านล่าง (7 คะแนน)**

Response จากการ Post เพื่อสร้าง Order ใหม่ (ในการทดสอบใช้เฉพาะ id เท่านั้น)

```
{
    "id": 842,
    "orderNumber": "0123456789",
    "orderDate": null,
    "customer": {
        "id": 5,
        "name": "null null"
    },
    "totalAmount": 505,
    "orderItems": null
}
```

Response เมื่อ Get Order by id

ตรวจสอบข้อมูลอื่นๆ โดยการใช้ URI ในข้อ 1 **(ต้องทำข้อ 1 สำเร็จ)**

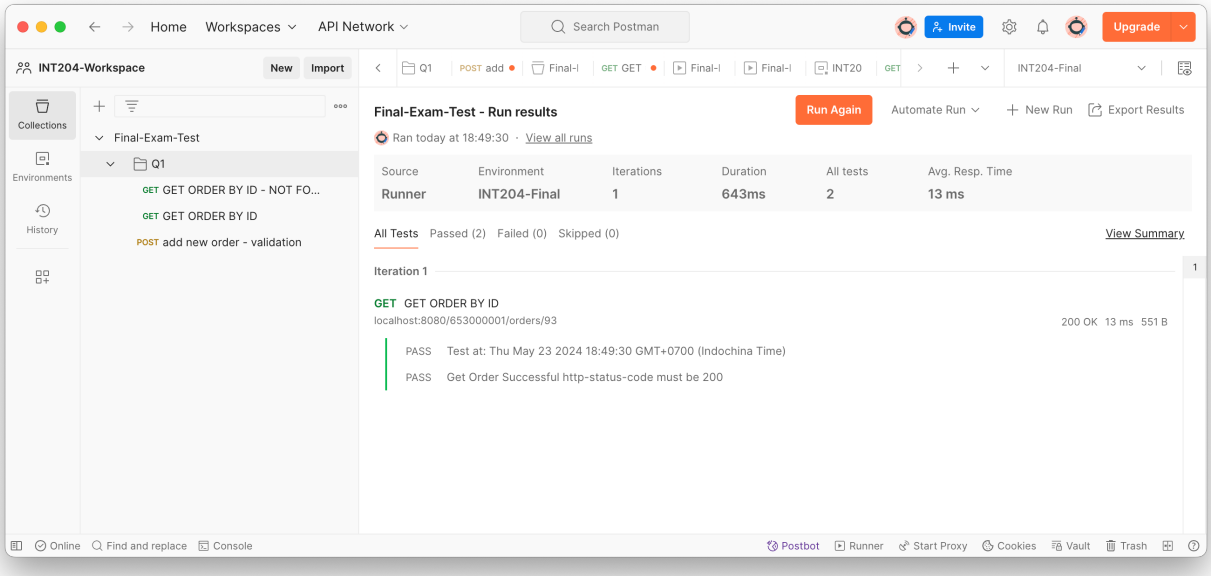
```
{
    "id": 842,
    "orderNumber": "0123456789",
    "orderDate": "2024-05-22T00:00:00Z",
    "customer": {
        "id": 5,
        "name": "Christina Berglund"
    },
    "totalAmount": 505.00,
    "orderItems": [
        {
            "id": 2161,
            "productName": "Queso Manchego La Pastora",
            "quantity": 5,
            "unitPrice": 100.00
        },
        {
            "id": 2162,
            "productName": "Ikura",
            "quantity": 2,
            "unitPrice": 2.50
        }
    ]
}
```

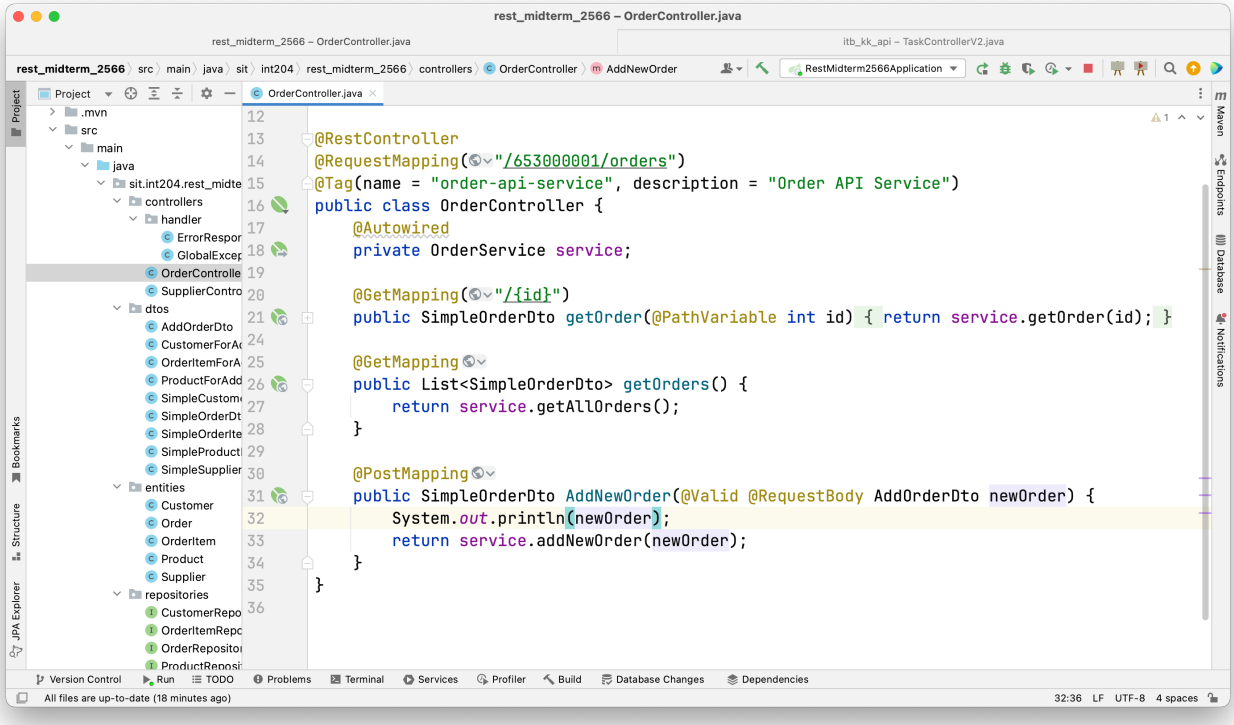
กรณีที่ทดสอบ ไม่ผ่าน นักศึกษา สามารถ ดูค่าตัวแปรต่างๆ ที่ใช้ ในการทดสอบได้จาก **console** ของ **postman**

B.Sc. (Information Technology), School of Information Technology, KMUTT
INT204 Server side programming II
3nd Examination 2/2023

ตัวอย่าง การ **capture** ผลการทดสอบ

- ต้องเป็นผลจากการ **Run (ไม่ใช่ send request)**
- ต้อง **capture** ทั้ง **windows (**ไม่ตัดมาเฉพาะ บางส่วน)





การแก้ไข Postman Test Script โดยไม่ได้รับอนุญาต เป็นการทุจริตร้ายแรง