# WebPos 实验报告

李岳 201840278

计算机科学与技术系

2022 年 7 月 7 日

## 摘要

实现了响应式的 WebPos，是对前面几次作业的综合。

# 一　结构

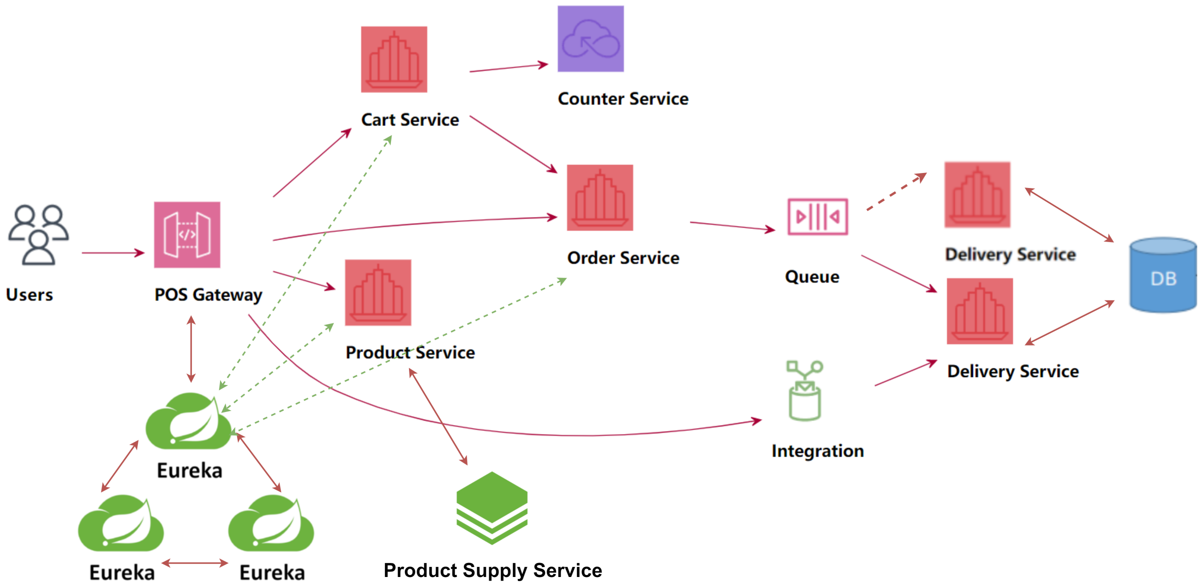


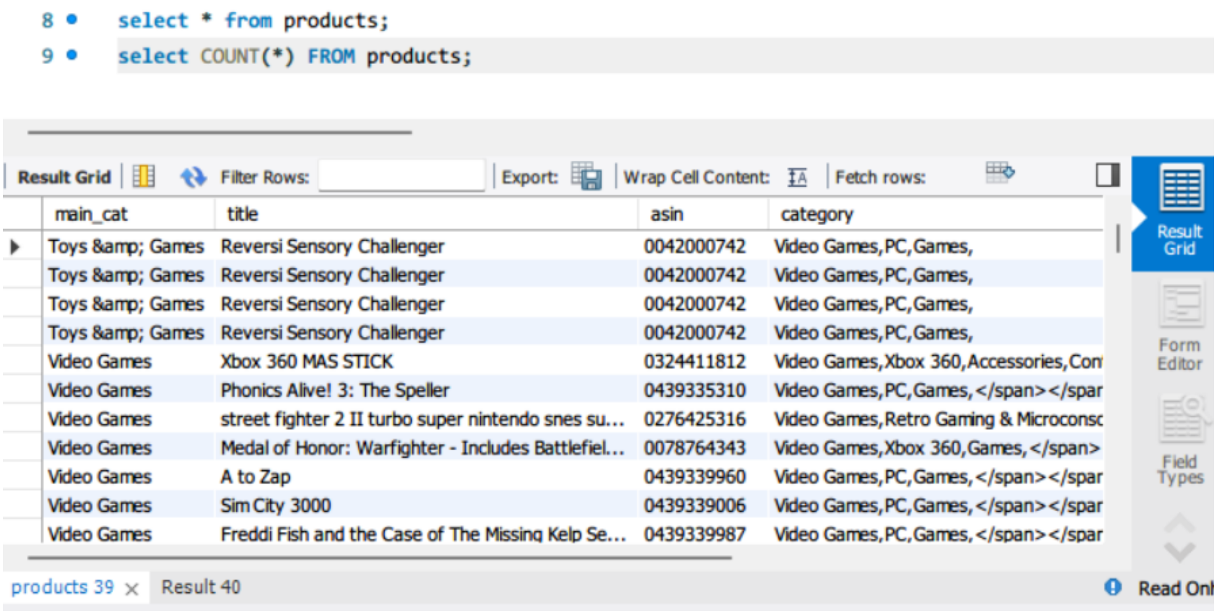
图 1.1: 系统结构

整体为微服务架构，API gateway 进行路由管理，实现对微服务部件的访问。Eureka Server 对微服务部件进行注册。Product Supply Service 实现大批量数据处理导入。Product Service 实现商品相关功能。Cart Service 实现基本的购物车增删查改功能。Order Service 对 Checkout 请求进行处理，产生订单通过 RabbitMQ 传至 Delivery Service。用户可以随时在 Delivery Service 查询订单状态。

# 二　功能演示

## 2.1　商品导入

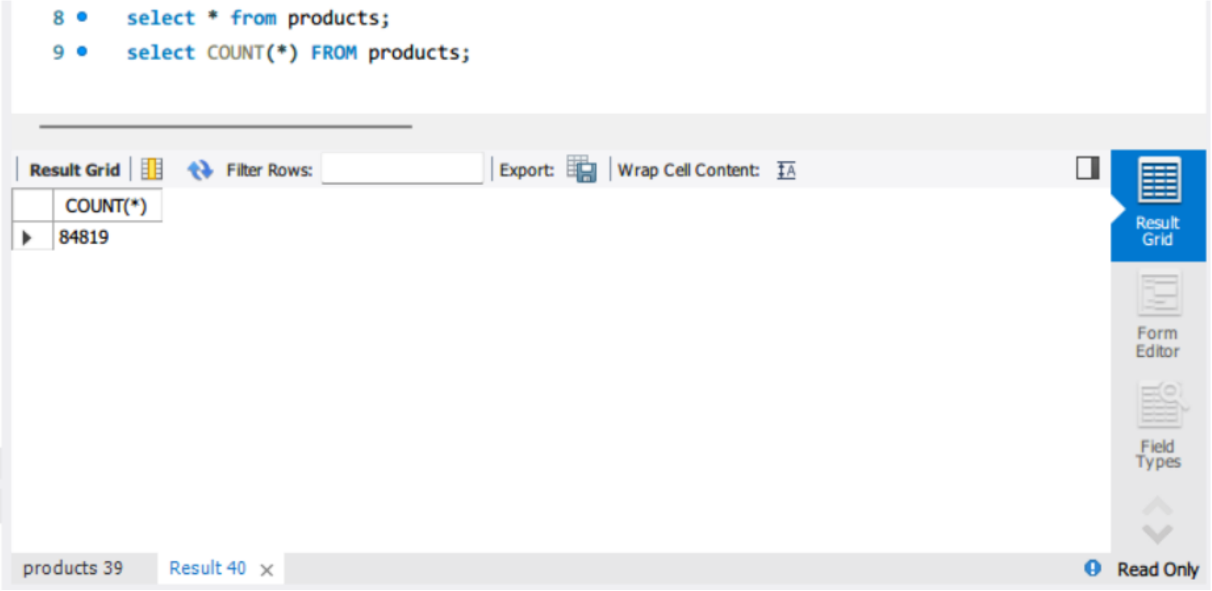
通过 Spring Batch 批处理读取 meta_ Video_ Games.json 的数据并导入数据库



```sql
select * from products;
select COUNT(*) FROM products;
```

| main_cat | title | asin | category |
|---|---|---|---|
| Toys &amp; Games | Reversi Sensory Challenger | 0042000742 | Video Games,PC,Games, |
| Toys &amp; Games | Reversi Sensory Challenger | 0042000742 | Video Games,PC,Games, |
| Toys &amp; Games | Reversi Sensory Challenger | 0042000742 | Video Games,PC,Games, |
| Toys &amp; Games | Reversi Sensory Challenger | 0042000742 | Video Games,PC,Games, |
| Video Games | Xbox 360 MAS STICK | 0324411812 | Video Games,Xbox 360,Accessories,Con |
| Video Games | Phonics Alive! 3: The Speller | 0439335310 | Video Games,PC,Games,</span></spar |
| Video Games | street fighter 2 II turbo super nintendo snes su... | 0276425316 | Video Games,Retro Gaming & Microconsc |
| Video Games | Medal of Honor: Warfighter - Includes Battlefiel... | 0078764343 | Video Games,Xbox 360,Games,</span> |
| Video Games | A to Zap | 0439339960 | Video Games,PC,Games,</span></spar |
| Video Games | Sim City 3000 | 0439339006 | Video Games,PC,Games,</span></spar |
| Video Games | Freddi Fish and the Case of The Missing Kelp Se... | 0439339987 | Video Games,PC,Games,</span></spar |

图 2.2: 商品导入



```sql
select * from products;
select COUNT(*) FROM products;
```

| COUNT(*) |
|---|
| 84819 |

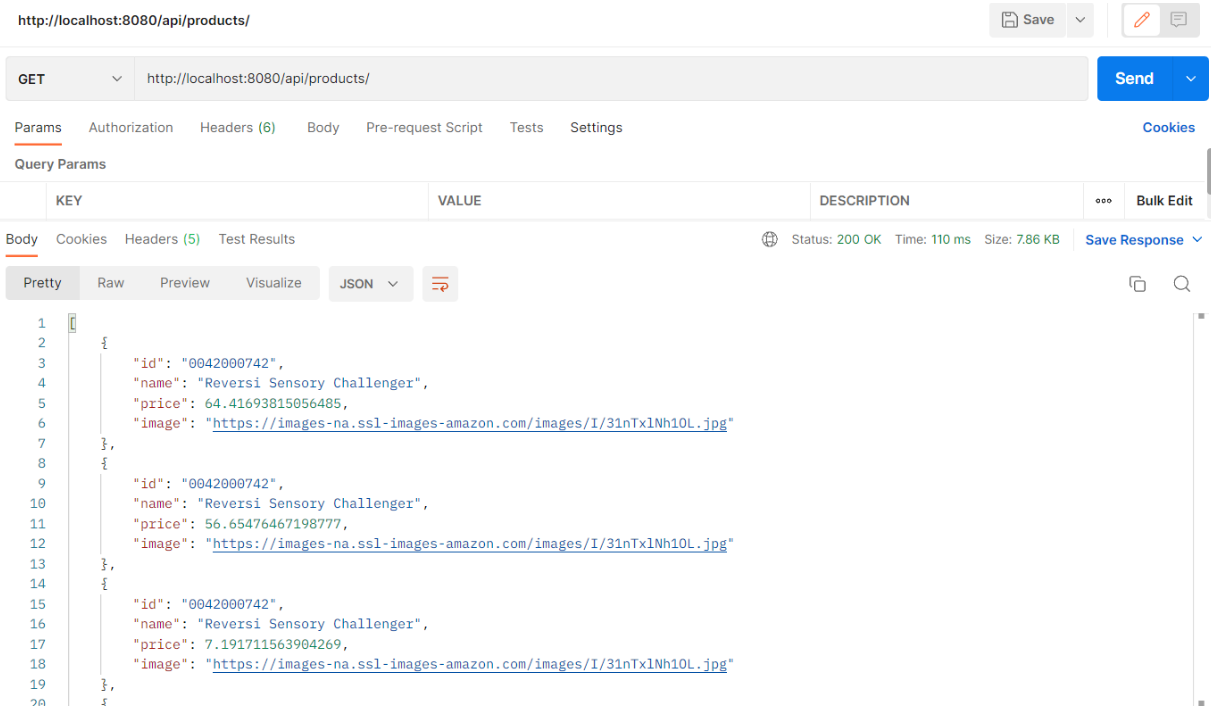
图 2.3: 商品数量, 共计 84819 条

## 2.2 Products 相关

GET - /api/products/ 返回全部商品

GET - /api/products/search/xxxname 搜索商品

GET - /api/products/xxxid 返回某个商品



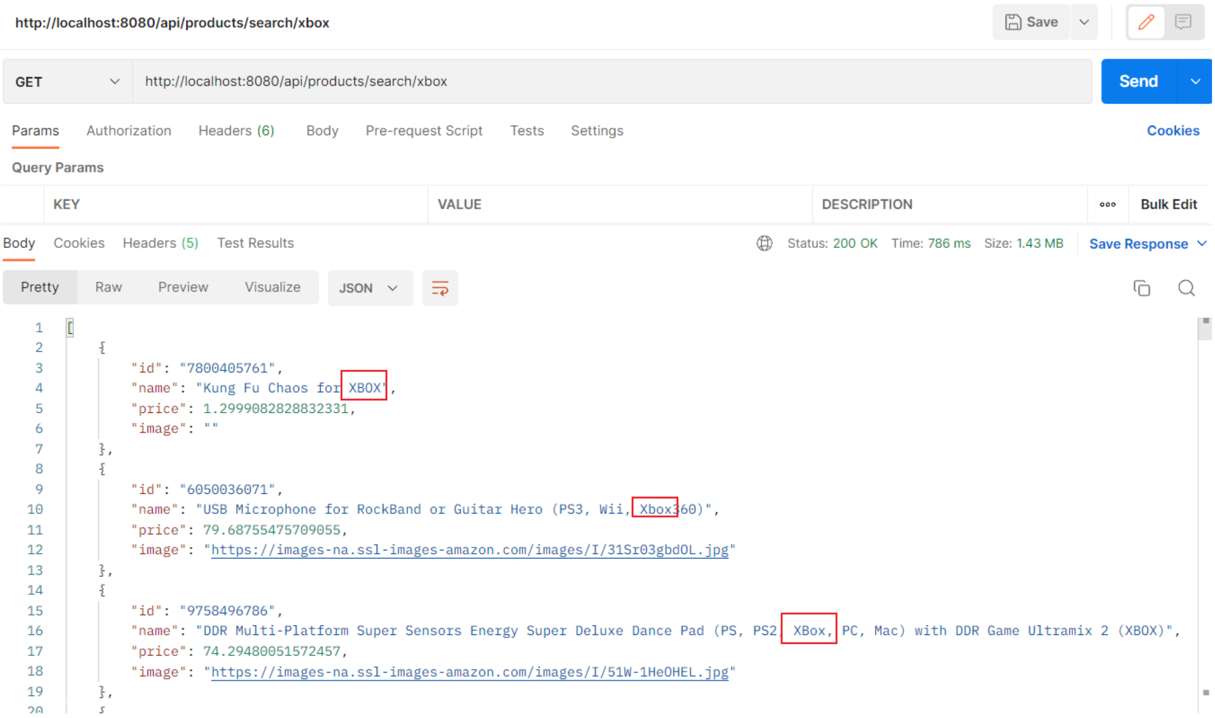
http://localhost:8080/api/products/

GET http://localhost:8080/api/products/

Status: 200 OK Time: 110 ms Size: 7.86 KB

```
[
    {
        "id": "0042000742",
        "name": "Reversi Sensory Challenger",
        "price": 64.41693815056485,
        "image": "https://images-na.ssl-images-amazon.com/images/I/31nTxlNh1OL.jpg"
    },
    {
        "id": "0042000742",
        "name": "Reversi Sensory Challenger",
        "price": 56.65476467198777,
        "image": "https://images-na.ssl-images-amazon.com/images/I/31nTxlNh1OL.jpg"
    },
    {
        "id": "0042000742",
        "name": "Reversi Sensory Challenger",
        "price": 7.191711563904269,
        "image": "https://images-na.ssl-images-amazon.com/images/I/31nTxlNh1OL.jpg"
    },
    {
```

图 2.4: GET - /api/products/ 返回全部商品



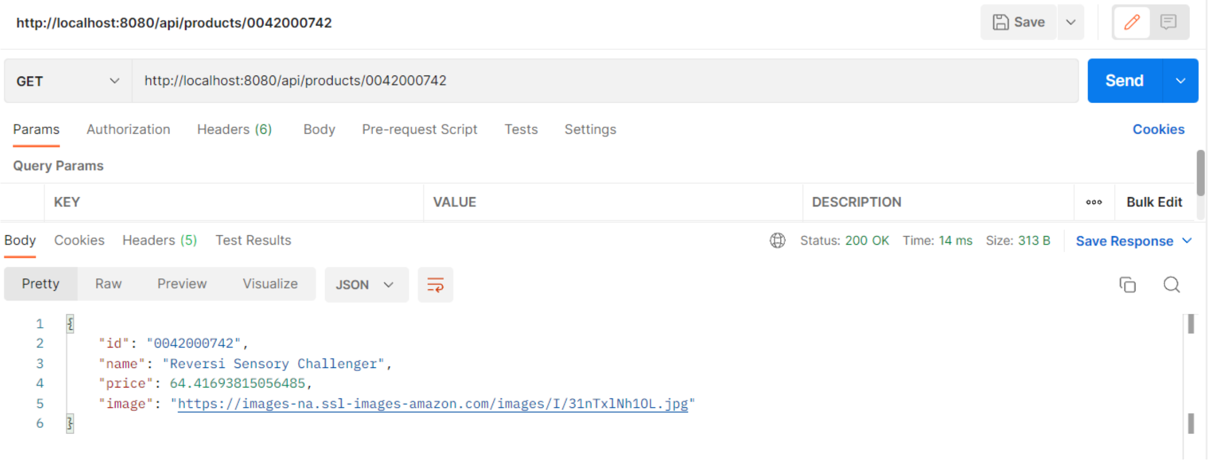
http://localhost:8080/api/products/search/xbox

GET http://localhost:8080/api/products/search/xbox

Status: 200 OK Time: 786 ms Size: 1.43 MB

```
[
    {
        "id": "7800405761",
        "name": "Kung Fu Chaos for XBOX",
        "price": 1.2999082828832331,
        "image": ""
    },
    {
        "id": "6050036071",
        "name": "USB Microphone for RockBand or Guitar Hero (PS3, Wii, Xbox360)",
        "price": 79.68755475709055,
        "image": "https://images-na.ssl-images-amazon.com/images/I/31Sr03gbdOL.jpg"
    },
    {
        "id": "9758496786",
        "name": "DDR Multi-Platform Super Sensors Energy Super Deluxe Dance Pad (PS, PS2, XBox, PC, Mac) with DDR Game Ultramix 2 (XBOX)",
        "price": 74.29480051572457,
        "image": "https://images-na.ssl-images-amazon.com/images/I/51W-1HeOHEL.jpg"
    },
    {
```

图 2.5: GET - /api/products/search/xxxname 搜索商品



```
{
    "id": "0042000742",
    "name": "Reversi Sensory Challenger",
    "price": 64.41693815056485,
    "image": "https://images-na.ssl-images-amazon.com/images/I/31nTxlNh1OL.jpg"
}
```

图 2.6: GET - /api/products/xxxid 返回某个商品

## 2.3 Cart 相关

GET - /api/carts/ 返回购物车全部商品

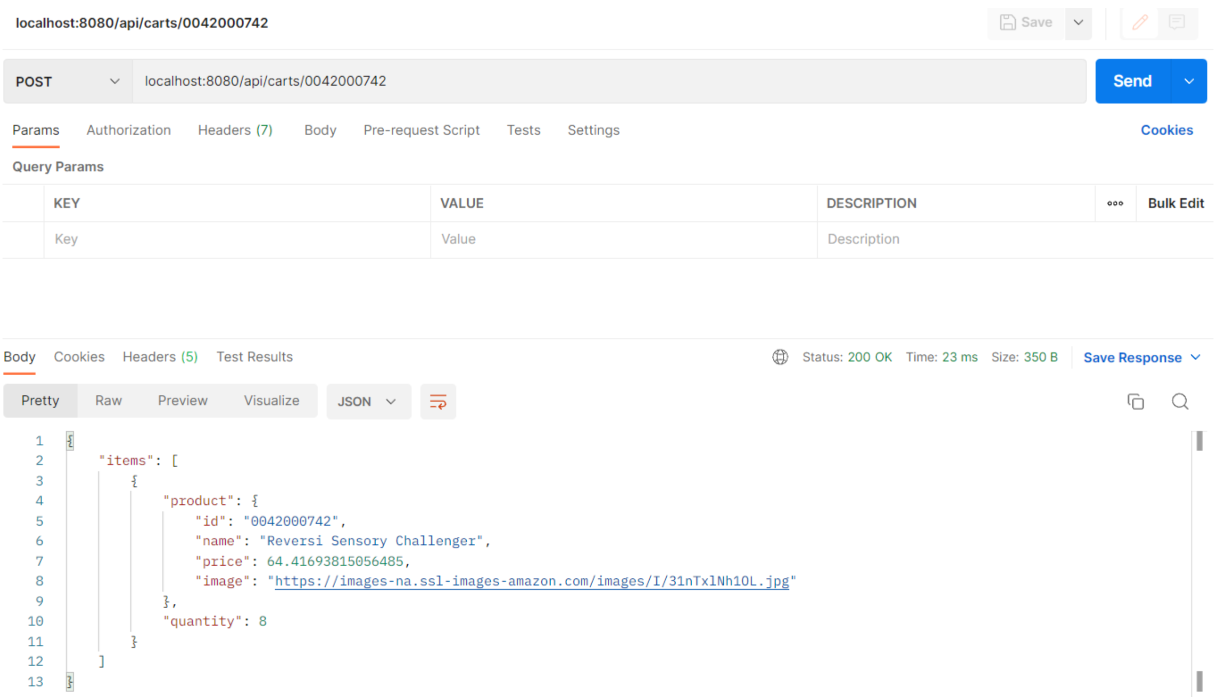
POST - /api/carts/xxxid 购物车增加一个商品

GET - api/carts/checkout 结账

DELETE - api/carts/xxxid 删除购物车中 id 为 xxx 的商品

DELETE - api/carts 清空购物车

仅展示部分，其余功能在 aw07/08 中展示



localhost:8080/api/carts/0042000742

POST localhost:8080/api/carts/0042000742

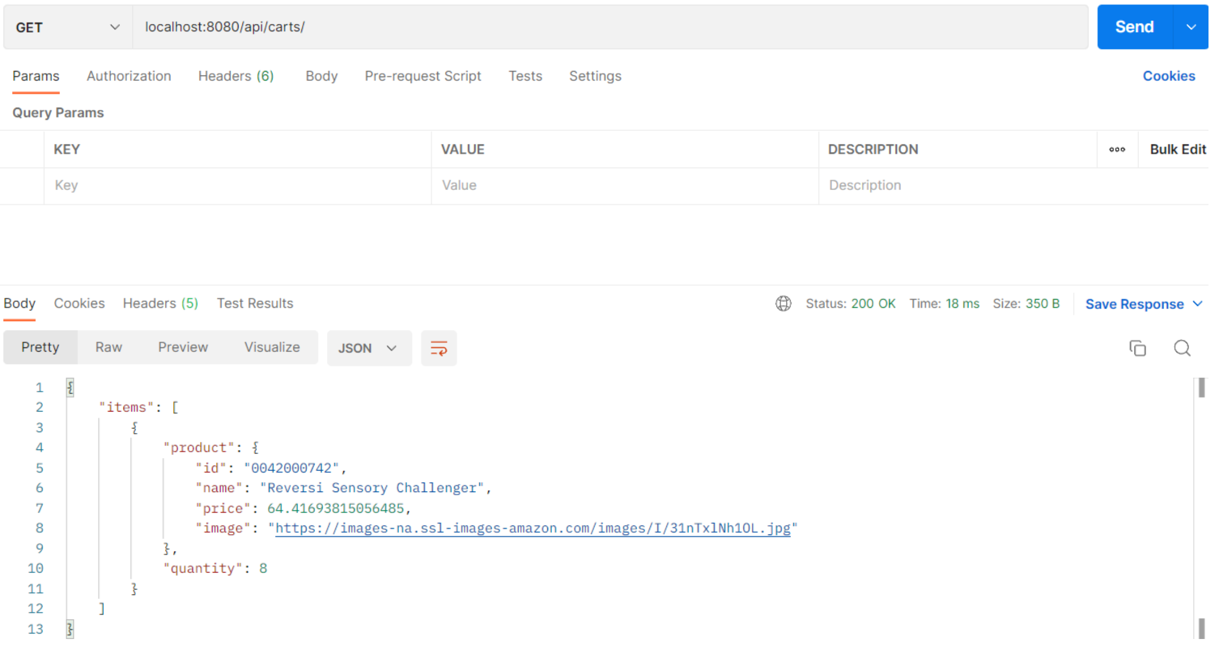
Query Params

| KEY | VALUE | DESCRIPTION |
|---|---|---|
| Key | Value | Description |

Status: 200 OK Time: 23 ms Size: 350 B

```
{
    "items": [
        {
            "product": {
                "id": "0042000742",
                "name": "Reversi Sensory Challenger",
                "price": 64.41693815056485,
                "image": "https://images-na.ssl-images-amazon.com/images/I/31nTxlNh1OL.jpg"
            },
            "quantity": 8
        }
    ]
}
```

图 2.7: POST - /api/carts/xxxid 购物车增加一个商品



GET localhost:8080/api/carts/

Query Params

| KEY | VALUE | DESCRIPTION |
|---|---|---|
| Key | Value | Description |

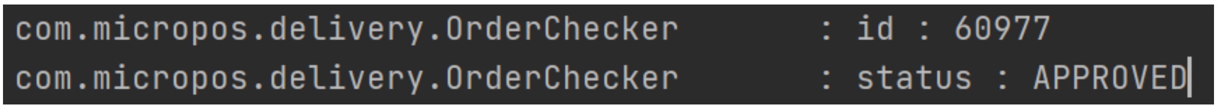
Status: 200 OK Time: 18 ms Size: 350 B

```
{
    "items": [
        {
            "product": {
                "id": "0042000742",
                "name": "Reversi Sensory Challenger",
                "price": 64.41693815056485,
                "image": "https://images-na.ssl-images-amazon.com/images/I/31nTxlNh1OL.jpg"
            },
            "quantity": 8
        }
    ]
}
```

图 2.8: GET - /api/carts/ 返回购物车全部商品

## 2.4 Delivery 相关

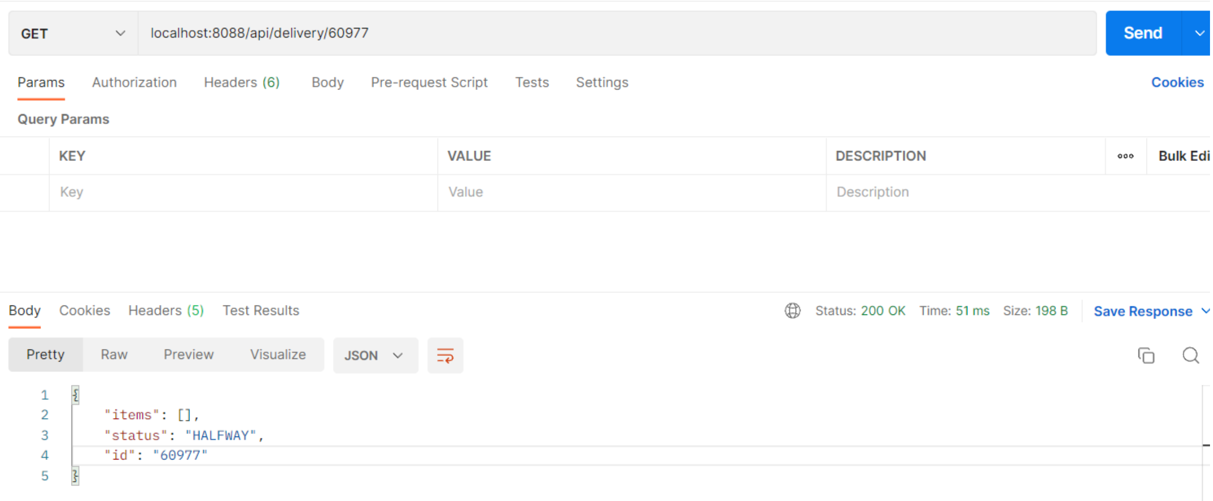
当调用 carts/checkout 后，会生成一个 Order，通过 RabbitMQ 送至 Delivery Service，Order 有状态和 id。



```
com.micropos.delivery.OrderChecker          : id : 60977
com.micropos.delivery.OrderChecker          : status : APPROVED
```

图 2.9: 生成 Order

GET - /api/delivery/xxxid 查询 Order 状态



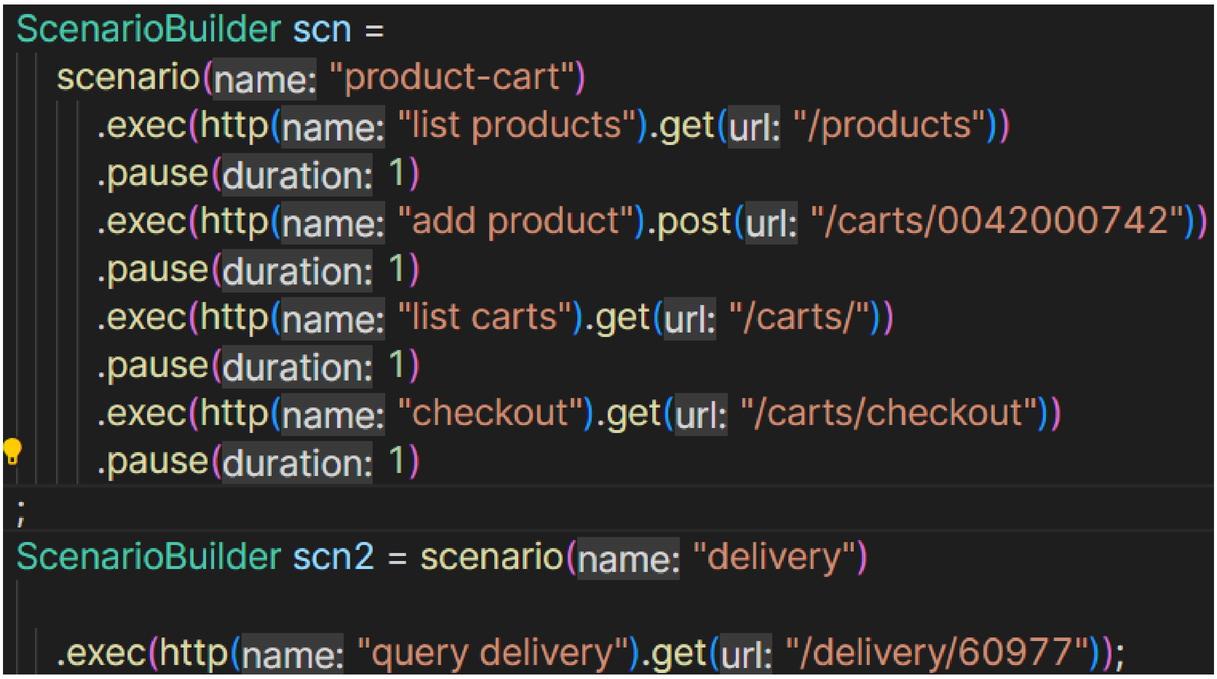
| KEY | VALUE | DESCRIPTION |
|---|---|---|
| Key | Value | Description |

```
{
    "items": [],
    "status": "HALFWAY",
    "id": "60977"
}
```

图 2.10: 订单状态

# 三 实验

做两组压测实验，第一组对 Product/Cart 进行测试。第二组对 Delivery 进行测试。



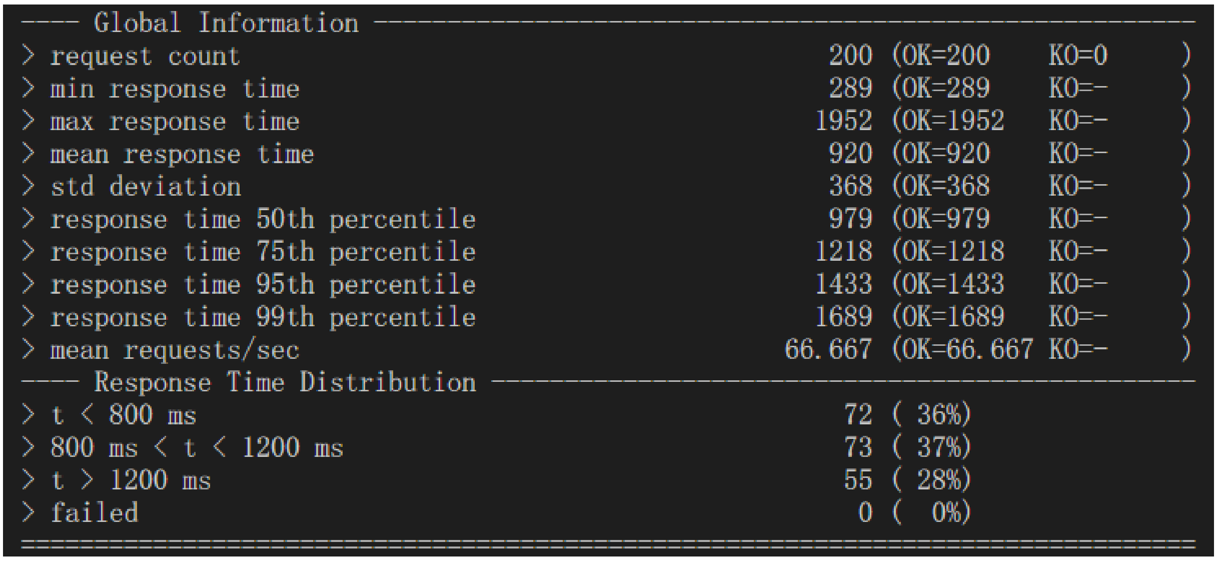
```
ScenarioBuilder scn =
    scenario(name: "product-cart")
        .exec(http(name: "list products").get(url: "/products"))
        .pause(duration: 1)
        .exec(http(name: "add product").post(url: "/carts/0042000742"))
        .pause(duration: 1)
        .exec(http(name: "list carts").get(url: "/carts/"))
        .pause(duration: 1)
        .exec(http(name: "checkout").get(url: "/carts/checkout"))
        .pause(duration: 1)
;
ScenarioBuilder scn2 = scenario(name: "delivery")

    .exec(http(name: "query delivery").get(url: "/delivery/60977"));
```

图 3.11: 实验

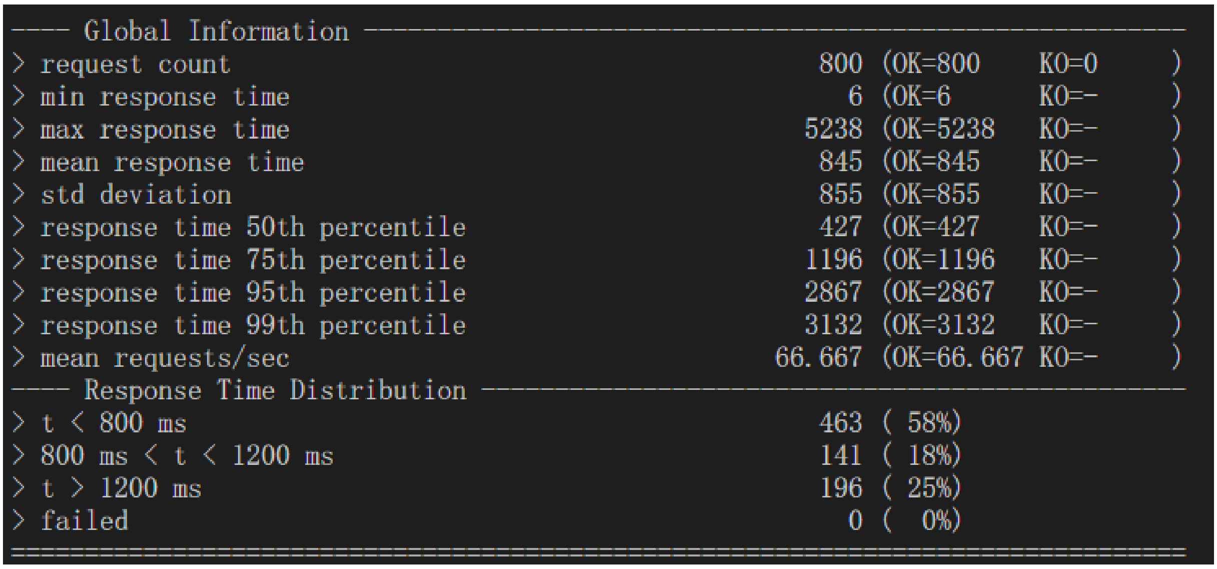
压测 1 内容为展示所有商品，购物车添加商品，展示购物车商品，结账。

压测 2 内容为对订单进行查询。



```
---- Global Information --------------------------------------------------------
> request count                                        200 (OK=200    KO=0     )
> min response time                                    289 (OK=289    KO=-     )
> max response time                                   1952 (OK=1952   KO=-     )
> mean response time                                   920 (OK=920    KO=-     )
> std deviation                                        368 (OK=368    KO=-     )
> response time 50th percentile                        979 (OK=979    KO=-     )
> response time 75th percentile                       1218 (OK=1218   KO=-     )
> response time 95th percentile                       1433 (OK=1433   KO=-     )
> response time 99th percentile                       1689 (OK=1689   KO=-     )
> mean requests/sec                                 66.667 (OK=66.667 KO=-     )
---- Response Time Distribution ------------------------------------------------
> t < 800 ms                                            72 ( 36%)
> 800 ms < t < 1200 ms                                  73 ( 37%)
> t > 1200 ms                                           55 ( 28%)
> failed                                                 0 (  0%)
================================================================================
```

图 3.12: 结果 1



```
---- Global Information --------------------------------------------------------
> request count                                        800 (OK=800    KO=0     )
> min response time                                      6 (OK=6      KO=-     )
> max response time                                   5238 (OK=5238   KO=-     )
> mean response time                                   845 (OK=845    KO=-     )
> std deviation                                        855 (OK=855    KO=-     )
> response time 50th percentile                        427 (OK=427    KO=-     )
> response time 75th percentile                       1196 (OK=1196   KO=-     )
> response time 95th percentile                       2867 (OK=2867   KO=-     )
> response time 99th percentile                       3132 (OK=3132   KO=-     )
> mean requests/sec                                 66.667 (OK=66.667 KO=-     )
---- Response Time Distribution ------------------------------------------------
> t < 800 ms                                           463 ( 58%)
> 800 ms < t < 1200 ms                                 141 ( 18%)
> t > 1200 ms                                          196 ( 25%)
> failed                                                 0 (  0%)
================================================================================
```

图 3.13: 结果 2

# 四　优化

优化有

- 使用 Reactive 架构
- 使用 Eureka 集群
- spring cache 进行缓存
- 对 Delivery 服务进行水平扩展，nginx 实现负载均衡
- RabbitMQ 采用 RoundRobin 分发消息

# 五　致谢

感谢老曹！感谢其他同学们！课程收获很大！