

测试和评价

目的

响应式架构

测试方法1

测试方法2

Not easily broken down

易扩展

代码易扩展

系统易扩展

消息驱动通信

集成作业6数据库

测试和评价

姓名：郭睿杰

学号：MG21330024

院系：计算机科学与技术

目的

- 测试是否为响应式架构（是否满足响应式特点）；
- 测试是否easily broken down；
- 测试是否易扩展；
- 测试是否为消息驱动。

响应式架构

测试方法1

通过 `http://localhost:8080/carts/newcart/{accountId}` 创建多个购物车，通过 `/carts` 对购物车数据库进行查询操作。首先看核心代码

```
@GetMapping(value = "", produces = MediaType.TEXT_EVENT_STREAM_VALUE)
public Flux<CartDto> streamListCarts() {
    return cartService.getAllCarts().map(cart -> cartMapper.toCartDto(cart));
}
```

按流的方式返回购物车：

```
    "items": []
  },
  {
    "cartId": "62b728d63e3a5578627a1a7e",
    "items": []
  },
  {
    "cartId": "62b728d73e3a5578627a1a7f",
    "items": []
  },
  {
    "cartId": "62b72980ee4b5471bc9ca3ab",
    "items": []
  },
  {
    "cartId": "62b72981ee4b5471bc9ca3ac",
    "items": []
  },
  {
    "cartId": "62b72e18d07c6b04a5a7dac4",
    "items": []
  },
  {
    "cartId": "62b7341142b9881305a222c8",
    "items": []
  },
  {
    "cartId": "guoruijie",
    "items": []
  }
}
```

测试方法2

对一个购物车进行结账操作。在Postman中，GET请求

`http://localhost:8080/carts/checkout/guoruijie`

由于结账操作也是以流的方式去完成（核心代码如下，见 `pos-carts` 模块 `\com.micropos.carts.service.CartServiceImpl`）：

```

@Override
public Mono<Double> checkout(String cartId) {
    return cartRepository.findById(cartId).flatMap(cart -> {
        double total = 0;
        for (int i = 0; i < cart.items().size(); i++) {
            total += cart.items().get(i).quantity() *
            cart.items().get(i).productPrice();
        }

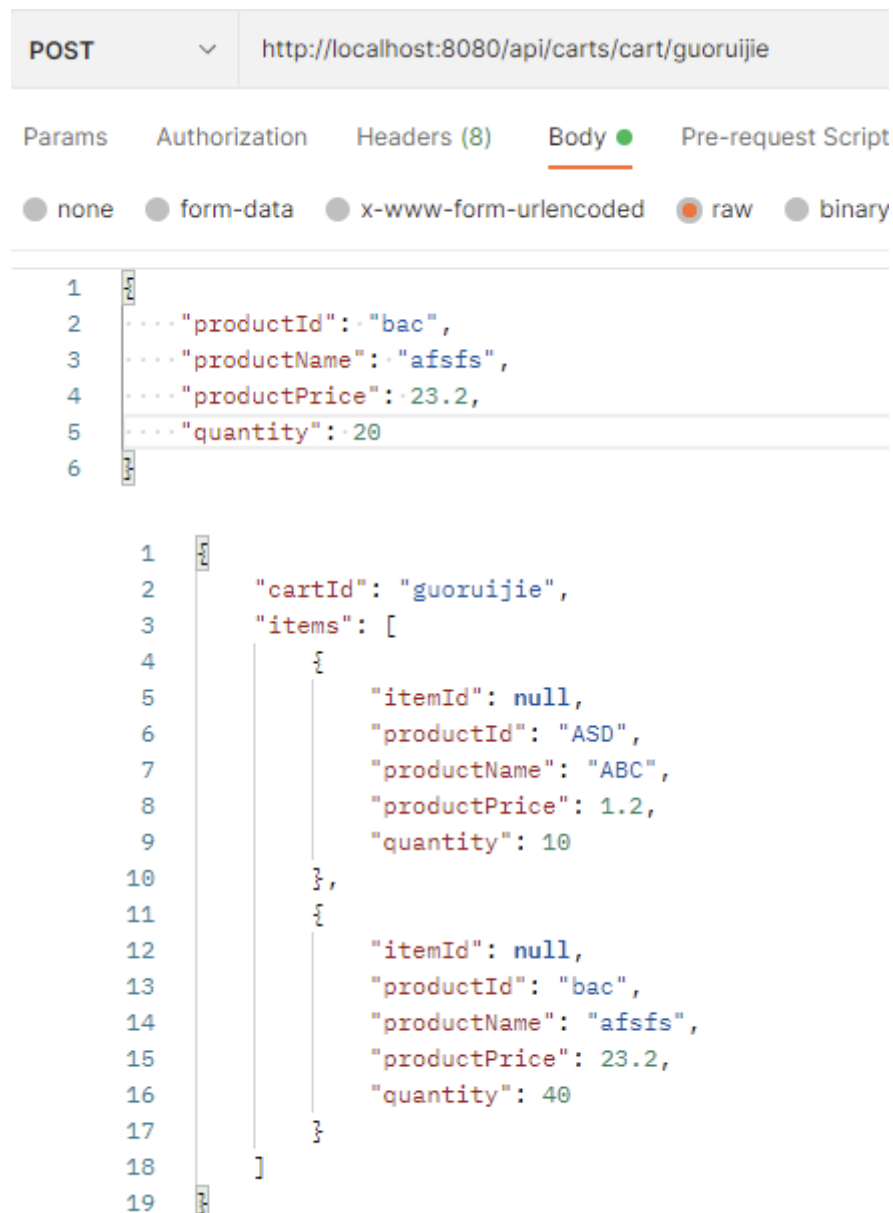
        orderFeignClient.newOrder(cartMapper.toCartDto(cart)).subscribe(System.out::println);

        return this.cartRepository.delete(cart)
            .then(Mono.just(total));
    }).map(d -> d);
}

```

打开断点调试，会发现，当结账结果最终返回用户界面时，checkout当中的代码才会被执行。具体测试如下：

- 我们先往用户购物车中添加一定的物品（可以看到，我们使用POST请求来添加，这是复合RESTFUL架构的）：



The screenshot shows a REST client interface with a POST request to `http://localhost:8080/api/carts/cart/guoruijie`. The 'Body' tab is selected, showing a JSON payload. The payload is a list of items, each with 'productId', 'productName', 'productPrice', and 'quantity' fields. The first item has 'productId': 'ASD', 'productName': 'ABC', 'productPrice': 1.2, and 'quantity': 10. The second item has 'productId': 'bac', 'productName': 'afsfs', 'productPrice': 23.2, and 'quantity': 40.

```

1  {
2    "productId": "bac",
3    "productName": "afsfs",
4    "productPrice": 23.2,
5    "quantity": 20
6  }

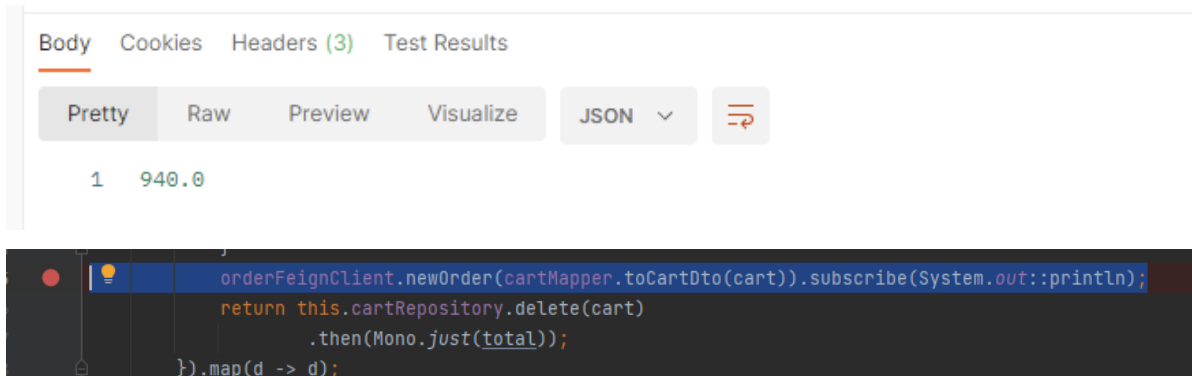
```

```

1  {
2    "cartId": "guoruijie",
3    "items": [
4      {
5        "itemId": null,
6        "productId": "ASD",
7        "productName": "ABC",
8        "productPrice": 1.2,
9        "quantity": 10
10     },
11     {
12       "itemId": null,
13       "productId": "bac",
14       "productName": "afsfs",
15       "productPrice": 23.2,
16       "quantity": 40
17     }
18   ]
19 }

```

然后, checkout (url为 `http://localhost:8080/carts/checkout/guoruijie`)



Not easily broken down

本实验通过熔断机制来保证系统的错误不会传播。

测试: 建立空数据库, 由 `http://localhost:8080/products` 查询。返回NOT_FOUND。

易扩展

易扩展性从架构风格即可观察。

代码易扩展

1. 如, 用户信息当前存储只有用户名, 因此完全可以使用String类型。但仍然封装在pos-api model 中的Account类中。

系统易扩展

不同服务直接几乎无耦合, 通过api模块中的DTO进行数据通信, 扩展性好。

消息驱动通信

在 `pos-orders` 和 `pos-delivery` 模块通过rabbitmq进行消息通信。

实验步骤:

- 启动注册中心、网关、 `pos-delivery` 服务。
- 运行 `pos-orders` 模块中本人已写好的测试代码:
`com.micropos.orders.OrdersApplicationTest`。

```
@SpringBootTest(classes = {OrdersApplication.class})
@RunWith(SpringRunner.class)
public class OrdersApplicationTest {
    @Autowired
    OrderMapper orderMapper;
    @Autowired
    private RabbitTemplate rabbitTemplate;
```

```

@Test
public void testSend() {
    Order order = new Order();
    order.orderId("testId");
    order.accountId("testAccountId");

    rabbitTemplate.convertAndSend("delivery", "order_delivery",
    orderMapper.toOrderDto(order));
}
}

```

- 观察到 pos-delivery 含有 @RabbitListener 注解的方法打印出如下消息：

```

2022-06-28 18:10:14.387 INFO 16948 --- [main] o.s.a.r.c.CachingConnectionFactory : Cr
接收者收到JSON格式消息：
订单编号: testId
订单账户: testAccountId
订单内容: []

```

- 消息驱动工作正常。

集成作业6数据库

	id	name	price	image
▶	1	Serendipity 3 \$100.00 Gift Card	12.00	http://aaabbbccc.img
	2	Serendipity 3 \$100.00 Gift Card	12.00	http://aaabbbccc.img
	3	Wood Puzzle Magic Box Gift Card	12.00	https://images-na.ssl-images-amazon.com/imag...
	4	Amazon.com Gift Cards, Pack of 50 (Old Version...	1250.00	https://images-na.ssl-images-amazon.com/imag...
	5	Whimsical Peacock GiftCard - \$100	12.00	http://aaabbbccc.img
	6	Blooming Tulip GiftCard - \$25	12.00	http://aaabbbccc.img
	7	Wilton Reception Gift Card Holder, White	29.99	https://images-na.ssl-images-amazon.com/imag...
	8	iTunes - Gift Card \$15	12.00	http://aaabbbccc.img
	9	\$25 Lobster Gram Gift Certificate	12.00	http://aaabbbccc.img
	10	Starbucks Gift Card	12.00	https://images-na.ssl-images-amazon.com/imag...
	11	ECA Membership Card	12.00	http://aaabbbccc.img
	12	Lane Bryant Gift Card Collection	12.00	http://aaabbbccc.img
	13	Build-A-Bear Gift Card Collection	12.00	http://aaabbbccc.img
	14	Spa Finder Gift Card	12.00	http://aaabbbccc.img
	15	Chili's Gift Card Collection	12.00	http://aaabbbccc.img
	16	Regal Entertainment Group Gift Card Collection	12.00	http://aaabbbccc.img