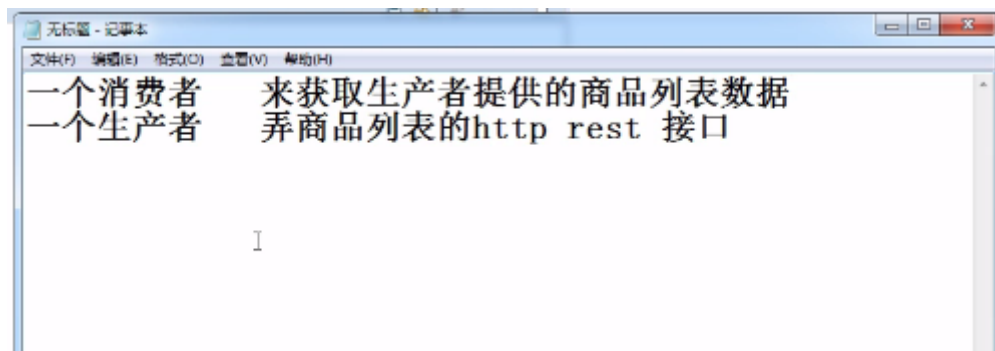


消费者去调用生产者



一,生产者(继承于父工程,需要连接数据库,热部署,)

1.yml语言中:主要是开启true,

```
1 server:
2   port: 13000
3   # 数据源的配置
4   spring:
5     application:
6       name: good-rest
7     datasource:
8       type: com.alibaba.druid.pool.DruidDataSource
9       username: root
10      password: root
11      driver-class-name: com.mysql.jdbc.Driver
12      url: jdbc:mysql://localhost:3306/taotao
13      dbcp2: # 进行数据库连接池的配置
14      min-idle: 5 # 数据库连接池的最小维持连接数
15      initial-size: 5 # 初始化提供的连接数
16      max-total: 5 # 最大的连接数
17      max-wait-millis: 200 # 等待连接获取的最大超时时间
18
19   eureka:
20     instance:
21       prefer-ip-address: true
22       instance-id: ${spring.application.name}:${spring.application.instance-id:${server.port}}
23     client:
24       register-with-eureka: true
25       fetch-registry: true
26       service-url:
27         defaultZone: http://admin:admin@localhost:12000/eureka/
```

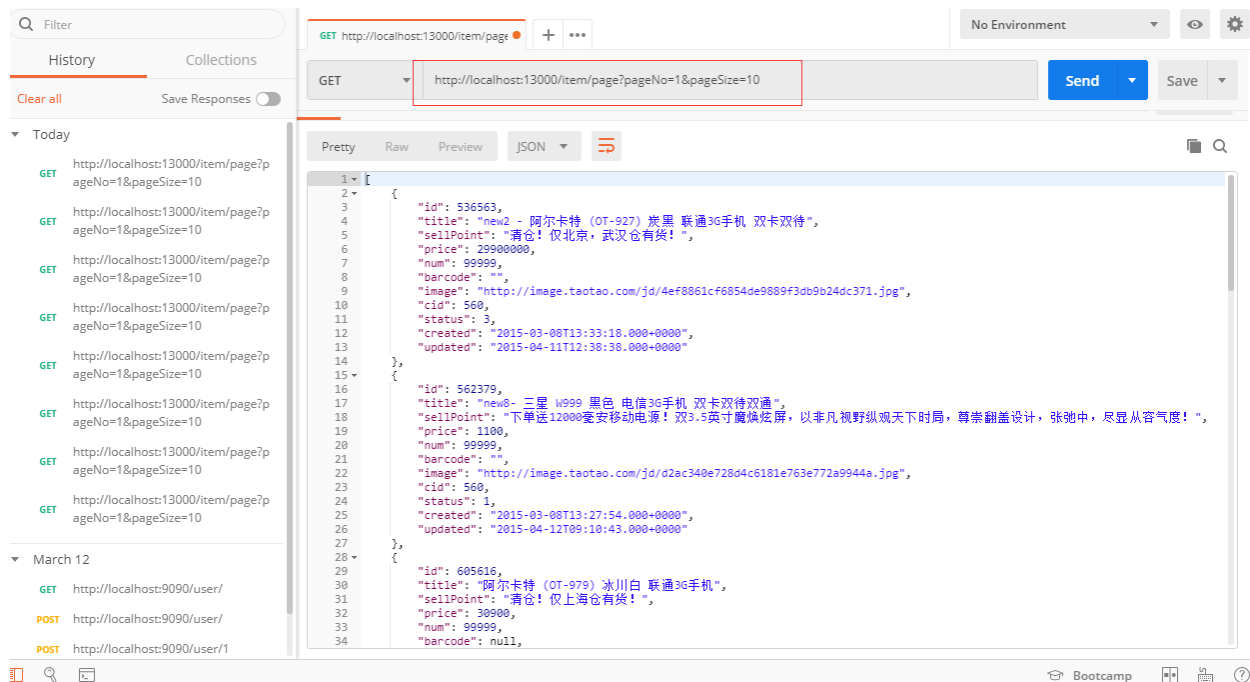
```
28
29 # mybatis 配置
30 mybatis:
31   mapper-locations: classpath:mapping/*.xml
32   type-aliases-package: com.lanou.rest.domain
33 # 显示sql
34 logging:
35   level:
36     com.lanou.rest.domain: debug
37
38
39 #pagehelper分页插件
40 pagehelper:
41   helperDialect: mysql
42   reasonable: true
43   supportMethodsArguments: true
44   params: count=countSql
45   returnPageInfo: check
```

2.消费者和生产者都是客户端jar都是client

```
1 <!-- c和p都属于客户端,注册中心属于服务端 -->
2 <dependency>
3   <groupId>org.springframework.cloud</groupId>
4   <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
5 </dependency>
```

3.我们在生产者中用到了分页

做完之后尽量用postman测试一下



二.消费者

仅仅是有个调用,springboot只需要一个热部署和web页面即可,可以加一个cloud架包需要改成client

1仅仅一个controller即可.

```

1 @RestController
2 @RequestMapping("item")
3 public class GoodController {
4
5     @Resource
6     private RestTemplate restTemplate;
7
8     @RequestMapping("page")
9     public Object getItem(@RequestParam("pageNo") Integer pageNo,
10     @RequestParam("pageSize") Integer pageSize) {
11         // http://localhost:13000/item/page?pageNo=1&pageSize=10
12         ResponseEntity<List> responseEntity =
13             restTemplate.getForEntity("http://GOOD-REST/item/page?
14             pageNo="+pageNo+"&pageSize="+pageSize+", List.class);
15         List<Item> list = responseEntity.getBody();
16         return list;
17     }
18 }

```

2.yml语言:

```

1 server:
2     port: 14000

```

```

3
4 spring:
5   application:
6     name: good-portal
7
8
9 eureka:
10  instance:
11    prefer-ip-address: true
12    instance-id: ${spring.application.name}:${spring.application.instance_id:${server.port}}
13  client:
14    register-with-eureka: true
15    fetch-registry: true
16    service-url:
17      defaultZone: http://admin:admin@localhost:12000/eureka/

```

3.我们要建一个@Bean,交给容器管理.

启动类:

```

1 @SpringBootApplication
2 @EnableEurekaClient
3 public class GoodCAplication {
4   //将RestTemplate交给容器管理,@LoadBalanced这是负载均衡注解.
5   解析http://GOOD-REST
6   @LoadBalanced
7   @Bean
8   public RestTemplate restTemplate() {
9     return new RestTemplate();
10  }
11
12  public static void main(String[] args) {
13    SpringApplication.run(GoodCAplication.class, args);
14  }

```

4.<https://www.cnblogs.com/duanxz/p/3510622.html>

RestTemplate:(强大:自动将json数据转成pojo对象,方法:getForObject)

在微服务都是以HTTP接口的形式暴露自身服务的,因此在调用远程服务时必须使用HTTP客户端。我们可以使用JDK原生的URLConnection、Apache的Http Client、Netty的异步HTTP

Client, Spring的RestTemplate。但是，用起来最方便、最优雅的还是要属Feign了。这里介绍的是RestTemplate。

什么是RestTemplate？

RestTemplate是Spring提供的用于访问Rest服务的客户端，

RestTemplate提供了多种便捷访问远程Http服务的方法,能够大大提高客户端的编写效率。

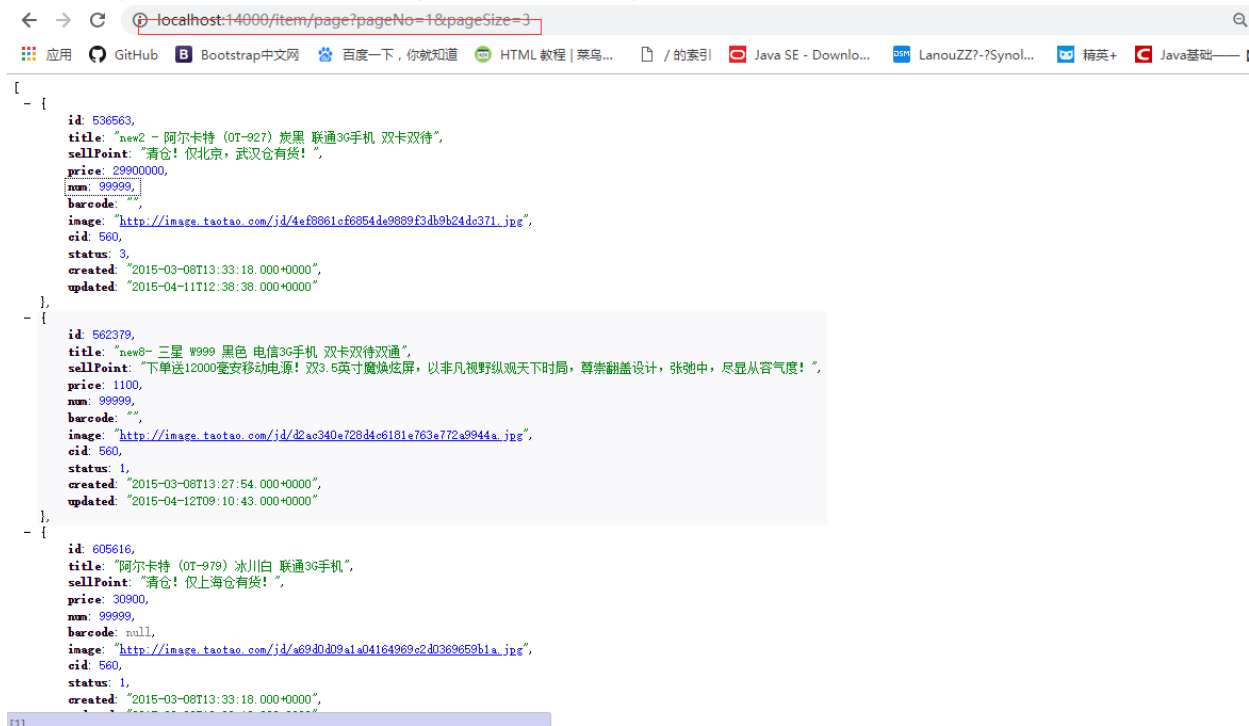
调用RestTemplate的默认构造函数，RestTemplate对象在底层通过使用java.net包下的实现创建HTTP 请求，

可以通过使用ClientHttpRequestFactory指定不同的HTTP请求方式。

ClientHttpRequestFactory接口主要提供了两种实现方式

1、一种是SimpleClientHttpRequestFactory，使用J2SE提供的方式（既java.net包提供的方式）创建底层的Http请求连接。

2、一种方式是使用HttpComponentsClientHttpRequestFactory方式，底层使用HttpClient访问远程的Http服务，使用HttpClient可以配置连接池和证书等信息。



是消费者的端口号