

EASYMALL项目整合

2020年2月10日 16:27

1.easymall项目整合

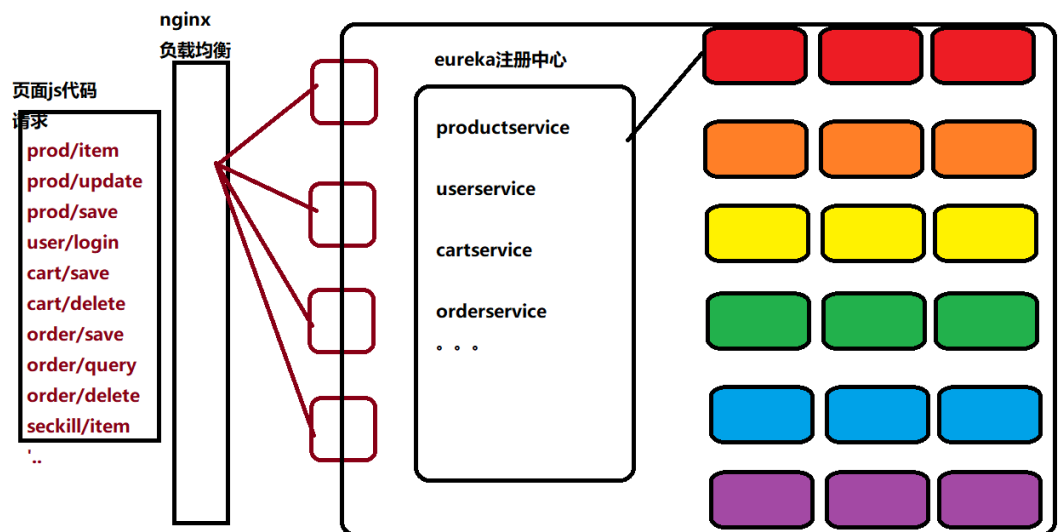
1.1介绍

easymall就是web阶段电商项目，具备的功能非常复杂，多，（web阶段只实现用户注册和登陆，没有使用架构技术）。

互联网框架中的easymall功能（微服务）

- 商品系统：处理商品分页查询，单个查询，商品新增，商品修改
- 图片系统：负责整个系统的所有图片的上传逻辑。（浏览交给nginx）
- 用户系统：注册，登录的逻辑交给用户系统，用户身份验证也是用户系统
- 购物车系统：用户的购物车数据
- 订单系统：用户提交删除订单
- 搜索系统：单独运行一个微服务工程，执行easymall中搜索商品功能
- 秒杀系统：单独处理瞬间高并发的秒杀逻辑。

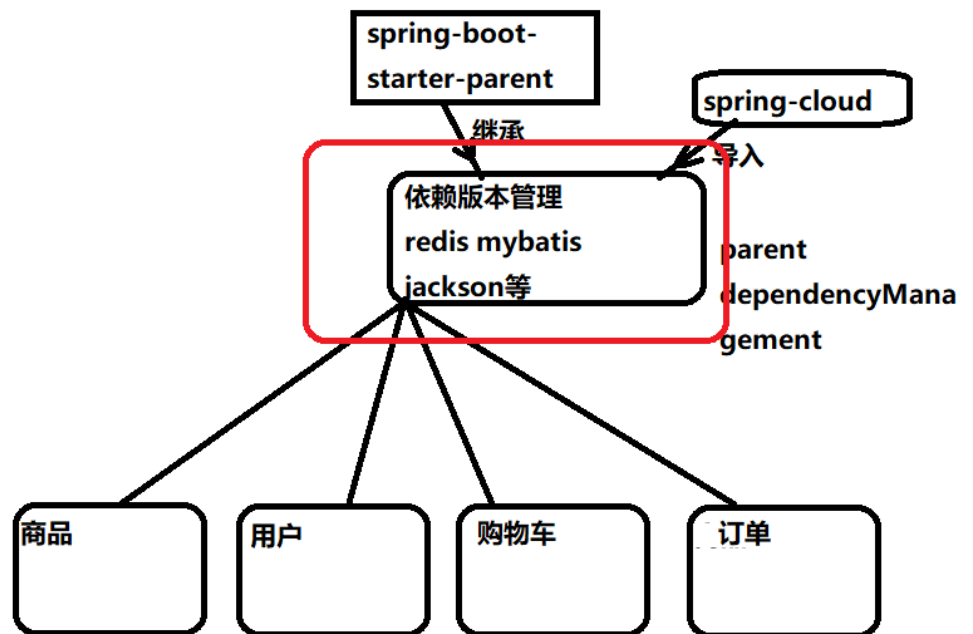
1.2springcloud微服务框架结构



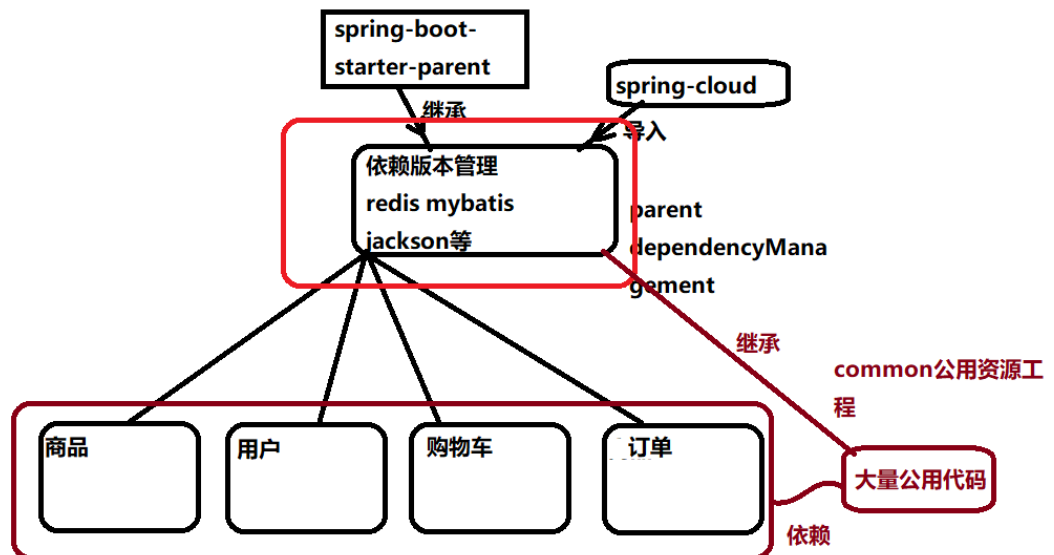
2.搭建一下基本的maven项目结构

2.1maven管理项目的结构

准备一个父级工程easymall-parent 管理所有easymall中子工程的资源版本（子工程中，看不到版本号的，依赖，依赖管理看不到，插件看不到）



对于每一个微服务中的公用代码可以提取到一个common工程通过依赖实现



2.3搭建easymall的管理结构

使用课前资料--06springcloud--07easymall项目--common工程代码粘贴

2.1easymall-parent工程

- 创建parent工程
- pom文件:
 - 继承spring-boot-starter-parent

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>1.5.9.RELEASE</version>
</parent>
```

- 导入springcloud
- 管理有版本号的资源 (dependencyManagement)
 - ◆ redis
 - ◆ mybatis
 - ◆ Elasticsearch

```
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-dependencies</artifactId>
      <version>Edgware.RELEASE</version>
      <scope>import</scope>
      <type>pom</type>
    </dependency>
    <!--子工程具有版本需求的资源在parent定义-->
    <!--redis依赖版本1.4.7.RELEASE-->
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-redis</artifactId>
      <version>1.4.7.RELEASE</version>
    </dependency>
    <!--mybatis-->
    <dependency>
      <groupId>org.mybatis.spring.boot</groupId>
      <artifactId>mybatis-spring-boot-starter</artifactId>
      <version>1.3.0</version>
    </dependency>
    <!--elasticsearch相关依赖-->
    <dependency>
      <groupId>org.elasticsearch</groupId>
      <artifactId>elasticsearch</artifactId>
      <version>5.5.2</version>
    </dependency>
    <dependency>
      <groupId>org.elasticsearch.client</groupId>
      <artifactId>transport</artifactId>
      <version>5.5.2</version>
    </dependency>
  </dependencies>
</dependencyManagement>
```

2.2公用工具资源工程

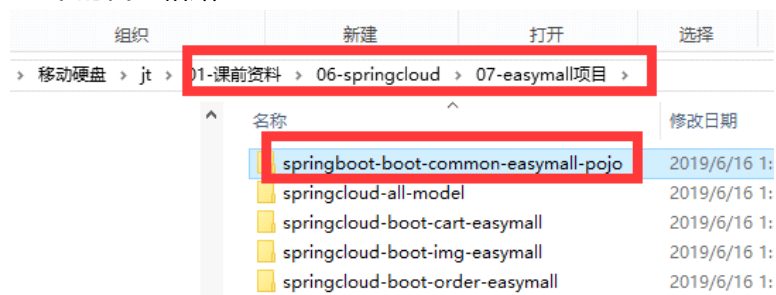
- 创建工程（idea的和maven的聚合管理）
- pom文件：
 - 继承easymall-parent

```
<parent>
  <artifactId>easymall-parent</artifactId>
  <groupId>cn.tedu</groupId>
  <version>1.0-SNAPSHOT</version>
</parent>
```

- 依赖：starter-web（但是common工程不需要启动）

```
<dependencies>
  <!--starter-web-->
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.11</version>
    <scope>test</scope>
  </dependency>
</dependencies>
```

- src下的代码粘贴



粘贴的文件夹，一定要放对地方

代码的内容：

pojo：表示一个工程具备业务含义的所有javabean

util:公用的一些工具类的代码。

cookieUtils：操作cookie，不用在写Cookie[]

MapperUtil:序列化操作对象和json字符串的工具类

MD5Util:加密工具类

UUIDUtil：生成唯一字符串的工具类

vo：view object,项目根据ajax接收的json字符串结构，

对应编写的视图对象，例如， ajax要接收json

```
{ "age" :18,"level":19,"driver":"王翠花"}==Object1
```

对象包含三个属性

integer age

Integer level

String driver

仅仅是按照ajax要求进行对话交互的逻辑。