目录

[1、商品相关接口 3](#_Toc29962)

[1.1、商品分页查询：sobeyMallProduct/V1/products（get） 3](#_Toc14436)

[1.1.1、逻辑流程 3](#_Toc13410)

[1.2、商品预新增：sobeyMallProduct/V1/products/pre（get） 3](#_Toc1074)

[1.2.1、逻辑流程 3](#_Toc18325)

[1.3、商品新增：sobeyMallProduct/V1/products（post） 3](#_Toc10853)

[1.3.1、逻辑流程 3](#_Toc27324)

[1.4、商品删除：sobeyMallProduct/V1/products/{uuid}（delete） 5](#_Toc24657)

[1.4.1、逻辑流程 5](#_Toc18678)

[1.5、商品修改：sobeyMallProduct/V1/products（patch） 6](#_Toc11086)

[1.5.1、逻辑流程 6](#_Toc1572)

[1.6、商品列表：sobeyMallProduct/V1/products/list（get） 9](#_Toc11255)

[1.6.1、逻辑流程 9](#_Toc9134)

[1.7、商品查询：sobeyMallProduct/V1/products/{uuid}（get） 9](#_Toc7272)

[1.7.1、逻辑流程 9](#_Toc75)

[1.8、商品统计：sobeyMallProduct/V1/products/statistic（get） 10](#_Toc24927)

[1.8.1、逻辑流程 10](#_Toc29468)

[2、商品素材相关接口 10](#_Toc19817)

[2.1、商品素材查询：sobeyMallProduct/V1/medias（get） 10](#_Toc17570)

[2.1.1、逻辑流程 10](#_Toc28298)

[2.2、商品素材新增：sobeyMallProduct/V1/medias（post） 11](#_Toc7247)

[2.2.1、逻辑流程 11](#_Toc16887)

[2.3、商品素材分页查询：sobeyMallProduct/V1/medias/page（post） 11](#_Toc28618)

[2.3.1、逻辑流程 11](#_Toc18145)

[2.4、商品素材删除：sobeyMallProduct/V1/medias/{uuid}（delete） 12](#_Toc10521)

[2.4.1、逻辑流程 12](#_Toc28776)

[3、商品类别相关接口 12](#_Toc9035)

[3.1、商品分类列表查询：sobeyMallProduct/V1/categorys/list（post） 12](#_Toc14365)

[3.1.1、逻辑流程 12](#_Toc22884)

[4、商品亮点相关接口 13](#_Toc24012)

[4.1、亮点预新增：sobeyMallProduct/V1/bright-spot/pre（get） 13](#_Toc16778)

[4.1.1、逻辑流程 13](#_Toc1636)

[5、商品按量相关接口 13](#_Toc24265)

[5.1、按量预新增：sobeyMallProduct/V1/metric/pre（get） 13](#_Toc15149)

[5.1.1、逻辑流程 13](#_Toc9717)

[5.2、按量类型查询：sobeyMallProduct/V1/metric/type（get） 13](#_Toc5044)

[5.2.1、逻辑流程 13](#_Toc30303)

[6、商品折扣相关接口 14](#_Toc26645)

[6.1、折扣分页查询：sobeyMallProduct/V1/discount/page（post） 14](#_Toc11649)

[6.1.1、逻辑流程 14](#_Toc5765)

[6.2、新增折扣：sobeyMallProduct/V1/discount（post） 14](#_Toc21713)

[6.2.1、逻辑流程 14](#_Toc32337)

[6.3、修改折扣：sobeyMallProduct/V1/discount（patch） 15](#_Toc9046)

[6.3.1、逻辑流程 15](#_Toc9130)

[6.4、删除折扣：sobeyMallProduct/V1/discount/{uuid}（delete） 15](#_Toc6324)

[6.4.1、逻辑流程 15](#_Toc19189)

[6.5、生效折扣：sobeyMallProduct/V1/discount/protocal/{uuid}（post） 16](#_Toc26056)

[6.5.1、逻辑流程 16](#_Toc3961)

[6.6、协议查看：sobeyMallProduct/V1/discount/protocal/{uuid}（get） 16](#_Toc14103)

[6.6.1、逻辑流程 16](#_Toc22688)

# 1、商品相关接口

## 1.1、商品分页查询：sobeyMallProduct/V1/products（get）

### 1.1.1、逻辑流程

1. 参数：  
    page：页码  
    size：每页条数  
    Product：商品类
2. 将page与size设置在Page类中；
3. 根据传递的参数查询出商品列表封装在Page类中，返回到前端。

## 1.2、商品预新增：sobeyMallProduct/V1/products/pre（get）

### 1.2.1、逻辑流程

①，参数：无

②，生成MediaDir（配置文件父路径+随机uuid），Uuid，Logo，MasterGraph，Banner，Scene，UserGuide，DevelopGuide，PriceTable，PriceLimited，Sla等字段值（除MediaDir外，其他均为随机uuid）；

③，将生成的字段封装在Product类中，返回到前端。

## 1.3、商品新增：sobeyMallProduct/V1/products（post）

### 1.3.1、逻辑流程

①，参数：

Product：商品类

BindingResult：spring自带的参数校验结果类

②，对BindingResult进行校验，如果BindingResult存在错误，则将错误信息封装到自定义的异常中进行抛出，提示相关错误信息；  
  
判断Product实例是否为空，若为空直接抛出异常提示“商品不能为空”；  
校验商品名和编码不能为空，若为空，抛出异常提示“商品名称和编码不能为空”；  
  
将MediaDir，uuid，请求头中的token和站点编码site获取到。

③，将商品参数新增到商品表。

④，如果存在关联商品的数据，则先判断关联是否存在（避免重复关联），如不存在就将关联商品相关数据新增到关联商品表（关联商品表的uuid是在何时生成的？前台页面填写了关联商品，却没有传递到后台？）。

⑤，将商品亮点相关数据新增到商品亮点表中。

⑥，如果存在商品功能数据，则将数据新增到商品功能表中（商品功能表的uuid是在何时生成的？）。

⑦，如果存在商品合作案例数据，则将数据新增到商品合作案例表中（商品合作案例表的uuid是在何时生成的？）。

⑧，将版本数据新增到商品版本表（在新增前生成了uuid）；

如果版本数据中存在商品版本自定义规格数据，则将数据新增到商品版本自定义规格表（在新增前生成了uuid）；

如果商品版本自定义规格数据中存在商品版本自定义规格选项数据，则将数据新增到商品版本自定义规格选项表（在新增前未生成了uuid）。

⑨，如果存在商品按量计费规则数据，则将数据新增到商品按量计费规则表中。

⑩，如果存在商品套餐数据，则将数据新增到商品套餐表中（在新增前生成了uuid）；

如果商品套餐数据中存在自定义套餐包资源名称数据，并且数据中包metricId，否者抛出异常提示“metricId必须传递”，将数据新增到自定义套餐包资源名称表（在新增前生成了uuid）；

如果自定义套餐包资源名称数据中存在商品套餐自定义选项数据，则将数据新增到商品套餐自定义选项表（在新增前未生成了uuid）。

⑪，如果存在商品自定义权限数据，并且自定义权限名称不能为空，否则抛出异常提示“自定义权限的名称不能为空”，将数据新增到商品自定义权限表中（在新增前未生成了uuid）。

⑫，在整个商品新增的过程中报错的话，就将素材父目录进行删除，抛出异常提示“新增商品失败”。

## 1.4、商品删除：sobeyMallProduct/V1/products/{uuid}（delete）

### 1.4.1、逻辑流程

①，参数：

uuid：商品id

②，将uuid封装到Product实例中，根据Product实例查询商品表，判断查询结果是否为空，若为空则抛出异常提示“商品不存在”；

③，遍历上一步查询出来的结果；

④，将商品id封装到CooperationCase中，对商品的合作案例进行删除；

⑤，将商品id封装到Function中，对商品的功能进行删除；

⑥，将商品id封装到BrightSpot中，根据BrightSpot查询出亮点列表；  
  
遍历亮点列表，将亮点uuid封装到Media中，根据Media查询出素材列表；  
  
遍历素材列表，将素材删除的同时也将素材中的地址取出将素材对应的文件进行删除；  
  
最后删除亮点信息；

⑦，将商品id封装到ProductPrivilege中，根据ProductPrivilege查询商品自定义权限列表；遍历商品自定义权限列表，根据商品自定义权限uuid查询出商品自定义权限，若查询出的商品自定义权限不为空，则根据uuid删除商品自定义权限（既然根据商品id查询出了商品自定义权限列表，为什么还要根据uuid再查询一次，直接删除不就行了？）；

⑧，将商品id封装到Version中，根据Version查询出版本列表；  
  
遍历版本列表，将版本uuid封装到VersionCustom中，根据VersionCustom查询出商品版本自定义规格列表；  
  
遍历商品版本自定义规格列表，将商品版本自定义规格uuid封装到VersionCustomOption中，根据VersionCustomOption删除商品版本自定义规格选项数据；  
  
再删除商品版本自定义规格数据；再删除版本数据；

⑨，将商品id封装到Media中，根据Media查询出素材列表；遍历素材列表，将素材删除的同时也将素材中的地址取出将素材对应的文件进行删除；

⑩，将商品id封装到Packages中，根据Packages查询出套餐包列表；  
  
遍历套餐包列表，将套餐包uuid封装到PackagesCustom中，根据PackagesCustom查询出自定义套餐包资源名称列表；  
  
遍历自定义套餐包资源名称列表，将自定义套餐包资源名称uuid封装到PackagesCustomOption中，根据PackagesCustomOption删除商品套餐自定义选项数据；  
  
再删除自定义套餐包资源名称数据；再删除套餐包数据；

⑪，删除素材的父目录；

⑫，根据商品id商品数据；

⑬，在商品的整个删除中，如果出现任何异常，就进行日志记录“删除商品报错”。（为什么删除商品后不想前端返回商品的结果信息）

## 1.5、商品修改：sobeyMallProduct/V1/products（patch）

### 1.5.1、逻辑流程

①，参数：

Product：商品类

②，根据商品uuid查询数据库，如果商品不存在则抛出异常提示“商品不存在”；

③，修改商品表数据；

④，如果关联商品数据列表不为空，遍历关联商品列表；判断关联商品数据中操作类型是否为delete，若是，并且关联商品数据的uuid不为空，则将关联商品数据删除；否则，检验是否进行了重复关联，若不是，就将关联商品数据插入或更新到数据库；

⑤，如果商品亮点数据列表不为空，遍历商品亮点列表；  
  
判断商品亮点数据中操作类型是否为delete，若是，并且商品亮点数据的uuid不为空，则将商品亮点数据删除，在删除之前将商品亮点uuid封装到Media中的mediaId中，根据Media查询出素材列表，遍历列表，删除素材数据，并将素材对应的文件进行删除；  
  
否则，将商品亮点数据插入或更新到数据库；

⑥，如果商品功能数据列表不为空，遍历商品功能数据列表；判断商品功能数据中操作类型是否为delete，若是，并且商品功能数据的uuid不为空，则将商品功能数据删除；否则，将商品功能数据插入或更新到数据库；

⑦，如果商品案例数据列表不为空，遍历商品案例数据列表；

判断商品案例数据中操作类型是否为delete，若是，并且商品案例数据的uuid不为空，则将商品案例数据删除；

否则，将商品案例数据插入或更新到数据库；

⑧，如果商品版本数据列表不为空，遍历商品版本数据列表；

判断商品版本数据中操作类型是否为delete，若是，并且商品版本数据的uuid不为空，则将商品版本相关数据（按商品版本自定义规格选项，商品版本自定义规格，商品版本的顺序）删除；

判断商品版本数据中操作类型是否为update，若是，将商品版本数据更新到数据库；

判断商品版本数据中操作类型是否为insert，若是，则生成uuid，将商品版本数据新增到数据库；

如果商品版本数据中商品版本自定义规格列表不为空，遍历商品版本自定义规格列表；

判断商品版本自定义规格数据中操作类型是否为delete，若是，并且商品版本自定义规格数据的uuid不为空，则将商品版本自定义规格相关数据（按商品版本自定义规格选项，商品版本自定义规格的顺序）删除；

判断商品版本自定义规格数据中操作类型是否为update，若是，将商品版本自定义规格数据更新到数据库；

判断商品版本自定义规格数据中操作类型是否为insert，若是，则生成uuid，将商品版本自定义规格数据新增到数据库；

如果商品版本自定义规格数据中商品版本自定义规格选项列表不为空，遍历商品版本自定义规格选项列表；

判断商品版本自定义规格选项数据中操作类型是否为delete，若是，并且商品版本自定义规格选项数据的uuid不为空，则将商品版本自定义规格选项数据删除；

判断商品版本自定义规格选项数据中操作类型是否为update，若是，将商品版本自定义规格选项数据更新到数据库；

判断商品版本自定义规格选项数据中操作类型是否为insert，若是，未生成uuid，将商品版本自定义规格数据新增到数据库；

⑨，如果商品按量数据列表不为空，遍历商品按按量数据列表；

判断商品按量数据中操作类型是否为delete，若是，并且商品按量数据的uuid不为空，则将商品按量数据删除；

否则，将商品按量数据插入或更新到数据库；

⑩，如果商品套餐数据列表不为空，遍历商品套餐数据列表；

判断商品套餐数据中操作类型是否为delete，若是，并且商品套餐数据的uuid不为空，则将商品套餐相关数据（按商品套餐自定义选项，自定义套餐包资源名称 ，商品套餐的顺序）删除；

判断商品套餐数据中操作类型是否为update，若是，将商品套餐数据更新到数据库；

判断商品套餐数据中操作类型是否为insert，若是，则生成uuid，将商品套餐数据新增到数据库；

如果商品套餐数据中自定义套餐包资源名称列表不为空，遍历自定义套餐包资源名称列表；

判断自定义套餐包资源名称数据中操作类型是否为delete，若是，并且自定义套餐包资源名称数据的uuid不为空，则将自定义套餐包资源名称相关数据（按商品套餐自定义选项，自定义套餐包资源名称的顺序）删除；

判断自定义套餐包资源名称数据中操作类型是否为update，若是，将自定义套餐包资源名称数据更新到数据库；

判断自定义套餐包资源名称数据中操作类型是否为insert，若是，则生成uuid，将自定义套餐包资源名称数据新增到数据库；

如果自定义套餐包资源名称数据中商品套餐自定义选项列表不为空，遍历商品套餐自定义选项列表；

判断商品套餐自定义选项数据中操作类型是否为delete，若是，并且商品套餐自定义选项数据的uuid不为空，则将商品套餐自定义选项数据删除；

判断商品套餐自定义选项数据中操作类型是否为update，若是，将商品套餐自定义选项数据更新到数据库；

判断商品套餐自定义选项数据中操作类型是否为insert，若是，未生成uuid，将商品套餐自定义选项数据新增到数据库；

⑪，如果商品自定义权限数据列表不为空，遍历商品套餐数据列表；

判断商品自定义权限数据中操作类型是否为delete，若是，并且商品自定义权限数据的uuid不为空，则将商品自定义权限数据删除；

判断商品自定义权限数据中操作类型是否为insert，若是，未生成uuid，将商品自定义权限数据新增到数据库；

判断商品自定义权限数据中操作类型是否为update，若是，将商品自定义权限数据更新到数据库；

⑫，在整个过程中，若出现任何异常，抛出异常提示“修改商品失败”。

## 1.6、商品列表：sobeyMallProduct/V1/products/list（get）

### 1.6.1、逻辑流程

①，参数：

Product：商品类

②，根据Product实例中封装的参数查询商品列表，将商品列表返回到前端。

## 1.7、商品查询：sobeyMallProduct/V1/products/{uuid}（get）

### 1.7.1、逻辑流程

①，参数：

uuid：商品id

②，将uuid封装到Product实例中，根据Product实例查询商品列表，将商品列表返回到前端。

## 1.8、商品统计：sobeyMallProduct/V1/products/statistic（get）

### 1.8.1、逻辑流程

①，参数：无

②，直接查询数据库，根据商品状态（上架或者未上架）进行分组查询出不通状态下商品的数量，封装到hashmap中，返回到前端。

# 2、商品素材相关接口

## 2.1、商品素材查询：sobeyMallProduct/V1/medias（get）

### 2.1.1、逻辑流程

①，参数：   
 Media：商品素材类

②，判断Media实例是否为空，若为空，直接返回空；

③，根据传递的参数查询出商品素材列表封装在Page类中，返回到前端。

## 2.2、商品素材新增：sobeyMallProduct/V1/medias（post）

### 2.2.1、逻辑流程

1. 参数：  
   Media：商品素材类  
   destPath：主目录
2. 判断Media实例是否为空，为空抛出自定义异常提示“商品素材不能为空”；
3. 判断商品id和素材id是否为空，若为空，抛出自定义异常提示“productId和mediaId必须传递”；
4. 对上传的文件进行储存和新增素材信息，判断上传的文件是否是单个文件，若不是，就抛出异常提示“一次只能上传一个文件”；
5. 储存文件；判断文件是否为空，为空直接返回null；  
     
   为防止文件重名，在父路径destPath下再增加一级uuid目录，使每个文件储存在不同的文件夹下，最终的文件路径为destPath+uuid+OriginalFilename，判断文件父路径是否存在，若不存在，则生成文件父路径；  
     
   最后进行文件上传；
6. 如果存储文件出现异常，则抛出异常提示“文件上传报错”。
7. 新增素材信息；判断储存文件时返回的储存路径是否为空，若为空直接返回null；生成素材的uuid，将素材相关数据新增到素材表；
8. 最后将素材的uuid返回给前端。

## 2.3、商品素材分页查询：sobeyMallProduct/V1/medias/page（post）

### 2.3.1、逻辑流程

①，参数：  
 offset：页码  
 limit：每页条数  
 Media：素材类

②，将offset与limit设置在Page类中；

③，根据传递的参数查询出素材列表封装在Page类中，返回到前端。

## 2.4、商品素材删除：sobeyMallProduct/V1/medias/{uuid}（delete）

### 2.4.1、逻辑流程

①，参数：  
 uuid：素材uuid  
②，将uuid封装到media实例中，根据media查询数据库，确认素材信息是否存在，若存在，则删除素材记录，并将与素材关联的文件进行删除。

# 3、商品类别相关接口

## 3.1、商品分类列表查询：sobeyMallProduct/V1/categorys/list（post）

### 3.1.1、逻辑流程

①，参数：   
 Category：商品类别类

②，设置查询结果按order（顺序）字段进行正序；

③，根据传递的参数查询出商品类别列表，返回到前端。

# 4、商品亮点相关接口

## 4.1、亮点预新增：sobeyMallProduct/V1/bright-spot/pre（get）

### 4.1.1、逻辑流程

①，参数：无

②，生成随机uuid并封装在BrightSpot实例中，返回到前端。

# 5、商品按量相关接口

## 5.1、按量预新增：sobeyMallProduct/V1/metric/pre（get）

### 5.1.1、逻辑流程

①，参数：无

②，生成随机uuid并封装在Metric实例中，返回到前端。

## 5.2、按量类型查询：sobeyMallProduct/V1/metric/type（get）

### 5.2.1、逻辑流程

①，参数：无

②，将所有按量计费的类型转换为json串返回到前端。

# 6、商品折扣相关接口

## 6.1、折扣分页查询：sobeyMallProduct/V1/discount/page（post）

### 6.1.1、逻辑流程

①，参数：

page：页码

size：每页条数

Discount：折扣类

②，将page与size设置在Page类中；

③，根据传递的参数（封装在Discount实例中）查询出折扣列表封装在Page类中，返回到前端。

## 6.2、新增折扣：sobeyMallProduct/V1/discount（post）

### 6.2.1、逻辑流程

①，参数：

Discount：折扣类

BindingResult：spring自带的参数校验结果类

②，检验BindingResult实例中是否寻在错误，若存在，将错误信息拼接后封装到自定义异常抛出，向前端提示相关错误信息；

③，校验该用户是否已经存在的未过期的将要新增折扣的商品的折扣信息，若存在，则抛出自定义异常提示“一个用户对一个商品只能有一个折扣信息,商品名称为:xxx”；

④，生成requestUuid；遍历商品id，新建Discount实例，将前端传递的参数封装到Discount实例中，将折扣状态设置为“未生效”，设置requestUuid，设置折扣（设置前将折扣保留四位小数）；向数据库新增折扣信息；  
  
⑤，新建DiscountRequestInfo实例，将上一步生成的requestUuid设置成uuid，设置请求类型为insert，将DiscountRequestInfo实例新增到折扣协议表；

⑥，远程请求消息服务的发送普通文本消息（系统消息）接口，告知用户“用户您好，您所申请的折扣已生成，请查看详情，确认后生效！”。

## 6.3、修改折扣：sobeyMallProduct/V1/discount（patch）

### 6.3.1、逻辑流程

①，参数：

Discount：折扣类

②，检验参数中的uuid、userCode、productId不能为空，任一参数为空就抛出异常提示参数为空；

③，根据uuid查询数据库是否存在此这折扣信息，若不存在，抛出异常提示“数据不存在，请核对”；

④，处理折扣（保留4位小数）；将折扣的状态设置为未生效；生成并设置新的requestUuid；将折扣信息更新到数据库；

⑤，新建DiscountRequestInfo实例，将上一步新生成的requestUuid设置为uuid，请求类型为update，将DiscountRequestInfo实例新增到折扣协议表；

⑥，远程请求消息服务的发送普通文本消息接口，告知用户“用户您好，您所申请的折扣已生成，请查看详情，确认后生效！”。

## 6.4、删除折扣：sobeyMallProduct/V1/discount/{uuid}（delete）

### 6.4.1、逻辑流程

①，参数：

Uuid：折扣uuid

②，根据uuid查询数据库，若查询结果为空，则抛出异常提示“数据不存在,请核对”；

③，根据uuid删除折扣信息；

④，根据查询出来的userCode和productId生成key，在redis中将对应的key删除。

## 6.5、生效折扣：sobeyMallProduct/V1/discount/protocal/{uuid}（post）

### 6.5.1、逻辑流程

①，参数：

Uuid：折扣协议uuid

②，根据uuid查询数据库，若查询结果为空，则抛出异常提示“数据不存在,请确认”；

③，判断上一步查询结果中折扣状态是否是已确认，若是，抛出异常提示“此协议已经确认过”；

④，将uuid作为requestUuid字段查询折扣表得到折扣集合，遍历折扣集合；将数据库折扣状态改为2（已确认）；将userCode和productId的组合作为key，将折扣信息的json串作为value存到redis中；

⑤，根据uuid将相关的折扣协议信息的状态更新为2（生效）。

## 6.6、协议查看：sobeyMallProduct/V1/discount/protocal/{uuid}（get）

### 6.6.1、逻辑流程

①，参数：

Uuid：折扣协议uuid

②，根据uuid查询协议信息，若查询结果为空，则抛出异常提示“数据不存在,请确认”；

③，判断协议的状态是否为生效，若是，则直接返回协议内容到前端；

④，将uuid作为折扣信息的requestUuid查询出折扣信息集合，遍历集合，根据折扣信息中商品id查询出商品信息，再按格式“产品：xxx，折扣：xxx折”生成折扣字符串信息；

⑤，根据④中查询出的折扣信息中用户id和用户站点编码，查询出用户的信息；

⑥，将④中查询出的折扣信息中expirationDate（折扣过期时间）转化为格式为"yyyy年MM月dd日 HH时mm分ss秒"的字符串，将当前时间转化为格式为“2018年10月24日”的字符串；

⑦，查询出折扣协议模板（目前只有一个模板）；

⑧，将协议中的占位符进行替换得到完整协议；

1. 新建DiscountRequestInfo实例，设置uuid，设置⑧中生成的协议，协议状态设置为1（未生效），设置协议编号；将折扣协议更新到数据库；
2. 将协议内容返回到前端。