

苍穹外卖day4

完成套餐管理模块所有业务功能，包括：

- 新增套餐
- 套餐分页查询
- 删除套餐
- 修改套餐
- 起售停售套餐

要求：

1. 根据产品原型进行需求分析，分析出业务规则
2. 设计接口
3. 梳理表之间的关系（分类表、菜品表、套餐表、口味表、套餐菜品关系表）
4. 根据接口设计进行代码实现
5. 分别通过swagger接口文档和前后端联调进行功能测试

1. 新增套餐

1.1 需求分析和设计

产品原型：

*套餐名称:

*套餐分类:

*套餐价格:

元

*套餐菜品:

1 添加菜品

菜品名称	原价	份数			操作
鱼香肉丝	¥44.00	-	1	+	删除

2 *套餐图片:

图片大小不超过2M
仅能上传 PNG JPEG JPG类型图片

套餐描述:

菜品描述, 最长200字

保存

保存并继续添加菜品

返回

添加菜品

请输入菜品名称进行搜索

荤菜

素菜

<input type="checkbox"/>	唐酥梨集	在售	¥40.00
<input type="checkbox"/>	唐酥梨集	在售	¥40.00

已选菜品 (1)

西红柿炒鸡蛋

¥15.00

添加

取消

业务规则：

- 套餐名称唯一
- 套餐必须属于某个分类
- 套餐必须包含菜品
- 名称、分类、价格、图片为必填项
- 添加菜品窗口需要根据分类类型来展示菜品
- 新增的套餐默认为停售状态

接口设计（共涉及到4个接口）：

- 根据类型查询分类（已完成）
- 根据分类id查询菜品
- 图片上传（已完成）
- 新增套餐

根据分类id查询菜品

基本信息

Path: /admin/dish/list

Method: GET

接口描述:

请求参数

Query

参数名称	是否必须	示例	备注
categoryId	是	101	分类id

返回数据

名称	类型	是否必须	默认值	备注	其他信息
code	integer	必须			format: int32
data	object []	非必须			item 类型: object
├ categoryId	integer	非必须			format: int64
├ createTime	string	非必须			format: date-time
├ createUser	integer	非必须			format: int64
├ description	string	非必须			
├ id	integer	非必须			format: int64
├ image	string	非必须			
├ name	string	非必须			
├ price	number	非必须			
├ status	integer	非必须			format: int32
├ updateTime	string	非必须			format: date-time
├ updateUser	integer	非必须			format: int64
msg	string	非必须			

新增套餐

基本信息

Path: /admin/setmeal

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

Body

名称	类型	是否必须	默认值	备注	其他信息
categoryId	integer	必须		分类id	format: int64
description	string	非必须		套餐描述	
id	integer	非必须		套餐id	format: int64
image	string	必须		套餐图片	
name	string	必须		套餐名称	
price	number	必须		套餐价格	
setmealDishes	object []	必须		套餐包含的菜品	item 类型: object
└─ copies	integer	必须		份数	format: int32
└─ dishId	integer	必须		菜品id	format: int64
└─ id	integer	非必须		套餐和菜品关系id	format: int64
└─ name	string	必须		菜品名称	
└─ price	number	必须		菜品价格	
└─ setmealId	integer	必须		套餐id	format: int64
status	integer	必须		套餐状态: 1位起售 0为停售	format: int32

返回数据

名称	类型	是否必须	默认值	备注	其他信息
code	integer	必须			format: int32
data	object	非必须			
msg	string	非必须			

数据库设计：

setmeal表为套餐表

```
1 CREATE TABLE `setmeal` (  
2   `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '主键',  
3   `category_id` bigint(20) NOT NULL COMMENT '菜品分类id',  
4   `name` varchar(32) CHARACTER SET utf8 COLLATE utf8_bin NOT NULL COMMENT '套餐名称',  
5   `price` decimal(10,2) NOT NULL COMMENT '套餐价格',  
6   `status` int(11) DEFAULT '1' COMMENT '售卖状态 0:停售 1:起售',
```

```

7      `description` varchar(255) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT
      NULL COMMENT '描述信息',
8      `image` varchar(255) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL
      COMMENT '图片',
9      `create_time` datetime DEFAULT NULL COMMENT '创建时间',
10     `update_time` datetime DEFAULT NULL COMMENT '更新时间',
11     `create_user` bigint(20) DEFAULT NULL COMMENT '创建人',
12     `update_user` bigint(20) DEFAULT NULL COMMENT '修改人',
13     PRIMARY KEY (`id`),
14     UNIQUE KEY `idx_setmeal_name` (`name`)
15 ) ENGINE=InnoDB AUTO_INCREMENT=32 DEFAULT CHARSET=utf8 COLLATE=utf8_bin
      COMMENT='套餐';

```

setmeal_dish表为套餐菜品关系表

```

1 CREATE TABLE `setmeal_dish` (
2     `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '主键',
3     `setmeal_id` bigint(20) DEFAULT NULL COMMENT '套餐id',
4     `dish_id` bigint(20) DEFAULT NULL COMMENT '菜品id',
5     `name` varchar(32) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL COMMENT
      '菜品名称（冗余字段）',
6     `price` decimal(10,2) DEFAULT NULL COMMENT '菜品单价（冗余字段）',
7     `copies` int(11) DEFAULT NULL COMMENT '菜品份数',
8     PRIMARY KEY (`id`)
9 ) ENGINE=InnoDB AUTO_INCREMENT=47 DEFAULT CHARSET=utf8 COLLATE=utf8_bin
      COMMENT='套餐菜品关系';

```

1.2 代码实现

1.2.1 DishController

```

1      /**
2       * 根据分类id查询菜品
3       * @param categoryId
4       * @return
5       */
6      @GetMapping("/list")
7      @ApiOperation("根据分类id查询菜品")
8      public Result<List<Dish>> list(@RequestParam Long categoryId){
9          //当Controller方法的参数名与请求中的参数名一致时，可以不用@RequestParam注解。
10         List<Dish> list = dishService.list(categoryId);
11         return Result.success(list);
12     }

```

1.2.2 DishService

```
1  /**
2      * 根据分类id查询菜品
3      * @param categoryId
4      * @return
5  */
6  List<Dish> list(Long categoryId);
```

1.2.3 DishServiceImpl

```
1  /**
2      * 根据分类id查询菜品
3      * @param categoryId
4      * @return
5  */
6  public List<Dish> list(Long categoryId) {
7      Dish dish = Dish.builder()
8          .categoryId(categoryId)
9          .status(StatusConstant.ENABLE)
10         .build();
11     return dishMapper.list(dish);
12 }
```

1.2.4 DishMapper

```
1  /**
2      * 动态条件查询菜品
3      * @param dish
4      * @return
5  */
6  List<Dish> list(Dish dish);
```

1.2.5 DishMapper.xml

在<if test="name != null">中通过OGNL表达式访问Dish对象的名字属性

```
1  <select id="list" resultType="Dish" parameterType="Dish">
2      select * from dish
3      <where>
4          <if test="name != null">
5              and name like concat('%',#{name},'%')
6          </if>
7          <if test="categoryId != null">
8              and category_id = #{categoryId}
9          </if>
10         <if test="status != null">
11             and status = #{status}
12         </if>
13     </where>
14     order by create_time desc
15 </select>
```

1.2.6 SetmealController

sky-server/src/main/java/com/sky/controller/admin/SetmealController.java

```
1  package com.sky.controller.admin;
2
3  import com.sky.dto.SetmealDTO;
4  import com.sky.result.Result;
5  import com.sky.service.SetmealService;
6  import io.swagger.annotations.Api;
7  import io.swagger.annotations.ApiOperation;
8  import lombok.extern.slf4j.Slf4j;
9  import org.springframework.beans.factory.annotation.Autowired;
10 import org.springframework.web.bind.annotation.PostMapping;
11 import org.springframework.web.bind.annotation.RequestBody;
12 import org.springframework.web.bind.annotation.RequestMapping;
13 import org.springframework.web.bind.annotation.RestController;
14
15 /**
16  * 套餐管理
17  */
18 @RestController
19 @RequestMapping("/admin/setmeal")
20 @Api(tags = "套餐相关接口")
21 @Slf4j
22 public class SetmealController {
23
24     @Autowired
25     private SetmealService setmealService;
26
27     /**
28      * 新增套餐
29      * @param setmealDTO
30      * @return
31      */
32     @PostMapping
33     @ApiOperation("新增套餐")
34     public Result save(@RequestBody SetmealDTO setmealDTO) {
35         setmealService.saveWithDish(setmealDTO);
36         return Result.success();
37     }
38 }
```

1.2.7 SetmealService

sky-server/src/main/java/com/sky/service/SetmealService.java

```

1 package com.sky.service;
2
3 import com.sky.dto.SetmealDTO;
4
5 public interface SetmealService {
6     /**
7      * 新增套餐，同时需要保存套餐和菜品的关联关系
8      * @param setmealDTO
9      */
10    void savewithDish(SetmealDTO setmealDTO);
11 }

```

1.2.8 SetmealServiceImpl

sky-server/src/main/java/com/sky/service/impl/SetmealServiceImpl.java

```

1 package com.sky.service.impl;
2
3 import com.sky.dto.SetmealDTO;
4 import com.sky.entity.Setmeal;
5 import com.sky.entity.SetmealDish;
6 import com.sky.mapper.DishMapper;
7 import com.sky.mapper.SetmealDishMapper;
8 import com.sky.mapper.SetmealMapper;
9 import com.sky.service.SetmealService;
10 import lombok.extern.slf4j.Slf4j;
11 import org.springframework.beans.BeanUtils;
12 import org.springframework.beans.factory.annotation.Autowired;
13 import org.springframework.stereotype.Service;
14 import org.springframework.transaction.annotation.Transactional;
15
16 import java.util.List;
17
18 /**
19  * 套餐业务实现
20  */
21 @Service
22 @Slf4j
23 public class SetmealServiceImpl implements SetmealService {
24
25     @Autowired
26     private SetmealMapper setmealMapper;
27     @Autowired
28     private SetmealDishMapper setmealDishMapper;
29     @Autowired
30     private DishMapper dishMapper;
31
32     /**
33      * 新增套餐，同时需要保存套餐和菜品的关联关系
34      * @param setmealDTO
35      */
36     @Transactional
37     public void savewithDish(SetmealDTO setmealDTO) {
38         Setmeal setmeal = new Setmeal();
39         BeanUtils.copyProperties(setmealDTO, setmeal);

```



```

40
41     //向套餐表插入数据
42     setmealMapper.insert(setmeal);
43
44     //获取生成的套餐id
45     Long setmealId = setmeal.getId();
46
47     List<SetmealDish> setmealDishes = setmealDTO.getSetmealDishes();
48     setmealDishes.forEach(setmealDish -> {
49         setmealDish.setSetmealId(setmealId); //套餐菜品关系设置套餐id
50     });
51
52     //保存套餐和菜品的关联关系
53     setmealDishMapper.insertBatch(setmealDishes);
54 }
55 }

```

1.2.9 SetmealMapper

```

1  /**
2      * 新增套餐
3      * @param setmeal
4  */
5  @AutoFill(OperationType.INSERT)
6  void insert(Setmeal setmeal);

```

1.2.10 SetmealMapper.xml

sky-server/src/main/resources/mapper/SetmealMapper.xml

```

1  <?xml version="1.0" encoding="UTF-8" ?>
2  <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
3      "http://mybatis.org/dtd/mybatis-3-mapper.dtd" >
4  <mapper namespace="com.sky.mapper.SetmealMapper">
5
6      <insert id="insert" parameterType="Setmeal" useGeneratedKeys="true"
7          keyProperty="id">
8          insert into setmeal
9              (category_id, name, price, status, description, image, create_time,
10             update_time, create_user, update_user)
11             values (#{categoryId}, #{name}, #{price}, #{status}, #{description},
12                 #{image}, #{createTime}, #{updateTime},
13                 #{createUser}, #{updateUser})
14      </insert>
15  </mapper>

```

1.2.11 SetmealDishMapper

```

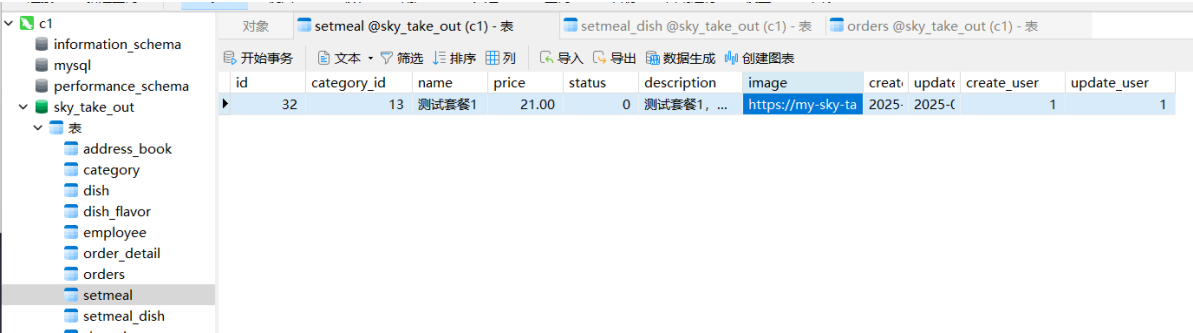
1  /**
2      * 批量保存套餐和菜品的关联关系
3      * @param setmealDishes
4  */
5  void insertBatch(List<SetmealDish> setmealDishes);

```

1.2.12 SetmealDishMapper.xml

使用小写 `list` 更符合MyBatis官方文档的推荐

```
1 <insert id="insertBatch" parameterType="list">
2     insert into setmeal_dish
3     (setmeal_id,dish_id,name,price,copies)
4     values
5     <foreach collection="setmealDishes" item="sd" separator=",">
6         (#{sd.setmealId},#{sd.dishId},#{sd.name},#{sd.price},#{sd.copies})
7     </foreach>
8 </insert>
```



2. 套餐分页查询

2.1 需求分析和设计

产品原型：

套餐名称

套餐分类

售卖状态

请选择

搜索

批量删除

新建套餐

<input type="checkbox"/>	套餐名称	套餐图片	套餐分类	套餐价	售卖状态	最后操作时间	操作
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	停售	2019-02-02 11:11	修改 删除 启售
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	启售	2019-02-02 11:11	修改 删除 停售
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	启售	2019-02-02 11:11	修改 删除 停售
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	启售	2019-02-02 11:11	修改 删除 停售
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	启售	2019-02-02 11:11	修改 删除 停售
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	启售	2019-02-02 11:11	修改 删除 停售
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	启售	2019-02-02 11:11	修改 删除 停售
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	启售	2019-02-02 11:11	修改 删除 停售
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	启售	2019-02-02 11:11	修改 删除 停售
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	启售	2019-02-02 11:11	修改 删除 停售

总共 85 个项目

12345

业务规则：

- 根据页码进行分页展示
- 每页展示10条数据
- 可以根据需要，按照套餐名称、分类、售卖状态进行查询

接口设计：

基本信息

Path: /admin/setmeal/page

Method: GET

接口描述：

请求参数

Query

参数名称	是否必须	示例	备注
categoryId	否		分类id
name	否		套餐名称
page	是		页码
pageSize	是		每页记录数
status	否		套餐起售状态

返回数据

名称	类型	是否必须	默认值	备注	其他信息
code	number	必须			
msg	null	非必须			
data	object	非必须			
└─ total	number	非必须			
└─ records	object []	非必须			item 类型: object
└─ id	number	必须			
└─ categoryId	number	必须			
└─ name	string	必须			
└─ price	number	必须			
└─ status	number	必须			
└─ description	string	必须			
└─ image	string	必须			
└─ updateTime	string	必须			
└─ categoryName	string	必须			

2.2 代码实现

2.2.1 SetmealController

```
1  /**
2      * 分页查询
3      * @param setmealPageQueryDTO
4      * @return
5  */
6  @GetMapping("/page")
7  @ApiOperation("分页查询")
8  public Result<PageResult> page(SetmealPageQueryDTO setmealPageQueryDTO) {
9      PageResult pageResult = setmealService.pageQuery(setmealPageQueryDTO);
10     return Result.success(pageResult);
11 }
```

2.2.2 SetmealService

```
1  /**
2      * 分页查询
3      * @param setmealPageQueryDTO
4      * @return
5  */
6  PageResult pageQuery(SetmealPageQueryDTO setmealPageQueryDTO);
```

2.2.3 SetmealServiceImpl

```
1  /**
2      * 分页查询
3      * @param setmealPageQueryDTO
4      * @return
5  */
6  public PageResult pageQuery(SetmealPageQueryDTO setmealPageQueryDTO) {
7      int pageNum = setmealPageQueryDTO.getPage();
8      int pageSize = setmealPageQueryDTO.getPageSize();
9
10     PageHelper.startPage(pageNum, pageSize);
11     Page<SetmealVO> page = setmealMapper.pageQuery(setmealPageQueryDTO);
12     return new PageResult(page.getTotal(), page.getResult());
13 }
```

2.2.4 SetmealMapper

```
1  /**
2      * 分页查询
3      * @param setmealPageQueryDTO
4      * @return
5  */
6  Page<SetmealVO> pageQuery(SetmealPageQueryDTO setmealPageQueryDTO);
```

2.2.5 SetmealMapper.xml

```
1  <select id="pageQuery" resultType="com.sky.vo.SetmealVO">
2      select
3          s.*,c.name categoryName
4      from
5          setmeal s
6      left join
7          category c
8      on
9          s.category_id = c.id
10     <where>
11         <if test="name != null">
12             and s.name like concat('%',{name},%')
13         </if>
14         <if test="status != null">
15             and s.status = #{status}
16         </if>
17         <if test="categoryId != null">
18             and s.category_id = #{categoryId}
19         </if>
20     </where>
21     order by s.create_time desc
22 </select>
```

套餐名称: 套餐分类: 售卖状态:

<input type="checkbox"/>	套餐名称	图片	套餐分类	套餐价	售卖状态	最后操作时间	操作
<input type="checkbox"/>	测试套餐1		人气套餐	¥ 21	● 停售	2025-05-20 10:29	修改 删除 启售

3. 删除套餐

3.1 需求分析和设计

产品原型:

1套餐名称

套餐分类

售卖状态

请选择

搜索

批量删除

2新建套餐

<input type="checkbox"/>	套餐名称	套餐图片	套餐分类	套餐价	售卖状态	最后操作时间	3操作
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	停售	2019-02-02 11:11	修改删除启售4
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	启售	2019-02-02 11:11	修改删除停售
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	启售	2019-02-02 11:11	修改删除停售
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	启售	2019-02-02 11:11	修改删除停售
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	启售	2019-02-02 11:11	修改删除停售
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	启售	2019-02-02 11:11	修改删除停售
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	启售	2019-02-02 11:11	修改删除停售
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	启售	2019-02-02 11:11	修改删除停售
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	启售	2019-02-02 11:11	修改删除停售
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	启售	2019-02-02 11:11	修改删除停售

业务规则：

- 可以一次删除一个套餐，也可以批量删除套餐
- 起售中的套餐不能删除

接口设计：

基本信息

Path： /admin/setmeal

Method： DELETE

接口描述：

请求参数

Query

参数名称	是否必须	示例	备注
ids	是		ids

返回数据

名称	类型	是否必须	默认值	备注	其他信息
code	integer	非必须			format: int32
data	object	非必须			
msg	string	非必须			

3.2 代码实现

3.2.1 SetmealController

```
1  /**
2      * 批量删除套餐
3      * @param ids
4      * @return
5  */
6  @DeleteMapping
7  @ApiOperation("批量删除套餐")
8  public Result delete(@RequestParam List<Long> ids){
9      setmealService.deleteBatch(ids);
10     return Result.success();
11 }
```

3.2.2 SetmealService

```
1  /**
2      * 批量删除套餐
3      * @param ids
4  */
5  void deleteBatch(List<Long> ids);
```

3.2.3 SetmealServiceImpl

```
1  /**
2      * 批量删除套餐
3      * @param ids
4  */
5  @Transactional
6  public void deleteBatch(List<Long> ids) {
7      ids.forEach(id -> {
8          Setmeal setmeal = setmealMapper.getById(id);
9          if(StatusConstant.ENABLE == setmeal.getStatus()){
10             //起售中的套餐不能删除
11             throw new
12             DeletionNotAllowedException(MessageConstant.SETMEAL_ON_SALE);
13         }
14     });
15     ids.forEach(setmealId -> {
16         //删除套餐表中的数据
17         setmealMapper.deleteById(setmealId);
18         //删除套餐菜品关系表中的数据
19         setmealDishMapper.deleteBySetmealId(setmealId);
20     });
21 }
```

3.2.4 SetmealMapper

```
1  /**
2      * 根据id查询套餐
3      * @param id
4      * @return
5  */
6  @Select("select * from setmeal where id = #{id}")
7  Setmeal getById(Long id);
8
9  /**
10     * 根据id删除套餐
11     * @param setmealId
12  */
13  @Delete("delete from setmeal where id = #{id}")
14  void deleteById(Long setmealId);
```

3.2.5 SetmealDishMapper

```
1  /**
2      * 根据套餐id删除套餐和菜品的关联关系
3      * @param setmealId
4  */
5  @Delete("delete from setmeal_dish where setmeal_id = #{setmealId}")
6  void deleteBySetmealId(Long setmealId);
```

3.3 功能测试

略。

4. 修改套餐

4.1 需求分析和设计

产品原型：

*套餐名称:

*套餐分类:

*套餐价格:

元

*套餐菜品:

1 添加菜品

菜品名称	原价	份数			操作
鱼香肉丝	¥44.00	-	1	+	删除

*套餐图片:

图片大小不超过2M
仅能上传 PNG JPEG JPG类型图片

套餐描述:

菜品描述，最长200字

保存

保存并继续添加菜品

返回

接口设计（共涉及到5个接口）：

- 根据id查询套餐
- 根据类型查询分类（已完成）
- 根据分类id查询菜品（已完成）
- 图片上传（已完成）
- 修改套餐

根据id查询套餐

基本信息

Path: /admin/setmeal/{id}

Method: GET

接口描述:

请求参数

路径参数

参数名称	示例	备注
id	101	套餐id

修改套餐

基本信息

Path: /admin/setmeal

Method: PUT

接口描述:

请求参数

Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

4.2 代码实现

SetmealController.java

```
1  /**
2      * 根据id查询套餐，用于修改页面回显数据
3      *
4      * @param id
5      * @return
6  */
7  @GetMapping("/{id}")
8  @ApiOperation("根据id查询套餐")
9  public Result<SetmealVO> getById(@PathVariable Long id) {
10      SetmealVO setmealVO = setmealService.getByIdWithDish(id);
11      return Result.success(setmealVO);
12  }
13
14  /**
15      * 修改套餐
16      *
17      * @param setmealDTO
18      * @return
19  */
20  @PutMapping
21  @ApiOperation("修改套餐")
22  public Result update(@RequestBody SetmealDTO setmealDTO) {
23      setmealService.update(setmealDTO);
24      return Result.success();
25  }
```

SetmealService.java

```

1  /**
2      * 根据id查询套餐和关联的菜品数据
3      * @param id
4      * @return
5  */
6  SetmealVO getByIdWithDish(Long id);
7
8  /**
9      * 修改套餐
10     * @param setmealDTO
11 */
12 void update(SetmealDTO setmealDTO);

```

SetmealServiceImpl.java

有漏洞，可以手动在请求包里修改 SetmealDTO 的 status

```

1  /**
2      * 根据id查询套餐和套餐菜品关系
3      *
4      * @param id
5      * @return
6  */
7  public SetmealVO getByIdWithDish(Long id) {
8      Setmeal setmeal = setmealMapper.getById(id);
9      List<SetmealDish> setmealDishes = setmealDishMapper.getBySetmealId(id);
10
11     SetmealVO setmealVO = new SetmealVO();
12     BeanUtils.copyProperties(setmeal, setmealVO);
13     setmealVO.setSetmealDishes(setmealDishes);
14
15     return setmealVO;
16 }
17
18 /**
19     * 修改套餐
20     *
21     * @param setmealDTO
22 */
23 //TODO 有漏洞，可以手动在请求包里修改`SetmealDTO`的`status`，但是应该管理员才能修改菜品，暂且忽略
24 @Transactional
25 public void update(SetmealDTO setmealDTO) {
26     Setmeal setmeal = new Setmeal();
27     BeanUtils.copyProperties(setmealDTO, setmeal);
28
29     //1、修改套餐表，执行update

```

```

30     setmealMapper.update(setmeal);
31
32     //套餐id
33     Long setmealId = setmealDTO.getId();
34
35     //2、删除套餐和菜品的关联关系，操作setmeal_dish表，执行delete
36     setmealDishMapper.deleteBySetmealId(setmealId);
37
38     List<SetmealDish> setmealDishes = setmealDTO.getSetmealDishes();
39     setmealDishes.forEach(setmealDish -> {
40         setmealDish.setSetmealId(setmealId);
41     });
42     //3、重新插入套餐和菜品的关联关系，操作setmeal_dish表，执行insert
43     setmealDishMapper.insertBatch(setmealDishes);
44 }

```

SetmealMapper.java

```

1  /**
2   * 修改套餐
3   * @param setmeal
4   */
5  @AutoFill(OperationType.UPDATE)
6  void update(Setmeal setmeal);

```

SetmealMapper.xml

```

1  <update id="update" parameterType="Setmeal">
2      update setmeal
3      <set>
4          <if test="name != null"> name = #{name}, </if>
5          <if test="categoryId != null"> category_id = #{categoryId},
6      </if>
7          <if test="price != null"> price = #{price}, </if>
8          <if test="status != null"> status = #{status}, </if>
9          <if test="description != null"> description = #{description},
10     </if>
11          <if test="image != null"> image = #{image}, </if>
12          <if test="updateTime != null"> update_time = #{updateTime},
13     </if>
14          <if test="updateUser != null"> update_user = #{updateUser} </if>
15     </set>
16     where id = #{id}
17 </update>

```

SetmealDishMapper.java

```
1  /**
2   * 根据套餐id查询套餐和菜品的关联关系
3   * @param setmealId
4   * @return
5   */
6  @Select("select * from setmeal_dish where setmeal_id = #{setmealId}")
7  List<SetmealDish> getBySetmealId(Long setmealId);
```

4.3 功能测试

名称	原价	份数	操作
馒头	1	<div>-5+</div>	删除
米饭	2	<div>-1+</div>	删除
王老吉	6	<div>-1+</div>	删除

5. 起售停售套餐

5.1 需求分析和设计

产品原型：

套餐名称 请输入套餐名称

套餐分类 请输入套餐分类

售卖状态 请选择

搜索

批量删除

2 新建套餐

<input type="checkbox"/>	套餐名称	套餐图片	套餐分类	套餐价	售卖状态	最后操作时间	3 操作
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	停售	2019-02-02 11:11	修改 删除 停售 4
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	启售	2019-02-02 11:11	修改 删除 停售
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	启售	2019-02-02 11:11	修改 删除 停售
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	启售	2019-02-02 11:11	修改 删除 停售
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	启售	2019-02-02 11:11	修改 删除 停售
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	启售	2019-02-02 11:11	修改 删除 停售
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	启售	2019-02-02 11:11	修改 删除 停售
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	启售	2019-02-02 11:11	修改 删除 停售
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	启售	2019-02-02 11:11	修改 删除 停售
<input type="checkbox"/>	午餐优惠套餐		午餐套	¥222.00	启售	2019-02-02 11:11	修改 删除 停售

业务规则：

- 可以对状态为起售的套餐进行停售操作，可以对状态为停售的套餐进行起售操作
- 起售的套餐可以展示在用户端，停售的套餐不能展示在用户端
- 起售套餐时，如果套餐内包含停售的菜品，则不能起售

接口设计：

基本信息

Path: /admin/setmeal/status/{status}

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

路径参数

参数名称	示例	备注
status	1	套餐状态，1表示起售，0表示停售

Query

参数名称	是否必须	示例	备注
id	是	101	套餐id

返回数据

名称	类型	是否必须	默认值	备注	其他信息
code	integer	必须			format: int32
data	object	非必须			
msg	string	非必须			

5.2 代码实现

SetmealController.java

```

1  /**
2      * 套餐起售停售
3      * @param status
4      * @param id
5      * @return
6  */
7  @PostMapping("/status/{status}")
8  @ApiOperation("套餐起售停售")
9  public Result startOrStop(@PathVariable Integer status, Long id) {
10      setmealService.startOrStop(status, id);
11      return Result.success();
12  }

```

SetmealService.java

```

1  /**
2      * 套餐起售、停售
3      * @param status
4      * @param id
5  */
6  void startOrStop(Integer status, Long id);

```

SetmealServiceImpl.java

```

1  /**
2      * 套餐起售、停售
3      * @param status
4      * @param id
5  */
6  public void startOrStop(Integer status, Long id) { //起售时先起售菜品，再起售套餐
7      //起售套餐时，判断套餐内是否有停售菜品，有停售菜品提示"套餐内包含未启售菜品，无法启售"
8      if(status == StatusConstant.ENABLE){
9          //select a.* from dish a left join setmeal_dish b on a.id =
10         b.dish_id where b.setmeal_id = ?
11         List<Dish> dishList = dishMapper.getBySetmealId(id);
12         if(dishList != null && dishList.size() > 0){
13             dishList.forEach(dish -> {
14                 if(StatusConstant.DISABLE == dish.getStatus()){
15                     throw new
16                     SetmealEnableFailedException(MessageConstant.SETMEAL_ENABLE_FAILED);
17                 }
18             });
19         }
20         Setmeal setmeal = Setmeal.builder()
21             .id(id)
22             .status(status) //停售时，是先停售套餐再停售菜品，所以不影响这个状态设置
23             .build();
24         setmealMapper.update(setmeal);

```

DishMapper.java

```
1  /**
2      * 根据套餐id查询菜品
3      * @param setmealId
4      * @return
5  */
6  @Select("select a.* from dish a left join setmeal_dish b on a.id = b.dish_id
7      where b.setmeal_id = #{setmealId}")
8  List<Dish> getBySetmealId(Long setmealId);
```

5.3 功能测试

✔ 套餐状态已经更改成功!