

# 苍穹外卖day9

## 1. 用户端历史订单

### 1. 查询历史订单



## 业务规则

- 分页查询历史订单
- 可以根据订单状态查询
- 展示订单数据时，需要展示的数据包括：下单时间、订单状态、订单金额、订单明细（商品名称、图片）

## OrderController.java

```
1      /**
2       * 历史订单查询
3       * @param page
4       * @param pageSize
5       * @param status
6       * @return
7       * @throws Exception
8       */
9      @GetMapping("/historyOrders")
10     @ApiOperation("历史订单查询")
11     public Result<PageResult> historyOrders(int page,int pageSize,Integer
status) throws Exception {
12         PageResult
pageResult=ordersService.pageQuery4User(page,pageSize,status);
13         return Result.success(pageResult);
14     }
```

## OrderService.java

```
1      /**
2       * 历史订单查询
3       * @param page
4       * @param size
5       * @param status
6       * @return
7       */
8      PageResult pageQuery4User(int page, int size, Integer status);
```

## OrderServiceImpl.java

```
1      /**
2       * 用户端订单分页查询
3       *
4       * @param pageNum
5       * @param pageSize
6       * @param status
7       * @return
8       */
```

```

9      public PageResult pageQuery4User(int pageNum, int pageSize, Integer
status) {
10          // 设置分页
11          PageHelper.startPage(pageNum, pageSize);
12
13          OrdersPageQueryDTO ordersPageQueryDTO = new OrdersPageQueryDTO();
14          ordersPageQueryDTO.setUserId(BaseContext.getCurrentId());
15          ordersPageQueryDTO.setStatus(status);
16
17          // 分页条件查询
18          Page<Orders> page = orderMapper.pageQuery(ordersPageQueryDTO);
19
20          List<OrderVO> list = new ArrayList();
21
22          // 查询出订单明细，并封装入OrderVO进行响应
23          if (page != null && page.getTotal() > 0) {
24              for (Orders orders : page) {
25                  Long orderId = orders.getId();// 订单id
26
27                  // 查询订单明细
28                  List<OrderDetail> orderDetails =
orderDetailMapper.getByOrderId(orderId);
29
30                  OrderVO orderVO = new OrderVO();
31                  BeanUtils.copyProperties(orders, orderVO);
32                  orderVO.setOrderDetailList(orderDetails);
33
34                  list.add(orderVO);
35              }
36          }
37          return new PageResult(page.getTotal(), list);
38      }

```

#### OrderDetailMapper.java

```

1      /**
2       * 根据订单id查询订单明细
3       * @param orderId
4       * @return
5       */
6      @Select("select * from order_detail where order_id = #{orderId}")
7      List<OrderDetail> getByOrderId(Long orderId);

```

#### OrderMapper.java

```

1      /**
2       * 分页条件查询并按下单时间排序
3       * @param ordersPageQueryDTO
4       */
5      Page<Orders> pageQuery(OrdersPageQueryDTO ordersPageQueryDTO);

```

## OrderMapper.xml

```
1      <select id="pageQuery" resultType="com.sky.entity.Orders">
2          select * from orders
3          <where>
4              <if test="number != null and number!=''">
5                  and number like concat('%',{number},'%')
6              </if>
7              <if test="phone != null and phone!=''">
8                  and phone like concat('%',{phone},'%')
9              </if>
10             <if test="userId != null">
11                 and user_id = #{userId}
12             </if>
13             <if test="status != null">
14                 and status = #{status}
15             </if>
16             <if test="beginTime != null">
17                 and order_time >= #{beginTime}
18             </if>
19             <if test="endTime != null">
20                 and order_time <= #{endTime}
21             </if>
22         </where>
23         order by order_time desc
24     </select>
```

<

历史订单

...

🕒

全部订单

待付款

已取消

2025-05-29 10:24

待接单

¥ 352.00

共4件

馋嘴牛蛙

香锅牛蛙

金汤酸菜牛蛙

再来一单

催单

2025-05-28 11:25

已取消



¥ 591.00

共9件

金汤酸菜牛蛙

香锅牛蛙

测试套餐a1

馋嘴牛蛙

再来一单

2. 查询订单详情

订单已完成

申请退款

再来一单

苍穹外卖		
	【春夏单人套餐】龙井虾仁+米饭+茶	¥99.8
× 1		
	秋日螃蟹	¥79.8
× 1		
	老鸭汤	¥59.9
微辣 × 1		
展开更多		
打包费		¥7
配送费		¥6
		合计 ¥252.5

联系商家

期望时间	立即配送
配送地址	惠** 185****7664 传智播客办公楼（放置4层电梯口）

订单号码	67335643196380192
订单时间	2022-04-15 18:24:46
支付方式	在线支付
订单备注	多给加点糖汁，感谢！
餐具数量	商家按餐量提供

# 查询订单详情

**GET** /user/order/orderDetail/{id} • 开发中 ▾

 共享 | C端-订单接口

创建时间 2025年5月16日    修改时间 13 天前    修改者 狐友4

Mock >

## 请求参数

Path 参数

**id** string 订单id  
示例值: 101

## 返回响应

成功(200)

HTTP 状态码: 200    内容格式: JSON    application/json

数据结构

- code** integer <int32>
- ▶ **data** object OrderVO
- msg** string

```

1      /**
2       * 查询订单详情
3       *
4       * @param id
5       * @return
6       */
7      @GetMapping("/orderDetail/{id}")
8      @ApiOperation("查询订单详情")
9      public Result<OrderVO> details(@PathVariable("id") Long id) {
10         OrderVO orderVO = orderService.details(id);
11         return Result.success(orderVO);
12     }

```

#### OrderService.java

```

1      /**
2       * 查询订单详情
3       * @param id
4       * @return
5       */
6      OrderVO details(Long id);

```

#### OrderServiceImpl.java

```

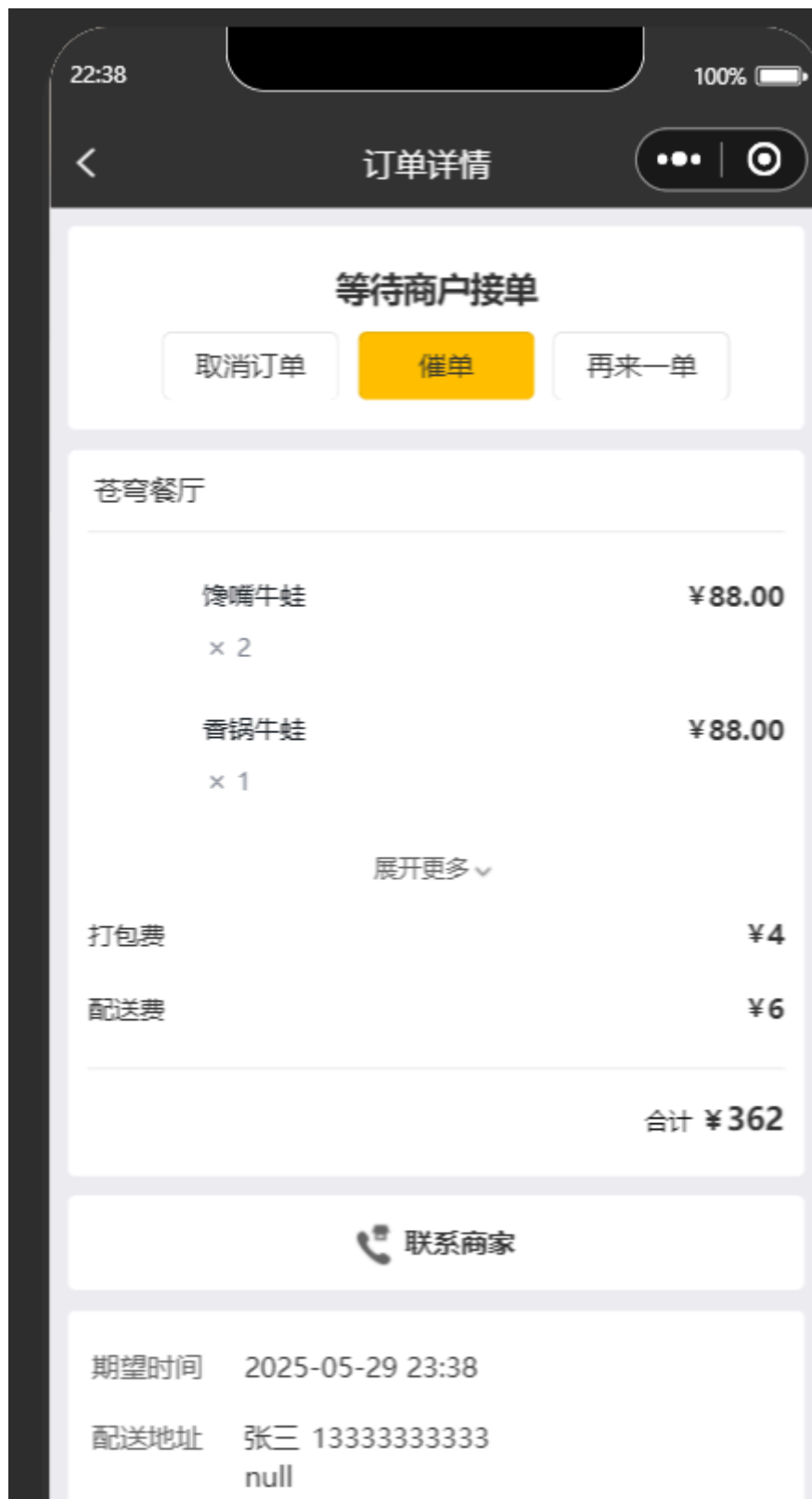
1      /**
2       * 查询订单详情
3       *
4       * @param id
5       * @return
6       */
7      @Override
8      public OrderVO details(Long id) {
9         // 根据id查询订单
10         Orders orders = orderMapper.getById(id);
11
12         // 查询该订单对应的菜品/套餐明细
13         List<OrderDetail> orderDetailList =
orderDetailMapper.getByOrderId(orders.getId());
14         OrderVO orderVO = new OrderVO();
15         BeanUtils.copyProperties(orders, orderVO);
16         orderVO.setOrderDetailList(orderDetailList);
17         return orderVO;
18     }

```

#### OrderMapper.java



```
1 /**
2  * 根据id查询订单
3  * @param id
4  */
5 @Select("select * from orders where id=#{id}")
6 Orders getById(Long id);
```



### 3. 取消订单



# 取消订单

PUT

/user/order/cancel/{id}

• 开发中

🌐 共享

C端-订单接口

创建时间 2025年5月16日

修改时间 13 天前

修改者 狐友4C0G

创建者

Mock >

## 请求参数

Path 参数
<div><div>id</div><div>string</div><div>订单id</div></div>
示例值: 101

## 返回响应

成功(200)

HTTP 状态码: 200      内容格式: JSON      application/json

数据结构

code

integer <int32>

data

object

msg

string

- 业务规则：
- 待支付和待接单状态下，用户可直接取消订单
  - 商家已接单状态下，用户取消订单需电话沟通商家
  - 派送中状态下，用户取消订单需电话沟通商家
  - 如果在待接单状态下取消订单，需要给用户退款

- 取消订单后需要将订单状态修改为“已取消”

#### OrderController.java

```
1      /**
2       * 用户取消订单
3       *
4       * @return
5       */
6      @PostMapping("/cancel/{id}")
7      @ApiOperation("取消订单")
8      public Result cancel(@PathVariable("id") Long id) throws Exception {
9          orderService.userCancelById(id);
10         return Result.success();
11     }
```

#### OrderService.java

```
1      /**
2       * 用户取消订单
3       * @param id
4       */
5      void userCancelById(Long id) throws Exception;
```

#### OrderServiceImpl.java

注释掉退款的调用

```
1      /**
2       * 用户取消订单
3       *
4       * @param id
5       */
6      public void userCancelById(Long id) throws Exception {
7          // 根据id查询订单
8          Orders ordersDB = orderMapper.getById(id);
9
10         // 校验订单是否存在
11         if (ordersDB == null) {
12             throw new
13             OrderBusinessException(MessageConstant.ORDER_NOT_FOUND);
14         }
15
16         //订单状态 1待付款 2待接单 3已接单 4派送中 5已完成 6已取消
17         if (ordersDB.getStatus() > 2) {
18             throw new
19             OrderBusinessException(MessageConstant.ORDER_STATUS_ERROR);
20         }
21
22         Orders orders = new Orders();
```

```
21     orders.setId(ordersDB.getId());
22
23     // 订单处于待接单状态下取消，需要进行退款
24     if (ordersDB.getStatus().equals(Orders.TO_BE_CONFIRMED)) {
25         //调用微信支付退款接口
26         //         weChatPayUtil.refund(
27         //             ordersDB.getNumber(), //商户订单号
28         //             ordersDB.getNumber(), //商户退款单号
29         //             new BigDecimal(0.01), //退款金额，单位 元
30         //             new BigDecimal(0.01)); //原订单金额
31
32         //支付状态修改为 退款
33         orders.setPayStatus(Orders.REFUND);
34     }
35
36     // 更新订单状态、取消原因、取消时间
37     orders.setStatus(Orders.CANCELLED);
38     orders.setCancelReason("用户取消");
39     orders.setCancelTime(LocalDateTime.now());
40     orderMapper.update(orders);
41 }
```



订单已取消

再来一单



您的订单已退款成功。

苍穹餐厅

馋嘴牛蛙

¥88.00

× 2

香锅牛蛙

¥88.00

× 1

展开更多

打包费

¥4

配送费

¥6

合计 ¥362



联系商家

#### 4. 再来一单

9:41

订单详情

订单已完成

申请退款

再来一单

苍穹外卖

【春夏单人套餐】龙井虾仁+米饭+茶

¥99.8

×1

秋日螃蟹

¥79.8

×1

老鸭汤

¥59.9

酸辣

×1

展开更多

打包费

¥7

配送费

¥6

合计 ¥252.5

联系商家

期望时间

立即配送

配送地址

惠\*\* 185\*\*\*\*7664  
传智播客办公楼（放置4层电梯口）

订单号码

67335643196380192

订单时间

2022-04-15 18:24:46

支付方式

在线支付

订单备注

多给加点糖汁，感谢！

# 再来一单

POST /user/order/repetition/{id}

• 开发中

共享

C端-订单接口

创建时间

2025年5月16日

修改时间

13 天前

修改者

狐友4C0G

创建者

Mock >

## 请求参数

Path 参数	
id	string 订单id
示例值:	101
Body 参数 text/plain	

## 返回响应

成功(200)		
HTTP 状态码:	200	内容格式: JSON application/json
数据结构		
code	integer <int32>	
data	object	
msg	string	



## OrderController.java

```
1  /**
2   * 再来一单
3   *
4   * @param id
5   * @return
6   */
7  @PostMapping("/repetition/{id}")
8  @ApiOperation("再来一单")
9  public Result repetition(@PathVariable Long id) {
10     orderService.repetition(id);
11     return Result.success();
12 }
```

## OrderService.java

```
1  /**
2   * 再来一单
3   *
4   * @param id
5   */
6  void repetition(Long id);
```

## OrderServiceImpl.java

```
1  /**
2   * 再来一单
3   *
4   * @param id
5   */
6  public void repetition(Long id) {
7      // 查询当前用户id
8      Long userId = BaseContext.getCurrentId();
9
10     // 根据订单id查询当前订单详情
11     List<OrderDetail> orderDetailList =
12         orderDetailMapper.getByOrderId(id);
13
14     // 将订单详情对象转换为购物车对象
15     List<ShoppingCart> shoppingCartList = orderDetailList.stream().map(x
16     -> {
17         ShoppingCart shoppingCart = new ShoppingCart();
18
19         // 将原订单详情里面的菜品信息重新复制到购物车对象中
20         BeanUtils.copyProperties(x, shoppingCart, "id");//忽略id属性
21         shoppingCart.setUserId(userId);
22         shoppingCart.setCreateTime(LocalDateTime.now());
23
24         return shoppingCart;
25     }).collect(Collectors.toList());
```

```

24         //stream()将orderDetailList（订单详情列表）转换为流（Stream），支持链式操作。
25         //map()对流中每个元素进行转换操作。
26         //collect()将流结果收集为新的集合（这里是List<ShoppingCart>）。
27
28         // 将购物车对象批量添加到数据库
29         shoppingCartMapper.insertBatch(shoppingCartList);
30     }

```

## ShoppingCartMapper.java

```

1     /**
2      * 批量插入购物车数据
3      *
4      * @param shoppingCartList
5      */
6     void insertBatch(List<ShoppingCart> shoppingCartList);

```

## ShoppingCartMapper.xml

```

1     <insert id="insertBatch" parameterType="list">
2         insert into shopping_cart
3         (name, image, user_id, dish_id, setmeal_id, dish_flavor, number,
4         amount, create_time)
5         values
6         <foreach collection="shoppingCartList" item="sc" separator=",">
7             ({sc.name},{sc.image},{sc.userId},{sc.dishId},{
8             {sc.setmealId},{sc.dishFlavor},{sc.number},{sc.amount},{sc.createTime}}
9         </foreach>
10    </insert>

```

## 2. 商家端订单管理

### 1. 订单搜索

苍穹外卖LOGO

工作台

订单管理

分类管理

菜品管理

套餐管理

数据统计

员工管理

营业中

营业状态

99+4

管理员

订单管理

全部订单

待接单(25)

待派送(50)

派送中(30)

已完成

已取消

1

订单号:手机号:下单时间:20220331 00:00-20220431 00:00

查询

重置

订单号	3	订单状态	用户名	手机号	地址	下单时间	实收金额	2	操作
2021010200001		待付款	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00		取消 查看
2021010200001		待接单	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00		接单 拒单 查看
2021010200001		待派送	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00		派送 取消 查看
2021010200001		已完成	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00		退单 查看
2021010200001		派送中	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00		完成 取消 查看
2021010200001		已取消	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00		查看

<

1

2

3

4

5

6

7

8

9

>

10条/页

跳至

5

页

苍穹外卖LOGO

工作台

订单管理

分类管理

菜品管理

套餐管理

数据统计

员工管理

营业中

营业状态

99+4

管理员

订单管理

全部订单

待接单(25)

待派送(50)

派送中(30)

已完成

已取消

1

订单号:手机号:下单时间:20220331 00:00-20220431 00:00

查询

重置

订单号	3	订单菜品	地址	预计送达时间	实收金额	备注	2	操作
2021010200001		宫保鸡丁* 3; 红烧带鱼* 2; 农家小炒肉* 1;	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	不要青菜		接单 拒单 查看
2021010200001		宫保鸡丁* 3; 红烧带鱼* 2; 农家小炒肉* 1;	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	不要青菜		接单 拒单 查看
2021010200001		宫保鸡丁* 3; 红烧带鱼* 2; 农家小炒肉* 1;	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00			接单 拒单 查看
2021010200001		宫保鸡丁* 3; 红烧带鱼* 2; 农家小炒肉* 1;	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	不要青菜		接单 拒单 查看
2021010200001		宫保鸡丁* 3; 红烧带鱼* 2; 农家小炒肉* 1;	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	不要青菜		接单 拒单 查看
2021010200001		宫保鸡丁* 3; 红烧带鱼* 2; 农家小炒肉* 1;	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00			接单 拒单 查看
2021010200001		宫保鸡丁* 3; 红烧带鱼* 2; 农家小炒肉* 1;	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00			接单 拒单 查看

<

1

2

3

4

5

6

7

8

9

>

10条/页

跳至

5

页

苍穹外卖LOGO

工作台

订单管理

分类管理

菜品管理

套餐管理

数据统计

员工管理

营业中

营业状态

99+4

管理员

订单管理

全部订单

待接单(25)

待派送(50)

派送中(30)

已完成

已取消

1

订单号:手机号:下单时间:20220331 00:00-20220431 00:00

查询

重置

订单号	3	订单菜品	地址	预计送达时间	备注	餐具数量	2	操作
2021010200001		宫保鸡丁* 3; 红烧带鱼* 2; 农家小炒肉* 1;	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	送到电梯口	1		派送 取消 查看
2021010200001		宫保鸡丁* 3; 红烧带鱼* 2; 农家小炒肉* 1;	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	不要青菜	1		派送 取消 查看
2021010200001		宫保鸡丁* 3; 红烧带鱼* 2; 农家小炒肉* 1;	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11		1		派送 取消 查看
2021010200001		宫保鸡丁* 3; 红烧带鱼* 2; 农家小炒肉* 1;	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	送到电梯口	5		派送 取消 查看
2021010200001		宫保鸡丁* 3; 红烧带鱼* 2; 农家小炒肉* 1;	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	不要青菜	3		派送 取消 查看
2021010200001		宫保鸡丁* 3; 红烧带鱼* 2; 农家小炒肉* 1;	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11		1		派送 取消 查看
2021010200001		宫保鸡丁* 3; 红烧带鱼* 2; 农家小炒肉* 1;	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11		1		派送 取消 查看

<

1

2

3

4

5

6

7

8

9

>

10条/页

跳至

5

页

苍穹外卖LOGO

工作台

订单管理

分类管理

菜品管理

套餐管理

数据统计

员工管理

营业中

营业状态

99+4

管理员

订单管理

全部订单

待接单(25)

待派送(50)

派送中(30)

已完成

已取消

1

订单号:手机号:下单时间:20220331 00:00-20220431 00:00

查询

重置

订单号	3	订单菜品	地址	预计送达时间	备注	餐具数量	2	操作
2021010200001		宫保鸡丁* 3; 红烧带鱼* 2; 农家小炒肉* 1;	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	送到电梯口	1	完成	取消 查看
2021010200001		宫保鸡丁* 3; 红烧带鱼* 2; 农家小炒肉* 1;	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	不要香菜	1	完成	取消 查看
2021010200001		宫保鸡丁* 3; 红烧带鱼* 2; 农家小炒肉* 1;	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11		1	完成	取消 查看
2021010200001		宫保鸡丁* 3; 红烧带鱼* 2; 农家小炒肉* 1;	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	送到电梯口	5	完成	取消 查看
2021010200001		宫保鸡丁* 3; 红烧带鱼* 2; 农家小炒肉* 1;	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	不要香菜	3	完成	取消 查看
2021010200001		宫保鸡丁* 3; 红烧带鱼* 2; 农家小炒肉* 1;	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11		1	完成	取消 查看
2021010200001		宫保鸡丁* 3; 红烧带鱼* 2; 农家小炒肉* 1;	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11		1	完成	取消 查看

<123456789>10条/页跳至5页

苍穹外卖LOGO

工作台

订单管理

分类管理

菜品管理

套餐管理

数据统计

员工管理

营业中

营业状态

99+4

管理员

订单管理

全部订单

待接单(25)

待派送(50)

派送中(30)

已完成

已取消

1

订单号:手机号:下单时间:20220331 00:00-20220431 00:00

查询

重置

订单号	3	用户名	手机号	地址	送达时间	实收金额	备注	2	操作
2021010200001		播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	不要香菜	退单	查看
2021010200001		播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	不要香菜	退单	查看
2021010200001		播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00		退单	查看
2021010200001		播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	不要香菜	退单	查看
2021010200001		播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	不要香菜	退单	查看
2021010200001		播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00		退单	查看

<123456789>10条/页跳至5页

苍穹外卖LOGO

工作台

订单管理

分类管理

菜品管理

套餐管理

数据统计

员工管理

营业中

营业状态

99+4

管理员

订单管理

全部订单

待接单(25)

待派送(50)

派送中(30)

已完成

已取消

1

订单号:手机号:下单时间:20220331 00:00-20220431 00:00

查询

重置

订单号	3	用户名	手机号	地址	下单时间	取消时间	取消原因	2	操作
2021010200001		播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	2021-01-02 11:11	用户取消	查看	
2021010200001		播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	2021-01-02 11:11	用户取消	查看	
2021010200001		播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	2021-01-02 11:11	菜品已销售完, 暂时无法接单	查看	
2021010200001		播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	2021-01-02 11:11	菜品已销售完, 暂时无法接单	查看	
2021010200001		播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	2021-01-02 11:11	用户取消	查看	
2021010200001		播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	2021-01-02 11:11	用户取消	查看	

<123456789>10条/页跳至5页

# 订单搜索

GET

/admin/order/conditionSearch

• 开发中 ▾

🌐 共享

订单管理接口

创建时间 2025年5月16日    修改时间 13 天前    修改者 狐友4CC

Mock >

## 请求参数

Query 参数		
beginTime	string	beginTime
endTime	string	endTime
number	string	number
page	string	page
pageSize	string	pageSize
phone	string	phone
status	string	status

## 返回响应

成功(200)		
HTTP 状态码: 200	内容格式: JSON	applicatio
数据结构		
code	number	
msg	null	
▼ data	object .....	
total	number	
▶ records	array [object {25}]	

业务规则：

- 输入订单号/手机号进行搜索，支持模糊搜索
- 根据订单状态进行筛选
- 下单时间进行时间筛选
- 搜索内容为空，提示未找到相关订单
- 搜索结果页，展示包含搜索关键词的内容
- 分页展示搜索到的订单数据

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

```
1  package com.sky.controller.admin;
2
3
4  import com.sky.dto.OrdersPageQueryDTO;
5  import com.sky.result.PageResult;
6  import com.sky.result.Result;
7  import com.sky.service.OrderService;
8  import io.swagger.annotations.Api;
9  import io.swagger.annotations.ApiOperation;
10 import lombok.extern.slf4j.Slf4j;
11 import org.springframework.beans.factory.annotation.Autowired;
12 import org.springframework.web.bind.annotation.GetMapping;
13 import org.springframework.web.bind.annotation.RequestMapping;
14 import org.springframework.web.bind.annotation.RestController;
15
16 /**
17  * 订单管理
18  */
19 @RestController("adminOrderController")
20 @RequestMapping("/admin/order")
21 @Slf4j
22 @Api(tags = "订单管理接口")
23 public class OrderController {
24
25     @Autowired
26     private OrderService orderService;
27
28     /**
29      * 订单搜索
30      *
31      * @param ordersPageQueryDTO
32      * @return
33      */
34     @GetMapping("/conditionSearch")
35     @ApiOperation("订单搜索")
36     public Result<PageResult> conditionSearch(OrdersPageQueryDTO
ordersPageQueryDTO) {
37         PageResult pageResult =
orderService.conditionSearch(ordersPageQueryDTO);
38         return Result.success(pageResult);
39     }
}
```

## OrderService.java

```

1      /**
2       * 条件搜索订单
3       * @param ordersPageQueryDTO
4       * @return
5       */
6      PageResult<OrderVO> conditionSearch(OrdersPageQueryDTO ordersPageQueryDTO);

```

## OrderServiceImpl.java

```

1      /**
2       * 订单搜索
3       *
4       * @param ordersPageQueryDTO
5       * @return
6       */
7      public PageResult<OrderVO> conditionSearch(OrdersPageQueryDTO ordersPageQueryDTO)
8      {
9          PageHelper.startPage(ordersPageQueryDTO.getPage(),
10             ordersPageQueryDTO.getPageSize());
11
12          Page<Orders> page = orderMapper.pageQuery(ordersPageQueryDTO);
13
14          // 部分订单状态，需要额外返回订单菜品信息，将Orders转化为OrderVO
15          List<OrderVO> orderVOList = getOrderVOList(page);
16
17          return new PageResult<OrderVO>(page.getTotal(), orderVOList);
18      }
19
20      private List<OrderVO> getOrderVOList(Page<Orders> page) {
21          // 需要返回订单菜品信息，自定义OrderVO响应结果
22          List<OrderVO> orderVOList = new ArrayList<>();
23
24          List<Orders> ordersList = page.getResult();
25          if (!CollectionUtils.isEmpty(ordersList)) {
26              for (Orders orders : ordersList) {
27                  // 将共同字段复制到OrderVO
28                  OrderVO orderVO = new OrderVO();
29                  BeanUtils.copyProperties(orders, orderVO);
30                  String orderDishes = getOrderDishesStr(orders);
31
32                  // 将订单菜品信息封装到orderVO中，并添加到orderVOList
33                  orderVO.setOrderDishes(orderDishes);
34                  orderVOList.add(orderVO);
35              }
36          }
37          return orderVOList;
38      }

```

```
37
38  /**
39   * 根据订单id获取菜品信息字符串
40   *
41   * @param orders
42   * @return
43   */
44  private String getOrderDishesStr(Orders orders) {
45      // 查询订单菜品详情信息（订单中的菜品和数量）
46      List<OrderDetail> orderDetailList =
orderDetailMapper.getByOrderId(orders.getId());
47
48      // 将每一条订单菜品信息拼接为字符串（格式：宫保鸡丁*3；）
49      List<String> orderDishList = orderDetailList.stream().map(x -> {
50          String orderDish = x.getName() + "*" + x.getNumber() + ";";
51          return orderDish;
52      }).collect(Collectors.toList());
53
54      // 将该订单对应的所有菜品信息拼接在一起
55      return String.join("", orderDishList);
56  }
```

全部订单

待接单<sup>1</sup>

待派送

派送中

已完成

订单号:

请填写订单号

手机号:

请填写手机号

下单时间:

📅

订单号

订单状态

用户名

手机号

地址

174852952350  
0

已取消

张三

13333333333

174848548748  
9

待接单

张三

13333333333

174840273165  
7

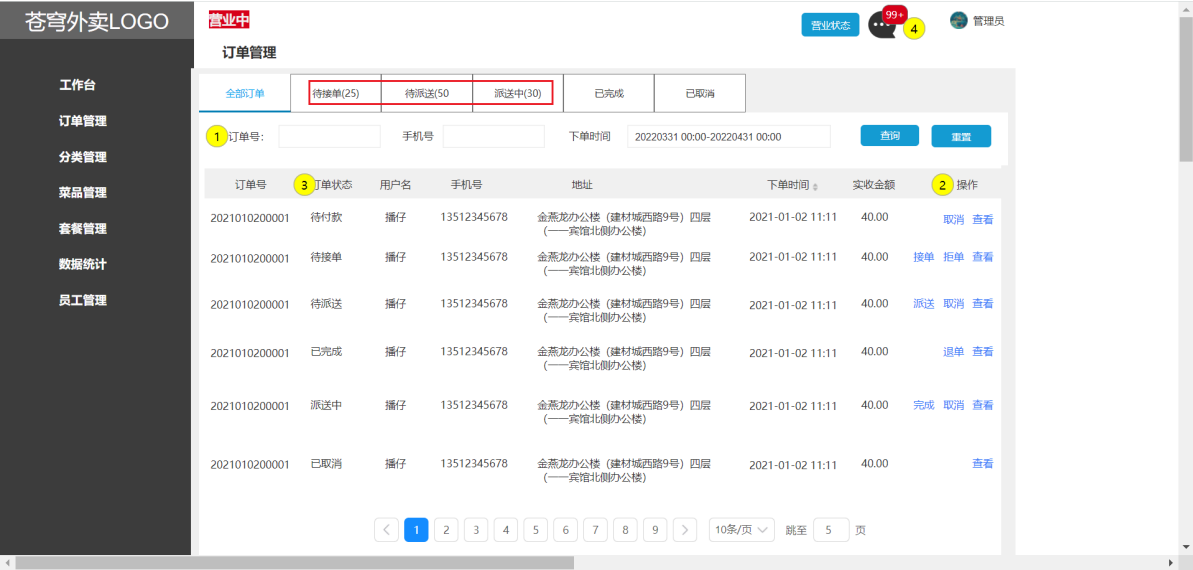
待付款

张三

13333333333



## 2. 各个状态的订单数量统计



### 各个状态的订单数量统计

**GET** /admin/order/statistics • 开发中

🌐 共享 订单管理接口

创建时间 2025年5月16日 修改时间 13 天前 修改者 狐友4C0G 创建者 狐友4C0G

Mock >

#### 请求参数

无

#### 返回响应

成功(200)

HTTP 状态码: 200 内容格式: JSON application/json

#### 数据结构

```
code integer <int32>
data object OrderStatisticsVO
  confirmed integer <int32> 待派送数量
  deliveryInProgress integer <int32> 派送中数量
  toBeConfirmed integer <int32> 待接单数量
msg string
```

com/sky/controller/admin/OrderController.java

```
1  /**
2   * 各个状态的订单数量统计
3   *
4   * @return
5   */
6  @GetMapping("/statistics")
7  @ApiOperation("各个状态的订单数量统计")
8  public Result<OrderStatisticsVO> statistics() {
9      OrderStatisticsVO orderStatisticsVO = orderService.statistics();
10     return Result.success(orderStatisticsVO);
11 }
```

OrderService.java

```
1  /**
2   * 各个状态的订单数量统计
3   * @return
4   */
5  OrderStatisticsVO statistics();
```

OrderServiceImpl.java

```
1  /**
2   * 各个状态的订单数量统计
3   *
4   * @return
5   */
6  public OrderStatisticsVO statistics() {
7      // 根据状态，分别查询出待接单、待派送、派送中的订单数量
8      Integer toBeConfirmed =
9      orderMapper.countStatus(Orders.TO_BE_CONFIRMED);
10     Integer confirmed = orderMapper.countStatus(Orders.CONFIRMED);
11     Integer deliveryInProgress =
12     orderMapper.countStatus(Orders.DELIVERY_IN_PROGRESS);
13
14     // 将查询出的数据封装到orderStatisticsVO中响应
15     OrderStatisticsVO orderStatisticsVO = new OrderStatisticsVO();
16     orderStatisticsVO.setToBeConfirmed(toBeConfirmed);
17     orderStatisticsVO.setConfirmed(confirmed);
18     orderStatisticsVO.setDeliveryInProgress(deliveryInProgress);
19     return orderStatisticsVO;
20 }
```

OrderMapper.java

```
1  /**
2   * 根据状态统计订单数量
3   * @param status
4   */
5  @Select("select count(id) from orders where status = #{status}")
6  Integer countStatus(Integer status);
```

全部订单

待接单<sup>1</sup>

待派送

订单号:

请填写订单号

手机号:

请填写

订单号	订单菜品	地址
1748485487489	馋嘴牛蛙*2;香 锅牛蛙*1;金汤 酸菜牛蛙*1;	

### 3. 查询订单详情

订单信息

X

订单号

20210505123012311

下单时间

2022年4月19日16:16:30

订单状态

待付款

订单金额

91.5

用户名

张先生

手机号

135 \*\*\*\* 1234

地址

金燕龙办公楼（建材城西路9号）四层（——宾馆北侧办公楼）

备注

不要葱花香菜、少盐少辣椒

菜品

鱼香肉丝	x1	¥ 30.50
宫保鸡丁	x1	¥ 30.50
北京烤鸭	x1	¥ 30.50

派送费

¥ 6

打包费

¥ 3

合计

¥ 91.50

取消订单

返回

- 业务规则：
- 订单详情页面需要展示订单基本信息（状态、订单号、下单时间、收货人、电话、收货地址、金额等）
  - 订单详情页面需要展示订单明细数据（商品名称、数量、单价）

# 查询订单详情

GET /admin/order/details/{id} 开发中

共享 订单管理接口

创建时间 2025年5月16日 修改时间 13 天前 修改者 狐友4C0G 创建者 狐友4C0G

Mock

## 请求参数

Path 参数

id

 string 订单id

示例值: 101

## 返回响应

成功(200)

HTTP 状态码: 200 内容格式: JSON application/json

数据结构

code

 integer <int32>

data

 object OrderVO

msg

 string

com/sky/controller/admin/OrderController.java

```
1  /**
2   * 订单详情
3   *
4   * @param id
5   * @return
6   */
7  @GetMapping("/details/{id}")
8  @ApiOperation("查询订单详情")
9  public Result<OrderVO> details(@PathVariable("id") Long id) {
10      OrderVO orderVO = orderService.details(id);
11      return Result.success(orderVO);
12  }
```

订单信息

订单号：1748571254898

待接单

下单时间：2025-05-30 10:14

用户名：张三

手机号：13333333333

预计送达时间：2025-05-30 11:14

地址：

备注

菜品

金汤酸菜牛蛙x3

¥88.00

菜品小计

¥264.00

费用

菜品小计：¥264

派送费：¥6

打包费：¥3

合计：¥273

支付渠道：微信支付

支付时间：2025-05-30 10:14

拒单

接单

4. 接单

苍穹外卖LOGO

营业中

营业状态

99+

4

管理员

工作台

订单管理

分类管理

菜品管理

套餐管理

数据统计

员工管理

订单管理

全部订单

待接单(25)

待派送(50)

派送中(30)

已完成

已取消

1

订单号：手机号：下单时间：

20220331 00:00-20220431 00:00

查询

重置

订单号	3	用户名	手机号	地址	下单时间	实收金额	2
2021010200001	待付款	播仔	13512345678	金燕龙办公楼（建材城西路9号）四层（——宾馆北侧办公楼）	2021-01-02 11:11	40.00	取消 查看
2021010200001	待接单	播仔	13512345678	金燕龙办公楼（建材城西路9号）四层（——宾馆北侧办公楼）	2021-01-02 11:11	40.00	接单 拒单 查看
2021010200001	待派送	播仔	13512345678	金燕龙办公楼（建材城西路9号）四层（——宾馆北侧办公楼）	2021-01-02 11:11	40.00	派送 取消 查看
2021010200001	已完成	播仔	13512345678	金燕龙办公楼（建材城西路9号）四层（——宾馆北侧办公楼）	2021-01-02 11:11	40.00	退单 查看
2021010200001	派送中	播仔	13512345678	金燕龙办公楼（建材城西路9号）四层（——宾馆北侧办公楼）	2021-01-02 11:11	40.00	完成 取消 查看
2021010200001	已取消	播仔	13512345678	金燕龙办公楼（建材城西路9号）四层（——宾馆北侧办公楼）	2021-01-02 11:11	40.00	查看

<

1

2

3

4

5

6

7

8

9

>

10条/页

跳至

5

页

订单信息

X

订单号

20210505123012311

下单时间

2022年4月19日16:16:30

订单状态

待接单

订单金额

91.5

用户名

张先生

手机号

135 \*\*\*\* 1234

预计送达时间

2022年4月19日16:16:30

地址

金燕龙办公楼（建材城西路9号）四层（——宾馆北侧办公楼）

备注

不要葱花香菜、少盐少辣椒

菜品

鱼香肉丝	x1	¥ 30.50
宫保鸡丁	x1	¥ 30.50
北京烤鸭	x1	¥ 30.50

派送费

¥ 6

打包费

¥ 3

合计

¥ 91.50

接单

拒单

☐ 处理完打开下一条信息

业务规则：

- 商家接单其实就是将订单的状态修改为“已接单”

# 接单

PUT /admin/order/confirm

• 开发中

共享

订单管理接口

创建时间

2025年5月16日

修改时间

13 天前

修改者

狐友4C00

Mock >

## 请求参数

Header 参数

Content-Type

string

示例值: application/json

Body 参数 application/json

id

integer <int64> 订单id

## 返回响应

成功(200)

HTTP 状态码: 200 内容格式: JSON application/json

数据结构

code

integer <int32>

data

object

msg

string

com/sky/controller/admin/OrderController.java



```

1      /**
2       * 接单
3       *
4       * @return
5       */
6      @PostMapping("/confirm")
7      @ApiOperation("接单")
8      public Result confirm(@RequestBody OrdersConfirmDTO ordersConfirmDTO) {
9          orderService.confirm(ordersConfirmDTO);
10         return Result.success();
11     }

```

## OrderService.java

```

1      /**
2       * 接单
3       *
4       * @param ordersConfirmDTO
5       */
6      void confirm(OrdersConfirmDTO ordersConfirmDTO);

```

## OrderServiceImpl.java

```

1      /**
2       * 接单
3       *
4       * @param ordersConfirmDTO
5       */
6      public void confirm(OrdersConfirmDTO ordersConfirmDTO) {
7          Orders orders = Orders.builder()
8              .id(ordersConfirmDTO.getId())
9              .status(Orders.CONFIRMED)
10             .build();
11
12         orderMapper.update(orders);
13     }

```

全部订单

待接单

待派送<sup>1</sup>

派

订单号:

请填写订单号

手机号:

请填写手机号

订单号	订单菜品	地址
1748485487489	馋嘴牛蛙*2;香锅牛蛙*1;金汤酸菜牛蛙*1;	

5. 拒单

苍穹外卖LOGO

营业中

营业状态 99 4 管理员

订单管理

全部订单

待接单(25)

待派送(50)

派送中(30)

已完成

已取消

1 订单号:

手机号

下单时间 20220331 00:00-20220431 00:00

查询

重置

订单号	3 订单状态	用户名	手机号	地址	下单时间	实收金额	2 操作
2021010200001	待付款	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	取消 查看
2021010200001	待接单	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	接单 拒单 查看
2021010200001	待派送	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	派送 取消 查看
2021010200001	已完成	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	退单 查看
2021010200001	派送中	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	完成 取消 查看
2021010200001	已取消	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	查看

< 1 2 3 4 5 6 7 8 9 >

10条/页 跳至 5 页

订单信息		X	
订单号	20210505123012311	下单时间	2022年4月19日16:16:30
订单状态	待接单	订单金额	91.5
用户名	张先生	手机号	135 **** 1234
预计送达时间	2022年4月19日16:16:30		
地址	金燕龙办公楼（建材城西路9号）四层（——宾馆北侧办公楼）		
备注	不要葱花香菜、少盐少辣椒		
菜品			
鱼香肉丝	x1	¥ 30.50	
宫保鸡丁	x1	¥ 30.50	
北京烤鸭	x1	¥ 30.50	
派送费		¥ 6	
打包费		¥ 3	
合计		¥ 91.50	
<button>接单</button>		<div><div>拒单</div></div>	
<input type="checkbox"/> 处理完打开下一条信息			

业务规则：

- 商家拒单其实就是将订单状态修改为“已取消”
- 只有订单处于“待接单”状态时可以执行拒单操作
- 商家拒单时需要指定拒单原因
- 商家拒单时，如果用户已经完成了支付，需要为用户退款

# 拒单

**PUT** /admin/order/rejection • 开发中

共享 | 订单管理接口

创建时间 2025年5月16日    修改时间 13 天前    修改者 狐友4C0G    创建者 狐友

Mock >

## 请求参数

Header 参数

Content-Type

 string

示例值: application/json

Body 参数 application/json

id

 integer <int64> 订单id

rejectionReason

 string 拒单原因

## 返回响应

成功(200)

HTTP 状态码: 200    内容格式: JSON    application/json

数据结构

code

 integer <int32>

data

 object

msg

 string

admin/OrderController

```

1      /**
2       * 拒单
3       *
4       * @return
5       */
6      @PostMapping("/rejection")
7      @ApiOperation("拒单")
8      public Result rejection(@RequestBody OrdersRejectionDTO
ordersRejectionDTO) throws Exception {
9          orderService.rejection(ordersRejectionDTO);
10         return Result.success();
11     }

```

## OrderService

```

1      /**
2       * 拒单
3       *
4       * @param ordersRejectionDTO
5       */
6      void rejection(OrdersRejectionDTO ordersRejectionDTO) throws Exception;

```

## OrderServiceImpl

```

1      /**
2       * 拒单
3       *
4       * @param ordersRejectionDTO
5       */
6      public void rejection(OrdersRejectionDTO ordersRejectionDTO) throws
Exception {
7          // 根据id查询订单
8          Orders ordersDB = orderMapper.getById(ordersRejectionDTO.getId());
9
10         // 订单只有存在且状态为2（待接单）才可以拒单
11         if (ordersDB == null ||
!ordersDB.getStatus().equals(Orders.TO_BE_CONFIRMED)) {
12             throw new
OrderBusinessException(MessageConstant.ORDER_STATUS_ERROR);
13         }
14
15         //支付状态
16         Integer payStatus = ordersDB.getPayStatus();
17         if (payStatus == Orders.PAID) {
18             //用户已支付，需要退款
19             //      String refund = weChatPayUtil.refund(
20             //          ordersDB.getNumber(),
21             //          ordersDB.getNumber(),
22             //          new BigDecimal(0.01),
23             //          new BigDecimal(0.01));
24             //      log.info("申请退款: {}", refund);

```

```

25     }
26
27     // 拒单需要退款，根据订单id更新订单状态、拒单原因、取消时间
28     Orders orders = new Orders();
29     orders.setId(ordersDB.getId());
30     orders.setStatus(Orders.CANCELLED);
31     orders.setRejectionReason(ordersRejectionDTO.getRejectionReason());
32     orders.setCancelTime(LocalDateTime.now());
33
34     orderMapper.update(orders);
35 }

```

## 6. 取消订单

苍穹外卖LOGO

工作台

订单管理

分类管理

菜品管理

套餐管理

数据统计

员工管理

营业中

营业状态

99+

4

管理员

订单管理

全部订单

待接单(25)

待派送(50)

派送中(30)

已完成

已取消

1 订单号:

手机号

下单时间

20220331 00:00-20220431 00:00

查询

重置

订单号	3 订单状态	用户名	手机号	地址	下单时间	实收金额	2 操作
2021010200001	待付款	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	取消 查看
2021010200001	待接单	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	接单 拒单 查看
2021010200001	待派送	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	派送 取消 查看
2021010200001	已完成	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	退单 查看
2021010200001	派送中	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	完成 取消 查看
2021010200001	已取消	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	查看

< 1 2 3 4 5 6 7 8 9 >

10条/页

跳至 5 页

业务规则：

- 取消订单其实就是将订单状态修改为“已取消”
- 商家取消订单时需要指定取消原因
- 商家取消订单时，如果用户已经完成了支付，需要为用户退款

# 取消订单

**PUT** /admin/order/cancel • 开发中 ▾

🌐 共享 订单管理接口

创建时间 2025年5月16日    修改时间 13 天前    修改者 狐友4C0G    创建

Mock >

## 请求参数

Header 参数

Content-Type

 string  
示例值: application/json

Body 参数 application/json

cancelReason

 string 订单取消原因

id

 integer <int64> 订单id

## 返回响应

成功(200)

HTTP 状态码: 200    内容格式: JSON    application/json

数据结构

code

 integer <int32>

data

 object

msg

 string

admin/OrderController

```

1      /**
2       * 取消订单
3       *
4       * @return
5       */
6      @PostMapping("/cancel")
7      @ApiOperation("取消订单")
8      public Result cancel(@RequestBody OrdersCancelDTO ordersCancelDTO)
9      throws Exception {
10         orderService.cancel(ordersCancelDTO);
11         return Result.success();
12     }

```

## OrderService

```

1      /**
2       * 商家取消订单
3       *
4       * @param ordersCancelDTO
5       */
6      void cancel(OrdersCancelDTO ordersCancelDTO) throws Exception;

```

## OrderServiceImpl

```

1      /**
2       * 取消订单
3       *
4       * @param ordersCancelDTO
5       */
6      public void cancel(OrdersCancelDTO ordersCancelDTO) throws Exception {
7          // 根据id查询订单
8          Orders ordersDB = orderMapper.getById(ordersCancelDTO.getId());
9
10         //支付状态
11         Integer payStatus = ordersDB.getPayStatus();
12         if (payStatus == 1) {
13             //用户已支付，需要退款
14             String refund = weChatPayUtil.refund(
15                 // String refund = weChatPayUtil.refund(
16                 //         ordersDB.getNumber(),
17                 //         ordersDB.getNumber(),
18                 //         new BigDecimal(0.01),
19                 //         new BigDecimal(0.01));
20                 // log.info("申请退款: {}", refund);
21             }
22
23             // 管理端取消订单需要退款，根据订单id更新订单状态、取消原因、取消时间
24             Orders orders = new Orders();
25             orders.setId(ordersCancelDTO.getId());
26             orders.setStatus(Orders.CANCELLED);
27             orders.setCancelReason(ordersCancelDTO.getCancelReason());

```



```
28 orders.setCancelTime(LocalDateTime.now());
29 orderMapper.update(orders);
30 }
```

订单信息

订单号: 1748571254898

已取消

下单时间: 2025-05-30 10:14

用户名: 张三

手机号: 13333333333

地址:

取消原因

订单量较多, 暂时无法接单

菜品

金汤酸菜牛蛙x3

¥88.00

菜品小计

¥264.00

费用

菜品小计: ¥264

派送费: ¥6

打包费: ¥3

合计: ¥273

支付渠道: 微信支付

支付时间: 2025-05-30 10:14

7. 派送订单

苍穹外卖LOGO

营业中

营业状态

99+

4

管理员

订单管理

全部订单

待接单(25)

待派送(50)

派送中(30)

已完成

已取消

1 订单号:

手机号

下单时间

20220331 00:00-20220431 00:00

查询

重置

订单号	3 订单状态	用户名	手机号	地址	下单时间	实收金额	2 操作
2021010200001	待付款	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	取消 查看
2021010200001	待接单	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	接单 拒单 查看
2021010200001	待派送	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	派送 取消 查看
2021010200001	已完成	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	退单 查看
2021010200001	派送中	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	完成 取消 查看
2021010200001	已取消	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	查看

<

1

2

3

4

5

6

7

8

9

>

10条/页

跳至

5

页

业务规则:

- 派送订单其实就是将订单状态修改为“派送中”
- 只有状态为“待派送”的订单可以执行派送订单操作

派送订单

PUT

/admin/order/delivery/{id}

• 开发中

共享

订单管理接口

创建时间

2025年5月16日

修改时间

13 天前

修改者

狐友4C0G

创建者

Mock

请求参数

Path 参数

id

string

订单id

示例值:

101

Body 参数

text/plain

返回响应

成功(200)

HTTP 状态码: 200

内容格式: JSON

application/json

数据结构

code

integer <int32>

data

object

msg

string

admin/OrderController

```

1      /**
2       * 派送订单
3       *
4       * @return
5       */
6      @PostMapping("/delivery/{id}")
7      @ApiOperation("派送订单")
8      public Result delivery(@PathVariable("id") Long id) {
9          orderService.delivery(id);
10         return Result.success();
11     }

```

## OrderService

```

1      /**
2       * 派送订单
3       *
4       * @param id
5       */
6      void delivery(Long id);

```

## OrderServiceImpl

```

1      /**
2       * 派送订单
3       *
4       * @param id
5       */
6      public void delivery(Long id) {
7          // 根据id查询订单
8          Orders ordersDB = orderMapper.getById(id);
9
10         // 校验订单是否存在，并且状态为3
11         if (ordersDB == null ||
12             !ordersDB.getStatus().equals(Orders.CONFIRMED)) {
13             throw new
14                 OrderBusinessException(MessageConstant.ORDER_STATUS_ERROR);
15         }
16
17         Orders orders = new Orders();
18         orders.setId(ordersDB.getId());
19         // 更新订单状态,状态转为派送中
20         orders.setStatus(Orders.DELIVERY_IN_PROGRESS);
21
22         orderMapper.update(orders);
23     }

```

全部订单

待接单

待派送

派送中1

已完成

已取消

订单号:

请填写订单号

手机号:

请填写手机号

下单时间:

开始日期

至

结束日期

查询

订单号	订单菜品	地址	预计送达时间	备注	餐具数量	操作
1748485487489	馋嘴牛蛙*2;香锅牛蛙*1;金汤酸菜牛蛙*1;		2025-05-29 11:24		0	<a href="#">完成</a> <a href="#">取消</a> <a href="#">查看</a>

8. 完成订单

苍穹外卖LOGO

营业中

营业状态

99+

4

管理员

订单管理

全部订单

待接单(25)

待派送(50)

派送中(30)

已完成

已取消

1

订单号:

手机号

下单时间

20220331 00:00-20220431 00:00

查询

重置

订单号	3 订单状态	用户名	手机号	地址	下单时间	实收金额	2 操作
2021010200001	待付款	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	<a href="#">取消</a> <a href="#">查看</a>
2021010200001	待接单	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	<a href="#">接单</a> <a href="#">拒单</a> <a href="#">查看</a>
2021010200001	待派送	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	<a href="#">派送</a> <a href="#">取消</a> <a href="#">查看</a>
2021010200001	已完成	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	<a href="#">退单</a> <a href="#">查看</a>
2021010200001	派送中	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	<a href="#">完成</a> <a href="#">取消</a> <a href="#">查看</a>
2021010200001	已取消	播仔	13512345678	金燕龙办公楼 (建材城西路9号) 四层 (——宾馆北侧办公楼)	2021-01-02 11:11	40.00	<a href="#">查看</a>

<

1

2

3

4

5

6

7

8

9

>

10条/页

跳至

5

页

业务规则：

- 完成订单其实就是将订单状态修改为“已完成”
- 只有状态为“派送中”的订单可以执行订单完成操作

# 完成订单

PUT /admin/order/complete/{id}

• 开发中

共享

订单管理接口

创建时间

2025年5月16日

修改时间

13 天前

修改者

狐友4C0G

1

Mock >

## 请求参数

Path 参数	
id	string 订单id
示例值:	101
Header 参数	
Content-Type	string
示例值:	application/json
Body 参数	
text/plain	

## 返回响应

成功(200)		
HTTP 状态码:	200	内容格式: JSON application/json
数据结构		
code	integer <int32>	
data	object	
msg	string	

admin/OrderController

```

1      /**
2       * 完成订单
3       *
4       * @return
5       */
6      @PostMapping("/complete/{id}")
7      @ApiOperation("完成订单")
8      public Result complete(@PathVariable("id") Long id) {
9          orderService.complete(id);
10         return Result.success();
11     }

```

## OrderService

```

1      /**
2       * 完成订单
3       *
4       * @param id
5       */
6      void complete(Long id);

```

## OrderServiceImpl

```

1      /**
2       * 完成订单
3       *
4       * @param id
5       */
6      public void complete(Long id) {
7          // 根据id查询订单
8          Orders ordersDB = orderMapper.getById(id);
9
10         // 校验订单是否存在，并且状态为4
11         if (ordersDB == null ||
12             !ordersDB.getStatus().equals(Orders.DELIVERY_IN_PROGRESS)) {
13             throw new
14                 OrderBusinessException(MessageConstant.ORDER_STATUS_ERROR);
15         }
16
17         Orders orders = new Orders();
18         orders.setId(ordersDB.getId());
19         // 更新订单状态,状态转为完成
20         orders.setStatus(Orders.COMPLETED);
21         orders.setDeliveryTime(LocalDateTime.now());
22
23         orderMapper.update(orders);
24     }

```

全部订单

待接单

待派送

派送中

已完成

已取消

订单号:

请填写订单号

手机号:

请填写手机号

下单时间:

开始日期

至

结束日期

订单号	用户名	手机号	地址	送达时间	实收金额
1748485487489	张三	13333333333		2025-05-30 10:12	¥ 362

### 3. 校验收货地址是否超出配送范围

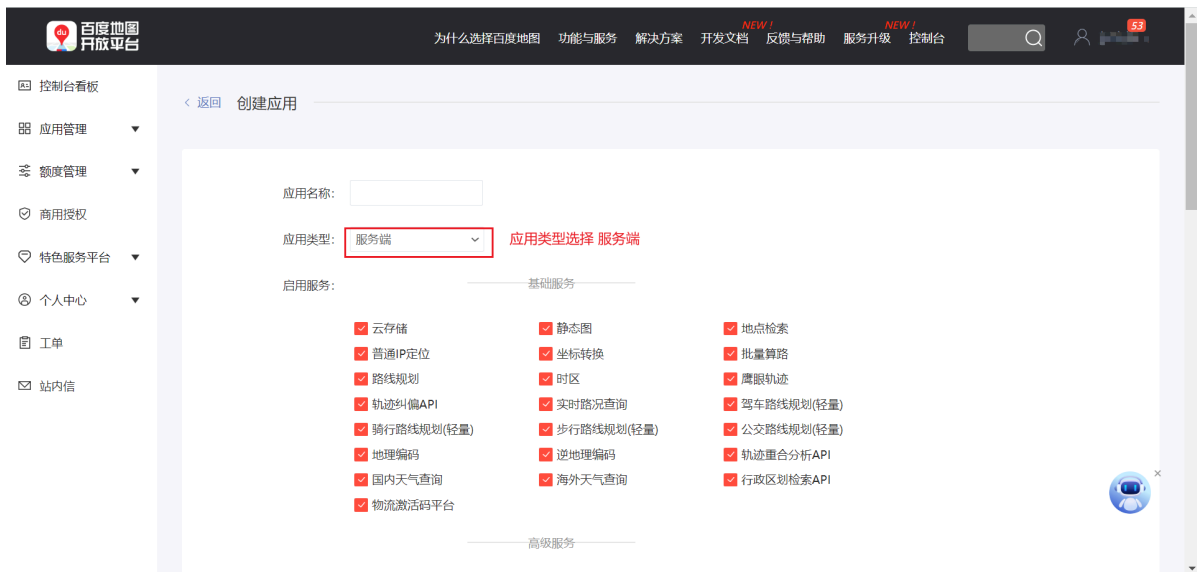
#### 1. 环境准备

注册账号：<https://passport.baidu.com/v2/?reg&tt=1671699340600&overseas=&gid=CF954C2-A3D2-417F-9FE6-B0F249ED7E33&tpl=pp&u=https%3A%2F%2Flbsyun.baidu.com%2Findex.php%3Ftitle%3D%E9%A6%96%E9%A1%B5>

登录百度地图开放平台：<https://lbsyun.baidu.com/>

进入控制台，创建应用，获取AK：





相关接口:

<https://lbsyun.baidu.com/index.php?title=webapi/guide/webservice-geocoding>

<https://lbsyun.baidu.com/index.php?title=webapi/directionlite-v1>

## 2. 代码开发

application.yml

```
1 sky:
2   shop:
3     address: ${sky.shop.address}
4   baidu:
5     ak: ${sky.baidu.ak}
```

application-dev.yml

```
1 sky:
2   shop:
3     address: 北京市西城区丰汇园21号
4   baidu:
5     ak: XBxxxxxxxxxxx
```

com/sky/service/impl/OrderServiceImpl.java

```
1     @Value("${sky.shop.address}") //这里要导入spring的包而不是lombok的包
2     private String shopAddress;
3
4     @Value("${sky.baidu.ak}")
5     private String ak;
6
7     /**
8      * 检查客户的收货地址是否超出配送范围
```



```

9      * @param address
10     */
11     private void checkOutOfRange(String address) {
12         Map map = new HashMap();
13         map.put("address", shopAddress);
14         map.put("output", "json");
15         map.put("ak", ak);
16
17         //获取店铺的经纬度坐标
18         String shopCoordinate =
HttpClientUtil.doGet("https://api.map.baidu.com/geocoding/v3", map);
19
20         JSONObject jsonObject = JSON.parseObject(shopCoordinate);
21         if(!jsonObject.getString("status").equals("0")){
22             throw new OrderBusinessException("店铺地址解析失败");
23         }
24
25         //数据解析
26         JSONObject location =
jsonObject.getJSONObject("result").getJSONObject("location");
27         String lat = location.getString("lat");
28         String lng = location.getString("lng");
29         //店铺经纬度坐标
30         String shopLngLat = lat + "," + lng;
31
32         map.put("address", address);
33         //获取用户收货地址的经纬度坐标
34         String userCoordinate =
HttpClientUtil.doGet("https://api.map.baidu.com/geocoding/v3", map);
35
36         jsonObject = JSON.parseObject(userCoordinate);
37         if(!jsonObject.getString("status").equals("0")){
38             throw new OrderBusinessException("收货地址解析失败");
39         }
40
41         //数据解析
42         location =
jsonObject.getJSONObject("result").getJSONObject("location");
43         lat = location.getString("lat");
44         lng = location.getString("lng");
45         //用户收货地址经纬度坐标
46         String userLngLat = lat + "," + lng;
47
48         map.put("origin", shopLngLat);
49         map.put("destination", userLngLat);
50         map.put("steps_info", "0");
51
52         //路线规划
53         String json =
HttpClientUtil.doGet("https://api.map.baidu.com/directionlite/v1/driving",
map);
54
55         jsonObject = JSON.parseObject(json);
56         if(!jsonObject.getString("status").equals("0")){
57             throw new OrderBusinessException("配送路线规划失败");
58         }

```

```

59
60         //数据解析
61         JSONObject result = jsonObject.getJSONObject("result");
62         JSONArray jsonArray = (JSONArray) result.get("routes");
63         Integer distance = (Integer) ((JSONObject)
        jsonArray.get(0)).get("distance");
64
65         if(distance > 5000){
66             //配送距离超过5000米
67             throw new OrderBusinessException("超出配送范围");
68         }
69     }

```

在OrderServiceImpl的submitOrder方法中调用上面的校验方法：

```

public class OrderServiceImpl implements OrderService {
    public OrderSubmitVO submitOrder(OrdersSubmitDTO orderSubmitDTO) {
        if(addressBook==null){
            //抛出业务异常
            throw new AddressBookBusinessException(MessageConstant.ADDRESS_BOOK_IS_NULL);
        }

        //检查用户的收货地址是否超出配送范围
        checkOutOfRange( address: addressBook.getCityName()+addressBook.getDistrictName()+addressBook.getDetail());

        //查询当前用户的购物车数据
        Long userId= BaseContext.getCurrentId();
        ShoppingCart shoppingCart=new ShoppingCart();
        shoppingCart.setUserId(userId);
    }
}

```

```

//路线规划
String json = HttpClientUtil.doGet( url: "https://api.map.baidu.com/directionlite/v1/driving", map);

JSONObject = JSON.parseObject(json);  json: {"status":0,"message":"ok","result":{"origin":{"lng":11
if(!jsonObject.getString( key: "status").equals("0")){  jsonObject:  size = 3
    throw new OrderBusinessException("配送路线规划失败");
}

//数据解析
JSONObject result = jsonObject.getJSONObject( key: "result");
JSONArray jsonArray = (JSONArray) result.get("routes");
Integer distance = (Integer) ((JSONObject) jsonArray.get(0)).get("distance");

if(distance > 5000){
    //配送距离超过5000米
    throw new OrderBusinessException("超出配送范围");
}

```

ables    Console    Actuator

ession (Enter) or add a watch (Ctrl+Shift+Enter)

- ✓ duration -> {Integer@14610} 0
- > "traffic\_condition" -> {Integer@12599} 0
- ✓ "distance" -> {Integer@14621} 20
  - > key = "distance"
  - > value = {Integer@14621} 20

测试成功后可以注释掉checkOutOfRange的调用，避免影响测试。

# 额外参考

---

<https://zhuanlan.zhihu.com/p/692714923>