

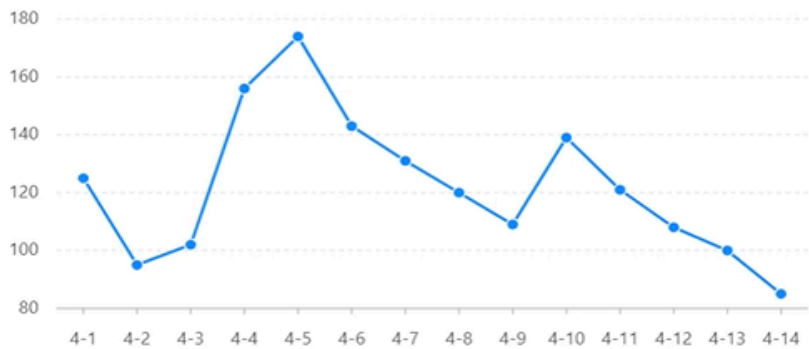
苍穹外卖day11

1. 营业额统计

产品原型：

昨日 近7日 近30日 本周 本月

营业额统计



业务规则：

- 营业额指订单状态为已完成的订单金额合计
- 基于可视化报表的折线图展示营业额数据，x轴为日期，y轴为营业额
- 根据时间选择区间，展示每天的营业额数据

接口设计：

基本信息

Path: /admin/report/turnoverStatistics

Method: GET

接口描述：

请求参数

Query

参数名称	是否必须	示例	备注
begin	是	2022-05-01	开始日期
end	是	2022-05-31	结束日期

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	必须			
└─ dateList	string	必须		日期列表，日期之间以逗号分隔	
└─ turnoverList	string	必须		营业额列表，营业额之间以逗号分隔	
msg	string	非必须			

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

```
1 package com.sky.controller.admin;  
2  
3 import com.sky.result.Result;  
4 import com.sky.service.ReportService;  
5 import com.sky.vo.TurnoverReportVO;
```

```

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.format.annotation.DateTimeFormat;
11 import org.springframework.web.bind.annotation.GetMapping;
12 import org.springframework.web.bind.annotation.RequestMapping;
13 import org.springframework.web.bind.annotation.RestController;
14
15 import java.time.LocalDate;
16
17
18 /**
19  * 数据统计相关接口
20  */
21 @RestController
22 @RequestMapping("/admin/report")
23 @Api(tags="数据统计相关接口")
24 @Slf4j
25 public class ReportController {
26
27
28     @Autowired
29     private ReportService reportService;
30
31     /**
32      * 营业额统计
33      * @param begin
34      * @param end
35      * @return
36      */
37     @GetMapping("/turnoverStatistics")
38     @ApiOperation("营业额统计")
39     public Result<TurnoverReportVO> turnoverStatistics(
40         @DateTimeFormat(pattern = "yyyy-MM-dd") LocalDate begin,
41         @DateTimeFormat(pattern = "yyyy-MM-dd") LocalDate end) {
42         log.info("营业额数据统计:{}", begin, end);
43         return Result.success(reportService.getTurnoverStatistics(begin,
44 end));
45     }
46 }

```

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

```

1  package com.sky.service;
2
3  import com.sky.vo.TurnoverReportVO;
4
5  import java.time.LocalDate;
6
7  public interface ReportService {
8

```

```

9      /**
10     * 统计指定时间区间内的营业额数据
11     * @param begin
12     * @param end
13     * @return
14     */
15     TurnoverReportVO getTurnoverStatistics(LocalDate begin, LocalDate end);
16 }
17

```

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

```

1  package com.sky.service.impl;
2
3  import com.sky.entity.Orders;
4  import com.sky.mapper.OrderMapper;
5  import com.sky.service.ReportService;
6  import com.sky.vo.TurnoverReportVO;
7  import lombok.extern.slf4j.Slf4j;
8  import org.apache.commons.lang.StringUtils;
9  import org.springframework.beans.factory.annotation.Autowired;
10 import org.springframework.stereotype.Service;
11
12 import java.time.LocalDate;
13 import java.time.LocalDateTime;
14 import java.time.LocalTime;
15 import java.util.ArrayList;
16 import java.util.HashMap;
17 import java.util.List;
18 import java.util.Map;
19
20 @Service
21 @Slf4j
22 public class ReportServiceImpl implements ReportService {
23
24     @Autowired
25     private OrderMapper orderMapper;
26
27
28     /**
29     * 统计指定时间区间内的营业额数据
30     * @param begin
31     * @param end
32     * @return
33     */
34     @Override
35     public TurnoverReportVO getTurnoverStatistics(LocalDate begin, LocalDate
end) {
36         //当前集合用于存放从begin到end范围内的每天日期
37         List<LocalDate> dateList= new ArrayList<>();
38
39         dateList.add(begin);
40         while(!begin.equals(end)) {

```

```

41         //日期计算，计算指定日期的后一天对应的日期
42         begin = begin.plusDays(1);
43         dateList.add(begin);
44     }
45
46     //存放每天的营业额
47     List<Double> turnoverList = new ArrayList<>();
48     for (LocalDate date : dateList) {
49         //查询date日期对应的营业额数据，营业额是指：状态为“已完成”的订单金额合计
50         LocalDateTime beginTime = LocalDateTime.of(date, LocalTime.MIN);
51         LocalDateTime endTime = LocalDateTime.of(date, LocalTime.MAX);
52
53         //select sum(amount) from orders where order_time > beginTime
54         //and order_time < endTime and status = 5
55         Map map=new HashMap();
56         map.put("begin",beginTime);
57         map.put("end",endTime);
58         map.put("status", Orders.COMPLETED);
59         Double turnover=orderMapper.sumByMap(map);
60         turnover = turnover == null ? 0.0 : turnover;//防止null的情况
61         turnoverList.add(turnover);
62     }
63
64     //封装返回结果
65     return TurnoverReportVO.builder()
66         .dateList(StringUtils.join(dateList,""))
67         .turnoverList(StringUtils.join(turnoverList,""))
68         .build();
69 }
70 }

```

OrderMapper.java

```

1  /**
2   * 根据动态条件统计营业额数据
3   * @param map
4   * @return
5   */
6  Double sumByMap(Map map);

```

OrderMapper.xml

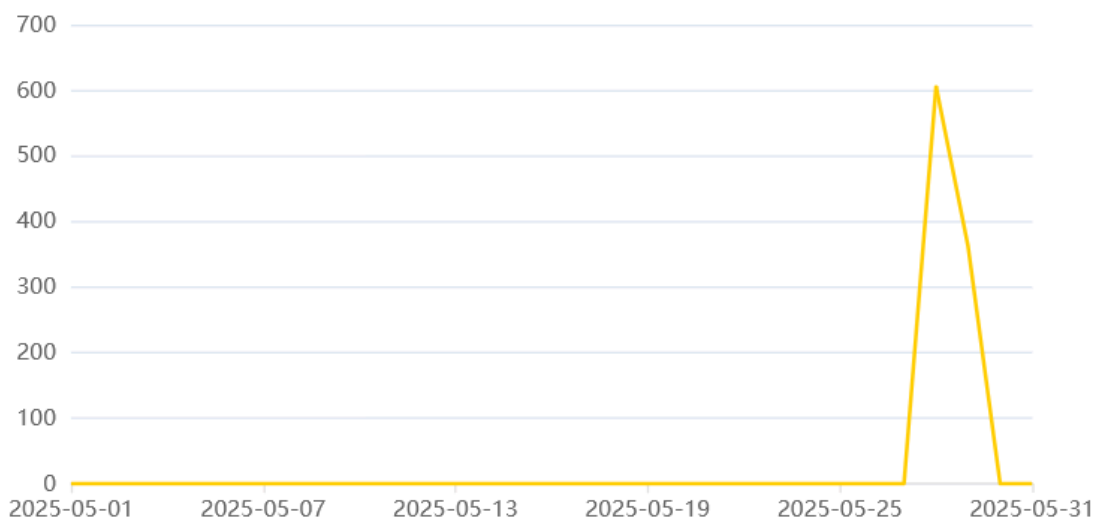
```

1  <select id="sumByMap" resultType="java.lang.Double">
2      select sum(amount) from orders
3      <where>
4          <if test="status != null">
5              and status = #{status}
6          </if>
7          <if test="begin != null">

```

```
8         and order_time >= #{begin} <!--使用转义字符-->
9     </if>
10    <if test="end != null">
11        and order_time <= #{end}
12    </if>
13 </where>
14 </select>
```

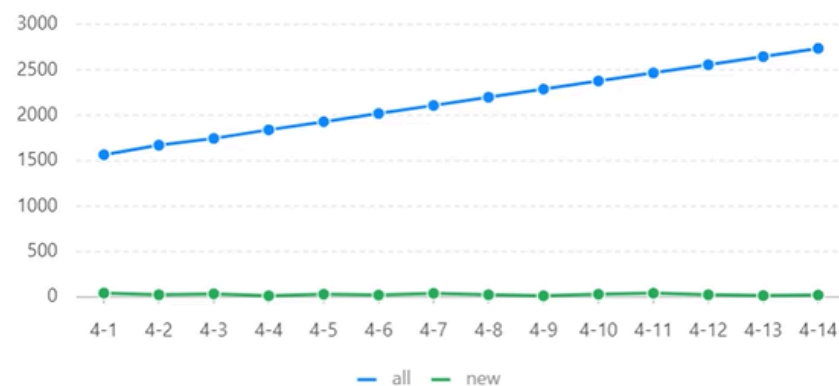
营业额统计



2. 用户统计

产品原型：

用户统计



业务规则：

- 基于可视化报表的折线图展示用户数据，x轴为日期，y轴为用户数
- 根据时间选择区间，展示每天的用户总量和新增用户量数据

接口设计:

基本信息

Path: /admin/report/userStatistics

Method: GET

接口描述:

请求参数

Query

参数名称	是否必须	示例	备注
begin	是	2022-05-01	开始日期
end	是	2022-05-31	结束日期

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	必须			
├ dateList	string	必须		日期列表, 以逗号分隔	
├ newUserList	string	必须		新增用户数列表, 以逗号分隔	
├ totalUserList	string	必须		总用户量列表, 以逗号分隔	
msg	string	非必须			

ReportController.java

```
1  /**
2   * 用户数据统计
3   *
4   * @param begin
5   * @param end
6   * @return
7   */
8  @GetMapping("/userStatistics")
9  @ApiOperation("用户数据统计")
10 public Result<UserReportVO> userStatistics(
11     @DateTimeFormat(pattern = "yyyy-MM-dd") LocalDate begin,
12     @DateTimeFormat(pattern = "yyyy-MM-dd") LocalDate end) {
13
14     return Result.success(reportService.getUserStatistics(begin, end));
15 }
```

ReportService.java

```
1  /**
2   * 统计指定时间区间内的用户数据
3   * @param begin
4   * @param end
5   * @return
6   */
7  UserReportVO getUserStatistics(LocalDate begin, LocalDate end);
```

ReportServiceImpl.java

```
1  @Autowired
2  private UserMapper userMapper;
3
4  /**
5   * 根据时间区间统计用户数量
6   *
```

```

7      * @param begin
8      * @param end
9      * @return
10     */
11    @Override
12    public UserReportVO getUserStatistics(LocalDate begin, LocalDate end) {
13        List<LocalDate> dateList = new ArrayList<>();
14        dateList.add(begin);
15
16        while (!begin.equals(end)) {
17            begin = begin.plusDays(1);
18            dateList.add(begin);
19        }
20        List<Integer> newUserList = new ArrayList<>(); // 新增用户数
21        List<Integer> totalUserList = new ArrayList<>(); // 总用户数
22
23        for (LocalDate date : dateList) {
24            LocalDateTime beginTime = LocalDateTime.of(date, LocalTime.MIN);
25            LocalDateTime endTime = LocalDateTime.of(date, LocalTime.MAX);
26            Map map=new HashMap();
27            map.put("end",endTime);
28            // 总用户数量 select count(id) from user where create_time < ?
29            Integer totalUser = userMapper.countByMap(map);
30
31            map.put("begin",beginTime);
32            // 新增用户数量 select count(id) from user where create_time > ?
33            // and create_time < ?
34            Integer newUser = userMapper.countByMap(map);
35
36            newUserList.add(newUser);
37            totalUserList.add(totalUser);
38        }
39
40        return UserReportVO.builder()
41            .dateList(StringUtils.join(dateList, ","))
42            .newUserList(StringUtils.join(newUserList, ","))
43            .totalUserList(StringUtils.join(totalUserList, ","))
44            .build();
45    }

```

UserMapper.java

```

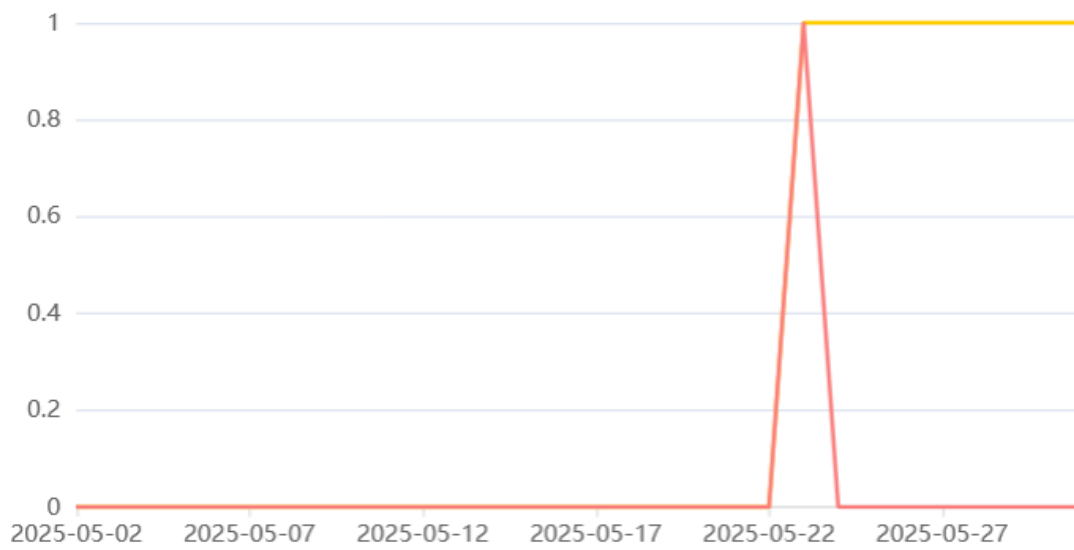
1    /**
2     * 根据动态条件统计用户数量
3     *
4     * @param map
5     * @return
6     */
7    Integer countByMap(Map<String, Object> map);

```

UserMapper.xml

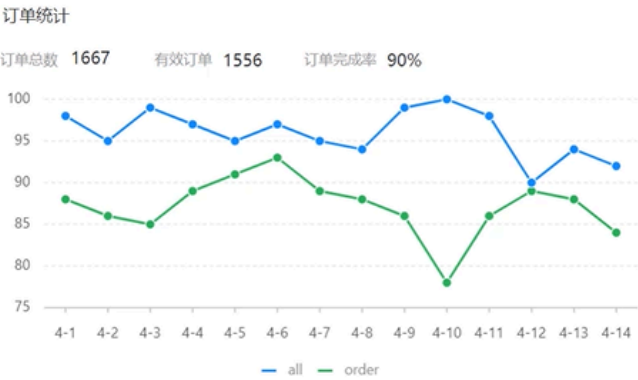
```
1 <select id="countByMap" resultType="java.lang.Integer">
2     select count(id) from user
3     <where>
4         <if test="begin != null">
5             and create_time >= #{begin}
6         </if>
7         <if test="end != null">
8             and create_time <= #{end}
9         </if>
10    </where>
11 </select>
```

用户统计



3. 订单统计

产品原型：



业务规则：

- 有效订单指状态为“已完成”的订单
- 基于可视化报表的折线图展示订单数据，x轴为日期，y轴为订单数量
- 根据时间选择区间，展示每天的订单总数和有效订单数
- 展示所选时间区间内的有效订单数、总订单数、订单完成率， $\text{订单完成率} = \text{有效订单数} / \text{总订单数} * 100\%$

接口设计：

基本信息				返回数据		
Path: /admin/report/ordersStatistics				名称	类型	其他信息
Method: GET				code	integer	format: int32
接口描述:				data	object	
请求参数				dateList	string	日期列表，以逗号分隔
				orderCompletionRate	number	订单完成率
				orderCountList	string	订单数列表，以逗号分隔
				totalOrderCount	integer	订单总数
				validOrderCount	integer	有效订单数
				validOrderCountList	string	有效订单数列表，以逗号分隔
				msg	string	
Query						
参数名称	是否必须	示例	备注			
begin	是	2022-05-01	开始日期			
end	是	2022-05-31	结束日期			

ReportController.java

```
1  /**
2   * 订单数据统计
3   *
4   * @param begin
5   * @param end
6   * @return
7   */
8  @GetMapping("/ordersStatistics")
9  @ApiOperation("订单数据统计")
10 public Result<OrderReportVO> orderStatistics(
11     @DateTimeFormat(pattern = "yyyy-MM-dd") LocalDate begin,
12     @DateTimeFormat(pattern = "yyyy-MM-dd") LocalDate end) {
13
14     return Result.success(reportService.getOrderStatistics(begin, end));
15 }
```

ReportService.java

```
1  /**
2   * 根据时间区间统计订单数量
3   * @param begin
4   * @param end
5   * @return
6   */
7  OrderReportVO getOrderStatistics(LocalDate begin, LocalDate end);
```

ReportServiceImpl.java

```
1  /**
2   * 根据时间区间统计订单数量
3   * @param begin
4   * @param end
5   * @return
6   */
7  @Override
8  public OrderReportVO getOrderStatistics(LocalDate begin, LocalDate end)
9  {
10     List<LocalDate> dateList = new ArrayList<>();
11     dateList.add(begin);
12
13     while (!begin.equals(end)) {
14         begin = begin.plusDays(1);
15         dateList.add(begin);
16     }
17
18     //存放每天的订单总数
19     List<Integer> orderCountList = new ArrayList<>();
20     //存放每天的有效订单数
21     List<Integer> validOrderCountList = new ArrayList<>();
22
23     for(LocalDate date : dateList) {
24         LocalDateTime beginTime = LocalDateTime.of(date, LocalTime.MIN);
25         LocalDateTime endTime = LocalDateTime.of(date, LocalTime.MAX);
26         // 查询每天的总订单数 select count(id) from orders where order_time
27         > ? and order_time < ?
28         Integer orderCount=getOrderCount(beginTime,endTime,null);
29         // 查询每天的有效订单数 select count(id) from orders where
30         order_time > ? and order_time < ? and status = ?
31         Integer
32         validOrderCount=getOrderCount(beginTime,endTime,Orders.COMPLETED);
33
34         orderCountList.add(orderCount);
35         validOrderCountList.add(validOrderCount);
36     }
37
38     //计算时间区间内的订单总数量
39     Integer
40     totalOrderCount=orderCountList.stream().reduce(Integer::sum).get();
41     //reduce()方法是用于对流元素进行归约操作的核心方法
```

```

37
38         //计算时间区间内的有效订单数量
39         Integer
validOrderCount=validOrderCountList.stream().reduce(Integer::sum).get();
40
41         //计算订单完成率
42         Double orderCompletionRate = 0.0;
43         if (totalOrderCount!=0){
44             orderCompletionRate=validOrderCount.doubleValue() /
totalOrderCount;
45         }
46
47         return OrderReportVO.builder()
48             .dateList(StringUtils.join(dateList, ","))
49             .orderCountList(StringUtils.join(orderCountList, ","))
50             .validOrderCountList(StringUtils.join(validOrderCountList,
","))
51             .totalOrderCount(totalOrderCount)
52             .validOrderCount(validOrderCount)
53             .orderCompletionRate(orderCompletionRate)
54             .build();
55
56     }
57
58     /**
59      * 根据条件统计订单数量
60      * @param begin
61      * @param end
62      * @param status
63      * @return
64      */
65     private Integer getOrderCount(LocalDateTime begin, LocalDateTime
end,Integer status) {
66         Map map=new HashMap();
67         map.put("begin",begin);
68         map.put("end",end);
69         map.put("status",status);
70         return orderMapper.countByMap(map);
71     }

```

OrderMapper.java

```

1         /**
2         * 根据动态条件统计订单数量
3         * @param map
4         * @return
5         */
6         Integer countByMap(Map map);

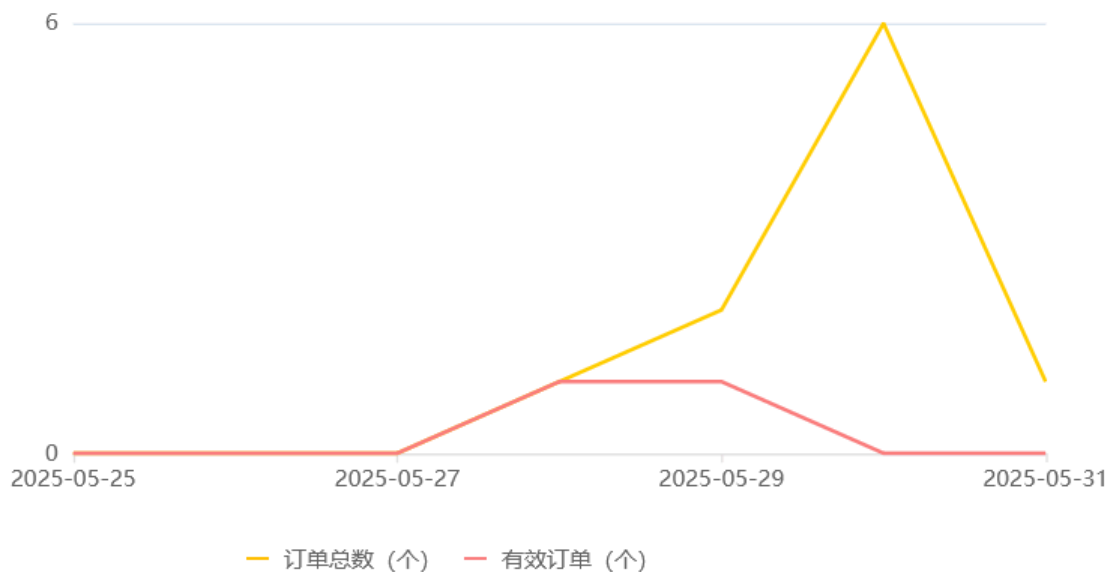
```

OrderMapper.xml

```
1 <select id="countByMap" resultType="java.lang.Integer">
2     select count(id) from orders
3     <where>
4         <if test="status != null">
5             and status = #{status}
6         </if>
7         <if test="begin != null">
8             and order_time &gt;= #{begin} <!--使用转义字符-->
9         </if>
10        <if test="end != null">
11            and order_time &lt;= #{end}
12        </if>
13    </where>
14 </select>
```

订单统计

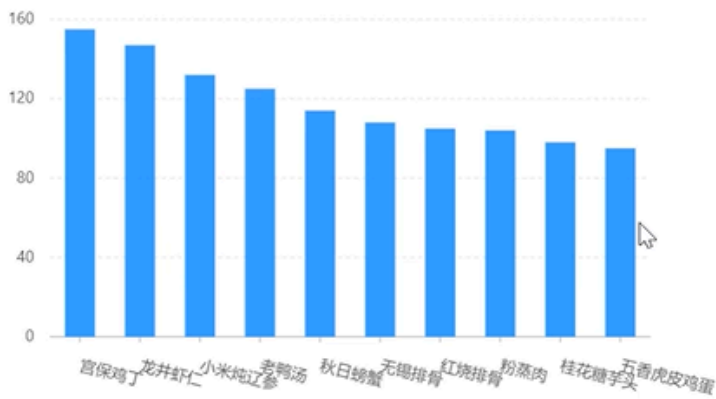
订单完成率 = 有效订单 / 订单总数
20.0% = **2** / **10**



4. 销量排名统计

产品原型：

销量排名TOP10



业务规则：

- 根据时间选择区间，展示销量前10的商品（包括菜品和套餐）
- 基于可视化报表的柱状图降序展示商品销量
- 此处的销量为商品销售的份数

接口设计：

基本信息

Path: /admin/report/top10

Method: GET

接口描述：

请求参数

Query

参数名称	是否必须	示例	备注
begin	是	2022-05-01	开始日期
end	是	2022-05-31	结束日期

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	必须			
nameList	string	必须		商品名称列表，以逗号分隔	
numberList	string	必须		销量列表，以逗号分隔	
msg	string	非必须			

ReportController.java

```
1  /**
2   * 销量排名统计
3   * @param begin
4   * @param end
5   * @return
6   */
7  @GetMapping("/top10")
8  @ApiOperation("销量排名top10")
9  public Result<SalesTop10ReportVO> top10(
10      @DateTimeFormat(pattern = "yyyy-MM-dd") LocalDate begin,
11      @DateTimeFormat(pattern = "yyyy-MM-dd") LocalDate end){
12      log.info("销量排名: :{},{}", begin, end);
13      return Result.success(reportService.getSalesTop10(begin, end));
```

```
14     }
```

ReportService.java

```
1  /**
2   * 销量排名统计
3   * @param begin
4   * @param end
5   * @return
6   */
7  SalesTop10ReportVO getSalesTop10(LocalDate begin, LocalDate end);
```

ReportServiceImpl.java

```
1  /**
2   * 销量排名统计
3   * @param begin
4   * @param end
5   * @return
6   */
7  @Override
8  public SalesTop10ReportVO getSalesTop10(LocalDate begin, LocalDate end)
9  {
10     //select od.name ,sum(od.number) from order_detail od,orders o where
    od.order_id=o.id and o.status = 5
11     //and o.order_time > ? and o.order_time < ? group by od.name order
    by number desc
12     //limit 0,10
13
14     LocalDateTime beginTime = LocalDateTime.of(begin, LocalTime.MIN);
15     LocalDateTime endTime = LocalDateTime.of(end, LocalTime.MAX);
16
17     List<GoodssSalesDTO> salesTop10 = orderMapper.getSalesTop(beginTime,
18     endTime);
19
20     List<String> names =
21     salesTop10.stream().map(GoodssSalesDTO::getName).collect(Collectors.toList())
22     ;
23     String nameList = StringUtils.join(names, ",");
24
25     List<Integer> numbers =
26     salesTop10.stream().map(GoodssSalesDTO::getNumber).collect(Collectors.toList(
27     ));
28     String numberList = StringUtils.join(numbers, ",");
29     //封装返回结果数据
30     return SalesTop10ReportVO.builder()
31         .nameList(nameList)
32         .numberList(numberList)
33         .build();
34 }
```

OrderMapper.java

```
1  /**
2   * 统计指定时间区间内的销量排名前10
3   *
4   * @param begin
5   * @param end
6   */
7  List<GoodsSalesDTO> getSalesTop(LocalDateTime begin, LocalDateTime end);
```

OrderMapper.xml

```
1  <select id="getSalesTop" resultType="com.sky.dto.GoodsSalesDTO">
2      select od.name name,sum(od.number) number from order_detail od
,orders o
3      where od.order_id = o.id
4      and o.status = 5
5      <if test="begin != null">
6          and order_time >= #{begin}
7      </if>
8      <if test="end != null">
9          and order_time <= #{end}
10     </if>
11     group by name
12     order by number desc
13     limit 0, 10
14 </select>
```

销量排名TOP10

