

2. ?????

2.1 ???????

```
wimoor-ui/src/views/amazon/listing/analysis/  
├─ index.vue #   
└─ components/  
    ├─ data_deatils.vue #   
    └─ dialog.vue #   
  
wimoor-ui/src/api/amazon/product/  
└─ productAnysApi.js #   

```

2.2 ???????

2.2.1 ???????index.vue?

□□□□ □ wimoor-ui/src/views/amazon/listing/analysis/index.vue

□□□□ □

- □□□□□□□□□□□□□□
- □□ SKU□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□

□□□□□ □

```
<template>  
  <div class="gird-line-head el-white-bg">  
    <!-- □□□□□□ -->  
    <div class="flex-center">  
      <el-space>  
        <Group @change="groupChange" ref="groupRef" :init="true"/>  
        <el-input v-model="queryParams.search" clearable @input="handleQuery" placeholder="□□□"  
" class="input-with-select">  
          <!-- □□□□□□ -->  
          <template #prepend>  
            <el-select v-model="queryParams.ftype" @change='handleQuery' style="width:100px;"  
placeholder="SKU">  
              <!-- □□□□□□ -->  

```

```

        </el-select>
    </template>
    <!-- 搜索 -->
    <template #append>
        <el-button @click="handleQuery">
            <el-icon class="font-base ic-cen"><search /></el-icon>
        </el-button>
    </template>
</el-input>
</el-space>
</div>
</div>

<!-- 商品列表 -->
<div class="grid-content">
    <!-- 商品SKU -->
    <div class="left-content el-white-bg ">
        <div class="con-header"><h4>SKU</h4></div>
        <el-scrollbar style="height:calc(100vh - 280px)">
            <div>
                <ul class="sku-list" v-infinite-scroll="load">
                    <li class="pointer" v-for="item in tableData" @click="selectSku(item)"
: class="{ 'active': item.active}">
                        {{item.sku}}
                        <div class="font-extraSmall">ASIN[{{item.asin}}]</div>
                        <div v-if="showmarket" class="font-extraSmall">{{item.groupname}}-
{{item.marketname}}</div>
                    </li>
                </ul>
            </div>
        </el-scrollbar>
    <!-- 分页 -->
    <pagination v-if="total > 0" :total="total" layout="total,prev,next" v-
model:page="queryParams.currentPage" v-model:limit="queryParams.pageSize"
@pagination="handleQuery" />
</div>

<!-- 右侧内容 -->
<div class="right-content">
    <el-scrollbar class="screen-height gary-bg">

```

```

        <DataDeatils ref="dataDeatilsRef"/>
    </el-scrollbar>
</div>
</div>
</template>

<script setup>
import {ref, reactive, toRefs, onMounted} from "vue";
import DataDeatils from "../components/data_deatils.vue"
import Group from '@/components/header/group.vue';
import productAnysApi from '@/api/amazon/product/productAnysApi.js';

// 初始化
let state=reactive({
    tableData:[],
    total:10,
    showmarket:false,
    queryParams:{
        pagesize:10,
        currentpage:1,
        ftype:'sku',
    }
})

// 取消选中
function selectSku(row){
    state.tableData.forEach((item)=>{
        item.active = false;
    })
    row.active = true;
    dataDeatilsRef.value.show(row);
}

// 切换分组
function groupChange(obj){
    state.queryParams.groupid=obj.groupid;
    state.queryParams.marketplaceid=obj.marketplaceid;
    if(state.queryParams.groupid&&state.queryParams.marketplaceid){
        state.showmarket=false;
    }else{

```

```

    state.showmarket=true;
  }
  handleQuery();
}

// 初始化
function handleQuery(){
  productAnysApi.productAsinList(state.queryParams).then((res)=>{
    state.tableData=res.data.records;
    state.total=res.data.total;
    if(state.total>0){
      selectSku(res.data.records[0]);
    }
  });
}
</script>

```

2.2.2 data_deatils.vue

文件路径: wimoor-ui/src/views/amazon/listing/analysis/components/data_deatils.vue

文件内容:

- 初始化
- 获取商品信息
- 获取商品评价
- 获取商品问答

模板内容:

```

<template>
  <div class="gird-line-right">
    <!-- 初始化 -->
    <el-card>
      <el-row gutter="16">
        <el-col :span="16">
          <div class="p-b-h">
            <!-- 获取商品信息 -->
            <div>
              <el-image v-if="infoMap.image" :src="infoMap.image" class="img-size"></el-image>
              <el-image v-else :src="$require('empty/noimage40.png')" class="img-size"></el-

```



```

    <template #label>
      <div @click.stop="showQueryDialog" class="custom-tabs-label pointer font-black"
style="padding-left: 10px; padding-right: 10px;margin-left:-10px;margin-right:-10px" v-
if="item.name=='none'">
        <el-icon><Plus /></el-icon>
      </div>
      <div class="custom-tabs-label" v-else>{{item.name}}</div>
    </template>
  </el-tab-pane>
</el-tabs>

```

```

<!-- 日期选择器 -->

```

```

<el-card class="p-a-card">
  <template #header>
    <div class="flex-center-between">
      <el-space>
        <el-radio-group v-model="times" @change="changeTimes">
          <el-radio-button label="7" />
          <el-radio-button label="30" />
          <el-radio-button label="90" />
        </el-radio-group>
        <Datepicker ref="datepickersRef" :days="1" @changedate="changedate" />
      </el-space>
    </div>
  </template>
  <div class="p-a-body">
    <div class="p-a-right">
      <div id="anaysis-mycharts" class="my-chart"></div>
    </div>
  </div>
</el-card>

```

```

<!-- 表格 -->

```

```

<el-card style="margin-top:10px;">
  <el-scrollbar style="width:calc(100vw - 350px);" always>
    <table class="sd-table">
      <tr>
        <td width="80px;"><td width="80px;"></td>
        <td width="80px;"></td>
      </tr>
    </table>
  </el-scrollbar>
</el-card>

```

```

        <td v-for="label in labels" width="60px;">{{label}}</td>
    </tr>
    <tr v-for="(legend,index) in legends">
        <td width="80px;">{{legend}}</td>
        <td width="80px;">{{summary[index]}}</td>
        <td width="60px;" v-if="series && series[index] && series[index].data" v-for="item
in series[index].data">
            {{item}}
        </td>
    </tr>
</table>
</el-scrollbar>
</el-card>

<!-- 对话框 -->
<el-dialog v-model="remarkVisable" title="">
    <el-input v-model="infoMap.remark2" type="textarea" :rows="3"></el-input>
    <template #footer>
        <el-button @click="remarkVisable=false"></el-button>
        <el-button type="primary" @click.stop="updateAnyRemark"></el-button>
    </template>
</el-dialog>

<!-- 对话框 -->
<Dialog ref="dialogRef" @change="loadQueryList"></Dialog>
</div>
</template>

<script setup>
import {ref,reactive,toRefs}from"vue"
import * as echarts from 'echarts';
import productAnysApi from '@api/amazon/product/productAnysApi.js';
import queryFieldApi from '@api/sys/tool/queryFieldApi.js';

// 
let state=reactive({
    remarkVisable:false,
    activeName:"sales",
    times:"07",

```

```

infoMap:{remark2:''},
isload:true,
queryParams:{},
labels:[],
series:[],
summary:[],
legends:[],
queryList:[{name:"none"}],
})

// 初始化
function loadChart(){
  var ftype="";
  if(state.activeName=="hisrank"||state.activeName=="sales"){
    ftype=state.activeName;
  }else{
    state.queryList.forEach(item=>{
      if(state.activeName==item.id){
        ftype=item.queryfield;
      }
    })
  }
  if(ftype!="none"){
    setTimeout(function(){

productAnysApi.getChartData({"sku":state.infoMap.sku,"marketplaceid":state.infoMap.marketplace
id,"groupid":state.infoMap.groupid,

"ftype":ftype,"fromDate":state.queryParams.fromDate,"endDate":state.queryParams.endDate}).then
((res)=>{
  if (res.data && res.data.length > 0) {
    var data=res.data;
    state.labels = data[0].labels;
    state.series = [];
    state.legends = [];
    var hasrightline=false;
    for (var i = 0; i < data.length; i++) {
      var name = data[i].name;
      if (name == 'xxxxx' || name == 'xxxxx' || name == 'xxxxx'

```

```

        || name == 'Acos' || name == "AcoAs" || name == "□□□□□"
        || name == "□□□□□□") {
    hasrightline=true;break;
}
}

for (var i = 0; i < data.length; i++) {
    state.legends.push(data[i].name);
    var datas = {};
    state.summary[i]=0;
    data[i].data.forEach(item=>{
        if(item){
            state.summary[i]=state.summary[i]+parseFloat(item);
        }
    })
    datas.name = data[i].name;
    datas.type = "line";
    if (datas.name == '□□□□□' || datas.name == '□□□□□' || datas.name == '□□□□□'
        || datas.name == 'Acos' || datas.name == "AcoAs" || datas.name == "□□□□□□"
        || datas.name == "□□□□□□") {
        state.summary[i]=formatFloat(state.summary[i]/data[i].data.length)+" (avg)";
        datas.yAxisIndex = 1;
    } else if(hasrightline==true){
        datas.type = "bar";
        datas.barGap = "0%";
        datas.boundaryGap = "0%";
        datas.barMaxWidth = 32, datas.itemStyle = {
            normal : {
                barBorderRadius : [ 4, 4, 0, 0 ]
            }
        };
    }
    if(hasrightline==false){
        datas.symbolSize = 0, datas.itemStyle = {
            normal : {
                lineStyle : {
                    width : 2
                }
            }
        }
    }
}

```

```

        };
    }
    datas.smooth = 0.5;
    datas.symbol = 'emptycircle';
    datas.data = data[i].data;
    datas.label={
        show:true,
    };
    datas.showAllSymbol=false;
    state.series.push(datas);
}
lineChart();
}
if(state.isload==true){
    state.isload=false;
}
});
},500);
}else{
    showQueryDialog();
}
}

// 初始化
function lineChart() {
    if(myChart!=null){
        myChart.clear()
    }else{
        myChart =echarts.init(document.getElementById('anaysis-mycharts'));
    }
    var option = {
        tooltip : {
            trigger : 'axis',
            formatter : function(params) {
                var showHtm = "";
                for (var i = 0; i < params.length; i++) {
                    var date = params[i].name;
                    var name = params[i].seriesName;
                    var value = params[i].value;

```

```

    if (name == '□□□□' || name == '□□□□□' || name == '□□□□□□'
        || name == 'Acos' || name == "AcoAs" || name == "□□□□□□"
            || name == "□□□□□□□") {
        showHtm += name + ": " + value + "%" + '<br>';
    } else {
        showHtm += name + ": " + value + '<br>';
    }
}
showHtm = date + '<br>' + showHtm;
return showHtm;
},
axisPointer : {
    type : 'line',
   LineStyle : {
        color : '#ccc',
        width : 1,
        type : 'solid'
    },
},
},
legend : {
    data : state.legends,
    y : 'top',
    x : 'center',
},
color : [ '#ffa400', '#75D6AA', '#EB6A79', '#7AA5DA', '#d69bf2',
    '#59f3e3', '#8875ff', '#e0e0e5', '#ff8559', '#00FF7F', '#00FF7F' ],
grid : {
    x : 50,
    x2 : 50,
    y : 50,
    y2 : 50,
    borderWidth : 0,
},
calculable : false,
xAxis : [ {
    axisLabel : {
        show : true,
        textStyle : {

```

```
        color : '#999'
      }
    },
    splitLine : {
     LineStyle : {
        color : '#f1f1f1',
        width : 1,
      }
    },
    axisTick : {
      show : false,
     LineStyle : {
        color : '#f1f1f1'
      }
    },
    axisLine : {
     LineStyle : {
        color : '#f1f1f1',
        width : 1,
      }
    },
    type : 'category',
    boundaryGap : true,
    data : state.labels
  } ],
  yAxis : [ {
    axisLabel : {
      show : true,
      textStyle : {
        color : '#999'
      },
    },
    splitLine : {
     LineStyle : {
        color : '#f1f1f1',
        width : 1,
      }
    },
    axisLine : {
```

```

        lineStyle : {
            color : '#f1f1f1',
            width : 1,
        }
    },
}, {
    axisLabel : {
        show : true,
        textStyle : {
            color : '#999'
        },
    },
},
splitLine : {
    show: false,
},
axisLine : {
    lineStyle : {
        color : '#f1f1f1',
        width : 1,
    }
},
}, 1,
series : state.series
};
myChart.setOption(option);
window.addEventListener('resize', ()=>{
    myChart.resize();
});
}

// 商品详情
function show(row){
    if(row.id){
        productAnysApi.productdetail({"pid":row.id}).then((res)=>{
            state.infoMap=res.data;
            state.infoMap.link="https://"+row.point_name+"/dp/"+res.data.asin;
            loadChart();
        });
    }
}

```

```

loadQueryList();
}

// 初始化
defineExpose({show})
</script>

```

2.3 API????

文件 `wimoor-ui/src/api/amazon/product/productAnysApi.js`

API列表

名称	URL	方法	说明
productAsinList	/amazon/api/v1/report/product/analysis/productAsinList	POST	获取 ASIN 列表
productdetail	/amazon/api/v1/report/product/analysis/productdetail	GET	获取产品详情
productdetailByInfo	/amazon/api/v1/report/product/analysis/productdetailByInfo	GET	根据 SKU 获取产品详情
updateAnyRemark	/amazon/api/v1/report/product/analysis/updateAnyRemark	GET	更新产品备注
getChartData	/amazon/api/v1/report/product/analysis/getChartData	GET	获取产品图表数据

3. ?????

3.1 ????????

```

wimoor-amazon/amazon-boot/src/main/java/com/wimoor/amazon/product/
├─ controller/
│   └─ ProductAnalysisController.java # 产品分析控制器
├─ service/
│   └─ IProductInOrderService.java # 产品订单服务接口
│       └─ impl/
│           └─ ProductInOrderServiceImpl.java # 产品订单服务实现类
└─ mapper/

```

```

├─ ProductInOrderMapper.java      # □□□□□□□□
├─ ProductRankMapper.java        # □□□□□□□□
└─ ProductInOptMapper.java       # □□□□□□□□

```

3.2 ???????

3.2.1 ??????ProductAnalysisController.java?

```

□□□ □ wimoor-amazon/amazon-
boot/src/main/java/com/wimoor/amazon/product/controller/ProductAnalysisController.java

```

□□□ □

- □□□□□□ HTTP□□
- □□□□□□□□□□□□□□
- □□□□□□□□

□□□ □

```

@Api(tags = "□□□□")
@RestController
@SystemControllerLog("□□□□")
@RequestMapping("/api/v1/report/product/analysis")
@RequiredArgsConstructor
public class ProductAnalysisController {
    @Autowired
    IProductInfoService iProductInfoService;
    final IAmazonAuthorityService amazonAuthorityService;
    final IProductInOrderService iProductInOrderService;
    final IProductInOptService iProductInOptService;

    @PostMapping("/productAsinList")
    public Result<IPage<Map<String, Object>>> productListAction(@RequestBody ProductListDTO
query) {
        String search = query.getSearch();
        String searchtype = query.getFtype();
        Map<String, Object> parameter = new HashMap<String, Object>();
        parameter.put("searchtype", searchtype);
        parameter.put("search", search != null && !search.isEmpty() ? "%" + search.trim() +
"% " : null);
        String marketplaceid = query.getMarketplaceid();

```

```

String groupid = query.getGroupid();
parameter.put("marketplaceid", marketplaceid != null && !marketplaceid.isEmpty() &&
!"all".equals(marketplaceid) ? marketplaceid.trim() : null);
UserInfo userinfo = UserInfoContext.get();
if (userinfo.isLimit(UserLimitDataType.operations)) {
    parameter.put("myself", userinfo.getId());
}

parameter.put("shopid", userinfo.getCompanyid());
if(!"all".equals(groupid)&&StrUtil.isNotEmpty(groupid)) {
    if(StrUtil.isNotEmpty(marketplaceid)) {
        AmazonAuthority auth = amazonAuthorityService.selectByGroupAndMarket(groupid,
marketplaceid);
        if(auth!=null) {
            parameter.put("amazonAuthId", auth.getId());
        }
    }
}

if (StrUtil.isBlankOrUndefined(groupid)||"all".equals(groupid)) {
    parameter.put("groupid", null);
    if(userinfo.getGroups()!=null&&userinfo.getGroups().size(>0) {
        parameter.put("groupList", userinfo.getGroups());
    }
} else {
    parameter.put("groupid", groupid);
}

IPage<Map<String, Object>> list = iProductInfoService.getAsinList(query.getPage(),
parameter);
return Result.success(list);
}

@GetMapping("/productdetail")
public Result<Map<String, Object>> productListAction(String pid) {
    UserInfo userinfo = UserInfoContext.get();
    return Result.success(iProductInOrderService.selectDetialById(pid,
userinfo.getCompanyid()));
}

@GetMapping("/productdetailByInfo")

```

```

    public Result<Map<String, Object>> productdetailByInfoAction(String sku, String
marketplaceid, String sellerid, String groupid) {
        UserInfo userinfo = UserInfoContext.get();
        AmazonAuthority auth = amazonAuthorityService.selectByGroupAndMarket(groupid,
marketplaceid);
        if(auth!=null) {
            LambdaQueryWrapper<ProductInfo> queryWrapper = new
LambdaQueryWrapper<ProductInfo>();
            queryWrapper.eq(ProductInfo::getAmazonAuthId, auth.getId());
            queryWrapper.eq(ProductInfo::getMarketplaceid, marketplaceid);
            queryWrapper.eq(ProductInfo::getSku, sku);
            ProductInfo info = iProductInfoService.getOne(queryWrapper);
            if(info!=null) {
                return Result.success(iProductInOrderService.selectDetialById(info.getId(),
userinfo.getCompanyid()));
            } else {
                return Result.failed();
            }
        } else {
            return Result.failed();
        }
    }

    @SystemControllerLog("□□□□□□")
    @GetMapping("/updateAnyRemark")
    public Result<?> updateAnyRemarkAction(String pid, String remark) {
        ProductInOpt opt = iProductInOptService.getById(pid);
        if(opt!=null) {
            opt.setRemarkAnalysis(remark);
            return Result.success(iProductInOptService.updateById(opt));
        } else {
            opt = new ProductInOpt();
            opt.setPid(new BigInteger(pid));
            opt.setRemarkAnalysis(remark);
            return Result.success(iProductInOptService.save(opt));
        }
    }

    @GetMapping("/getChartData")
    public Result<List<Map<String, Object>>> getChartDataAction(String sku, String

```

```

marketplaceid, String groupid, String ftype, String fromDate, String endDate) {
    List<Map<String, Object>> maps = null;
    UserInfo user = UserInfoContext.get();
    Map<String, Object> parameter = new HashMap<String, Object>();
    parameter.put("shopid", user.getCompanyid());
    parameter.put("marketplace", marketplaceid != null && !marketplaceid.isEmpty() ?
marketplaceid.trim() : null);
    parameter.put("groupid", groupid != null && !groupid.isEmpty() ? groupid.trim() :
null);

    if (StrUtil.isEmpty(groupid) || StrUtil.isEmpty(marketplaceid)) {
        throw new BizException("□□□□□□□□");
    }
    AmazonAuthority amazonAuthority =
amazonAuthorityService.selectByGroupAndMarket(groupid, marketplaceid);
    if (amazonAuthority != null) {
        parameter.put("amazonAuthId", amazonAuthority.getId());
    }
    parameter.put("userid", user.getId());
    String beginDate = fromDate;
    Map<String, Integer> ftypeset = new HashMap<String, Integer>();
    if ("sales".equals(ftype)) {
        ftypeset.put("uns", 0);
        ftypeset.put("ods", 1);
    } else if ("hisrank".equals(ftype)) {
        ftypeset.put("rnks", 0);
    } else {
        String[] ftypeStr = ftype.split(",");
        for (int i = 0; i < ftypeStr.length; i++) {
            if(StrUtil.isNotEmpty(ftypeStr[i])) {
                ftypeset.put(ftypeStr[i], i);
            }
        }
    }
    if (StrUtil.isNotEmpty(ftype)) {
        maps = iProductInOrderService.getChartData(ftypeset, parameter, user);
    } else {
        maps = null;
    }
    return Result.success(maps);
}

```



```

parameter.get("beginDate");
    String endDate = parameter.get("endDate") == null ? null : (String)
parameter.get("endDate");
    String amazonAuthId = parameter.get("amazonAuthId") == null ? null :
parameter.get("amazonAuthId").toString();

    // []
    List<AmzProductPageviews> sessionlist = null;
    List<ProductRank> rnclist = null;
    List<Map<String, Object>> advlist = null;
    List<OrdersSummary> orderlist = null;
    List<Map<String, Object>> ftypeList = null;
    List<Map<String, Object>> rankList = null;

    if (typesMap.containsKey("rnk")) { // sales rank
        rnclist = this.productRank(sku, marketplace, beginDate, endDate, amazonAuthId,
user, ftype);
    }
    if (typesMap.containsKey("rnks")) { // sales rank []
        ftypeList = this.getCountRankBySku(sku, marketplace, amazonAuthId, user);
        if (ftypeList != null && ftypeList.size() > 0) {
            rankList = this.getRankBySku(sku, marketplace, beginDate, endDate,
amazonAuthId, user);
            for (int i = 0; i < ftypeList.size(); i++) {
                typesMap.put(ftypeList.get(i).get("name").toString(), i + 1);
            }
        } else {
            rankList = new ArrayList<Map<String, Object>>();
            typesMap.put("[]", 1);
        }
    }
    if (typesMap.containsKey("uns") || typesMap.containsKey("ods")
        || typesMap.containsKey("pts") || typesMap.containsKey("aups")) {
        // total order item [] // units orders [], [], []
        orderlist = iOrdersSumService.orderSummaryBySkuDate(sku, marketplace, beginDate,
endDate, amazonAuthId, user, ftype);
    }
    if (typesMap.containsKey("cks") // clicks []
        || typesMap.containsKey("imp") // impressions []

```

```

        || typesMap.containsKey("ctr") // ctr
        || typesMap.containsKey("spd") // adv spend
        || typesMap.containsKey("cpc") // CPC
        || typesMap.containsKey("cr") // total order/click
        || typesMap.containsKey("acos") // total order/click
        || typesMap.containsKey("tos") // total sales
        || typesMap.containsKey("acoas")
        || typesMap.containsKey("aus") //
        || typesMap.containsKey("aups")) { //
advlist = this.advInfo(sku, marketplace, beginDate, endDate, amazonAuthId, user,
ftype);
    if (typesMap.containsKey("acoas") && orderlist == null) {
        orderlist = iOrdersSumService.orderSummaryBySkuDate(sku, marketplace,
beginDate, endDate, amazonAuthId, user, ftype);
    }
}
if (typesMap.containsKey("ses") // session
    || typesMap.containsKey("pgv") // pageview,
    || typesMap.containsKey("bbp") || typesMap.containsKey("osp")) { //
Unit_Session_Percentage
    sessionlist = this.sessionPage(sku, marketplace, beginDate, endDate, amazonAuthId,
user, ftype);
}

List<Map<String, Object>> listMap = new ArrayList<Map<String, Object>>();
for (Entry<String, Integer> typeentry : typesMap.entrySet()) {
    ftype = typeentry.getKey();
    Map<String, Object> map = new HashMap<String, Object>();
    SimpleDateFormat sdf = new SimpleDateFormat("MM.dd");
    Map<String, Object> tempmap = new HashMap<String, Object>();

    if ("ses".equals(ftype)) { // session
        for (int i = 0; i < sessionlist.size(); i++) {
            AmzProductPageviews item = sessionlist.get(i);
            tempmap.put(sdf.format(GeneralUtil.getDate(item.getByday())),
item.getSessions());
        }
        map.put("name", "session");
    } else if ("pgv".equals(ftype)) { // pageview

```

```

        for (int i = 0; i < sessionlist.size(); i++) {
            AmzProductPageviews item = sessionlist.get(i);
            tempmap.put(sdf.format(GeneralUtil.getDate(item.getByday())),
item.getPageViews());
        }
        map.put("name", "????");
    } else if ("bbp".equals(ftype)) { // BuyBox percentage ????
        for (int i = 0; i < sessionlist.size(); i++) {
            AmzProductPageviews item = sessionlist.get(i);
            tempmap.put(sdf.format(GeneralUtil.getDate(item.getByday())),
item.getBuyBoxPercentage());
        }
        map.put("name", "????");
    } else if ("osp".equals(ftype)) { // Unit_Session_Percentage ????
        for (int i = 0; i < sessionlist.size(); i++) {
            AmzProductPageviews item = sessionlist.get(i);
            tempmap.put(sdf.format(GeneralUtil.getDate(item.getByday())),
item.getUnitSessionPercentage());
        }
        map.put("name", "????");
    } else if ("uns".equals(ftype)) { // units orders ????
        for (int i = 0; i < orderlist.size(); i++) {
            OrdersSummary item = orderlist.get(i);
            tempmap.put(sdf.format(item.getPurchaseDate()), item.getQuantity());
        }
        map.put("name", "?");
    } else if ("pts".equals(ftype)) { // ????
        for (int i = 0; i < orderlist.size(); i++) {
            OrdersSummary item = orderlist.get(i);
            tempmap.put(sdf.format(item.getPurchaseDate()), item.getOrderprice());
        }
        map.put("name", "???");
    } else if ("ods".equals(ftype)) { // total order item ???
        for (int i = 0; i < orderlist.size(); i++) {
            OrdersSummary item = orderlist.get(i);
            tempmap.put(sdf.format(item.getPurchaseDate()), item.getOrdersum());
        }
        map.put("name", "????");
    } else if (ftype.equals("rnk")) {

```

```

        for (int i = 0; i < rnklist.size(); i++) {
            ProductRank item = rnklist.get(i);
            tempmap.put(sdf.format(item.getByday()), item.getRank());
        }
        map.put("name", "□□□□");
    } else if (typesMap.containsKey("rnks")) {
        if (ftype.equals("rnks")) {
            continue;
        }
        for (Map<String, Object> rankMap : rankList) {
            String rankName = rankMap.get("name").toString();
            if (ftype.equals(rankName)) {
                tempmap.put(sdf.format(rankMap.get("byday")), rankMap.get("rank"));
            }
            continue;
        }
        map.put("name", ftype);
    } else {
        if ("acoas".equals(ftype)) {
            long diff = GeneralUtil.getDatez(endDate).getTime() -
GeneralUtil.getDatez(beginDate).getTime();
            long daysize = diff / (1000 * 60 * 60 * 24);
            Calendar c = Calendar.getInstance();
            c.setTime(GeneralUtil.getDatez(beginDate));
            Map<String, Object> costmap = new HashMap<String, Object>();
            Map<String, Object> salesmap = new HashMap<String, Object>();
            for (int i = 0; i < advlist.size(); i++) {
                Map<String, Object> item = advlist.get(i);
                costmap.put(sdf.format(item.get("bydate")), item.get("spd"));
            }
            for (int i = 0; i < orderlist.size(); i++) {
                OrdersSummary item = orderlist.get(i);
                salesmap.put(sdf.format(item.getPurchaseDate()),
item.getOrderprice());
            }
            for (int i = 1; i <= daysize; i++, c.add(Calendar.DATE, 1)) {
                String tempkey = sdf.format(c.getTime());
                BigDecimal sales = new BigDecimal("0");
                Object obj = salesmap.get(tempkey);

```

```

        if (obj != null) {
            sales = new BigDecimal(obj.toString());
        }
        BigDecimal cost = new BigDecimal("0");
        Object obj2 = costmap.get(tempkey);
        if (obj2 != null) {
            cost = new BigDecimal(obj2.toString());
        }
        BigDecimal acoas = new BigDecimal("0");
        if (sales.compareTo(new BigDecimal("0")) != 0) {
            acoas = cost.multiply(new BigDecimal("100")).divide(sales, 2,
RoundingMode.HALF_DOWN);
        }
        tempmap.put(tempkey, acoas);
    }
} else if ("aups".equals(ftype)) { //□□□□□□=□□□□□□/□□□□
    long diff = GeneralUtil.getDatez(endDate).getTime() -
GeneralUtil.getDatez(beginDate).getTime();
    long daysize = diff / (1000 * 60 * 60 * 24);
    Calendar c = Calendar.getInstance();
    c.setTime(GeneralUtil.getDatez(beginDate));
    Map<String, Object> adUnitsmap = new HashMap<String, Object>();
    Map<String, Object> totalUnitsmap = new HashMap<String, Object>();
    for (int i = 0; i < advlist.size(); i++) {
        Map<String, Object> item = advlist.get(i);
        adUnitsmap.put(sdf.format(item.get("bydate")), item.get("aus"));
    }
    for (int i = 0; i < orderlist.size(); i++) {
        OrdersSummary item = orderlist.get(i);
        totalUnitsmap.put(sdf.format(item.getPurchaseDate()),
item.getQuantity());
    }
    for (int i = 1; i <= daysize; i++, c.add(Calendar.DATE, 1)) {
        String tempkey = sdf.format(c.getTime());
        BigDecimal totalUnits = new BigDecimal("0");
        Object obj = totalUnitsmap.get(tempkey);
        if (obj != null) {
            totalUnits = new BigDecimal(obj.toString());
        }
    }
}

```

```

        BigDecimal adUnits = new BigDecimal("0");
        Object obj2 = adUnitsmap.get(tempkey);
        if (obj2 != null) {
            adUnits = new BigDecimal(obj2.toString());
        }
        BigDecimal aups = new BigDecimal("0");
        if (totalUnits.compareTo(new BigDecimal("0")) != 0) {
            aups = adUnits.multiply(new BigDecimal("100")).divide(totalUnits,
2, RoundingMode.HALF_EVEN);
        }
        tempmap.put(tempkey, aups);
    }
    map.put("name", "□□□□□");
} else if(advlist!=null) {
    for (int i = 0; i < advlist.size(); i++) {
        Map<String, Object> item = advlist.get(i);
        tempmap.put(sdf.format(item.get("bydate")), item.get(ftype));
    }
}
if (ftype.equals("cks")) {
    map.put("name", "□□□□□");
} else if (ftype.equals("imp")) {
    map.put("name", "□□□□□");
} else if (ftype.equals("ctr")) {
    map.put("name", "□□□□□");
} else if (ftype.equals("spd")) {
    map.put("name", "□□□□");
} else if (ftype.equals("cpc")) {
    map.put("name", "□□□□□□");
} else if (ftype.equals("acos")) {
    map.put("name", "Acos");
} else if (ftype.equals("acoas")) {
    map.put("name", "AcoAs");
} else if (ftype.equals("cr")) {
    map.put("name", "□□□□□");
} else if (ftype.equals("tos")) {
    map.put("name", "□□□□□");
} else if (ftype.equals("aus")) {
    map.put("name", "□□□□");
}

```

```

    }
}

// 日期差值
long diff = GeneralUtil.getDatez(endDate).getTime() -
GeneralUtil.getDatez(beginDate).getTime();
long daysize = diff / (1000 * 60 * 60 * 24);
Calendar c = Calendar.getInstance();
c.setTime(GeneralUtil.getDatez(beginDate));
BigDecimal summary = new BigDecimal("0");
List<String> listLabel = new ArrayList<String>();
List<String> listData = new ArrayList<String>();
for (int i = 1; i <= daysize+1; i++, c.add(Calendar.DATE, 1)) {
    String tempkey = sdf.format(c.getTime());
    String value = tempmap.get(tempkey) == null ? "0" :
tempmap.get(tempkey).toString();
    listLabel.add(tempkey);
    listData.add(value);
    summary = summary.add(new BigDecimal(value));
}
map.put("summary", summary);
map.put("labels", listLabel);
map.put("data", listData);
listMap.add(map);
}
return listMap;
}

// 日期差值
private List<ProductRank> productRank(String sku, String marketplace, String beginDate,
String endDate, String amazonAuthId, UserInfo user, String ftype) {
    Map<String, Object> param = new HashMap<String, Object>();
    param.put("sku", sku);
    param.put("marketplaceid", marketplace);
    param.put("beginDate", beginDate);
    param.put("enddate", endDate);
    param.put("amazonAuthId", amazonAuthId);
    param.put("shopid", user.getCompanyid());
    List<ProductRank> list = productRankMapper.selectBySku(param);
}
}

```

```

        return list;
    }

    // 商品销量
    private List<Map<String, Object>> getCountRankBySku(String sku, String marketplace, String
amazonAuthId, UserInfo user) {
        Map<String, Object> param = new HashMap<String, Object>();
        param.put("sku", sku);
        param.put("marketplaceid", marketplace);
        param.put("amazonAuthId", amazonAuthId);
        param.put("shopid", user.getCompanyid());
        List<Map<String, Object>> list = productRankMapper.selectCountRankBySku(param);
        return getProductRank(list);
    }

    // 商品排名
    private List<Map<String, Object>> getRankBySku(String sku, String marketplace, String
beginDate, String endDate, String amazonAuthId, UserInfo user) {
        Map<String, Object> param = new HashMap<String, Object>();
        param.put("sku", sku);
        param.put("marketplaceid", marketplace);
        param.put("beginDate", beginDate);
        param.put("enddate", endDate);
        param.put("amazonAuthId", amazonAuthId);
        param.put("shopid", user.getCompanyid());
        List<Map<String, Object>> list = productRankMapper.selectRankBySku(param);
        return getProductRank(list);
    }

    // 商品广告
    private List<Map<String, Object>> advInfo(String sku, String marketplace, String
beginDate, String endDate, String amazonAuthId, UserInfo user, String ftype) {
        Map<String, Object> param = new HashMap<String, Object>();
        param.put("sku", sku);
        param.put("marketplaceid", marketplace);
        param.put("startdate", beginDate);
        param.put("enddate", endDate);
        param.put("amazonAuthId", amazonAuthId);
        param.put("shopid", user.getCompanyid());
    }

```

```

List<Map<String, Object>> list = null;
AmazonAuthority auth = iAmazonAuthorityService.getById(amazonAuthId);
param.put("sellerid", auth.getSellerid());
list=productInOptMapper.findAdvert(param);
return list;
}

// 商品排名
public List<Map<String, Object>> getProductRank(List<Map<String, Object>> list) {
    if (list != null) {
        Iterator<Map<String, Object>> iterator = list.iterator();
        while (iterator.hasNext()) {
            Map<String, Object> map = iterator.next();
            Object name = map.get("name");
            if (name == null) {
                Object categoryId = map.get("categoryId");
                try {
                    new BigInteger(categoryId.toString());
                    iterator.remove();
                } catch (Exception e) {
                    String strName = categoryId.toString().replace("_display_on_website",
"".replace("_and_", "&").replace("_", " ");
                    String[] categoryName = strName.toString().split("&");
                    String str = "";
                    for (int i = 0; i < categoryName.length; i++) {
                        String str1 = categoryName[i].substring(0, 1).toUpperCase() +
categoryName[i].substring(1);
                        if (i == categoryName.length - 1) {
                            str += str1;
                        } else {
                            str += str1 + " & ";
                        }
                    }
                    map.put("name", str);
                }
            }
        }
    }
    return list;
}

```

```

}

// 商品列表
public List<AmzProductPageviews> sessionPage(String sku, String marketplaceid, String
startdate, String enddate, String amazonAuthId, UserInfo user, String ftype) {
    Map<String, Object> param = new HashMap<String, Object>();
    param.put("sku", sku);
    param.put("marketplaceid", marketplaceid);
    param.put("startDate", startdate);
    param.put("endDate", enddate);
    param.put("amazonAuthId", amazonAuthId);
    param.put("shopid", user.getCompanyid());
    List<AmzProductPageviews> list = iAmzProductPageviewsService.findPageviews(param);
    return list;
}
}

```

4. ???????

4.1 ???????

商品 商品列表

商品 商品

1. 商品 `productAnysApi.productAsinList()` 商品
2. 商品 `ProductAnalysisController.productListAction()` 商品
3. 商品列表
4. 商品 `iProductInfoService.getAsinList()` 商品
5. 商品

4.2 ???????

商品 商品 ID商品

商品 商品

1. 商品 `productAnysApi.productdetail()` 商品
2. 商品 `ProductAnalysisController.productListAction()` 商品
3. 商品 `iProductInOrderService.selectDetialById()` 商品

5. ?????

5.1 ???????

1. [] [] Vue 3 Composition API []
2. [] [] Element Plus []
3. [] [] ECharts []
4. [] [] []
5. [] [] []

5.2 ???????

1. [] [] → [] → []
2. [] []
3. [] []
4. [] []
5. [] []

5.3 ???????

1. [] [] RESTful API []
2. [] []
3. [] []
4. [] []
5. [] []

6. ???????

6.1 ???????

1. [] []
 - [] DOM []
 - []
 - []
2. [] []
 - []
 - []
 - [] TypeScript []
3. [] []

