



1.按价格区间筛选

1.1 需求分析

点击搜索面板上的价格区间，实现按价格筛选

价格	0-500元	500-1000元	1000-1500元	1500-2000元	2000-3000元	3000元以上
----	--------	-----------	------------	------------	------------	---------

1.2 前端代码

1.2.1 前端控制层

(1) 修改 pinyougou-search-web 的 searchController.js 搜索条件的定义

```
$scope.searchMap={'keywords':'','category':'','brand':'','spec':{},'price':'' };//搜索条件封装对象
```

(2) 修改 pinyougou-search-web 的 searchController.js 添加搜索项和删除搜索项的方法

```
//添加搜索项

$scope.addSearchItem=function(key,value){

    if(key=='category' || key=='brand' || key=='price'){//如果点击的是分类或品牌

        $scope.searchMap[key]=value;

    }else{//如果是规格

        $scope.searchMap.spec[key]=value;

    }

    $scope.search();

}
```



```
//移除复合搜索条件

$scope.removeSearchItem=function(key){

    if(key=="category" || key=="brand" || key=='price'){//如果是分类或品牌

        $scope.searchMap[key]="";

    }else{//否则是规格

        delete $scope.searchMap.spec[key];//移除此属性

    }

    $scope.search();

}
```

1.2.2 页面

(1) 修改页面 search.html,在标签上调用方法

```
<ul class="type-list" ng-if="searchMap.price==''>

    <li>

        <a ng-click="addSearchItem('price', '0-500')">0-500 元</a>

    </li>

    <li>

        <a ng-click="addSearchItem('price', '500-1000')">500-1000 元</a>

    </li>

    <li>

        <a ng-click="addSearchItem('price', '1000-1500')">1000-1500 元</a>

    </li>

    <li>
```



```

        <a ng-click="addSearchItem('price', '1500-2000')">1500-2000 元</a>

    </li>

    <li>

        <a ng-click="addSearchItem('price', '2000-3000')">2000-3000 元 </a>

    </li>

    <li>

        <a ng-click="addSearchItem('price', '3000-*')">3000 元以上</a>

    </li>

</ul>

```

(2) 修改 search.html，增加面包屑

```

<li class="tag" ng-if="searchMap.price!=''" ng-click="removeSearchItem('price')">价格:
{{searchMap.price}}<i class="sui-icon icon-tb-close"></i></li>

```

1.3 后端代码

修改 pinyougou-search-service 的 ItemSearchServiceImpl.java

```

/**

 * 根据关键字搜索列表

 * @param keywords

 * @return

 */

private Map searchList(Map searchMap){

    .....

    //1.1 关键字查询.....

    //1.2 按分类筛选.....

```



```
//1.3 按品牌筛选.....

//1.4 过滤规格.....

//1.5 按价格筛选.....

if(!"".equals(searchMap.get("price"))){

    String[] price = ((String) searchMap.get("price")).split("-");

    if(!price[0].equals("0")){//如果区间起点不等于 0

        Criteria filterCriteria=new
Criteria("item_price").greaterThanEqual(price[0]);

        FilterQuery filterQuery=new SimpleFilterQuery(filterCriteria);

        query.addFilterQuery(filterQuery);

    }

    if(!price[1].equals("*")){//如果区间终点不等于*

        Criteria filterCriteria=new
Criteria("item_price").lessThanEqual(price[1]);

        FilterQuery filterQuery=new SimpleFilterQuery(filterCriteria);

        query.addFilterQuery(filterQuery);

    }

}

//高亮显示处理      .....

}
```

2.搜索结果分页

2.1 需求分析

在上述功能基础上实现分页查询



«上一页»

1

2

3

4

...

下一页»

共4页 到第

1

页

确定

2.2 后端代码

修改 pinyougou-search-service 工程 ItemSearchServiceImpl.java

```
/**
 * 根据关键字搜索列表
 * @param keywords
 * @return
 */

private Map searchList(Map searchMap){

    .....

    //1.1 关键字查询

    .....

    //1.2 按分类筛选

    .....

    //1.3 按品牌筛选

    .....

    //1.4 过滤规格

    .....

    //1.5 按价格筛选

    .....

    //1.6 分页查询

    Integer pageNo= (Integer) searchMap.get("pageNo");//提取页码
```



```
        if(pageNo==null){

            pageNo=1;//默认第一页

        }

        Integer pageSize=(Integer) searchMap.get("pageSize");//每页记录数

        if(pageSize==null){

            pageSize=20;//默认 20

        }

        query.setOffset((pageNo-1)*pageSize);//从第几条记录查询

        query.setRows(pageSize);

        //高亮显示处理

        .....

        Map map=new HashMap<>();

        map.put("rows", page.getContent());

        map.put("totalPages", page.getTotalPages());//返回总页数

        map.put("total", page.getTotalElements());//返回总记录数

        return map;

    }
```

2.3 前端代码

2.3.1 构建分页标签

需求：

(1) 如果我们需要修改默认页码和每页记录数，可以修改 searchController.js 的 searchMap，为搜索对象添加属性。



```
$scope.searchMap={ 'keywords': '', 'category': '', 'brand': '', 'spec': {}, 'price': '',  
  
    'pageNo':1, 'pageSize':40 };//搜索条件封装对象
```

(2) 修改 searchController.js 实现页码的构建

```
//构建分页标签(totalPages 为总页数)  
  
buildPageLabel=function(){  
  
    $scope.pageLabel=[];//新增分页栏属性  
  
    var maxPageNo= $scope.resultMap.totalPages;//得到最后页码  
  
    var firstPage=1;//开始页码  
  
    var lastPage=maxPageNo;//截止页码  
  
    if($scope.resultMap.totalPages> 5){ //如果总页数大于 5 页,显示部分页码  
  
        if($scope.searchMap.pageNo<=3){//如果当前页小于等于 3  
  
            lastPage=5; //前 5 页  
  
        }else if( $scope.searchMap.pageNo>=lastPage-2 ){//如果当前页大于等于最大  
页码-2  
  
            firstPage= maxPageNo-4; //后 5 页  
  
        }else{ //显示当前页为中心的 5 页  
  
            firstPage=$scope.searchMap.pageNo-2;  
  
            lastPage=$scope.searchMap.pageNo+2;  
  
        }  
  
    }  
  
    //循环产生页码标签  
  
    for(var i=firstPage;i<=lastPage;i++){  
  
        $scope.pageLabel.push(i);  
  
    }  
}
```



```
}
```

(3) 在查询后调用此方法

```
//搜索

$scope.search=function(){

    searchService.search( $scope.searchMap ).success(

        function(response){

            $scope.resultMap=response;//搜索返回的结果

            buildPageLabel();//调用

        }

    );

}
```

(4) 修改 search.html, 循环产生页码

```
<ul>

    <li class="prev disabled">

        <a href="#">«上一页</a>

    </li>

    <li ng-repeat="p in pageLabel">

        <a href="#" >{{p}}</a>

    </li>

    <li class="next">

        <a href="#">下一页»</a>

    </li>

</ul>
```




```
<div>

    <span>&nbsp;共{{resultMap.totalPages}}页&nbsp;</span>

    .....

</div>
```

(5) 显示总条数

搜索结果: {{resultMap.total}}条记录

2.3.2 提交页码查询

(1) 在 searchController.js 增加方法，修改页码执行查询

```
//根据页码查询

$scope.queryByPage=function(pageNo){

    //页码验证

    if(pageNo<1 || pageNo>$scope.resultMap.totalPages){

        return;

    }

    $scope.searchMap.pageNo=pageNo;

    $scope.search();

}
```

(2) 修改页码调用方法

```
<div class="sui-pagination pagination-Large">

    <ul>

        <li class="prev">

            <a href="#" ng-click="queryByPage(searchMap.pageNo-1)"><</a>
```



```

        </li>

        <li ng-repeat="p in pageLabel">

            <a href="#" ng-click="queryByPage(p)">{{p}}</a>

        </li>

        <li class="next">

            <a href="#" ng-click="queryByPage(searchMap.pageNo+1)">>></a>

        </li>

    </ul>

    <div>

        <span>&nbsp;共{{resultMap.totalPages}}页&nbsp;</span>

        <span> 到第 <input type="text" class="page-num" ng-model="searchMap.pageNo">页

            <button class="page-confirm" ng-click="queryByPage(searchMap.pageNo)">确定
    </button></span>

    </div>

</div>

```

(3) 修改 search 方法, 在执行查询前, 转换为 int 类型, 否则提交到后端有可能变成字符串

```

//搜索

$scope.search=function(){

    $scope.searchMap.pageNo= parseInt($scope.searchMap.pageNo) ;

    .....

}

```

2.3.3 显示省略号

```

//构建分页栏

```



```
buildPageLabel=function(){  
  
    //构建分页栏  
  
    $scope.pageLabel=[];  
  
    var firstPage=1;//开始页码  
  
    var lastPage=$scope.resultMap.totalPages;//截止页码  
  
    $scope.firstDot=true;//前面有点  
  
    $scope.lastDot=true;//后边有点  
  
    if($scope.resultMap.totalPages>5){ //如果页码数量大于 5  
  
        if($scope.searchMap.pageNo<=3){//如果当前页码小于等于 3 ， 显示前 5 页  
  
            lastPage=5;  
  
            $scope.firstDot=false;//前面没点  
  
        }else if( $scope.searchMap.pageNo>= $scope.resultMap.totalPages-2 ){//  
显示后 5 页  
  
            firstPage=$scope.resultMap.totalPages-4;  
  
            $scope.lastDot=false;//后边没点  
  
        }else{ //显示以当前页为中心的 5 页  
  
            firstPage=$scope.searchMap.pageNo-2;  
  
            lastPage=$scope.searchMap.pageNo+2;  
  
        }  
  
    }else{  
  
        $scope.firstDot=false;//前面无点  
  
        $scope.lastDot=false;//后边无点  
  
    }  
  
    //构建页码
```



```
for(var i=firstPage;i<=lastPage;i++){  
  
    $scope.pageLabel.push(i);  
  
}  
  
}
```

修改页面：页码前的省略号

```
<li class="dotted" ng-if="firstDot==true"><span>...</span></li>
```

页码后的省略号

```
<li class="dotted" ng-if="lastDot==true"><span>...</span></li>
```

2.3.4 页码不可用样式

修改 searchController.js 增加方法

```
//判断当前页为第一页  
  
$scope.isTopPage=function(){  
  
    if($scope.searchMap.pageNo==1){  
  
        return true;  
  
    }else{  
  
        return false;  
  
    }  
  
}  
  
  
  
//判断当前页是否未最后一页  
  
$scope.isEndPage=function(){  
  
    if($scope.searchMap.pageNo==$scope.resultMap.totalPages){
```



```
        return true;

    }else{

        return false;

    }

}
```

(2) 修改页面

```
<ul>

    <li class="prev {{isTopPage()? 'disabled': ''}}">

        <a href="#" ng-click="queryByPage(searchMap.pageNo-1)">« 上一页</a>

    </li>

    <li ng-repeat="p in resultMap.pageLabel">

        <a href="#" ng-click="queryByPage(p)">{{p}}</a>

    </li>

    <li class="dotted"><span>...</span></li>

    <li class="next {{isEndPage()? 'disabled': ''}}">

        <a href="#" ng-click="queryByPage(searchMap.pageNo+1)">下一页»</a>

    </li>

</ul>
```

2.3.5 搜索起始页码处理

测试：如果我们先按照“手机”关键字进行搜索，得出的页数是 19 页，然后我们点击第 18 页进行查询，然后我们再根据“三星”关键字搜索，会发现没有结果显示。是因为当前页仍然为 18，而三星的结果只有 4 页，所以无法显示。我们需要在每次点击查询时将页码设置为 1。

修改搜索按钮，调用搜索前将起始页码设置为 1



```
<button class="sui-btn btn-xlarge btn-danger" ng-click="searchMap.pageNo=1;search()" type="button">搜索</button>
```