**搜索引擎找房业务逻辑说明**

**Search Room Document**

|  |  |
| --- | --- |
| **项目名称：** | **新官网、租客App** |
| **项目编号：** | cmd-20160811 |
| **作者：** | 雷传盛 |

**版权所有©上海墨果资产管理有限公司**

**Copyright ©Mogoroom Forture Management Co.,Ltd**.

目录

[1. 全局说明 2](#_Toc458760211)

[1.1 代码版本 2](#_Toc458760212)

[1.1.1 代码版本 2](#_Toc458760213)

[2. 官网找房 2](#_Toc458760214)

[2.1 官网找房 2](#_Toc458760215)

[2.1.1 代码导读 2](#_Toc458760216)

[2.1.2 处理流程 2](#_Toc458760217)

[2.1.3 参数解析 3](#_Toc458760218)

[2.2 App找房 6](#_Toc458760219)

[2.2.1 代码导读 6](#_Toc458760220)

[2.2.2 处理流程 6](#_Toc458760221)

[2.2.3 参数解析 7](#_Toc458760222)

# 全局说明

## 代码版本

### 代码版本

本文档说明全部基于cross\_svc\_eq分支的代码，因此阅读本文档前请pull一下最新代码。

# 搜索说明

## 官网找房

### 代码导读

接口：com.mogoroom.service.flat.IRoomSearchService

实现类：com.mogoroom.service.flat.impl.RoomSearchServiceImpl

方法定义：public JSONObject searchRoomForPage(Map<String, String> paras, int pageNum, int pageSize, boolean isNeedAttached, int aggregationTerm) throws Exception

测试类：com.mogoroom.test.service.searchRoomForHttpTest

### 处理流程

1. 获取城市Id，若不存在则默认为289-上海；
2. 运用Elasticsearch的java api组装查询条件；
3. 分页参数组装；
4. 根据sort参数设置排序规则；
5. 根据aggregationTerm组装汇总参数，请注意：不管aggregationTerm是什么值，每次请求都会按优选品牌进行汇总，因为在新官网中每次都要显示当前搜索结果涵盖了哪些优选品牌如下：



1. 执行搜索；
2. 循环处理列表记录，具体如下：
3. 重新定义title；
4. 设置蘑菇认证标识；
5. 如果搜索条件中包含地铁线或地铁站，则过滤掉不符合当前搜索条件的地铁信息metroInfo；
6. 设置汇总信息
7. 若按小区汇总(aggregationTerm=1)，循环从bucket获取小区Id，然后：

* 从缓存中获取其对应的小区信息，若缓存中没有，则从es的rooms.community中获取其对应的小区信息且存入缓存中；
* 设置分散式或集中式公寓标识
* 设置房源总数；
* 设置小区最低价格；

1. 若按区域汇总(aggregationTerm=2)，循环从bucket获取区域Id，然后：

* 从缓存中获取其对应的区域信息，若缓存中没有，则从es的rooms.district中获取其对应的区域信息且存入缓存中；
* 设置区域房源总数；
* 重新定义经纬度的key名，且删除旧名；

1. 若按城市汇总(aggregationTerm=3)，从es.rooms.city中获取所有已开发的城市，循环遍历这些城市，然后：

* 获取城市Id，若城市Id与当前切换城市Id一致，则从直接当前bucket中获取城市信息且设置房源总数；
* 若城市Id与当前切换城市Id不一致，则需要根据城市Id和优选品牌Id(如果存在的话)从搜索引擎中获取城市信息及其房源总数；
* 设置待开发城市信息(参照枚举MapCitiesEnum)，且将房源总数设置成0，这些城市在地图上会以小蘑菇的形式出现。

1. 若按商圈汇总(aggregationTerm=4)，循环从bucket获取商圈Id，然后：

* 从缓存中获取其对应的商圈信息，若缓存中没有，则es的rooms.business中获取其对应的区域信息且存入缓存中；
* 设置商圈房源总数；

1. 设置优选品牌汇总信息，首先循环从优选品牌对应的bucket中获取优选品牌Id，然后：

* 从缓存中获取其对应的优选品牌信息，若缓存中没有，则从es的rooms.brand中获取其对应的区域信息且存入缓存中；
* 设置房源总数；
* 设置最低价格好最高价格；
* 设置品牌描述；

1. 添加附属信息(isNeedAttached=true)，将新官网的查询条件返回给页面：城市列表、区域、地铁信息(地铁线、地铁站)、默认城市。
2. 返回结果。

### 参数解析

1. paras

以下是对其对应的key进行说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数** | **名称** | **类型** | **是否必输** | **说明** |
| cityId | 城市Id | int | 是 |  |
| sort | 排序 | int | 否 | 0-推荐排序，1-价格升 2-价格降，3-面积升 4-面积降， 5-房态释放时间降， 6-预计入住时间升， 7-按距离，由近及远 |
| roomId | 房间Id | int | 否 |  |
| districtId | 区域Id集合 | jsonArray | 否 | 如：[3,4,5] |
| areaId | 商圈Id | int | 否 |  |
| areaIds | 商圈Id集合 | jsonArray | 否 | 如：[34,35,36] |
| nearSearch | 附近找房 | json | 否 | 如：{  "lng": 121.490857,  "lat": 31.243738,  "distance": 1000 } |
| multi\_nearSearch | 公司找房 | json | 否 | 如：{  "stationIds": [  17,  18,  19  ],  "distance": 1000 } |
| stationId | 地铁站Id | int | 否 |  |
| stationIds | 地铁站Id集合 | jsonArray | 否 | 如：[17,18,19] |
| subwayId | 地铁线Id | int | 否 |  |
| subwayIds | 地铁线Id集合 | jsonArray | 否 | 如：[1,2,3] |
| metro | 地铁线和地铁站集合 | jsonArray | 否 | 如：{  "subwayIds": [  1,  2,  3  ],  "stationIds": [  17,  18,  19  ] } |
| zone | 区域与商圈集合 | jsonArray | 否 | 如：{  "districtIds": [  2,  3,  4  ],  "areaIds": [  34,  35,  36  ] } |
| brandId | 优选品牌Id | int | 否 |  |
| communityId | 小区Id | int | 否 |  |
| minPrice | 最低价 | int | 否 |  |
| maxPrice | 最高价 | int | 否 |  |
| houseType | 户型 | jsonArray | 否 | 如：[1,2,3] |
| brandType | 品牌类型 | int | 否 | 0-非品牌， 1-优选品牌， 2-普通品牌 |
| roomMateGender | 室友性别 | int | 否 | 1-全是美女， 2-全是帅哥， 3-男女搭配 |
| roomMateCnstl | 室友星座 | jsonArray | 否 | 如：[ 'constellation-1', 'constellation-2' ] |
| readyDate | 入住时间 | int | 否 | 1-立即入住， 2-一周内入住， 3-两周内入住， 4-一个月内入住 |
| decoration | 装修风格 | string | 否 | 参考字典groupName='style' |
| toilet | 厕所 | int | 否 | 1-有，0-无 |
| veranda | 阳台 | int | 否 | 1-有，0-无 |
| bayWindow | 飘窗 | int | 否 | 1-有，0-无 |
| hasTV | 电视 | int | 否 | 1-有，0-无 |
| airCond | 空调 | int | 否 | 1-有，0-无 |
| hasPic | 图片 | int | 否 | 1-有，0-无 |
| hasWifi | WIFI | int | 否 | 1-有，0-无 |
| rentType | 租赁方式集合 | jsonArray | 否 | 如：[1,2,3] |
| orientation | 朝向 | string | 否 |  |
| level | 房源等级 | jsonArray | 否 | 如：[1,2,3,4] |
| hasService | 是否为酒店式公寓 | int | 否 | 即判断是否为集中式公寓，1-是，0-否 |
| landlordId | 房东Id | int | 否 |  |
| searchWord | 搜索关键字 | string | 否 | 匹配区域名称、小区名称、 商圈名称、地铁信息、房间号码 |
| checkIn | 入住时间 | string | 否 |  |
| hobby | 室友爱好 | string | 否 |  |
| career | 室友职业 | string | 否 |  |
| statusUpdateDate | 房态更新时间 | int | 否 | 1-最近上新 |
| minLng | 最小纬度 | double | 否 |  |
| maxLng | 最大纬度 | double | 否 |  |
| minLat | 最小经度 | double | 否 |  |
| maxLat | 最大经度 | double | 否 |  |

1. pageNum

下一页码

1. pageSize

每页显示记录数

1. isNeedAttached

是否需要附加信息，true-是，false-否，通常第一次进入官网列表页面为true。

1. aggregationTerm

汇总类型，请参考枚举AggregationTermEnum

## App找房

### 代码导读

接口：com.mogoroom.service.flat.IRoomSearchService

实现类：com.mogoroom.service.flat.impl.RoomSearchServiceImpl

方法定义：public JSONObject searchRoomForApp(Map<String, String> paras, int pageNum, int pageSize, boolean isNeedSummary) throws Exception

测试类：com.mogoroom.test.service.searchRoomForAppTest

### 处理流程

1. 获取城市Id，若不存在则默认为289-上海；
2. 运用Elasticsearch的java api组装查询条件；
3. 分页参数组装；
4. 根据sort参数设置排序规则；
5. 按小区和行政区域组装汇总请求参数
6. 执行搜索请求；
7. 循环处理列表记录，具体如下：
8. 重新定义title；
9. 重新定义image，使其符合手机尺寸；
10. 重新定义蘑菇认证标识(label)，若brandId不为空(即优选品牌房源)，则获得蘑菇认证(即label=1)，否则不能获得蘑菇认证(label=0)；
11. 计算房间距离，若为附近找房(nearbyMark=1)，则计算目标经纬度与当前房源所在小区的距离；
12. 如果搜索条件中包含地铁线或地铁站，则过滤掉不符合当前搜索条件的地铁信息metroInfo；
13. 将找房列表对象数据的某几个字段封装成房间详情页需要的几个字段；
14. 设置汇总数据(即isNeedSummary=true时)，具体如下：
15. 设置小区汇总信息，从bucket中获取小区汇总数据，且循环遍历作如下处理：

* 检查小区是否存在；
* 设置小区房源数量；
* 设置小区房源最低价格；

1. 设置行政区域汇总信息，从bucket中获取区域汇总数据，且循环遍历作如下处理：

* 检查区域是否存在；
* 设置区域房源数量；
* 设置区域房源最低价格；

1. 设置地铁站汇总信息，从bucket中获取地铁站汇总数据，且循环遍历作如下处理：

* 检查地铁站是否存在；
* 设置房间数量；
* 设置最低价格；
* 地铁站对应的小区汇总，且按最低价格升序存储；

1. 返回结果。

### 参数解析

1. paras

以下是对其对应的key进行说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数** | **名称** | **类型** | **是否必输** | **说明** |
| cityId | 城市Id | int | 是 |  |
| sort | 排序 | int | 否 | 0-推荐排序，1-价格升 2-价格降，3-面积升 4-面积降， 5-房态释放时间降， 6-预计入住时间升， 7-按距离，由近及远 |
| roomId | 房间Id | int | 否 |  |
| districtId | 区域Id集合 | jsonArray | 否 | 如：[3,4,5] |
| areaId | 商圈Id | int | 否 |  |
| areaIds | 商圈Id集合 | jsonArray | 否 | 如：[34,35,36] |
| nearSearch | 附近找房 | json | 否 | 如：{  "lng": 121.490857,  "lat": 31.243738,  "distance": 1000 } |
| multi\_nearSearch | 公司找房 | json | 否 | 如：{  "stationIds": [  17,  18,  19  ],  "distance": 1000 } |
| stationId | 地铁站Id | int | 否 |  |
| stationIds | 地铁站Id集合 | jsonArray | 否 | 如：[17,18,19] |
| subwayId | 地铁线Id | int | 否 |  |
| subwayIds | 地铁线Id集合 | jsonArray | 否 | 如：[1,2,3] |
| metro | 地铁线和地铁站集合 | jsonArray | 否 | 如：{  "subwayIds": [  1,  2,  3  ],  "stationIds": [  17,  18,  19  ] } |
| zone | 区域与商圈集合 | jsonArray | 否 | 如：{  "districtIds": [  2,  3,  4  ],  "areaIds": [  34,  35,  36  ] } |
| brandId | 优选品牌Id | int | 否 |  |
| communityId | 小区Id | int | 否 |  |
| minPrice | 最低价 | int | 否 |  |
| maxPrice | 最高价 | int | 否 |  |
| houseType | 户型 | jsonArray | 否 | 如：[1,2,3] |
| brandType | 品牌类型 | int | 否 | 0-非品牌， 1-优选品牌， 2-普通品牌 |
| roomMateGender | 室友性别 | int | 否 | 1-全是美女， 2-全是帅哥， 3-男女搭配 |
| roomMateCnstl | 室友星座 | jsonArray | 否 | 如：[ 'constellation-1', 'constellation-2' ] |
| readyDate | 入住时间 | int | 否 | 1-立即入住， 2-一周内入住， 3-两周内入住， 4-一个月内入住 |
| decoration | 装修风格 | string | 否 | 参考字典groupName='style' |
| toilet | 厕所 | int | 否 | 1-有，0-无 |
| veranda | 阳台 | int | 否 | 1-有，0-无 |
| bayWindow | 飘窗 | int | 否 | 1-有，0-无 |
| hasTV | 电视 | int | 否 | 1-有，0-无 |
| airCond | 空调 | int | 否 | 1-有，0-无 |
| hasPic | 图片 | int | 否 | 1-有，0-无 |
| hasWifi | WIFI | int | 否 | 1-有，0-无 |
| rentType | 租赁方式集合 | jsonArray | 否 | 如：[1,2,3] |
| orientation | 朝向 | string | 否 |  |
| level | 房源等级 | jsonArray | 否 | 如：[1,2,3,4] |
| hasService | 是否为酒店式公寓 | int | 否 | 即判断是否为集中式公寓，1-是，0-否 |
| landlordId | 房东Id | int | 否 |  |
| searchWord | 搜索关键字 | string | 否 | 匹配区域名称、小区名称、 商圈名称、地铁信息、房间号码 |
| checkIn | 入住时间 | string | 否 |  |
| hobby | 室友爱好 | string | 否 |  |
| career | 室友职业 | string | 否 |  |
| statusUpdateDate | 房态更新时间 | int | 否 | 1-最近上新 |
| minLng | 最小纬度 | double | 否 |  |
| maxLng | 最大纬度 | double | 否 |  |
| minLat | 最小经度 | double | 否 |  |
| maxLat | 最大经度 | double | 否 |  |
| nearbyMark | 附近找房 | int | 否 | 1-附近找房 |
| lng | 纬度 | double | 否 | 若为附近找房，该值不能 为空 |
| lat | 经度 | double | 否 | 若为附近找房，该值不能 为空 |

1. pageNum

下一页码

1. pageSize

每页显示记录数

1. isNeedSummary

是否需要汇总数据，true-是，fale-否

## 关键字搜索

### 代码导读

URI：/elastic/keyword4j(renter、renterpc都有)

Controller：com.mogoroom.renterpc.controller.ElasticSearchController(renter、renterpc都有)

Facade接口：com.mogoroom.facade.IElasticSearchFacade

Facade实现类：com.mogoroom.facade.impl.ElasticSearchFacadeImpl

Facade方法：public List<Map<String, Object>> searchWordHintByHttp(Long cityId, String searchWord, ExtKeywordParamCriteria criteria)

### 处理流程

1. 收集好请求参数：城市Id(cityId)、搜索关键字(searchWord)、搜索额外条件(ExtKeywordParamCriteria)；
2. 向搜索引擎发送请求，索引(keywords)、类型(wordHint)，按以下步骤发送请求：
3. 对域中文名称(name)作分词匹配(match)

* 检查搜索额外条件(ExtKeywordParamCriteria)是否存在排除类型(excludeTypes)，若存在，则需要在请求条件中加上相关的非操作(mustNot)；
* 按条件执行搜索请求，若返回结果为空，则执行第2)步；

1. 对域短拼(shortPinyin)作前缀匹配(prefix)

* 检查搜索额外条件(ExtKeywordParamCriteria)是否存在排除类型(excludeTypes)，若存在，则需要在请求条件中加上相关的非操作(mustNot)；
* 按条件执行搜索请求，若返回结果为空，则执行第3)步；

1. 对域短拼(shortPinyin)作分词匹配(match)

* 检查搜索额外条件(ExtKeywordParamCriteria)是否存在排除类型(excludeTypes)，若存在，则需要在请求条件中加上相关的非操作(mustNot)；
* 按条件执行搜索请求，若返回结果为空，则执行第4)步；

1. 对域全拼(fullPinyin)作前缀匹配(prefix)

* 检查搜索额外条件(ExtKeywordParamCriteria)是否存在排除类型(excludeTypes)，若存在，则需要在请求条件中加上相关的非操作(mustNot)；
* 按条件执行搜索请求，若返回结果为空，则执行第5)步；

1. 对域全拼(fullPinyin)作分词匹配(prefix)

* 检查搜索额外条件(ExtKeywordParamCriteria)是否存在排除类型(excludeTypes)，若存在，则需要在请求条件中加上相关的非操作(mustNot)；
* 按条件执行搜索请求。

1. 计算每个词条对应的房间数量

按以下方式组装请求条件

1. 必需条件：城市Id(cityId)；
2. 根据词条类型(type)，找出其在rooms索引和room类型中对应的域名(请参考枚举KeywordEnum)，然后组装其条件；
3. 若搜索额外条件(ExtKeywordParamCriteria)中存在优选品牌Id(brandId)，则表示该搜索是在品牌主页进行的搜索，需要组装品牌Id作为请求条件；

执行搜索请求；

1. 返回结果。

### 参数解析

1. cityId

城市Id，每次请求必输参数

1. searchWord

搜索关键字，必输参数

1. criteria

额外搜索条件，非必输参数，其内包含单词联想排除类型(excludeTypes)和优选品牌Id(brandId)，其中若包含优选品牌Id，则表示该单词联想是在品牌主页中进行的搜索，请参考类：

com.mogoroom.service.flat.domain.ExtKeywordParamCriteria