# 黑马旅游网-聚合

## 需求

同学们，接下来就是检验我们的聚合的学习成果的时候了，这里我们提出了黑马旅游网的需求：

在IUserService中定义方法，实现对品牌、城市、星级的聚合！

需求解读：在搜索页面的品牌、城市等信息不应该在页面写死，而是通过聚合hotel索引库中的酒店数据得来的：



为什么会有这样一个需求呢？可以看上面页面的红框，是我们黑马旅游网的搜索页面的顶部导航部分，这块有很多的过滤条件，用户点击这里的过滤条件就会对搜索结果做过滤！筛选出符合条件的那些酒店数据。那这些过滤条件是从何而来呢？是不是直接写死在页面的呢？目前是写死在页面的，但是在真实情况中，城市信息、品牌信息、星级信息肯定不可能是写死的！这里展示的过滤条件不能是静态的，而必须是hotel索引库中包含了哪些城市信息、哪些品牌信息、哪些星级信息，网页就应该展示哪些信息！

那么问题来了，我们怎么知道hotel索引库中包含了哪些城市信息、哪些品牌信息、哪些星级信息？简单，对hotel索引库做聚合啊！这个问题其实回到了聚合知识之初：如何在海量数据中知道品牌数据有哪些？这不就是聚合功能吗？？所以我们只需要对hotel索引库中的数据，针对品牌字段、城市字段、星级字段去做聚合，就能得到hotel索引库中包含哪些品牌、城市、星级了，而后我们再把这些结果数据封装返回给前端，一渲染不就可以解决问题了吗！

## 代码实现

废话不多说，看代码：

**@Override**

**public Map<String, List<String>> filters() {**

**try {**

**// 准备Request**

**SearchRequest request = new SearchRequest("hotel");**

**// 准备DSL**

**request.source().size(0);**

**// 城市聚合**

**request.source().aggregation(**

**AggregationBuilders**

**.terms("cityAgg")**

**.field("city")**

**.size(100)**

**);**

**// 品牌聚合**

**request.source().aggregation(**

**AggregationBuilders**

**.terms("brandAgg")**

**.field("brand")**

**.size(100)**

**);**

**// 品牌聚合**

**request.source().aggregation(**

**AggregationBuilders**

**.terms("starAgg")**

**.field("starName")**

**.size(100)**

**);**

**// 发送请求**

**SearchResponse response = client.search(request, RequestOptions.DEFAULT);**

**// 结果解析**

**Aggregations aggregations = response.getAggregations();**

**Map<String, List<String>> filters = new HashMap<>();**

**// 解析城市聚合**

**Terms cityAgg = aggregations.get("cityAgg");**

**List<? extends Terms.Bucket> cityBuckets = cityAgg.getBuckets();**

**List<String> cites = new ArrayList<>();**

**for (Terms.Bucket bucket : cityBuckets) {**

**String city = bucket.getKeyAsString();**

**cites.add(city);**

**}**

**filters.put("城市", cites);**

**// 解析品牌聚合**

**Terms brandAgg = aggregations.get("brandAgg");**

**List<? extends Terms.Bucket> brandBuckets = brandAgg.getBuckets();**

**List<String> brands = new ArrayList<>();**

**for (Terms.Bucket bucket : brandBuckets) {**

**String brand = bucket.getKeyAsString();**

**brands.add(brand);**

**}**

**filters.put("品牌", brands);**

**// 解析星级聚合**

**Terms starAgg = aggregations.get("starAgg");**

**List<? extends Terms.Bucket> starBuckets = starAgg.getBuckets();**

**List<String> starNames = new ArrayList<>();**

**for (Terms.Bucket bucket : starBuckets) {**

**String starName = bucket.getKeyAsString();**

**starNames.add(starName);**

**}**

**filters.put("星级", starNames);**

**return filters;**

**} catch (Exception e) {**

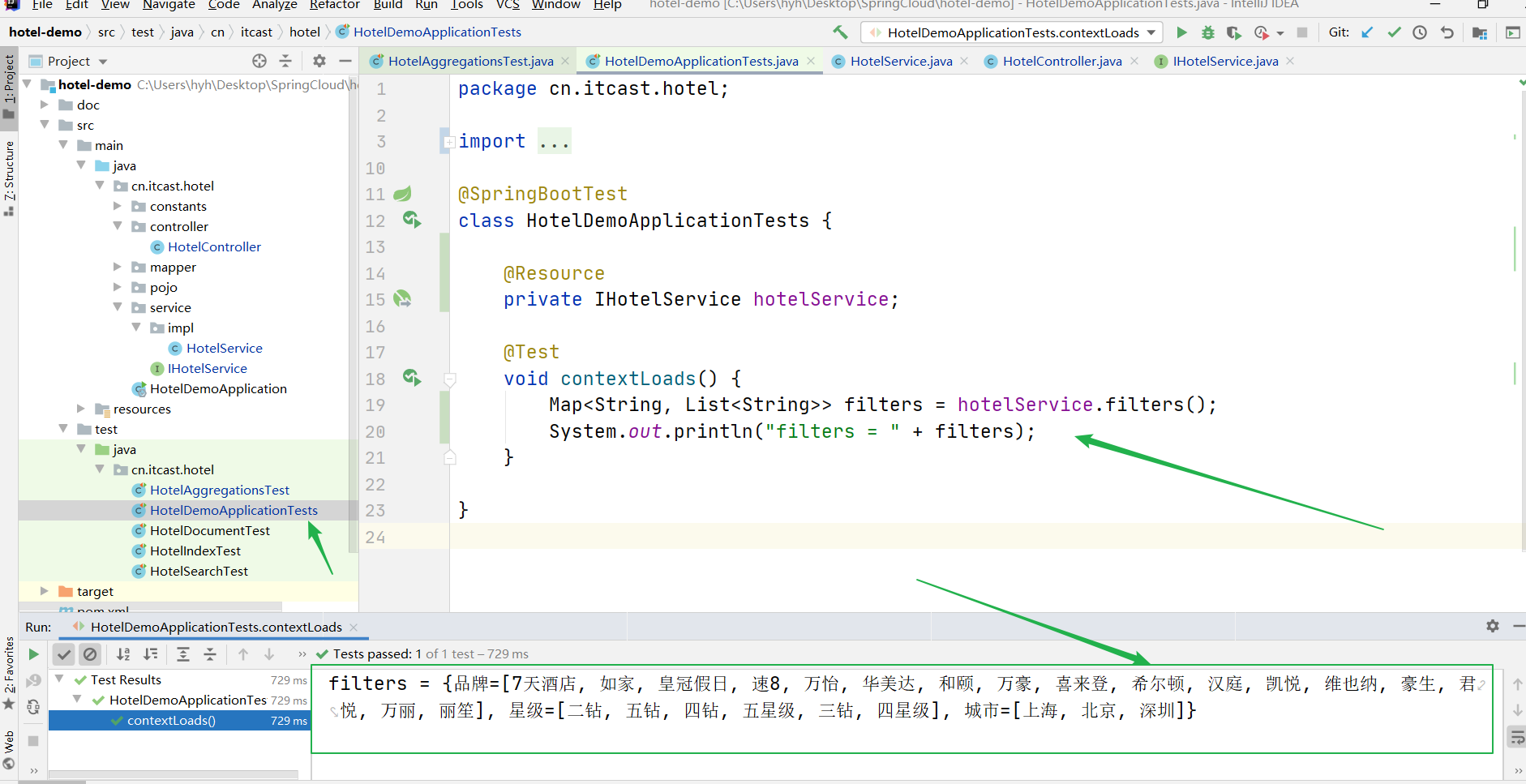
**throw new RuntimeException(e);**

**}**

**}**

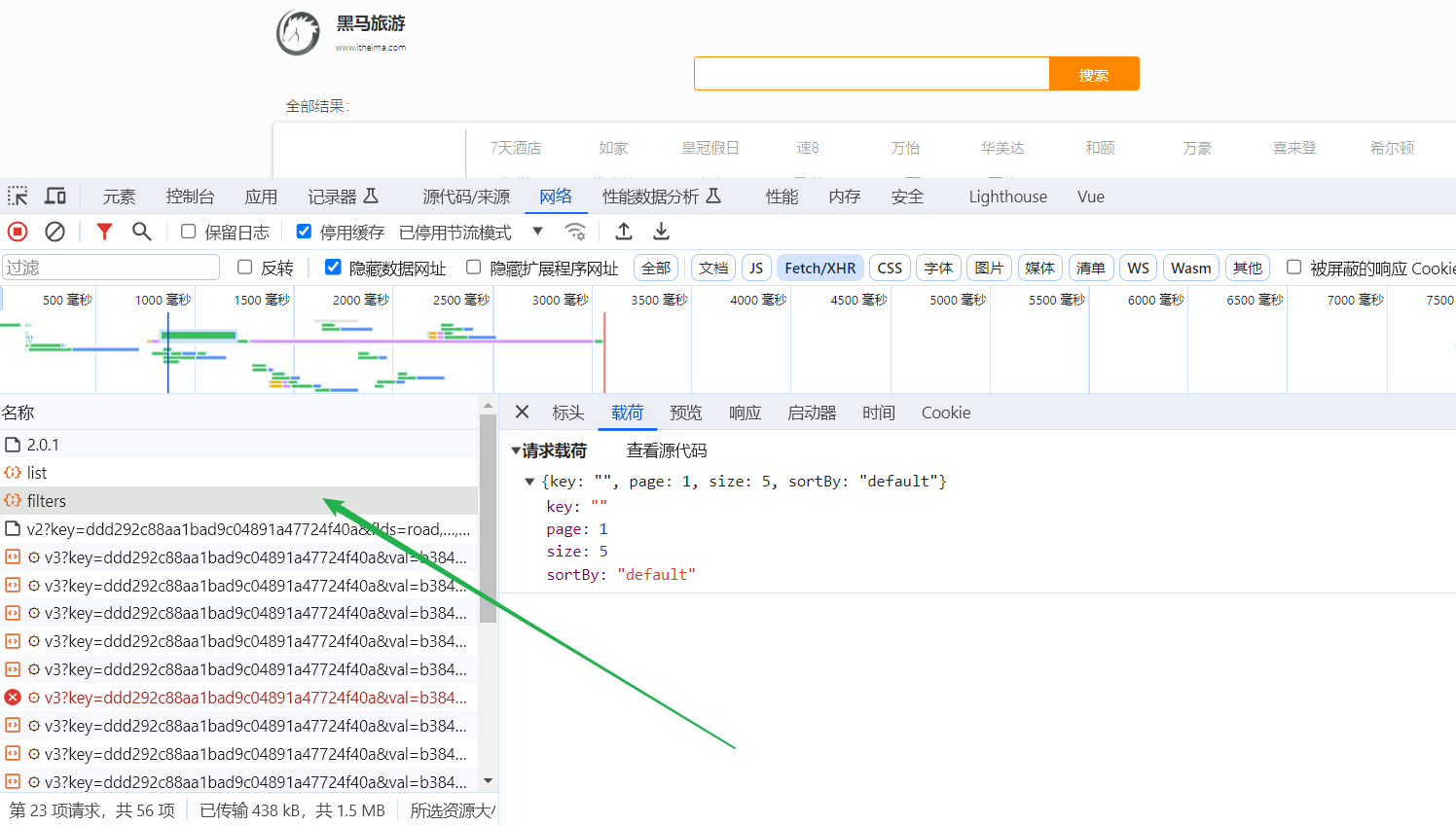
## 运行测试用例方法，测试filters方法

可以看到，完全没问题！



上面我们已经在黑马旅游网中实现了对城市、品牌、星级三个字段的聚合，只不过这个聚合的结果我们还没有返回给前端，所以我们还没有在前端看到最终效果！现在我们先来分析一下前端的请求，然后在服务端接收请求，把最终结果返回给前端，最终实现想要的效果！

先看浏览器：



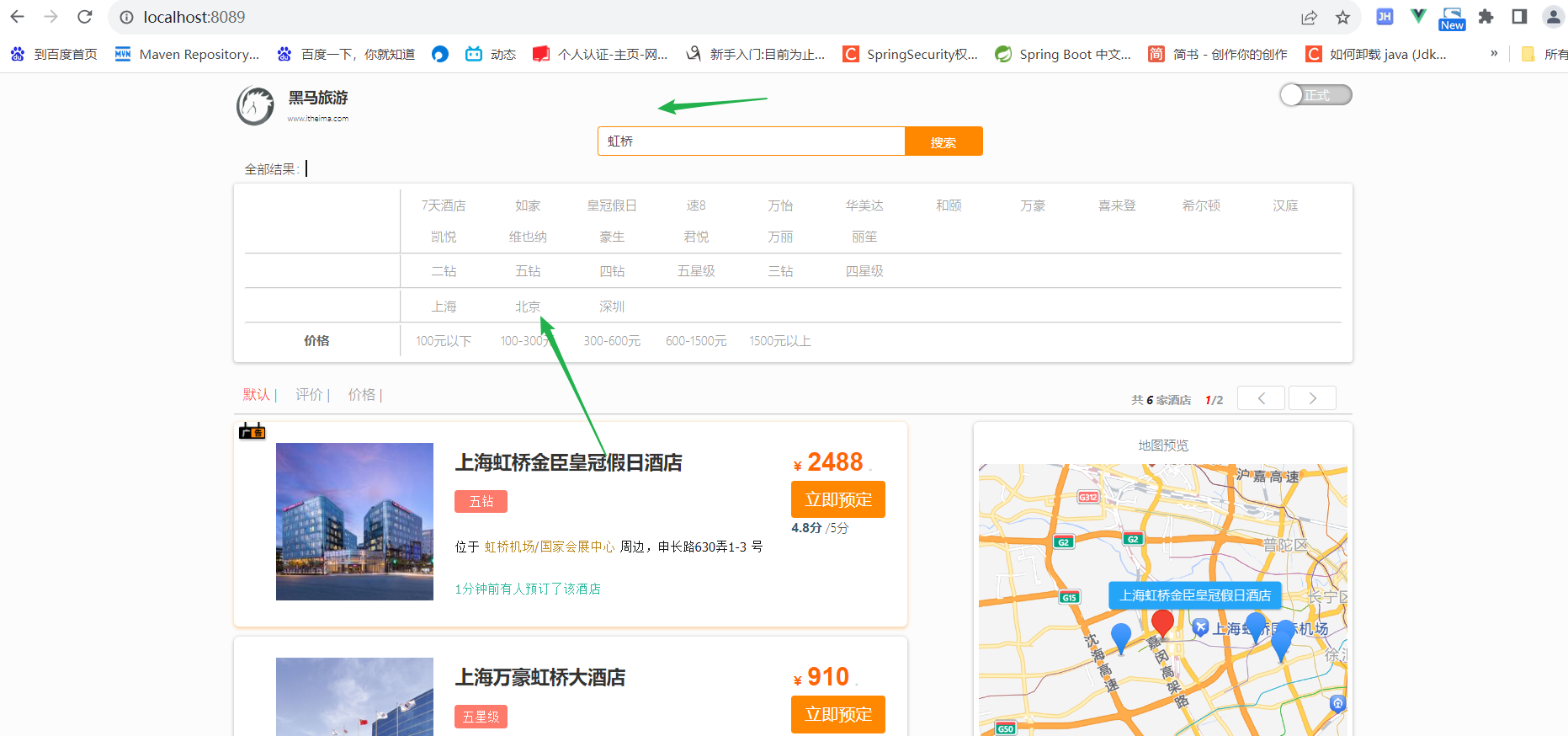
可以看到点击搜索按钮会发起了两个请求，一个是很熟悉的查询酒店数据的请求list，一个就是获取酒店聚合数据的请求filters！可以看到filters也有请求呢！为什么呢？为什么我要在查过滤项的时候也要带条件呢？大家知道过滤项查询将来是要通过聚合来实现的，而聚合一旦加了条件，就是在限定聚合的范围（就是我们之前学习DSL实现聚合的时候的限定范围，通过query实现哦）！

黑马旅游网这里为什么要限定范围呢？我直接对整个hotel索引库做聚合不行吗？还真就不行！

你想一下，如果我们在搜索时没有加任何条件，那我搜到的是索引库的所有数据，这个时候你对索引库的所有数据去做聚合得到城市、品牌、星级是没问题的！但是如果你现在搜索的时候加了条件，比如：虹桥，那我得到的结果一定得是跟上海虹桥火车站有关的数据！这个时候上海虹桥火车站有关的酒店数据对应的城市一定是上海！到那时你却对hotel索引库的所有数据去做聚合，你得到的城市一定包括了所有的城市，这个时候用户不久很奇怪了吗？我搜索的虹桥明明是上海的，怎么还出现北京的城市聚合结果呢？如果这时候你再点击北京，再结合搜索框中的”虹桥“，你想想看还能搜素到任何东西吗！不能啊！肯定不能啊，北京都没有虹桥火车站。这就出事了。

所以说你不应该对索引库中的所有数据去做聚合，这个时候用户条件是虹桥，你就应该对虹桥相关的酒店去做聚合，也就是说要去限定聚合的范围！

那怎么限定聚合范围啊？就是在做聚合的时候加查询条件呗！你在请求list查酒店的时候用的是什么条件，你在请求filters做聚合时就也用什么条件！这样在搜索酒店结果的基础上做聚合了！因此！我们在查询过滤项filters请求时，要加list请求相同的查询条件！



## 限定聚合范围-带过滤项查询

限定聚合范围该怎么写呢？准确来说，你在list请求搜索酒店的时候查询怎么写，这里就怎么写，这样我就可以确保我在做聚合时候的查询条件跟搜索酒店时候的查询条件是完全一致的！

可以看到跟上面相比，只加了两行代码而已！

**@Override**

**public Map<String, List<String>> filters(RequestParams params) {**

**try {**

**// 准备Request**

**SearchRequest request = new SearchRequest("hotel");**

**// 限定聚合范围**

**QueryBuilder boolQuery = buildBasicQuery(params);**

**request.source().query(boolQuery);**

**// 准备DSL**

**request.source().size(0);**

**// 城市聚合**

**request.source().aggregation(**

**AggregationBuilders**

**.terms("cityAgg")**

**.field("city")**

**.size(100)**

**);**

**// 品牌聚合**

**request.source().aggregation(**

**AggregationBuilders**

**.terms("brandAgg")**

**.field("brand")**

**.size(100)**

**);**

**// 品牌聚合**

**request.source().aggregation(**

**AggregationBuilders**

**.terms("starAgg")**

**.field("starName")**

**.size(100)**

**);**

**// 发送请求**

**SearchResponse response = client.search(request, RequestOptions.DEFAULT);**

**// 结果解析**

**Aggregations aggregations = response.getAggregations();**

**Map<String, List<String>> filters = new HashMap<>();**

**// 解析城市聚合**

**Terms cityAgg = aggregations.get("cityAgg");**

**List<? extends Terms.Bucket> cityBuckets = cityAgg.getBuckets();**

**List<String> cites = new ArrayList<>();**

**for (Terms.Bucket bucket : cityBuckets) {**

**String city = bucket.getKeyAsString();**

**cites.add(city);**

**}**

**filters.put("城市", cites);**

**// 解析品牌聚合**

**Terms brandAgg = aggregations.get("brandAgg");**

**List<? extends Terms.Bucket> brandBuckets = brandAgg.getBuckets();**

**List<String> brands = new ArrayList<>();**

**for (Terms.Bucket bucket : brandBuckets) {**

**String brand = bucket.getKeyAsString();**

**brands.add(brand);**

**}**

**filters.put("品牌", brands);**

**// 解析星级聚合**

**Terms starAgg = aggregations.get("starAgg");**

**List<? extends Terms.Bucket> starBuckets = starAgg.getBuckets();**

**List<String> starNames = new ArrayList<>();**

**for (Terms.Bucket bucket : starBuckets) {**

**String starName = bucket.getKeyAsString();**

**starNames.add(starName);**

**}**

**filters.put("星级", starNames);**

**return filters;**

**} catch (Exception e) {**

**throw new RuntimeException(e);**

**}**

**}**