# RestClient实现搜索结果处理

朋友们，之前使用JavaRestClient实现了文档搜索，我们把文档搜索出来了，接下来就轮到对搜索结果处理了：排序、分页、高亮！

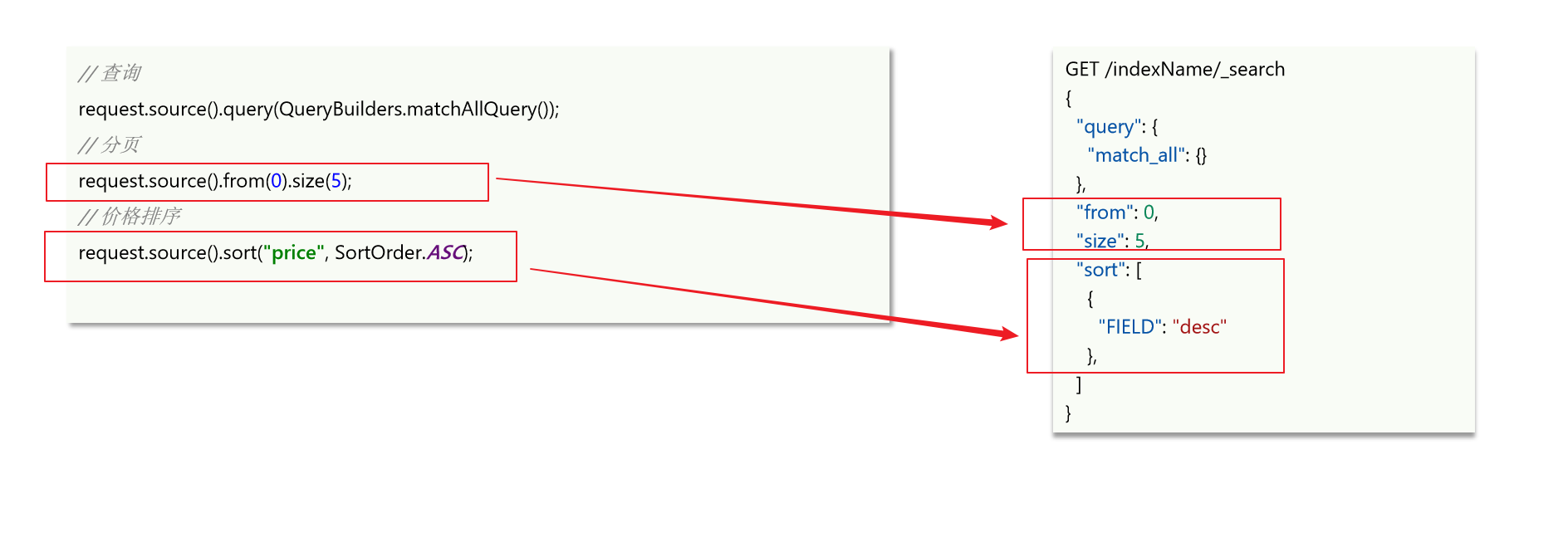
## 排序和分页

一并实现排序和分页！

之前我们使用DSL对搜索结果处理的时候，我们就知道了，分页、排序、查询是同一级别的，如下图：



之前我们学习DSL实现查询的时候，主要就是修改query内部的属性！所以对应JavaRestClient实现查询的java代码，就是在修改request.source().query(QueryBuilders.XXX)内部的参数！那你现在分页的from和size，排序的sort都是和query同级，所以对应JavaRestClient实现分页、排序也就不能在request.source().query()内部去做了！应该和query()方法同级，即应该是对应request.source()，这个source()就是对应DSL语句最外层的{}！

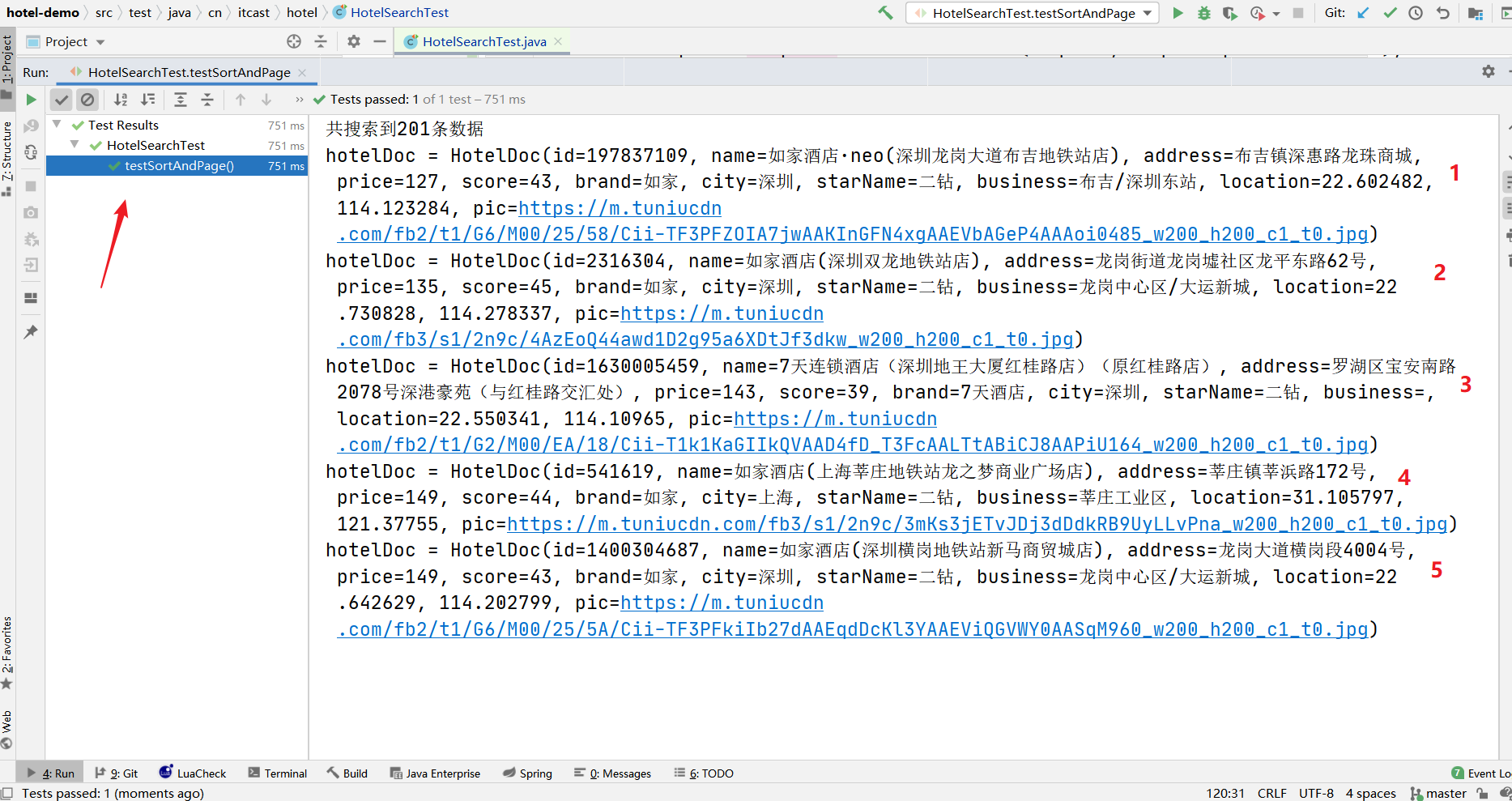


因此，我们无论是做现在的排序、分页，还是做将来的高亮显示等等，都是用这个request.source()来实现的！

### 代码实现



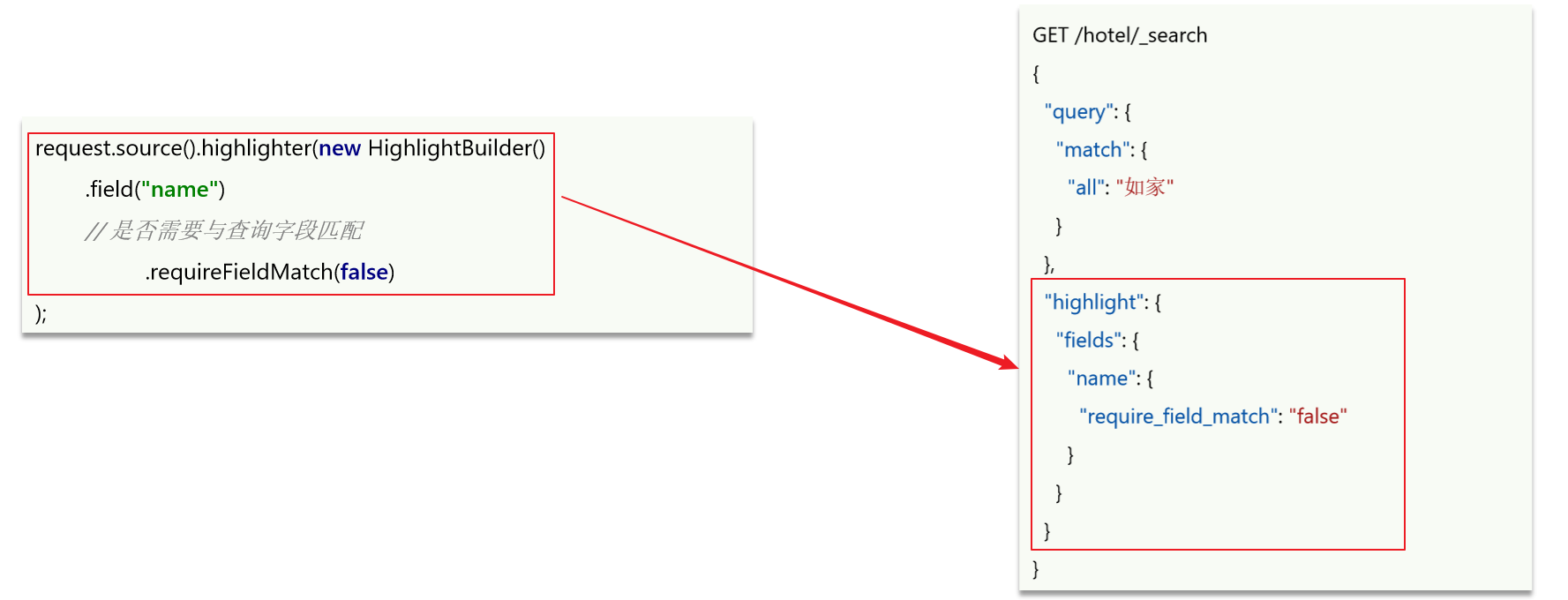
### 运行测试方法，查看搜索结果：



## 高亮

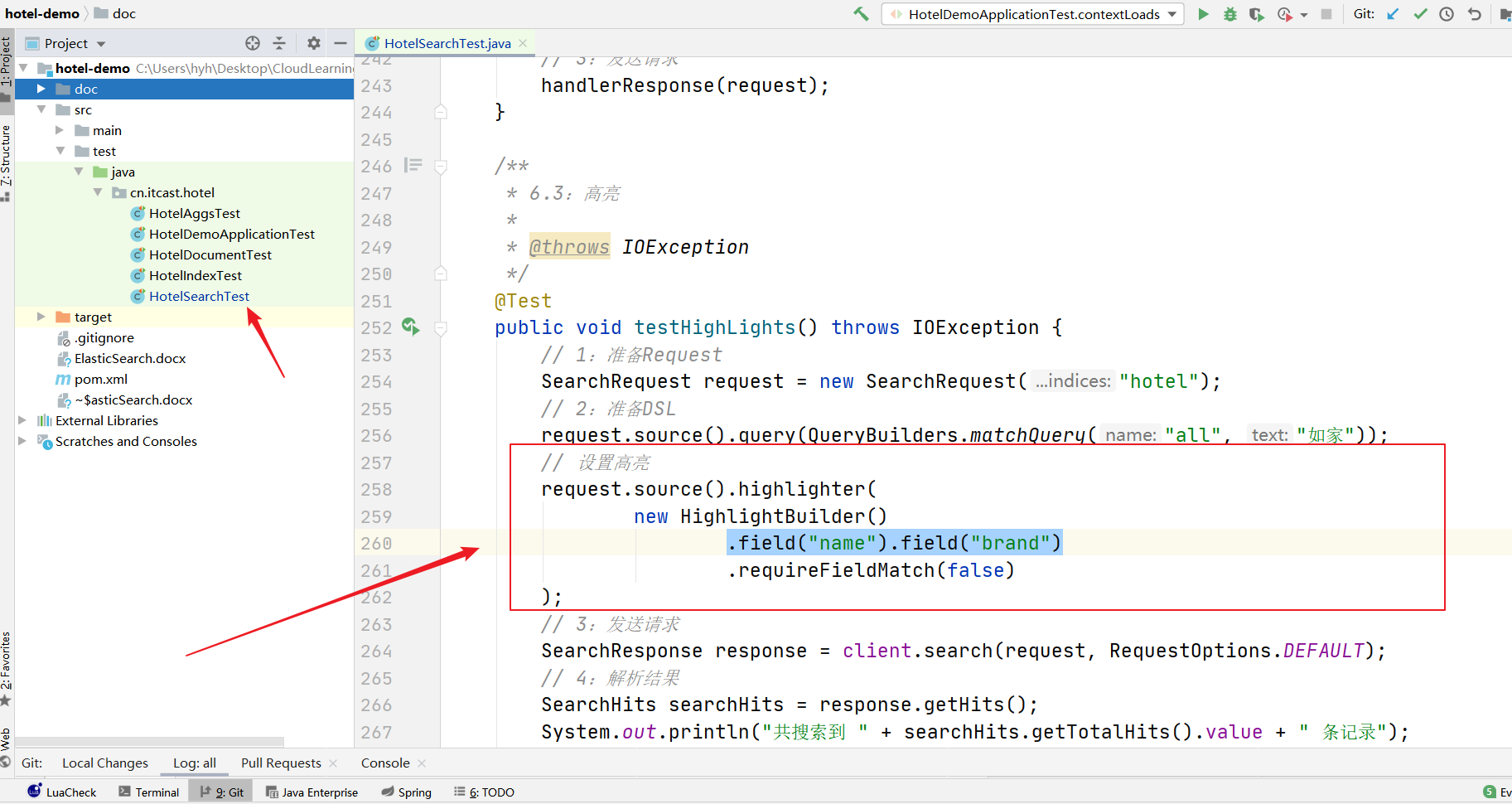
朋友们，上面刚刚学习了JavaRestClient如何实现ES搜索结果的排序、分页。现在来看一下JavaRestClient实现ES搜索结果高亮的API。

高亮API包括**高亮请求DSL构建**和**高亮结果解析**两部分。我们先看请求的DSL构建



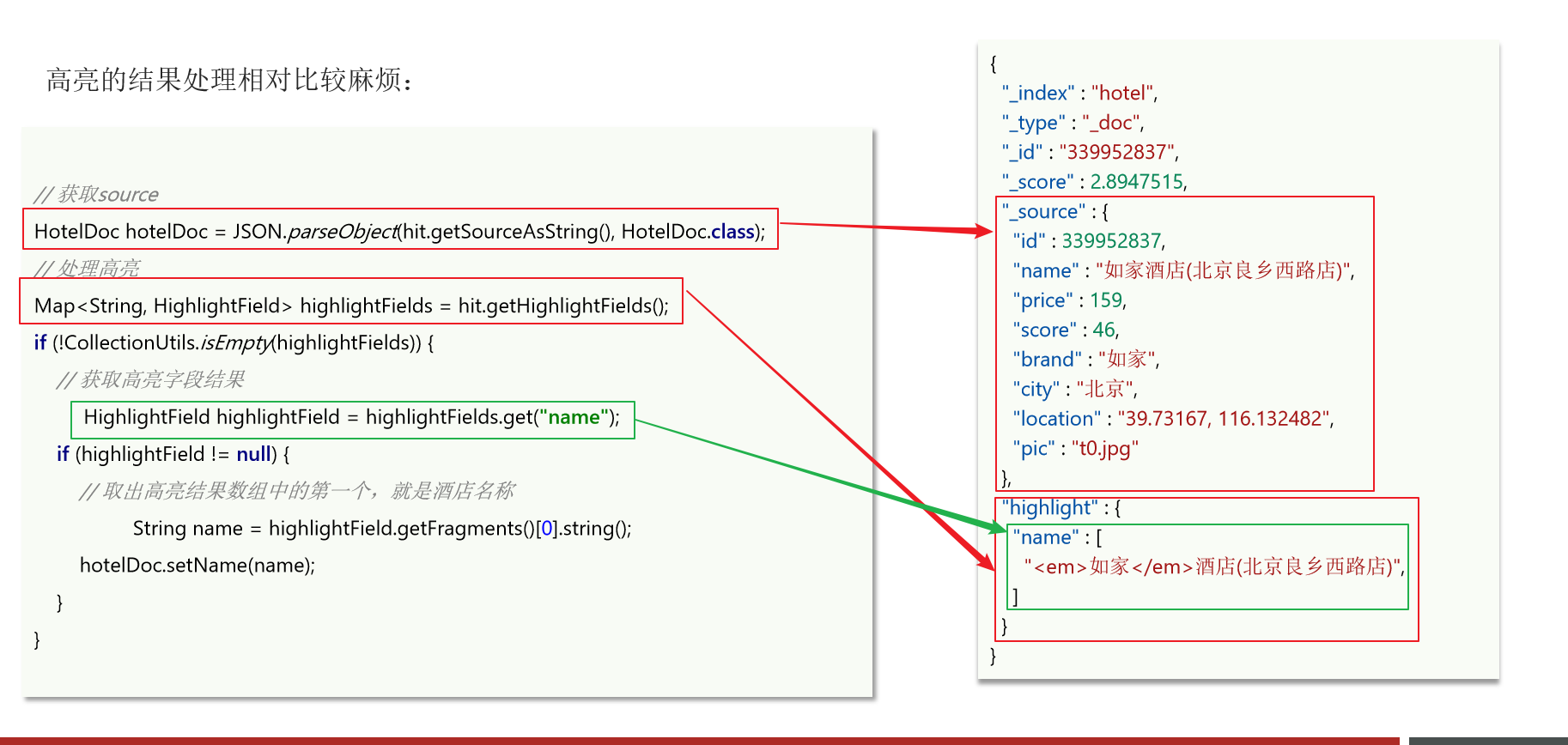
我们发现，DSL语句中高亮highlight同样是和搜索query同级的！所以对应的高亮java代码也是request.source().highlighter()中构建！即通过HighlightBuilder构建！

### 高亮请求代码实现



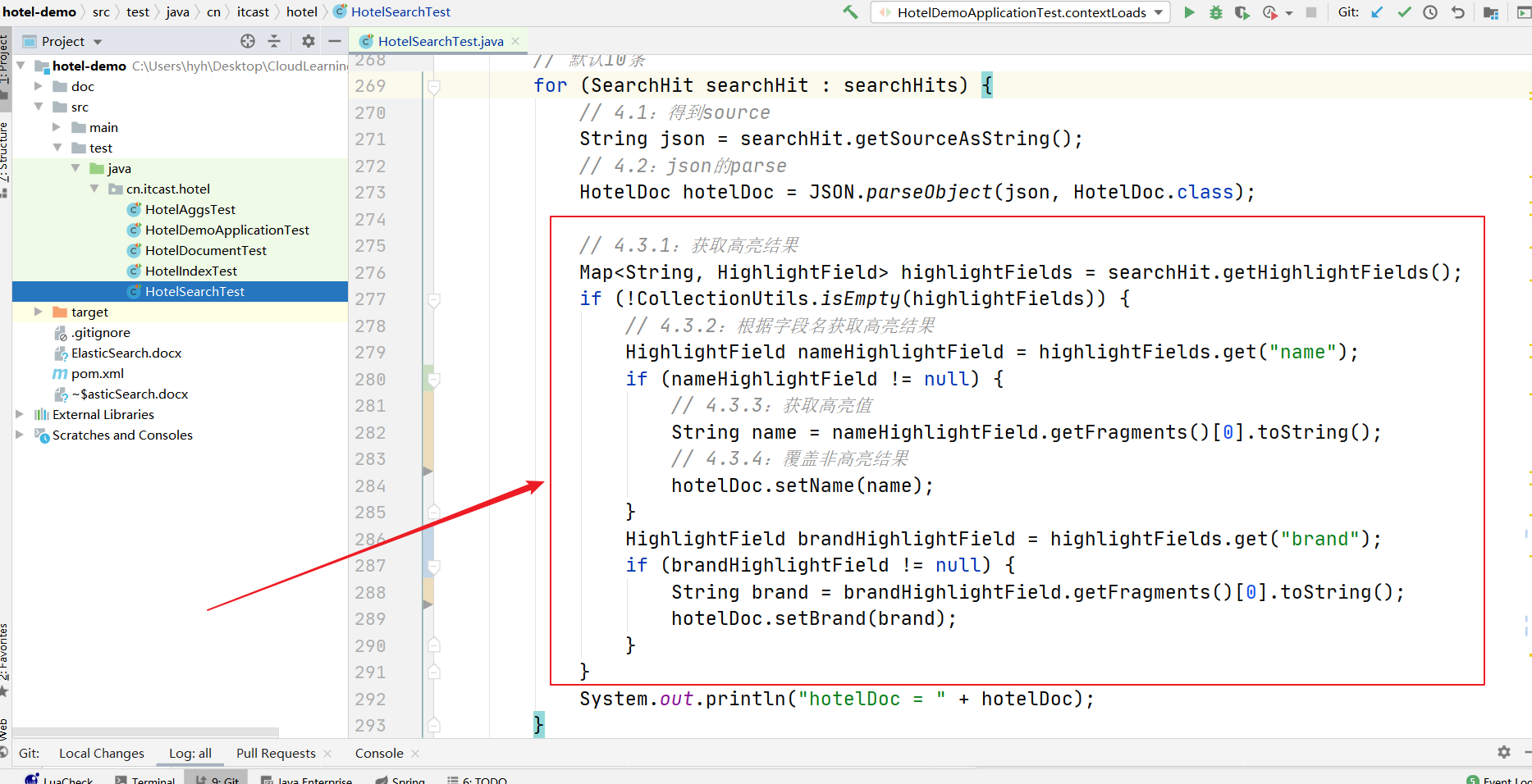
### 高亮结果解析代码实现

上面的高亮请求代码只是在做请求DSL的构建，结果解析部分还没做！

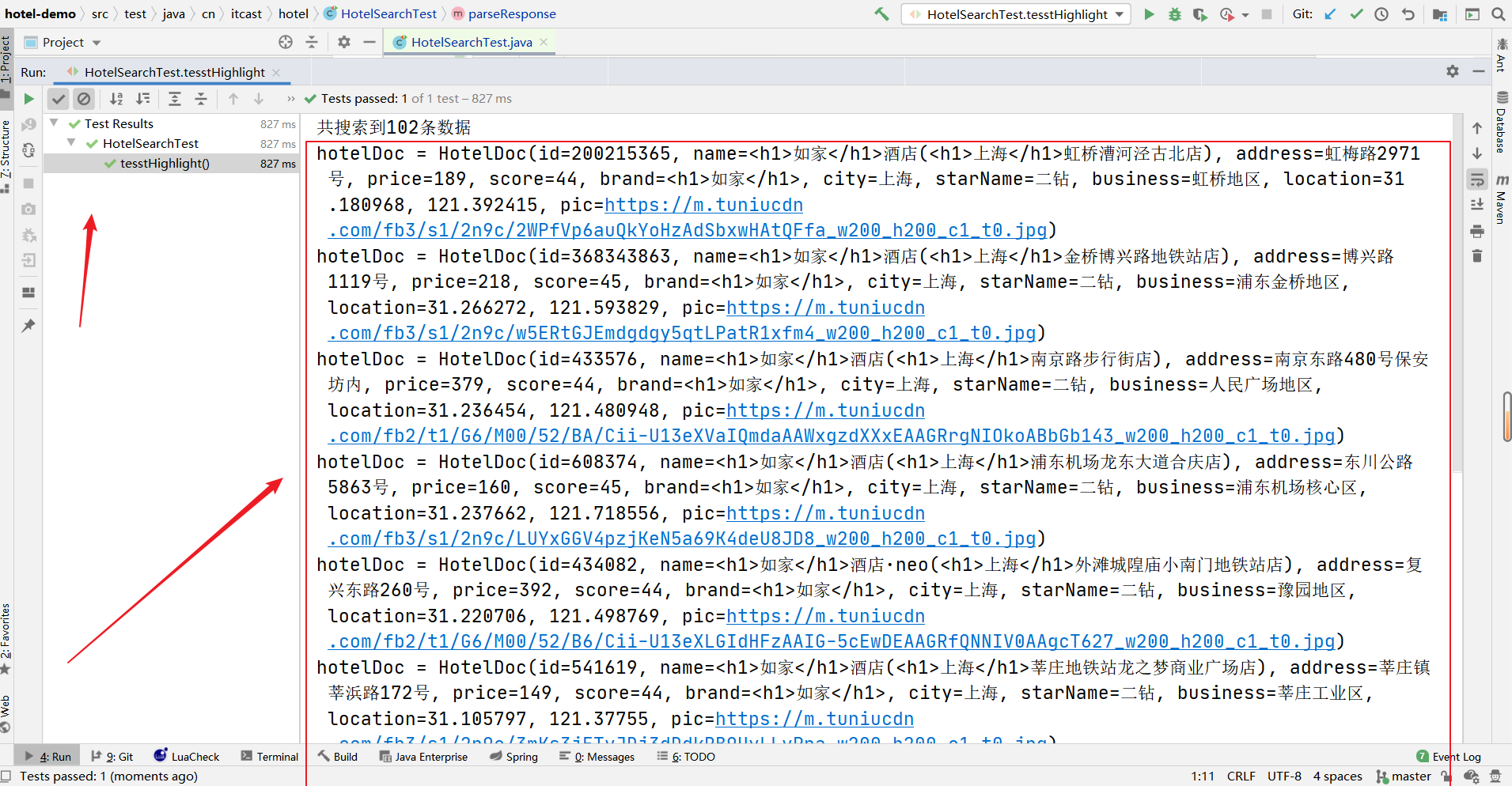


hit.getSourceAsString()获取的是原始文档\_source，是不能被修改的，搜索结果中的高亮部分highlight是与原始文档\_source分离的，可以看到高亮部分highlight与原始文档\_source同级！因此要获取高亮结果就是要获取highlight部分，所以hit.getHighlightFields()就是获取高亮结果的，得到的结果是一个Map，因为highlight是一个JSON对象，对应在Java中可不就是map！

高亮结果解析代码如下：



### 运行测试方法，在控制台发现字段已经被高亮结果替换



### 总结

所有搜索DSL的构建，记住一个API，SearchRequest的source()方法，拿到source，就等于拿到了DSL语句的最大最外面的JSON，就具备了query、form、size、sort、highlight等所有功能。

高亮结果解析是参考JSON结果，逐层解析。