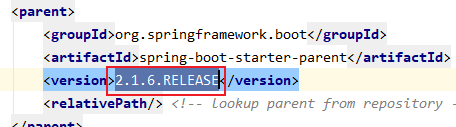
# ----2021-12-21

【花费时间：一个多小时】**applica.yml 与.java文件没有提示**

原因，勾选了省电模式

需要使用springBoot的版本：2.1.6.RELEASE



推荐一个软件(记笔记用)：LibreOffice 7.2

【花费了半个小时】**springBoot测试需要引入的测试类**

|  |
| --- |
| @RunWith(SpringRunner.**class**) |

|  |
| --- |
| <**dependency**>  <**groupId**>org.testng</**groupId**>  <**artifactId**>testng</**artifactId**>  <**version**>RELEASE</**version**>  <**scope**>test</**scope**> </**dependency**> |

|  |
| --- |
| **import** org.junit.Test; |

【已解决】**@Test方法上没有可运行的图标**

因为class前面没有加上public所以检测不到

# ------ 2021-12-22

**index等于false什么意思？**

|  |
| --- |
| @Field(type = FieldType.***Keyword***,index = **false**) **private** String **images**; *// 图片地址* |

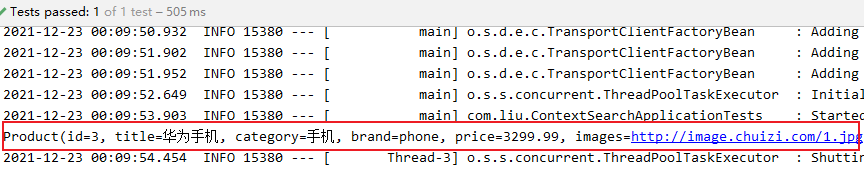
图片的路径不索引？ 不索引，没有索引的价值

|  |
| --- |
| **spring**:  **data**:  **elasticsearch**:  **cluster-name**: lagou-elastic  **cluster-nodes**: 127.0.0.1:9301,127.0.0.1:9302,127.0.0.1:9303 |
| @Autowired **private** ElasticsearchTemplate **template**;  **template**.createIndex(Product.**class**); // 创建索引  **template**.putMapping(Product.**class**); // 指定类型 |
| @Document(indexName = **"lagou01"**,type = **"product"**,shards = 3,replicas = 1)  @Id *//作为document的id*  @Field(type = FieldType.***Text***,analyzer = **"ik\_max\_word"**)  @Field(type = FieldType.***Keyword***,index = **false**) **private** String **images**; *// 图片地址* |

**【花费时间较长-未解决】如何查看保存后的索引数据？数据浏览中没有值？**

|  |
| --- |
| **extends** ElasticsearchRepository<Product,Long>  **productRepository**.save(product);  **productRepository**.saveAll(list); |

使用插件-数据查不到，感觉是索引库存在问题（或者是自己操作的问题）

**查询时数据是存在的**

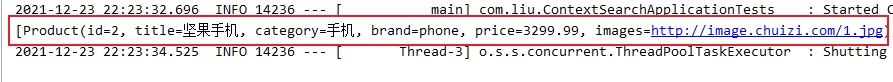
|  |
| --- |
| Optional<Product> optional = **productRepository**.findById(3L); *//取出数据* *//orElse方法的作用：如果optional中封装的实体对象为空也就是没有从索引库中查询出匹配的文档，返回orElse方法的参数* Product defaultResult = **new** Product(); defaultResult.setTitle(**"default data"**); Product product = optional.orElse(defaultResult); *// 查询不到，显示后背方案的值* |
| Iterable<Product> iterable = **productRepository**.findAll(); iterable.forEach(System.***out***::println); |

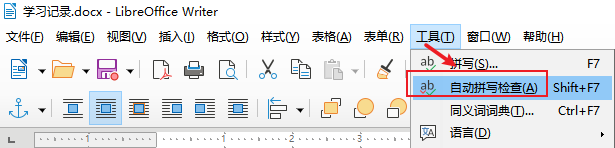
## ----------- 2021-12-23

**如何自定义方法？**

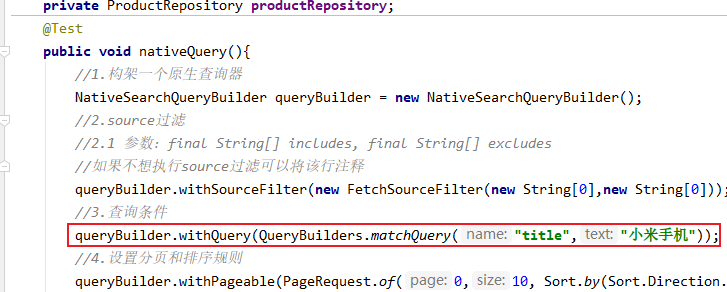
提示是一段一段的，不用担心，没有错误

|  |
| --- |
| *// 查询价格区间的产品数据* List<Product> findByPriceBetween(**double** from,**double** to); |
| List<Product> productList = **productRepository**.findByPriceBetween(0.00, 10000.00); System.***out***.println(productList); |



**libreoffice去除英文红色下划线**

**原生查询-根据条件查询为什么还可以查询出其他的数据？**

【source过滤】：

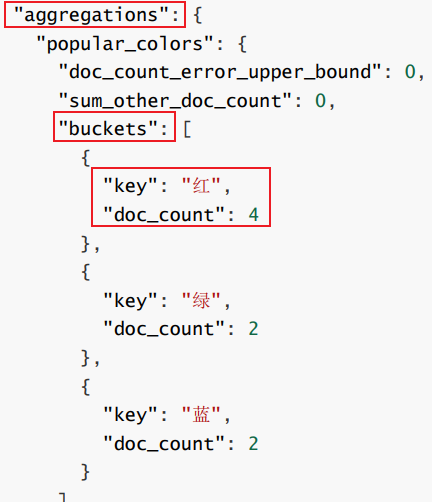
第一个参数：保留某列的值

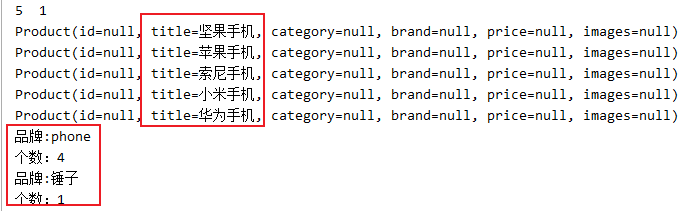
第二个参数：去掉某列的值

【查询条件】：

分词查询（详情看高亮显示）

没有生效



【聚合】

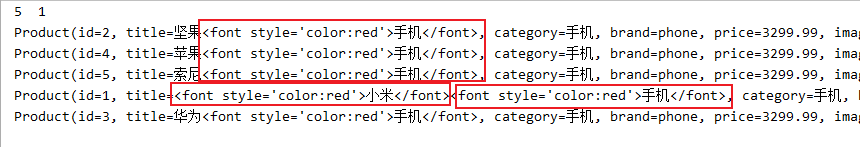
可以理解为分组

聚合中有桶，

桶中有key与doc\_count

|  |
| --- |
| *//1.构架一个原生查询器* NativeSearchQueryBuilder queryBuilder = **new** NativeSearchQueryBuilder(); *//2.source过滤* *//2.1 参数：final String[] includes, final String[] excludes* *//如果不想执行source过滤可以将该行注释* *//queryBuilder.withSourceFilter(new FetchSourceFilter(new String[0],new String[0]));* queryBuilder.withSourceFilter(**new** FetchSourceFilter(**new** String[]{**"title"**},**null**)); *//3.查询条件* queryBuilder.withQuery(QueryBuilders.*matchQuery*(**"title"**,**"小米手机"**)); *//4.设置分页和排序规则* queryBuilder.withPageable(PageRequest.*of*(0,10, Sort.*by*(Sort.Direction.***DESC***,**"price"**))); *//5.高亮 后面演示* *//6.聚合* queryBuilder.addAggregation(AggregationBuilders.*terms*(**"brandAgg"**).field(**"brand"**)); *//7.查询* AggregatedPage<Product> result = **template**.queryForPage(queryBuilder.build(), Product.**class**); *//获取结果* **long** total = result.getTotalElements(); *//页码* **int** totalPages = result.getTotalPages(); *//获得本页的数据集合* List<Product> content = result.getContent(); System.***out***.println(total+**" "**+totalPages); content.stream().forEach(product -> System.***out***.println(product));  *// 获取聚合的结果* Aggregations aggregations = result.getAggregations(); Terms terms = aggregations.get(**"brandAgg"**); *// 获取桶及桶内容* terms.getBuckets().forEach(b->{  System.***out***.println(**"品牌:"**+b.getKeyAsString());  System.***out***.println(**"个数："**+b.getDocCount());  } ); |

**【没有解决】高亮显示原理？**



|  |
| --- |
| **public class** ESSearchResultMapper **implements** SearchResultMapper {  @Override  **public** <T> AggregatedPage<T> mapResults(SearchResponse searchResponse, Class<T> aClass, Pageable pageable) {  *//获得总记录数* **long** totalHits = searchResponse.getHits().getTotalHits();  *//记录列表* List<T> list = **new** ArrayList<>();  *//获取原始的搜索结果* SearchHits hits = searchResponse.getHits();  **if** (hits.getHits().**length** <= 0) {  **return null**;  }  **for** (SearchHit hit : hits) {  *//获取\_source属性中的所有数据*  Map<String, Object> map = hit.getSourceAsMap();  *//获得高亮的字段*  Map<String, HighlightField> highlightFields = hit.getHighlightFields();  *//每个高亮字段都需要进行设置*  **for**(Map.Entry<String,HighlightField> highlightField : highlightFields.entrySet()){  *//获得高亮的key：高亮字段*  String key = highlightField.getKey();  *//获得value：高亮之后的效果*  HighlightField value = highlightField.getValue();  *//将高亮字段和文本效果放入到map中*  map.put(key,value.getFragments()[0].toString());  }  *//将map转换为对象*  Gson gson = **new** Gson();  *//map-->jsonString-->对象* T t = gson.fromJson(gson.toJson(map), aClass);  list.add(t);  }  **return new** AggregatedPageImpl<>(list,pageable,totalHits);  } } |
| <**dependency**>  <**groupId**>com.google.code.gson</**groupId**>  <**artifactId**>gson</**artifactId**>  <**version**>2.8.5</**version**> </**dependency**> |
| *//5.高亮 后面演示* HighlightBuilder.Field field = **new** HighlightBuilder.Field(**"title"**); field.preTags(**"<font style='color:red'>"**); field.postTags(**"</font>"**); queryBuilder.withHighlightFields(field);  *//7.查询* AggregatedPage<Product> result = **template**.queryForPage(queryBuilder.build(), Product.**class**,**new** ESSearchResultMapper()); |

聚合获取失败了，不知道是否是没有做自定义结果映射的原因