

filter 使用语法:

```
1 GET /ecommerce/product/_search
2 {
3   "query": {
4     "bool": {
5       "must": [
6         {"match": {
7           "desc": "shouji"
8         }}
9       ],
10      "filter": {           // filter 仅用于过滤,不会做相关性的比较
11        "range": {         // filter 放在 bool 内,作为筛选条件
12          "price": {
13            "gte": 2500
14          }
15        }
16      }
17    }
18  }
19 }
```

filter 与 query 对比

filter 仅仅只是按照搜索条件过滤出需要的数据而已, 不计算任何相关分数, 对相关度没有任何影响

query 会去计算每个 document 相对于搜索条件的**相关度**, 并且走一些复杂的**算法**, 按照相关度进行排序

一般来说, 如果你的在进行搜索, 需要将最匹配的搜索条件的数据先返回, 那么用 query; 如果你只是根据一些条件筛选出一部分数据, 不关注其排序, 那么用 filter

除非是你的这些搜索条件, 你希望越符合这些搜索条件的document 越排在前面返回, 那么这些搜索条件要放在 query 中; 如果你不希望一些搜索条件来影响你的 document排序, 那么就放在 filter 中即可

filter 与 query 性能 (filter 性能 > query 性能)

filter, 不需要计算相关度分数, 不需要按照相关度分数进行排序, 同时还有内置的自动 cache 最常使用的 filter 的数据
query, 相反, 要计算相关度分数, 按照分数进行排序, 而且无法 cache 结果

总结: 做筛选用 filter 做搜索用 query, filter 的性能要好于 query 的性能