

ElasticSearch

```
curl -XGET "localhost:9200/_all/_mapping?pretty"
```

```
$ curl -XDELETE 'http://localhost:9200/twitter/'
```

```
$ curl -s -XPOST localhost:9200/_bulk --data-binary "@requests"; echo
```

```
GET /test-field/dailywellprd/_search?size=50
```

```
{
  "query": {
    "match": {
      "region": "A"
    }
  }
}
```

```
POST /test-field/region/_search?q=name:B
```

```
POST /test-field/dailywellprd/_search
```

```
{ "query":
  { "bool":
    { "must":
      { "range":
        { "date": { "from": "2015-12-12T12:00:00", "to": "2015-12-25T12:00:00" } }
      }
    }
  }
}
```

```
POST /test-field/dailywellprd/_search?size=1000
```

```
{ "query":
  { "bool":
    { "must": [
      { "range":
        { "date": { "from": "2015-01-01T12:00:00", "to": "2015-12-31T12:00:00" } }
      },
      { "match":
        { "status": "换泵" }
      }
    ]
  }
}
```

```
POST /test-field/dailywellprd/_search?size=10000
```

```
{ "query":
  { "bool":
    { "must": [
      { "range":
        { "date": { "from": "2015-01-01T12:00:00", "to": "2015-12-31T12:00:00" } }
      }
    ]
  },
  "sort": { "oil" : { "order" : "asc" } }
}
```

<https://www.elastic.co/blog/found-beginner-troubleshooting>

```
POST /test-field/dailywellprd/_search?size=100
```

```
{
  "query": {
    "filtered": {
      "filter": { "bool": { "must": [ { "range": { "date": { "from": "2015-07-01T12:00:00", "to": "2015-07-31T12:00:00" } } ] } }
    },
    "query": { "match": {
```

```

        "well": "PA1"
    }
}
}
}

//Get production of PA1 and PA2 in July
POST /test-field/dailywellprd/_search?size=100
{
  "query": {
    "filtered": {
      "filter" : {
        "bool" : {
          "must": { "range":
            { "date":{ "from":"2015-07-01T12:00:00", "to":"2015-07-31T12:00:00" } }
          },
          "should" : [
            { "match" : {"well" : "PA1"}},
            { "match" : {"well" : "PA2"}}
          ]
        }
      }
    }
  }
}
}

```

Should: Or
 Must: And
 Must_Not: Not

Child/Parent: doc can update independently, lower performance and higher memory cost
 Nest: can't update independently. Higher performance.

```
bin/plugin install elasticsearch/elasticsearch-mapper-attachments/3.1.1
```

ElasticSearch Aggregation - 聚合, only on not analyzed field.

PUT test-field

```

{
  "mappings": {
    "well": {
      "properties": {
        "name": {
          "type": "string",
          "fields": {
            "raw": { "type": "string", "index": "not_analyzed" }}
        },
        "region": {
          "type": "string",
          "fields": {
            "raw": { "type": "string", "index": "not_analyzed" }}
        },
        "type": {
          "type": "string",
          "fields": {
            "raw": { "type": "string", "index": "not_analyzed" }}
        },
        "prdlayer": {

```



```
      "field": "oil"
    }
  },
  "water": {
    "sum": {
      "field": "water"
    }
  },
  "gas": {
    "sum": {
      "field": "gas"
    }
  }
}
}
```

GET /test-field/well/_search

```
{
  "query": {
    "has_parent": {
      "type": "region",
      "query": {
        "match": {
          "name": "C"
        }
      }
    }
  }
}
```

GET /test-field/dailywellprd/_search

```
{
  "size": 0,
  "query": {
    "has_parent": {
      "type": "well",
      "query": {
        "has_parent": {
          "type": "region",
          "query": {
            "match": {
              "name": "C"
            }
          }
        }
      }
    }
  }
}
```

```

    }
  }
}
},
"aggs": {
  "dailyproduction": {
    "date_histogram": {
      "field": "date",
      "interval": "1d",
      "time_zone": "Asia/Shanghai",
      "min_doc_count": 1
    },
    "aggs": {
      "oil": {
        "sum": {
          "field": "oil"
        }
      },
      "water": {
        "sum": {
          "field": "water"
        }
      },
      "gas": {
        "sum": {
          "field": "gas"
        }
      }
    }
  }
}
}

//more accurate filter
GET /test-field/dailywellprd/_search
{
  "size": 0,
  "query": {
    "has_parent": {
      "type": "well",
      "query": {
        "has_parent": {
          "type": "region",
          "query": {
            "filtered": {

```

```

        "filter": {
            "term": { "name.raw": "C" }
        }
    }
}

},
"aggs": {
    "dailyproduction": {
        "date_histogram": {
            "field": "date",
            "interval": "1d",
            "time_zone": "Asia/Shanghai",
            "min_doc_count": 1
        },
        "aggs": {
            "oil": {
                "sum": {
                    "field": "oil"
                }
            },
            "water": {
                "sum": {
                    "field": "water"
                }
            },
            "gas": {
                "sum": {
                    "field": "gas"
                }
            }
        }
    }
}

//Get production agrretation by date from both region A and region C
GET /test-field/dailywellprd/_search
{
    "size": 0,
    "query": {
        "has_parent": {
            "type": "well",
            "query": {

```

```
    "has_parent": {
      "type": "region",
      "query": {
        "filtered": {
          "filter": {
            "terms": { "name.raw": ["A","C"] }
          }
        }
      }
    }
  },
}
```

```
"aggs": {
  "dailyproduction": {
    "date_histogram": {
      "field": "date",
      "interval": "1d",
      "time_zone": "Asia/Shanghai",
      "min_doc_count": 1
    },

```

```
    "aggs": {
      "oil": {
        "sum": {
          "field": "oil"
        }
      },
      "water": {
        "sum": {
          "field": "water"
        }
      },
      "gas": {
        "sum": {
          "field": "gas"
        }
      }
    }
  }
}
```

```
}
}
```

//with date

POST /test-field/dailywellprd/_search

```
{
  "size": 0,
```

```
"query": {
  "filtered": {
    "query": {
      "has_parent": {
        "type": "well",
        "query": {
          "has_parent": {
            "type": "region",
            "query": {
              "match": {
                "name": "A"
              }
            }
          }
        }
      }
    }
  },
  "filter": {
    "bool": {
      "must": [
        {
          "range": {
            "date": {
              "gte": "2015-07-31",
              "lte": "2015-08-06"
            }
          }
        }
      ]
    }
  }
}
```

```
},
"aggs": {
  "2": {
    "date_histogram": {
      "field": "date",
      "interval": "1d",
      "time_zone": "Asia/Shanghai",
      "min_doc_count": 1
    },
    "aggs": {
      "oil": {
        "sum": {
```



```

        "field": "oil"
      }
    },
    "water": {
      "sum": {
        "field": "water"
      }
    },
    "gas": {
      "sum": {
        "field": "gas"
      }
    }
  }
}

//Query wells with production >??
POST /test-field/_search?
{
  "query": {
    "filtered": {
      "query": {
        "has_child": {
          "type": "dailywellprd",
          "query": {
            "filtered": {
              "filter": {
                "bool": {
                  "must": [
                    {"range": {
                      "oil": {
                        "from": 100,
                        "to": 20000
                      }
                    }
                  ]
                }
              },
              {"range": {
                "date": {
                  "gte": "2015-07-01",
                  "lte": "2015-08-31"
                }
              }
            }
          ]
        }
      }
    }
  }
}

```



```

    }
  }
}

  ]]
}

}

}

}

}

}

```

中文

```
sudo bin/plugin install analysis-smartcn
```

```
GET /_analyze
{
  "analyzer" : "smartcn",
  "text" : "松辽盆地南部王府断陷山东屯构造带局部构造高点城深102井"
}
```

```
sudo bin/plugin install mapper-attachments
```

```
PUT /docx
{
  "index" : {
    "analysis" : {
      "analyzer" : {
        "default" : {
          "type" : "smartcn"
        }
      }
    }
  }
}
```

```
POST /docx
{
  "index" : {
    "analysis" : {
      "analyzer" : {
        "default" : {
          "type" : "smartcn"
        }
      }
    }
  },
  "mappings": {
    "word": {
      "properties": {
        "content": { "type": "attachment" }
      }
    }
  }
}
```

```
curl XPOST http://localhost:9200/docx/word/read_doc?pretty=1 -d '{"content": "'`base64 -w 0 1.doc | perl -pe 's/\n/\\n/g'`"}'
```

```
base64 -w 0 1.doc | perl -pe 's/\n/\\n/g' >1.txt, turn 1.txt to json object
```

```
curl PUT 'http://localhost:9200/docx/word/1?pretty=1' -d @1.txt
```

```
POST /docx/word/_search
{
  "query": {
    "query_string": {
      "query": "你好"
    }
  }
}
```

