

# 聚合

我可以利用以下几种不同的聚合和它的语法，但学习聚合的最佳途径就是用例子来阐明。一旦我得到了聚合的思想，以及如何合理地嵌套使用它，那语法就得不那么重要了。

**NOTE** 聚合的桶操作和度量的完整用法可以在[{ref}/search-aggregations.html](#)[Elasticsearch 参考]中找到。本章中会涵盖其中很多内容，但在读完本章后看它会有助于它的整体能力有所了解。

所以我先看一个例子。我将会构建一些汽车商有用的聚合，数据是关于汽车交易的信息：车型、制造商、何地被出售等。

首先我批量索引一些数据：

```
POST /cars/transactions/_bulk
{ "index": {} }
{ "price" : 10000, "color" : "red", "make" : "honda", "sold" : "2014-10-28" }
{ "index": {} }
{ "price" : 20000, "color" : "red", "make" : "honda", "sold" : "2014-11-05" }
{ "index": {} }
{ "price" : 30000, "color" : "green", "make" : "ford", "sold" : "2014-05-18" }
{ "index": {} }
{ "price" : 15000, "color" : "blue", "make" : "toyota", "sold" : "2014-07-02" }
{ "index": {} }
{ "price" : 12000, "color" : "green", "make" : "toyota", "sold" : "2014-08-19" }
{ "index": {} }
{ "price" : 20000, "color" : "red", "make" : "honda", "sold" : "2014-11-05" }
{ "index": {} }
{ "price" : 80000, "color" : "red", "make" : "bmw", "sold" : "2014-01-01" }
{ "index": {} }
{ "price" : 25000, "color" : "blue", "make" : "ford", "sold" : "2014-02-12" }
```

有了数据，开始构建我的第一个聚合。汽车商可能会想知道哪个颜色的汽车销量最好，用聚合可以轻易得到结果，用 **terms** 桶操作：

```
GET /cars/transactions/_search
{
  "size" : 0,
  "aggs" : { ①
    "popular_colors" : { ②
      "terms" : { ③
        "field" : "color"
      }
    }
  }
}
```

① 聚合操作被置于参数 **aggs** 之下（如果愿意，完整形式 **aggregations** 同样有效）。

② 然后，可以聚合指定一个我想要名称，本例中是：`popular_colors`。

③ 最后，定义一个桶的类型 `terms`。

聚合是在特定搜索结果背景下行的，也就是它只是请求的另外一个参数（例如，使用 `/_search` 端点）。聚合可以与 `size` 一起使用，但我会在 [限定聚合的 \(Scoping Aggregations\)](#) 中来解决这个问题。

#### NOTE

可能会注意到我将 `size` 置成 0。我并不是关心搜索结果的具体内容，所以将返回数置 0 来提高速度。置 `size: 0` 与 Elasticsearch 1.x 中使用 `count` 搜索类型等。

然后我聚合定义一个名字，名字取决于使用者，聚合的结果会以我定义的名字来使用，这样就可以解析得到的结果。

随后我定义聚合本身，在本例中，我定义了一个 `terms` 桶。一个 `terms` 桶会返回到的唯一新建的桶。因为我告诉它使用 `color` 字段，所以 `terms` 桶会返回一个颜色新建桶。

我运行聚合并看结果：

```
{
  ...
  "hits": {
    "hits": [] ①
  },
  "aggregations": {
    "popular_colors": { ②
      "buckets": [
        {
          "key": "red", ③
          "doc_count": 4 ④
        },
        {
          "key": "blue",
          "doc_count": 2
        },
        {
          "key": "green",
          "doc_count": 2
        }
      ]
    }
  }
}
```

① 因为我置了 `size` 参数，所以不会有 hits 搜索结果返回。

② `popular_colors` 聚合是作为 `aggregations` 字段的一部分被返回的。

③ 一个桶的 `key` 都与 `color` 字段里到的唯一值。它会包含 `doc_count` 字段，告诉我包含

的文 数量。

④ 个桶的数量代表 色的文 数量。

包含多个桶， 个 一个唯一 色（例如： 或 ）。 个桶也包括 聚合 桶的所有文 的数量。例如，有四 色的 。

前面的 个例子完全是 行的：一旦文 可以被搜到，它就能被聚合。 也就意味着我 可以直接将聚合的 果源源不断的 入 形 ，然后生成 的 表 。 不久， 又 了一 色的 ，我 的 形就会立即 更新 色 的 信息。

！ 就是我 的第一个聚合！

## 添加度量指

前面的例子告 我 个桶里面的文 数量， 很有用。但通常，我 的用需要提供更 的文 度量。例如， 色汽 的平均 格是多少？

了 取更多信息，我 需要告 Elasticsearch 使用 个字段， 算何 度量。 需要将度量 嵌套在桶内，度量会基于桶内的文 算 果。

我 汽 的例子加入 **average** 平均度量：

```
GET /cars/transactions/_search
{
  "size" : 0,
  "aggs": {
    "colors": {
      "terms": {
        "field": "color"
      },
      "aggs": { ①
        "avg_price": { ②
          "avg": {
            "field": "price" ③
          }
        }
      }
    }
  }
}
```

① 度量新 **aggs** 。

② 度量指定名字：**avg\_price**。

③ 最后， **price** 字段定 **avg** 度量。

正如所 ，我 用前面的例子加入了新的 **aggs** 。 个新的聚合 我 可以将 **avg** 度量嵌套置于 **terms** 桶内。 上， 就 个 色生成了平均 格。

正如 **色** 的例子，我 需要 度量起一个名字（ **avg\_price** ） 可以 后根据名字 取它的。最后，我 指定度量本身（ **avg** ）以及我 想要 算平均 的字段（ **price** ）：

```
{
  ...
  "aggregations": {
    "colors": {
      "buckets": [
        {
          "key": "red",
          "doc_count": 4,
          "avg_price": { ①
            "value": 32500
          }
        },
        {
          "key": "blue",
          "doc_count": 2,
          "avg_price": {
            "value": 20000
          }
        },
        {
          "key": "green",
          "doc_count": 2,
          "avg_price": {
            "value": 21000
          }
        }
      ]
    }
  }
  ...
}
```

① 中的新字段 **avg\_price**。

尽管 只 生很小改 ， 上我 得的数据是 了。之前，我 知道有四 色的 ， 在， 色 的平均 格是 \$32, 500 美元。 个信息可以直接 示在 表或者 形中。

## 嵌套桶

在我 使用不同的嵌套方案 ， 聚合的力量才能真正得以 。 在前例中，我 以及看到如何将一个度量嵌入桶中，它的功能已 十分 大了。

但真正令人激 的分析来自于将桶嵌套 外一个桶 所能得到的 果。 在，我 想知道 个 色的汽 制造商的分布：

```
GET /cars/transactions/_search
{
  "size" : 0,
  "aggs": {
    "colors": {
      "terms": {
        "field": "color"
      },
      "aggs": {
        "avg_price": { ①
          "avg": {
            "field": "price"
          }
        },
        "make": { ②
          "terms": {
            "field": "make" ③
          }
        }
      }
    }
  }
}
```

- ① 注意前例中的 `avg_price` 度量 然保持原位。
- ② 一个聚合 `make` 被加入到了 `color` 色桶中。
- ③ 一个聚合是 `terms` 桶，它会 一个汽 制造商生成唯一的桶。

里 生了一些有趣的事。 首先，我 可能会 察到之前例子中的 `avg_price` 度量完全没有 化，在原来的位置。 一个聚合的 个 都可以有多个度量或桶， `avg_price` 度量告 我 色汽的平均 格。它与其他桶和度量相互独立。

我 的 用非常重要，因 里面有很多相互 ，但又完全不同的度量需要收集。聚合使我 能 用一次数据 求 得所有的 些信息。

外一件 得注意的重要事情是我 新 的 个 `make` 聚合，它是一个 `terms` 桶（嵌套在 `colors` 、 `terms` 桶内）。 意味着它会 数据集中的 个唯一 合生成（`color` 、 `make`）元 。

我 看看返回的 （了 我 只 示部分 果）：

```
{
  ...
  "aggregations": {
    "colors": {
      "buckets": [
        {
          "key": "red",
          "doc_count": 4,
          "make": { ①
            "buckets": [
              {
                "key": "honda", ②
                "doc_count": 3
              },
              {
                "key": "bmw",
                "doc_count": 1
              }
            ]
          },
          "avg_price": {
            "value": 32500 ③
          }
        },
        ...
      ]
    }
  }
}
```

- ① 正如期望的那样，新的聚合嵌入在一个颜色桶中。
- ② 在我看按不同制造商分解的颜色下信息。
- ③ 最后，我看到前例中的 `avg_price` 度量虽然持平。

报告我以下几点：

- 颜色有四。
- 颜色的平均是 \$32,500 美元。
- 其中三 是 Honda 本田制造，一 是 BMW 宝马制造。

## 最后的修改

我回到原点，在入新之前，我的示例做最后一个修改，

个汽车生成商

算最低和最高的价格：

```

GET /cars/transactions/_search
{
  "size" : 0,
  "aggs": {
    "colors": {
      "terms": {
        "field": "color"
      },
      "aggs": {
        "avg_price": { "avg": { "field": "price" } },
      },
      "make" : {
        "terms" : {
          "field" : "make"
        },
        "aggs" : { ①
          "min_price" : { "min": { "field": "price" } }, ②
          "max_price" : { "max": { "field": "price" } } ③
        }
      }
    }
  }
}

```

① 我 需要 加 外一个嵌套的 **aggs** 。

② 然后包括 **min** 最小度量。

③ 以及 **max** 最大度量。

得到以下 出（只 示部分 果）：

```

{
  ...
  "aggregations": {
    "colors": {
      "buckets": [
        {
          "key": "red",
          "doc_count": 4,
          "make": {
            "buckets": [
              {
                "key": "honda",
                "doc_count": 3,
                "min_price": {
                  "value": 10000 ①
                },
                "max_price": {
                  "value": 20000 ①
                }
              },
              {
                "key": "bmw",
                "doc_count": 1,
                "min_price": {
                  "value": 80000
                },
                "max_price": {
                  "value": 80000
                }
              }
            ]
          },
          "avg_price": {
            "value": 32500
          }
        },
        ...
      ]
    }
  }
}

```

① min 和 max 度量 在出 在 个汽 制造商 ( make ) 下面。

有了 个桶, 我 可以 的 果 行 展并得到以下信息 :

- 有四 色 。
- 色 的平均 是 \$32, 500 美元。
- 其中三 色 是 Honda 本田制造, 一 是 BMW 宝 制造。
- 最便宜的 色本田 \$10, 000 美元。
- 最 的 色本田 \$20, 000 美元。