elasticsearch

加入尚学堂，一起进步！

- CURL命令
  - 简单认为是可以在命令行下访问url的一个工具
  - curl是利用URL语法在命令行方式下工作的开源文件传输工具，使用curl可以简单实现常见的get/post请求。
  - curl
  - -x 指定http请求的方法
  - HEAD GET POST PUT DELETE
  - -d 指定要传输的数据

- CURL建立索引库
- curl -XPUT 'http://localhost:9200/index_name/'
- PUT/POST都可以

- CURL创建索引
- #创建索引

```
curl -XPOST http://localhost:9200/bjsxt/employee/1 -d
    '{
"first_name" : "John",
"last_name" : "Smith",
"age" : 25,
"about" : "I love to go rock climbing",
"interests": [ "sports", "music" ]
}'
```

- PUT 和POST用法
- PUT是幂等方法，POST不是。所以PUT用于更新、POST用于新增比较合适。
  - PUT，DELETE操作是幂等的。所谓幂等是指不管进行多少次操作，结果都一样。比如我用PUT修改一篇文章，然后在做同样的操作，每次操作后的结果并没有不同，DELETE也是一样。
  - POST操作不是幂等的，比如常见的POST重复加载问题：当我们多次发出同样的POST请求后，其结果是创建出了若干的资源。
  - 还有一点需要注意的就是，创建操作可以使用POST，也可以使用PUT，区别在于POST是作用在一个集合资源之上的（/articles），而PUT操作是作用在一个具体资源之上的（/articles/123），比如说很多资源使用数据库自增主键作为标识信息，而创建的资源的标识信息到底是什么只能由服务端提供，这个时候就必须使用POST。

- 创建索引注意事项

- **索引库名称必须要全部小写，不能以下划线开头，也不能包含逗号**
- 如果没有明确指定索引数据的ID，那么es会自动生成一个随机的ID,需要使用POST参数

  curl -XPOST http://localhost:9200/bjsxt/emp/ -d '{"first_name" : "John"}'
- 如果想要确定我们创建的都是全新的内容
  - 1：使用自增ID
  - 2：在url后面添加参数
    - curl -XPUT http://localhost:9200/bjsxt/emp/2?op_type=create -d '{"name": “zs","age":25}'
    - curl -XPUT http://localhost:9200/bjsxt/emp/2/_create -d '{"name": “laoxiao","age":25}'
  - 如果成功创建了新的文档，Elasticsearch将会返回常见的元数据以及201 Created的HTTP反馈码。而如果存在同名文件，Elasticsearch将会返回一个409 Conflict的HTTP反馈码

- GET查询索引
  - 根据员工id查询
    - curl -XGET http://localhost:9200/bjsxt/employee/1?pretty
  - 在任意的查询字符串中添加pretty参数，es可以得到易于识别的json结果。
  - curl后添加-i 参数，这样你就能得到反馈头文件
    - curl -i 'http://192.168.1.170:9200/bjsxt/emp/1?pretty'

- GET查询索引
  - 检索文档中的一部分，如果只需要显示指定字段，
    - curl -XGET http://localhost:9200/bjsxt/employee/1?_source=name,age
  - 如果只需要source的数据
    - curl –XGET http://localhost:9200/bjsxt/employee/1/_source
  - 查询所有
    - curl -XGET http://localhost:9200/bjsxt/employee/_search
  - 你可以再返回的 hits 中发现我们录入的文档。搜索会默认返回最前的10个数值。
  - 根据条件进行查询
    - curl -XGET http://localhost:9200/elasticsearch/employee/_search?q=last_name:Smith

- DSL查询
- Domain Specific Language
  - 领域特定语言
- curl -XGET http://localhost:9200/bjsxt/employee/_search -d
  - '{"query":
    - {"match":
      - {"last_name":"Smith"}
    - }
  - }'

- MGET查询
- 使用mget API获取多个文档
  - curl -XGET http://localhost:9200/_mget?pretty -d '{"docs":[{"_index": “bjsxt","_type":"emp","_id":2,"_source":"name"},{"_index":"website","_type":"blog","_id":2}]}'
- 如果你需要的文档在同一个_index或者同一个_type中，你就可以在URL中指定一个默认的/_index或者/_index/_type
  - curl -XGET http://localhost:9200/bjsxt/emp/_mget?pretty -d '{"docs":[{"_id":1},{"_type":"blog","_id":2}]}'
- 如果所有的文档拥有相同的_index 以及 _type，直接在请求中添加ids的数组即可
  - curl -XGET http://localhost:9200/bjsxt/emp/_mget?pretty -d '{"ids":["1","2"]}'
- 注意：如果请求的某一个文档不存在，不会影响其他文档的获取结果。HTTP返回状态码依然是200，这是因为mget这个请求本身已经成功完成。要确定独立的文档是否被成功找到，需要检查found标识

- HEAD使用
- 如果只想检查一下文档是否存在，你可以使用HEAD来替代GET方法，这样就只会返回HTTP头文件
  - curl -i -XHEAD http://localhost:9200/bjsxt/emp/1

- Elasticsearch的更新
- ES可以使用PUT或者POST对文档进行更新，如果指定ID的文档已经存在，则执行更新操作
- 注意:执行更新操作的时候
  - ES首先将旧的文档标记为删除状态
  - 然后添加新的文档
  - 旧的文档不会立即消失，但是你也无法访问
  - ES会在你继续添加更多数据的时候在后台清理已经标记为删除状态的文档

- Elasticsearch的更新
- 局部更新，可以添加新字段或者更新已有字段（必须使用POST）
  - curl -XPOST http://localhost:9200/bjsxt/emp/1/_update -d '{"doc":{"city":"beijing","car":"BMW"}}'

- Elasticsearch的删除
- curl -XDELETE http://localhost:9200/bjsxt/emp/4/
- 如果文档存在，es会返回200 ok的状态码，found属性值为true，_version属性的值+1
- found属性值为false，但是_version属性的值依然会+1，这个就是内部管理的一部分，它保证了我们在多个节点间的不同操作的顺序都被正确标记了
- 注意：删除一个文档也不会立即生效，它只是被标记成已删除。Elasticsearch将会在你之后添加更多索引的时候才会在后台进行删除内容的清理

- Elasticsearch的删除
- 通过查询API删除指定索引库下指定类型下的数据

```
curl -XDELETE 'http://localhost:9200/bjsxt/emp/_query?q=user:kimchy'

curl -XDELETE 'http://localhost:9200/bjsxt/emp/_query' -d
'{
    "query" : {
        "term" : { "user" : "kimchy" }
    }
}'
```

- Elasticsearch的删除
- 通过查询API删除指定索引库下多种类型下的数据

curl -XDELETE 'http://localhost:9200/bjsxt/emp,user/_query?q=user:kimchy ‘

- 通过查询API删除多个索引库下多种类型下的数据

curl -XDELETE 'http://localhost:9200/bjsxt,index/emp,user/_query?q=user:kimchy'

或者删除所有索引库中的匹配的数据

curl -XDELETE 'http://localhost:9200/_all/_query?q=tag:wow'

- Elasticsearch的批量操作bulk
- 与mget类似，bulk API可以帮助我们同时执行多个请求
- 格式：
  - action：index/create/update/delete
  - metadata：_index,_type,_id
  - request body：_source(删除操作不需要)

```
{ action: { metadata }}\n
{ request body        }\n
{ action: { metadata }}\n
{ request body        }\n
```

- 使用curl -XPOST -d 时注意，不能直接在json字符串中添加\n字符，应该按回车
- create 和index的区别
  - 如果数据存在，使用create操作失败，会提示文档已经存在，使用index则可以成功执行。
- 使用文件的方式
  - vi requests
  - curl -XPOST/PUT localhost:9200/_bulk --data-binary @request;
- bulk请求可以在URL中声明/_index 或者/_index/_type
- bulk一次最大处理多少数据量
  - bulk会把将要处理的数据载入内存中，所以数据量是有限制的
  - 最佳的数据量不是一个确定的数值，它取决于你的硬件，你的文档大小以及复杂性，你的索引以及搜索的负载
  - 一般建议是1000-5000个文档，如果你的文档很大，可以适当减少队列，大小建议是5-15MB，默认不能超过100M，可以在es的配置文件中修改这个值http.max_content_length: 100mb

- Elasticsearch的版本控制
- 普通关系型数据库使用的是（悲观并发控制（PCC））
  - 当我们在读取一个数据前先锁定这一行，然后确保只有读取到数据的这个线程可以修改这一行数据
- ES使用的是（乐观并发控制（OCC））
  - ES不会阻止某一数据的访问，然而，如果基础数据在我们读取和写入的间隔中发生了变化，更新就会失败，这时候就由程序来决定如何处理这个冲突。它可以重新读取新数据来进行更新，又或者将这一情况直接反馈给用户。
- ES如何实现版本控制(使用es内部版本号)
  - 1：首先得到需要修改的文档，获取版本(_version)号
    - curl -XGET http://localhost:9200/elasticsearch/emp/1
  - 2：在执行更新操作的时候把版本号传过去
    - curl -XPUT http://localhost:9200/elasticsearch/emp/1?version=1 -d '{"name": “zs","age":25}'(覆盖)
    - curl -XPOST http://localhost:9200/elasticsearch/emp/1/_update?version=1 -d '{"doc":{"city":"beijing","car":"BMW"}}'(部分更新)
  - 3：如果传递的版本号和待更新的文档的版本号不一致，则会更新失败

- Elasticsearch的版本控制
- ES如何实现版本控制(使用外部版本号)
  - 如果你的数据库已经存在了版本号，或者是可以代表版本的时间戳。这时就可以在es的查询url后面添加version_type=external来使用这些号码。
  - 注意：版本号码必须要是大于0小于9223372036854775807（Java中long的最大正值）的整数。
  - es在处理外部版本号的时候，它不再检查_version是否与请求中指定的数值是否相等，而是检查当前的_version是否比指定的数值小，如果小，则请求成功。
  - example：
    - curl -XPUT 'http://localhost:9200/bjsxt/emp/20?version=10&version_type=external' -d '{"name": "laoxiao"}'
    - 注意：此处url前后的引号不能省略，否则执行的时候会报错

- Elasticsearch的插件
- 站点插件（以网页形式展现）
  - BigDesk Plugin (作者 Lukáš Vlček)
  - 简介：监控es状态的插件，推荐！
  - Elasticsearch Head Plugin (作者 Ben Birch)
  - 简介：很方便对es进行各种操作的客户端。
  - Paramedic Plugin (作者 Karel Minařík)
  - 简介：es监控插件
  - SegmentSpy Plugin (作者 Zachary Tong)
  - 简介：查看es索引segment状态的插件
  - Inquisitor Plugin (作者 Zachary Tong)
  - 简介：这个插件主要用来调试你的查询。

- Elasticsearch的插件
- 这个主要提供的是节点的实时状态监控，包括jvm的情况，linux的情况，elasticsearch的情况
  - 安装bin/plugin install lukas-vlcek/bigdesk
  - 删除bin/plugin remove bigdesk
- http://192.168.57.4:9200/_plugin/bigdesk/
- http://192.168.57.4:9200/_cluster/state?pretty

- Elasticsearch的插件
- 这个主要提供的是健康状态查询
  - 安装bin/plugin -install mobz/elasticsearch-head
- http://192.168.57.4:9200/_plugin/head/
- http://192.168.57.4:9200/_cluster/health?pretty