### 一．搭建环境

ElasticSearch下载地址：<https://www.elastic.co/cn/downloads/elasticsearch>

配置elasticsearch

修改一下es使用的参数。编辑config/elasticsearch.yml：

# 换个集群的名字，免得跟别人的集群混在一起

cluster.name: gqm

# 换个节点名字

node.name: node-001

# 修改一下ES的监听地址，这样别的机器也可以访问

network.host: 0.0.0.0

# 默认的就好

http.port: 9200

# 增加新的参数，这样head插件可以访问es

http.cors.enabled: true

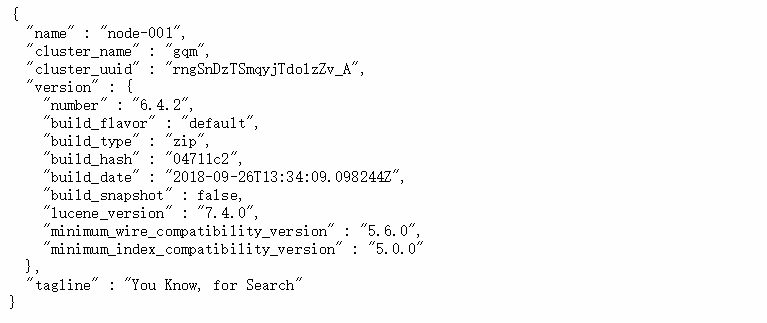
http.cors.allow-origin: "\*”

启动elasticsearch

双击elasticseatch目录下bin里的elasticsearch.bat

访问elasticsearch

在浏览器输入地址：127.0.0.1：9200



安装ElasticSearch6.3.1的Head插件

ElasticSearch-Head 下载地址：<https://github.com/mobz/elasticsearch-head.git>

一、安装nodejs

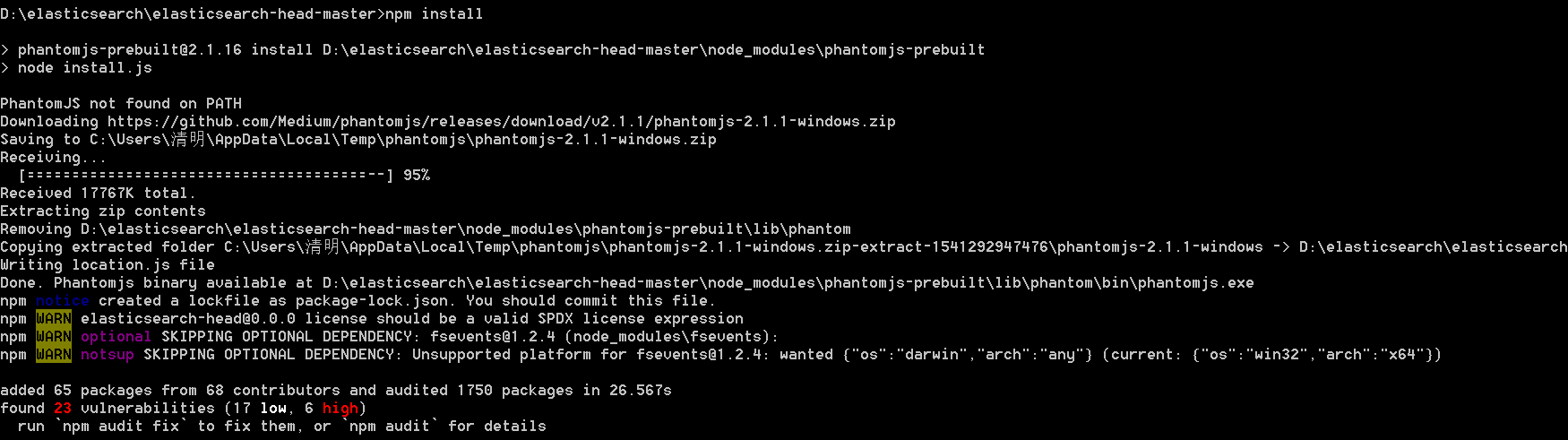
二、安装grunt

路径切到nodejs安装目录下下。

执行：npm install -g grunt-cli （-g代表全局安装。）

然后在head源码目录中，执行npm install 下载的包：

npm install



最后，在head源代码目录下启动nodejs：

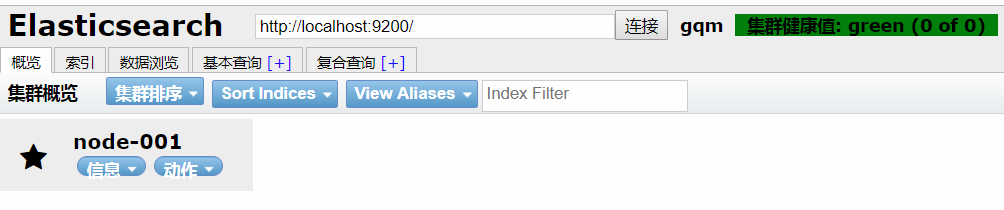
grunt server



访问ElasticSearch-Head

在浏览器输入地址：127.0.0.1：9100

然后在ElasticSearch-Head插件的输入框里输入127.0.0.1：9200就可以查看到我们的elasticsearch服务



### 一基本概念

索引：含有相同属性的文档的集合 （数据库名）

类型：索引可以定义一个或多个类型文档必须属于类型 （数据库表）

文档：是可以被索引的基本数据单位 (数据库记录)

分片：每个索引都有多个分片每个分片是一个Lucene索引

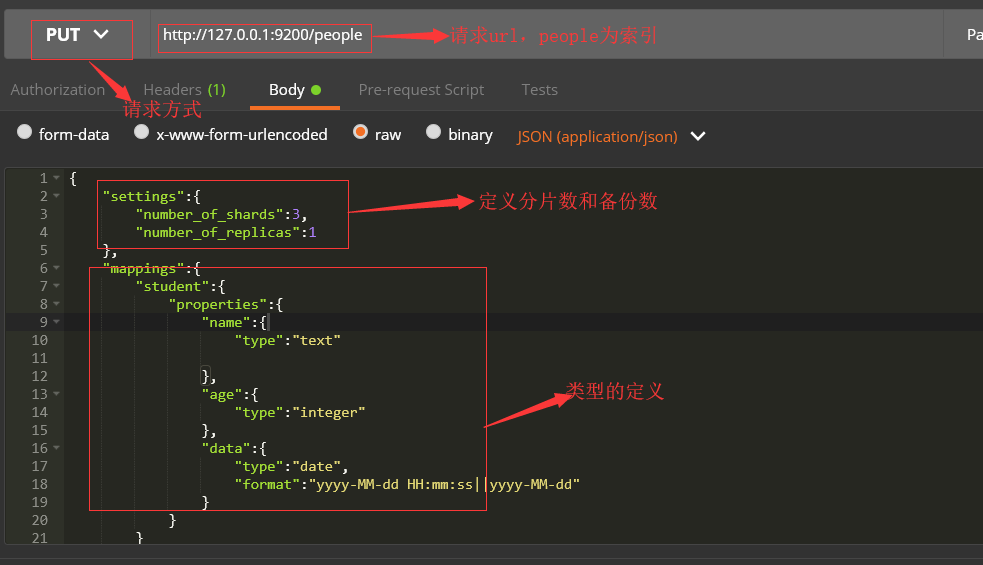
备份：拷贝一份分片就完成列分片的备份

### 二．简单使用

API基本格式：http://<ip>:<port>/<索引>/<类型>/<文档id>

工具:postman

1. 创建索引user和类型



创建一个people索引和一个student类型，

{

"settings":{

"number\_of\_shards":3,

"number\_of\_replicas":1

},

"mappings":{

"student":{

"properties":{

"name":{

"type":"text"

},

"age":{

"type":"integer"

},

"date":{

"type":"date",

"format":"yyyy-MM-dd HH:mm:ss||yyyy-MM-dd"

}

}

}

}

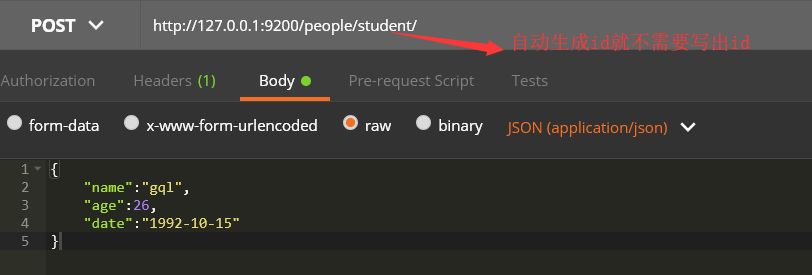
}

1. 插入数据

指定文档id插入

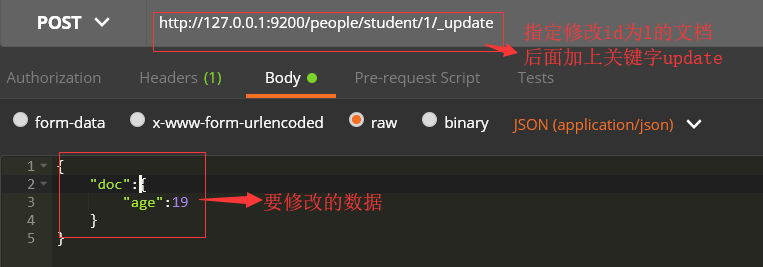


自动生成id

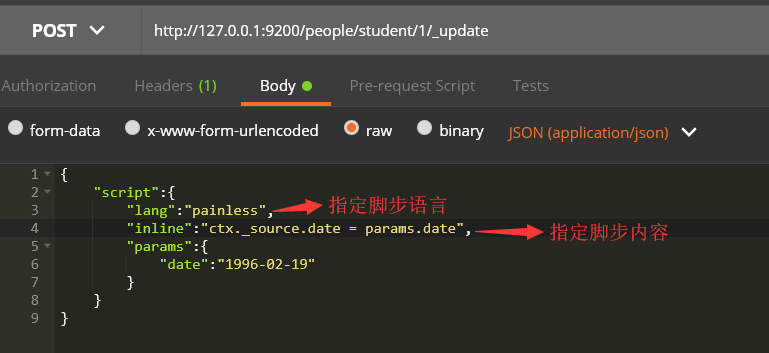


1. 修改数据

直接修改文档

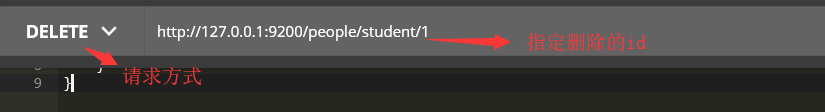


通过脚本修改

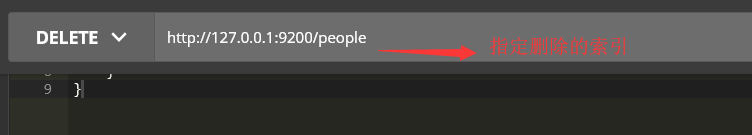


1. 删除操作

删除文档

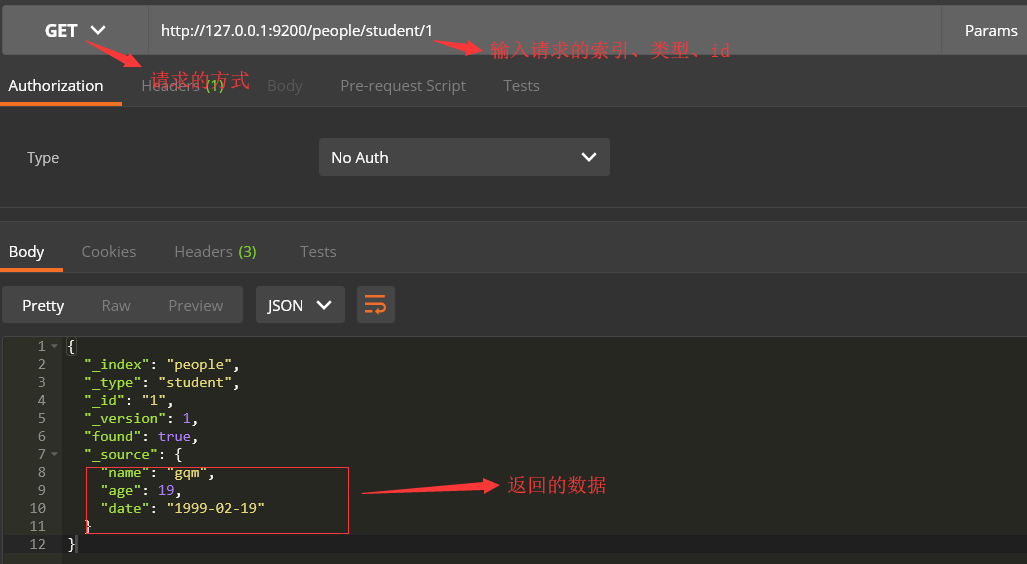


删除索引

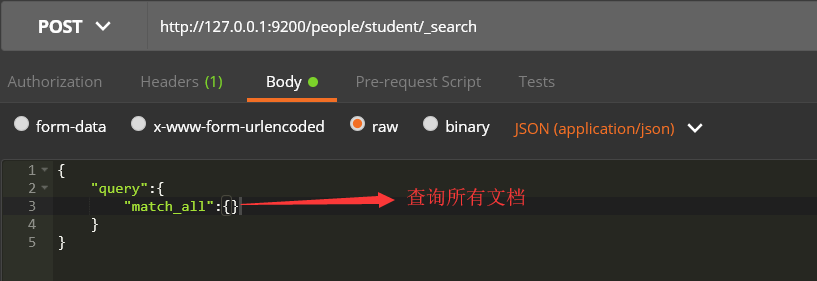


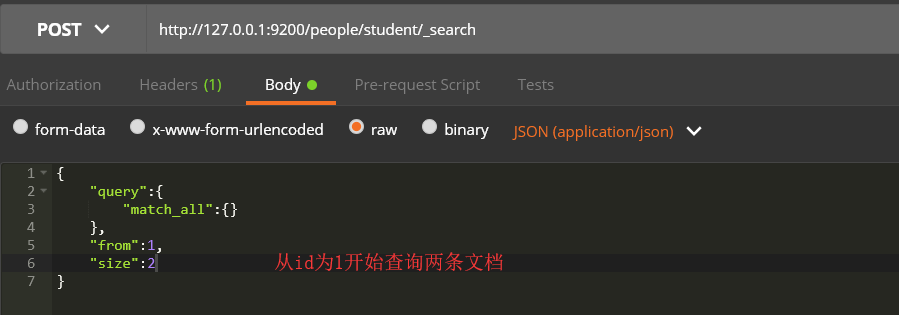
1. 查询语法

简单查询



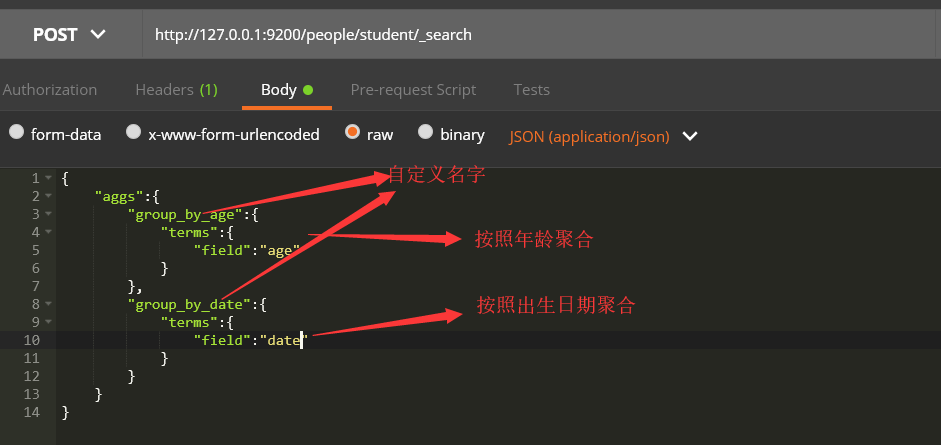
条件查询



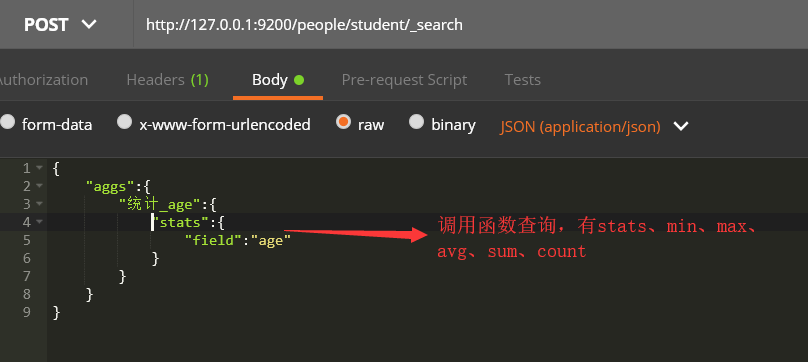




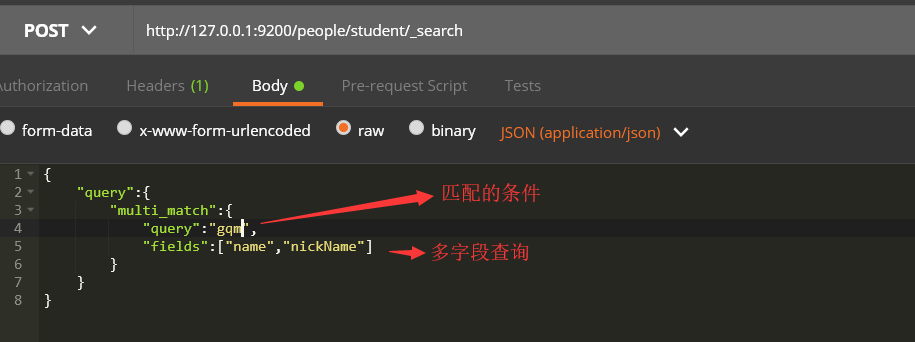
聚合查询

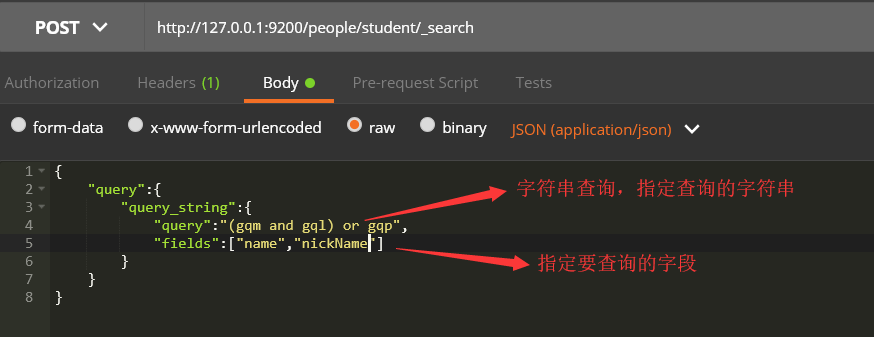


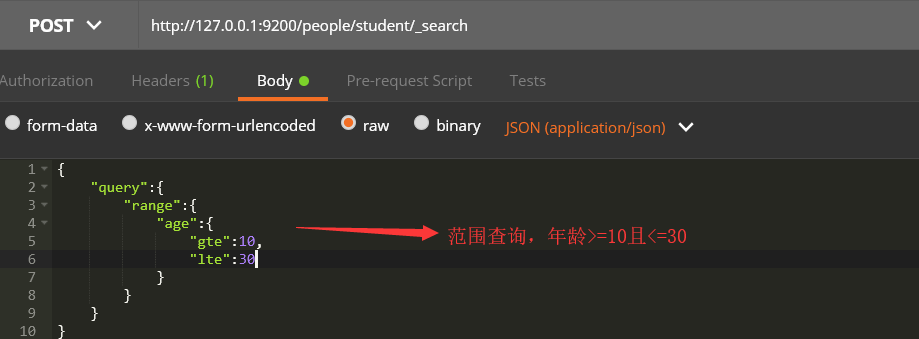
调用函数查询



高级查询

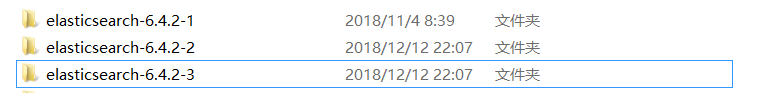






### 三．集群搭建

1.先在本地解压多个（我这里三个）elasticsearch，其中1为主节点，其他为从节点



2.配置主节点的yml

# 集群的名字，免得跟别人的集群混在一起

cluster.name: gqm

# 换个节点名字

node.name: node-001

# 修改一下ES的监听地址，这样别的机器也可以访问

network.host: 0.0.0.0

# 默认的就好

http.port: 9200

#tcp端口

transport.tcp.port: 9300

# 增加新的参数，这样head插件可以访问es

http.cors.enabled: true

http.cors.allow-origin: "\*"

#指定了该节点可能成为 master 节点，还可以是数据节点

node.master: true

node.data: true

# 集群个节点IP地址，也可以使用els、els.shuaiguoxia.com等名称，需要各节点能够解析

discovery.zen.ping.unicast.hosts: ["127.0.0.1:9300","127.0.0.1:9301", "127.0.0.1:9302"]

2.配置第一个从节点的yml

# 集群的名字，免得跟别人的集群混在一起

cluster.name: gqm

# 换个节点名字与其他不一样

node.name: node-002

# 修改一下ES的监听地址，这样别的机器也可以访问

network.host: 0.0.0.0

# 与其他不一样

http.port: 9201

#tcp端口与其他不一样

transport.tcp.port: 9301

# 增加新的参数，这样head插件可以访问es

http.cors.enabled: true

http.cors.allow-origin: "\*"

# 集群个节点IP地址，也可以使用els、els.shuaiguoxia.com等名称，需要各节点能够解析

discovery.zen.ping.unicast.hosts: ["127.0.0.1:9300","127.0.0.1:9301", "127.0.0.1:9302"]

3.配置第二个从节点的yml

# 集群的名字，免得跟别人的集群混在一起

cluster.name: gqm

# 换个节点名字与其他不一样

node.name: node-003

# 修改一下ES的监听地址，这样别的机器也可以访问

network.host: 0.0.0.0

# 与其他不一样

http.port: 9202

#tcp端口与其他不一样

transport.tcp.port: 9302

# 增加新的参数，这样head插件可以访问es

http.cors.enabled: true

http.cors.allow-origin: "\*"

# 集群个节点IP地址，也可以使用els、els.shuaiguoxia.com等名称，需要各节点能够解析

discovery.zen.ping.unicast.hosts: ["127.0.0.1:9300","127.0.0.1:9301", "127.0.0.1:9302"]

4.依次启动主从节点通过head插件访问主节点或通过<http://127.0.0.1:9200/_cat/nodes?v> 查看集群的节点

