模式

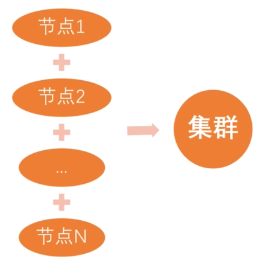
工厂：通过抽象类实现不同工厂需求 策略

简单工厂：工厂的工厂 通过两个接口实现不同需求 构建一组相互依赖的对象

建造者：通过将建造对象（产品 产品生产方法）传入指导类（组件产品流程）中，创建产品

  1> 集群和节点

一个es集群是由一个或多和es节点组成的集合



每一个集群都有一个名字, 如之前的wali

每个节点都有自己的名字, 如之前的master, slave1, slave2

节点是可以存储数据, 参与索引数据等的独立服务

2> 索引(类似于数据库里面的database)

索引是含有相同属性的文档集合

索引在es中是通过一个名字来识别的, 必须是英文字母小写, 且不含中划线

3> 类型(相当于sql中的table)

一个索引可以定义一个或多个类型, 文档必须属于一个类型

4> 文档(相当于sql中的一行记录)

文档是可以被索引的基本数据单位

5> 分片

每个索引都有多个分片, 每个分片都是一个luncene索引

分片的好处: 分摊索引的搜索压力, 分片还支持水平的拓展和拆分以及分布式的操作, 可以提高搜索和其他处理的效率

6> 备份

拷贝一个分片就完成了分片的备份

备份的好处: 当主分片失败或者挂掉, 备份就可以代替分片进行操作, 进而提高了es的可用性, 备份的分片还可以进行搜索操作, 以分摊搜索的压力.

ES在创建索引时, 默认创建5个分片, 一份备份, 可以修改, 分片的数量只能在创建索引的时候指定, 索引创建后就不能修改分片的数量了, 而备份是可以动态修改的

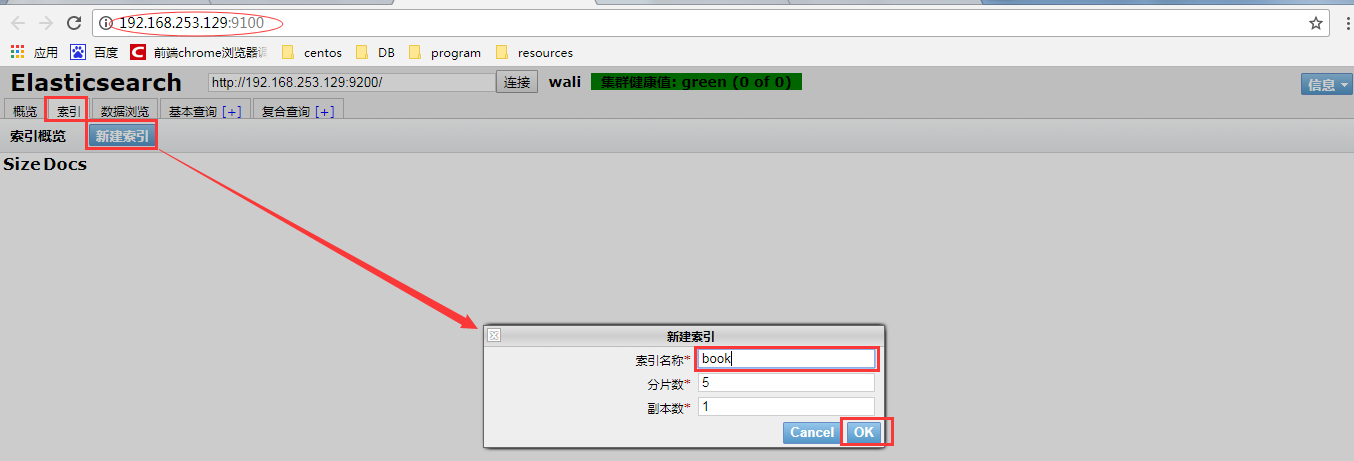
## **2.2 ES的基本用法**

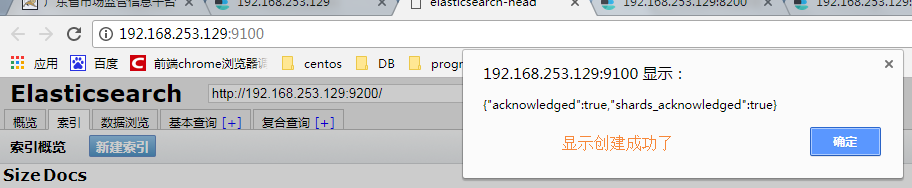
ES以RESTFul风格来命名API的, 其API的基本格式如下

http://<ip>:<port>/<索引>/<类型>/<文档id>

ES的动作是以http方法来决定的: 常用的http方法: GET/PUT/POST/DELETE

Eg1, 创建索引, 结合head插件创建索引的实例





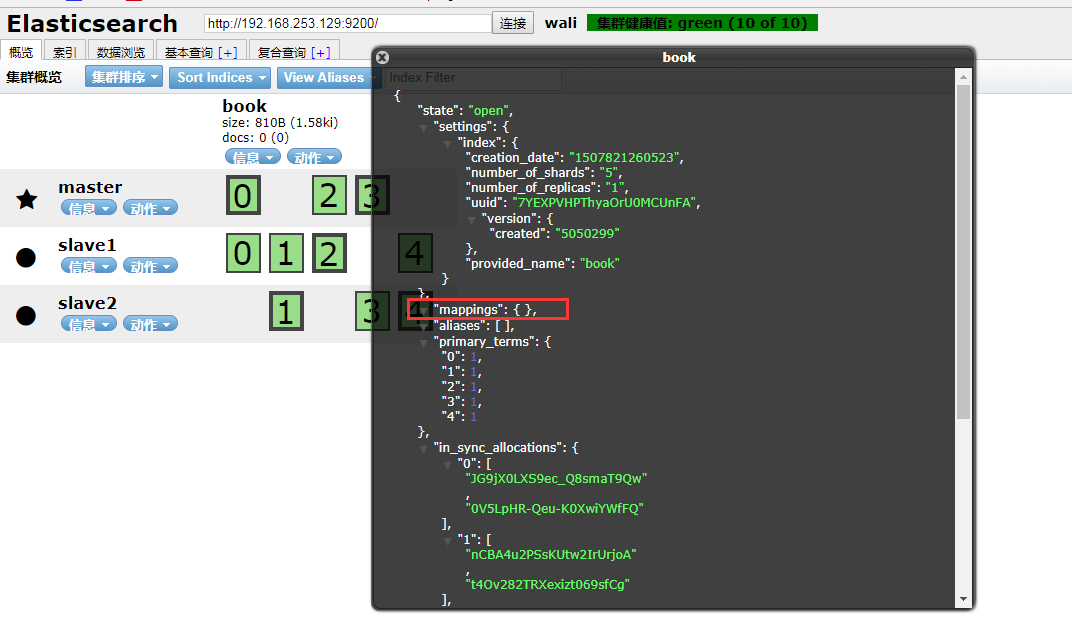
返回”概览”查看创建情况



创建索引分为: **结构化创建与非结构化创建**

查看索引是否是结构化的方法

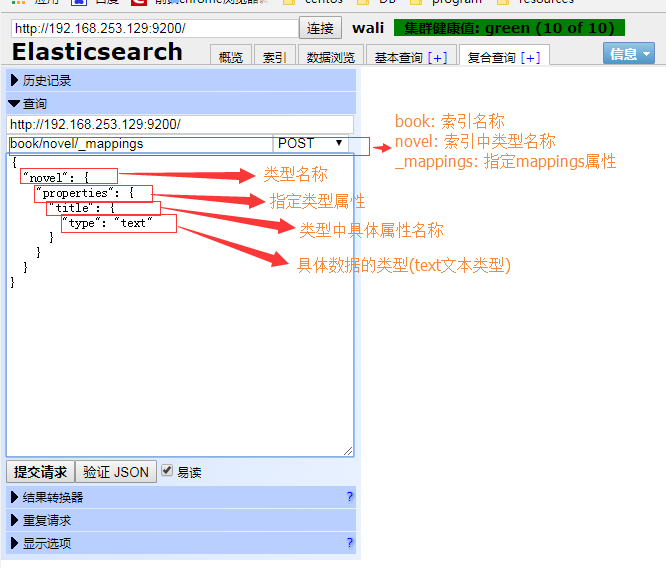




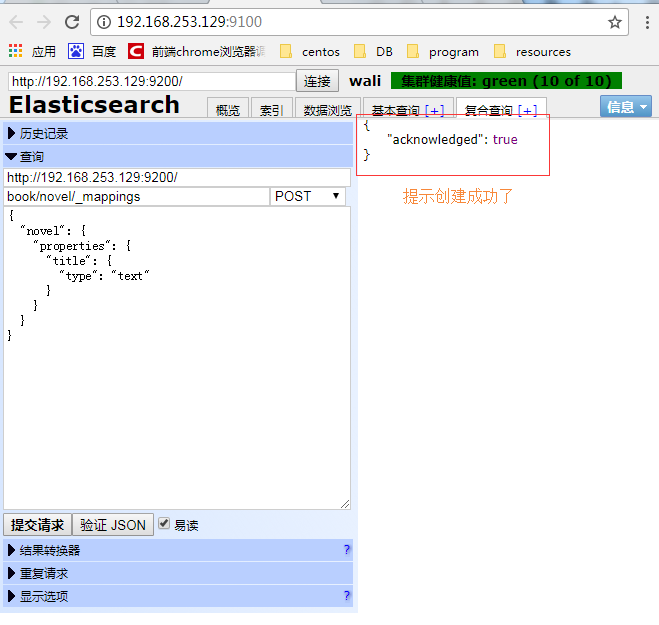
**Mappings**是结构化的一个关键词, 其后内容是空的, 说明这个索引是一个非结构化的索引

创建结构化索引的方法

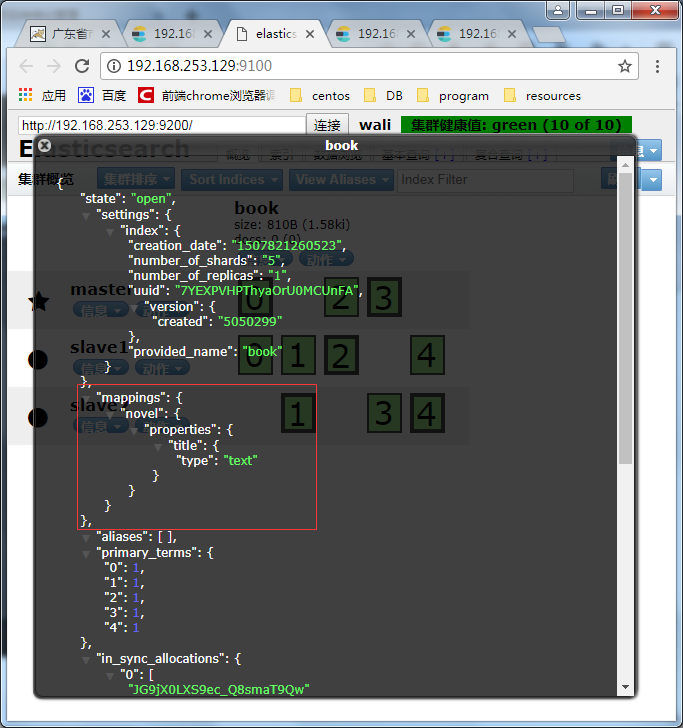
点击head插件的”复合查询”



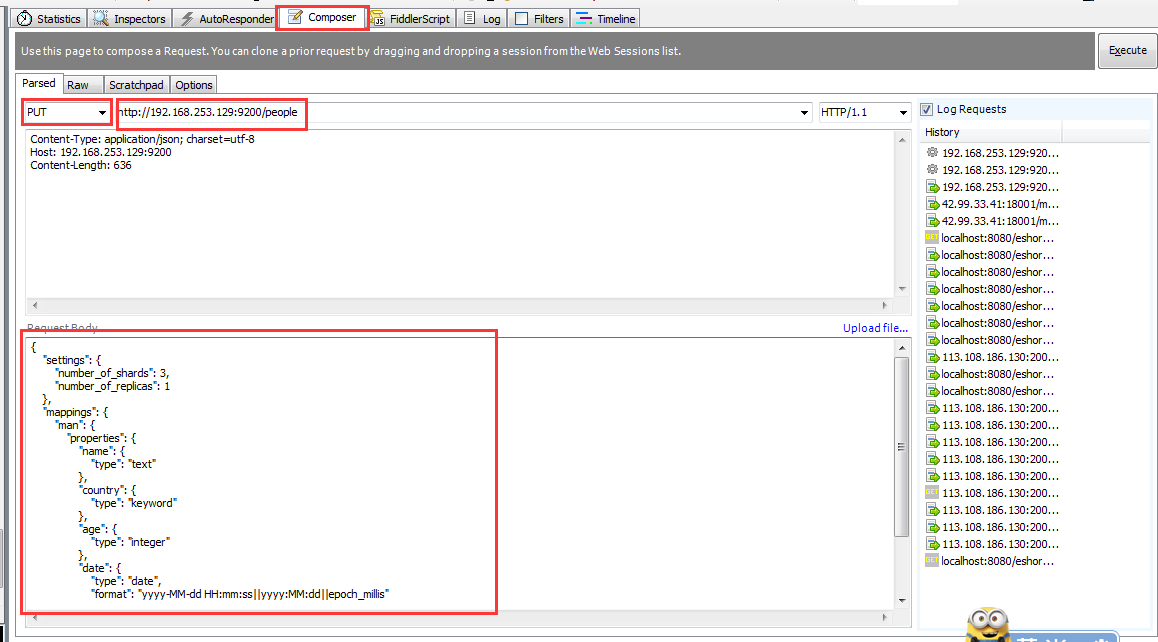
验证json后, 提交请求,



返回head插件”概览查看”



Head插件中编写json发送请求比较复杂, 可以使用postman或者是fiddler等工具进行请求的发送



请求数据的json格式

{

    "settings": {

        "number\_of\_shards": 3,

        "number\_of\_replicas": 1

    },

    "mappings": {

        "man": {

            "properties": {

                "name": {

                    "type": "text"

                },

                "country": {

                    "type": "keyword"

                },

                "age": {

                    "type": "integer"

                },

                "date": {

                    "type": "date",

                    "format": "yyyy-MM-dd HH:mm:ss||yyyy:MM:dd||epoch\_millis"

                }

            }

        },

        "woman": {}

    }

}

### **2.2.1> 数据插入**

文档id, 唯一索引值, 指向文档数据

#### **2.2.1.1 指定文档id插入**

使用http中的**put**方法,

插入时输入的ip地址, 如<http://192.168.253.129:9200/people/man/1>

请求参数依次为:索引名称/类型名称/文档id

请求参数

{

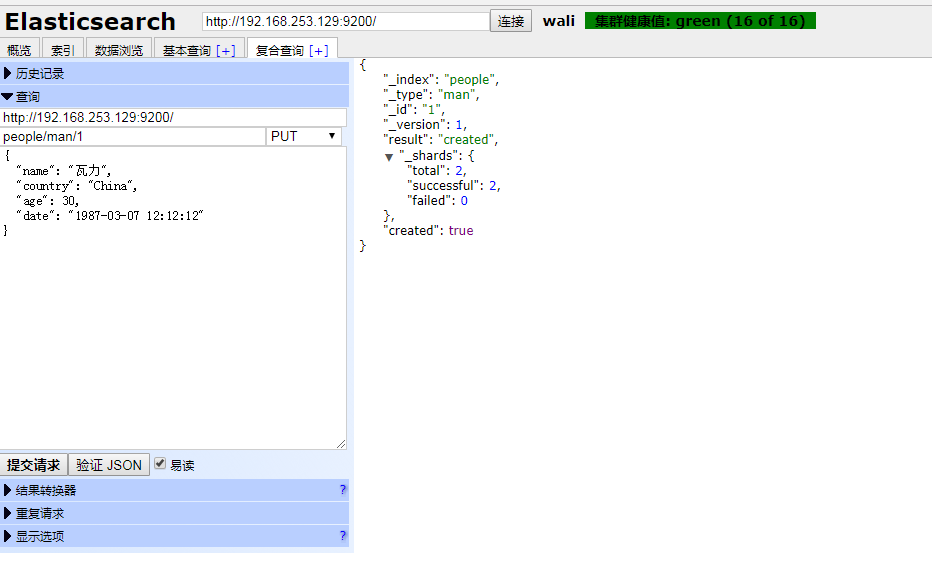
  "name": "瓦力",

  "country": "China",

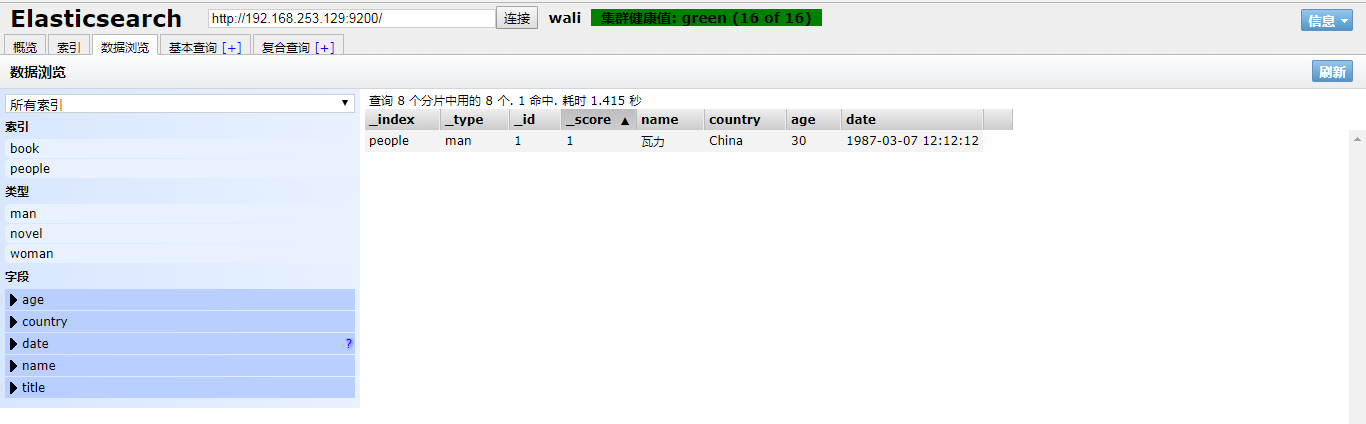
  "age": 30,

  "date": "1987-03-07 12:12:12"

}



Head插件中”数据浏览”



#### **2.2.1.2 自动产生文档id插入**

使用http中的**post**方法,

插入时输入的ip地址, 如http://192.168.253.129:9200/people/man

请求参数

{

  "name": "超重瓦力",

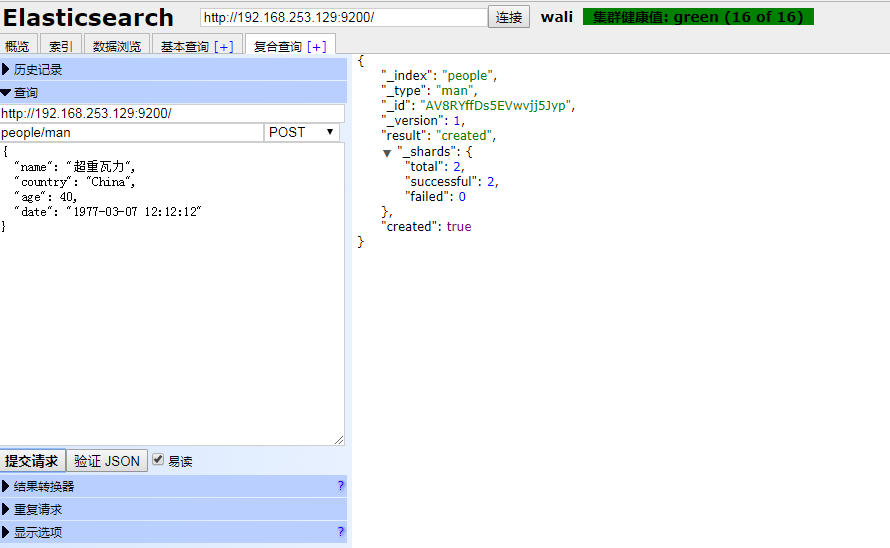
  "country": "China",

  "age": 40,

  "date": "1977-03-07 12:12:12"

}

响应结果



### **2.2.2> 修改文档数据**

#### **2.2.2.1 直接修改文档**

http方法: **post**方法

请求地址:  <http://192.168.253.129:9200/people/man/1/_update>

关键词: \_update, doc

请求参数

{

    "doc": {

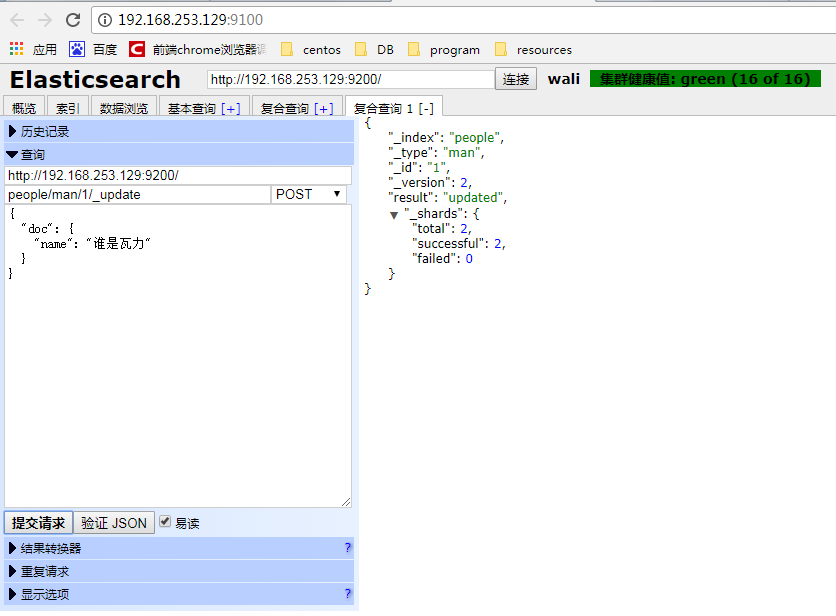
        "name": "谁是瓦力"

    }

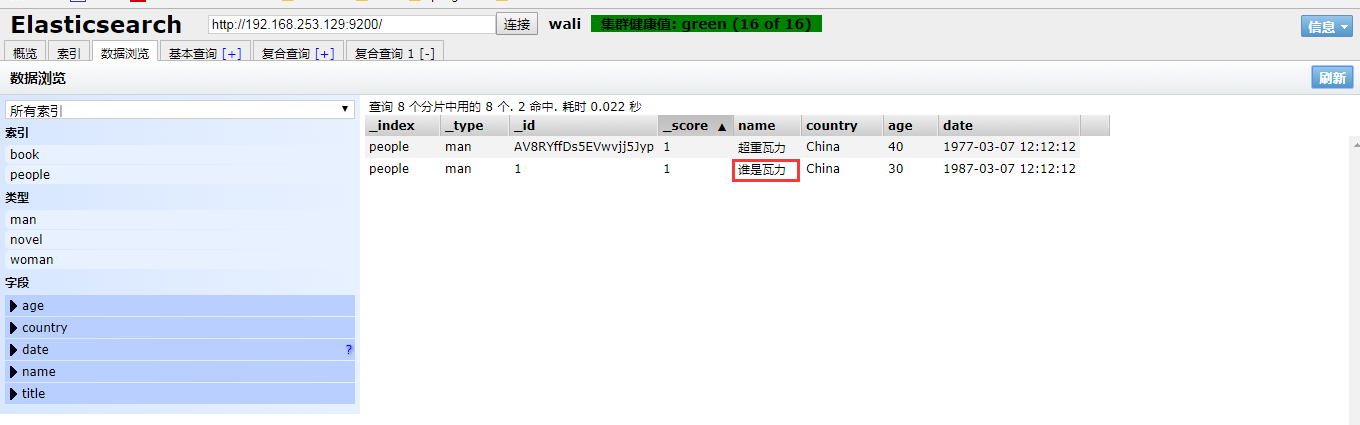
}

“doc”为关键字, 要修改的文档放在doc中, 实例修改了type为people索引下man中id为1 的name属性

返回的结果



可以在head插件中”数据浏览”中验证修改结果



#### **2.2.2.2 脚本修改文档**

通过脚本修改的api格式与直接修改的是一致的

http方法: **post**方法

请求地址:  <http://192.168.253.129:9200/people/man/1/_update>

请求参数

{

  "script": {

    "lang": "painless",

    "inline": "ctx.\_source.age += 30"

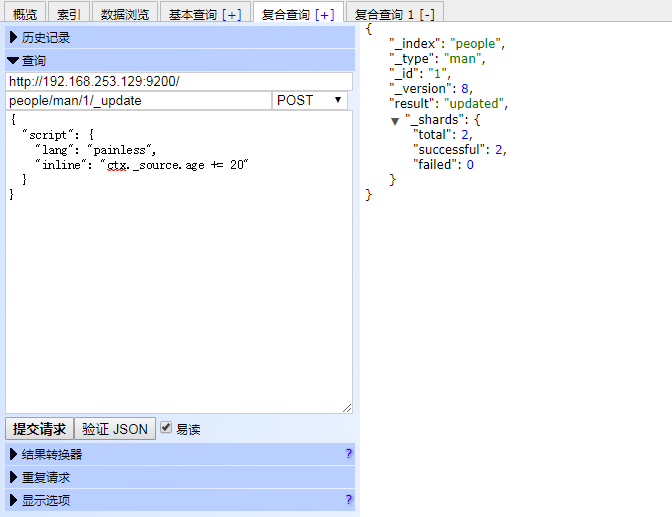
  }

}

关键字”script”: 标志以脚本的方式修改文档

“lang”: 表示以何种脚本语言进行修改, “painless”表示以es内置的脚本语言进行修改. 此外es还支持多种脚本语言, 如Python, js等等

“inline”:指定脚本内容 “ctx”代表es上下文, \_source 代表文档



可以在head插件中”数据浏览”检查修改数据

此外还有其他的参数设置方式, 例如将参数放到外面

{

  "script": {

    "lang": "painless",

    "inline": "ctx.\_source.name = params.name; ctx.\_source.age = params.age",

    "params": {

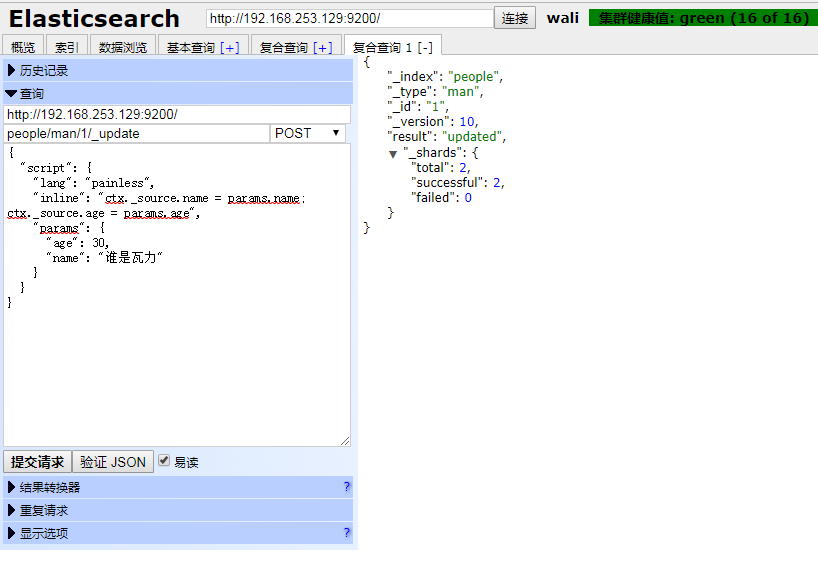
      "age": 30,

      "name": "谁是瓦力"

    }

  }

}



### **2.2.3> 删除**

#### **2.2.3.1 删除文档**

http方法: **delete**

请求路径http://192.168.253.129:9200/people/man/1



#### **2.2.3.1 删除索引**

1. 可以通过head插件进行删除



2. 通过api删除

Http方法: delete

链接地址: <http://192.168.253.129:9200/book>

