GET \_search

{

"query": {

"match\_all": {}

}

}

//单纯添加一个索引

PUT /student

//为索引添加结构

POST /student/\_mapping

{

"properties":{

"id":{

"type":"integer"

},

"name":{

"type":"text"

},

"age":{

"type":"integer"

}

}

}

//创建有结构的索引

PUT /student1

{

"mappings": {

"properties": {

"id": {

"type": "integer"

},

"name": {

"type": "text"

},

"age": {

"type": "integer"

}

}

}

}

DELETE /student1

//新增修改文档(id存在修改 不存在新增 id不写自动生成)

POST /student/\_doc/1

{

"id":1,

"name":"Heria",

"age":22

}

POST /student/\_doc/3

{

"id":3,

"name":"John",

"age":22

}

//查询文档

GET /student/\_doc/2

//删除文档

DELETE /student/\_doc/8sSog4kBHlU0A2aCug1W

//批量查询

GET /student/\_mget

{

"docs":[

{"\_id":1},

{"\_id":2},

{"\_id":3}

]

}

//查询所有文档

GET /student2/\_search

{

"query": {

"match\_all": {}

}

}

//修改文档部分字段

POST /student/\_doc/2/\_update

{

"doc":{

"name":"LiuYiPing"

}

}

POST /student2/\_doc/1/\_update

{

"doc":{

"name":"my life is broking"

}

}

-------------------------添加索引,根据索引查询字段-------------------

PUT /student1

{

"mappings": {

"properties": {

"age":{

"type": "integer"

},

"name": {

"type": "text",

"index": true,

"store": true

}

}

}

}

PUT /student2

{

"mappings": {

"properties": {

"age":{

"type": "integer"

},

"name": {

"type": "text",

"index": true,

"store": false

}

}

}

}

-----------------2个索引中添加2个文档-------------------

POST /student1/\_doc/1

{

"name":"My Life is broking",

"age":32

}

POST /student2/\_doc/1

{

"name":"My Life is broking",

"age":12

}

---------------------------------------------------------

----------------------根据字段查询---------------------

GET /student1/\_search

{

"query": {

"term": {

"name":"life"

}

}

}

GET /student2/\_search

{

"query": {

"term": {

"name":"my life is broking"

}

}

}

---------------------------------------------------------

----------------------根据ID查询---------------------

GET /student2/\_doc/1

---------------------------------------------------------

----------------------根据字段查询---------------------

GET /student2/\_search

{

"query": {

"term": {

"name":"my life is broking"

}

}

}

----------------------批量查询----------------------

GET /student2/\_mget

{

"docs":[

{"\_id":1}

]

}

--------------------单独查询某个域----------------------

GET /student1/\_search

{

"stored\_fields": ["name"]

}

//查不到

GET /student2/\_search

{

"stored\_fields": ["name"]

}

---------------------------------------------------------

GET /\_analyze

{

"text": "I love Spring",

"analyzer": "standard"

}

GET /\_analyze

{

"text": "我是你爹",

"analyzer": "standard"

}

---------------------------------------------------------

------------------测试ik分词器---------------------------------------

GET /\_analyze

{

"text": "我真的还想再活五百年",

"analyzer": "standard"

}

GET /\_analyze

{

"text": "我爱玩英雄联盟",

"analyzer": "ik\_smart"

}

GET /\_analyze

{

"text": "我爱玩英雄联盟",

"analyzer": "ik\_max\_word"

}

------------------测试pinyin分词器---------------------------------------

GET /\_analyze

{

"text": "我爱玩英雄联盟",

"analyzer": "pinyin"

}

------------------测试自定义分词器---------------------------------------

PUT /student3

{

"settings": {

"analysis": {

"analyzer": {

"ik\_pinyin":{

"tokenizer":"ik\_max\_word",

"filter":"pinyin\_filter"

}

},

"filter": {

"pinyin\_filter":{

"type":"pinyin",

"keep\_separate\_first\_letter" : false, // 是否分词每个字的首字母

"keep\_full\_pinyin" : true, // 是否分词全拼

"keep\_original" : true, // 是否保留原始输入

"remove\_duplicated\_term" : true // 是否删除重复项

}

}

}

},

"mappings": {

"properties": {

"name":{

"type": "text",

"store": true,

"index": true,

"analyzer": "ik\_pinyin"

},

"age":{

"type": "integer"

}

}

}

}

GET /student3/\_analyze

{

"text": "你好混日子的选手",

"analyzer": "ik\_pinyin"

}