elasticsearch
for developer

CRUD operation

book_02.txt

การจัดการข้อมูล

- Indexing document

- Retrieve document

- Check existing document

- Update document

- Create new document

- Delete document

- Partial update document

Elasticsearch

โดยปกติทุกๆ field จะถูก indexed เสมอ

Document

Top-level ของ object (JSON)

Document metadata

- `_index`
- `_type`
- `_id`

```
"hits": [
    {
        "_index": "store",
        "_type": "book",
        "_id": "2",
        "_score": 0.058131136,
        "_source": {
            "title": "Clean Code: A Handbook of Agile
```

Compare with RDBMS

DATABASE	TABLE	ROW	COLUMN
INDEX	TYPE	DOCUMENT	FIELD

Indexing document

กำหนด ID เอง

PUT /{index}/{type}/{id}

Indexing document

สร้าง ID แบบอัตโนมัติ

POST /{index}/{type}/

```
"hits": [
    {
        "_index": "store",
        "_type": "book",                    UUID
        "_id": "AUrnsc0Ny8oVZ-EGkX9f",
        "_score": 1,
        "_source": {
```

Retrieve document

GET /{index}/{type}/{id}

Retrieve document

ถ้าต้องการข้อมูลบางส่วนล่ะ

GET /store/book/1?_source=title,description

```json
{
    "_index": "store",
    "_type": "book",
    "_id": "1",
    "_version": 2,
    "found": true,
    "_source": {
        "title": "Elasticsearch: The Definitive Guide",
        "description": "A Distributed Real-Time Search and Analytics Engine"
    }
}
```

Check existing document

ตรวจสอบว่ามี document หรือไม่

curl -i -XHEAD http://localhost:9200/store/book/1

```
somkiatpui:elasticsearch-1.4.2 somkiat$ curl -i -XHEAD http://localhost:9200/store/book/1
HTTP/1.1 200 OK
Content-Type: text/plain; charset=UTF-8
Content-Length: 0

somkiatpui:elasticsearch-1.4.2 somkiat$ curl -i -XHEAD http://localhost:9200/store/book/100
HTTP/1.1 404 Not Found
Content-Type: text/plain; charset=UTF-8
Content-Length: 0
```

Update document

ทำการแก้ไขข้อมูลทั้ง document

PUT /store/book/123

```
{
    "title" : "Update",

    "author_name" : ["user1", "user2"],

    "tag" : ["update", "book"]

}
```

```
{
    "_index": "store",
    "_type": "book",
    "_id": "1",
    "_version": 8,
    "created": false
}
```

versioning

Partial update document

ทำการแก้ไขข้อมูลบาง fields

POST /store/book/1/_update
```
{
    "doc" : {
        "title" : "partial update",
        "tag" : [ "test", "computer" ],
        "views": 0
} }
```

result
```
{
    "_index": "store",
    "_type": "book",
    "_id": "1",
    "_version": 11
}
```

Partial update document

ทำการดูรายละเอียด

GET /store/book/1

```
"found": true,
"_source": {
    "title": "partial update",
    "author_name": [
        "Clinton Gormley",
        "Zachary Tong"
    ],
    "tag": [
        "test",
        "computer"
    ],
    "isbn-13": "978-1449358549",
    "isbn-10": "1449358543",
    "price": 44.3,
    "page": 724,
    "description": "A Distributed Real-Time Search and Analytics
    "detail": "Learn how to use Elasticsearch, an open source, di
RESTful search engine built on top of Apache Lucene. Each chapter i
tackles a particular facet of Elasticsearch with separate sections
more advanced programmers. If you're a beginner, advanced technique
required reading, but you can revisit them once you have a solid un
the basics.",
    "views": 0
}
```

Partial update document

แก้ไขด้วย Script ให้เพิ่มค่าจำนวน view ของหนังสือ

```
POST /store/book/1/_update
{

    "script" : "ctx._source.views+=1"

}
```

Partial update document

แก้ไขด้วย Script ให้เพิ่ม tag ของหนังสือ

POST /store/book/1/_update

```
{
    "script": "ctx._source.tag+=new_tag",
    "params": {
            "new_tag": "search"
    }
}
```

Partial update document

ถ้าบาง document ไม่มี views ล่ะ ?

POST /store/book/1/_update
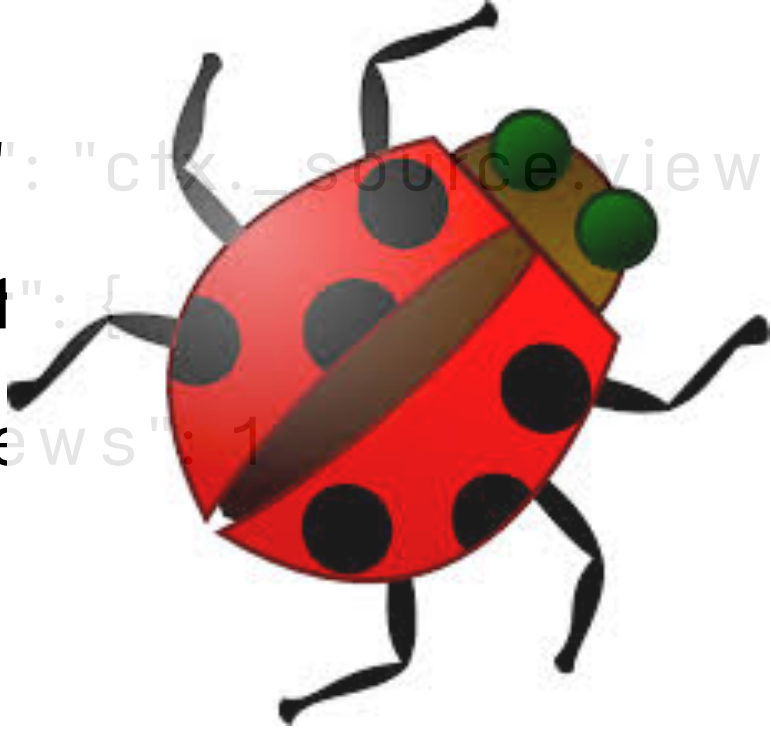
```
{
    "script": "ctx._source.views+=1",
    "upsert": {
        "views": 1
    }
}
```

Partial update document

ถ้าบาง document ไม่มี views ล่ะ ?

POST /store/book/1/_update

```
{
    "script": "ctx._source.views+=1",
    "upsert": {
        "views": 1
    }
}
```

Partial update document

ถ้าบาง document ไม่มี views ล่ะ ?

POST /store/book/1/_update

```
{
    "script": "if (!ctx._source.views)
                    {ctx._source.views = 1};
            ctx._source.views += views",
    "params": {
        "views": 5
    }
}
```

Create new document

POST /{index}/{type}/

Create new document

ถ้า ID ซ้ำกันล่ะ ป้องกันอย่างไร?

PUT /store/book/123?**op_type=create**

```
{
    "title" : "New",
    "author_name" : ["user1", "user2"],
    "tag" : ["new", "book"]
}
```

Create new document

ลองสร้างอีกครั้ง

PUT /store/book/123/**_create**

```
{
    "title" : "New",
    "author_name" : ["user1", "user2"],
    "tag" : ["new", "book"]
}
```

Result with error

```
{
    "error": "DocumentAlreadyExistsException[[store][4]
[book][123]: document already exists]",
    "status": 409
}
```

Delete document

DELETE /store/book/123

First

```json
{
    "found": true,
    "_index": "store",
    "_type": "book",
    "_id": "123",
    "_version": 3
}
```

Second

```json
{
    "found": false,
    "_index": "store",
    "_type": "book",
    "_id": "123",
    "_version": 1
}
```

Delete by query

```
DELETE /store/book/_query
{
  "query": {
    "term": {
      "tag": "elasticsearch"
    }
  }
}
```