

Elasticsearch 입력/조회/삭제

Elasticsearch	Realational DB	CRUD
GET	Select	Read
PUT	Update	Update
POST	Insert	Create
DELETE	Delete	Delete

실습 환경 구축

- 리눅스 명령어를 사용해서 쿼리문을 작성하기 위해 설치하는 윈도우에서 하고 WSL을 사용해서 WSL 환경에서 실습을 진행한다.
- 이를 위해 /config/elasticsearch.yml에 다음 내용을 추가한다.

```
network.host: 0.0.0.0
http.port: 9200

cluster.name: my-cluster
node.name: node-1
cluster.initial_master_nodes: ["node-1"]

xpack.security.enabled: false
```

- 이후 Elasticsearch.bat를 재시작한다. 이러면 모든 로컬 IP에서 접속이 허용된다. 테스트 용으로만 사용한다.
- 윈도우 powershell에서 ipconfig를 수행하고 wsl에서 window로 통신할 때 사용하는 ip주소를 확인한다.
- curl http://172.22.208.1:9200 와 같은 형식으로 접속 가능 여부를 확인할 수 있다.

입력/조회/삭제 실습

- curl 커맨드로 API를 보낼 때 -X라는 prefix를 붙인다.
- curl -X{http Method} {elasticsearch 주소}/{index} 형식으로 접속한다.

Index 조회

- curl -XGET http://172.22.208.1:9200/classes
- 결과값을 더 직관적으로 확인하려면 ?pretty를 붙이면 된다. json 포맷으로 들여쓰기를 잘 해준다.
- curl -XGET http://172.22.208.1:9200/classes?pretty

Index 생성

- Index는 리소스 자체의 고유한 url이기 때문에 PUT을 사용한다.
- `curl -XPUT http://172.22.208.1:9200/classes`

Index 삭제

- `curl -XDELETE http://172.22.208.1:9200/classes`

Document 생성

- Index가 없어도 바로 생성 가능하다.
- 7.x 이상부터는 `_type`이 지원되지 않는다. `_type`이름으로 `_doc`을 고정적으로 사용한다.
- Content-Type을 명시해주면 알 수 없는 오류를 방지할 수 있다.

POST, PUT 모두 사용 가능하다.

- ID 1로 문서 생성/업데이트 (PUT)
 - `curl -X PUT "http://172.22.208.1:9200/classes/_doc/1" -H "Content-Type: application/json" -d '{"title":"Algorithm", "professor":"john"}'`
- 자동 ID 부여해서 문서 추가 (POST)
 - `curl -XPOST http://172.22.208.1:9200/classes/_doc -H "Content-Type: application/json" -d '{"title":"Algorithm", "professor":"john"}'`
- 결과 확인
 - `curl -X GET http://172.22.208.1:9200/classes/_doc/1?pretty`
- Document json 데이터를 파일로 입력해주는 것도 가능하다.
 - `curl -XPOST http://172.22.208.1:9200/classes/_doc -H "Content-Type: application/json" -d @oneclass.json`
 - 현재 디렉토리의 oneclass.json 파일의 내용을 읽어서 전송.
 - 만약 PUT으로 전송할 때 ID가 겹치면 데이터를 덮어쓴다.