MongoDB Reference

작성자 : 양동호 (Rex), 20160125

https://www.mongodb.org/

< 설치하기 >

1. 위의 링크에서 download 를 받는다.

2. msi파일에는 다른 의존 소프트웨어들이 포함되어있으며, 자동으로 설치할수 있다. 설치해보자.

- 나는 D:₩Program Files₩MongoDB₩ 폴더에 설치하였으며,
  데이터가 저장될 디렉토리는 D:₩testMongo₩dbTest 로 설정할 것이다. (이거는 따로 폴더생성 하면 됨)

* 주의 : 인증 설정으로 "Secure Mode"로 실행하지 않고 공용 네트워크에 mongod.exe 표시하지 말 것. MongoDB를 신뢰할 환경에서 실행되도록 설계되었고, 데이터베이스는 기본적으로 "Secure Mode"를 사용하지 않음.

3. MongoDB 환경설정 하기

(0) 윈도우의 환경변수 path 를 잡아주는데, MongoDB 설치디렉토리의 bin 폴더까지 잡아준다.

(1) DB의 모든 데이터가 저장될 디렉토리를 만들기
아무 폴더나 들어가서 Shift + 우클릭, "여기서 명령창 열기" 를 한 뒤,
`md ₩data₩db` 를 타이핑한다, 그럼 해당 드라이브 root에 data₩db₩ 폴더가 만들어지는데,
그 상태에서 D:₩Program Files₩MongoDB₩Server₩3.2₩bin₩mongod.exe 파일을 실행시키면, db폴더에 파일을 넣는다.

(1)번을 진행했다면, (2)-(3)-(4) 는 전혀 진행할 필요 없다.
(1)번까지가 default설정이고 나머지는 커스터마이징 부분임

(2) 위의 경로는 MongoDB의 default data directory path인데, 그것을 변경하고자 한다면,
cmd창에서 `mongod --dbpath "D:₩testMongo₩dbTest"` 를 타이핑한다.

(3) 데이터를 저장할 폴더에 mongod.cfg 파일(.conf파일을 만들어도 됨)을 메모장으로 만들고
아래의 내용을 넣은 뒤 저장한다. YAML 형식이라고 한다. (tab사용불가, space 사용)

```
systemLog:
    destination: file
    path: D:₩testMongo₩log₩mongod.log
storage:
    dbPath: D:₩testMongo₩dbTest
```

(4) 아래의 명령어를 입력하면, cmd창에서 `net start mongodb` 명령어로 간편하게 서버를 실행할 수 있게 된다.
`mongod --config "D:₩testMongo₩mongod.cfg" --install`
끌 때에는 `net stop mongodb` 라고 타이핑하면 된다.

* 명령 프롬프트(cmd)를 반드시 '관리자 권한으로 실행' 해야 한다.

아래의 명령어로도 install이 가능하다. (왜 2가지 방법이 있는지는 모르겠다. 아래 것이 좀더 수동적인 방법임)

```
sc.exe create MongoDB binPath= "D:₩Program Files₩MongoDB₩Server₩3.2₩bin₩mongod.exe --service --config=₩"D:₩testMongo₩mongod.cfg₩"" DisplayName= "MongoDB" start= "auto"
```

4. 브라우저의 주소창에 http://localhost:27017/ 를 입력한다. 접속이 되면 성공!
(default설정은 여기까지만 하면 됨)

5. 만약, 현재의 경로가 마음에 들지 않거나, 경로를 바꾸고 싶다면, 아래의 명령어를 입력하고 다시 install 해준다.
`mongod --remove` 또는 `sc.exe delete MongoDB`

< MongoDB 실행하기 >

(1) cmd창을 관리자권한으로 열고, `cd D:` (설치드라이브)로 변경한다음에 `mongod` 를 타이핑한다.

(2) cmd창을 관리자권한으로 열고, `mongo` 를 타이핑한다. 그냥 mongo만 입력하면 localhost:27017로 접속함
mongod.exe → 서버 // mongo.exe → 클라이언트(shell)
관련 명령어는 (https://docs.mongodb.org/manual/reference/program/mongo/) 링크를 참조하자.
* mongo --port 9999 만 입력하면 127.0.0.1:9999/test 로 들어가진다.
mongo --port 9999 admin -u root -p <root 계정 암호> 를 입력하면 127.0.0.1:9999/admin 으로 들어간다.
mongo --port 9999 mongoDBtest 를 입력해야 127.0.0.1:9999/mongoDBtest 로 들어갈 수 있다.
어떻게 들어갔던지 관리자권한을 이용하려면, db.createUser({}) 를 해줘야한다. (default 설정에는 필요x)

(3) DB의 이름 설정하기 : `use DBname` 나는 use mongoDBtest 를 입력했다. 그러면 switch가 된다. (해당 DB 로 접속)

< MongoDB의 mongo.exe(shell) 에서 기본적으로 CRUD 를 해보기 >

- 기존의 RDBMS와 MongoDB의 용어 및 사용법 비교하기 (https://docs.mongodb.org/manual/reference/sql-comparison/)

- 기본적으로 db 인스턴스를 사용하며, db 인스턴스는 "데이터베이스 그 자체" 이다.
- db.collectionName → db객체 내부에는 컬렉션(테이블) 들이 field로 존재한다.
* 여기서 매우 중요한사항! Javascript 의 Object 특성상 . 찍고 필드명을 쓰면, 존재하지 않을경우 생성한다는 점!
➔ 이 말은, 새로운 컬렉션이 필요할 경우, 언제든지 점 찍고 생성하면 된다는 것이다!
- db.collectionName.메소드() → 컬렉션(테이블) 객체에는 CRUD를 가능케 해주는 메소드가 존재한다.

```
db.컬렉션.find({});
db.컬렉션.findOne({}); → 최상단 1개의 row만 반환? 아니면 반드시 1개만 반환되는 select구문이어야 하나?
db.컬렉션.insert({});
db.컬렉션.insertMany([{}, {}, {}]);
db.컬렉션.update({});
db.컬렉션.remove({});
```

이 외에 모든 메소드의 정보는, 아래의 링크에 모두 있다.
➔ https://docs.mongodb.org/manual/reference/method/

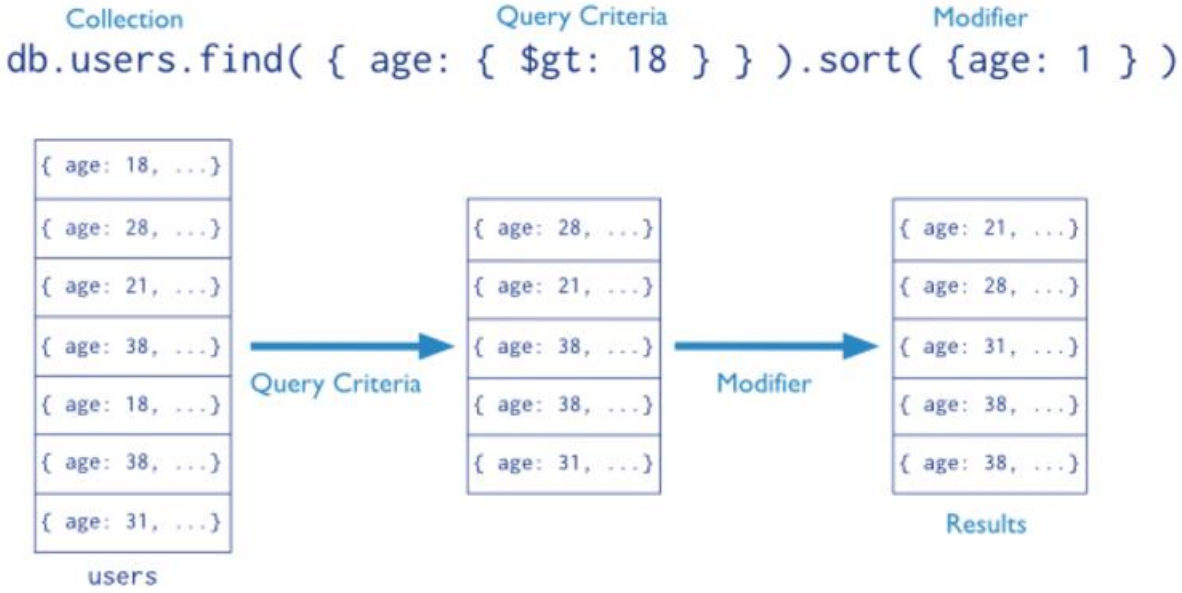
- 특별하게 알아둘 것은, 몽고DB에는 Primary Key가 없다는 것이다.
그것은 해당 row를 insert() 하는 시점에서 _id 라는 컬럼으로 자동으로 정의되는데,
랜덤한 해시 값으로 설정된다. (예를들면, 53d98f133bb604791249ca99 와 같이.)

- 기본적으로 MongoDB는 데이터의 조작을 Javascript Object Notation (JSON)을 사용한다. key: value

```
{
  name: "sue",                    ← field: value
  age: 26,                        ← field: value
  status: "A",                    ← field: value
  groups: [ "news", "sports" ]    ← field: value
}
```

- 컬렉션이란, 관계형 데이터베이스 (RDBMS) 의 테이블과 유사하다.

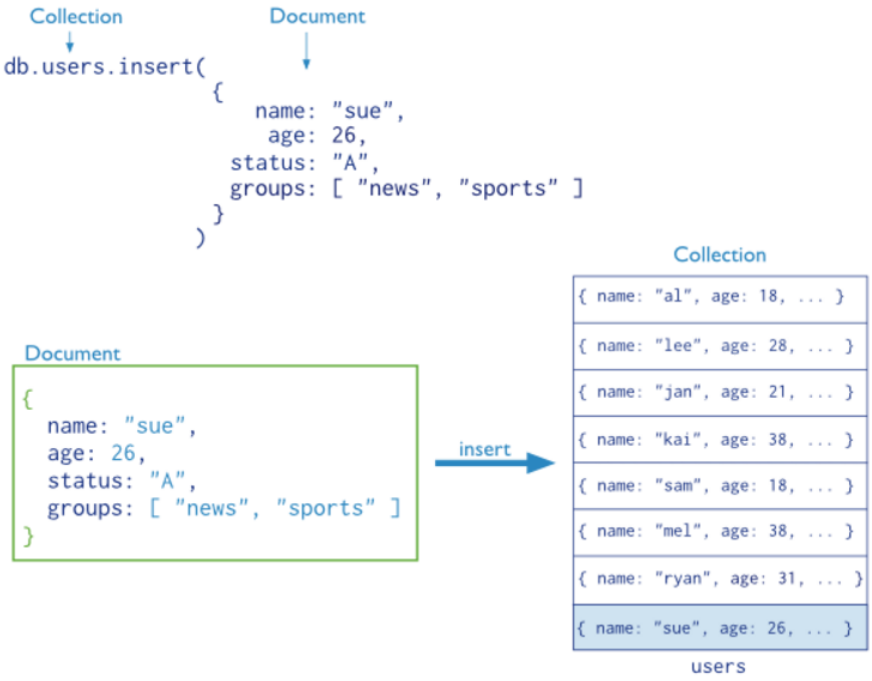
- 쿼리 또한, 기존의 RDBMS에서 사용하던 것 처럼, document(row)를 식별하는 **기준과 조건**을 지정한다.



- 자세한 내용은 그림이 모든 것을 말해주므로… 그림만 넣겠다.

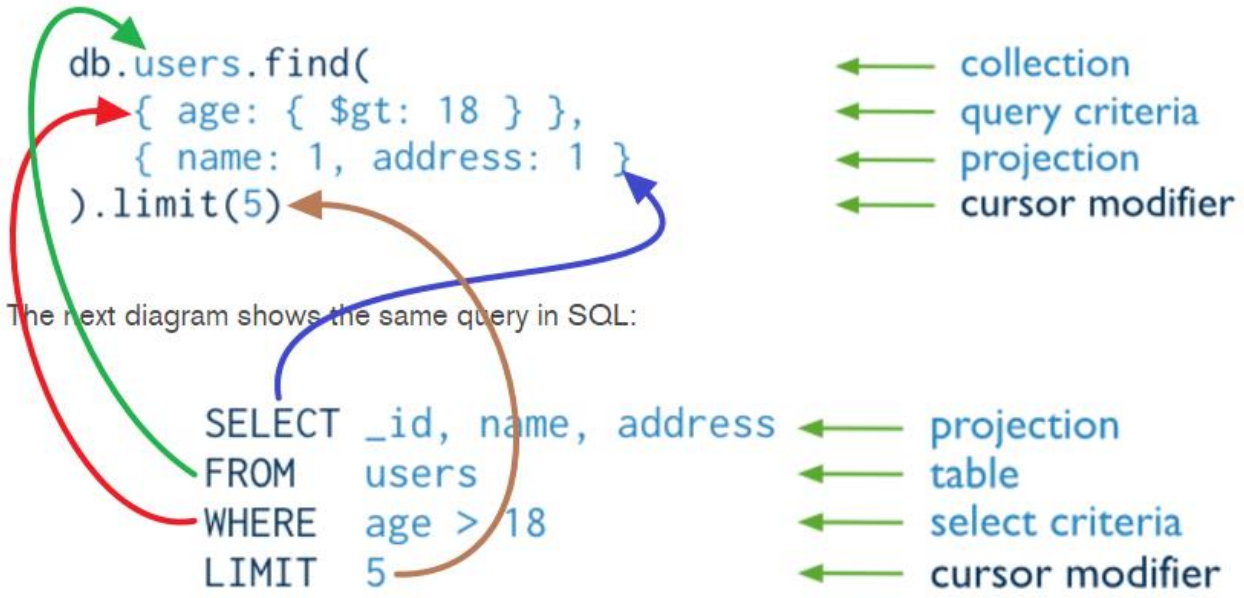
https://docs.mongodb.org/manual/core/crud-introduction/

- insert(Create) 하는 장면

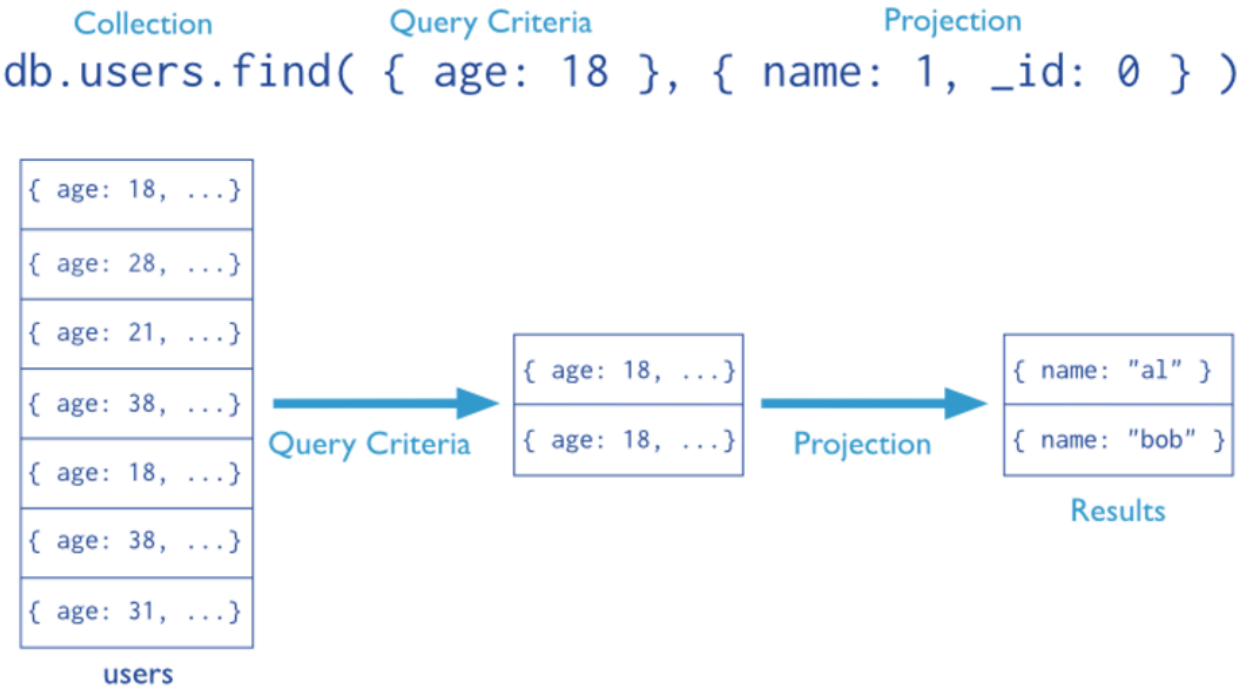


- find(select, Read) 하는 장면

The following diagram highlights the components of a MongoDB query operation:



The next diagram shows the same query in SQL:



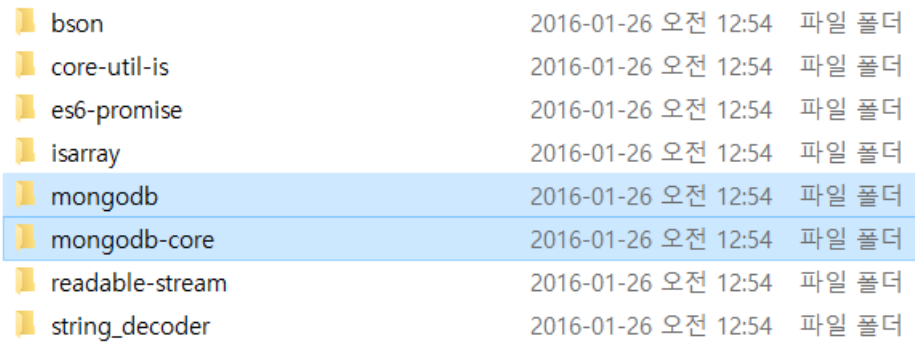
이 외에 모든 메소드의 정보는, 아래의 링크에 모두 있다.

➔ https://docs.mongodb.org/manual/reference/method/

< NodeJS 의 서버와 연동하기 >

* NodeJS 에서 구동중인 expressJS 서버를 기준으로 설명한다. (20160126 기준)
  express 모듈버전 : 4.13.1
  mongodb 모듈버전 : 2.1.4

1. 구동중인 expressJS 서버의 디렉토리에서 cmd창을 열고 `npm install mongodb --save` 를 타이핑한다.
   그러면 node_modules 폴더에는 mongodb 외에 여러 모듈들이 함께 설치되는데, 그 모듈들은 아래와 같다.



2. app.js 에는 반드시 아래의 코드가 있어야한다.

< 최상단에 넣는 코드 >

```
var MongoClient = require("mongodb").MongoClient  //몽고DB
  , mongoURL = "mongodb://localhost:27017/mongoDBtest";  //몽고DB의 서버URL과 DB의 주소
```

3. NodeJS + Express 서버에서 CRUD 로직을 작성하기 (참고 : https://docs.mongodb.org/getting-started/node/)

(1) user에 대한 Create (insert)

```
MongoClient.connect(mongoURL, function(err, db) {
  db.collection('user').insert({
    id: req.body.id,
    pw: req.body.pw,
  }, function(err, result) {
    db.close();
    res.send(result);
  });
});
```

(2) user에 대한 Read (find)

```
MongoClient.connect(mongoURL, function(err, db) {
  db.collection('user').findOne(  //1개만 찾음
    { id: req.body.id },  //where 조건
    function(err, document) {
      db.close();
      res.send(document);
    }
  );
});
```

(3) user에 대한 Update (update)

```
MongoClient.connect(mongoURL, function(err, db) {
  db.collection("user").updateOne(
    { id: req.body.id },  //where조건
    { $set: {
      pw: req.body.pw,
      email: req.body.email,
      like: req.body.like
    }},
    function(err, data) {
      db.close();
      res.send(data);
    }
  );
});
```

(4) user에 대한 Delete (delete)

```
MongoClient.connect(mongoURL, function(err, db) {
  db.collection("user").deleteOne(
    { id: req.body.id },  //where조건
    function(err, data) {
      db.close();
      res.send(data);
    }
  );
});
```

- 기본적으로, NoSQL의 메소드를 작성하고, 콜백function을 마지막 인자로 집어넣어주면 된다.
 - db.collectionName.메소드() → 컬렉션(테이블) 객체에는 CRUD를 가능케 해주는 메소드가 존재한다.
 db.컬렉션.find({}).toArray(function(err, result) { });
 db.컬렉션.findOne({}, function(err, result) {}); → 최상단 1개의 row만 반환? 아니면 반드시 1개만 반환되는 select구문이어야 하나?
 → ★그냥 find일경우에는 toArray() 를 해줘야하고, findOne() 이면 그냥 써도된다.
 db.컬렉션.insert({}, function(err, result) {});
 db.컬렉션.insertOne({}, function(err, result) {});
 db.컬렉션.insertMany([{}, {}, {}], function(err, result) {});
 db.컬렉션.updateOne({}, function(err, result) {});
 db.컬렉션.updateMany({}, function(err, result) {});
 db.컬렉션.replaceOne({}, function(err, result) {});
 db.컬렉션.deleteOne({}, function(err, result) {});
 db.컬렉션.deleteMany({}, function(err, result) {});
 db.컬렉션.drop({}, function(err, result) {});
 db.컬렉션.remove({}, function(err, result) {});