

1. mongoDB 설치 및 환경 구축

1.1 mongodb 설치(Ubunto Linux)

설치 가이드

- ubuntu 18.04
- mongodb Community 4.4
<https://docs.mongodb.com/v4.4/tutorial/install-mongodb-on-ubuntu/>
- Supported Platforms(참고)
<https://docs.mongodb.com/manual/administration/production-notes/#std-label-prod-notes-supported-platforms-ARM64>
- 설치 가이드 한글 사이트 정리 잘될 곳
<https://coterie.tistory.com/20>

1.2 mongodb 설치 방법 (윈도우 환경)

- <https://www.mongodb.com/download-center/community> 이동 후, .msi 파일 다운로드 및 설치
- 참고 블로그: <https://javapro.tistory.com/64>

1.3 MongoDB 실행 및 shell 명령

- MongoDB 실행

```
$ sudo systemctl start mongod
```
- MongoDB 실행 상태 확인

```
$ sudo systemctl status mongod
```
- MongoDB 재실행(conf 파일 수정 후 재실행 해야함)

```
$ sudo service mongod restart
```
- 시스템 booting 시 자동 실행되도록 하기

```
$ sudo systemctl enable mongod
```
- 포트확인 (몽고db 기본포트(27017))

```
$ sudo netstat -plntu | grep mongod
```

mongo shell 실행 사용자 추가

```
$ mongo
> use admin
> db.createUser(
  {  user: "rapa00",
```

```

        pwd: "1234",
        roles: [ "userAdminAnyDatabase",
                  "dbAdminAnyDatabase",
                  "readWriteAnyDatabase" ]
    }
)

```

시큐리티 인증 활성화

\$ sudo nano /etc/mongod.conf

- mongod.conf 파일에 다음 내용 추가하기
 - 들여쓰기 꼭 해야함.

```

security:
  authorization: enabled

```

- mongod.conf 파일 수정 후 mongod 재수행

\$ sudo service mongod restart

2. mongoDB를 위한 GUI 클라이언트

- mongodb의 데이터 구성
 - db, collection으로 구성
 - 데이터는 각 collection에 document 형식(python dictionary)으로 저장 됨
 - collection들의 논리적인 집합이 database

2-1 Robo3T 리눅스에 설치

Robo3T 설치

- 다운로드
<https://robomongo.org/download>

[설치방법 가이드]

압축 풀기

```
$ tar -xvzf -xvzf robo3t-1.4.4-linux-x86_64-e6ac9ec.tar.gz
```

Make a new folder in usr/local/bin from the package

```
$ sudo mkdir /usr/local/bin/robo3t
```

Move the extracted package to usr/local/bin

```
$ sudo mv robo3t-1.2.1-linux-x86_64-3e50a65/*
/usr/local/bin/robo3t
```

Change directory to `cd /usr/local/bin/robo3t/bin`

Now, We need to give permission to newly created directory using `chmod`

```
$ sudo chmod +x robo3t ./robo3t
```

Now we can run Robo3t `./robo3t`

We can download the icon for Robo3t from and put it here as we will need to make desktop icon later

홈페이지에서 icon.png 다운 받기

For example save it on /bin with name icon.png

`/usr/local/bin/robo3t/bin/icon.png`

```
$ mv icon.png /usr/local/bin/robo3t/bin
```

- 데스크탑 아이콘 추가하기

To make desktop icon for Robo3t, we can make a file in `usr/share/applications`

```
$ sudo nano /usr/share/applications/robo3t.desktop
```

Paste these there and save

```
[Desktop Entry]
Encoding=UTF-8
Type=Application
Name=Robo3t
Icon=/usr/local/bin/robo3t/bin/icon.png
Exec="/usr/local/bin/robo3t/bin/robo3t"
Comment=Robo3t
Categories=Development;
Terminal=false
StartupNotify=true
```

Now, we can find the icon in application launcher menu by search for robo3t

</pre>

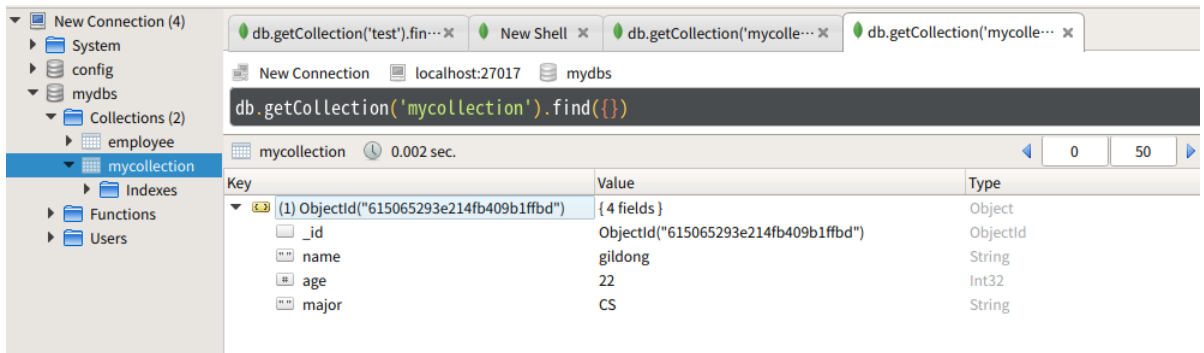
- robo3T 설치 가이드 참고 사이트

<https://gist.github.com/abdallahokasha/37911a64ad289487387e2d1a1446>

```

{
  "name" : "gildong",
  "age" : 22,
  "major" : "CS"
}
{
  "name" : "sunhee",
  "age" : 24,
  "major" : "Japanense"
}

```

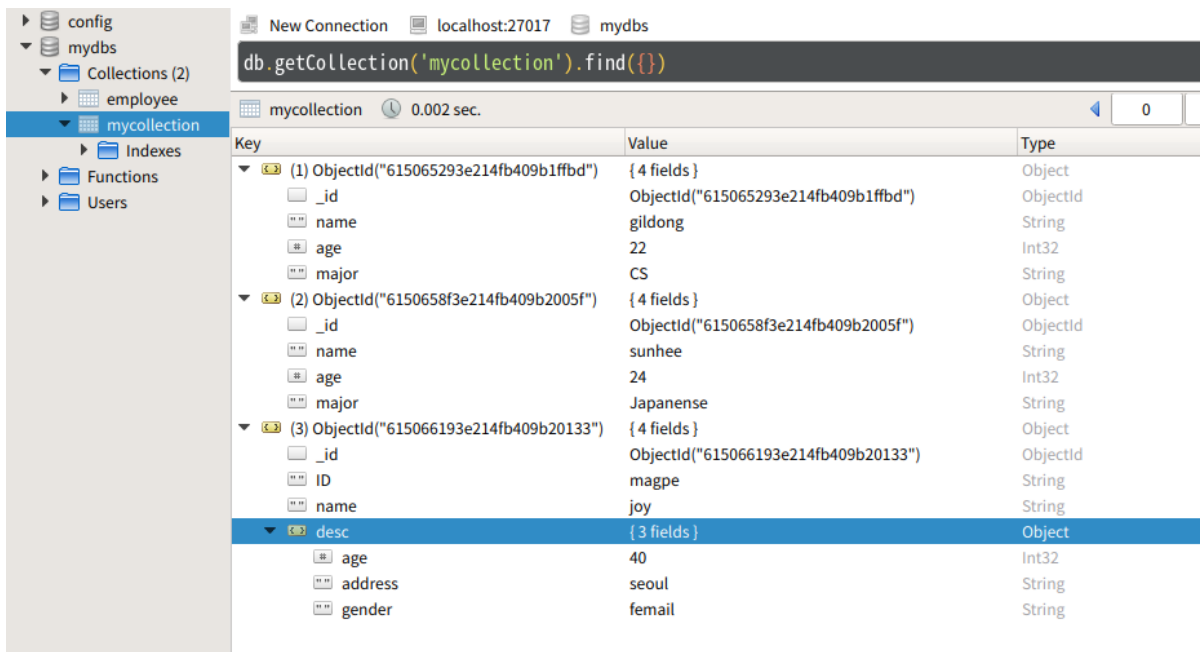


- Insert Document (different JSON fields) & Click View Documents in test collection

```

{
  "name" : "김철수",
  "age" : 54,
  "minor" : "CS",
  "nickname" : "joy"
}

```



2.2 Robomongo에서 커멘드 입력해보기 (실습)

- Right-click (Server) -> Open Shell
- show dbs - 전체 데이터베이스 열람
- use [DB 이름] - 데이터베이스 선택 (없으면 데이터베이스 생성)

- 예) use mydbs - mydbs 데이터베이스 선택
- show collections - 선택된 데이터베이스의 컬렉션 열람
- db.[Collection 이름].함수() 로 해당 컬렉션의 데이터 열람 및 조작
 - 예) db.employees.find() - employees 컬렉션에서 전체 Document 검색
- 데이터베이스 선택
 - use mydbs
- 데이터베이스 현황 확인
 - db
 - db.stats()
- collection 생성 및 삭제
 - db.createCollection("employees", {capped:true, size:10000})
 - Right-click (server) -> Click Refresh -> Go to mydbs -> Go to Collections -> Check employees
 - capped:true 최초 제한된 크기로 생성된 공간에서만 데이터를 저장하는 설정 (고성능, 저장공간차면 기존 공간 재사용, 일정시간만 저장하는 로그에 적합)
 - db.employees.isCapped()
 - db.employees.drop()
 - db.createCollection("employees")
 - db.employees.isCapped()
 - db.employees.drop()
 - db.createCollection("employees", {capped:true, size:10000})
- collection 확인
 - show collections
 - db.employees.stats()
- collection 이름 변경
 - db.employees.renameCollection("emp")
- collection 삭제
 - db.emp.drop()

2.3 SQL과 간단 비교 - mongodb collection 생성/변경

- collection 생성 (원하는 타임으로 데이터를 바로 넣으면 됨)
 - PRIMARY KEY를 위한 별도 컬럼 만들 필요 없음.
 - mongodb는 collection에서 _id가 각 Document마다 자동생성되어 primary key 역할을 함
 - 컬럼마다 데이터 타입을 정할 필요 없음 ("컬럼명": 컬럼값 이 기본 형태임)

```
CREATE TABLE people (  
  id MEDIUMINT NOT NULL  
    AUTO_INCREMENT,  
  user_id Varchar(30),  
  age Number,  
  status char(1),  
  PRIMARY KEY (id)  
)
```

Implicitly created on first `insertOne()` or `insertMany()` operation. The primary key `_id` is automatically added if `_id` field is not specified.

```
db.people.insertOne( {  
  user_id: "abc123",  
  age: 55,  
  status: "A"  
} )
```

However, you can also explicitly create a collection:

```
db.createCollection("people")
```

- collection 구조 변경 (기존 Document에 컬럼 추가/삭제 필요없을 시는 새로운 Document에만 필요한 컬럼을 추가 또는 삭제해서 넣으면됨)
 - ALTER TABLE은 기본적으로 collection에서는 필요 없음
 - 일부 기존 Document에도 컬럼과 컬럼값을 넣거나 삭제해야 한다면 다음과 같은 형태로는 가능함
- 기존 Document에도 컬럼과 컬럼값 추가시
 - SQL: ALTER TABLE people ADD COLUMN join_date DATETIME
 - mongodb: `db.people.updateMany({}, { $set: { join_date: new Date() } })`
- 기존 Document에도 컬럼과 컬럼값 삭제시
 - SQL: ALTER TABLE people DROP COLUMN join_date
 - mongodb: `db.people.updateMany({}, { $unset: { "join_date": "" } })`