**MongoDb User 만들기**

# 목차

[1. 배경지식 2](file:///C:\Users\wkdtn\Downloads\01%20Spring%20boot%20and%20JPA%20(1).docx#_Toc483936755)

[1) User 2](file:///C:\Users\wkdtn\Downloads\01%20Spring%20boot%20and%20JPA%20(1).docx#_Toc483936756)

[2. User 생성과 권한 4](file:///C:\Users\wkdtn\Downloads\01%20Spring%20boot%20and%20JPA%20(1).docx#_Toc483936760)

[1) User 생성 4](file:///C:\Users\wkdtn\Downloads\01%20Spring%20boot%20and%20JPA%20(1).docx#_Toc483936761)

[2) User 권한 6](file:///C:\Users\wkdtn\Downloads\01%20Spring%20boot%20and%20JPA%20(1).docx#_Toc483936762)

[3) User 권한 접속 7](file:///C:\Users\wkdtn\Downloads\01%20Spring%20boot%20and%20JPA%20(1).docx#_Toc483936763)

[4) User 제거 7](file:///C:\Users\wkdtn\Downloads\01%20Spring%20boot%20and%20JPA%20(1).docx#_Toc483936763)

[5) User Update 7](file:///C:\Users\wkdtn\Downloads\01%20Spring%20boot%20and%20JPA%20(1).docx#_Toc483936763)

1. 배경 지식

# User

Spring에서 디비 연동을 할 떄 Properties에 데이터베이스 설정을 해주었다. 그 설정에 mySql User 설정을 해주므로써 그 user 권한을 통하여 데이터베이스에 연결이 되어 crud를 진행할 수 있었다. 이번 강의노트에서는 MongoDb에 관한 User 설정을 하도록 하겠습니다. 우리가 Relational Database에서 쓰인 User와 MongoDB에서 쓰는 User의 개념이 색달라진다. 또한 Relational Database와 MongoDB의 보안성의 차이를 본다면 현저히 Relational Database가 월등한 편이지만 MongoDB에서는 NoSQL의 보안을 향상시키기 위해서는 다행히 User 개념이 존재한다. 초기에 MongoDB를 설치할 때에는 Admin User를 생성하지 않고 바로 넘어가서 mongod 명령어로 그냥 접속을 할 때에는 NoSQL Database의 보안성을 보장하지 않았었다. 그렇기에 연동설정에서 User설정은 더욱 중요하다

1. User 생성과 권한

# User 생성

# 

명령어 : use admin 입력 후

# 명령어

|  |  |
| --- | --- |
| 1  2  3  4  5 | db.createUser( { user: "<username>",  pwd: "<password>",  roles: [ "userAdminAnyDatabase",  "dbAdminAnyDatabase",  "readWriteAnyDatabase" ] } ) |

# 

치면 관리자 계정의 user가 생성된다.

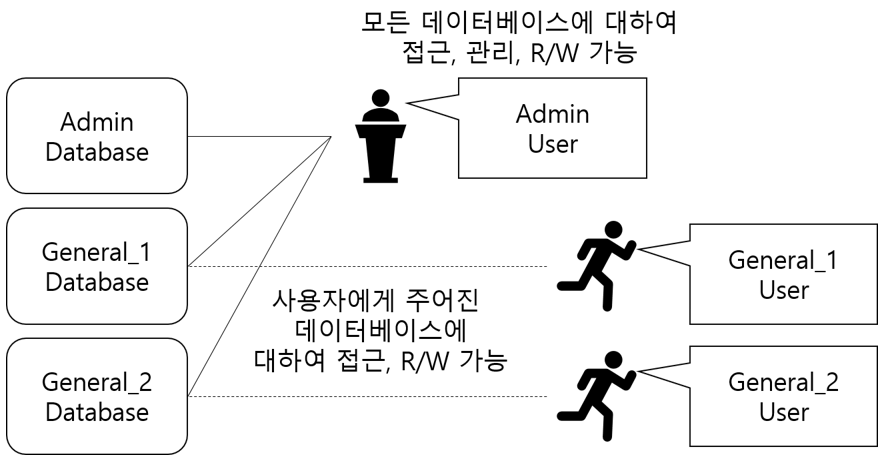


# 명령어

|  |  |
| --- | --- |
| 1 | Show users |

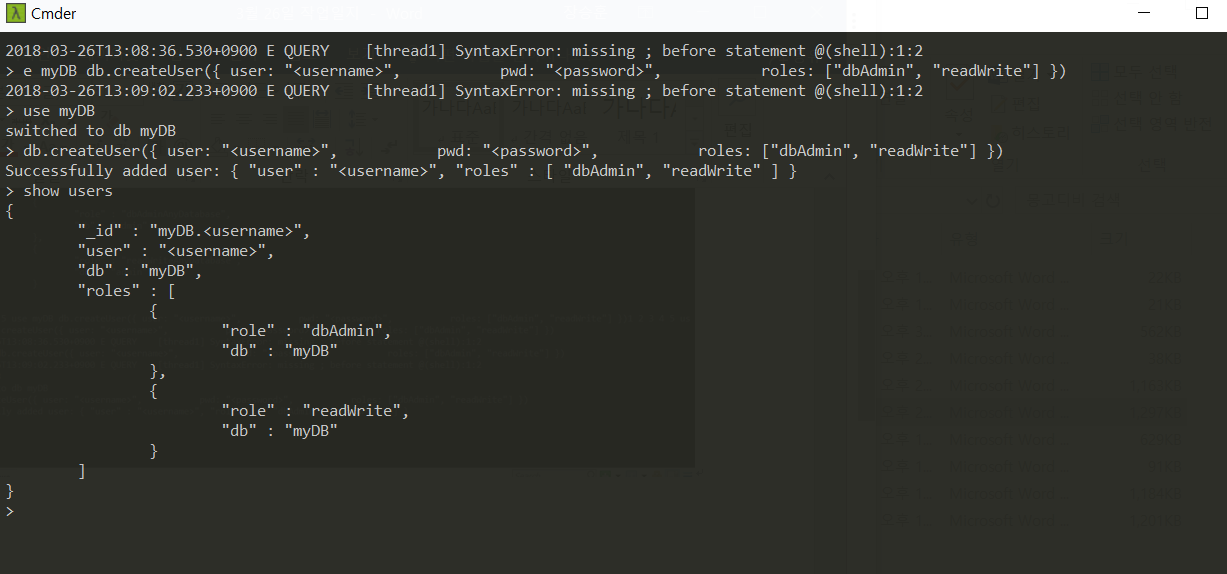
Show users를 치면 확인이 가능하다 .

다음은 사용자 계정 및 권한 추가이다. 방금은 관리자용 계정을 생성하였다. 사용자 계정은 한 데이터베이스에서만 접근이 가능하다. 관리자용 계정은 다른 데이터베이스에서 접근이 가능하다.



위 그림을 보면 이해가 쉽다.

# 사용자 계정 및 권한 추가



명령어 : use myDB입력 후

# 명령어

|  |  |
| --- | --- |
| 1  2  3  4  5 | db.createUser({ user: "<username>",  pwd: "<password>",  roles: ["dbAdmin", "readWrite"] }) |

로 사용자 계정을 추가했다.

# User 권한

1. MongoDB에서 현재 제공하고 있는 권한 목록들

* Document에 대해 쓰는 권한

read : Document들을 오로지 읽을 수만 있는 권한만 주어짐.

readWrite : Document들을 읽을 수 있는 동시에 삽입, 삭제, 수정이 가능하다.

* 하나의 Database에 대해 쓰는 권한

dbAdmin : 현재 데이터베이스에 대한 Schema 작업과 Index 생성과 통계 관련 작업을 실행 할 수 있다. 그렇지만 타 User들에게 권한을 줄 수 있지는 않다. 그런데 이 작업만 주었다고 하더라도 Document들을 읽고 쓸 수는 없는 일이다.

dbOwner : 이는 데이터베이스에 대한 모든 관리를 할 수 있도록 하는 권한이다. 이 권한은 dbAdmin, readWrite와 결합한 개념과 마찬가지인 셈이다.

userAdmin : 각기 다른 일반 User들에게 권한을 주거나 걷어 들일 수 있도록 한다.

* 모든 Database에 대해 쓰는 권한

여기에 있는 모든 권한은 local, config Database에 대해서는 해당되지 않는다.

readAnyDatabase : 말 그대로 모든 Database에 있는 Document들을 읽을 수만 있는 권한이다.

readWriteAnyDatabase : 모든 Database에 있는 Document들을 읽고 쓸 수 있는 권한이다.

userAdminAnyDatabase : 모든 Database에 현존하는 User들에 대해서 권한을 주고 뺏을 수 있는 권한이다.

dbAdminAnyDatabase : 모든 Database에 대해 Index를 생성하거나 통계 관련 작업을 할 수 있는 권한이다.

* Clustering 관리에 대한 권한

clusterAdmin : 클러스터 관리를 할 수 있는 User 중 최고봉. 이는 clusterManager, clusterMonitor, hostManager에 대한 권한을 줄 수 있고 데이터베이스를 삭제하는(dropDatabase()) 권한도 주어진다.

clusterManager : 클러스터에서 실시간으로 확인되는 모니터링을 관리할 수 있고 local, config 데이터베이스에 접근이 가능하다.

clusterMonitor : 모니터링을 관리할 수 있는 수준은 아니고 어떻게 돌아가는지에 대해 읽어 들일 수만 있다.

hostManager : 모니터링과 서버를 공급할 수 있도록 하는 권한이다.

* 백업과 복원 관련 권한

backup : 이는 현재 Database에 적재된 데이터들에 대해 mongodump를 이용해서 백업을 시켜줄 수 있는 권한이다.

restore : 이는 이전에 작성했던 Database의 시점을 설정하여 mongorestore를 토대로 가져와서 복원을 할 수 있는 권한이다.

Role에 들어가는 옵션들은

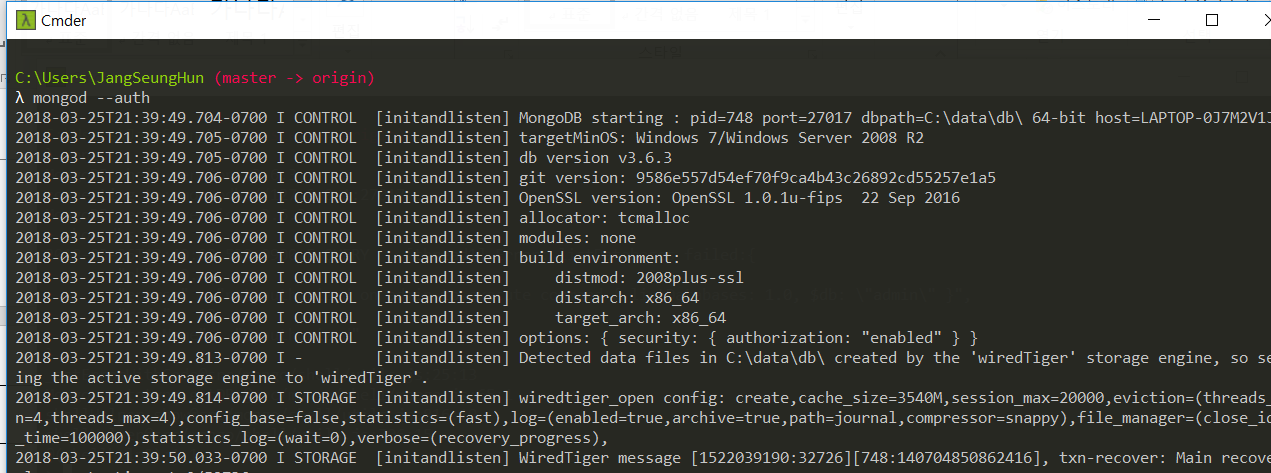
<https://docs.mongodb.com/manual/core/security-built-in-roles/>

위 사이트에 들어가면 알아 볼 수 있다

# User 권한 접속

자 그러면 방금 만든 계정으로 몽고 디비에 접속해 보겠다.

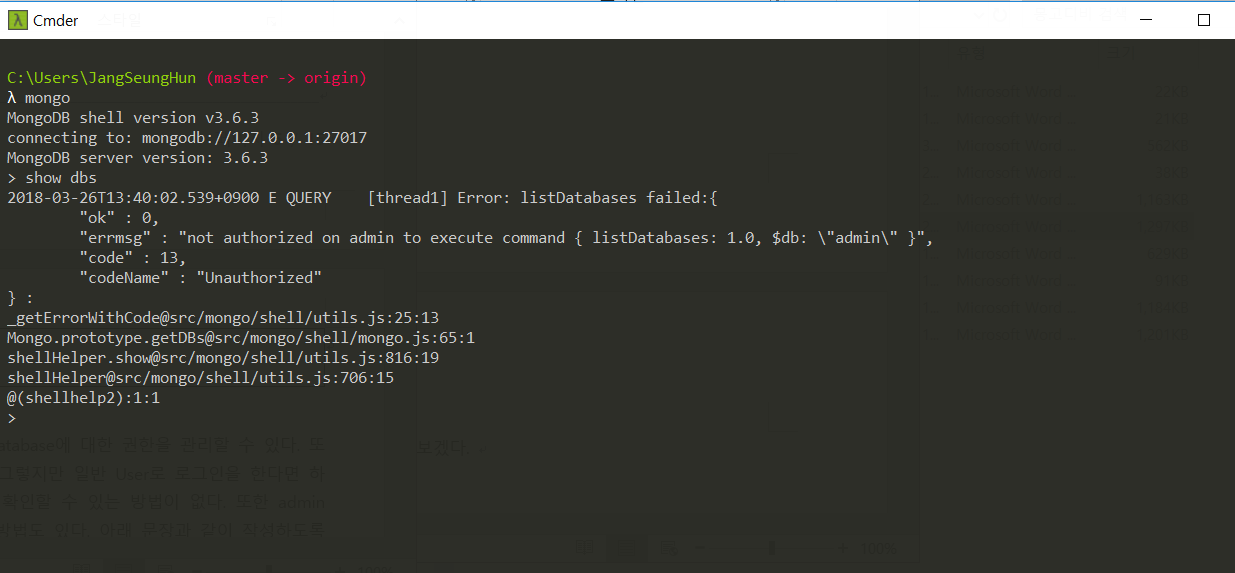
Ctrl + c 로 서버를 다 종료하고



# 명령어

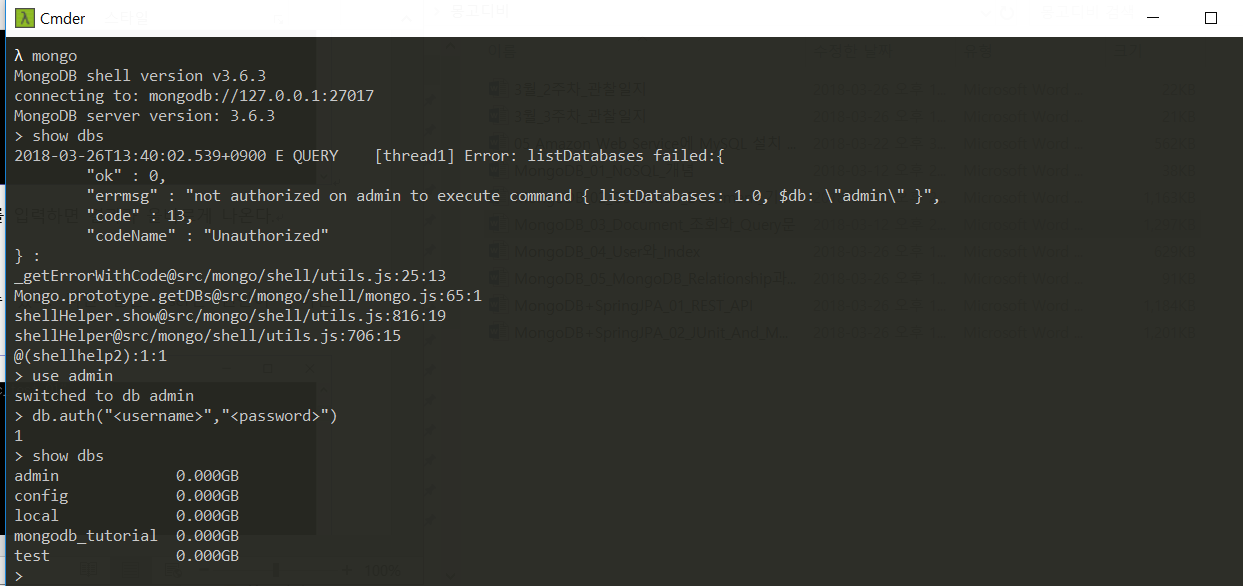
|  |  |
| --- | --- |
| 1 | mongod –auth |

명령어: mongod –auth로 다시 서버에 접속한다.



Mongo로 접속 show dbs하면 권한이 없다고 뜨는 걸 알 수 있다.

그러면 다시 권한이 있는 관리자 계정으로 접속



명령어 : use admin으로 접속 후

# 명령어

|  |  |
| --- | --- |
| 1 | db.auth(“<username>”,<password>”) |

로 들어가면 관리자 계정으로 접속되어서 show dbs를 했을 떄 정상 작동 되는 것을 확인 할 수 있다.

# User 제거

# 

# 명령어

|  |  |
| --- | --- |
| 1 | db.dropUser(“<username>”) |

명령어 db.dropUser(“<username>”)

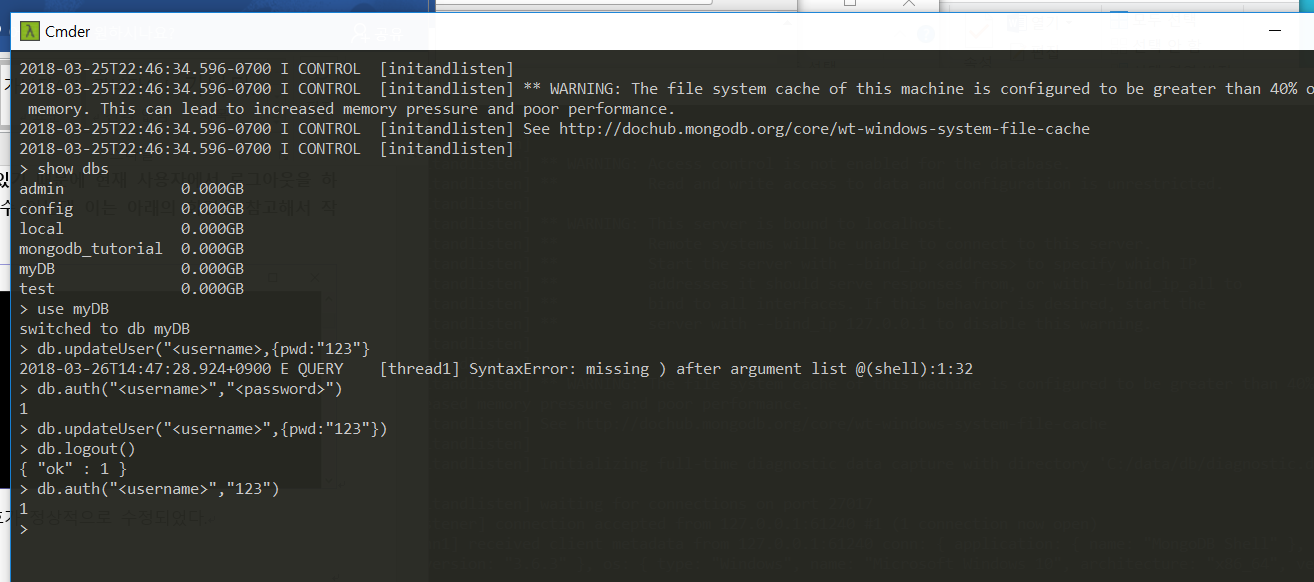
으로 계정 삭제가 가능하다 .

그리고 관리자 계정에서 사용자 계정을 삭제할 수 있지만 사용자 계정에 접속해서 관리자 계정을 삭제할 수는 없다.

방금 계정 삭제가 안된 분들은 계정 로그인을 잘 확인하고

오타를 확인하시길 바랍니다. 저도 계속 삭제가 안되서 계정 정보 확인해서 삭제 완료 했습니다.

# User Update



# 명령어

계정으로 접속된 상태에서

|  |  |
| --- | --- |
| 1  2  3  4 | : db.updateUser(“<username>”,{pwd:”123”})  db.logout()  db.auth(“<username>”,”123”) |

명령어 : db.updateUser(“<username>”,{pwd:”123”})

db.logout()

db.auth(“<username>”,”123”)

로 패스워드를 바꾼 것을 알 수 있다.