

MongoDB 실습

Database Programming



MongoDB 소개



MongoDB 설치



MongoDB 실행



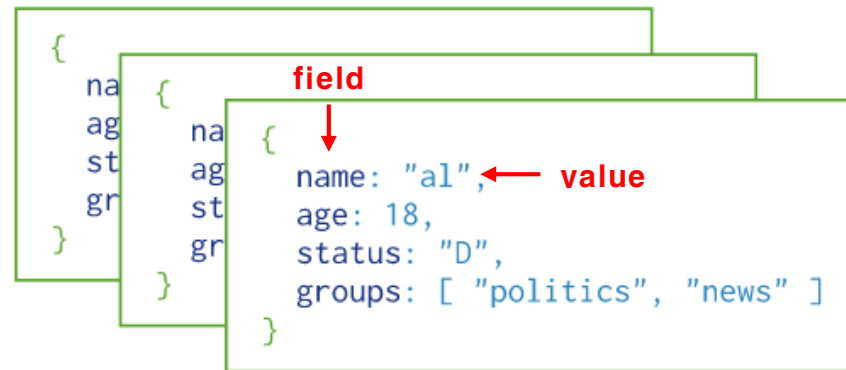
MongoDB 명령어



어플리케이션 프로그래밍

■ MongoDB

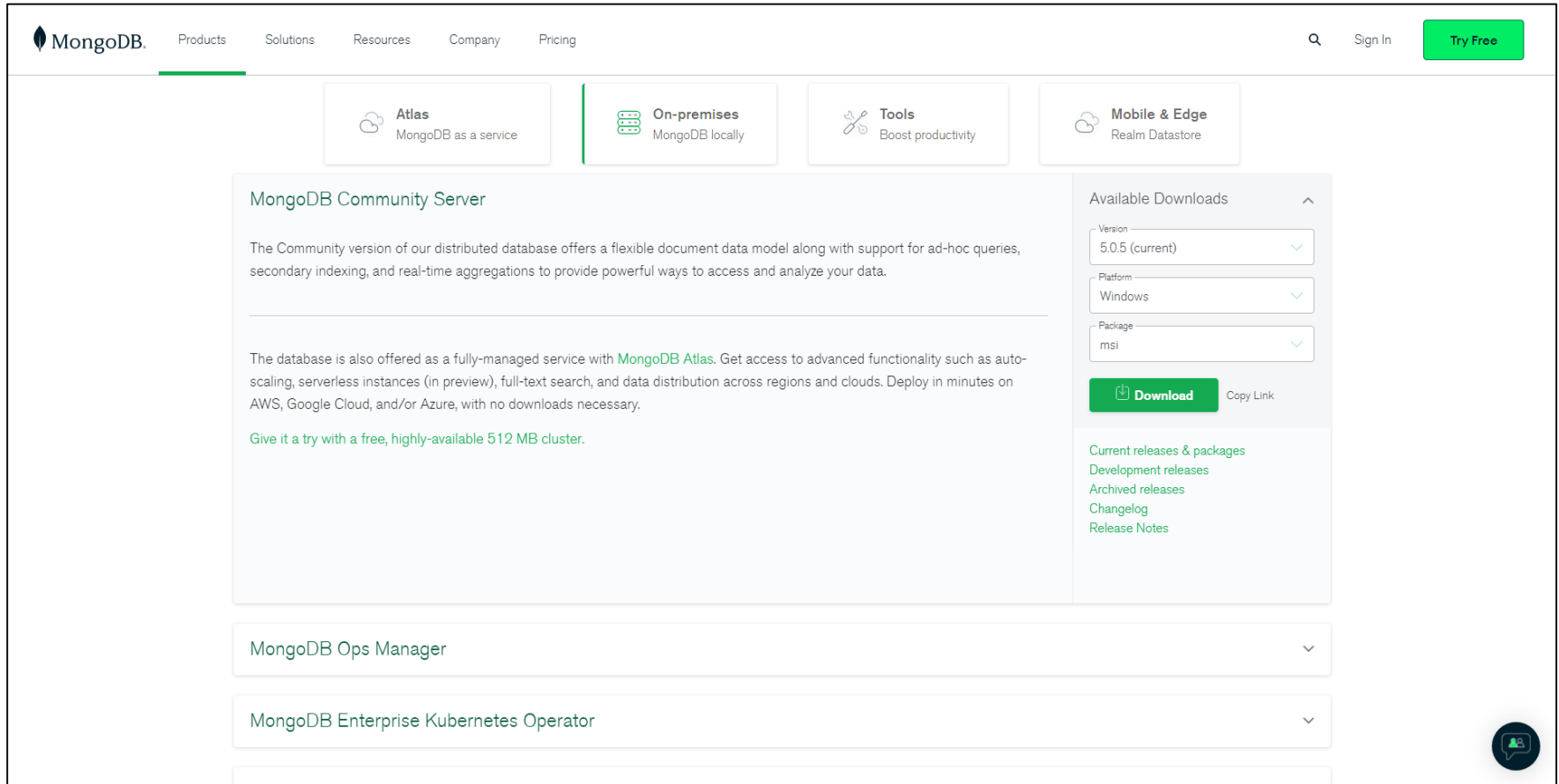
- 크로스 플랫폼 문서를 지향하는 NoSQL 데이터베이스 시스템
- 데이터를 필드(field)와 값(value)의 쌍으로 구성된 JSON 객체와 유사한 구조인 BSON(Binary JSON) 문서로 저장함
- 문서(데이터)를 컬렉션에 저장하고 데이터베이스에 하나 이상의 컬렉션을 저장함
- 무료 오픈 소스 소프트웨어



[그림] Collection 구조^[1]

[1] <https://docs.mongodb.com/manual/core/databases-and-collections/>

1) <https://www.mongodb.com/try/download/community> 접속



The screenshot shows the MongoDB website's download page for the Community Server. The navigation bar at the top includes the MongoDB logo, links for Products, Solutions, Resources, Company, and Pricing, a search icon, a Sign In link, and a green Try Free button. Below the navigation bar, there are four main product categories: Atlas (MongoDB as a service), On-premises (MongoDB locally), Tools (Boost productivity), and Mobile & Edge (Realm Datastore). The On-premises category is currently selected, indicated by a green vertical line. The main content area for the On-premises category is titled "MongoDB Community Server" and contains the following text: "The Community version of our distributed database offers a flexible document data model along with support for ad-hoc queries, secondary indexing, and real-time aggregations to provide powerful ways to access and analyze your data." Below this, it states: "The database is also offered as a fully-managed service with MongoDB Atlas. Get access to advanced functionality such as auto-scaling, serverless instances (in preview), full-text search, and data distribution across regions and clouds. Deploy in minutes on AWS, Google Cloud, and/or Azure, with no downloads necessary." A green link says: "Give it a try with a free, highly-available 512 MB cluster." To the right of the main content, there is a section titled "Available Downloads" with a dropdown menu for Version (5.0.5 (current)), Platform (Windows), and Package (msi). Below the dropdowns is a green Download button and a Copy Link button. Further down, there are links for Current releases & packages, Development releases, Archived releases, Changelog, and Release Notes. At the bottom of the page, there are two more product categories: MongoDB Ops Manager and MongoDB Enterprise Kubernetes Operator, each with a dropdown arrow. A small chat icon is visible in the bottom right corner.

MongoDB. Products Solutions Resources Company Pricing

Atlas
MongoDB as a service

On-premises
MongoDB locally

Tools
Boost productivity

Mobile & Edge
Realm Datastore

MongoDB Community Server

The Community version of our distributed database offers a flexible document data model along with support for ad-hoc queries, secondary indexing, and real-time aggregations to provide powerful ways to access and analyze your data.

The database is also offered as a fully-managed service with [MongoDB Atlas](#). Get access to advanced functionality such as auto-scaling, serverless instances (in preview), full-text search, and data distribution across regions and clouds. Deploy in minutes on AWS, Google Cloud, and/or Azure, with no downloads necessary.

[Give it a try with a free, highly-available 512 MB cluster.](#)

Available Downloads

Version
5.0.5 (current)

Platform
Windows

Package
msi

[Download](#) [Copy Link](#)

[Current releases & packages](#)
[Development releases](#)
[Archived releases](#)
[Changelog](#)
[Release Notes](#)

MongoDB Ops Manager

MongoDB Enterprise Kubernetes Operator

2) 5.0.5 버전과 msi 패키지 선택 후 다운로드 클릭 (Platform의 경우 자신의 OS에 맞게 선택)

MongoDB Community Server

The Community version of our distributed database offers a flexible document data model along with support for ad-hoc queries, secondary indexing, and real-time aggregations to provide powerful ways to access and analyze your data.

The database is also offered as a fully-managed service with [MongoDB Atlas](#). Get access to advanced functionality such as auto-scaling, serverless instances (in preview), full-text search, and data distribution across regions and clouds. Deploy in minutes on AWS, Google Cloud, and/or Azure, with no downloads necessary.

[Give it a try with a free, highly-available 512 MB cluster.](#)

Available Downloads

Version

5.0.5 (current) ✓

Platform

Windows ✓

Package

msi ✓

Download

Copy Link

Current releases & packages

Development releases

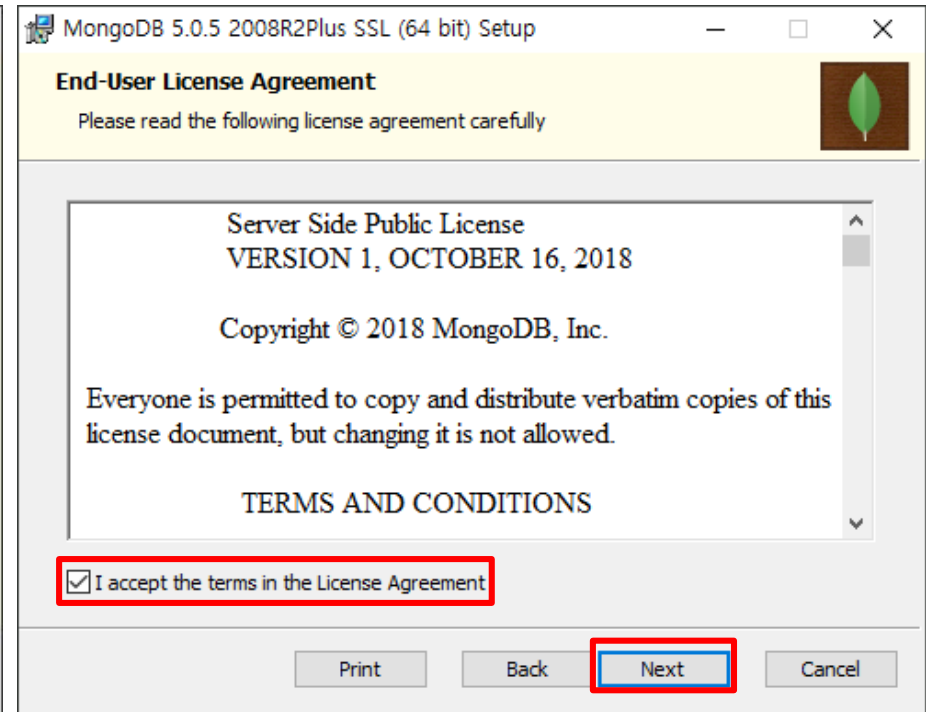
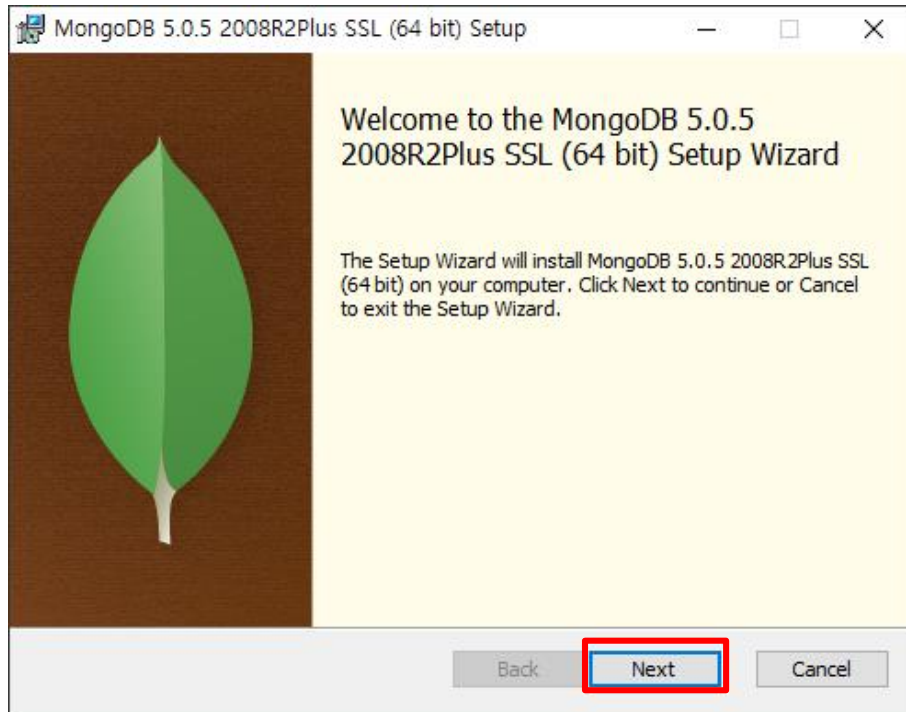
Archived releases

Changelog

Release Notes

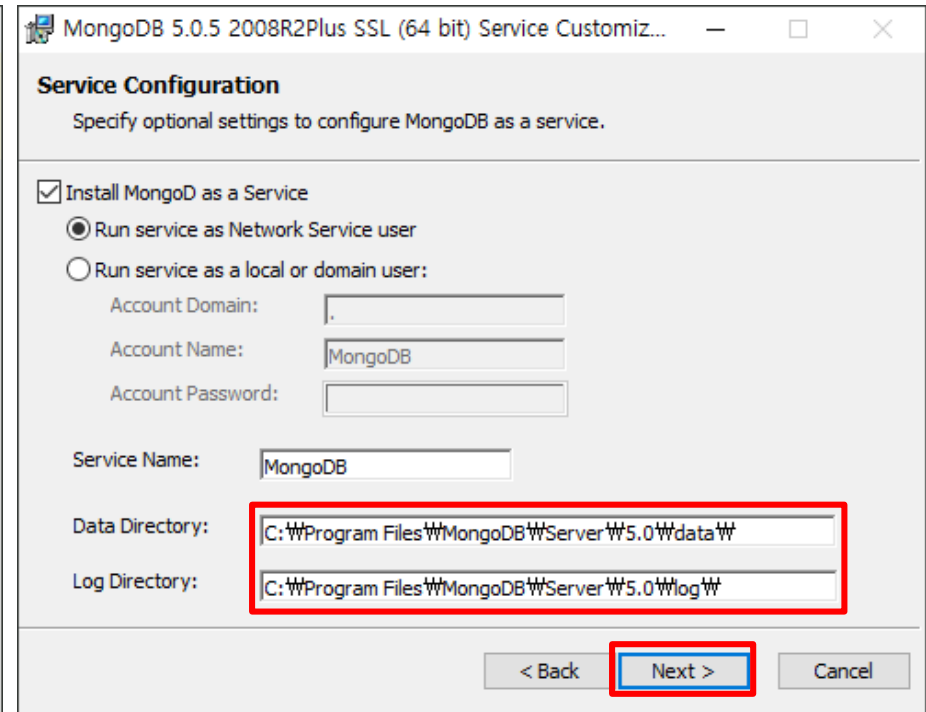
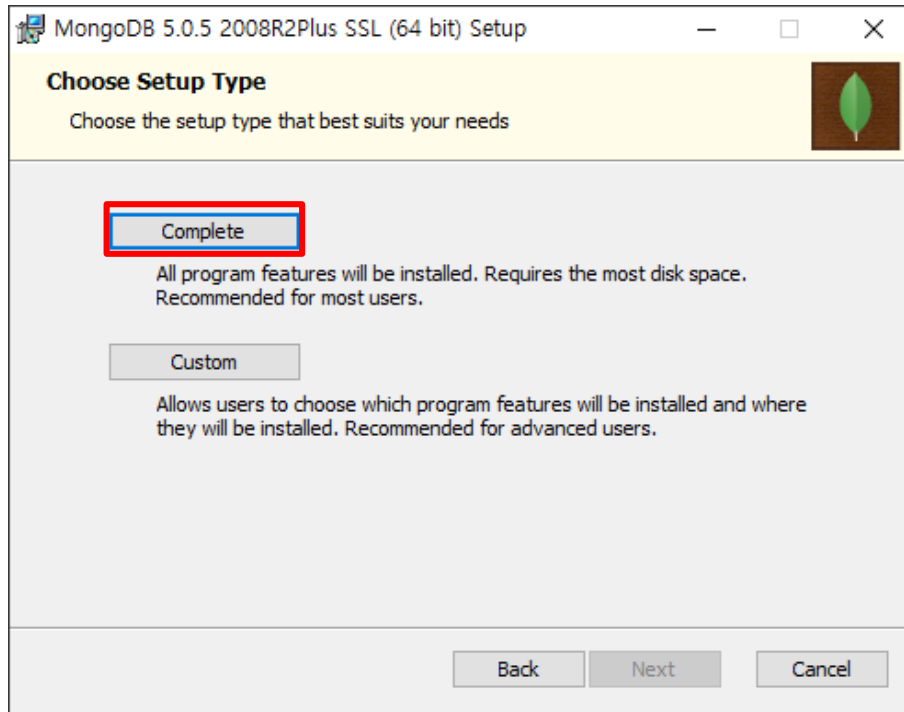
3) 다운로드 받은 설치 파일 실행 → Next 클릭

4) 라이선스 동의 → Next 클릭



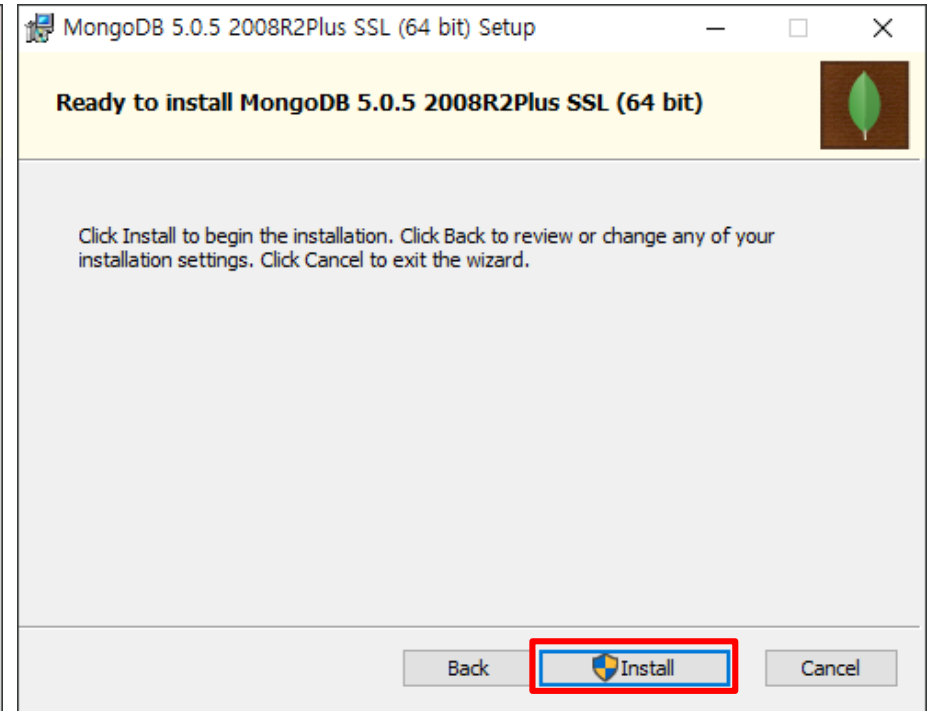
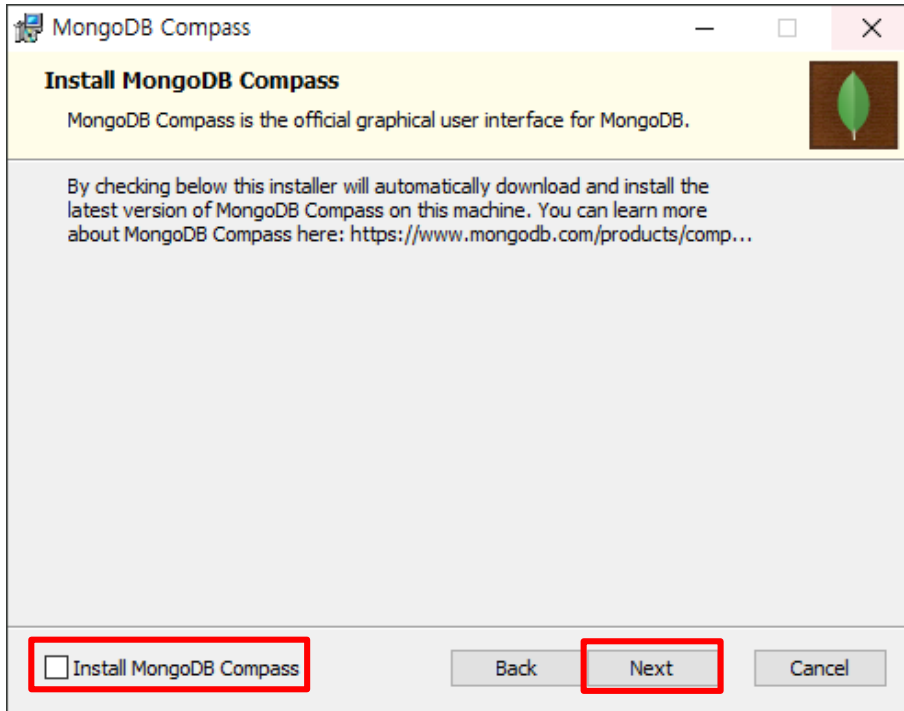
5) Complete 클릭

6) 설치 경로 확인 → Next 클릭

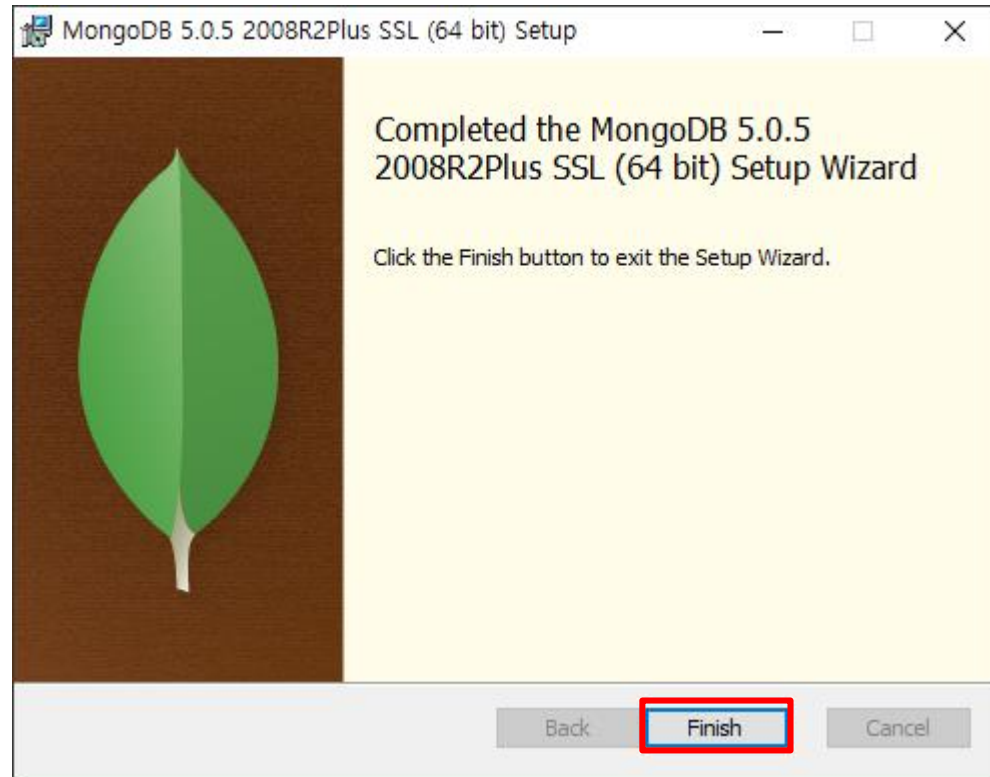


7) Install MongoDB Compass 체크 해제 → Next 클릭

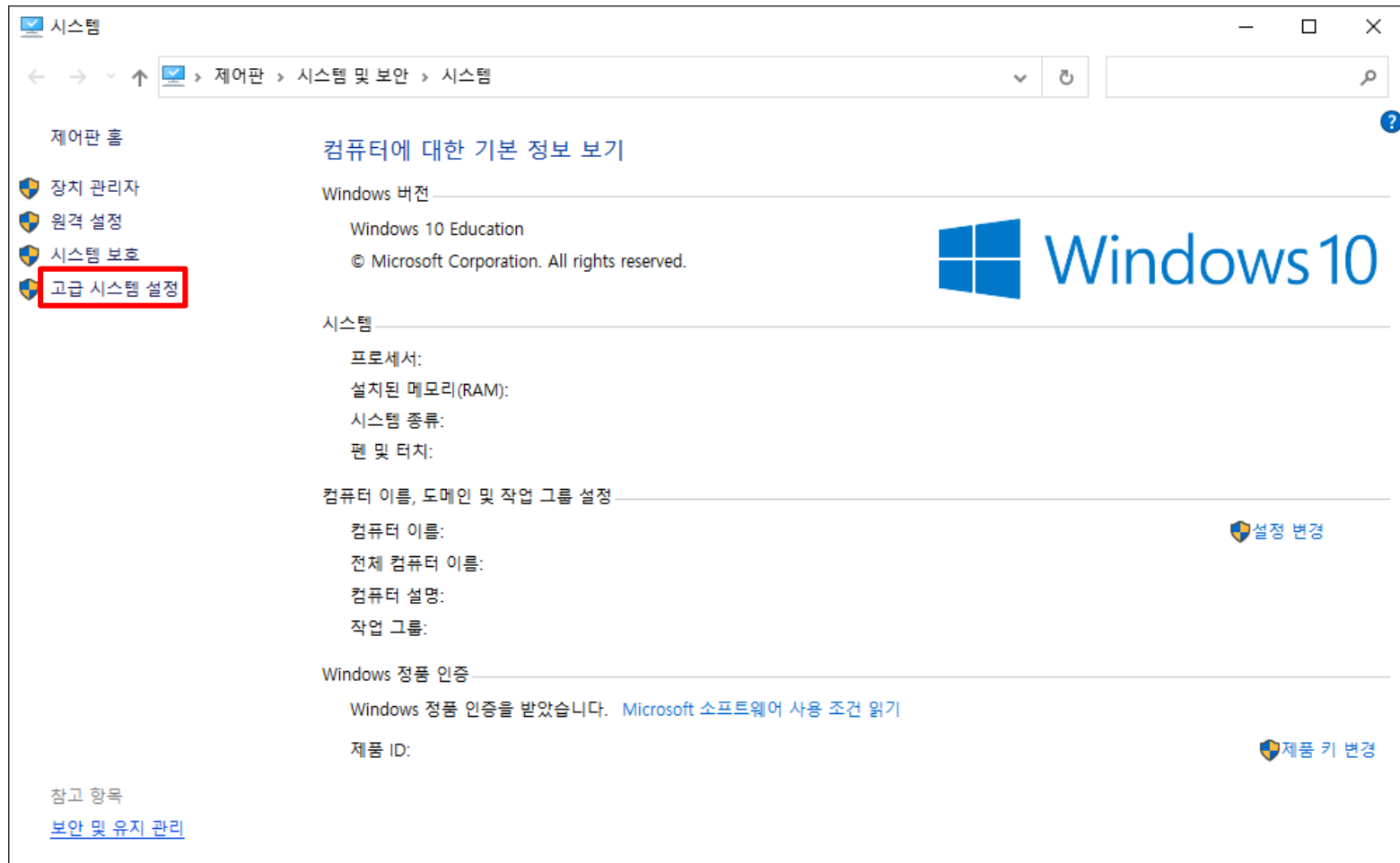
8) Install 클릭



9) Finish 클릭

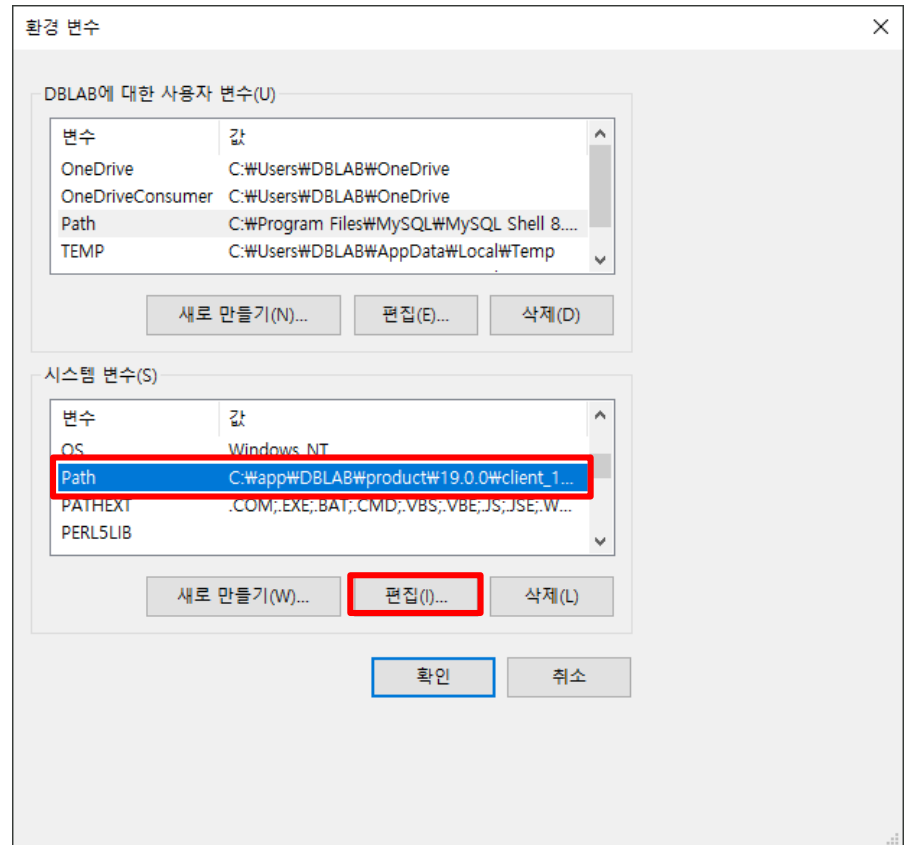
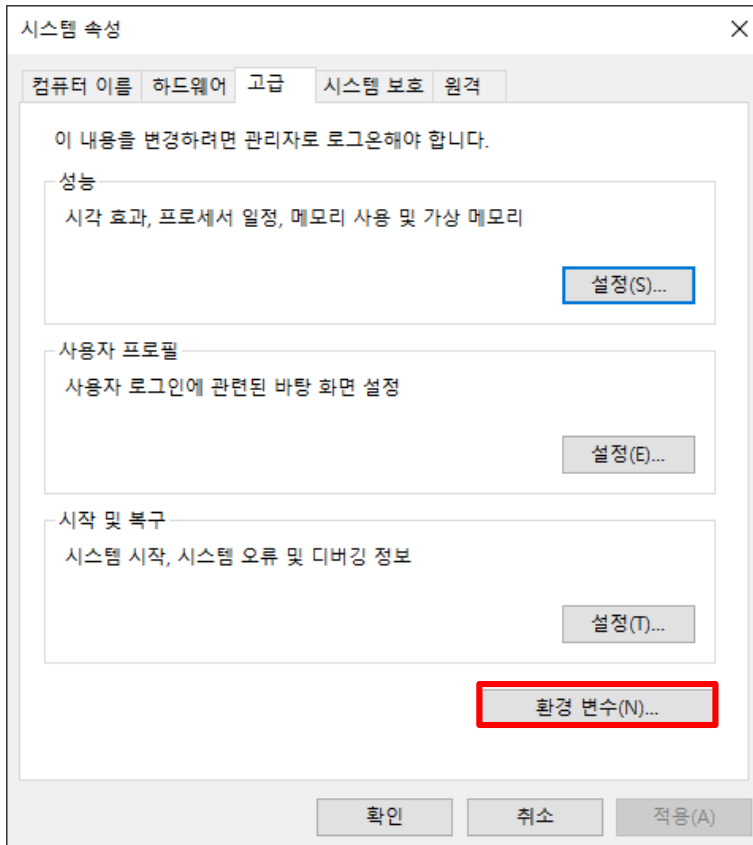


11) 내 PC 우클릭 → 속성 → 고급 시스템 설정 클릭



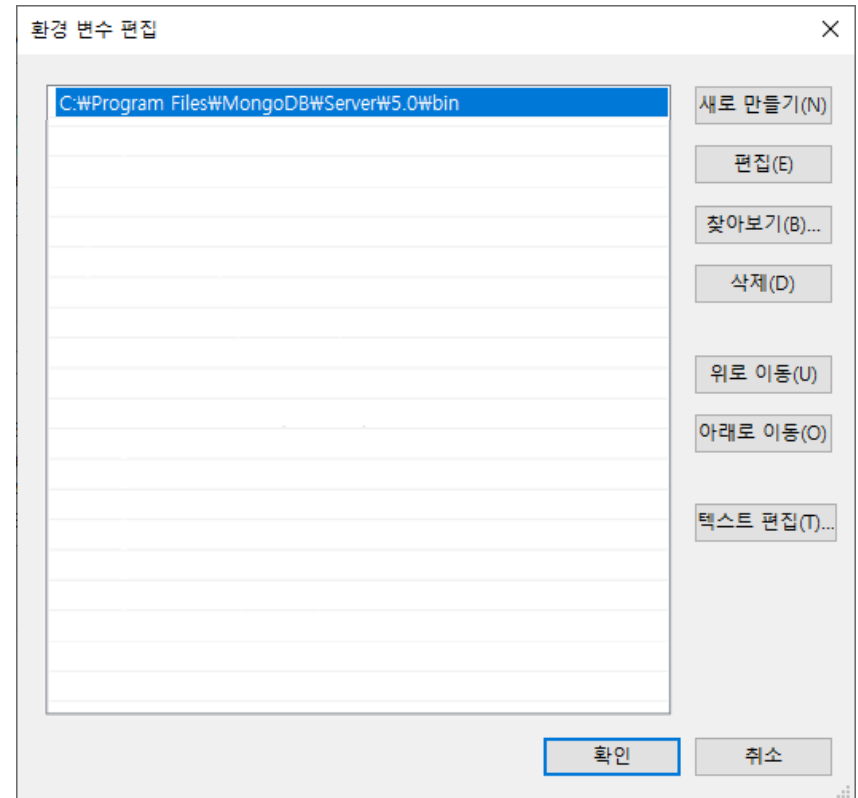
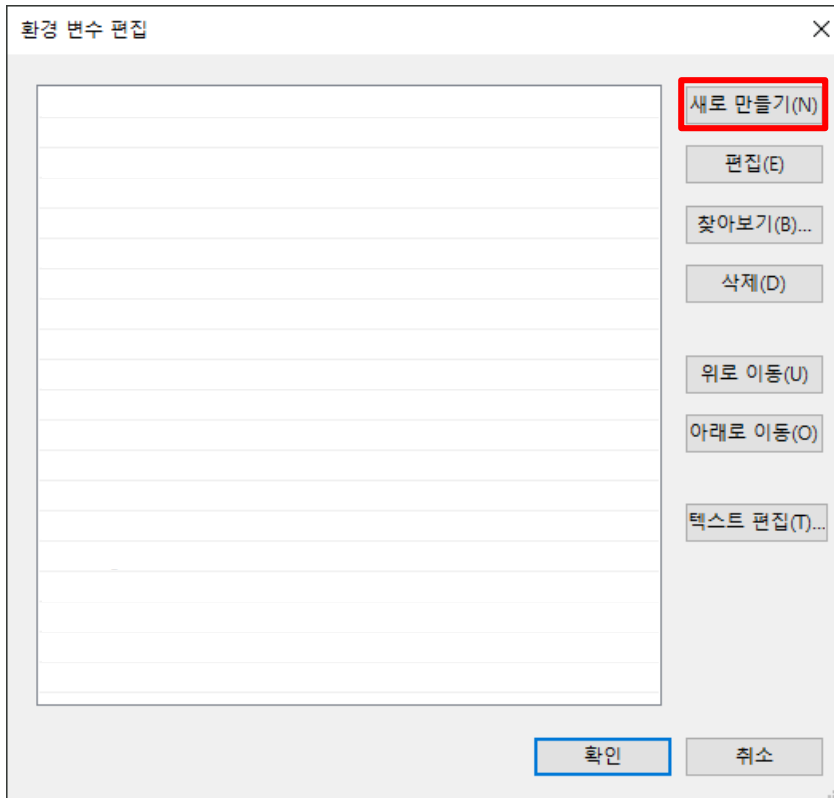
12) 환경 변수 클릭

13) 시스템 변수 → Path 선택 → 편집 클릭



14) 새로 만들기 클릭

15) C:\Program Files\MongoDB\Server\5.0\bin 입력 → 확인 클릭



16) cmd 실행 → mongo --version 입력하여 설치 확인



```

C:\ 명령 프롬프트
Microsoft Windows [Version 10.0.19041.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\WDBLAB>mongo --version
MongoDB shell version v5.0.5
Build Info: {
  "version": "5.0.5",
  "gitVersion": "d65fd89df3fc039b5c55933c0f71d647a54510ae",
  "modules": [],
  "allocator": "tcmalloc",
  "environment": {
    "distmod": "windows",
    "distarch": "x86_64",
    "target_arch": "x86_64"
  }
}
```

1) <https://docs.mongodb.com/mongodb-shell/install/> 접속

The screenshot shows the MongoDB documentation website for the MongoDB Shell installation. The page has a green header with a navigation bar and a sidebar. The main content area is titled "Install mongosh" and includes sections for prerequisites, supported versions, and the installation procedure. A "NOTE" box highlights that on Windows, preferences are stored in a specific directory. The "Install from MSI" section is partially visible at the bottom.

EVENT Interested in speaking at MongoDB World 2022? - Become a speaker >>

MongoDB. Products Solutions Resources Company Pricing

Search Sign In Try Free

MongoDB Documentation

- Back To View & Analyze Data
- MongoDB Shell**
 - Install mongosh**
 - Connect to a Deployment
 - Client-Side Field Level Encryption
 - Retrieve Shell Logs
 - mongosh Usage
 - Snippets
 - Write Scripts for mongosh
 - Reference
 - Access mongosh Help
 - Compatibility Changes with Legacy mongo Shell
 - mongosh Changelog

Docs Home → View & Analyze Data → MongoDB Shell

Install mongosh

Prerequisites

To use the MongoDB Shell, you must have a MongoDB deployment to connect to.

- For a free cloud-hosted deployment, you can use [MongoDB Atlas](#).
- To learn how to run a local MongoDB deployment, see [Install MongoDB](#).

Supported MongoDB Versions

You can use the MongoDB Shell to connect to MongoDB version 4.0 or greater.

Procedure

Select the appropriate tab for your operating system:

Windows macOS Linux

NOTE

On Windows, `mongosh` preferences and configuration options are stored in the `%APPDATA%/mongodb/mongosh` directory.

Install from MSI

Give Feedback

2) MongoDB Download Center 클릭

Install from MSI

- 1 Open the MongoDB Shell download page. 1

Open the [MongoDB Download Center](#) .

- 2 In the *Platform* dropdown, select *Windows 64-bit (8.1+) (MSI)*
- 3 Click *Download*.
- 4 Double-click the installer file.
- 5 Follow the prompts to install `mongosh`.

3) 1.1.6 버전과 msi 패키지 선택 후 다운로드 클릭 (Platform의 경우 자신의 OS에 맞게 선택)

MongoDB Shell

MongoDB Shell is the quickest way to connect to (and work with) MongoDB. Easily query data, configure settings, and execute other actions with this modern, extensible command-line interface — replete with syntax highlighting, intelligent autocomplete, contextual help, and error messages.

Note: MongoDB Shell is an open source (Apache 2.0), standalone product developed separately from the MongoDB Server.

Available Downloads

Version

1.1.6

Platform

Windows 64-bit (8.1+) (MSI) ✓

Package

msi

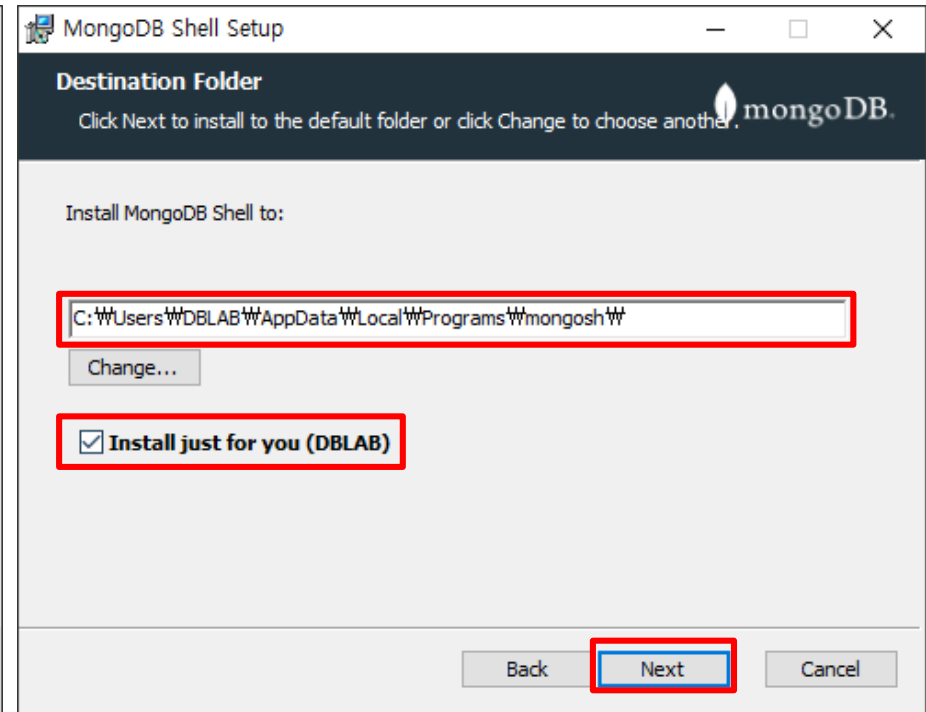
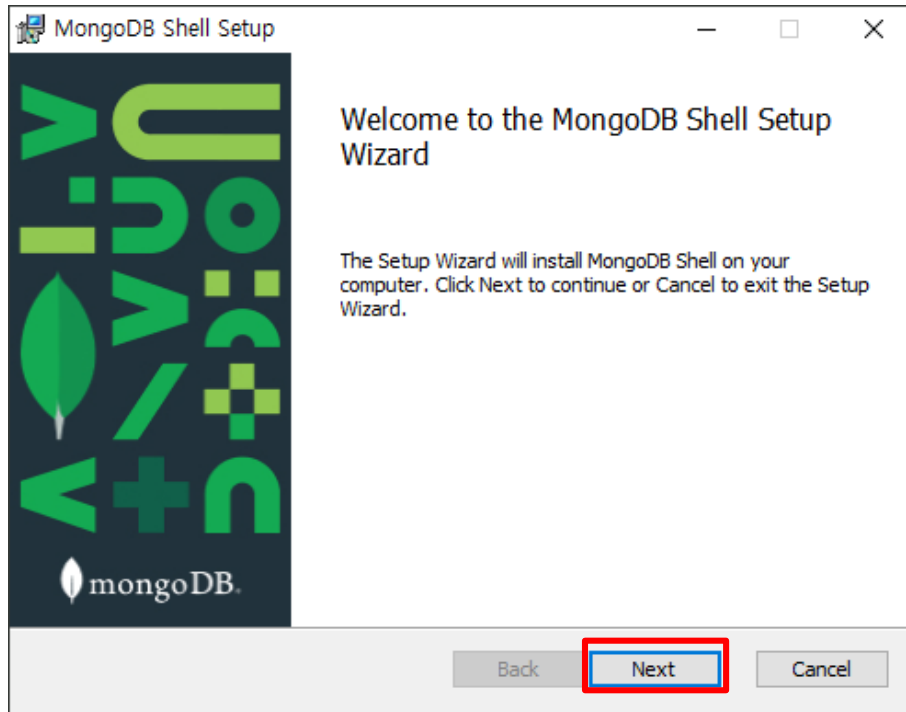
Download

Copy Link

[Documentation](#)

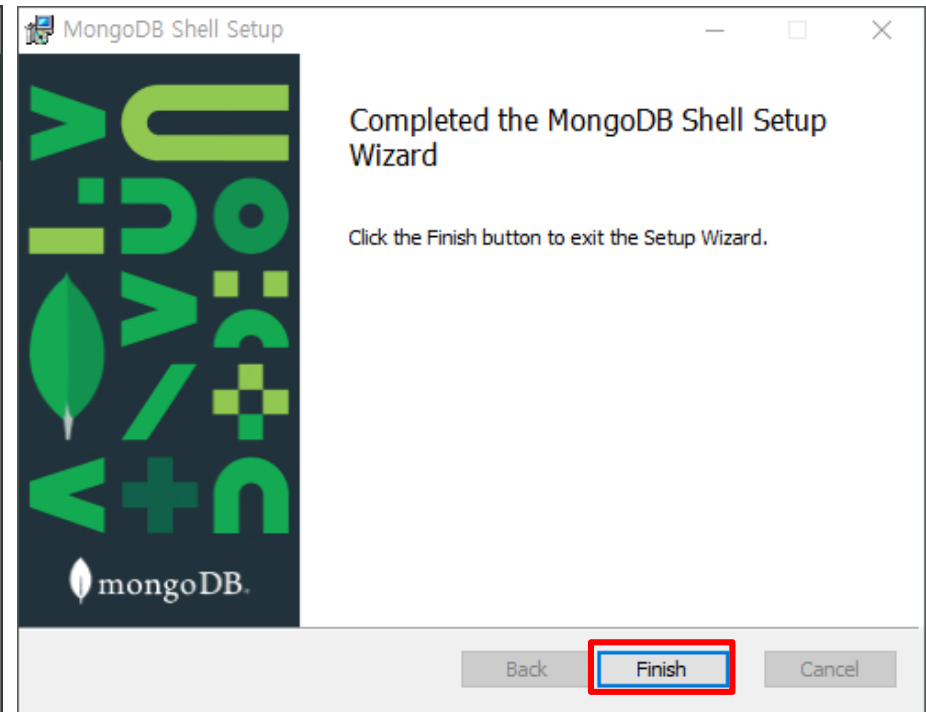
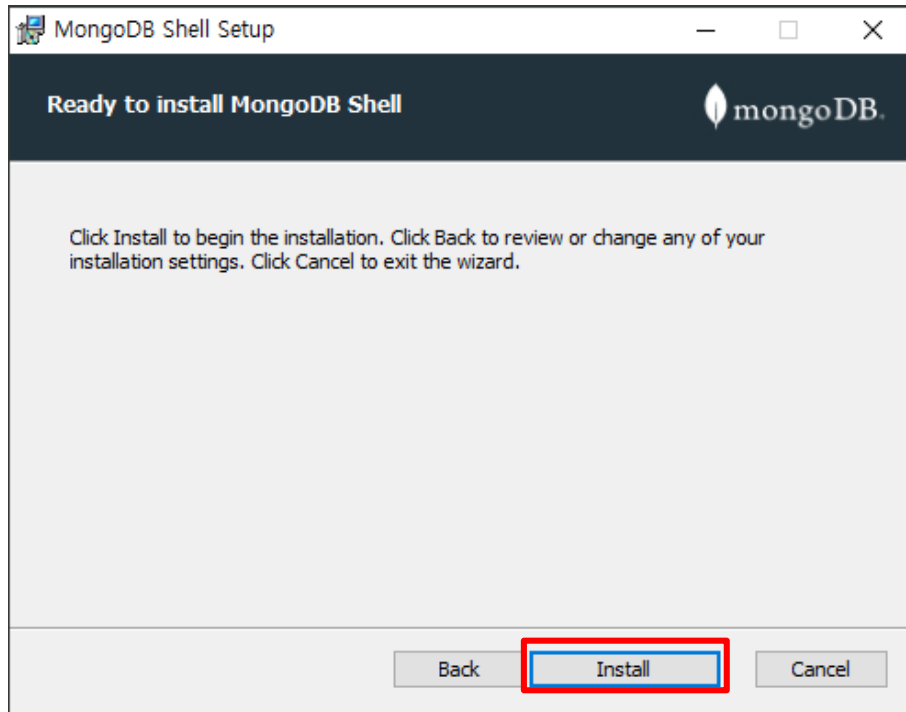
4) 다운로드 받은 설치 파일 실행 → Next 클릭

5) 설치 경로 확인 → Install just for you 체크 → Next 클릭

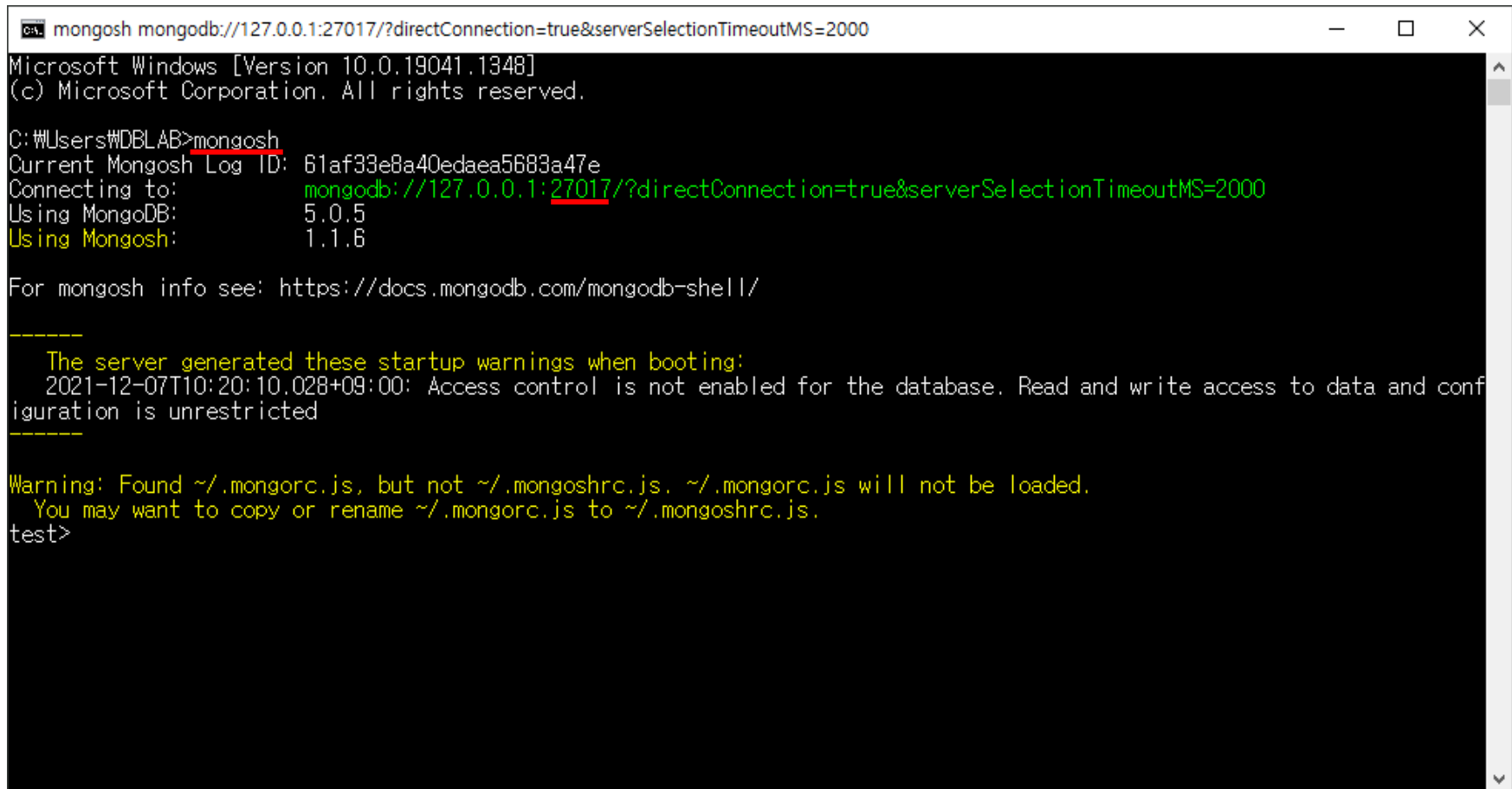


6) Install 클릭

7) 설치 완료 후 Finish 클릭



1) cmd창에 "mongosh" 입력 → "27017" 포트로 실행됨을 확인



```
cmd mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Microsoft Windows [Version 10.0.19041.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\WDBLAB>mongosh
Current Mongosh Log ID: 61af33e8a40edaea5683a47e
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Using MongoDB:      5.0.5
Using Mongosh:       1.1.6

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

-----
  The server generated these startup warnings when booting:
  2021-12-07T10:20:10.028+09:00: Access control is not enabled for the database. Read and write access to data and conf
  igation is unrestricted
-----

Warning: Found ~/.mongorc.js, but not ~/.mongoshrc.js. ~/.mongorc.js will not be loaded.
  You may want to copy or rename ~/.mongorc.js to ~/.mongoshrc.js.
test>
```

2) "show dbs" 입력

```
cmd mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Microsoft Windows [Version 10.0.19041.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\WDBLAB>mongosh
Current Mongosh Log ID: 61af33e8a40edaea5683a47e
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Using MongoDB:      5.0.5
Using Mongosh:       1.1.6

For mongosh info see: https://docs.mongodb.com/mongodbs-shell/

-----
  The server generated these startup warnings when booting:
  2021-12-07T10:20:10.028+09:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

Warning: Found ~/.mongorc.js, but not ~/.mongoshrc.js. ~/.mongorc.js will not be loaded.
You may want to copy or rename ~/.mongorc.js to ~/.mongoshrc.js.
test> show dbs
admin      41 kB
config     36.9 kB
local      41 kB
test>
```

- 데이터베이스 생성
 - 명령어
 - use <생성할 데이터베이스 이름>
 - 예제) testdb 라는 이름의 데이터베이스 생성
 - use testdb

```
test> use testdb
switched to db testdb
testdb>
```

- 현재 사용하고 있는 데이터베이스 조회
 - 명령어
 - db

```
testdb> db
testdb
```

- Collection 생성

- Collection

- 데이터베이스 내에 실제로 사용되는 도큐먼트

- 명령어

- db.createCollection(“<생성할 collection 이름>”)

- 예제) customers 라는 이름의 collection 생성

- db.createCollection(“customers”)

```
testdb> db.createCollection("customers")
{ ok: 1 }
testdb>
```

- 데이터 입력

- 명령어

- db.<collection 이름>.insertOne(<생성할 데이터>)
 - db.컬렉션명.insertOne({필드명:"값", ...})

- 예제) customers 라는 collection 안에 이름은 gildong이고 나이는 22인 고객의 데이터를 생성

```
testdb> db.customers.insertOne({name:"gildong", age: 22})
{
  acknowledged: true,
  insertedId: ObjectId("61af1d91a8497cd40f6ce5ad")
}
testdb>
```


■ 데이터 조회

• 명령어

- `db.<collection 이름>.find`
(<검색 조건>)
- `db.컬렉션명.find`
({필드명 : "조건값", ...})
- `db.customers.find()` 를 사용
하는 경우 전체 내용 검색

- 예제) customers 라는 collection
안에 이름이 gildong인 데이터 검색

```
testdb> db.customers.find()
[
  {
    _id: ObjectId("61af19d3a8497cd40f6ce5a6"),
    name: 'gildong',
    age: 22
  },
  {
    _id: ObjectId("61af1a78a8497cd40f6ce5a7"),
    name: 'cheolsu',
    age: 33
  },
  {
    _id: ObjectId("61af1a7ba8497cd40f6ce5a8"),
    name: 'yeonghui',
    age: 32
  }
]
testdb> db.customers.find({name: "gildong"})
[
  {
    _id: ObjectId("61af19d3a8497cd40f6ce5a6"),
    name: 'gildong',
    age: 22
  }
]
testdb>
```

■ 데이터 갱신

• 명령어

- db.<collection 이름>.updateOne

(<갱신할 데이터 선택>, <갱신할 데이터 입력>)

- db.컬렉션명.updateOne

({필드명:"조건값", ...}, {\$set: {필드명:"변경값", ...}})

```
testdb> db.customers.updateOne({name:"gildong"}, {$set: {name:"hana"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
testdb> db.customers.find()
[
  { _id: ObjectId("61af1d91a8497cd40f6ce5ad"), name: 'hana', age: 22 },
  {
    _id: ObjectId("61af1dd4a8497cd40f6ce5ae"),
    name: 'cheolsu',
    age: 33
  },
  {
    _id: ObjectId("61af1dd4a8497cd40f6ce5af"),
    name: 'yeonghui',
    age: 32
  }
]
```

- 데이터 삭제

- 명령어

- `db.<collection 이름>.deleteOne(<삭제할 데이터 선택>)`
- `db.컬렉션명.deleteOne({필드명:"조건값", ...})`

- 예제) customers 라는 collection 안에 이름이 hana인 데이터를 삭제

```
testdb> db.customers.deleteOne({name:"hana"});
{ acknowledged: true, deletedCount: 1 }
testdb> db.customers.find()
[
  {
    _id: ObjectId("61af1dd4a8497cd40f6ce5ae"),
    name: 'cheolsu',
    age: 33
  },
  {
    _id: ObjectId("61af1dd4a8497cd40f6ce5af"),
    name: 'yeonghui',
    age: 32
  }
]
testdb>
```

Name	Methods	Description	Example
insert	insertOne	단일의 도큐먼트를 collection에 insert	<pre>db.collection.insertOne(<document>, { writeConcern: <document> })</pre>
	insertMany	다수의 도큐먼트를 collection에 insert	<pre>db.collection.insertMany([<document 1>, <document 2>, ...], { writeConcern: <document>, ordered: <boolean> })</pre>
update	updateOne	특정 조건에 맞는 다수의 도큐먼트가 있더라도 하나의 도큐먼트만 update	<pre>db.collection.updateOne(<filter>, <update>, { upsert: <boolean>, writeConcern: <document>, collation: <document>, arrayFilters: [<filterdocument1>, ...], hint: <document string> })</pre>
	updateMany	특정 조건에 맞는 모든 도큐먼트를 update	<pre>db.collection.updateMany(<filter>, <update>, { upsert: <boolean>, writeConcern: <document>, collation: <document>, arrayFilters: [<filterdocument1>, ...], hint: <document string> })</pre>

Name	Methods	Description	Example
update	replaceOne	특정 조건에 맞는 다수의 도큐먼트가 있더라도 하나의 도큐먼트만 replace	<pre>db.collection.replaceOne(<filter>, <replacement>, { upsert: <boolean>, writeConcern: <document>, collation: <document>, hint: <document string> })</pre>
delete	deleteOne	특정 조건에 맞는 다수의 도큐먼트가 있더라도 하나의 도큐먼트만 delete	<pre>db.collection.deleteOne(<filter>, { writeConcern: <document>, collation: <document>, hint: <document string> })</pre>
	deleteMany	특정 조건에 맞는 모든 도큐먼트를 delete	<pre>db.collection.deleteMany(<filter>, { writeConcern: <document>, collation: <document> })</pre>

■ 비교 연산자

Name	Description	Example
\$eq(equals)	주어진 값과 일치하는 값	Syntax: (field: { \$eq: value })
\$gt(greater than)	주어진 값보다 큰 값	Syntax: (field: { \$gt: value })
\$gte(greater than or equals)	주어진 값보다 크거나 같은 값	Syntax: (field: { \$gte: value })
\$in	주어진 배열 안에 속하는 값	Syntax: (field: { \$in: [value1, ... , valueN] })
\$lt(less than)	주어진 값보다 작은 값	Syntax: (field: { \$lt: value })
\$lte(less than or equals)	주어진 값보다 작거나 같은 값	Syntax: (field: { \$lte: value })
\$ne(not equal)	주어진 값과 일치하지 않는 값	Syntax: (field: { \$ne: value })
\$nin(not in)	주어진 배열 안에 속하지 않는 값	Syntax: (field: { \$nin: [value1, ... , valueN] })

■ 논리 연산자


Name	Example
\$or	Syntax: (\$or: [{ expression1 }, ... , { expressionN }])
\$and	Syntax: (\$and: [{ expression1 }, ... , { expressionN }])
\$not	Syntax: ({ field: { \$not: { operator-expression } } })
\$nor	Syntax: ({ \$not: [{ expression1 }, ... , { expressionN }] })

■ 참고 사이트

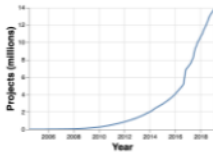
- <https://docs.mongodb.com/manual/reference/operator/query/>

```
[  
  {  
    field      value  
    ↓          ↓  
    "id" : "아이디",  
    "name" : "이름",  
    "addr" : "주소",  
    "major" : "전공",  
    "pwd" : "비밀번호"  
  },  
  ...  
]
```

- 1) <https://mvnrepository.com/artifact/org.mongodb/mongodb-driver/3.4.3> 접속
- 2) jar 파일 다운로드




Categories | Popular | Contact Us

Indexed Artifacts (23.3M)**Popular Categories**

- Aspect Oriented
- Actor Frameworks
- Application Metrics
- Build Tools
- Bytecode Libraries
- Command Line Parsers
- Cache Implementations
- Cloud Computing
- Code Analyzers
- Collections
- Configuration Libraries
- Core Utilities
- Date and Time Utilities
- Dependency Injection
- Embedded SQL Databases
- HTML Parsers
- HTTP Clients
- I/O Utilities

[Home](#) » [org.mongodb](#) » [mongodb-driver](#) » **3.4.3**

**MongoDB Java Driver » 3.4.3**

The MongoDB Driver uber-artifact that combines mongodb-driver-sync and the legacy driver

License	Apache 2.0
HomePage	http://www.mongodb.org
Date	(Jul 31, 2017)
Files	pom (1 KB) jar (358 KB) View All
Repositories	Central Sonatype Spring Plugins
Used By	319 artifacts

Note: There is a new version for this artifact

New Version	3.12.10
-------------	---------

[Maven](#) [Gradle](#) [Gradle \(Short\)](#) [Gradle \(Kotlin\)](#) [SBT](#) [Ivy](#) [Grape](#) [Leiningen](#) [Buildr](#)

```
// https://mvnrepository.com/artifact/org.mongodb/mongodb-driver
@Grapes(
    @Grab(group='org.mongodb', module='mongodb-driver', version='3.4.3')
)
```

☒ Include comment with link to declaration

Compile Dependencies (3)

Category / License	Group / Artifact	Version	Updates
--------------------	------------------	---------	---------

Indexed Repositories (1344)


- Central
- Sonatype
- Atlassian
- Spring Plugins
- Spring Lib M
- Hortonworks
- JCenter
- Atlassian Public
- JBossEA
- BeDataDriven

Popular Tags

android apache api application assets aws build build-system camel client clojure cloud config data database eclipse example extension framework github gradle groovy http integration io jboss library logging maven module osgi persistence platform plugin repository rest rlang scala sdk security server service spring starter streaming testing tools ui web webapp

Web site developed by @frodriquez


- 3) <https://mvnrepository.com/artifact/org.mongodb/mongodb-driver-core/3.4.3> 접속
- 4) jar 파일 다운로드

[Categories](#) | [Popular](#) | [Contact Us](#)

Indexed Artifacts (23.3M)**Popular Categories**

- Aspect Oriented
- Actor Frameworks
- Application Metrics
- Build Tools
- Bytecode Libraries
- Command Line Parsers
- Cache Implementations
- Cloud Computing
- Code Analyzers
- Collections
- Configuration Libraries
- Core Utilities
- Date and Time Utilities
- Dependency Injection
- Embedded SQL Databases
- HTML Parsers
- HTTP Clients
- I/O Utilities

[Home](#) » [org.mongodb](#) » [mongodb-driver-core](#) » **3.4.3**

**MongoDB Java Driver Core » 3.4.3**

The Java operations layer for the MongoDB Java Driver. Third parties can wrap this layer to provide custom higher-level APIs

License	Apache 2.0
HomePage	http://www.mongodb.org
Date	(Jul 31, 2017)
Files	pom (2 KB) jar (1.0 MB) View All
Repositories	Central Sonatype Spring Plugins
Used By	138 artifacts

Note: There is a new version for this artifact

New Version	4.3.1
-------------	-------

[Maven](#) [Gradle](#) [Gradle \(Short\)](#) [Gradle \(Kotlin\)](#) [SBT](#) [Ivy](#) [Grape](#) [Leiningen](#) [Buildr](#)

```
// https://mvnrepository.com/artifact/org.mongodb/mongodb-driver-core
@Grapes(
    @Grab(group='org.mongodb', module='mongodb-driver-core', version='3.4.3')
)
```

☒ Include comment with link to declaration

Compile Dependencies (5)

Indexed Repositories (1344)

- Central
- Sonatype
- Atlassian
- Spring Plugins
- Spring Lib M
- Hortonworks
- JCenter
- Atlassian Public
- JBossEA
- BeDataDriven

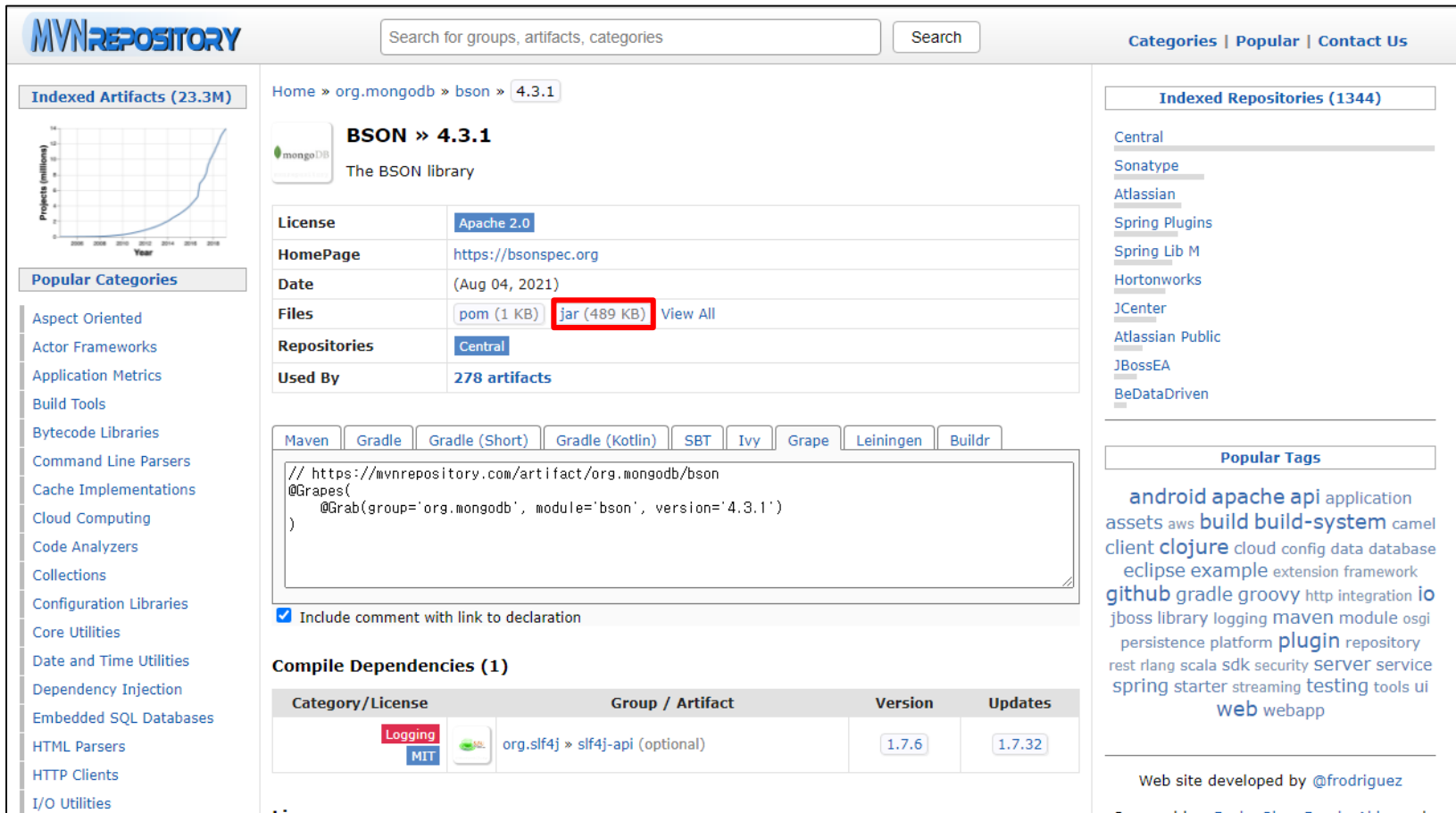
Popular Tags

android apache api application assets aws build build-system camel client clojure cloud config data database eclipse example extension framework github gradle groovy http integration io jboss library logging maven module osgi persistence platform plugin repository rest rlang scala sdk security server service spring starter streaming testing tools ui web webapp

Web site developed by [@frodriquez](#)

5) <https://mvnrepository.com/artifact/org.mongodb/bson/4.3.1> 접속

6) jar 파일 다운로드



MVNREPOSITORY Search for groups, artifacts, categories Search Categories | Popular | Contact Us

Indexed Artifacts (23.3M)

Home » org.mongodb » bson » 4.3.1

BSON » 4.3.1
The BSON library

License: Apache 2.0
HomePage: <https://bsonspec.org>
Date: (Aug 04, 2021)
Files: [pom \(1 KB\)](#) **[jar \(489 KB\)](#)** View All
Repositories: Central
Used By: 278 artifacts

Maven Gradle Gradle (Short) Gradle (Kotlin) SBT Ivy Grape Leiningen Buildr

```
// https://mvnrepository.com/artifact/org.mongodb/bson
@Grapes(
    @Grab(group='org.mongodb', module='bson', version='4.3.1')
)
```

☒ Include comment with link to declaration

Compile Dependencies (1)

Category/License	Group / Artifact	Version	Updates
Logging MIT	org.slf4j » slf4j-api (optional)	1.7.6	1.7.32

Indexed Repositories (1344)

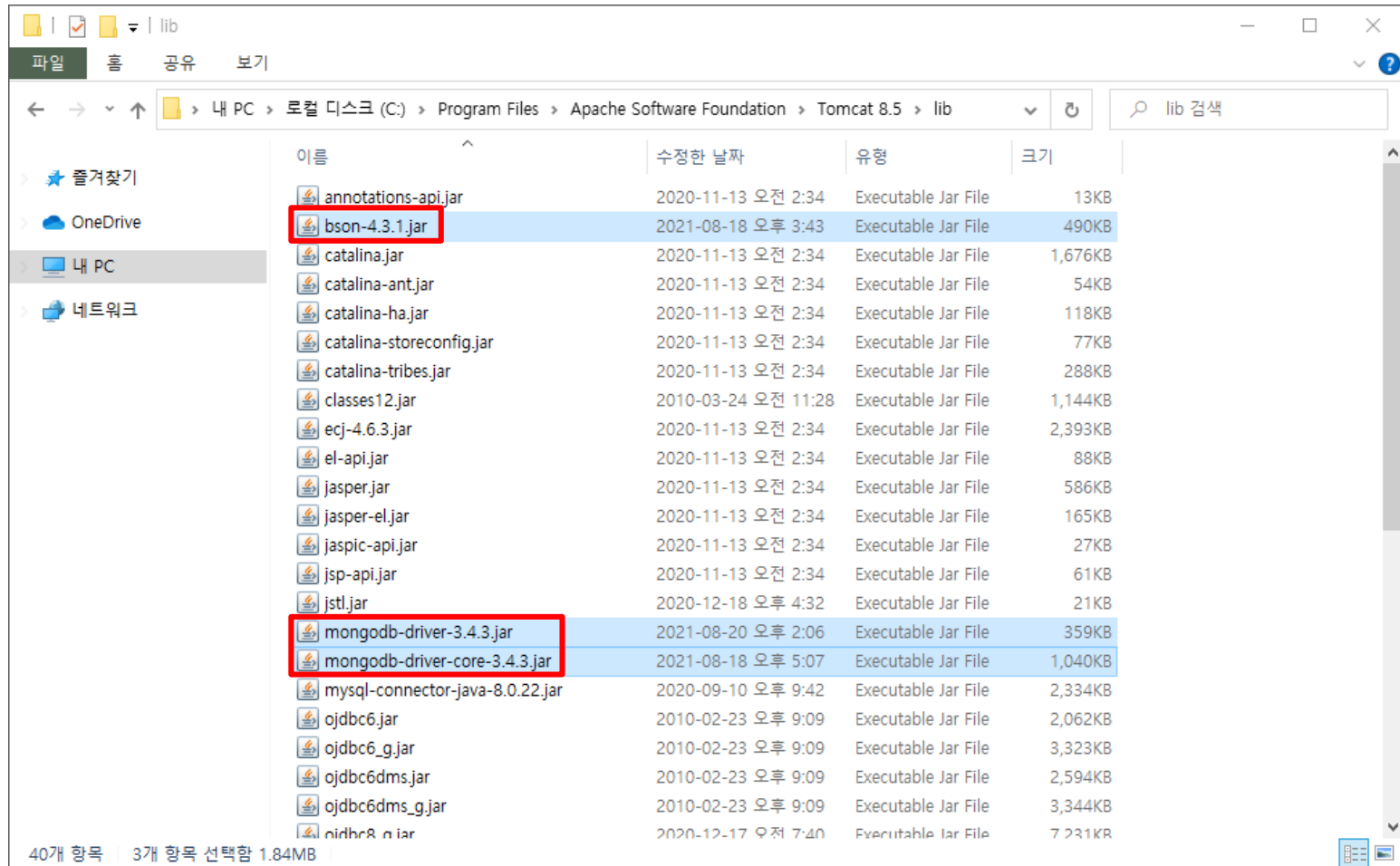
Central
Sonatype
Atlassian
Spring Plugins
Spring Lib M
Hortonworks
JCenter
Atlassian Public
JBossEA
BeDataDriven

Popular Tags

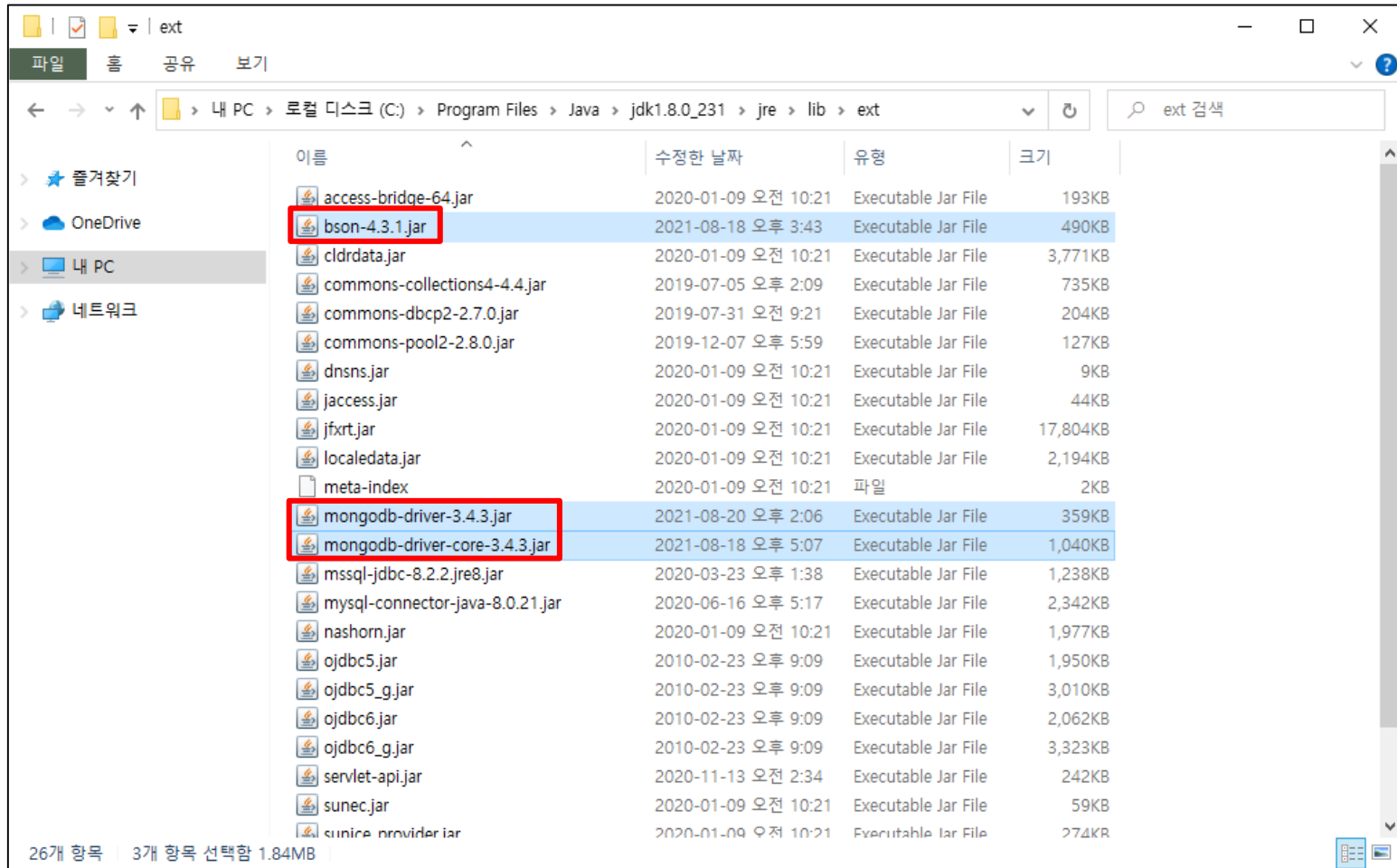
android apache api application assets aws build build-system camel client clojure cloud config data database eclipse example extension framework github gradle groovy http integration io jboss library logging maven module osgi persistence platform plugin repository rest rang scala sdk security server service spring starter streaming testing tools ui web webapp

Web site developed by @frodriguez
Powered by Scala, Play, Grails, Akka and

7) C:\Program Files\Apache Software Foundation\Tomcat 8.5\lib에 다운로드한 jar 파일 복사



8) C:\Program Files\Java\jdk1.8.0_231\jre\lib\ext에 다운로드한 jar 파일 복사



■ MongoDB 드라이버를 이용한 데이터베이스 연결

- MongoClient mongoClient = new MongoClient("서버 주소", "서버 포트");
 - 서버 주소 : localhost (외부 서버 사용 시 외부 서버 주소 입력)
 - 서버 포트 : 27017 (포트 변경 시 변경한 포트 입력)
- MongoDB db = mongoClient.getDatabase("DB 이름");
 - DB 이름 : testdb

■ MongoDB 드라이버를 이용한 컬렉션 접근

- MongoClient<Document> collection = db.getCollection("컬렉션명");
 - 컬렉션명 : 접근할 컬렉션명 입력 (ex. member)

■ 연결 확인 예시 (conncet.jsp)

```
<%@ page import="org.bson.Document"%>
<%@ page import="com.mongodb.*"%>
<%@ page import="com.mongodb.client.MongoCollection"%>
<%@ page import="com.mongodb.client.MongoDatabase"%>
<%@ page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
<%

    Boolean connect = false;

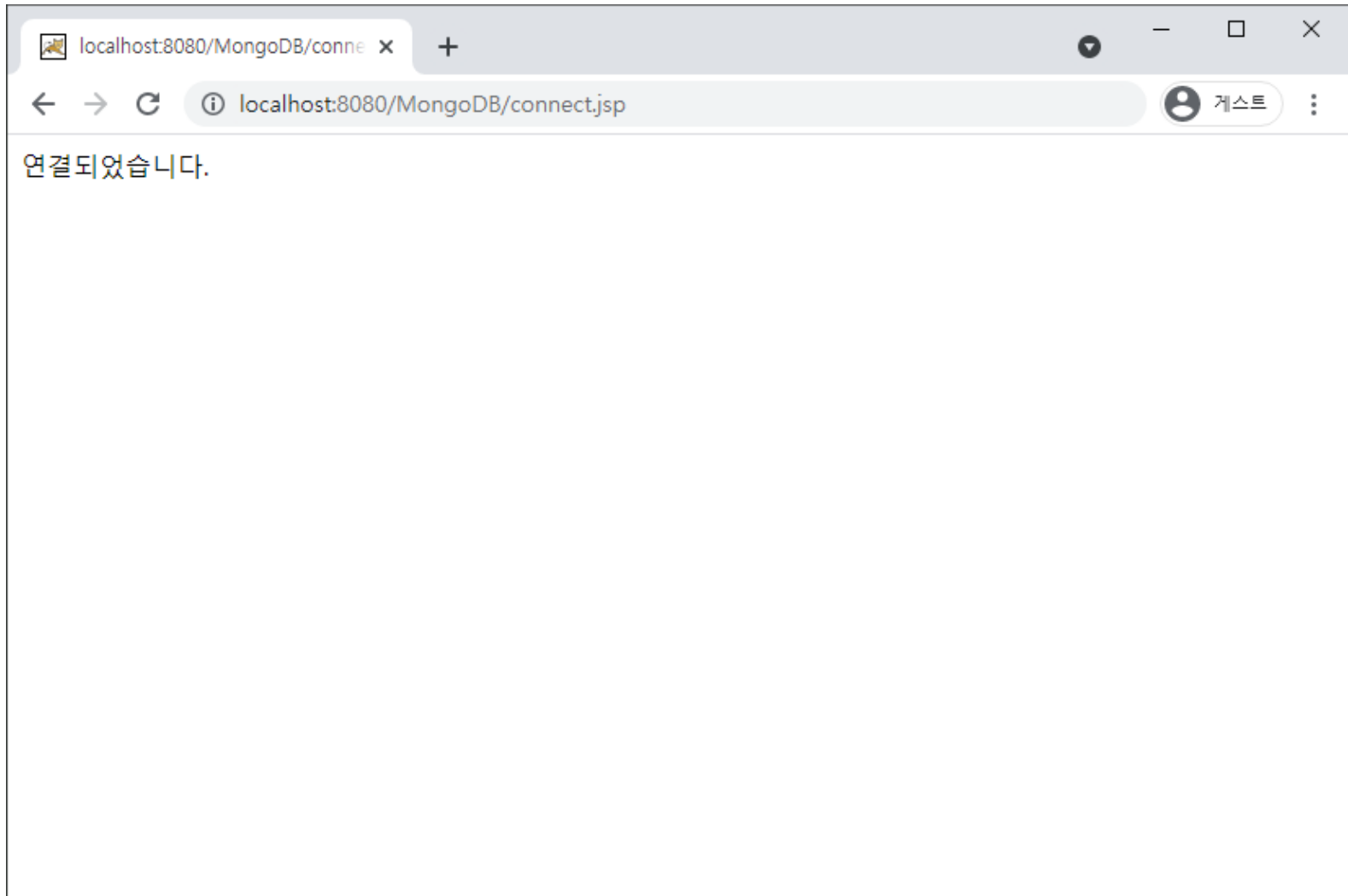
    try{
        MongoClient mongoClient = new MongoClient("localhost", 27017); // MongoDB 연결
        MongoDatabase db = mongoClient.getDatabase("testdb"); // 데이터베이스 선택
        connect = true;
        mongoClient.close();
    }catch(Exception e){
        connect = false;
        out.print(e);
    }

%>
<html>
<head></head>
<body>
<%

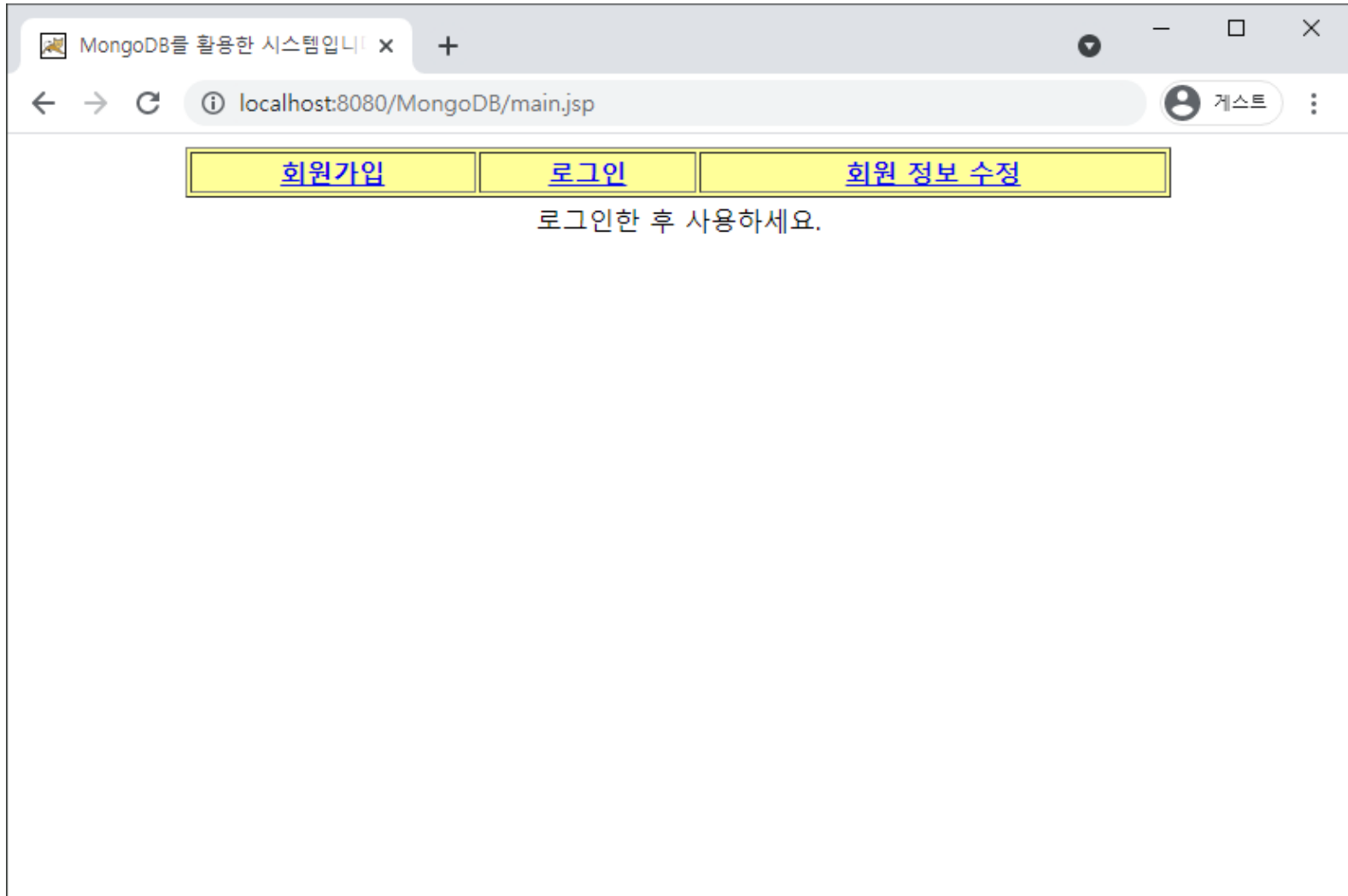
    if (connect == true){
        out.print("연결되었습니다.");
    } else {
        out.print("연결에 실패하였습니다.");
    }

%>
</body>
</html>
```

■ 연결 확인 예시



■ 메인 화면



■ 회원가입

회원가입 x +

localhost:8080/MongoDB/signUp.jsp 게스트

회원가입		로그인	회원 정보 수정
아이디	<input type="text"/>		
패스워드	<input type="password"/>		
이름	<input type="text"/>		
주소	<input type="text"/>		
전공	<input type="text"/>		
		가입	취소

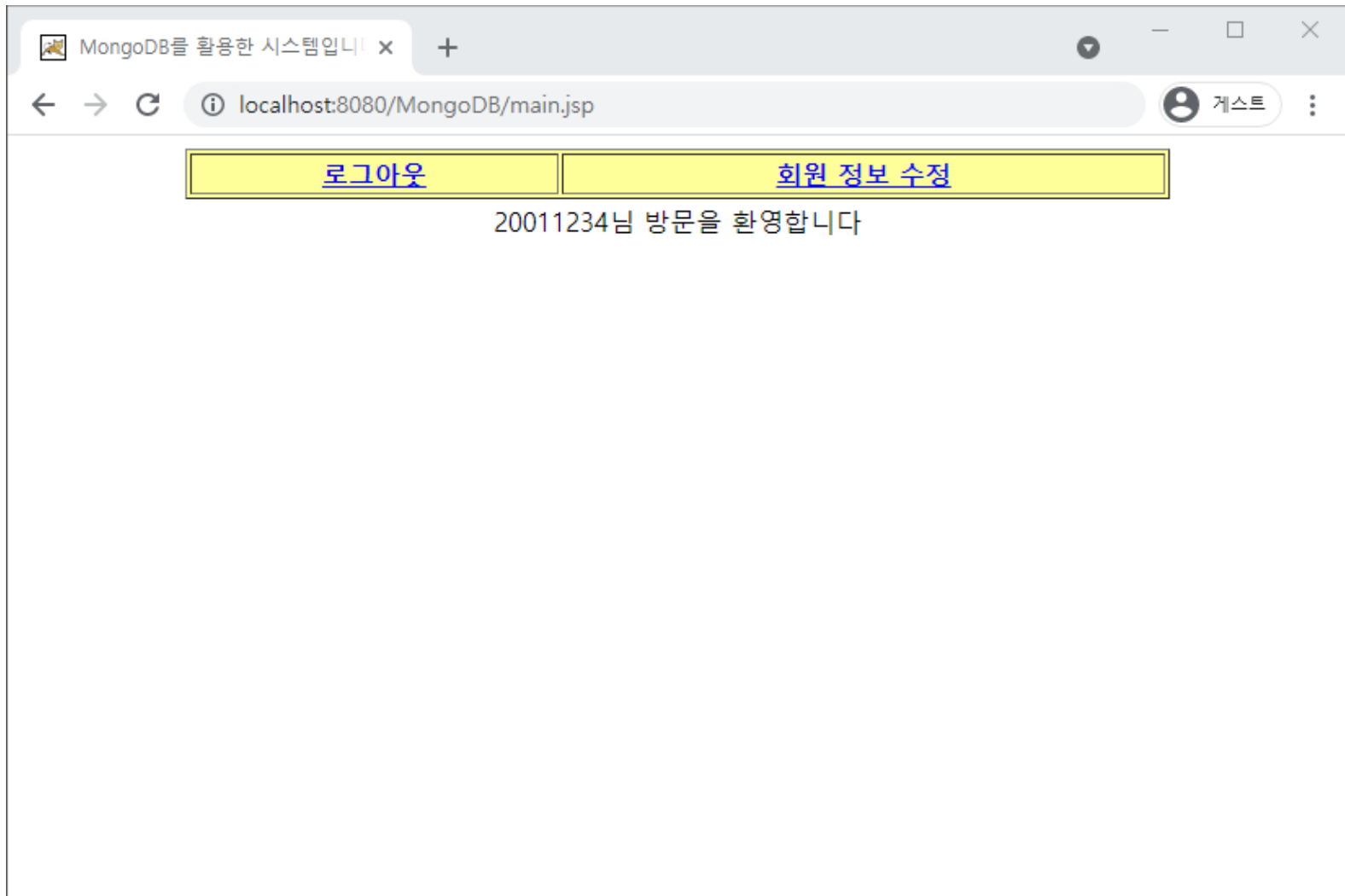
로그인

로그인

← → ↻ ⓘ localhost:8080/MongoDB/login.jsp 게스트 ⋮

회원가입	로그인	회원 정보 수정
아이디	20011234	
패스워드	
로그인		취소

로그인 성공한 메인화면



■ 회원 정보 수정

사용자 정보 수정

localhost:8080/MongoDB/update.jsp

게스트

로그아웃		회원 정보 수정	
이름	<input type="text" value="신경화"/>		
주소	<input type="text" value="경기도 화성군 송산면 고정1리 540-2"/>		
패스워드	<input type="password" value="...."/>		
전공	<input type="text" value="컴퓨터공학"/>		
		<input type="button" value="수정"/>	<input type="button" value="회원탈퇴"/>

```
<%@ page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
<html>
<head>
<title>Insert title here</title>
</head>
<body>
    <%
        String session_id = (String)session.getAttribute("user");
        String log;

        if(session_id==null) log="<a href=login.jsp>로그인</a>";
        else                  log="<a href=logout.jsp>로그아웃</a>";
    %>
    <table width="75%" align="center" bgcolor="#FFFF99" border>
        <tr>
            <%
                if(session_id==null){
            %>
            <td align="center"><b><a href="signUp.jsp">회원가입</b></td>
            <%
                }
            %>
            <td align="center"><b><%=log%></b></td>
            <td align="center"><b><a href="update.jsp">회원 정보 수정</b></td>
        </tr>
    </table>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
<!DOCTYPE html>
<html>
<head>
<title>MongoDB를 활용한 시스템입니다.</title>
</head>
<body>
    <%@ include file="top.jsp"%>
    <table width="75%" align="center" height="100%">
        <%
            if (session_id != null) {
        %>
        <tr>
            <td align="center"><%=session_id%>님 방문을 환영합니다</td>
        </tr>
        <%
            } else {
        %>
        <tr>
            <td align="center">로그인한 후 사용하세요.</td>
        </tr>
        <%
            }
        %>
    </table>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
<!DOCTYPE html>
<html>
<head>
<title>로그인</title>
</head>
<body>
  <%@ include file="top.jsp"%>
  <table width="75%" align="center" border>
    <form method="post" action="login_verify.jsp">
      <tr>
        <td>
          <div align="center">아이디</div>
        </td>
        <td>
          <div align="center"><input type="text" name="userID"></div>
        </td>
      </tr>
      <tr>
        <td>
          <div align="center">패스워드</div>
        </td>
        <td>
          <div align="center"><input type="password" name="userPassword"></div>
        </td>
      </tr>
      <tr>
        <td colspan=2>
          <div align="center">
            <input type="submit" name="submit" value="로그인">
            <input type="reset" value="취소">
          </div>
        </td>
      </tr>
    </form>
  </table>
</body>
</html>
```

```
<%@ page import="org.bson.Document"%>
<%@ page import="com.mongodb.MongoClient"%>
<%@ page import="com.mongodb.client.MongoCollection"%>
<%@ page import="com.mongodb.client.MongoDatabase"%>
<%@ page import="com.mongodb.client.model.Filters"%>
<%@ page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
<%

    String userID = request.getParameter("userID");
    String userPassword = request.getParameter("userPassword");

    MongoClient mongoClient = new MongoClient("localhost", 27017); // MongoDB 연결
    MongoDatabase db = mongoClient.getDatabase("testdb"); // 데이터베이스 선택
    MongoCollection<Document> collection = db.getCollection("member"); // 컬렉션 선택

    Document myResultSet = collection.find(Filters.and(Filters.eq("id", userID), Filters.eq("pwd", userPassword))).first();

    if (myResultSet!=null) {
        session.setAttribute("user", userID);
        session.setAttribute("name", myResultSet.get("name"));
        response.sendRedirect("main.jsp");
    } else {

%>

<script>
    alert("사용자아이디 혹은 암호가 틀렸습니다.");
    location.href = "login.jsp";
</script>
<%
    }
    mongoClient.close();
%>

<!DOCTYPE html>
<html>
<head>
<title>Insert title here</title>
</head>
<body>
</body>
</html>
```



```
<%@ page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
<% session.invalidate(); %>

<script>
    alert("로그아웃 되었습니다.");
    location.href="main.jsp";
</script>

<!DOCTYPE html>
<html>
<head>
<title>Insert title here</title>
</head>
<body>
</body>
</html>
```

```
<%@ page contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
<html>
<head>
<title>회원가입</title>
</head>
<body>
<%@ include file="top.jsp"%>
<table width="75%" align="center" border>
  <form method="post" action="signUp_verify.jsp">
    <tr>
      <th>아이디</th>
      <td><input type="text" name="m_id" size="20"></td>
    </tr>
    <tr>
      <th>패스워드</th>
      <td><input type="password" name="m_pwd" size="20"></td>
    </tr>
    <tr>
      <th>이름</th>
      <td><input type="text" name="m_name" size="20"></td>
    </tr>
    <tr>
      <th>주소</th>
      <td><input type="text" name="m_addr" size="50"></td>
    </tr>
    <tr>
      <th>학과</th>
      <td><input type="text" name="m_major" size="20"></td>
    </tr>
    <tr>
      <td colspan="2" align="center">
        <input type="submit" value="가입">
        <a href="main.jsp"><input type="button" value="취소"></a>
      </td>
    </tr>
  </form>
</table>
</body>
</html>
```

```
<%@page import="org.bson.Document"%>
<%@page import="com.mongodb.MongoClient"%>
<%@page import="com.mongodb.client.MongoDatabase"%>
<%@page import="com.mongodb.client.MongoCollection"%>
<%@ page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
<html>
<head>
<title>Insert title here</title>
</head>
<body>
    <%
        String m_id = new String(request.getParameter("m_id").getBytes("8859_1"), "utf-8");
        String m_pwd = new String(request.getParameter("m_pwd"));
        String m_name = new String(request.getParameter("m_name").getBytes("8859_1"), "utf-8");
        String m_addr = new String(request.getParameter("m_addr").getBytes("8859_1"), "utf-8");
        String m_major = new String(request.getParameter("m_major").getBytes("8859_1"), "utf-8");

        MongoClient mongoClient = new MongoClient("localhost", 27017); // MongoDB 연결
        MongoDatabase db = mongoClient.getDatabase("testdb"); // 데이터베이스 선택
        MongoCollection<Document> collection = db.getCollection("member"); // 컬렉션 선택

        Document query = new Document("id", m_id)
            .append("name", m_name)
            .append("addr", m_addr)
            .append("major", m_major)
            .append("pwd", m_pwd);

        try{
            collection.insertOne(query);
            mongoClient.close();
        }
    %>
```

← 컬렉션 구조 생성

```
{
    "id" : m_id,
    "name" : m_name,
    "addr" : m_addr,
    "major" : m_major,
    "pwd" : m_pwd
}
```

```
<script>
    alert("회원가입이 완료되었습니다.");
    location.href="main.jsp";
</script>
<%
    }catch(Exception e){
%>
<script>
    alert("회원가입에 문제가 발생했습니다.");
</script>
<%
    mongoClient.close();
    }
%>
</body>
</html>
```

```
<%@page import="com.mongodb.MongoClient"%>
<%@page import="com.mongodb.client.MongoDatabase"%>
<%@page import="com.mongodb.client.MongoCollection"%>
<%@page import="com.mongodb.client.model.Filters"%>
<%@page import="org.bson.Document"%>
<%@ page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>

<html>
<head>
<title>사용자 정보 수정</title>
</head>
<body>
    <%@ include file="top.jsp"%>
    <%
        if (session_id == null) {
            response.sendRedirect("login.jsp");
        } else {

MongoClient mongoClient = new MongoClient("localhost", 27017); // MongoDB 연결
MongoDatabase db = mongoClient.getDatabase("testdb"); // 데이터베이스 선택
MongoCollection<Document> collection = db.getCollection("member"); // 컬렉션 선택

Document mem = new Document();

        try {
            mem = collection.find(Filters.eq("id", session_id)).first();
        } catch (Exception ex) {
            System.out.println("Exception" + ex);
        }
    %>
```

```

<table width="75%" align="center" border>
  <form method="post" action="update_verify.jsp">
    <input type="hidden" name="m_id" size="30" value="<%=session_id%>">
    <tr>
      <th>이름</th>
      <td><input type="text" name="m_name" size="20"
        value="<%=mem.get("name") %>"></td>
    </tr>
    <tr>
      <th>주소</th>
      <td><input type="text" name="m_addr" size="50"
        value="<%=mem.get("addr") %>"></td>
    </tr>
    <tr>
      <th>패스워드</th>
      <td><input type="password" name="m_pwd" size="20"
        value="<%=mem.get("pwd") %>"></td>
    </tr>
    <tr>
      <th>전공</th>
      <td><input type="text" name="m_major" size="20"
        value="<%=mem.get("major") %>"></td>
    </tr>
    <tr>
      <td colspan="2" align="center">
        <input type="submit" value="수정">
        <a href="withdraw_verify.jsp"><input type="button" value="회원탈퇴"></a>
      </td>
    </tr>
  </table>
</form>
<%
  MongoClient.close();
%>
</body>
</html>

```

```
<%@page import="org.bson.Document"%>
<%@page import="com.mongodb.MongoClient"%>
<%@page import="com.mongodb.client.MongoDatabase"%>
<%@page import="com.mongodb.client.MongoCollection"%>
<%@page import="com.mongodb.client.model.Filters"%>
<%@ page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
<html>
<head>
<title>Insert title here</title>
</head>
<body>
    <%
        String m_id = request.getParameter("m_id");
        String m_name = new String(request.getParameter("m_name").getBytes("8859_1"), "utf-8");
        String m_addr = new String(request.getParameter("m_addr").getBytes("8859_1"), "utf-8");
        String m_pwd = new String(request.getParameter("m_pwd"));
        String m_major = new String(request.getParameter("m_major").getBytes("8859_1"), "utf-8");

        MongoClient mongoClient = new MongoClient("localhost", 27017); // MongoDB 연결
        MongoDatabase db = mongoClient.getDatabase("testdb"); // 데이터베이스 선택
        MongoCollection<Document> collection = db.getCollection("member"); // 컬렉션 선택

        Document query = new Document("name", m_name)
            .append("addr", m_addr)
            .append("major", m_major)
            .append("pwd", m_pwd);

        try {
            collection.updateOne(Filters.eq("id", m_id), new Document("$set", query));
        }
    %>
```

\$set을 사용하지 않으면
query에 사용한 데이터로만
덮어쓰기 되어 id필드가 사라짐

```
%>
<script>
    alert("회원 정보가 수정되었습니다.");
    location.href="update.jsp";
</script>
<%
    mongoClient.close();
} catch (Exception ex) {
    System.err.println("Exception: " + ex.getMessage());
    mongoClient.close();
}
%>
</body>
</html>
```



```
<%@page import="org.bson.Document"%>
<%@page import="com.mongodb.MongoClient"%>
<%@page import="com.mongodb.client.MongoDatabase"%>
<%@page import="com.mongodb.client.MongoCollection"%>
<%@page import="com.mongodb.client.model.Filters"%>
<%@ page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
<html>
<head>
<title>Insert title here</title>
</head>
<body>
    <%
        String m_id = session.getAttribute("user").toString();

        MongoClient mongoClient = new MongoClient("localhost", 27017); // MongoDB 연결
        MongoDatabase db = mongoClient.getDatabase("testdb"); // 데이터베이스 선택
        MongoCollection<Document> collection = db.getCollection("member"); // 컬렉션 선택

        try {
            collection.deleteOne(Filters.eq("id", m_id));
            session.invalidate();
        }

    %>
    <script>
        alert("회원 정보가 삭제되었습니다.");
        location.href="main.jsp";
    </script>
    <%
        mongoClient.close();
    } catch (Exception ex) {
        System.err.println("Exception: " + ex.getMessage());
        mongoClient.close();
    }
    %>
</body>
</html>
```