**Initialisation Mongo**

C:\Users\LENOVO>cd C:\Program Files\MongoDB\Server\4.2\bin

C:\Program Files\MongoDB\Server\4.2\bin>mongod

2020-04-18T15:40:09.573+0100 I CONTROL [main] Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'

2020-04-18T15:40:09.577+0100 W ASIO [main] No TransportLayer configured during NetworkInterface startup

2020-04-18T15:40:09.578+0100 I CONTROL [initandlisten] MongoDB starting : pid=2424 port=27017 dbpath=C:\data\db\ 64-bit host=DESKTOP-CIOT8A7

2020-04-18T15:40:09.578+0100 I CONTROL [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2

2020-04-18T15:40:09.578+0100 I CONTROL [initandlisten] db version v4.2.5

2020-04-18T15:40:09.578+0100 I CONTROL [initandlisten] git version: 2261279b51ea13df08ae708ff278f0679c59dc32

2020-04-18T15:40:09.578+0100 I CONTROL [initandlisten] allocator: tcmalloc

2020-04-18T15:40:09.578+0100 I CONTROL [initandlisten] modules: none

2020-04-18T15:40:09.578+0100 I CONTROL [initandlisten] build environment:

2020-04-18T15:40:09.578+0100 I CONTROL [initandlisten] distmod: 2012plus

2020-04-18T15:40:09.578+0100 I CONTROL [initandlisten] distarch: x86\_64

2020-04-18T15:40:09.578+0100 I CONTROL [initandlisten] target\_arch: x86\_64

2020-04-18T15:40:09.579+0100 I CONTROL [initandlisten] options: {}

2020-04-18T15:40:09.581+0100 I STORAGE [initandlisten] exception in initAndListen: NonExistentPath: Data directory C:\data\db\ not found., terminating

2020-04-18T15:40:09.581+0100 I NETWORK [initandlisten] shutdown: going to close listening sockets...

2020-04-18T15:40:09.581+0100 I - [initandlisten] Stopping further Flow Control ticket acquisitions.

2020-04-18T15:40:09.581+0100 I CONTROL [initandlisten] now exiting

2020-04-18T15:40:09.581+0100 I CONTROL [initandlisten] shutting down with code:100

C:\Program Files\MongoDB\Server\4.2\bin>mongo

MongoDB shell version v4.2.5

connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb

Implicit session: session { "id" : UUID("7d0cd230-c308-46bd-8393-347f4b6c60b2") }

MongoDB server version: 4.2.5

Server has startup warnings:

2020-04-18T10:24:00.963+0100 I CONTROL [initandlisten]

2020-04-18T10:24:00.963+0100 I CONTROL [initandlisten] \*\* WARNING: Access control is not enabled for the database.

2020-04-18T10:24:00.963+0100 I CONTROL [initandlisten] \*\* Read and write access to data and configuration is unrestricted.

2020-04-18T10:24:00.971+0100 I CONTROL [initandlisten]

---

Enable MongoDB's free cloud-based monitoring service, which will then receive and display

metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you

and anyone you share the URL with. MongoDB may use this information to make product

improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()

To permanently disable this reminder, run the following command: db.disableFreeMonitoring()

---

>

**Create a database called "contact"**

**> use contact**

**switched to db contact**

**> db**

**contact**

**Create a collection called "contactlist"**

**> db.createCollection('contactlist')**

**{ "ok" : 1 }**

**> show collections**

**contactlist**

**Insert 5 persons in "contactlist" :**

**> db.contactlist.insert([{"Last name":"Ben Lahmer","First name":"Fares","Email":"fares@gmail.com","age":"26"},{"Last name":"Kefi","First name":"Seif","Email":"kefi@gmail.com","age":"15"},{"Last name":"Fatnassi","First name":"Sarra","Email":"sarra@gmail.com","age":"40"},{"Last name":"Ben Yahia","First name":"Rym","age":"4"},{"Last name":"Cherif","First name":"Sami","age":"3"}])**

**BulkWriteResult({**

**"writeErrors" : [ ],**

**"writeConcernErrors" : [ ],**

**"nInserted" : 5,**

**"nUpserted" : 0,**

**"nMatched" : 0,**

**"nModified" : 0,**

**"nRemoved" : 0,**

**"upserted" : [ ]**

**})**

**Display all the contact lis**

**> db.contactlist.find()**

**{ "\_id" : ObjectId("5e9b122e7710199d3a25bb1e"), "Last name" : "Ben Lahmer", "First name" : "Fares", "Email" : "fares@gmail.com", "age" : "26" }**

**{ "\_id" : ObjectId("5e9b122e7710199d3a25bb1f"), "Last name" : "Kefi", "First name" : "Seif", "Email" : "kefi@gmail.com", "age" : "15" }**

**{ "\_id" : ObjectId("5e9b122e7710199d3a25bb20"), "Last name" : "Fatnassi", "First name" : "Sarra", "Email" : "sarra@gmail.com", "age" : "40" }**

**{ "\_id" : ObjectId("5e9b122e7710199d3a25bb21"), "Last name" : "Ben Yahia", "First name" : "Rym", "age" : "4" }**

**{ "\_id" : ObjectId("5e9b122e7710199d3a25bb22"), "Last name" : "Cherif", "First name" : "Sami", "age" : "3" }**

**Display all the information about only one person using his'  
id**

**>db.contactlist.find({"\_id":ObjectId("5e9b122e7710199d3a25bb1e")},{"Last name":1,"First name":1,"Email":1,"age":1})**

**{ "\_id" : ObjectId("5e9b122e7710199d3a25bb1e"), "Last name" : "Ben Lahmer", "First name" : "Fares", "Email" : "fares@gmail.com", "age" : "26" }**

**Display all the contact list having age>18**

**> db.contactlist.find({"age":{$gt:18}})**

**{ "\_id" : ObjectId("5e9b122e7710199d3a25bb1e"), "Last name" : "Ben Lahmer", "First name" : "Fares", "Email" : "fares@gmail.com", "age" : "26" }**

**{ "\_id" : ObjectId("5e9b122e7710199d3a25bb20"), "Last name" : "Fatnassi", "First name" : "Sarra", "Email" : "sarra@gmail.com", "age" : "40" }**

**Display all the contact list having age>18 and name'  
containing "ah"**

**> db.contactlist.find({"$and"  : [{"age":{$gt:18}}, {"Last name":{$regex : ".\*ah.\*"}} ] )**

**{ "\_id" : ObjectId("5e9b122e7710199d3a25bb1e"), "Last name" : "Ben Lahmer", "First name" : "Fares", "Email" : "fares@gmail.com", "age" : "26" }**

**Change the contact first name of "kefi Seif" by'  
"Kefi Anis"**

**> db.contactlist.find.findAndModify({**

**query: { "First name" : "Seif" },**

**sort: { "First name": 1 },**

**update: { $rename: { " First name ": " Anis" } },**

**new : true**

**});**

**{ "\_id" : ObjectId("5e9b122e7710199d3a25bb1f"), "Last name" : "Kefi", "First name" : "Anis", "Email" : "kefi@gmail.com", "age" : "15" }**

**Delete the the contact list having age < 5**

**> db.contactlist.remove({"age":{$lt:5}})**

**WriteResult({ "nRemoved" : 2 })**

**Display all the contact list**

**> db.contactlist.find()**

**{ "\_id" : ObjectId("5e9b122e7710199d3a25bb1e"), "Last name" : "Ben Lahmer", "First name" : "Fares", "Email" : "fares@gmail.com", "age" : "26" }**

**{ "\_id" : ObjectId("5e9b122e7710199d3a25bb1f"), "Last name" : "Kefi", "First name" : "Anis", "Email" : "kefi@gmail.com", "age" : "15" }**

**{ "\_id" : ObjectId("5e9b122e7710199d3a25bb20"), "Last name" : "Fatnassi", "First name" : "Sarra", "Email" : "sarra@gmail.com", "age" : "40" }**