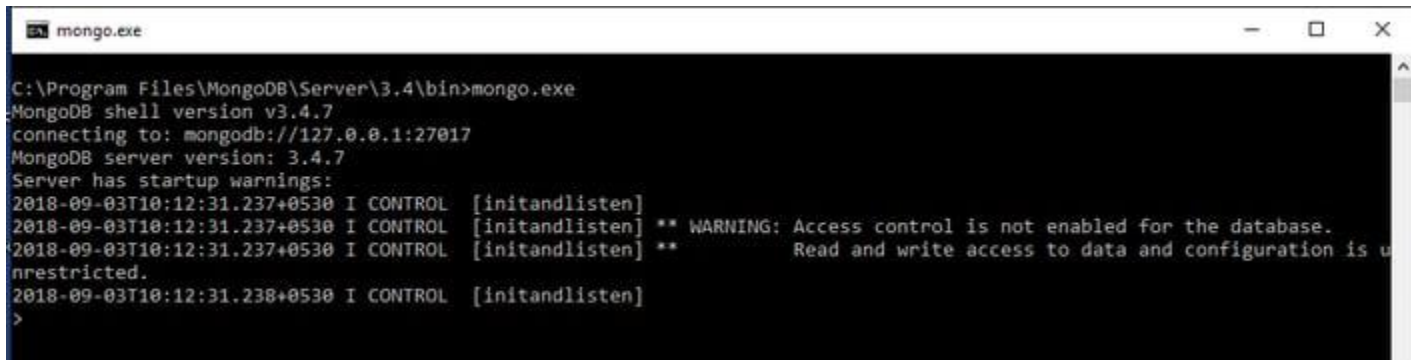


1. Install MongoDB On Windows

To install MongoDB on Windows, first download the latest release of MongoDB from <https://www.mongodb.org/downloads>. Make sure you get correct version of MongoDB depending upon your Windows version.

2. Running mongoDB

Go to location where mongo DB is installed “C:\Program Files\MongoDB\Server\3.4\bin” and open command prompt at that location and execute following command in command prompt.

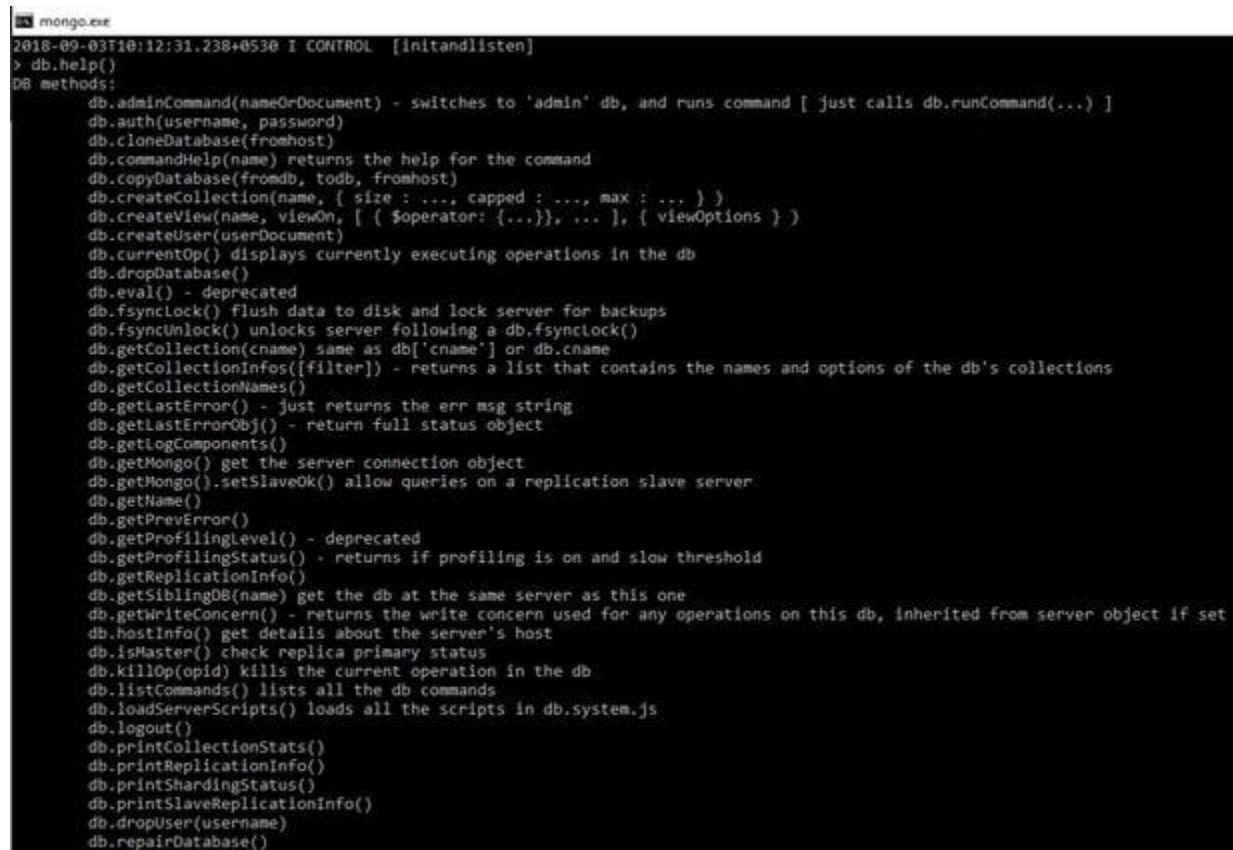


```
C:\Program Files\MongoDB\Server\3.4\bin>mongo.exe
MongoDB shell version v3.4.7
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.4.7
Server has startup warnings:
2018-09-03T10:12:31.237+0530 I CONTROL [initandlisten]
2018-09-03T10:12:31.237+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2018-09-03T10:12:31.237+0530 I CONTROL [initandlisten] **      Read and write access to data and configuration is u
nrestricted.
2018-09-03T10:12:31.238+0530 I CONTROL [initandlisten]
>
```

3. MongoDB Help

To get a list of commands, type `db.help()` in MongoDB client. This will give you a list of commands as shown in the following screenshot.

`db.help`



```
mongo.exe
2018-09-03T10:12:31.238+0530 I CONTROL [initandlisten]
> db.help()
DB methods:
  db.adminCommand(nameOrDocument) - switches to 'admin' db, and runs command [ just calls db.runCommand(...) ]
  db.auth(username, password)
  db.cloneDatabase(fromhost)
  db.commandHelp(name) returns the help for the command
  db.copyDatabase(fromdb, todb, fromhost)
  db.createCollection(name, { size : ..., capped : ..., max : ... } )
  db.createView(name, viewOn, [ { $operator: {...}}, ... ], { viewOptions } )
  db.createUser(userDocument)
  db.currentOp() displays currently executing operations in the db
  db.dropDatabase()
  db.eval() - deprecated
  db.fsyncLock() flush data to disk and lock server for backups
  db.fsyncUnlock() unlocks server following a db.fsyncLock()
  db.getCollection(cname) same as db['cname'] or db.cname
  db.getCollectionInfos([filter]) - returns a list that contains the names and options of the db's collections
  db.getCollectionNames()
  db.getLastErrorMessage() - just returns the err msg string
  db.getLastStatusObj() - return full status object
  db.getLogComponents()
  db.getMongo() get the server connection object
  db.getMongo().setSlaveOk() allow queries on a replication slave server
  db.getName()
  db.getPrevError()
  db.getProfilingLevel() - deprecated
  db.getProfilingStatus() - returns if profiling is on and slow threshold
  db.getReplicationInfo()
  db.getSiblingDB(name) get the db at the same server as this one
  db.getWriteConcern() - returns the write concern used for any operations on this db, inherited from server object if set
  db.hostInfo() get details about the server's host
  db.isMaster() check replica primary status
  db.killOp(opid) kills the current operation in the db
  db.listCommands() lists all the db commands
  db.loadServerScripts() loads all the scripts in db.system.js
  db.logout()
  db.printCollectionStats()
  db.printReplicationInfo()
  db.printShardingStatus()
  db.printSlaveReplicationInfo()
  db.dropUser(username)
  db.repairDatabase()
```

4. Show All Databases

Use below command to get list of all databases.

`show dbs`

```
db.version() current version of the
> show dbs
charts          0.006GB
chartts1        0.001GB
comp            0.005GB
drilldown       0.006GB
export          0.005GB
export1         0.005GB
incomp          0.005GB
local           0.000GB
page            0.007GB
relationship    0.005GB
>
```

5. Create new database

To create a new database execute the following command.

`use DATABASE_NAME`

```
mongo.exe
> use myTestDB
switched to db myTestDB
>
```

6. Know your current selected database

To know your current working/selected database execute the following command
`db`

```
> db
myTestDB
>
```

7. Drop database

To drop the database execute following command, this will drop the selected database
`db.dropDatabase()`

```
> db.dropDatabase()
{ "dropped" : "myTestDB", "ok" : 1 }
>
```

8. Create collection

To create the new collection execute the following commands
`db.createCollection(name)`

```
> db.createCollection("Employee");
{ "ok" : 1 }
>
```

9. To check collections list

To get the list of collections created execute the following command

Show collections

```
> show collections
Employee
>
```

10. Drop collection

To drop the selected collection execute the following command

db.COLLECTION_NAME.drop()

```
> show collections
Department
Employee
> db.Department.drop()
true
> show collections
Employee
>
```

11. Insert document in collection

>db.COLLECTION_NAME.insert(document)

To insert single document in selected collection execute the following command

```
> db.Employee.insert({name: 'Emp1',address: 'Pune'})
WriteResult({ "nInserted" : 1 })
> db.Employee.insert({name: 'Emp2',address: 'Mumbai'})
WriteResult({ "nInserted" : 1 })
>
```

To insert multiple documents in selected collection execute following command

```
> db.Employee.insertMany([{name : 'Emp1', address:'Pune'}, {name: 'Emp2', address: 'Mumbai'}])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("5b920da763cdcea45e0ac334"),
    ObjectId("5b920da763cdcea45e0ac335")
  ]
}
```

12. Get collection document

To get the list documents in collection execute the following command

```
> db.COLLECTION_NAME.find()
```

```
> db.Employee.find().pretty()
{
  "_id" : ObjectId("5b920c9863cdcea45e0ac332"),
  "name" : "Emp1",
  "address" : "Pune"
}
{
  "_id" : ObjectId("5b920c9d63cdcea45e0ac333"),
  "name" : "Emp2",
  "address" : "Mumbai"
}
```

13. Update document

To update the document in collection execute the following command

```
>db.COLLECTION_NAME.update(SELECTION_CRITERIA, UPDATED_DATA)
```

```
> db.Employee.find().pretty()
{
  "_id" : ObjectId("5b920da763cdcea45e0ac334"),
  "name" : "Emp1",
  "address" : "Pune"
}
{
  "_id" : ObjectId("5b920da763cdcea45e0ac335"),
  "name" : "Emp2",
  "address" : "Mumbai"
}
> db.Employee.update({'name':'Emp1'},{$set: {'name':'New Emp1'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Employee.find().pretty()
{
  "_id" : ObjectId("5b920da763cdcea45e0ac334"),
  "name" : "New Emp1",
  "address" : "Pune"
}
{
  "_id" : ObjectId("5b920da763cdcea45e0ac335"),
  "name" : "Emp2",
  "address" : "Mumbai"
}
>
```

14. Save document

To save document in collection execute the following command

```
>db.COLLECTION_NAME.save({_id:ObjectId(),NEW_DATA})
```

```
> db.Employee.save({_id: new ObjectId("5b920da763cdcea45e0ac337"), name: "Emp3", address: "Banglore"});
WriteResult({
  "nMatched" : 0,
  "nUpserted" : 1,
  "nModified" : 0,
  "_id" : ObjectId("5b920da763cdcea45e0ac337")
})
> db.Employee.find().pretty()
{
  "_id" : ObjectId("5b920da763cdcea45e0ac334"),
  "name" : "New Emp1",
  "address" : "Pune"
}
{
  "_id" : ObjectId("5b920da763cdcea45e0ac335"),
  "name" : "Emp2",
  "address" : "Mumbai"
}
{
  "_id" : ObjectId("5b920da763cdcea45e0ac337"),
  "name" : "Emp3",
  "address" : "Banglore"
}
>
```


15. Delete document

To delete document in selected collection execute the following command

```
>db.COLLECTION_NAME.remove(DELETION_CRITTERIA)
```

```
> db.Employee.find().pretty()
{
  "_id" : ObjectId("5b920da763cdcea45e0ac334"),
  "name" : "New Emp1",
  "address" : "Pune"
}
{
  "_id" : ObjectId("5b920da763cdcea45e0ac335"),
  "name" : "Emp2",
  "address" : "Mumbai"
}
{
  "_id" : ObjectId("5b920da763cdcea45e0ac337"),
  "name" : "Emp3",
  "address" : "Banglore"
}
> db.Employee.remove({'name':'Emp3'})
WriteResult({ "nRemoved" : 1 })
> db.Employee.find().pretty()
{
  "_id" : ObjectId("5b920da763cdcea45e0ac334"),
  "name" : "New Emp1",
  "address" : "Pune"
}
{
  "_id" : ObjectId("5b920da763cdcea45e0ac335"),
  "name" : "Emp2",
  "address" : "Mumbai"
}
>
```

If you would like for more details on the content of this blog, please comment on the section given below and I would try my best to answer your all queries.