

# 1. Set up your mongod

1. If you are using linux or mac, download and install mongod and open mongod in your terminal. Download mongod.

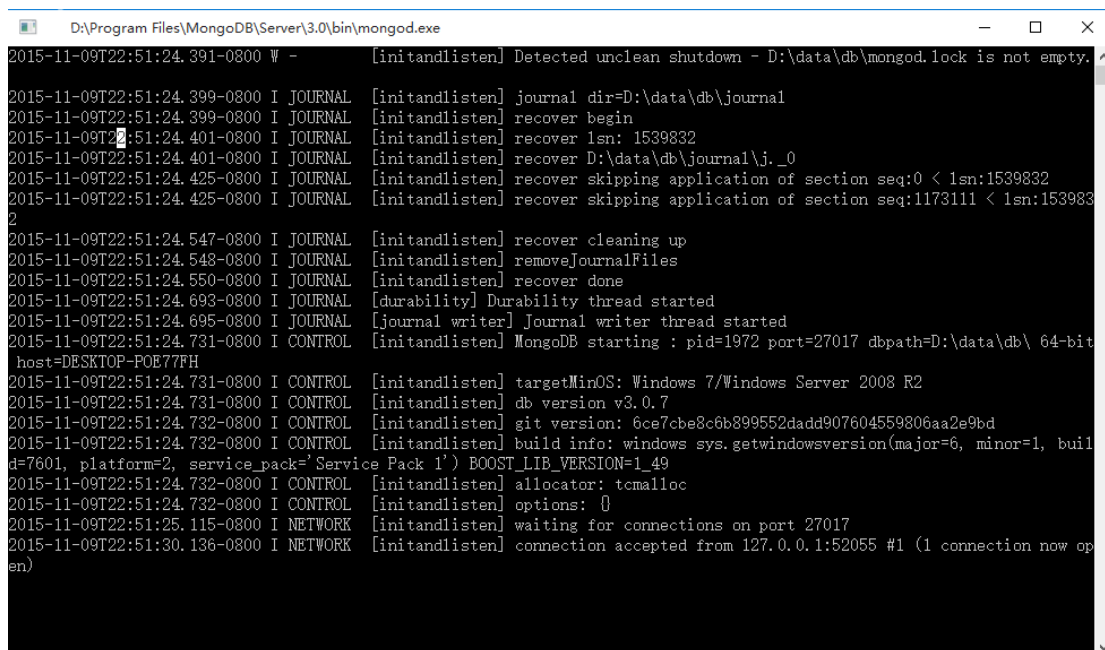
<https://www.mongodb.org/downloads#production>

2. If you are using windows, open mongod at first

D:\Program Files\MongoDB\Server\3.0\bin

Open mongod, which means you open mongo database service and wait users to connect to mongo.

If you get a problem “mongod data/db does not exist”, you can create a folder “c:/data/db” (or “d:/data/db”, it depends on which disk drive the mongo is installed in) or **mongod.exe --dbpath "your data path"** to reset your data path.



```
D:\Program Files\MongoDB\Server\3.0\bin\mongod.exe
2015-11-09T22:51:24.391-0800 W - [initandlisten] Detected unclean shutdown - D:\data\db\mongod.lock is not empty.
2015-11-09T22:51:24.399-0800 I JOURNAL [initandlisten] journal dir=D:\data\db\journal
2015-11-09T22:51:24.399-0800 I JOURNAL [initandlisten] recover begin
2015-11-09T22:51:24.401-0800 I JOURNAL [initandlisten] recover lsn: 1539832
2015-11-09T22:51:24.401-0800 I JOURNAL [initandlisten] recover D:\data\db\journal\j._0
2015-11-09T22:51:24.425-0800 I JOURNAL [initandlisten] recover skipping application of section seq:0 < lsn:1539832
2015-11-09T22:51:24.425-0800 I JOURNAL [initandlisten] recover skipping application of section seq:1173111 < lsn:1539832
2015-11-09T22:51:24.547-0800 I JOURNAL [initandlisten] recover cleaning up
2015-11-09T22:51:24.548-0800 I JOURNAL [initandlisten] removeJournalFiles
2015-11-09T22:51:24.550-0800 I JOURNAL [initandlisten] recover done
2015-11-09T22:51:24.693-0800 I JOURNAL [durability] Durability thread started
2015-11-09T22:51:24.695-0800 I JOURNAL [journal writer] Journal writer thread started
2015-11-09T22:51:24.731-0800 I CONTROL [initandlisten] MongoDB starting : pid=1972 port=27017 dbpath=D:\data\db\ 64-bit
host=DESKTOP-POE77FH
2015-11-09T22:51:24.731-0800 I CONTROL [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2015-11-09T22:51:24.731-0800 I CONTROL [initandlisten] db version v3.0.7
2015-11-09T22:51:24.732-0800 I CONTROL [initandlisten] git version: 6ce7cbe8c6b899552dadd907604559806aa2e9bd
2015-11-09T22:51:24.732-0800 I CONTROL [initandlisten] build info: windows sys.getwindowsversion(major=6, minor=1, build=7601, platform=2, service_pack='Service Pack 1') BOOST_LIB_VERSION=1_49
2015-11-09T22:51:24.732-0800 I CONTROL [initandlisten] allocator: tcmalloc
2015-11-09T22:51:24.732-0800 I CONTROL [initandlisten] options: {}
2015-11-09T22:51:25.115-0800 I NETWORK [initandlisten] waiting for connections on port 27017
2015-11-09T22:51:30.136-0800 I NETWORK [initandlisten] connection accepted from 127.0.0.1:52055 #1 (1 connection now open)
```

3. Set mongo bin file in your computer environment path

Find your mongo bin folder then add that after your environment path(right click on my computer then attribute and then advanced setting and then environment variable).

4. Then open cmd, in cmd use command “mongo”, now you are connected to mongod.

```
C:\WINDOWS\system32\cmd.exe - mongo
at connect (src/mongo/shell/mongo.js:179:14)
at (connect):1:6 at src/mongo/shell/mongo.js:179
exception: connect failed

C:\Users\zhihan>mongod
2015-11-09T22:50:49.475-0800 I STORAGE [initandlisten] exception in initAndListen: 29 Data directory C:\data\db\ not fo
und., terminating
2015-11-09T22:50:49.476-0800 I CONTROL [initandlisten] dbexit: rc: 100

C:\Users\zhihan>mongod
2015-11-09T22:51:10.411-0800 I STORAGE [initandlisten] exception in initAndListen: 29 Data directory C:\data\db\ not fo
und., terminating
2015-11-09T22:51:10.412-0800 I CONTROL [initandlisten] dbexit: rc: 100

C:\Users\zhihan>mongos
BadValue error: no args for --configdb
try 'mongos --help' for more information

C:\Users\zhihan>mongo
MongoDB shell version: 3.0.7
connecting to: test
> show dbs
local 0.078GB
test 0.078GB
> use cs336
switched to db cs336
>
```

5. Show all database instances in the database

**show dbs**

6. Create a new database instance

**use cs336**

Mind that now you are under cs336 database instance.

7. Under the database instance, create two collections

**db.createCollection("unlabel\_review")**

**db.createCollection("unlabel\_review\_after\_splitting")**

After that, if you use “show collections”, you can find three new collections have been created.

You can check those two collections are all empty by using command find() or count(), find will return all contents in the collection and count will return how many documents in the collection(the number of documents).

using **db.unlabel\_review.count()**, **db.unlabel\_review\_after\_splitting.count()**, or replace “count” with “find”. The results are as follows:

```
zhihan@zhihan-pc ~  
  
> db.createCollection("unlabel_review")  
{ "ok" : 1 }  
> db.createCollection("unlabel_review_after_splitting")  
{ "ok" : 1 }  
> db.unlabel_review.count()  
0  
> db.unlabel_review_after_splitting.count()  
0  
> 
```

8. Import json data in your database collections, use two json files to import the data in your database.

Open a new cmd console and then use mongoimport command.

`mongoimport -d cs336 -c unlabel_review UnlabelReview.json`

`mongoimport -d cs336 -c unlabel_review_after_splitting UnlabelReviewAfterSplitting.json`

```
zhihan@zhihan-pc ~/Dropbox/cs336_mongo  
  
cs336_mongo> mongoimport -d cs336 -c unlabel_review UnlabelReview.json  
connected to: 127.0.0.1  
Thu Nov 12 19:17:47.991 check 9 500  
Thu Nov 12 19:17:48.000 imported 500 objects  
cs336_mongo> mongoimport -d cs336 -c unlabel_review_after_splitting UnlabelReviewAfterSplitting.json  
connected to: 127.0.0.1  
Thu Nov 12 19:18:21.432 check 9 500  
Thu Nov 12 19:18:21.441 imported 500 objects  
cs336_mongo> 
```

Then check data again.

```
db.unlabel_review.count()
```

```
db.unlabel_review_after_splitting.count()
```

```
zhihan@zhihan-pc ~
> db.unlabel_review.count()
500
> db.unlabel_review_after_splitting.count()
500
>
```

We have import 500 documents in each unlabeled collection.

9. If want to check what it contain in each collection, you can use find() function.

```
zhihan@zhihan-pc ~
> db.unlabel_review.count()
500
> db.unlabel_review_after_splitting.count()
500
> db.unlabel_review.find()
{ "_id" : ObjectId("564528e0d40a732e502f33b1"), "review" : "\"Watching Time Chas
ers, it obvious that it was made by a bunch of friends. Maybe they were sitting
around one day in film school and said, \\\"Hey, let's pool our money together a
nd make a really bad movie!\\\" Or something like that. What ever they said, the
y still ended up making a really bad movie--dull story, bad script, lame acting,
poor cinematography, bottom of the barrel stock music, etc. All corners were cu
t, except the one that would have prevented this film's release. Life's like tha
t.\""}, "id" : "\"9999_0\""}
{ "_id" : ObjectId("564528e0d40a732e502f33b2"), "review" : "\"I saw this film ab
out 20 years ago and remember it as being particularly nasty. I believe it is ba
sed on a true incident: a young man breaks into a nurses' home and rapes, tortur
es and kills various women.It is in black and white but saves the colour for one
shocking shot.At the end the film seems to be trying to make some political sta
tement but it just comes across as confused and obscene.Avoid.\""}, "id" : "\"450
57_0\""}
{ "_id" : ObjectId("564528e0d40a732e502f33b3"), "review" : "\"Minor SpoilersIn N
```

10. Connect mongo from other programming language. Mongodb support js as default programming language. If you want to connect it from java, refer to following link.

<https://docs.mongodb.org/getting-started/java/>

other programming languages can be found in the following link.

<https://docs.mongodb.org/ecosystem/drivers/>

unlabel\_review is raw data. unlabel\_review\_after\_splitting is processed data, containing words in reviews and its corresponding frequency(count).

Its format is as follows:

```
{
  "_id" : ObjectId("5644077d0b539b0990979d13"),
  "review" : [
    {
      "count" : 1,
      "word" : "everyone"
    },
    {
      "count" : 1,
      "word" : "version"
    },
    {
      "count" : 2,
      "word" : "entertaining"
    },
    {
      "count" : 3,
      "word" : "classic"
    }
  ]
}
```

```
}],  
  "id" : "2381_9"  
}
```

As you see, each review has been split to words and its frequency. Because the number of words may be different in each review, it is not a good way to store such a data structure in classic relational database system.

review: customers' reviews about a movie

id: the review's id

You do not need to concern about “\_id”, it's a special id automatically generated by mongodb for each document (record).