NIELIT Gorakhpur

Course name: A level **SUBJECT: DATABASE TECHNOLOGIES**

Topic: MongoDB **DATE: 09/06/2020**

mongoimport command to Import Collection

mongoimport command is used to restore (import) a database from a backup(export) taken with mongoexport command.

Importing from JSON File

To Import a collection from a JSON file, the syntax is as under

Syntax

mongoimport --db DB_NAME --collection COLLECTION_name --type=json --file=Name-of-file-to-import

where,

DB_NAME – Name of the Database of the Collection to be exported

COLLECTION_name - Name of Collection of DB_NAME to be exported

Type –JSON, it is optional. System by default take it as a JSON

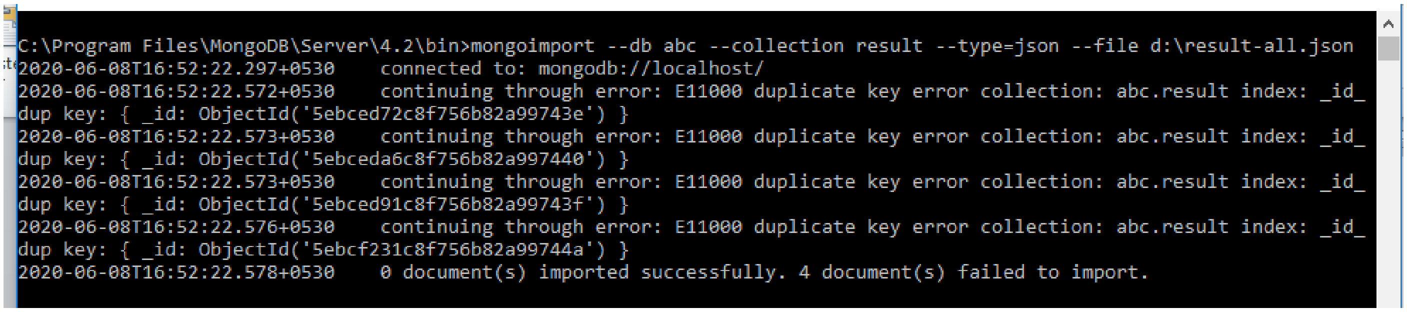
Name-of-file-to-import – Name and path of the JSON file to be imported (restore).

Note: We may specify a new collection name as well as a new database name while importing CSV or JSON file to the collection file.

Example :

1. Let’s assume that our backup file name is result-all.JSON which is in D:\ and the db name is abc and collection name is result.

mongoimport --db abc --collection result --type=json --file d:\result-all.json

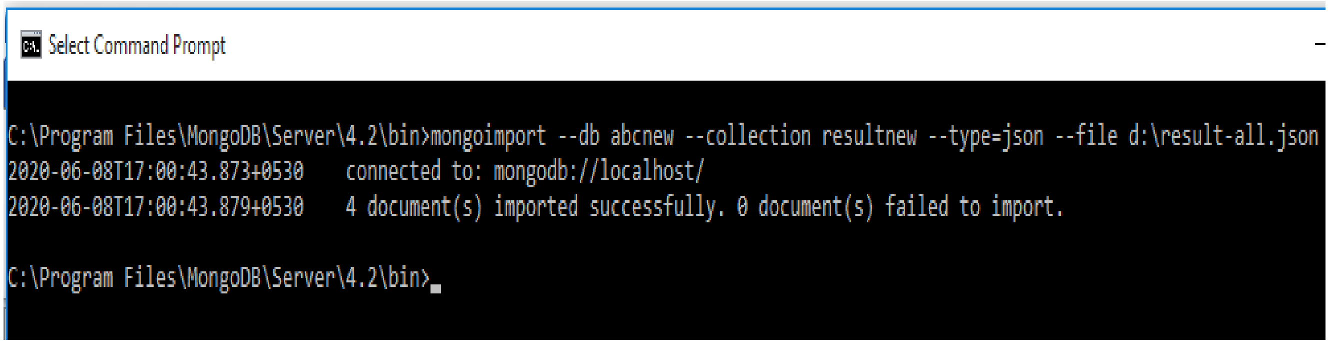


```
C:\Program Files\MongoDB\Server\4.2\bin>mongoimport --db abc --collection result --type=json --file d:\result-all.json
2020-06-08T16:52:22.297+0530    connected to: mongodb://localhost/
2020-06-08T16:52:22.572+0530    continuing through error: E11000 duplicate key error collection: abc.result index: _id_
dup key: { _id: ObjectId('5ebced72c8f756b82a99743e') }
2020-06-08T16:52:22.573+0530    continuing through error: E11000 duplicate key error collection: abc.result index: _id_
dup key: { _id: ObjectId('5ebceda6c8f756b82a997440') }
2020-06-08T16:52:22.573+0530    continuing through error: E11000 duplicate key error collection: abc.result index: _id_
dup key: { _id: ObjectId('5ebced91c8f756b82a99743f') }
2020-06-08T16:52:22.576+0530    continuing through error: E11000 duplicate key error collection: abc.result index: _id_
dup key: { _id: ObjectId('5ebcf231c8f756b82a99744a') }
2020-06-08T16:52:22.578+0530    0 document(s) imported successfully. 4 document(s) failed to import.
```

In the above example, mongoimport imports the data in the JSON data from the result-all.json file into the collection result in the abc database. The result is showing duplicate key error because the same records are existing in the collection specified.

2. We can give a new name to the collection as well as to database to which a JSON data file is imported. In the code below, the same JSON file of above example is used but imported to abcnew database as resultnew collection.

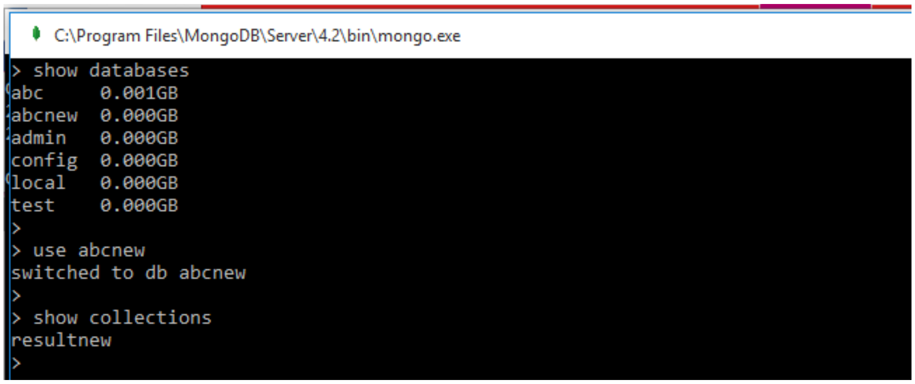
mongoimport --db abcnew --collection resultnew --type=json --file d:\result-all.json



```
C:\Program Files\MongoDB\Server\4.2\bin>mongoimport --db abcnew --collection resultnew --type=json --file d:\result-all.json
2020-06-08T17:00:43.873+0530    connected to: mongodb://localhost/
2020-06-08T17:00:43.879+0530    4 document(s) imported successfully. 0 document(s) failed to import.

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

Now list the databases in MongoDB and collection in the abcnew database



```
> show databases
abc     0.001GB
abcnew  0.000GB
admin   0.000GB
config  0.000GB
local   0.000GB
test    0.000GB
>
> use abcnew
switched to db abcnew
>
> show collections
resultnew
>
```

Importing from CSV File

mongoimport may also be used to import the csv formatted data into the collection. **–headerline option** is used to instructs mongoimport to determine the name of the fields using the first line in the CSV file. **If the CSV file don't have _id field, then while importing MongoDB automatically insert _id field to all the documents**. To Import a collection from a CSV file, the syntax is as under

Syntax

mongoimport --db DB_NAME --collection COLLECTION_name --type=csv --headerline --file=Name-of-file-to-import

where,

DB_NAME – Name of the Database of the Collection to be exported

COLLECTION_name - Name of Collection of DB_NAME to be exported

Type –csv

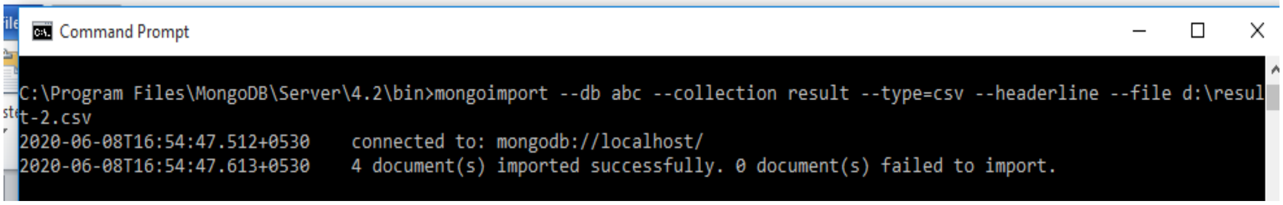
headerline - to take 1st record of CSV file as field names.

Name-of-file-to-import – Name and path of the JSON file to be imported (restore).

Example:

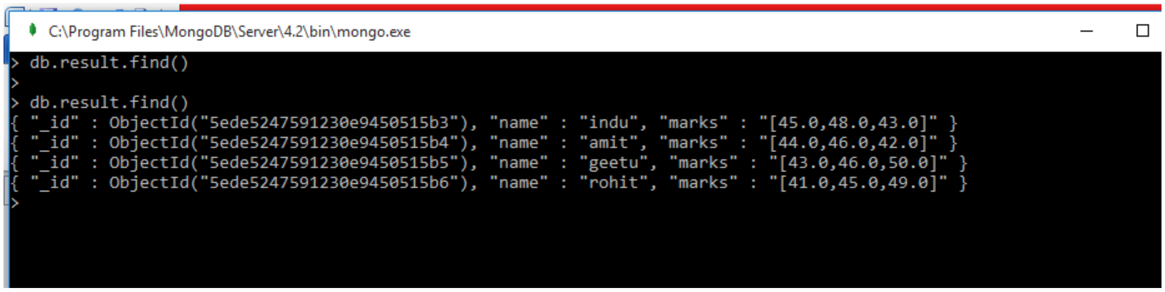
3. If our backup file name is result-2.csv which is in D:\ and the db name is abc and collection name is result.

mongoimport --db abc --collection result --type=csv --headerline --file d:\result-2.csv



```
C:\Program Files\MongoDB\Server\4.2\bin>mongoimport --db abc --collection result --type=csv --headerline --file d:\resul
t-2.csv
2020-06-08T16:54:47.512+0530    connected to: mongodb://localhost/
2020-06-08T16:54:47.613+0530    4 document(s) imported successfully. 0 document(s) failed to import.
```

The result shows the records imported to the collection. As the out CSV file don’t have any _id field but while importing MongoDB automatically insert _id field to all the documents. We may see that

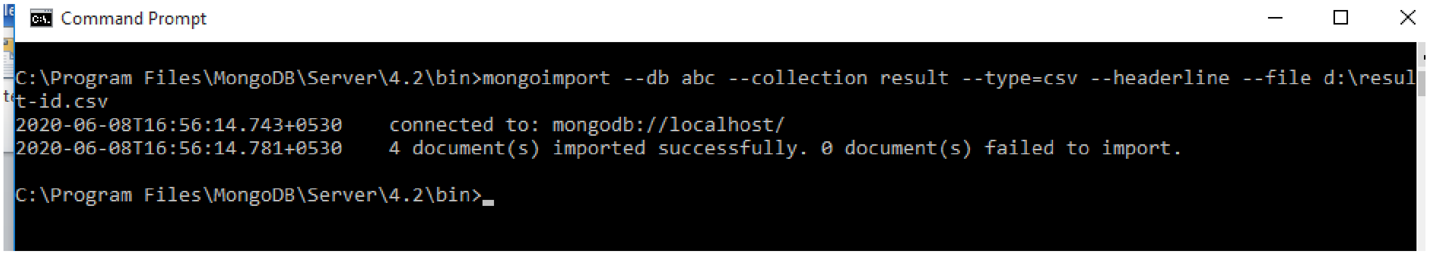


C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe

```
> db.result.find()
>
> db.result.find()
{ "_id" : ObjectId("5ede5247591230e9450515b3"), "name" : "indu", "marks" : "[45.0,48.0,43.0]" }
{ "_id" : ObjectId("5ede5247591230e9450515b4"), "name" : "amit", "marks" : "[44.0,46.0,42.0]" }
{ "_id" : ObjectId("5ede5247591230e9450515b5"), "name" : "geetu", "marks" : "[43.0,46.0,50.0]" }
{ "_id" : ObjectId("5ede5247591230e9450515b6"), "name" : "rohit", "marks" : "[41.0,45.0,49.0]" }
>
```

4. Now we are again importing a backup file result-id.csv having _id filed from D:\ and the db name is abc and collection name is result.

mongoimport --db abc --collection result --type=csv --headerline --file d:\result-id.csv

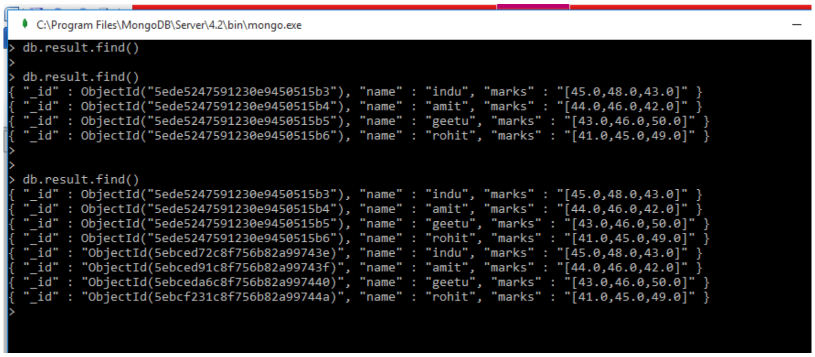


Command Prompt

```
C:\Program Files\MongoDB\Server\4.2\bin>mongoimport --db abc --collection result --type=csv --headerline --file d:\resul
t-id.csv
2020-06-08T16:56:14.743+0530    connected to: mongodb://localhost/
2020-06-08T16:56:14.781+0530    4 document(s) imported successfully. 0 document(s) failed to import.

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

Now if we display the collection result, it will have the records as



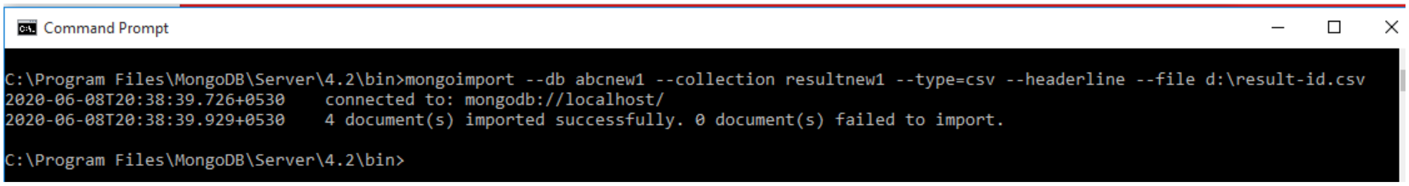
C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe

```
> db.result.find()
>
> db.result.find()
{ "_id" : ObjectId("5ede5247591230e9450515b3"), "name" : "indu", "marks" : "[45.0,48.0,43.0]" }
{ "_id" : ObjectId("5ede5247591230e9450515b4"), "name" : "amit", "marks" : "[44.0,46.0,42.0]" }
{ "_id" : ObjectId("5ede5247591230e9450515b5"), "name" : "geetu", "marks" : "[43.0,46.0,50.0]" }
{ "_id" : ObjectId("5ede5247591230e9450515b6"), "name" : "rohit", "marks" : "[41.0,45.0,49.0]" }
>
>
> db.result.find()
{ "_id" : ObjectId("5ede5247591230e9450515b3"), "name" : "indu", "marks" : "[45.0,48.0,43.0]" }
{ "_id" : ObjectId("5ede5247591230e9450515b4"), "name" : "amit", "marks" : "[44.0,46.0,42.0]" }
{ "_id" : ObjectId("5ede5247591230e9450515b5"), "name" : "geetu", "marks" : "[43.0,46.0,50.0]" }
{ "_id" : ObjectId("5ede5247591230e9450515b6"), "name" : "rohit", "marks" : "[41.0,45.0,49.0]" }
{ "_id" : "ObjectId(5ebced72c8f756b82a99743e)", "name" : "indu", "marks" : "[45.0,48.0,43.0]" }
{ "_id" : "ObjectId(5ebced91c8f756b82a99743f)", "name" : "amit", "marks" : "[44.0,46.0,42.0]" }
{ "_id" : "ObjectId(5ebceda6c8f756b82a997440)", "name" : "geetu", "marks" : "[43.0,46.0,50.0]" }
{ "_id" : "ObjectId(5ebcf231c8f756b82a99744a)", "name" : "rohit", "marks" : "[41.0,45.0,49.0]" }
>
```

Shows total 8 records, 4 imported in example-3 (system assigned _id) and 4 imported in example-4.

5. Now, we have taken database name as abcnew1 and collection name as resultnew1. In this case the collection will be imported as a new collection name and in a different database.

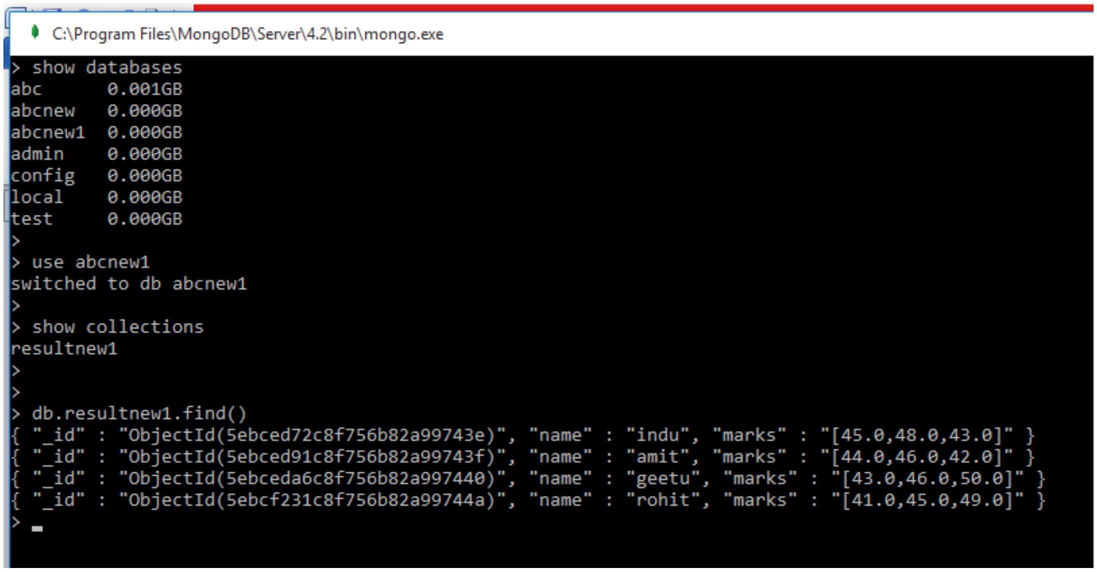
mongoimport --db abcnew1 --collection resultnew1 --type=csv --headerline --file d:\result-id.csv



```
C:\Program Files\MongoDB\Server\4.2\bin>mongoimport --db abcnew1 --collection resultnew1 --type=csv --headerline --file d:\result-id.csv
2020-06-08T20:38:39.726+0530    connected to: mongodb://localhost/
2020-06-08T20:38:39.929+0530    4 document(s) imported successfully. 0 document(s) failed to import.

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

And if we see the new database and collection, it will be like



```
> show databases
abc       0.001GB
abcnew    0.000GB
abcnew1   0.000GB
admin     0.000GB
config    0.000GB
local     0.000GB
test      0.000GB
>
> use abcnew1
switched to db abcnew1
>
> show collections
resultnew1
>
>
> db.resultnew1.find()
{ "_id" : "ObjectId(5ebced72c8f756b82a99743e)", "name" : "indu", "marks" : "[45.0,48.0,43.0]" }
{ "_id" : "ObjectId(5ebced91c8f756b82a99743f)", "name" : "amit", "marks" : "[44.0,46.0,42.0]" }
{ "_id" : "ObjectId(5ebceda6c8f756b82a997440)", "name" : "geetu", "marks" : "[43.0,46.0,50.0]" }
{ "_id" : "ObjectId(5ebcf231c8f756b82a99744a)", "name" : "rohit", "marks" : "[41.0,45.0,49.0]" }
> _
```

<u>Assignment</u>

1. How a JSON data file is imported as a collection?
2. How a CSV data file is imported as a collection?
3. Can we assign new name to collection as well as database while importing into the collection? If yes, how?