

Import a JSON file from the command line. Apply the following actions with the data present in the JSON file where, projection, aggregation, remove, count, limit, skip and sort

AIM:

To Import a JSON file from the command line. Apply the following actions with the data present in the JSON file where, projection, aggregation, remove, count, limit, skip and sort

PROCEDURE:

1. Open the terminal and run mongod. This will run the MongoDB server locally in your system.
2. In another terminal window run mongosh. This will activate the MongoDB Shell.
3. Create a database using the command use <database_name>.
4. Then in another window run this command
mongoimport --db mydatabase --collection employees --file /path/to/employees.json --jsonArray
5. The above command will import the JSON file into the database you created.
6. Then use the MongoDB commands to perform the given operations.

OUTPUT:

```
nohamedhussainshahhamed@Mohameds-Laptop ~ % mongo
MongoDB shell version v5.0.28
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("5d62f8b8-c441-4927-ab73-5d8d33346c98") }
MongoDB server version: 6.0.0
WARNING: shell and server versions do not match
=====
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====
The server generated these startup warnings when booting:
2024-08-29T17:02:43.505+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2024-08-29T17:02:43.505+05:30: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it sh
ould serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning
2024-08-29T17:02:43.505+05:30: Soft rlimits for open file descriptors too low
2024-08-29T17:02:43.505+05:30:      currentValue: 256
2024-08-29T17:02:43.505+05:30:      recommendedMinimum: 64000
>
```

```
[> show databases
admin    0.000GB
config  0.000GB
iisj     0.000GB
local    0.000GB
[> use iisj;
switched to db iisj
>
```

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

```
[> show collections;
employees
```

```
Users > mohamedhussainshahulhameed > Desktop > DA > ex5 > {} DAmo.json > ...
```

```
1 [
2   {"name": "sajjad", "age": 21, "city": "Dubai"},
3   {"name": "mukku", "age": 21, "city": "Chennai"},
4   {"name": "jesper", "age": 21, "city": "Arani"}
5 ]
```

```
mohamedhussainshahulhameed@mohameds-Laptop ~ % mongoimport --db iisj --collection employees --file /Users/mohamedhussainshahulhameed/Desktop/DA/ex5/DAmo.json --jsonArray
2024-08-29T17:20:15.606+0530   connected to: mongodb://localhost/
2024-08-29T17:20:15.616+0530   3 document(s) imported successfully. 0 document(s) failed to import.
```

```
> db.employees.find({$where:"this.age>20"});
{ "_id" : ObjectId("66d06077a73285d48c721fd9"), "name" : "jesper", "age" : 21, "city" : "Arani" }
{ "_id" : ObjectId("66d06077a73285d48c721fda"), "name" : "sajjad", "age" : 21, "city" : "Dubai" }
{ "_id" : ObjectId("66d06077a73285d48c721fdb"), "name" : "mukku", "age" : 21, "city" : "Chennai" }
```

Find()

```
> db.employees.find({}, {name:1, city:1, _id:0})
{ "name" : "jesper", "city" : "Arani" }
{ "name" : "sajjad", "city" : "Dubai" }
{ "name" : "mukku", "city" : "Chennai" }
```

Aggregate()

```
> db.employees.aggregate([
...   { $match: { city: "Dubai" } },
...   { $group: { _id: "$city", totalAge: { $sum: "$age" } } }
... ])
{ "_id" : "Dubai", "totalAge" : 21 }
```

Delete()

```
> db.yourcollection.deleteOne({ name: "jesper" })
{ "acknowledged" : true, "deletedCount" : 1 }
```

Count()

```
[> db.employees.countDocuments({ age: { $gt: 20 } })
3
```

limit()

```
> db.employees.find().limit(2)
{ "_id" : ObjectId("66d06077a73285d48c721fd9"), "name" : "jesper", "age" : 21, "city" : "Arani" }
{ "_id" : ObjectId("66d06077a73285d48c721fda"), "name" : "sajjad", "age" : 21, "city" : "Dubai" }
```

skip()

```
[> db.employees.find().skip(2)
{ "_id" : ObjectId("66d06077a73285d48c721fdb"), "name" : "mukku", "age" : 21, "city" : "Chennai" }
```

sort()

```
> db.employees.find().sort({ age: 1 })
{ "_id" : ObjectId("66d06077a73285d48c721fd9"), "name" : "jesper", "age" : 21, "city" : "Arani" }
{ "_id" : ObjectId("66d06077a73285d48c721fda"), "name" : "sajjad", "age" : 21, "city" : "Dubai" }
{ "_id" : ObjectId("66d06077a73285d48c721fdb"), "name" : "mukku", "age" : 21, "city" : "Chennai" }
```

RESULT:

Thus, to Import a JSON file from the command line. Apply the following actions with the data present in the JSON file where, projection, aggregation, remove, count, limit, skip and sort has been successfully executed.