Command Line Help

To see the list of options and help for starting the mongo shell, use the --help option from the command line:

mongo --help

Shell Help

To see the list of help, in the mongo shell, type help:

help

Database Help

In the mongo shell:

To see the list of databases on the server, use the show dbs command:

show dbs

New in version 2.4: show databases is now an alias for show dbs

To see the list of help for methods you can use on the db object, call the db.help() method:

db.help()

To see the implementation of a method in the shell, type the db.<method name> without the parenthesis (()), as in the following example which will return the implementation of the method db.updateUser():

db.updateUser

Collection Help

In the mongo shell:

To see all the database

show dbs

To see the list of collections in the current database, use the show collections command:

show collections

To see the help for methods available on the collection objects (e.g. db.<collection>), use the db.<collection>.help() method:

db.collection.help()

<collection> can be the name of a collection that exists, although you may specify a collection that doesn’t exist.

To see the collection method implementation, type the db.<collection>.<method> name without the parenthesis (()), as in the following example which will return the implementation of the save() method:

db.listcollections()

db.collection.save

Cursor Help

When you perform read operations with the find() method in the mongo shell, you can use various cursor methods to modify the find() behavior and various JavaScript methods to handle the cursor returned from the find() method.

To list the available modifier and cursor handling methods, use the db.collection.find().help() command:

db.collection.find().help()

<collection> can be the name of a collection that exists, although you may specify a collection that doesn’t exist.

To see the implementation of the cursor method, type the db.<collection>.find().<method> name without the parenthesis (()), as in the following example which will return the implementation of the toArray() method:

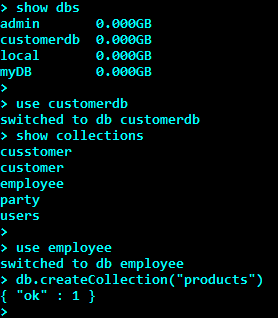
db.collection.find().toArray

The [mongo](https://docs.mongodb.com/manual/reference/program/mongo/#bin.mongo) shell is an interactive JavaScript interface to MongoDB. You can use the [mongo](https://docs.mongodb.com/manual/reference/program/mongo/#bin.mongo) shell to query and update data as well as perform administrative operations.

mongo also provides a fully functional JavaScript environment for use with a MongoDB.

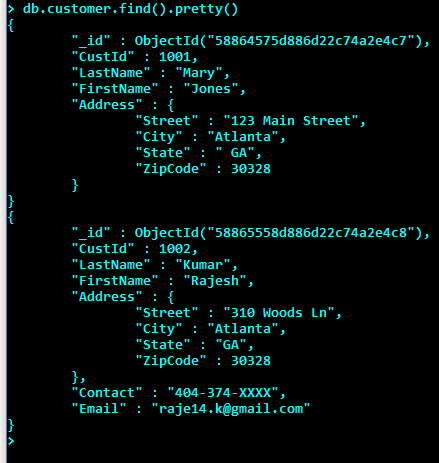
Working with the mongo Shell ; The operation should return test, which is the default database

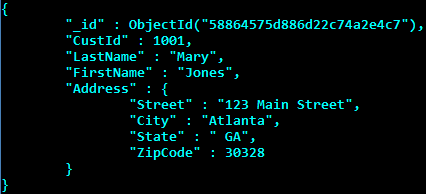
To switch databases, issue the use <db>

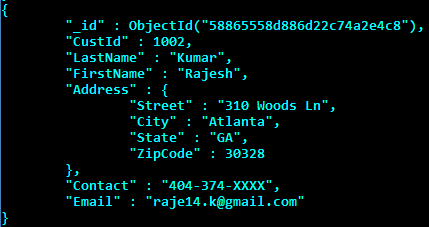


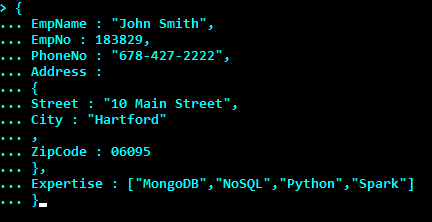
> db.customer.insert( { CustId : 1002, LastName : "Kumar", FirstName : "Rajesh", Address : { Street : "310 Woods Ln", City : "Atlanta", State : "GA", ZipCode : 30328, }, Contact: "404-374-XXXX", Email:"raje14.k@gmail.com"})

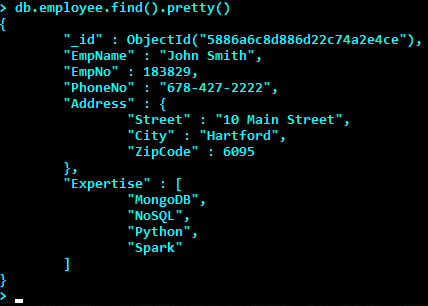












> db.employee.find().pretty()

{

"\_id" : ObjectId("5886a6c8d886d22c74a2e4ce"),

"EmpName" : "John Smith",

"EmpNo" : 183829,

"PhoneNo" : "678-427-2222",

"Address" : {

"Street" : "10 Main Street",

"City" : "Hartford",

"ZipCode" : 6095

},

"Expertise" : [

"MongoDB",

"NoSQL",

"Python",

"Spark"

]

}

{

"\_id" : ObjectId("5886a8d6d886d22c74a2e4cf"),

"EmpName" : "Rajesh Kumar",

"EmpNo" : 374528,

"PhoneNo" : "678-427-2222",

"Address" : {

"Street" : "123 Marsh Trail",

"City" : "Atlanta",

"ZipCode" : "30328"

},

"Expertise" : [

"MongoDB",

"NoSQL",

"Python",

"Spark"

]

}

{

"\_id" : ObjectId("5886aa27d886d22c74a2e4d0"),

"EmpName" : "Rajesh Kumar",

"EmpNo" : 374528,

"PhoneNo" : "678-427-2222",

"Address" : {

"Street" : "123 Marsh Trail",

"City" : "Atlanta",

"ZipCode" : "30328"

},

"Expertise" : [

"MongoDB",

"NoSQL",

"Python",

"Spark"

],

"CurrentRole" : "Data Modeler"

}

{

"\_id" : ObjectId("5886aaa8d886d22c74a2e4d1"),

"EmpName" : "Rajesh Kumar",

"EmpNo" : 374528,

"PhoneNo" : "678-427-2222",

"Address" : {

"Street" : "123 Marsh Trail",

"City" : "Atlanta",

"ZipCode" : "30328"

},

"Expertise" : [

"MongoDB",

"NoSQL",

"Python",

"Spark"

],

"CurrentRole" : "Data Modeler",

"ClientLocation" : "Hartford"

}

>

> db.employee.insert( { EmpName : "Rajesh Kumar", EmpNo : 374528, PhoneNo : "678-427-2222", Address : { Street : "123 Marsh Trail", City : "Atlanta" , ZipCode : "30328" }, Expertise : ["MongoDB","NoSQL","Python","Spark"]})

WriteResult({ "nInserted" : 1 })

> db.employee.insert( { EmpName : "Rajesh Kumar", EmpNo : 374528, PhoneNo : "678-427-2222", Address : { Street : "123 Marsh Trail", City : "Atlanta" , ZipCode : "30328" }, Expertise : ["MongoDB","NoSQL","Python","Spark"],CurrentRole:"Data Modeler",ClientLocation :"Hartford"})

WriteResult({ "nInserted" : 1 })

