MongoDB

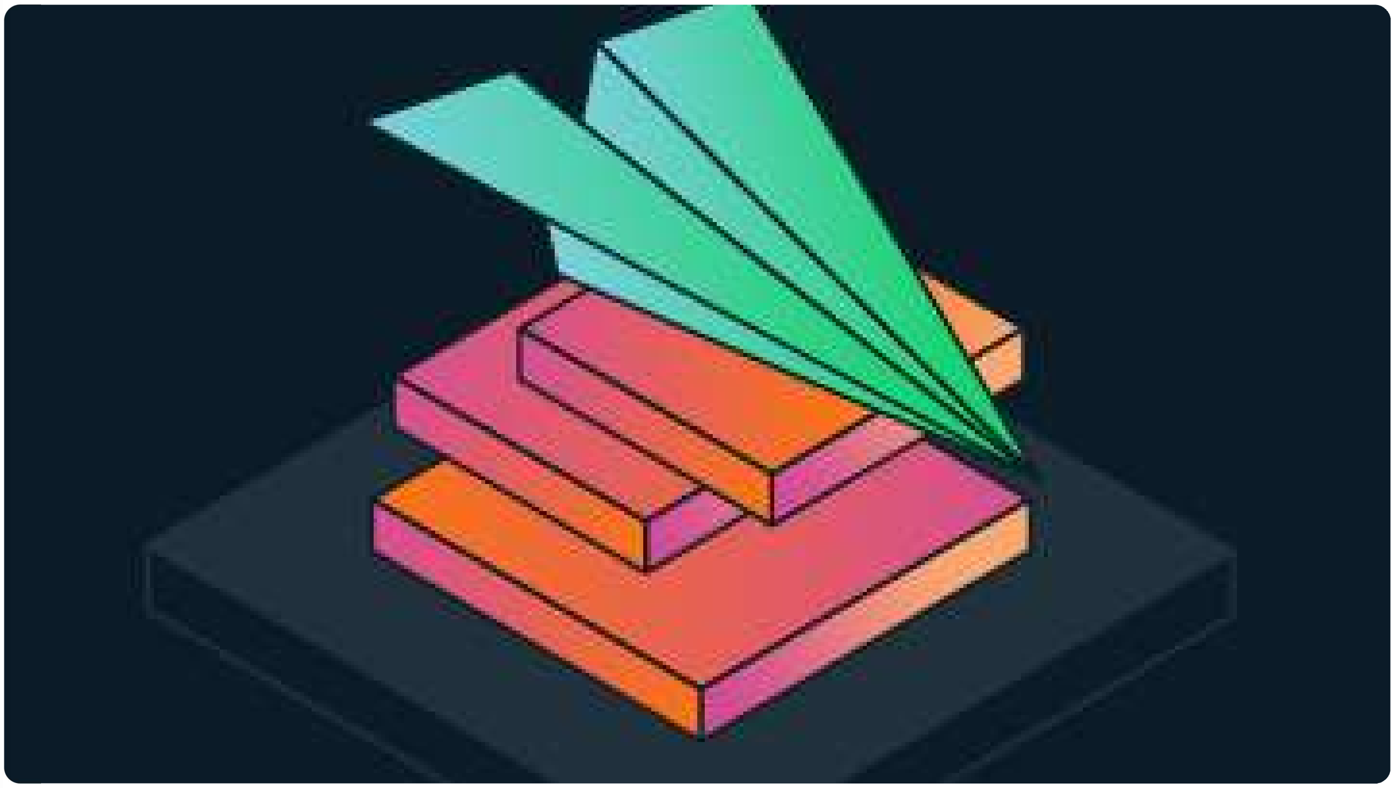
MongoDB Developer

MongoDB Cheat Sheet

Maxime Beugnet

Published Jan 31, 2022 • Updated May 13, 2022

MongoDB



Rate this quickstart ☆ ☆ ☆ ☆ ☆

First steps in the MongoDB World? This cheat sheet is filled with some handy tips, commands, and quick references to get you connected and CRUD'ing in no time!

- Get a free MongoDB cluster in MongoDB Atlas.
- Follow a course in MongoDB University.

Table of Contents

MongoDB Developer

- Helpers
- CRUD
- Databases and Collections
- Indexes
- Handy commands
- Change Streams
- Replica Set
- Sharded Cluster
- Wrap-up

Connect MongoDB Shell

```
mongo # connects to mongodb://127.0.0.1:27017 by default
mongo --host <host> --port <port> -u <user> -p <pwd> # omit the password if you want a prompt
mongo "mongodb://192.168.1.1:27017"


mongo "mongodb+srv://cluster-name.abcde.mongodb.net/<dbname>" --username <username> # MongoDB Atlas
```

- More documentation about the MongoDB Shell.
- To connect with the new mongosh, just replace `mongo` by `mongosh`.

🔝 Table of Contents 🔝

Helpers

Show Databases

MongoDB Developer

```
show dbs
db // prints the current database
```

Switch Database

```
use <database_name>
```

Show Collections

```
show collections
```

MongoDB Developer

```
load("myScript.js")
```

🔝 Table of Contents 🔝

CRUD

Create

```
db.coll.insertOne({name: "Max"})
db.coll.insert([{name: "Max"}, {name:"Alex"}]) // ordered bulk insert
db.coll.insert([{name: "Max"}, {name:"Alex"}], {ordered: false}) // unordered bulk insert
db.coll.insert({date: ISODate()})
db.coll.insert({name: "Max"}, {"writeConcern": {"w":
```

Read

```
db.coll.findOne() // returns a single document
db.coll.find()    // returns a cursor - show 20 results - "it"
to display more
```

MongoDB Developer

```
db.coll.find({date: ISODate("2020-09-25T13:57:17.180Z")})
db.coll.find({name: "Max", age: 32}).explain("executionStats")
// or "queryPlanner" or "allPlansExecution"
db.coll.distinct("name")

// Count
db.coll.count({age: 32})          // estimation based on
collection metadata
db.coll.estimatedDocumentCount()  // estimation based on
collection metadata
db.coll.countDocuments({age: 32}) // alias for an aggregation
pipeline - accurate count

// Comparison
db.coll.find({"year": {$gt: 1970}})
db.coll.find({"year": {$gte: 1970}})
db.coll.find({"year": {$lt: 1970}})
```

- db.collection.find()
- Query and Projection Operators
- BSON types
- Read Concern

Update

```
db.coll.update({"_id": 1}, {"year": 2016}) // WARNING! Replaces
db.coll.update({"_id": 1}, {"year": 2016}) // WARNING! Replaces
the entire document
```

MongoDB Developer

```
db.coll.update({"_id": 1}, {$rename: {"year": "date"} })
db.coll.update({"_id": 1}, {$inc: {"year": 5}})
db.coll.update({"_id": 1}, {$mul: {price:
NumberDecimal("1.25"), qty: 2}})
db.coll.update({"_id": 1}, {$min: {"imdb": 5}})
db.coll.update({"_id": 1}, {$max: {"imdb": 8}})
db.coll.update({"_id": 1}, {$currentDate: {"lastModified":
true}})
db.coll.update({"_id": 1}, {$currentDate: {"lastModified":
{$type: "timestamp"}}})

// Array
db.coll.update({"_id": 1}, {$push :{"array": 1}})
db.coll.update({"_id": 1}, {$pull :{"array": 1}})
db.coll.update({"_id": 1}, {$addToSet :{"array": 2}})
db.coll.update({"_id": 1}, {$pop: {"array": 1}})  // last
element
db.coll.update({"_id": 1}, {$pop: {"array": -1}}) // first
```

Delete

```
db.coll.remove({name: "Max"})
db.coll.remove({name: "Max"}, {justOne: true})
db.coll.remove({}) // WARNING! Deletes all the docs but not the
collection itself and its index definitions
db.coll.remove({name: "Max"}, {"writeConcern": {"w":
"majority", "wtimeout": 5000}})
```

```
majority", wtimeout: 5000}})
db.coll.findOneAndDelete({"name": "Max"})
```

MongoDB Developer

Databases and Collections

Drop

```
db.coll.drop()    // removes the collection and its index definitions
db.dropDatabase() // double check that you are *NOT* on the PROD cluster... :-)
```

Create Collection

```
// Create collection with a $jsonschema
db.createCollection("contacts", {
   validator: {$jsonSchema: {
      bsonType: "object",
      required: ["phone"],
      properties: {
         phone: {
            bsonType: "string",
            description: "must be a string and is required"
         },
         email: {
            bsonType: "string",
            pattern: "@mongodb\.com$",
```

```
         },
         status: {
            enum: [ "Unknown", "Incomplete" ],
            description: "can only be one of the enum values"
         }
      }
```

Other Collection Functions

```
db.coll.stats()
db.coll.storageSize()
db.coll.totalIndexSize()
db.coll.totalSize()
db.coll.validate({full: true})
db.coll.renameCollection("new_coll", true) // 2nd parameter to drop the target collection if exists
```

🔝 Table of Contents 🔝

Indexes

List Indexes

```
db.coll.getIndexes()
db.coll.getIndexKeys()
```

MongoDB Developer

Create Indexes

```
// Index Types
db.coll.createIndex({"name": 1})                // single field index
db.coll.createIndex({"name": 1, "date": 1})     // compound index
db.coll.createIndex({foo: "text", bar: "text"}) // text index
db.coll.createIndex({"$**": "text"})            // wildcard text index
db.coll.createIndex({"userMetadata.$**": 1})    // wildcard index
db.coll.createIndex({"loc": "2d"})              // 2d index
db.coll.createIndex({"loc": "2dsphere"})        // 2dsphere index
db.coll.createIndex({"_id": "hashed"})          // hashed index


// Index Options
db.coll.createIndex({"lastModifiedDate": 1}, {expireAfterSeconds: 3600})      // TTL index
db.coll.createIndex({"name": 1}, {unique: true})
db.coll.createIndex({"name": 1}, {partialFilterExpression: {age: {$gt: 18}}}) // partial index
```

Drop Indexes

```
db.coll.dropIndex("name_1")
```

MongoDB Developer

Hide/Unhide Indexes

```
db.coll.hideIndex("name_1")
db.coll.unhideIndex("name_1")
```

- Indexes documentation

🔝 Table of Contents 🔝

Handy commands

```
use admin
db.createUser({"user": "root", "pwd": passwordPrompt(),
"roles": ["root"]})
db.dropUser("root")
db.auth( "user", passwordPrompt() )

```

MongoDB Developer

```
db.currentOp()
db.killOp(123) // opid

db.fsyncLock()
db.fsyncUnlock()

db.getCollectionNames()
db.getCollectionInfos()
db.printCollectionStats()
db.stats()

db.getReplicationInfo()
db.printReplicationInfo()
db.isMaster()
```

🔝 Table of Contents 🔝

Change Streams

```
watchCursor = db.coll.watch( [ { $match : {"operationType" :
"insert" } } ] )

while (!watchCursor.isExhausted()){
   if (watchCursor.hasNext()){
      print(tojson(watchCursor.next()));
   }
}
```

Table of Contents

```
rs.status()
rs.initiate({"_id": "replicaTest",
  members: [
    { _id: 0, host: "127.0.0.1:27017" },
    { _id: 1, host: "127.0.0.1:27018" },
    { _id: 2, host: "127.0.0.1:27019", arbiterOnly:true }]
})
rs.add("mongodbd1.example.net:27017")
rs.addArb("mongodbd2.example.net:27017")
rs.remove("mongodbd1.example.net:27017")
rs.conf()
rs.isMaster()
rs.printReplicationInfo()
rs.printSlaveReplicationInfo()
rs.reconfig(<valid_conf>)
rs.slaveOk()
rs.stepDown(20, 5) // (stepDownSecs, secondaryCatchUpPeriodSecs)
```

Table of Contents

Sharded Cluster

```
sh.status()
sh.addShard("rs1/mongodbd1.example.net:27017")
sh.shardCollection("mydb.coll", {zipcode: 1})
```

```
sh.splitAt("mydb.coll", {x: 70})
sh.splitFind("mydb.coll", {x: 70})
sh.disableAutoSplit()
sh.enableAutoSplit()

sh.startBalancer()
sh.stopBalancer()
sh.disableBalancing("mydb.coll")
sh.enableBalancing("mydb.coll")
sh.getBalancerState()
sh.setBalancerState(true/false)
sh.isBalancerRunning()

sh.addTagRange("mydb.coll", {state: "NY", zip: MinKey }, { state: "NY", zip: MaxKey }, "NY")
sh.removeTagRange("mydb.coll", {state: "NY", zip: MinKey }, { state: "NY", zip: MaxKey }, "NY")
```

🔝 Table of Contents 🔝

Wrap-up

I hope you liked my little but - hopefully - helpful cheat sheet. Of course, this list isn't exhaustive at all. There are a lot more commands but I'm sure you will find them in the MongoDB documentation.

If you feel like I forgot a critical command in this list, please send me a tweet and I will make sure to fix it.

Check out our free courses on MongoDB University if you are not too sure what some of the above commands are doing.

MongoDB Developer

If you have questions, please head to our developer community website where the MongoDB engineers and the MongoDB community will help you build your next big idea with MongoDB.

🔝 Table of Contents 🔝

Rate this quickstart

Related

ARTICLE

Window Functions & Time Series Collections

May 13, 2022

ARTICLE

Building with Patterns: The Polymorphic Pattern

May 31, 2022

QUICKSTART

Aggregation Framework with Node.js Tutorial

MongoDB Developer

ARTICLE

Securing MongoDB with TLS

May 10, 2022

Request a Quickstart

MongoDB.

About

Careers

Investor Relations

Legal Notices

Privacy Notices

Security Information

Trust Center

Support

Contact Us

Customer Portal

MongoDB Developer

Social

Github

Stack Overflow

LinkedIn

Youtube

Twitter

Twitch

Facebook