

MongoDB Cheatsheet

Connection

mongo

Start the MongoDB shell client

mongo --host <hostname> --port <port>

Connect to a MongoDB instance

mongo "mongodb://<hostname>:<port>/<database>"

Connect with connection string

mongo --username <username> --password <password> --authenticationDatabase <authdb>

Connect with authentication

use <database>

Switch to a specific database

Database Operations

show dbs

List all databases

show collections

List all collections in current database

db.stats()

Display statistics for current database

db.dropDatabase()

Delete the current database

Collection Operations

db.createCollection("<name>")

Create a new collection

db.<collection>.stats()

Show statistics for a collection

db.<collection>.drop()

Delete a collection

CRUD Operations

db.<collection>.find()

Query documents in a collection

db.<collection>.find().pretty()

Format query results for readability

db.<collection>.findOne()

Retrieve first matching document

db.<collection>.insertOne({})

Insert a single document

db.<collection>.insertMany([], {})

Insert multiple documents

db.<collection>.updateOne({query}, {\$set: {}})

Update a single document

db.<collection>.updateMany({query}, {\$set: {}})

Update multiple documents

db.<collection>.deleteOne({query})

Delete a single document

db.<collection>.deleteMany({query})

Delete multiple documents

Query Operations

db.<collection>.find({field: value})

Filter documents by field value

db.<collection>.find({field: {\$gt: value}})

Greater than comparison

db.<collection>.find({field: {\$lt: value}})

Less than comparison

db.<collection>.find().sort({field: 1})

Sort results (1 for ascending, -1 for descending)

db.<collection>.find().limit(n)

Limit results to n documents

db.<collection>.find().skip(n)

Skip n documents in results

db.<collection>.count()

Count documents in collection

db.<collection>.distinct("field")

Get distinct values for a field

Indexing

db.<collection>.createIndex({field: 1})

Create an index (1 for ascending, -1 for descending)

db.<collection>.createIndex({field: "text"})

Create a text index

db.<collection>.getIndexes()

List all indexes on a collection

db.<collection>.dropIndex("index_name")

Remove an index

Aggregation

db.<collection>.aggregate([{\$match: {}}, {\$group: {}}])

Run an aggregation pipeline

db.<collection>.mapReduce(map, reduce, {out: "collection"})

Perform map-reduce operation

Monitoring

db.serverStatus()

Get server status information

db.stats()

Current database statistics

db.currentOp()

View current operations

db.killOp(opid)

Kill a specific operation by ID

rs.status()

Check replica set status

Administrative

db.shutdownServer()

Shutdown the MongoDB server

db.runCommand({command})

Run a database command

db.createUser()

Create a new user

db.getUsers()

List users in current database

db.auth("username", "password")

Authenticate as user