**MongoDB**

<https://www.mongodb.com/try/download/community>

Download the latest version and make sure it is the MSI package.

A screenshot of a phone

Description automatically generated

Make a note of where the installation files are going to be installed.

Data Directory:

A screenshot of a computer

Description automatically generated

Uncheck Install MongoDB Compass

A screenshot of a computer

Description automatically generated

Setup data folder in the C:\

C:\data\db – make sure the filenames are lowercase.

A screenshot of a computer

Description automatically generated

Open git bash terminal

Type: cd ~ to change into the home folder.

Create a new file .bash\_profile: touch .bash\_profile

ls -a 🡪 see if the file is created properly.

A screenshot of a computer

Description automatically generated

Edit the .bash\_profile file

vim .bash\_profile

A black and white screen with green and purple text

Description automatically generated

Hit ‘**i**’ to enable editing of file.

MongoDB installation documentation:

<https://medium.com/@LondonAppBrewery/how-to-download-install-mongodb-on-windows-4ee4b3493514>

A screenshot of a computer

Description automatically generated

Copy the 2 lines:

alias mongod="/c/Program\ files/MongoDB/Server/7.0/bin/mongod.exe"  
alias mongo="/c/Program\ Files/MongoDB/Server/7.0/bin/mongo.exe"

Paste it into the .bash\_profile

A computer screen with text and numbers

Description automatically generated

Save and Exit

Hit esc key

:wq!

A computer screen with text on it

Description automatically generated

**Test to see if the installation has succeeded.**

Close the bash terminal.

Re-open the bash terminal.

Type: mongo --version 🡪 you should see the version and Build Info.

A computer screen shot of a black screen

Description automatically generated

\*Note: if you get this error

A screenshot of a computer

Description automatically generated

The file in our programfiles is mongos.exe.

Change the filename to mongo without the ‘s’.

**CRUD**

<https://www.mongodb.com/docs/manual/crud/>

Open bash terminal:

Type: mongod

You should see: "Waiting for connections","attr":{"port":27017,"ssl":"off"}}

A computer screen with text on it

Description automatically generated

Open a new bash terminal and leave the other one the way it is. It is listening on port 27017.

Type mongo and you should get a prompt that says test>

A screenshot of a computer screen

Description automatically generated

**\*Note: if you get an error, that means we need to download the mongo shell.**

<https://www.mongodb.com/try/download/shell>

Download the file as a zip.

A screenshot of a phone

Description automatically generated

Extract the file.

Paste the extracted file in the C:\ folder.

Copy the path of the bin folder and add it to the environment variables.

C:\Program Files\mongoshell\bin

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

Now run the command mongosh.

You should get a new prompt that says test>

A screenshot of a computer screen

Description automatically generated

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CRUD Documentation**

<https://www.mongodb.com/docs/manual/crud/>

show dbs – displays all databases. By default mongoDB creates 3 databases: admin, config, and local.

**Create Data**

use <databaseName> – creates a new database

The database will not show up when you run the command show dbs because there is no data in it.

Add data to our database showDB:

db.products.insertOne({\_id: 1, name: “Pen”, price: 1.20}) – creates a collection named products in the shopDB database.

A screen shot of a computer screen

Description automatically generated

show collections – shows the collection(s) in the database.

A screenshot of a computer screen

Description automatically generated

**Read Data**

db.collection.find() – finds everything.

db.products.find()

A computer screen shot of a black and white screen

Description automatically generated

db.collection.find(query) – query is optional.

db.products.find({name: “Pencil”}) – finds name that is Pencil

A computer screen shot of a black screen

Description automatically generated

db.products.find({price: {$gt: 1}}) – find item where price is greater than 1.

A screen shot of a computer

Description automatically generated

db.collection.find(query, projection) – query and projection are optional.

db.products.find({\_id: 1}, {name: 1}) – finds the id: 1 and name.

A computer screen shot of a black screen

Description automatically generated

When we search for data, the ID will always get returned.

We can prevent the ID from being returned.

db.products.find({\_id: 1}, {name: 1, \_id: 0}) – finds the id: 1 and name.

A screenshot of a computer screen

Description automatically generated

**Update Data**

db.collection.updateOne()

db.products.updateOne({\_id: 1}, {$set:{stock: 32}}) – creates a new field ‘stock’ and sets it to 32.

A computer screen shot of a program

Description automatically generated

db.products.find() – now it shows the new field ‘stock’.

A computer screen shot of a computer program

Description automatically generated

**Delete Data**

db.collection.deleteOne()

db.products.deleteOne({\_id: 2})

A computer screen shot of a program

Description automatically generated

**Delete Entire Database**

Switch to the database you want to delete.

use fruitsDB

Delete the database.

db.dropDatabase()

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Relationships**

**One to Many**

Here we have “Rubber” with many ‘reviews’.

db.products.insertOne(

    {

        \_id: 3,

        name: "Rubber",

        price: 1.30,

        stock: 43,

        reviews: [

            {

                authorName: "Sally",

                rating: 5,

                review: "Best rubber ever!"

            },

            {

                authorName: "John",

                rating: 5,

                review: "Awesome rubber!"

            }

        ]

    }

)

A screen shot of a computer

Description automatically generated

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**MongoDB with Node.JS using Native MongoDB Driver**

Use the Mongoose package.

Documentation to use MongoDB with Node.JS

<https://mongodb.github.io/node-mongodb-native/3.5/quick-start/quick-start/>

**Mongo Installation**

Create a new project folder.

npm init -y

Install mongoose driver.

npm i mongodb

run mongod at bash terminal

app.js Mongo

// jshint esversion:6

const MongoClient = require('mongodb').MongoClient;

const assert = require('assert');

// Connection URL

const url = 'mongodb://localhost:27017';

// Database Name

const dbName = 'fruitsDB';

// Create a new MongoClient

const client = new MongoClient(url, { useNewUrlParser: true });

// Use connect method to connect to the Server

client.connect(function (err) {

    assert.equal(null, err);

    console.log("Connected successfully to server");

    const db = client.db(dbName);

    findDocuments(db, function () {

        client.close();

    });

})

const insertDocuments = function (db, callback) {

    // Get the documents collection

    const collection = db.collection('fruits');

    // Insert some documents

    collection.insertMany([

        {

            name: "Apple",

            score: 8,

            review: "Great fruit"

        },

        {

            name: "Orange",

            score: 6,

            review: "Kinda sour"

        },

        {

            name: "Banana",

            score: 9,

            review: "Great stuff"

        }

    ], function (err, result) {

        assert.equal(err, null);

        assert.equal(3, result.result.n);

        assert.equal(3, result.ops.length);

        console.log("Inserted 3 documents into the collection");

        callback(result);

    });

}

const findDocuments = function (db, callback) {

    // Get the documents collection

    const collection = db.collection('fruits');

    // Find some documents

    collection.find({}).toArray(function (err, fruits) {

        assert.equal(err, null);

        console.log("Found the following records");

        console.log(fruits)

        callback(fruits);

    });

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Mongoose**

Mongoose is an object document mapper (ODM).

Allows communication between mongoDB database and node JS app.

Install the mongoose package:

npm i mongoose