

Crud NodeJS

(Part two - Crud NodeJS with Mongo DB)

Versión : 0.1.1(20190303)

Author :

Ing. Duván Albeiro Mejia Cortes (software.bugs.io@gmail.com)

Objective:

In this short course (tutorial) we are going to build a Crud NodeJS web application with one NoSQL (MongoDB). To access the database we will use JavaScript programming language. Also we will build a server uses NodeJS.

Note: This is the Second part of the tutorial.

Software Requirements:

Bellow there is the list of the required Software to build a Crud Nodejs With Mongo DB.

- JavaScript
- NodeJS v11.10.1
- Visual Studio code ([Version 1.31](#))
- Express
- MongoDB(NoSQL)

Download the folder

Download the [crud-nodejs-mongodb](#) folder:

Clone this repository

```
git clone https://github.com/DuvanSGF/NodeJS.git
```

Or download the zip file and unpack it:

```
https://github.com/DuvanSGF/NodeJS/archive/master.zip/
```

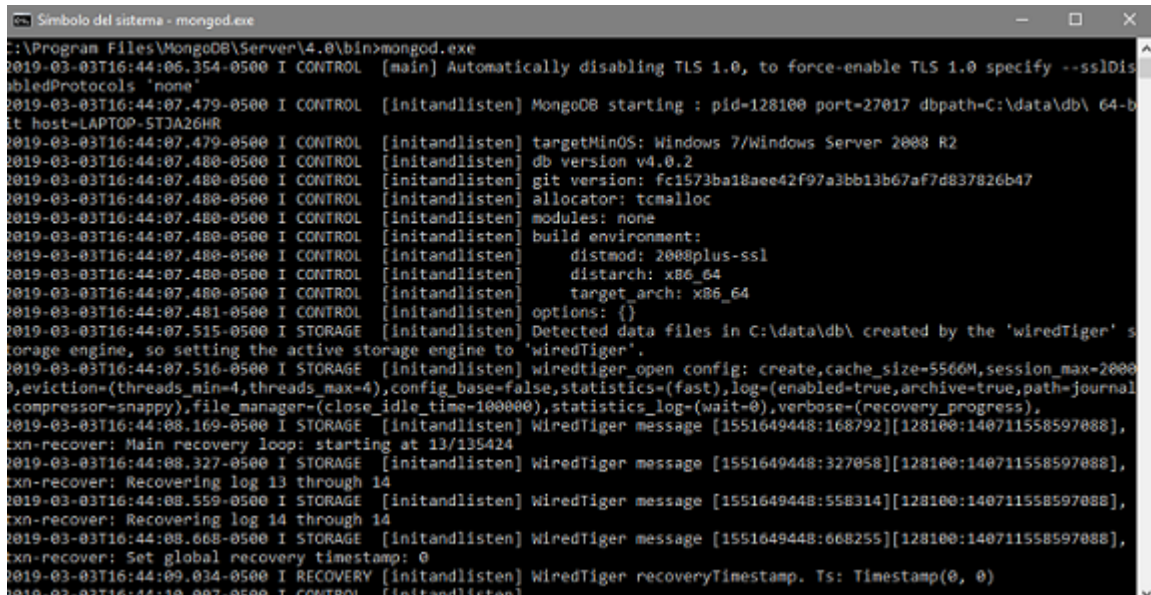
unzip master.zip

1. Database

1.1 NoSQL database

Run the MongoDB server

To create a database in MongoDB, use the `mongod.exe` command in a terminal to start (run) the server

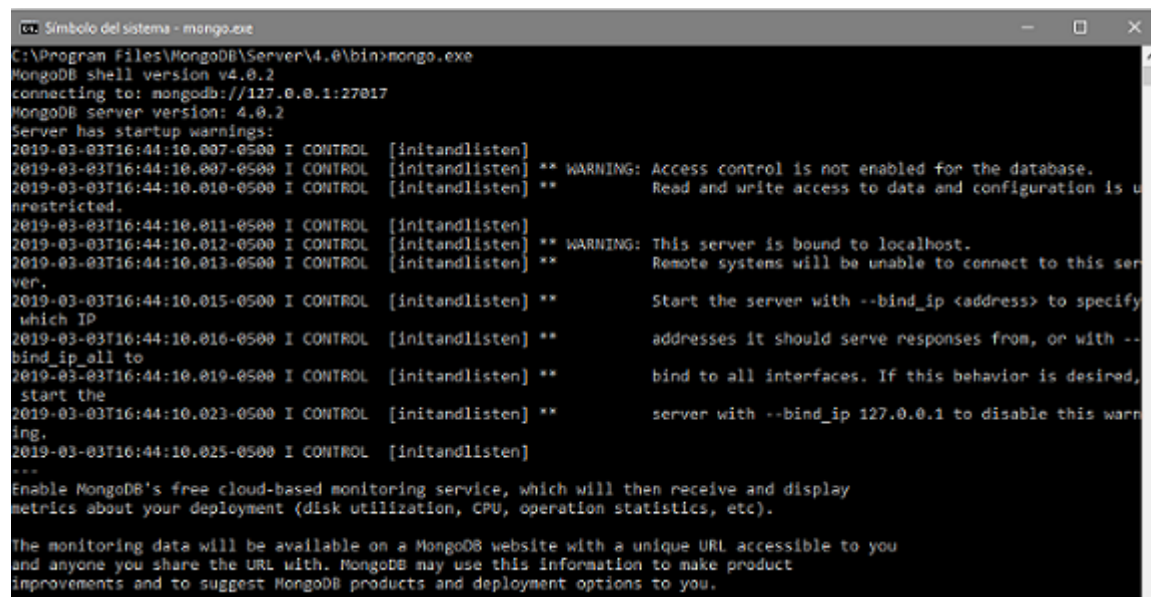


```
Símbolo del sistema - mongod.exe
C:\Program Files\MongoDB\Server\4.0\bin>mongod.exe
2019-03-03T16:44:06.354-0500 I CONTROL [main] Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDis
abledProtocols 'none'
2019-03-03T16:44:07.479-0500 I CONTROL [initandlisten] MongoDB starting : pid=128100 port=27017 dbpath=C:\data\db\ 64-b
it host=LAPTOP-5T3A26HR
2019-03-03T16:44:07.479-0500 I CONTROL [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2019-03-03T16:44:07.480-0500 I CONTROL [initandlisten] db version v4.0.2
2019-03-03T16:44:07.480-0500 I CONTROL [initandlisten] git version: fc1573ba18aee42f97a3bb13b67af7d837826b47
2019-03-03T16:44:07.480-0500 I CONTROL [initandlisten] allocator: tcmalloc
2019-03-03T16:44:07.480-0500 I CONTROL [initandlisten] modules: none
2019-03-03T16:44:07.480-0500 I CONTROL [initandlisten] build environment:
2019-03-03T16:44:07.480-0500 I CONTROL [initandlisten] distmod: 2008plus-ssl
2019-03-03T16:44:07.480-0500 I CONTROL [initandlisten] distarch: x86_64
2019-03-03T16:44:07.480-0500 I CONTROL [initandlisten] target_arch: x86_64
2019-03-03T16:44:07.481-0500 I CONTROL [initandlisten] options: {}
2019-03-03T16:44:07.515-0500 I STORAGE [initandlisten] Detected data files in C:\data\db\ created by the 'wiredTiger' s
torage engine, so setting the active storage engine to 'wiredTiger'.
2019-03-03T16:44:07.516-0500 I STORAGE [initandlisten] wiredtiger open config: create,cache_size=5566M,session_max=2000
0,eviction=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal
,compressor=snappy),file_manager=(close_idle_time=100000),statistics_log=(wait=0),verbose=(recovery_progress),
2019-03-03T16:44:08.169-0500 I STORAGE [initandlisten] WiredTiger message [1551649448:168792][128100:140711558597088],
txn-recover: Main recovery loop: starting at 13/135424
2019-03-03T16:44:08.327-0500 I STORAGE [initandlisten] WiredTiger message [1551649448:327058][128100:140711558597088],
txn-recover: Recovering log 13 through 14
2019-03-03T16:44:08.559-0500 I STORAGE [initandlisten] WiredTiger message [1551649448:558314][128100:140711558597088],
txn-recover: Recovering log 14 through 14
2019-03-03T16:44:08.668-0500 I STORAGE [initandlisten] WiredTiger message [1551649448:668255][128100:140711558597088],
txn-recover: Set global recovery timestamp: 0
2019-03-03T16:44:09.034-0500 I RECOVERY [initandlisten] WiredTiger recoveryTimestamp. Ts: Timestamp(0, 0)
2019-03-03T16:44:10.007-0500 I CONTROL [initandlisten]
```

This terminal windows must be kept open. if you close this console, you kill the MongoDB server process (stops the server).

Run the MongoDB client

To run the MongoDB client, open a new terminal windows and execute the `mongo.exe` command.



```
Símbolo del sistema - mongo.exe
C:\Program Files\MongoDB\Server\4.0\bin>mongo.exe
MongoDB shell version v4.0.2
connecting to: mongod://127.0.0.1:27017
MongoDB server version: 4.0.2
Server has startup warnings:
2019-03-03T16:44:10.007-0500 I CONTROL [initandlisten]
2019-03-03T16:44:10.007-0500 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2019-03-03T16:44:10.010-0500 I CONTROL [initandlisten] ** Read and write access to data and configuration is u
nrestricted.
2019-03-03T16:44:10.011-0500 I CONTROL [initandlisten]
2019-03-03T16:44:10.012-0500 I CONTROL [initandlisten] ** WARNING: This server is bound to localhost.
2019-03-03T16:44:10.013-0500 I CONTROL [initandlisten] ** Remote systems will be unable to connect to this ser
ver.
2019-03-03T16:44:10.015-0500 I CONTROL [initandlisten] ** Start the server with --bind_ip <address> to specify
which IP
2019-03-03T16:44:10.016-0500 I CONTROL [initandlisten] ** addresses it should serve responses from, or with --
bind_ip_all to
2019-03-03T16:44:10.019-0500 I CONTROL [initandlisten] ** bind to all interfaces. If this behavior is desired,
start the
2019-03-03T16:44:10.023-0500 I CONTROL [initandlisten] ** server with --bind_ip 127.0.0.1 to disable this warn
ing.
2019-03-03T16:44:10.025-0500 I CONTROL [initandlisten]
***
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

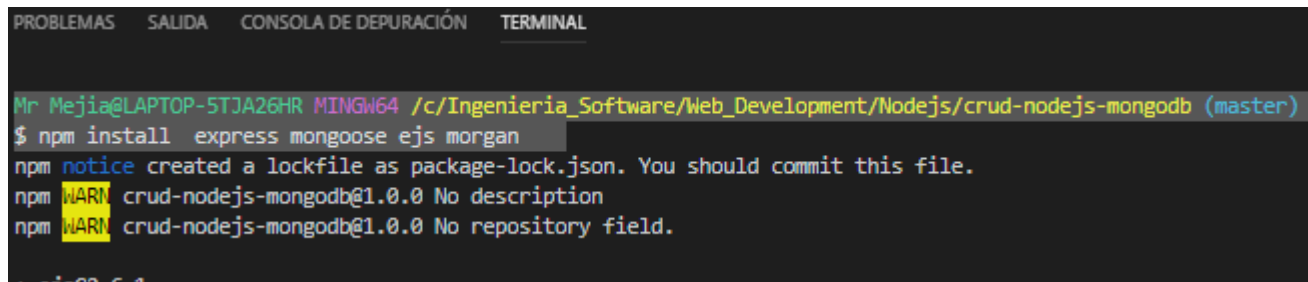
The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.
```

NodeJS web application will create the database and the collection.

2. Express Test Project

2.1 Required Software - Libraries

Inside crud-nodejs-mongo-db folder you must open the terminal.

A screenshot of a terminal window with a dark background. At the top, there are tabs labeled 'PROBLEMAS', 'SALIDA', 'CONSOLA DE DEPURACIÓN', and 'TERMINAL'. The terminal shows a command prompt for a user named 'Mr Mejia' on a machine named 'LAPTOP-5TJA26HR' running 'MINGW64'. The current directory is '/c/Ingenieria_Software/Web_Development/Nodejs/crud-nodejs-mongodb' and the branch is '(master)'. The command '\$ npm install express mongoose ejs morgan' has been executed. The output shows 'npm notice' about a lockfile, followed by two 'npm WARN' messages: 'crud-nodejs-mongodb@1.0.0 No description' and 'crud-nodejs-mongodb@1.0.0 No repository field.'.

```
Mr Mejia@LAPTOP-5TJA26HR MINGW64 /c/Ingenieria_Software/Web_Development/Nodejs/crud-nodejs-mongodb (master)
$ npm install express mongoose ejs morgan
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN crud-nodejs-mongodb@1.0.0 No description
npm WARN crud-nodejs-mongodb@1.0.0 No repository field.
```

2.1.2 Express

Express.js, or simply **Express**, is a [web application framework](#) for [Node.js](#), released as [free and open-source software](#) under the [MIT License](#). It is designed for building [web applications](#) and [APIs](#).^[3] It has been called the de facto standard server framework for Node.js.^[4]

```
$ npm install express
```

2.1.3 Mongoose

Mongoose is an object data modeling (ODM) library that provides a rigorous modeling environment for your data, enforcing structure as needed while still maintaining the flexibility that makes MongoDB powerful.

```
$ npm install mongoose
```

2.1.4 Ejs

EJS is a simple templating language that lets you generate HTML markup with plain JavaScript.

```
$ npm install ejs
```

2.1.5 Morgan

HTTP request logger middleware for node.js.

```
$ npm install morgan
```

2.1.6 Nodemon

Nodemon is a utility that will monitor for any changes in your source and automatically restart your server. Perfect for development. Install it using npm. Just use **nodemon** instead of node to run your code, and now your process will automatically restart when your code changes.

```
$ npm i nodemon -D
```

3. Run the Express Server

Before you need to do the step One .

To run the Express server you need to execute the following command in a terminal.

```
$ npm run dev
```

```
Mr Mejia@LAPTOP-5TJA26HR MINGW64 /c/Ingenieria_Software/Web_Development/Nodejs/Crud-nodejs-mongodb (master)
$ npm run dev

> crud-nodejs-mongodb@1.0.0 dev C:\Ingenieria_Software\Web_Development\Nodejs\Crud-nodejs-mongodb
> nodemon src/app.js

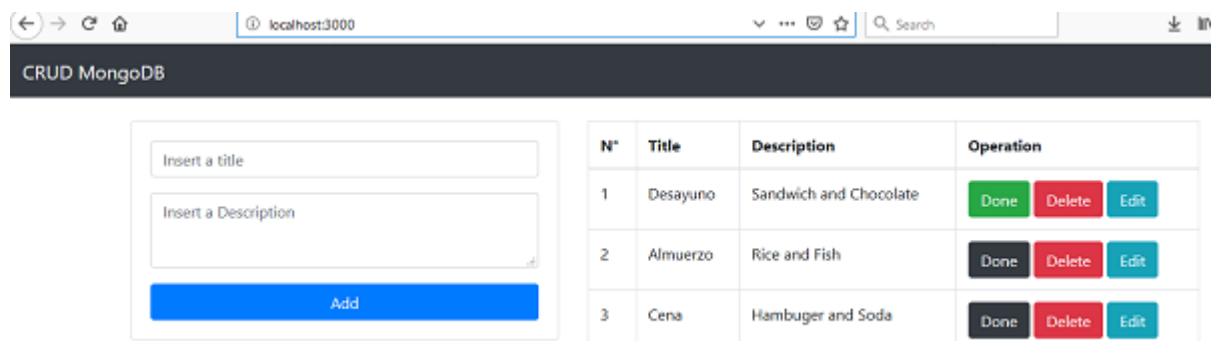
[nodemon] 1.18.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node src/app.js`
(node:7744) DeprecationWarning: current URL string parser is deprecated, and will be removed in a future version.
Use the new StringParser: true } to MongoClient.connect.
server on port 3000
Succesfull conection to DB
```

To test the Express server open the web service URL in a browser:

<http://127.0.0.1:3000/>

or

<http://localhost:3000/>



N°	Title	Description	Operation
1	Desayuno	Sandwich and Chocolate	<button>Done</button> <button>Delete</button> <button>Edit</button>
2	Almuerzo	Rice and Fish	<button>Done</button> <button>Delete</button> <button>Edit</button>
3	Cena	Hamburger and Soda	<button>Done</button> <button>Delete</button> <button>Edit</button>

MongoDB database access test : See the terminal

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL

POST /add 302 302.279 ms - 46
[ { status: true,
  _id: 5c7b15b740e5257cb5fdf7c0,
  title: 'Desayuno',
  description: 'Sandwich and Chocolate',
  __v: 0 },
  { status: false,
    _id: 5c7b1c4369125f94ed66982e,
    title: 'Almuerzo',
    description: 'Rice and Fish',
    __v: 0 },
  { status: false,
    _id: 5c7b349054b452eda50e0a76,
    title: 'Cena',
    description: 'Hamburger and Soda',
    __v: 0 },
  { status: false,
    _id: 5c7d86e6741f0c1e40db509b,
    title: 'Test',
    description: 'Es un test\r\n',
    __v: 0 } ]
GET / 200 10.671 ms - 5193
□
```