# How to connect MongoDB using mongoose in Node JS

*Mongoose.js* is a Node.js library that connects your **MongoDB** clusters or collections to your Node.js application. You can use it to construct schemas for your documents. When it comes to generating and working with schemas, Mongoose has a lot to offer. We'll look at how to connect a MongoDB instance to a Node.js application in this tutorial.

## What is MongoDB

MongoDB is one of the most popular No-SQL databases in use today among developers. Instead of SQL objects, NoSQL databases allow developers to transmit and receive data as JSON documents. Mongoose can be used to work with MongoDB in a Node.js application.

## Prerequisites:

Before we go any further, you'll need the following:

1. You have Node.js installed on your computer.
2. On your PC, there is a MongoDB instance running. If you want to utilize [Mongo DB Atlas](https://www.mongodb.com/), you won't need this.
3. Some experience with Node.js and Express.js is required.

## Setting Up Environment:

1. Create an empty directory with any name you want.
2. Open the folder with any code editor. I would like to prefer VS Code.
   1. You can download vs code from here. <https://code.visualstudio.com/>
3. Now press ctrl + shift + ~ to open terminal.
4. Now type ***npm init -y*** and hit enter. It will create package.json file and install necessary node modules
5. Now type ***npm install express nodemon cors dotenv mongoose***
6. Now create an empty file namely server.js and put below code in it.

const express = require("express");

const cors = require("cors");

const dotEnv = require("dotenv");

const app = express();

//registering middlewares

dotEnv.config();

app.use(cors());

app.use(express.json());

//listening to port

const port = process.env.PORT || 5000;

app.listen(port, () => {

  console.log(`Server is running on ${port}`);

});

## Creating the Connection:

1. In your root directory, create folder “config”
2. Create an empty file, “db.js”
3. Create .env file in root directory and add variable MONGO\_URI = url here
   1. Directory Structure:

Graphical user interface, text

Description automatically generated

1. If you want to connect db locally replace MONGO\_URI with ***mongodb://localhost:27017/usersdb***

Where usersdb is database name, you can give yours.

1. If you want to connect using MONGO\_ATLAS, then do the following steps
   1. Signup or login for Mongo DB atlas via <https://account.mongodb.com/account/register?_ga=2.174802465.288724901.1646250007-570639881.1643975138>
   2. Graphical user interface, text, application

      Description automatically generatedOnce you login, click on create button
   3. Again, click on ***Create Cluster*** Button

Graphical user interface, text, application, chat or text message

Description automatically generated

* 1. Wait for few minutes, and then following dashboard will appear

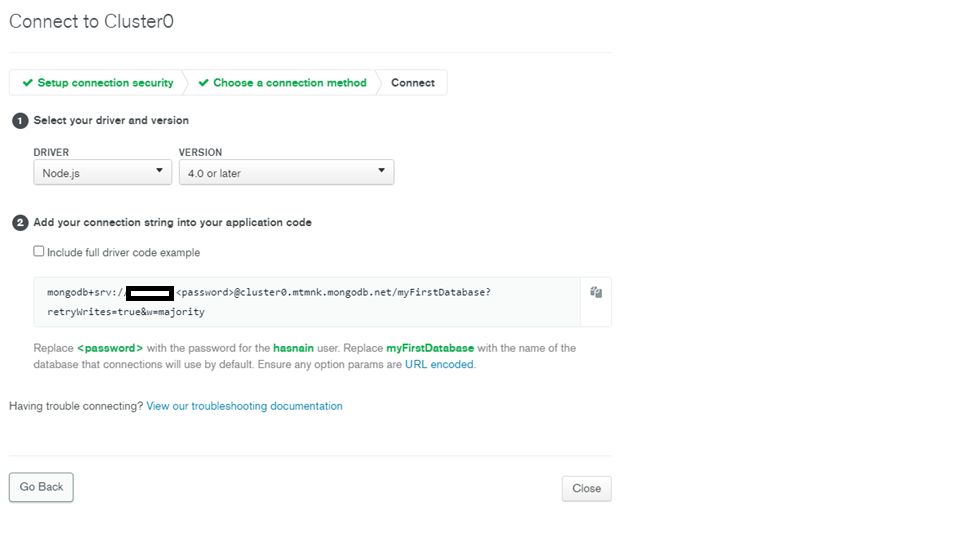
Graphical user interface, text, application

Description automatically generated

* 1. Click on connect
  2. Choose Connect your application

Graphical user interface, text, application, email

Description automatically generated

* 1. And finally copy the url from
  2. Finally add your name and password in url, which you have set when creating cluster and put it .env file

1. Inside config folder, In db.js put following code

const mongoose = require("mongoose");

const getConnection = async () => {

  try {

    const conn = await mongoose.connect(process.env.MONGO\_URI, {

      useNewUrlParser: true,

      useUnifiedTopology: true,

    });

    if (conn) {

      console.log(`MongoDB Connected on ${conn.connection.host}`);

    } else {

      console.log("Failed to connect DB");

    }

  } catch (error) {

    console.log(`Failed with error: ${error.message}`);

  }

};

module.exports = getConnection;

1. Now update server.js file as

const express = require("express");

const cors = require("cors");

const dotEnv = require("dotenv");

const getConnection = require("./config/db");

const app = express();

//registering middlewares

dotEnv.config();

app.use(cors());

app.use(express.json());

//connecting TO DB

getConnection();

//listening to port

const port = process.env.PORT || 5000;

app.listen(port, () => {

  console.log(`Server is running on ${port}`);

});

In package.json file add under scripts,

Graphical user interface

Description automatically generated

1. Finally go to terminal, and type ***npm start*** and hit enter.

Text

Description automatically generated

Hurrah! 😊 we are connected to our Data Base now. I hope you enjoyed this tutorial. You can read more about it at <https://mongoosejs.com/>