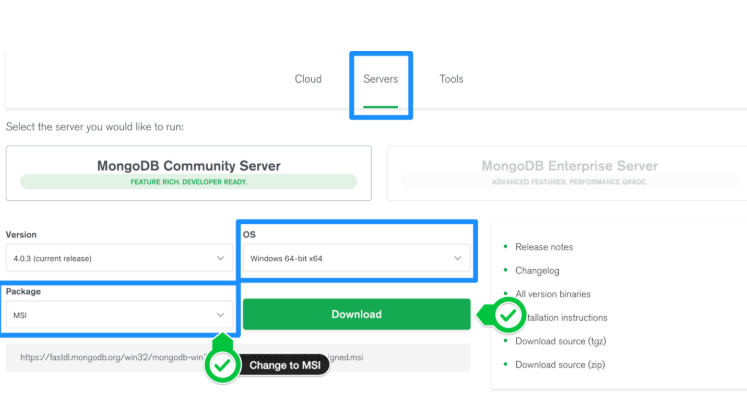
MagoDB(No sql) and mysql

1. Setup MagoDB:

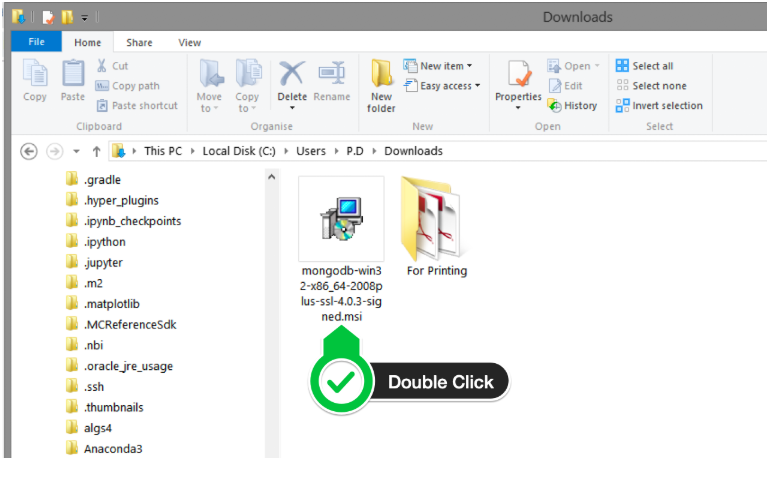
Step 1 : Download the MongoDB MSI Installer Package

Head over here and download the current version of MongoDB. Make sure you select MSI as the package you want to download.

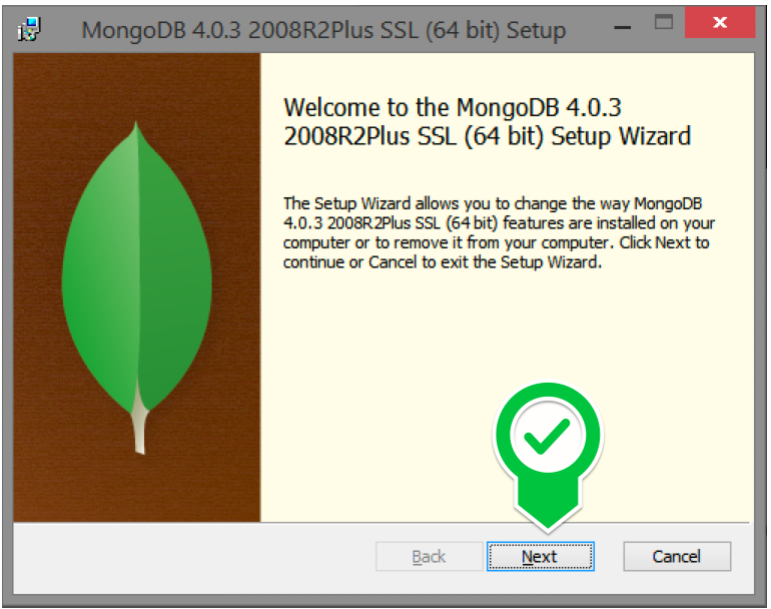


Step 2 : Install MongoDB with the Installation Wizard

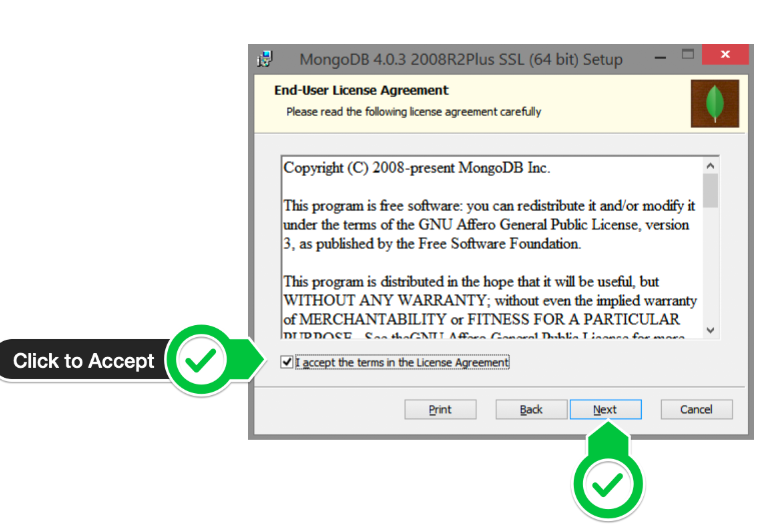
A. Make sure you are logged in as a user with Admin privileges. Then navigate to your downloads folder and double click on the .msi package you just downloaded. This will launch the installation wizard.



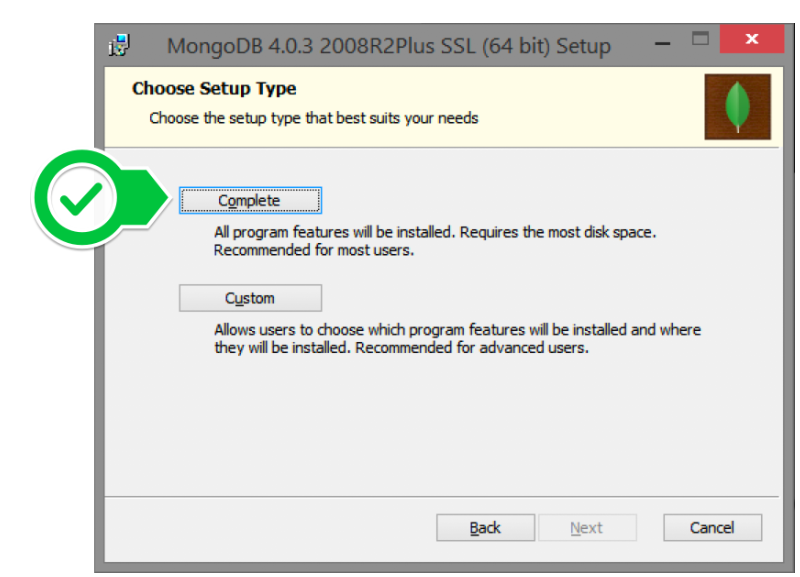
B. Click Next to start installation.



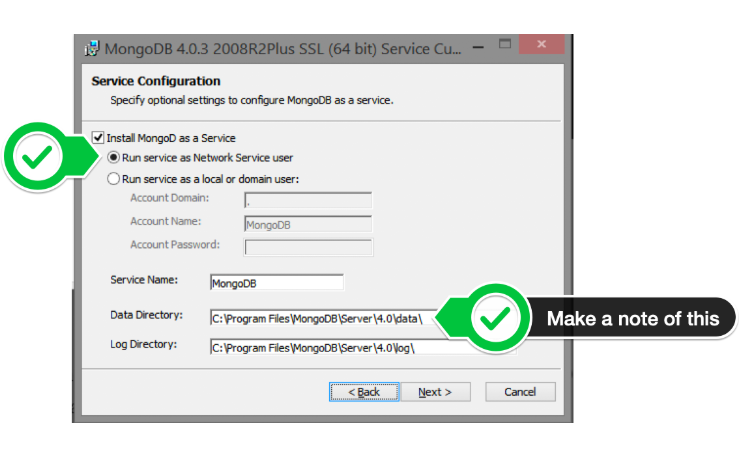
C. Accept the licence agreement then click Next.



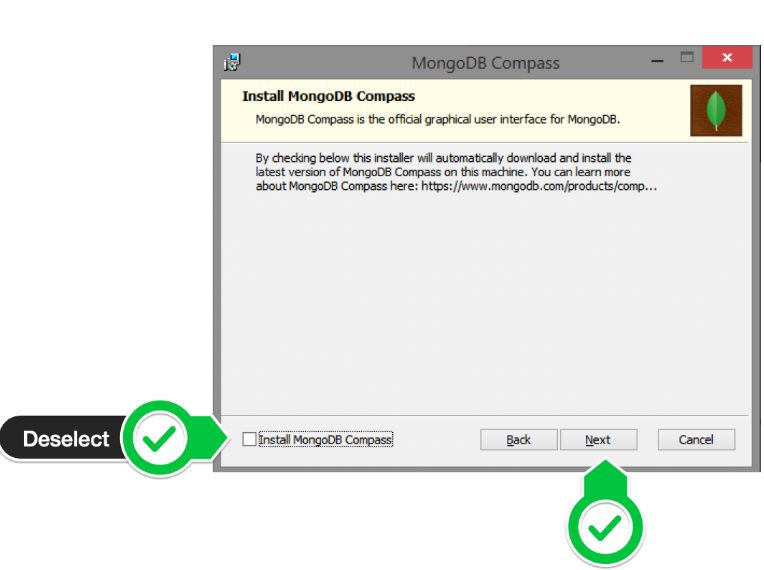
D. Select the Complete setup.



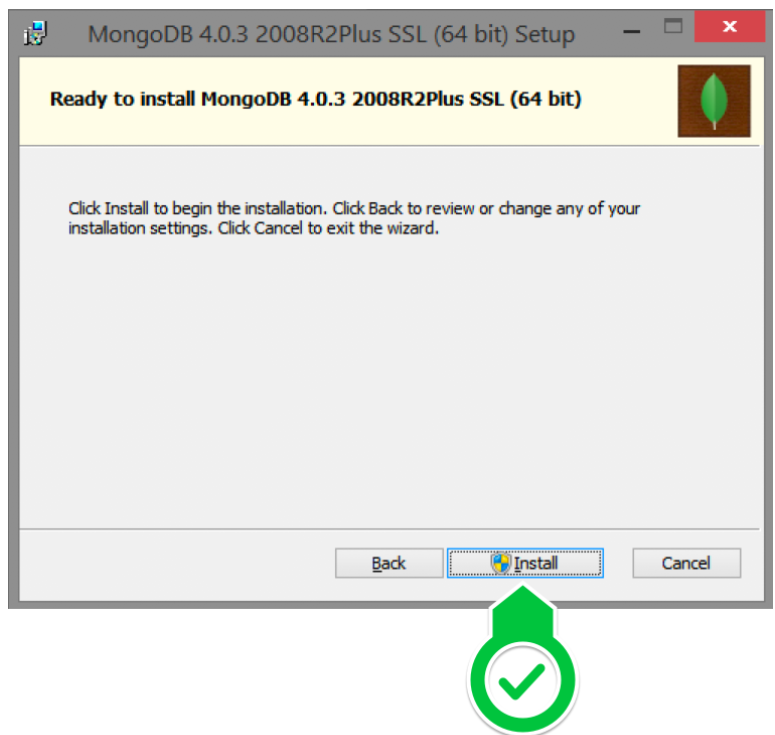
E. Select “Run service as Network Service user” and make a note of the data directory, we’ll need this later.



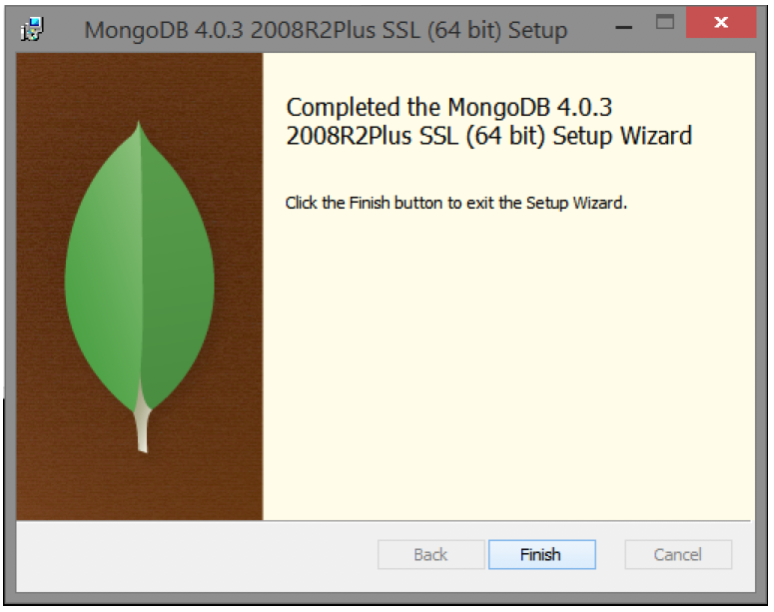
F. We won’t need Mongo Compass, so deselect it and click Next.



G. Click Install to begin installation.

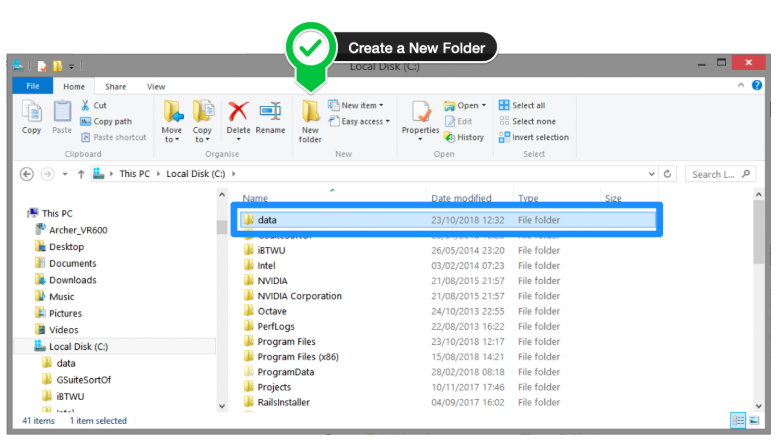


F. Hit Finish to complete installation.

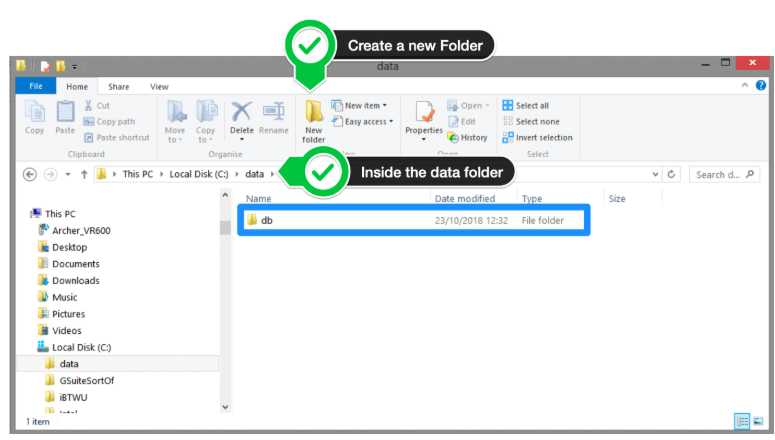


Step 3: Create the Data Folders to Store our Databases

1. Navigate to the C Drive on your computer using Explorer and create a new folder called data here.



1. Inside the data folder you just created, create another folder called db.

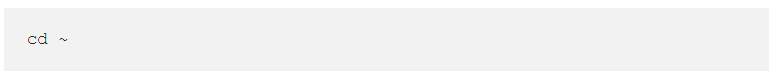


Step 4: Setup Alias Shortcuts for Mongo and Mongod

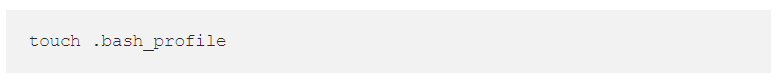
Once installation is complete, we’ll need to set up MongoDB on the local system.

A. Open up your Hyper terminal running Git Bash.

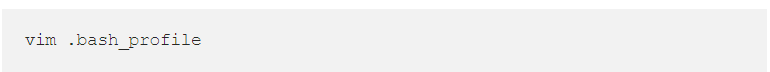
B. Change directory to your home directory with the following command:



1. Here, we’re going to create a file called .bash\_profile using the following command:



1. Open the newly created .bash\_profile with vim using the following command:

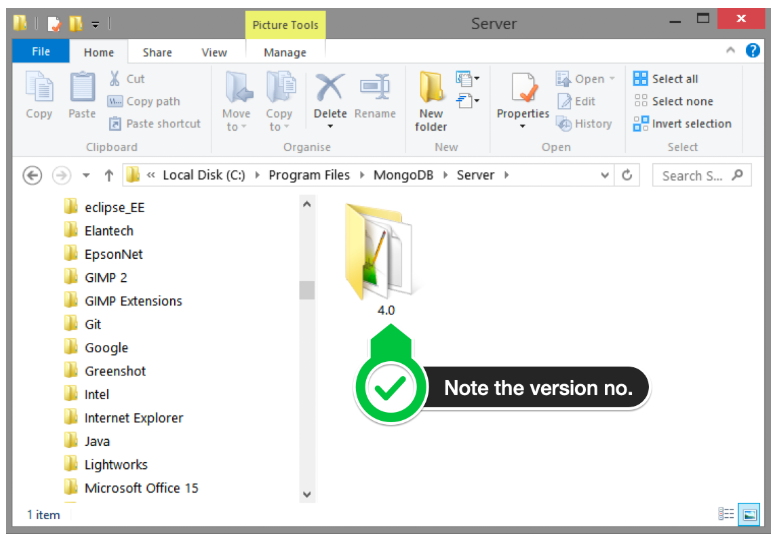


E. In vim, hit the I key on the keyboard to enter insert mode.

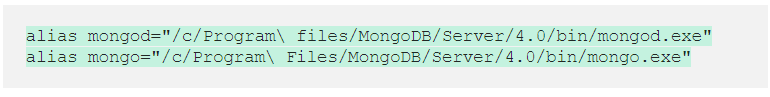


F. In your explorer go to C → Program Files → MongoDB → Server

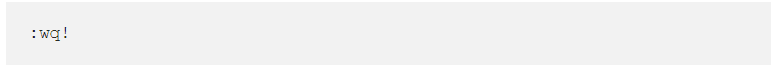
Now you should see the version of your MongoDB.



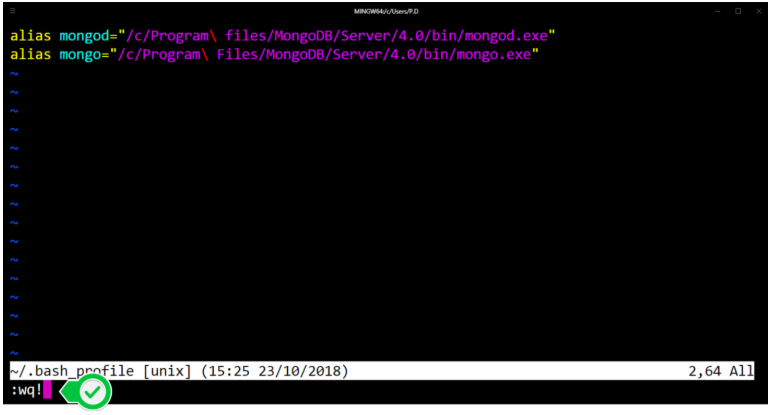
G. Paste in the following code into vim, make sure your replace the 4.0 with your version that you see in explorer



1. Hit the Escape key on your keyboard to exit the insert mode. Then type



to save and exit Vim.



Step 5 : Verify That Setup was Successful

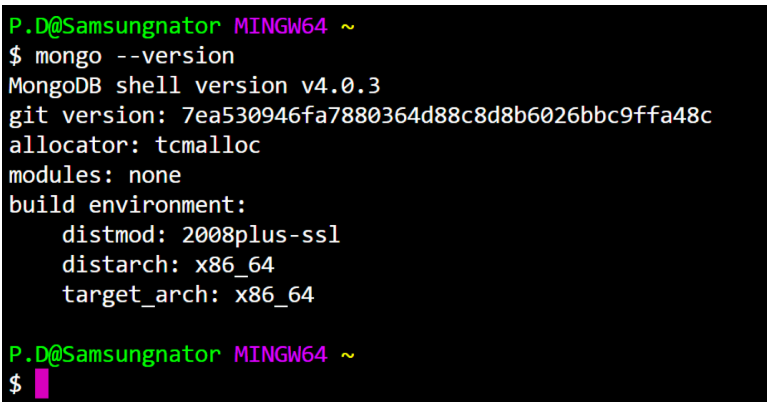
A. Close down the current Hyper terminal and quit the application.

B. Re-launch Hyper.

C. Type the following commands into the Hyper terminal:



Once you’ve hit enter, you should see something like this:



This means that you have successfully installed and setup MongoDB on your local system!

If you see something that looks like bash mongo command not found, then make sure you check back at all the steps above and follow it step-by-step making sure there are no typos and you haven’t missed any of the steps.

1. How to use MagoDB
2. Compare between No sql and mysql

MySQL NoSQL

1. Nature

Relational database in nature. Non-relational databse in nature.

1. Design

Modeled based on the concept Modeled based on the concept

Of “document”

Of “table”.

1. Scalable

Begin relational in nature Easily scalable big data as compared to relational.

,can be tough task to scale

bigg data.

1. Model

Detailed databse model needs No need to develop a detailed database model.

to be in place before creation.

1. Community

Vast and expert community Community is growing rapidly and compare to MyDQL.

is available.

1. Standardization

SQL is standard language. Lack of a standard query language.

1. Schema

Schema is rigid.. Dynamic schema is key benefit of NoSQL.

1. Flexibility

Not so flexible design-wise, New column or fields can be inserted without existing design

new column insertion

affects design.

1. Benefits of MagoDB

* Free
* Easy to learn
* Flexible (schema less)
* Accessible with a wide variety of platforms (JavaScript, python, etc.)
* Great documentation
* Very scalable
* Supports very complex queries
* Probably many others.

1. How to use it with NodeJS

Set up

Before we begin, we need to ensure you’ve completed a few prerequisite steps.

1. Install Node.js

First, make sure you have a supported version of Node.js installed (the MongoDB Node.js Driver requires Node 4.x or greater and for these examples, I've used Node.js 10.16.3).

1. Install the MongoDB Node.js Driver

The MongoDB Node.js Driver allows you to easily interact with MongoDB databases from within.

If you don’t have the MongoDB Node.js Driver installed, you can install it with the following command. Node.js applications.

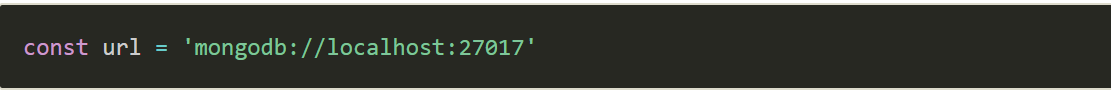


1. Connecting to MongoDB

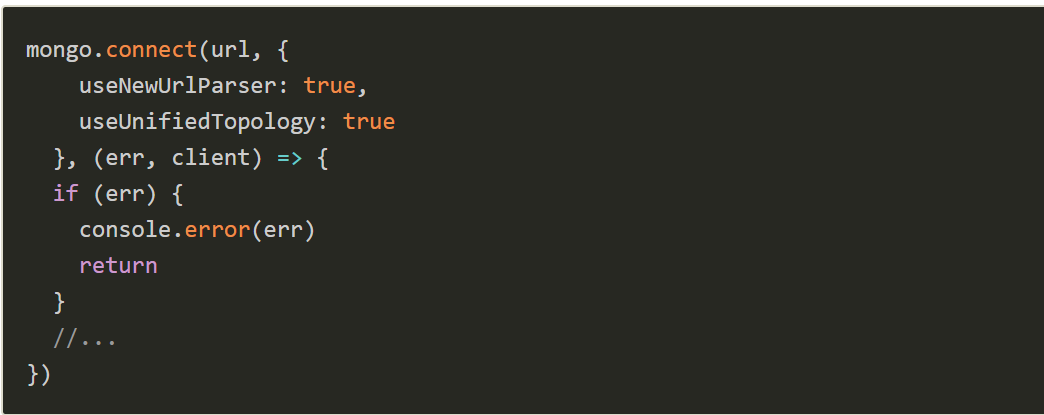
You require the mongodb package and you get the MongoClient object from it.



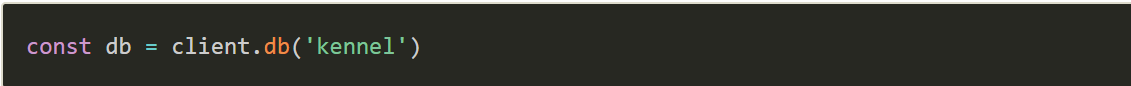
Create a URL to the MongoDB server. If you use MongoDB locally, the URL will be something like mongodb://localhost:27017, as 27017 is the default port.



Then use the mongo.connect() method to get the reference to the MongoDB instance client:



Now you can select a database using the client.db() method:



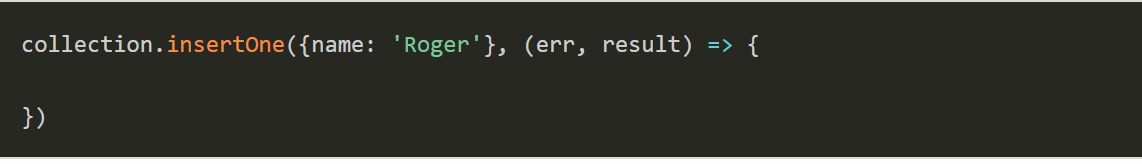
1. Create and get a collection

You can get a collection by using the db.collection() method. If the collection does not exist yet, it’s created.

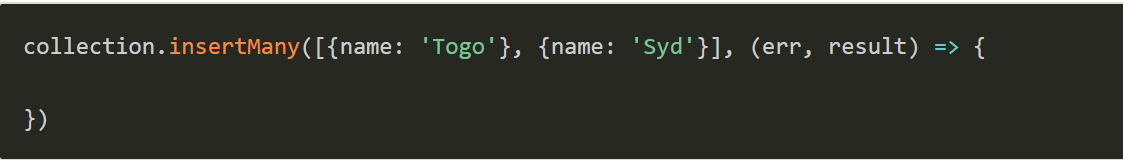


1. Insert data into a collection a Document

Add to app.js the following function which uses the insertOne() method to add an object dogs collection.

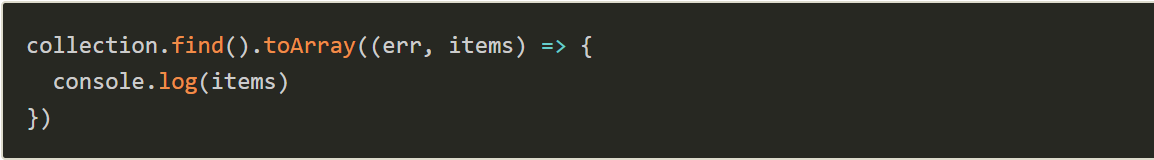


You can add multiple items using insertMany(), passing an array as the first parameter:

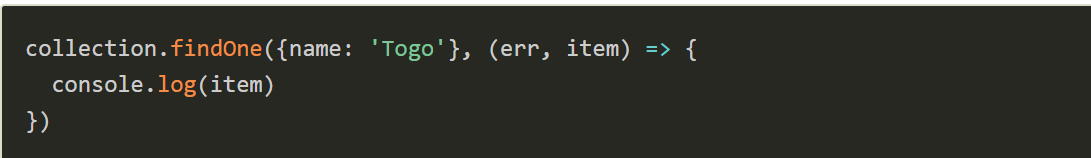


1. Find all documents

Use the find() method on the collection to get all the documents added to the collection:

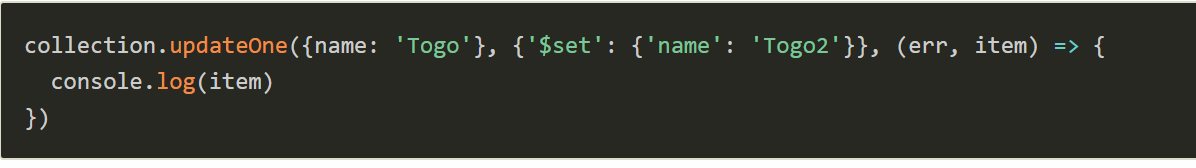


If you know you are going to get one element, you can skip the toArray() conversion of the cursor by calling findOne():



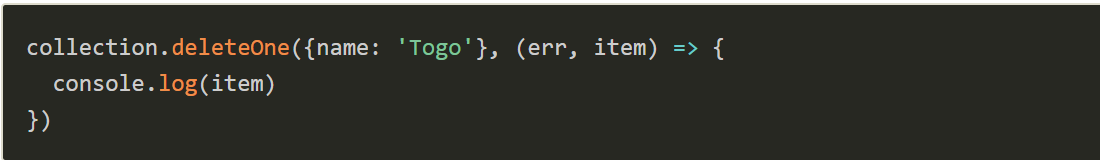
1. Update an existing document

Use the updateOne() method to update a document:



1. Delete a document

Use the deleteOne() method to delete a document:



1. Closing the connection

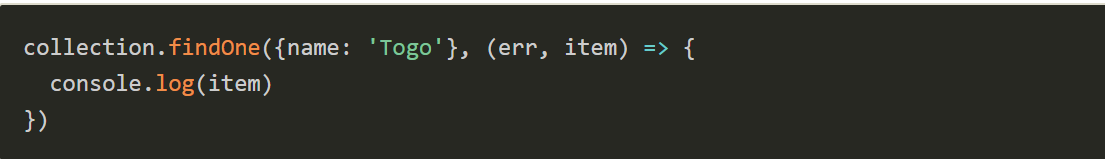
Once you are done with the operations you can call the close() method on the client object:



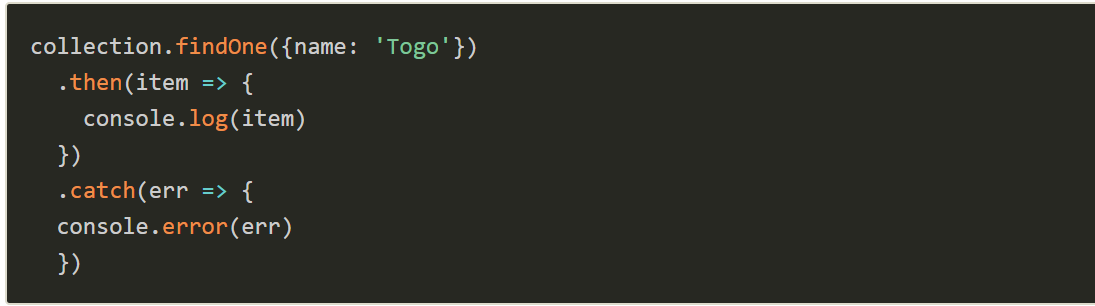
1. Use promises or async/await

I posted all those examples using the callback syntax. This API supports promises (and async/await) as well.

For example this



Can be used with promises:



or async/await:

