Connecting to mongoDB

<https://www.npmjs.com/package/mongodb>

help at <http://mongodb.github.io/node-mongodb-native/contents.html>

download npm at <https://www.npmjs.com/package/mongodb>

to access the database locally using 3rd part software

download <https://robomongo.org/>

<http://www.informit.com/articles/article.aspx?p=2301560>

<https://www.youtube.com/watch?v=Do_Hsb_Hs3c>

* **What is MongoDB**
* **Connect MongoDB database with node**
* **Other commands:**
  + **Finding**
  + **Inserting**
  + **Deleting**
  + **Updating**
* **Robomongo**

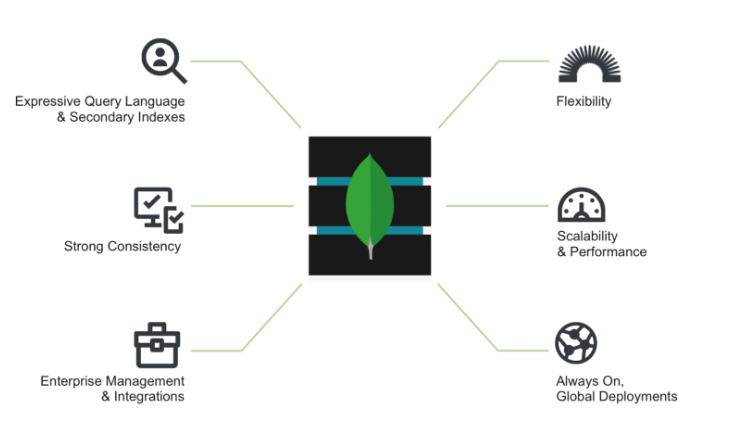
**What is MongoDB**

The latest in industries growing database. MongoDB is free and open source cross platform document-orientated database program. MongoDB uses JSON like documents and is considered to be a no-SQL database. This allows schemas to be changed on the fly and do not require downtime. This approach to document-orientated database allows you to spend less time prepping and organizing your data and makes it easy to store and combine data of any structure.

Main Points:

* Fast, Iterative Development
* Flexible Data Model
* Multi-Datacenter Scalability
* Integrated Feature Set
* Lower TCO
* Long-Term commitment

More @ <https://www.mongodb.com/mongodb-architecture>



From: https://www.mongodb.com/mongodb-architecture

High level diagram view of MongoDB’s characteristics

**Connect MongoDB database with node**

First install mongoDB dependacy, use command:

**“npm install mongodb –save”**

**Info @** [**https://www.npmjs.com/package/mongodb**](https://www.npmjs.com/package/mongodb)

Here I will create a seperate file to handle database functions and then use this in the server.js, this will allow me to split up different functionality into separate files.

Require the Npm module we just installed

**MongoClient = require('mongodb').MongoClient,**

Set variables for the URL, username and password (if necessary)

**dburl= "mongodb://URL:PortNumber/yourDatabaseName";**

**usr = "UserName";**

**pass = "Password";**

connect to mongo database

**exports.connect = function(){**

**MongoClient.connect(dburl, function(err, db) {**

**console.log("Connected correctly to server");**

**})**

**}**

Or connect and authenticate

**exports.connect = function(){**

**MongoClient.connect(dburl, function(err, db) {**

**db.authenticate(usr, pass, function(err, result) {**

**doSomething(db, function() {**

**console.log("Connected correctly to server");**

**});**

**})**

**})**

**}**

Example:

Connect, Authenticate, and read from database table “test” and return all results

**exports.findDocuments = function(req, res) {**

**MongoClient.connect(dburl, function(err, db) {**

**console.log("Connected correctly to server")**

**db.authenticate(usr, pass, function(err, result) {**

**console.log("Authenticated correctly to server")**

**var collection = db.collection('test');**

**collection.find({}).toArray(function(err, docs) {**

**db.close();**

**res.send( { data: docs } ) ;**

**});**

**})**

**})**

**};**

**Other commands:**

**Finding documents**

**exports.findDocuments = function(req, res) {**

**MongoClient.connect(dburl, function(err, db) {**

**console.log("Connected correctly to server")**

**db.authenticate(usr, pass, function(err, result) {**

**console.log("Authenticated correctly to server")**

**var collection = db.collection('test');**

**collection.find({}).toArray(function(err, docs) {**

**db.close();**

**res.send( { data: docs } ) ;**

**});**

**})**

**})**

**};**

**Inserting Documents**

**exports.insertDocuments = function(record) {**

**MongoClient.connect(dburl, function(err, db) {**

**console.log("Connected correctly to server")**

**db.authenticate(usr, pass, function(err, result) {**

**console.log("Authenticated correctly to server")**

**var collection = db.collection('test');**

**collection.insertMany( [ record ], function(err, result) {**

**console.log("Inserted document into the document collection");**

**db.close();**

**});**

**})**

**})**

**};**

**Deleting Documents**

**exports.deleteDocument = function(record) {**

**MongoClient.connect(dburl, function(err, db) {**

**console.log("Connected correctly to server")**

**db.authenticate(usr, pass, function(err, result) {**

**console.log("Authenticated correctly to server")**

**var collection = db.collection('test');**

**collection.deleteOne( record , function(err, result) {**

**console.log("Deleted document from document collection");**

**db.close();**

**});**

**})**

**})**

**};**

**Updating Documents**

**Robomongo**

Robomongo is a tool that can be used to manage your MongoDB, it provides a UI and GUI to allow the user to interact with and manage their database.

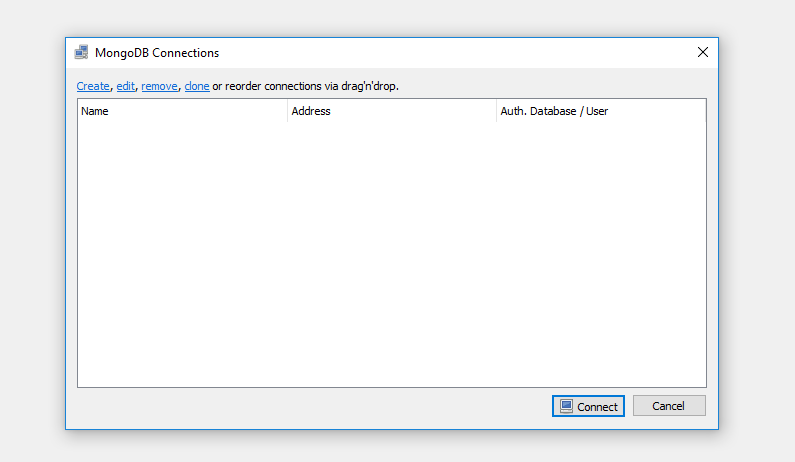
More @ <https://robomongo.org/>

**How to connect Robomongo to your database**

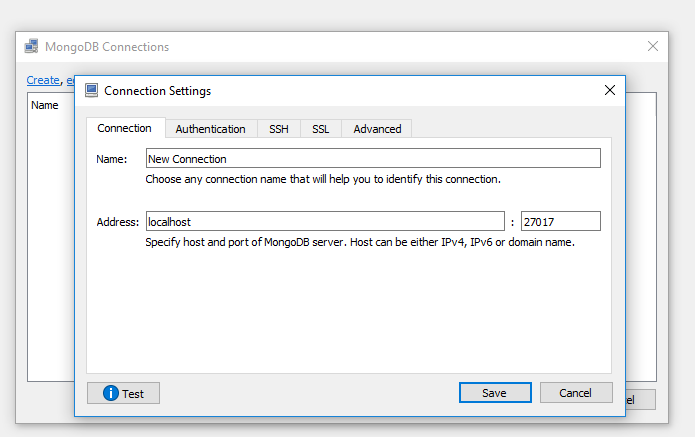
Download Robomongo at <https://robomongo.org/download> , you will use robomongo locally to manage your database.

To Connect:

create a new connection, by clicking the top left “create” button

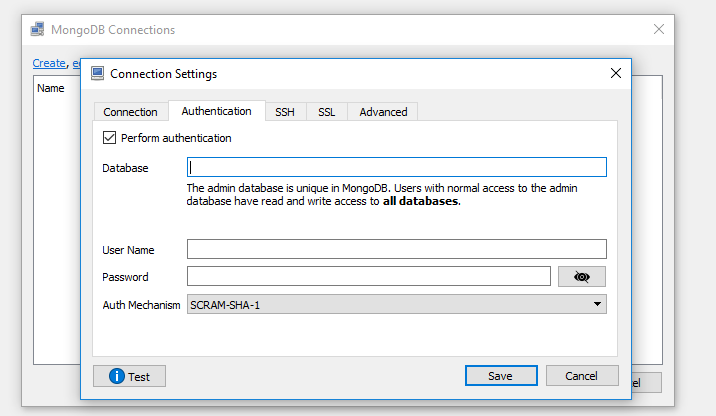


Input the name you would like to call this connection, and then supply it with the correct address and port number



Next we need to add authentication

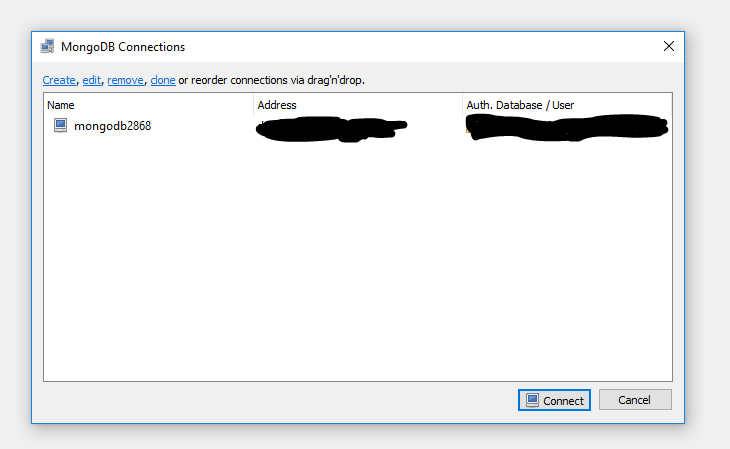
Enter the database name, username and password.



For me I changed the security to MONGODB-CR



Then finally you have robomongo setup to connect to your database.



This is what you will see once you connect

