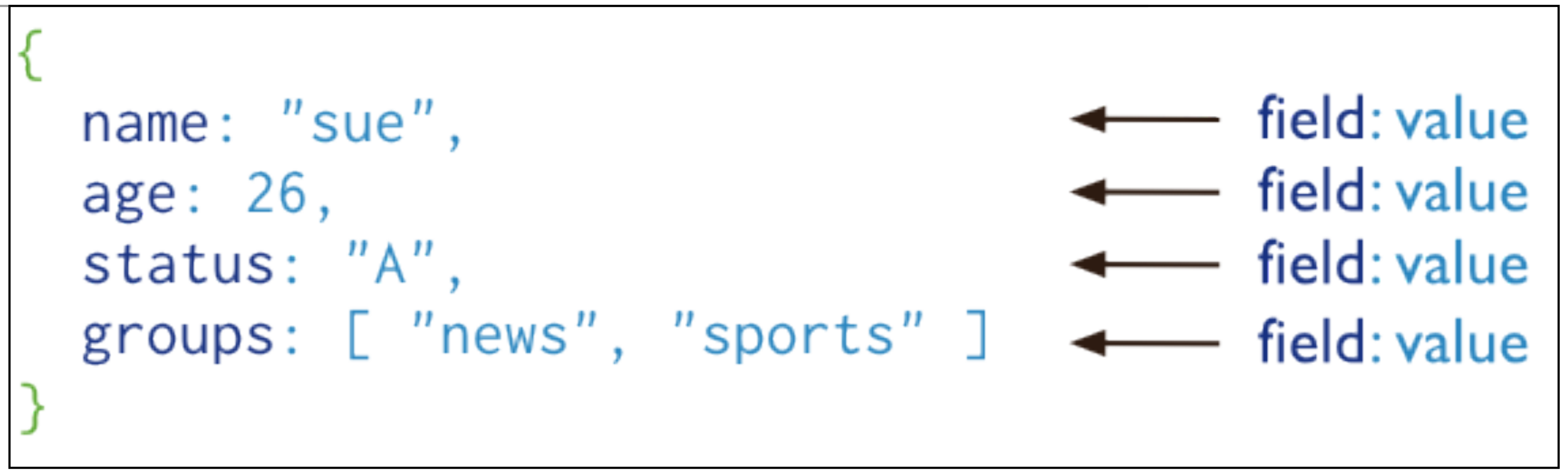
Setting up Mongo

mongoDB

Full Stack Web Development

MongoDB

- MongoDB is an open-source document database that provides high performance, high availability, and automatic scaling.
- A record in MongoDB is a document, which is a data structure composed of field and value pairs.
- MongoDB documents are similar to JSON objects.
- The values of fields may include other documents, arrays, and arrays of documents.



```
{
   name: "sue",                                 ←— field: value
   age: 26,                                     ←— field: value
   status: "A",                                 ←— field: value
   groups: [ "news", "sports" ]                 ←— field: value
}
```

- Documents (i.e. objects) correspond to native data types in JavaScript
- Embedded documents and arrays reduce need for expensive joins.

mongoose

elegant mongodb object modeling for node.js

“Mongoose provides a straight-forward, schema-based solution to model your application data. It includes built-in type casting, validation, query building, business logic hooks and more, out of the box”

Why Mongoose?

- Node focussed Object Document Manager Mongo & Node

- Can simplify MongoDB development, particularly as a learning tool for first contact with Mongo

- Re-introduces Schema to node, and simplifies considerably evolution of moderately complex applications.

- Can be viewed as an easier ‘way into’ mongo, but understanding and familiarity with MongoDB documentation still required.

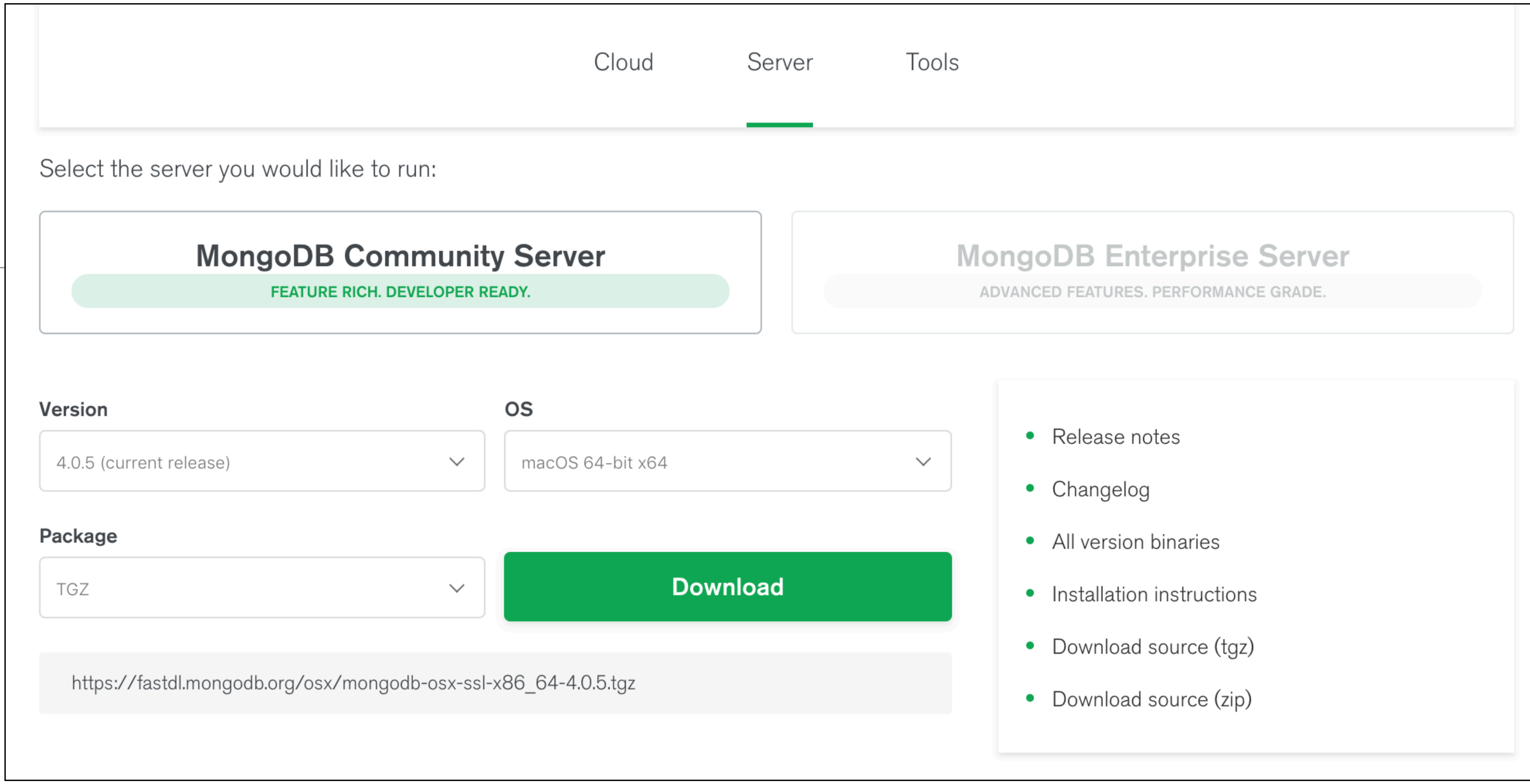
Learning Mongo via Mongoose

- Shortcut to understanding the basics
- Closer to Object Relational Mapping libraries like JPA/Hibernate
- Will require deeper understanding of core Mongo API at a later stage, but Mongoose First is a useful approach for now.

elegant mongodb object modeling for node.js

Setup (1)

- Install MongoDB
- Launch Mongo



Select the server you would like to run:

MongoDB Community Server
FEATURE RICH. DEVELOPER READY.

MongoDB Enterprise Server
ADVANCED FEATURES. PERFORMANCE GRADE.

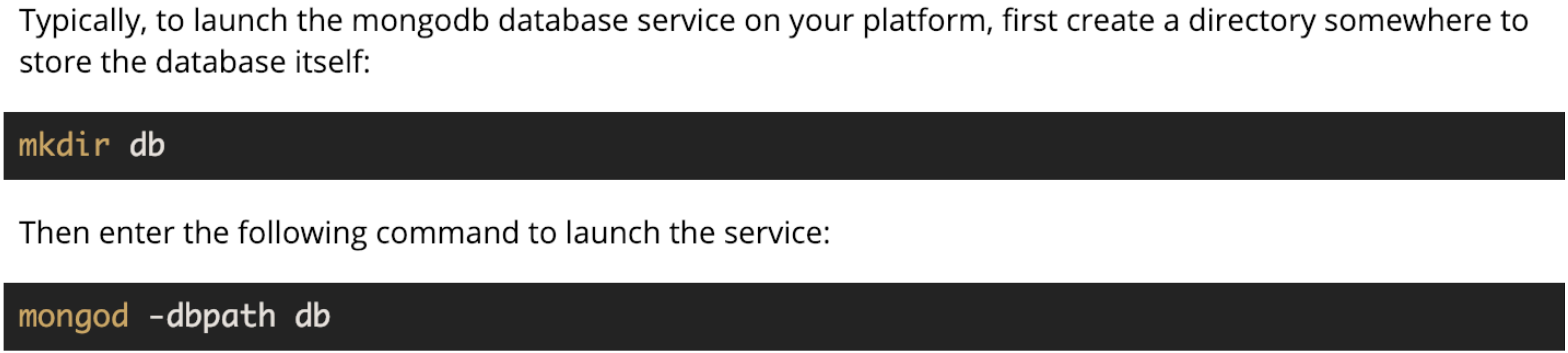
Version: 4.0.5 (current release)

OS: macOS 64-bit x64

Package: TGZ

https://fastdl.mongodb.org/osx/mongodb-osx-ssl-x86_64-4.0.5.tgz

- Release notes
- Changelog
- All version binaries
- Installation instructions
- Download source (tgz)
- Download source (zip)

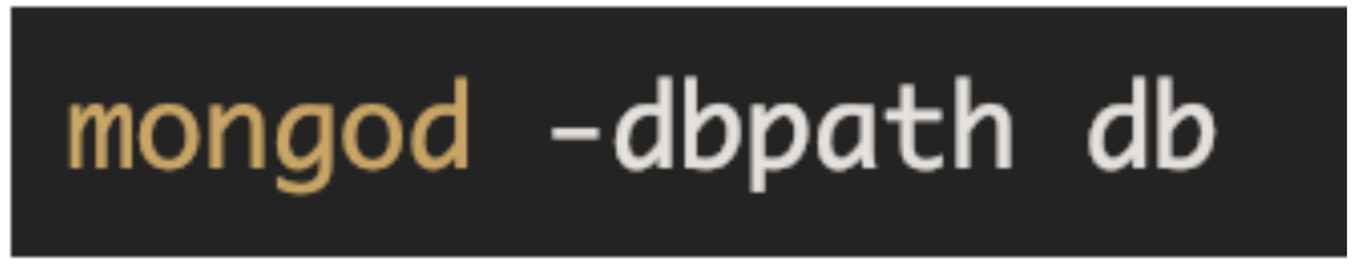


Typically, to launch the mongodb database service on your platform, first create a directory somewhere to store the database itself:

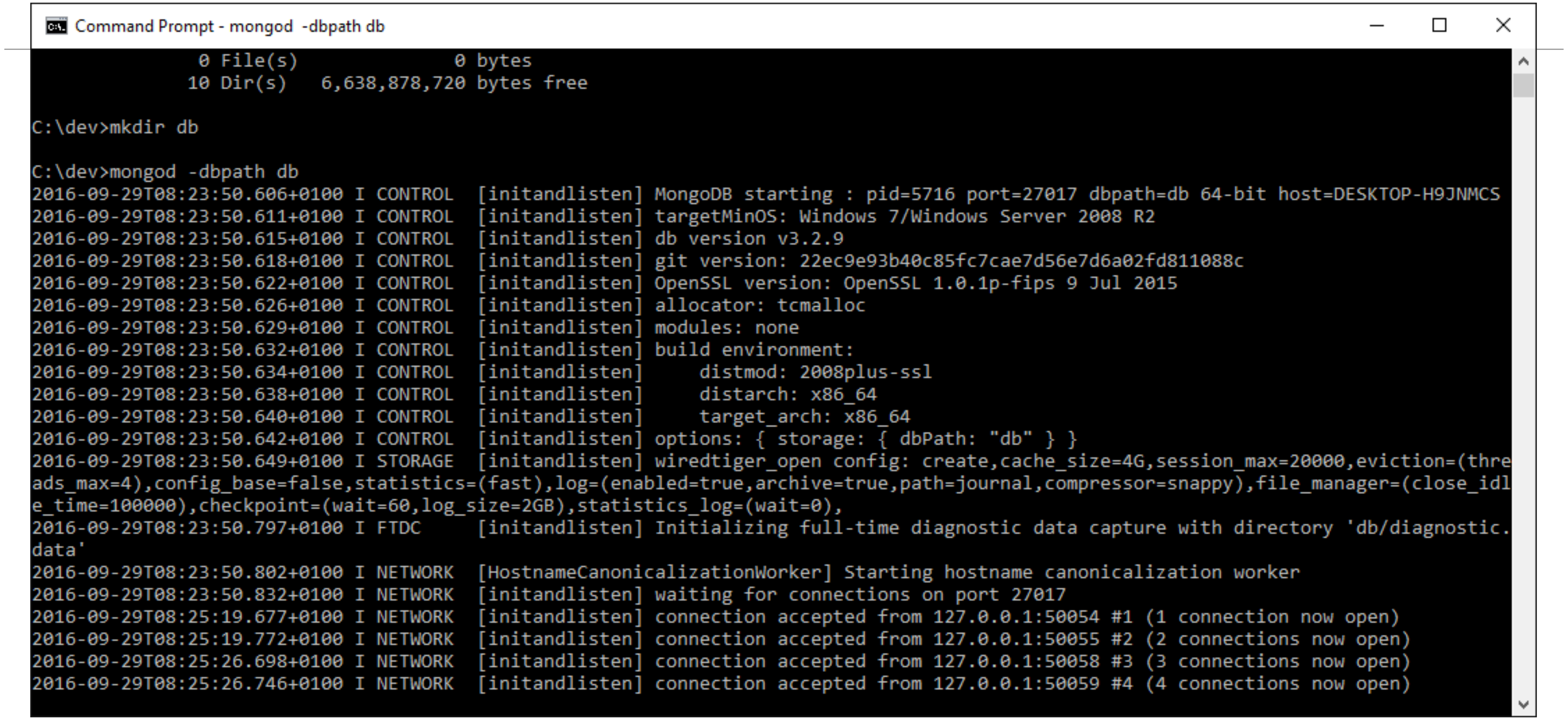
```
mkdir db
```

Then enter the following command to launch the service:

```
mongod -dbpath db
```



```
mongod -dbpath db
```

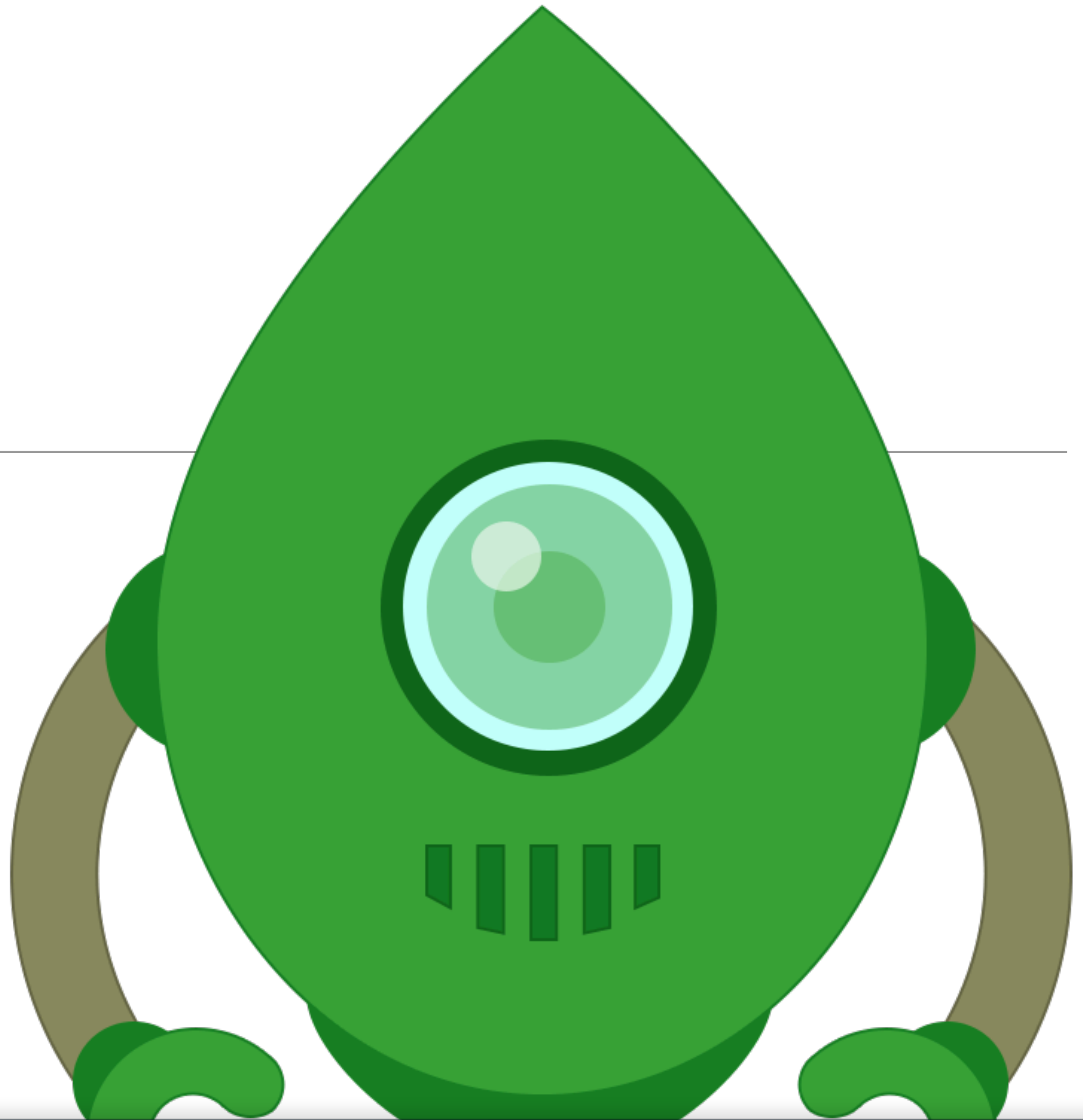


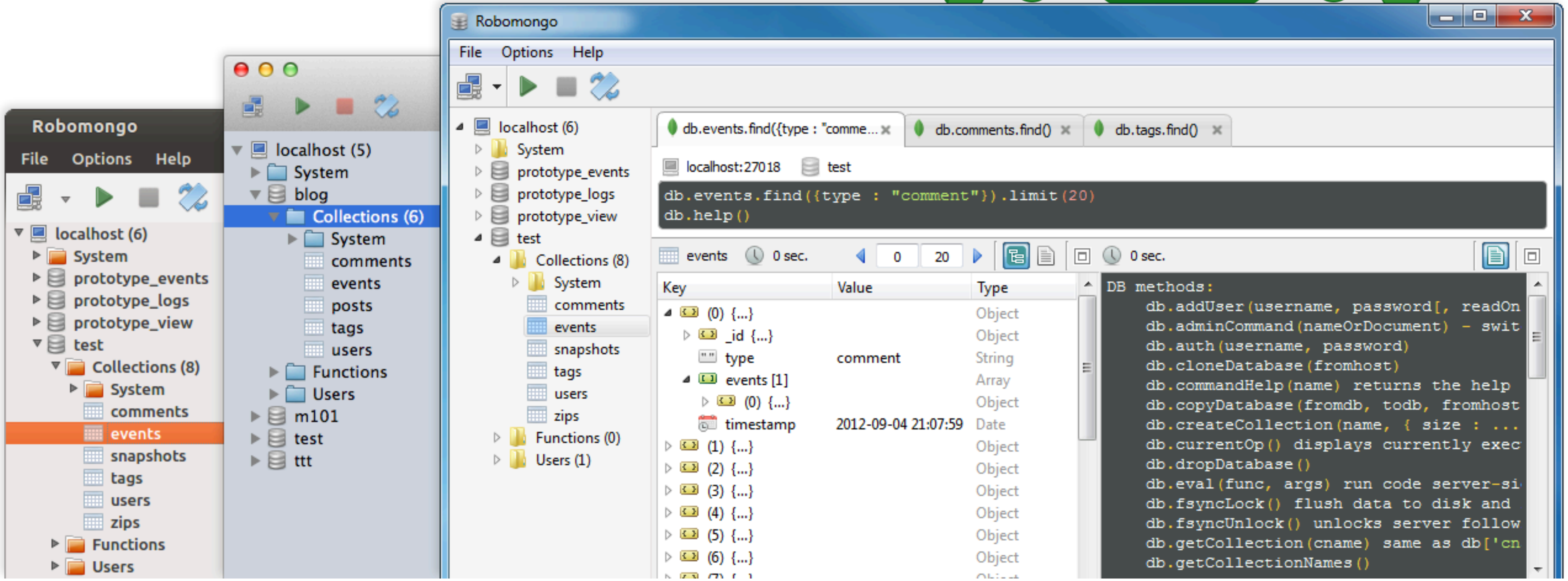
```
               0 File(s)              0 bytes
              10 Dir(s)   6,638,878,720 bytes free

C:\dev>mkdir db

C:\dev>mongod -dbpath db
2016-09-29T08:23:50.606+0100 I CONTROL  [initandlisten] MongoDB starting : pid=5716 port=27017 dbpath=db 64-bit host=DESKTOP-H9JNMCS
2016-09-29T08:23:50.611+0100 I CONTROL  [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2016-09-29T08:23:50.615+0100 I CONTROL  [initandlisten] db version v3.2.9
2016-09-29T08:23:50.618+0100 I CONTROL  [initandlisten] git version: 22ec9e93b40c85fc7cae7d56e7d6a02fd811088c
2016-09-29T08:23:50.622+0100 I CONTROL  [initandlisten] OpenSSL version: OpenSSL 1.0.1p-fips 9 Jul 2015
2016-09-29T08:23:50.626+0100 I CONTROL  [initandlisten] allocator: tcmalloc
2016-09-29T08:23:50.629+0100 I CONTROL  [initandlisten] modules: none
2016-09-29T08:23:50.632+0100 I CONTROL  [initandlisten] build environment:
2016-09-29T08:23:50.634+0100 I CONTROL  [initandlisten]     distmod: 2008plus-ssl
2016-09-29T08:23:50.638+0100 I CONTROL  [initandlisten]     distarch: x86_64
2016-09-29T08:23:50.640+0100 I CONTROL  [initandlisten]     target_arch: x86_64
2016-09-29T08:23:50.642+0100 I CONTROL  [initandlisten] options: { storage: { dbPath: "db" } }
2016-09-29T08:23:50.649+0100 I STORAGE  [initandlisten] wiredtiger_open config: create,cache_size=4G,session_max=20000,eviction=(thre
ads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idl
e_time=100000),checkpoint=(wait=60,log_size=2GB),statistics_log=(wait=0),
2016-09-29T08:23:50.797+0100 I FTDC     [initandlisten] Initializing full-time diagnostic data capture with directory 'db/diagnostic.
data'
2016-09-29T08:23:50.802+0100 I NETWORK  [HostnameCanonicalizationWorker] Starting hostname canonicalization worker
2016-09-29T08:23:50.832+0100 I NETWORK  [initandlisten] waiting for connections on port 27017
2016-09-29T08:25:19.677+0100 I NETWORK  [initandlisten] connection accepted from 127.0.0.1:50054 #1 (1 connection now open)
2016-09-29T08:25:19.772+0100 I NETWORK  [initandlisten] connection accepted from 127.0.0.1:50055 #2 (2 connections now open)
2016-09-29T08:25:26.698+0100 I NETWORK  [initandlisten] connection accepted from 127.0.0.1:50058 #3 (3 connections now open)
2016-09-29T08:25:26.746+0100 I NETWORK  [initandlisten] connection accepted from 127.0.0.1:50059 #4 (4 connections now open)
```

- This will log event to the console and serves as a useful check on the status of the service

Setup (2)

- Install Robo 3T
- A visual too for exploring and manipulation MongoDB Databases





```
db.events.find({type : "comment"}).limit(20)
db.help()
```

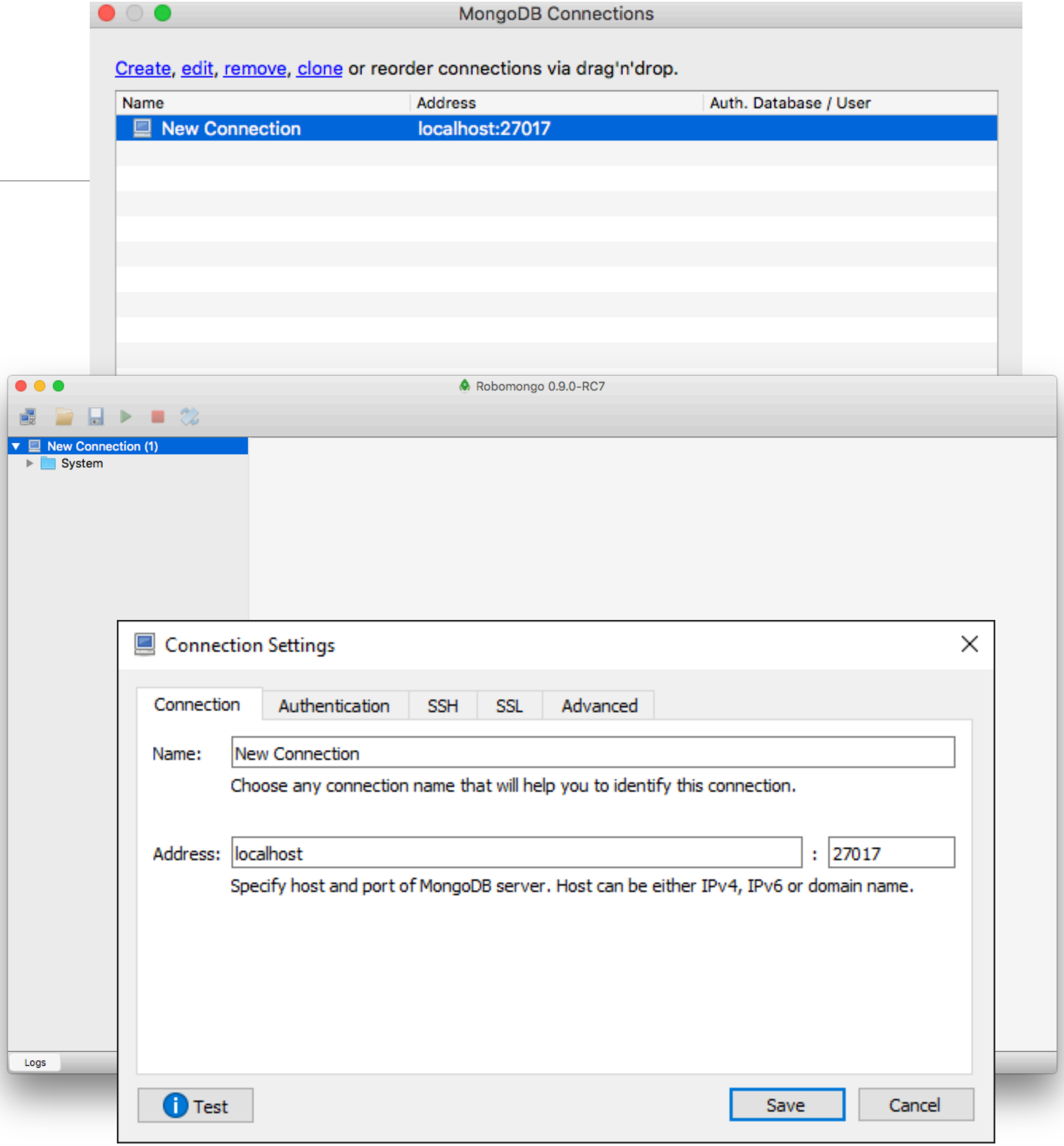
Key	Value	Type
(0) {...}		Object
_id {...}		Object
type	comment	String
events [1]		Array
(0) {...}		Object
timestamp	2012-09-04 21:07:59	Date
(1) {...}		Object
(2) {...}		Object
(3) {...}		Object
(4) {...}		Object
(5) {...}		Object
(6) {...}		Object

```
DB methods:
    db.addUser(username, password[, readOn
    db.adminCommand(nameOrDocument) - swit
    db.auth(username, password)
    db.cloneDatabase(fromhost)
    db.commandHelp(name) returns the help
    db.copyDatabase(fromdb, todb, fromhost
    db.createCollection(name, { size : ...
    db.currentOp() displays currently exec
    db.dropDatabase()
    db.eval(func, args) run code server-si
    db.fsyncLock() flush data to disk and
    db.fsyncUnlock() unlocks server follow
    db.getCollection(cname) same as db['cn
    db.getCollectionNames()
```

Setup (3)

- Launch Robo 3T and connect to the database you have already started

- Default may be localhost:27017



MongoDB Connections

Create, edit, remove, clone or reorder connections via drag'n'drop.

Name	Address	Auth. Database / User
New Connection	localhost:27017	

Robomongo 0.9.0-RC7

Connection Settings

Name: New Connection

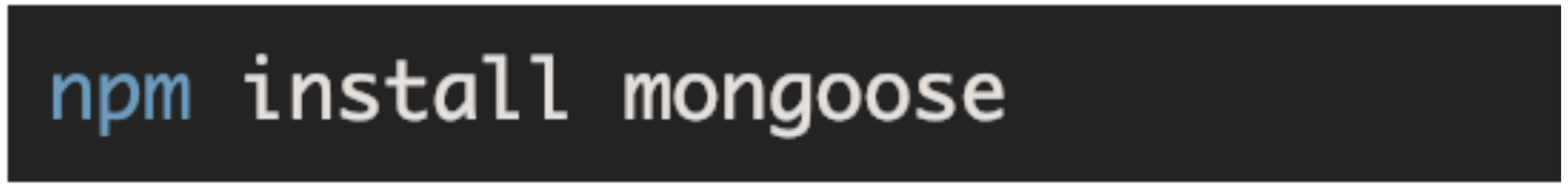
Choose any connection name that will help you to identify this connection.

Address: localhost : 27017

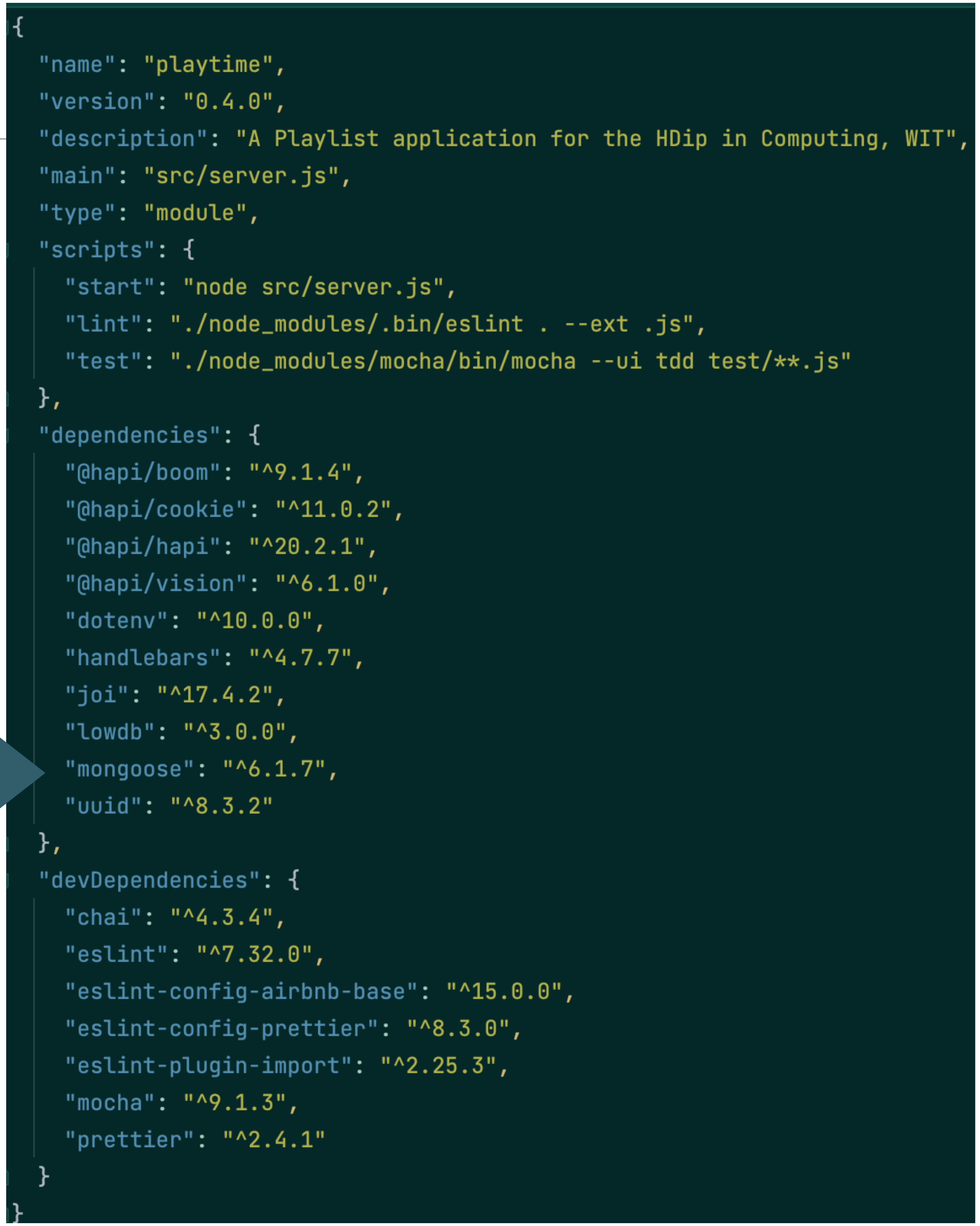
Specify host and port of MongoDB server. Host can be either IPv4, IPv6 or domain name.

Node & Mongoose

- mongoose package is our primary gateway to mongodb
- Packages includes complete API for our purposes



```
npm install mongoose
```



```
{
  "name": "playtime",
  "version": "0.4.0",
  "description": "A Playlist application for the HDip in Computing, WIT",
  "main": "src/server.js",
  "type": "module",
  "scripts": {
    "start": "node src/server.js",
    "lint": "./node_modules/.bin/eslint . --ext .js",
    "test": "./node_modules/mocha/bin/mocha --ui tdd test/**.js"
  },
  "dependencies": {
    "@hapi/boom": "^9.1.4",
    "@hapi/cookie": "^11.0.2",
    "@hapi/hapi": "^20.2.1",
    "@hapi/vision": "^6.1.0",
    "dotenv": "^10.0.0",
    "handlebars": "^4.7.7",
    "joi": "^17.4.2",
    "lowdb": "^3.0.0",
    "mongoose": "^6.1.7",
    "uuid": "^8.3.2"
  },
  "devDependencies": {
    "chai": "^4.3.4",
    "eslint": "^7.32.0",
    "eslint-config-airbnb-base": "^15.0.0",
    "eslint-config-prettier": "^8.3.0",
    "eslint-plugin-import": "^2.25.3",
    "mocha": "^9.1.3",
    "prettier": "^2.4.1"
  }
}
```

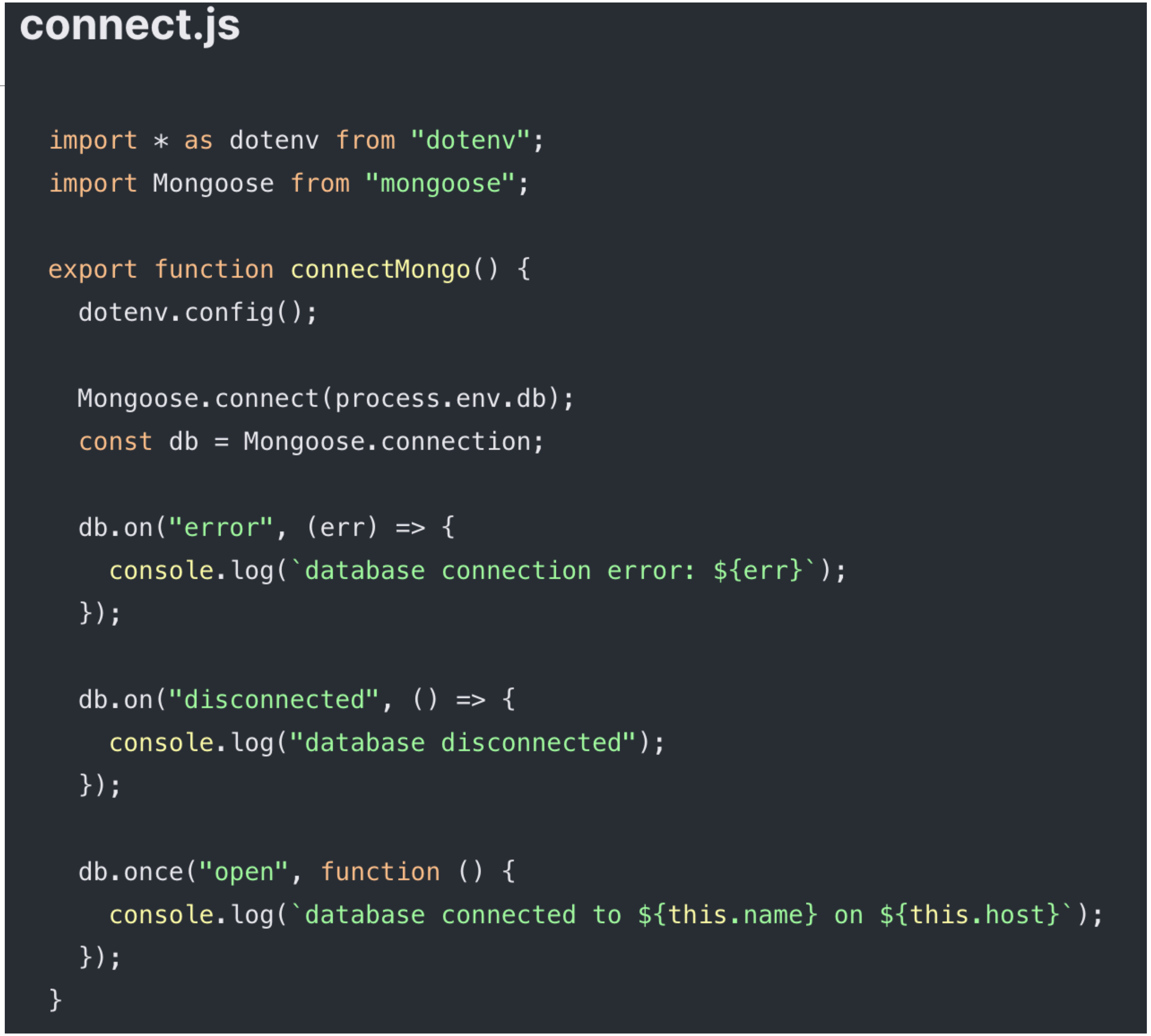
Connecting to Mongo (via Mongoose)

import .env

import & configure mongoose

connect to the database service

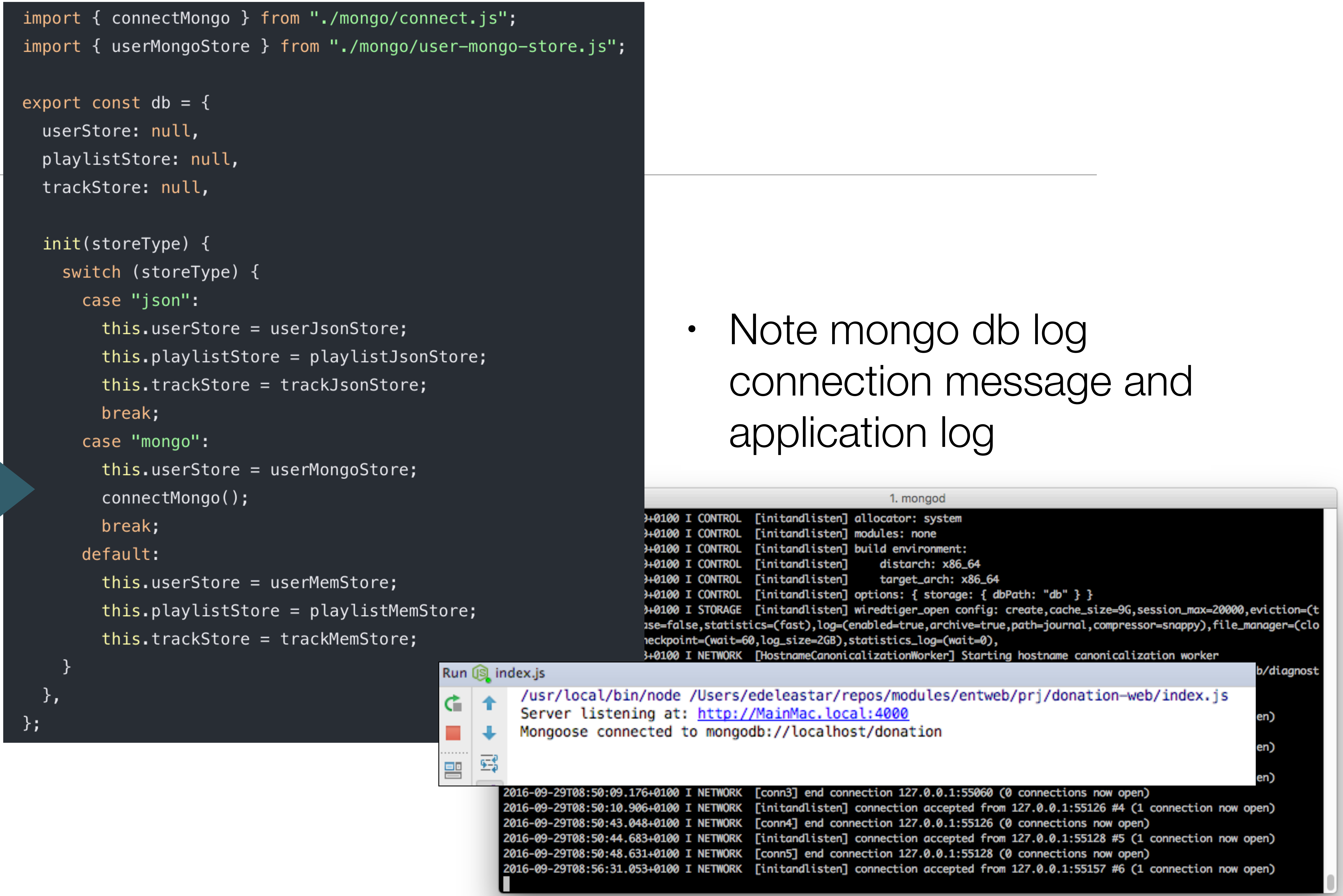
Log success/fail/ disconnect



connect.js

```
import * as dotenv from "dotenv";
import Mongoose from "mongoose";

export function connectMongo() {
  dotenv.config();

  Mongoose.connect(process.env.db);
  const db = Mongoose.connection;

  db.on("error", (err) => {
    console.log(`database connection error: ${err}`);
  });

  db.on("disconnected", () => {
    console.log("database disconnected");
  });

  db.once("open", function () {
    console.log(`database connected to ${this.name} on ${this.host}`);
  });
}
```

- Trigger connection if mongo selected:



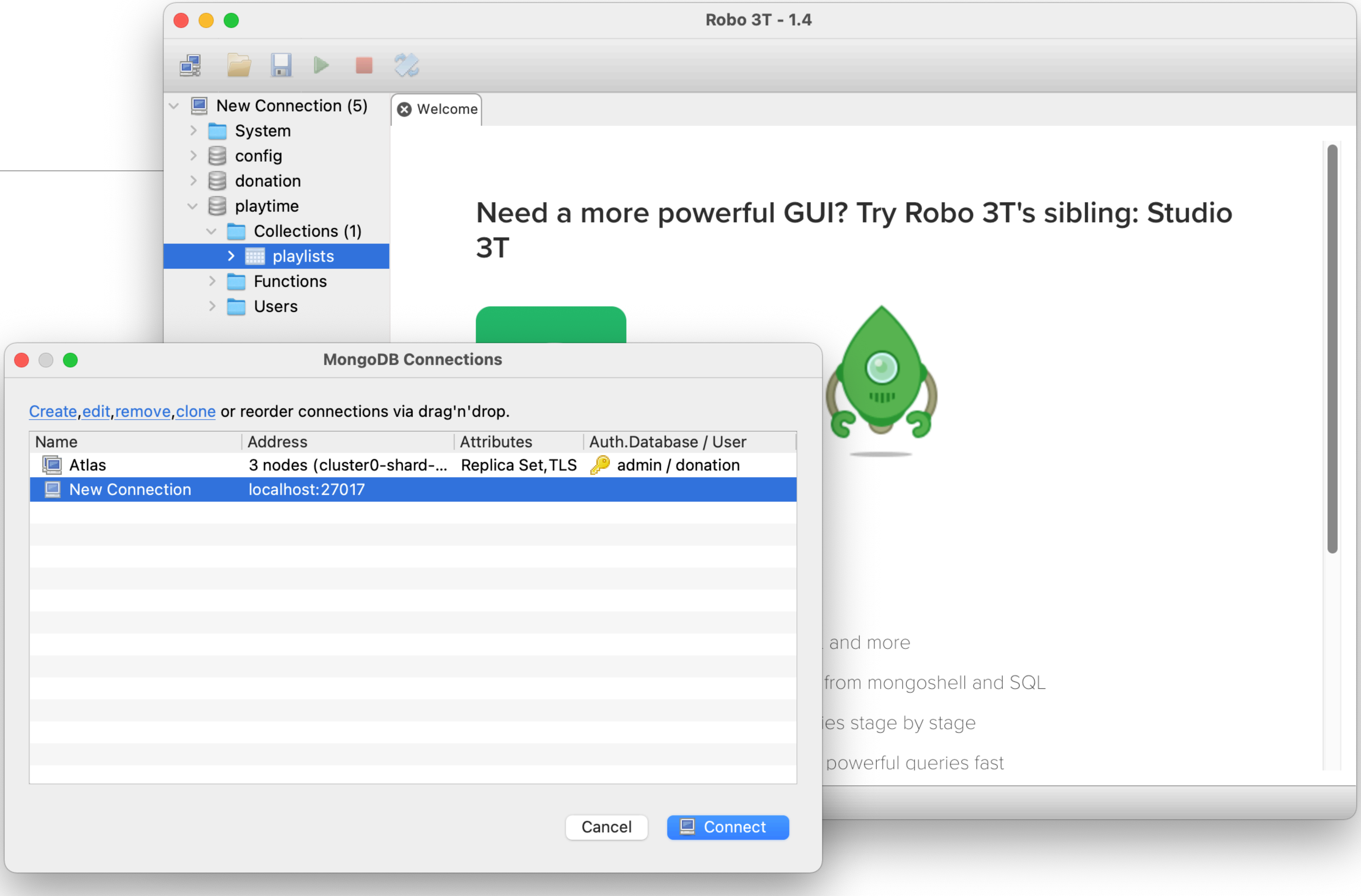
```
import { connectMongo } from "./mongo/connect.js";
import { userMongoStore } from "./mongo/user-mongo-store.js";

export const db = {
  userStore: null,
  playlistStore: null,
  trackStore: null,

  init(storeType) {
    switch (storeType) {
      case "json":
        this.userStore = userJsonStore;
        this.playlistStore = playlistJsonStore;
        this.trackStore = trackJsonStore;
        break;
      case "mongo":
        this.userStore = userMongoStore;
        connectMongo();
        break;
      default:
        this.userStore = userMemStore;
        this.playlistStore = playlistMemStore;
        this.trackStore = trackMemStore;
    }
  },
};
```

- Note mongo db log connection message and application log

```
)+0100 I CONTROL  [initandlisten] allocator: system
)+0100 I CONTROL  [initandlisten] modules: none
)+0100 I CONTROL  [initandlisten] build environment:
)+0100 I CONTROL  [initandlisten]     distarch: x86_64
)+0100 I CONTROL  [initandlisten]     target_arch: x86_64
)+0100 I CONTROL  [initandlisten] options: { storage: { dbPath: "db" } }
)+0100 I STORAGE  [initandlisten] wiredtiger_open config: create,cache_size=9G,session_max=20000,eviction=(t
use=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(clo
heckpoint=(wait=60,log_size=2GB),statistics_log=(wait=0),
3+0100 I NETWORK  [HostnameCanonicalizationWorker] Starting hostname canonicalization worker
                                                                                                  b/diagnost
                                                                                                  en)
                                                                                                  en)
                                                                                                  en)
2016-09-29T08:50:09.176+0100 I NETWORK  [conn3] end connection 127.0.0.1:55060 (0 connections now open)
2016-09-29T08:50:10.906+0100 I NETWORK  [initandlisten] connection accepted from 127.0.0.1:55126 #4 (1 connection now open)
2016-09-29T08:50:43.048+0100 I NETWORK  [conn4] end connection 127.0.0.1:55126 (0 connections now open)
2016-09-29T08:50:44.683+0100 I NETWORK  [initandlisten] connection accepted from 127.0.0.1:55128 #5 (1 connection now open)
2016-09-29T08:50:48.631+0100 I NETWORK  [conn5] end connection 127.0.0.1:55128 (0 connections now open)
2016-09-29T08:56:31.053+0100 I NETWORK  [initandlisten] connection accepted from 127.0.0.1:55157 #6 (1 connection now open)
```

```
/usr/local/bin/node /Users/edeleastar/repos/modules/entweb/prj/donation-web/index.js
Server listening at: http://MainMac.local:4000
Mongoose connected to mongodb://localhost/donation
```

- Connect Robo 3T to database to be able to browse contents



Need a more powerful GUI? Try Robo 3T's sibling: Studio 3T

MongoDB Connections

Create,edit,remove,clone or reorder connections via drag'n'drop.

Name	Address	Attributes	Auth.Database / User
Atlas	3 nodes (cluster0-shard-...	Replica Set,TLS	admin / donation
New Connection	localhost:27017		

Setting up Mongo

Full Stack Web Development