

# Setting up Mongo

---

# 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",  
  age: 26,  
  status: "A",  
  groups: [ "news", "sports" ]  
}
```

← field: value  
← field: value  
← field: value  
← field: value

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

# 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.

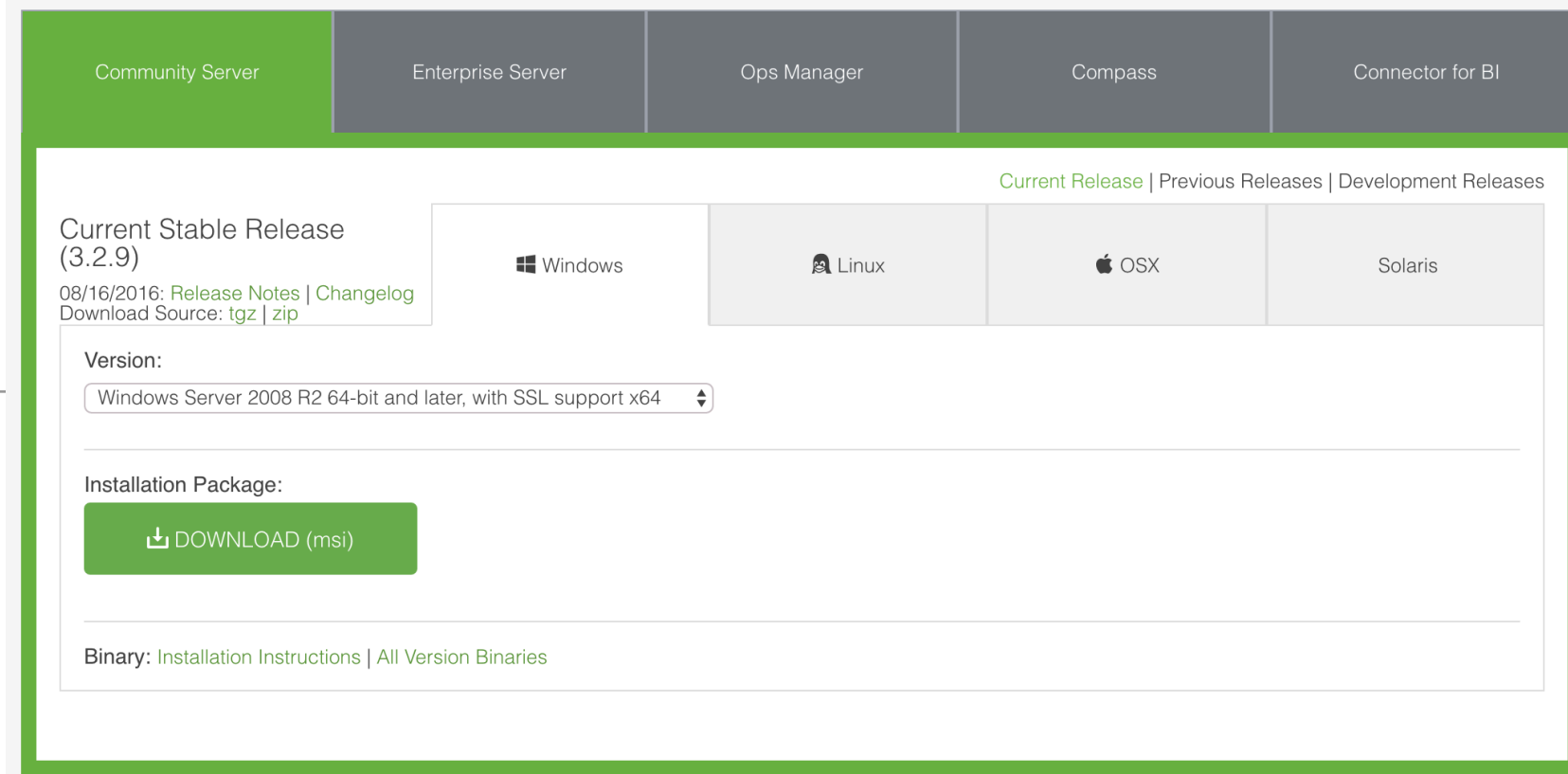
# Learning Mongo via Mongoose

---

- 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.

# Setup (1)

- Install MongoDB
- Launch Mongo



Typically, to launch the mongod 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

```
Command Prompt - mongod -dbpath db

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

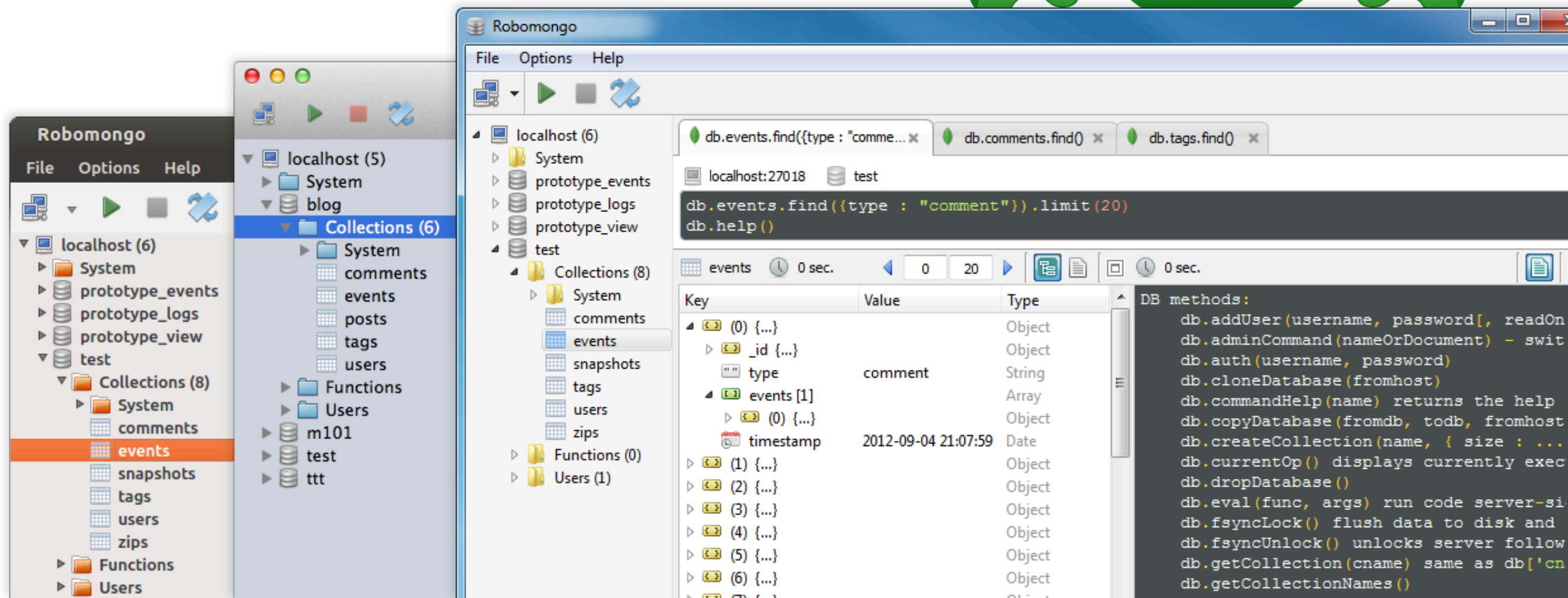
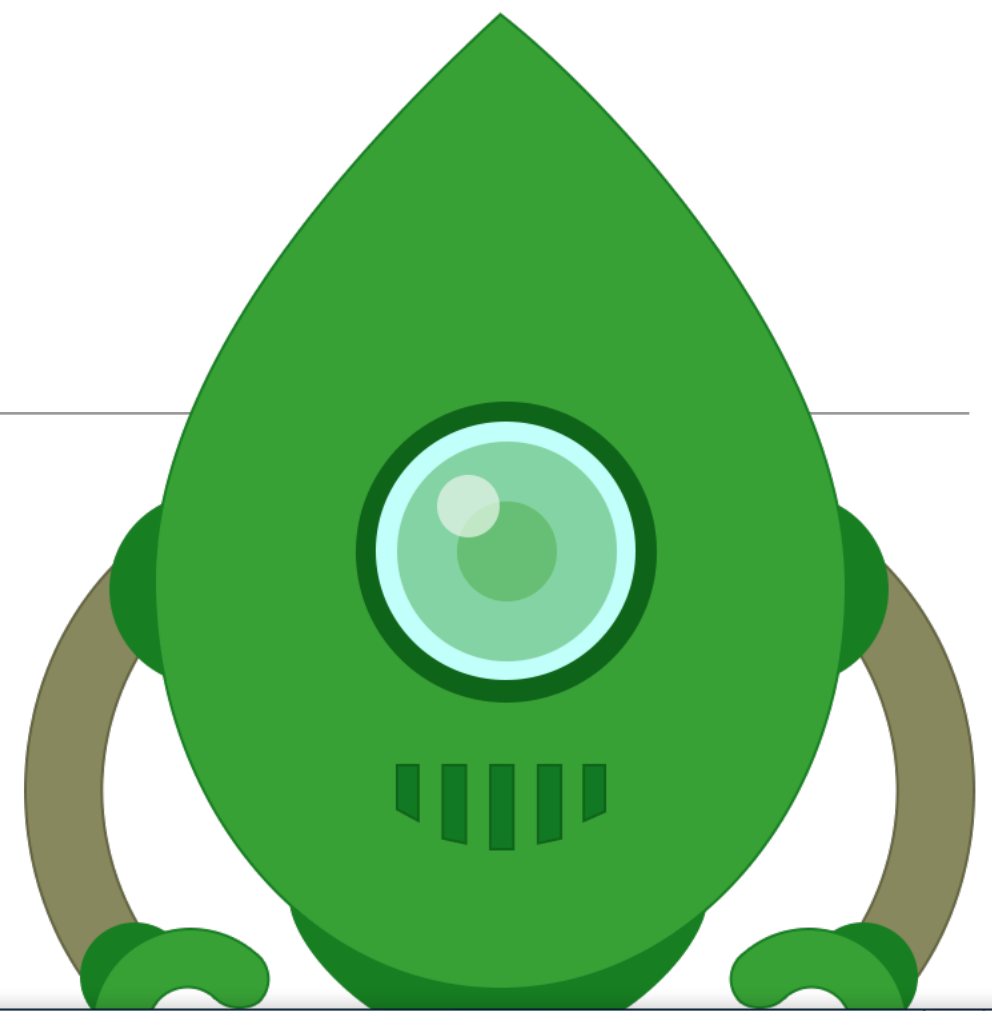
C:\dev>mkdir db

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=(threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_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 RoboMongo
- A visual tool 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

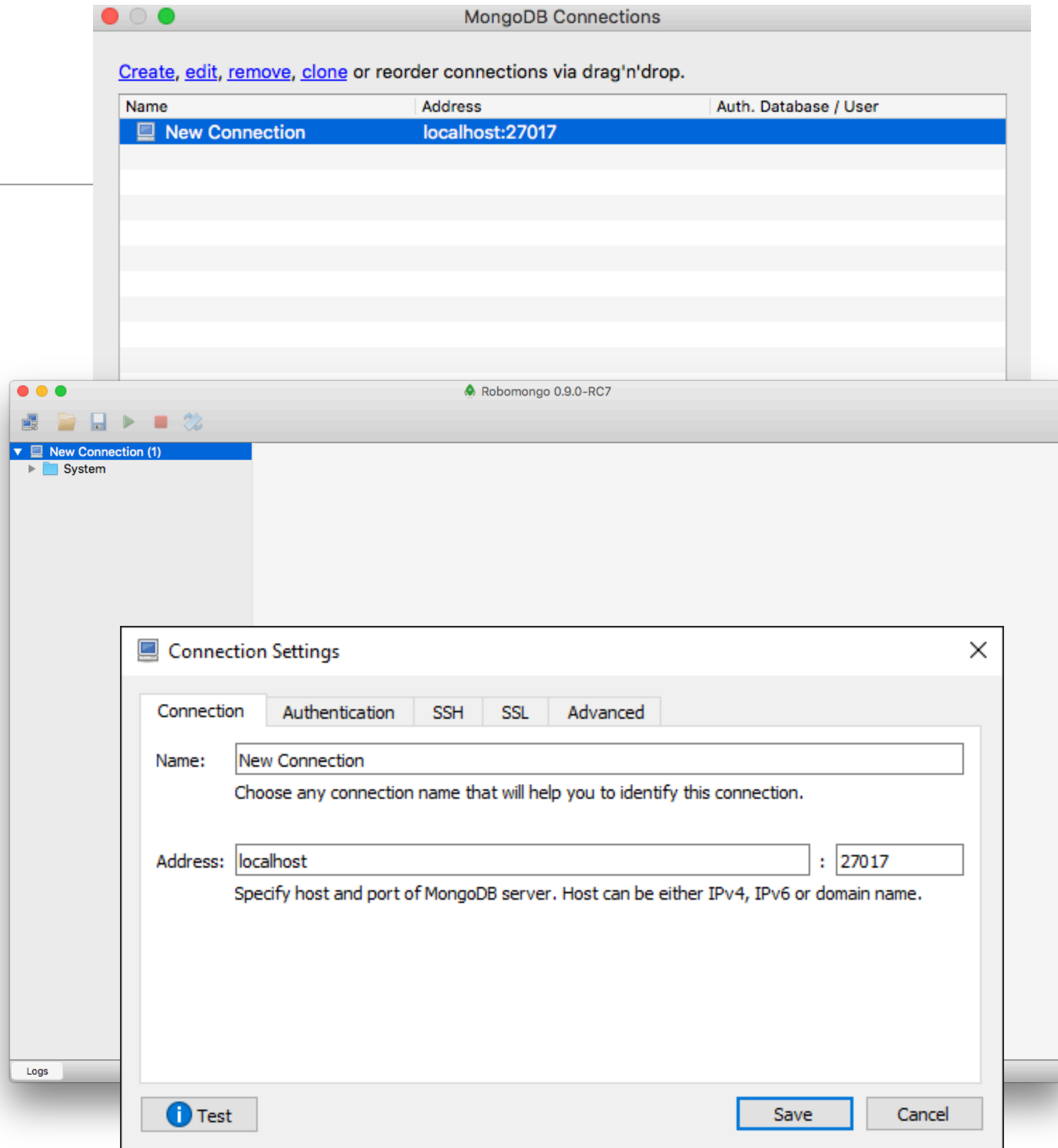
DB methods:

```
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 Robomongo and connect to the database you have already started
- Default may be localhost:27017



# Node & Mongoose


---

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

```
npm install mongoose -save
```

package.json

```
{
  "name": "donation-web",
  "version": "1.0.0",
  "description": "an application to host donations for candidates",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "handlebars": "^4.0.5",
    "hapi": "^14.1.0",
    "hapi-auth-cookie": "^6.1.1",
    "inert": "^4.0.1",
    "mongoose": "^4.5.8",
    "vision": "^4.1.0"
  }
}
```



# Connecting to Mongo (via Mongoose)

db.js

import mongoose

reset the 'promise' library

declare the connection string

connect to the database

Log success/fail/disconnect

```
'use strict';

const mongoose = require('mongoose');
mongoose.Promise = global.Promise;

let dbURI = 'mongodb://localhost/donation';
if (process.env.NODE_ENV === 'production') {
  dbURI = process.env.MONGODB_URI;
}

mongoose.connect(dbURI);

mongoose.connection.on('connected', function () {
  console.log('Mongoose connected to ' + dbURI);
});

mongoose.connection.on('error', function (err) {
  console.log('Mongoose connection error: ' + err);
});

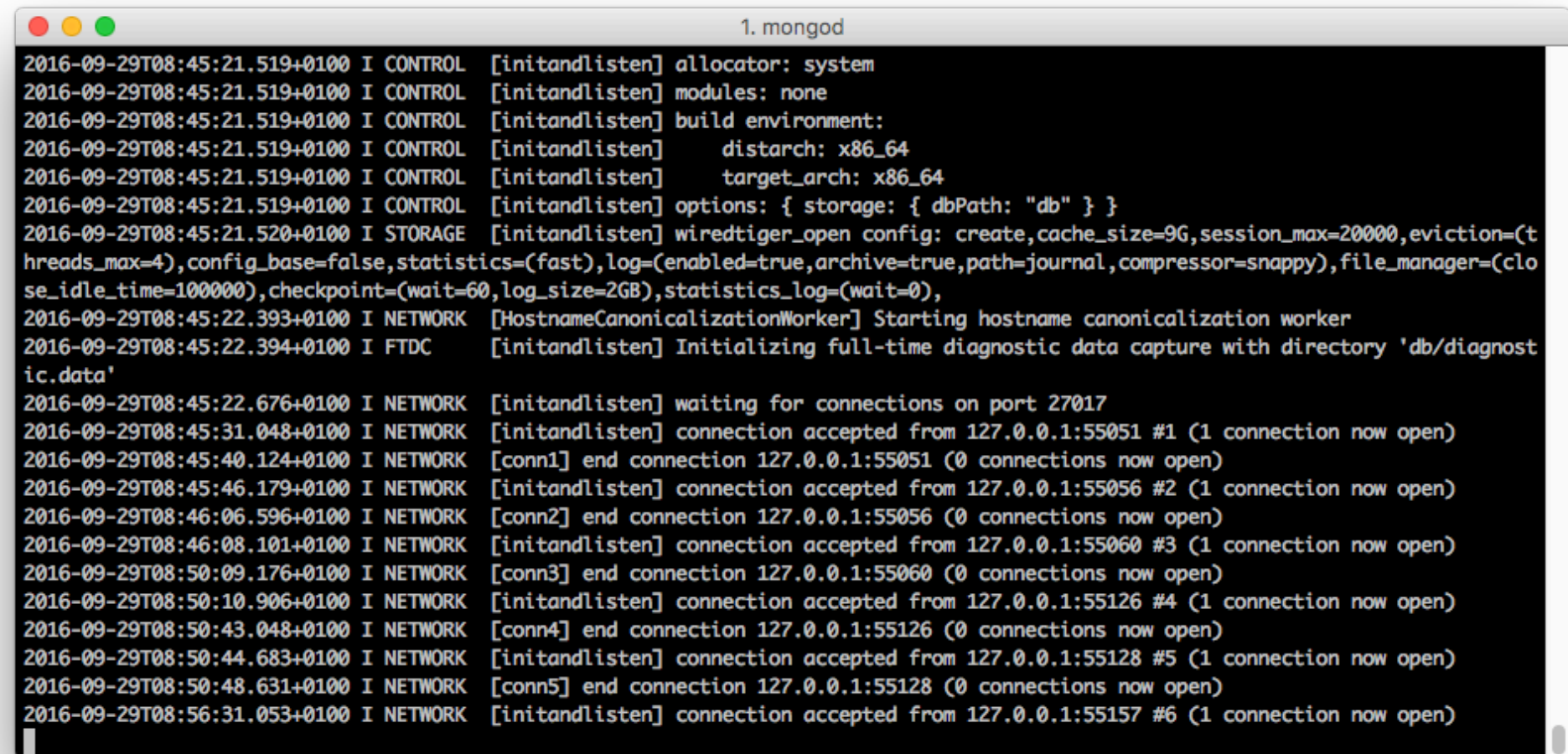
mongoose.connection.on('disconnected', function () {
  console.log('Mongoose disconnected');
});
```

- Trigger connection just by importing db.js from index

index.js

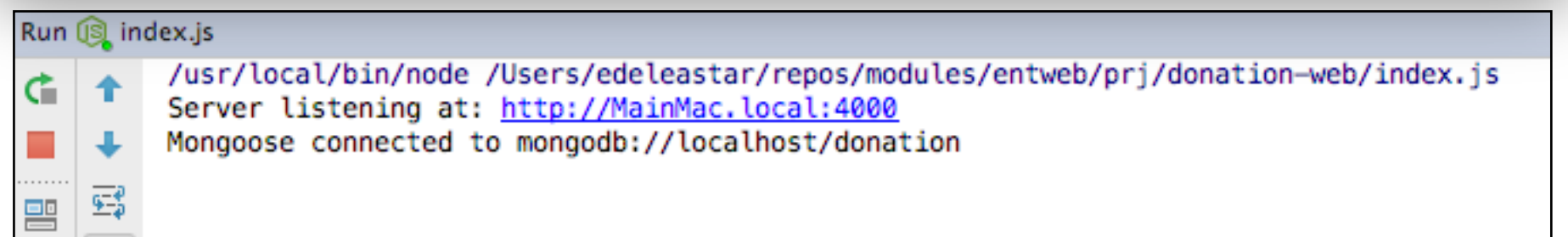
```
...  
require( './app/models/db' );  
...
```

- Note mongod log connection message



```
1. mongod  
2016-09-29T08:45:21.519+0100 I CONTROL [initandlisten] allocator: system  
2016-09-29T08:45:21.519+0100 I CONTROL [initandlisten] modules: none  
2016-09-29T08:45:21.519+0100 I CONTROL [initandlisten] build environment:  
2016-09-29T08:45:21.519+0100 I CONTROL [initandlisten] distarch: x86_64  
2016-09-29T08:45:21.519+0100 I CONTROL [initandlisten] target_arch: x86_64  
2016-09-29T08:45:21.519+0100 I CONTROL [initandlisten] options: { storage: { dbPath: "db" } }  
2016-09-29T08:45:21.520+0100 I STORAGE [initandlisten] wiredtiger_open config: create,cache_size=9G,session_max=20000,eviction=(t  
hreadsafe_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(clo  
se_idle_time=100000),checkpoint=(wait=60,log_size=2GB),statistics_log=(wait=0),  
2016-09-29T08:45:22.393+0100 I NETWORK [HostnameCanonicalizationWorker] Starting hostname canonicalization worker  
2016-09-29T08:45:22.394+0100 I FTDC [initandlisten] Initializing full-time diagnostic data capture with directory 'db/diagnost  
ic.data'  
2016-09-29T08:45:22.676+0100 I NETWORK [initandlisten] waiting for connections on port 27017  
2016-09-29T08:45:31.048+0100 I NETWORK [initandlisten] connection accepted from 127.0.0.1:55051 #1 (1 connection now open)  
2016-09-29T08:45:40.124+0100 I NETWORK [conn1] end connection 127.0.0.1:55051 (0 connections now open)  
2016-09-29T08:45:46.179+0100 I NETWORK [initandlisten] connection accepted from 127.0.0.1:55056 #2 (1 connection now open)  
2016-09-29T08:46:06.596+0100 I NETWORK [conn2] end connection 127.0.0.1:55056 (0 connections now open)  
2016-09-29T08:46:08.101+0100 I NETWORK [initandlisten] connection accepted from 127.0.0.1:55060 #3 (1 connection now open)  
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)
```

- ..and application log



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