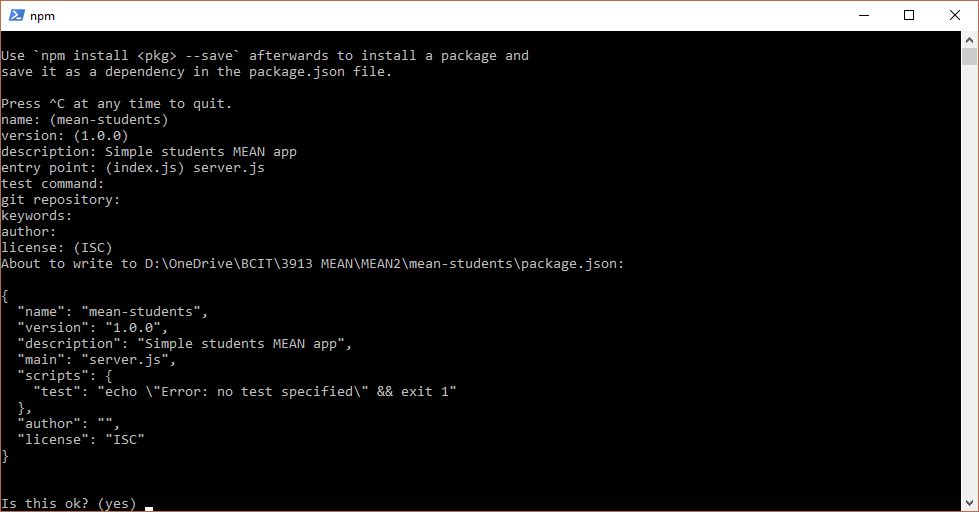
### Setting up our node express app

mkdir mean-bcit

cd mean-bcit

npm init



npm install express body-parser ejs ejs-locals mongojs --save

Create *server.js* in the app root

var express = require("express");

var path = require("path");

var bodyParser = require("body-parser");

var index = require("./routes/index");

var students = require("./routes/students");

var port = 3000;

var app = express();

// View engine

var ejsEngine = require("ejs-locals");

app.engine("ejs", ejsEngine); // support master pages

app.set("view engine", "ejs"); // ejs view engine

// Set static folder

app.use(express.static(path.join(\_\_dirname, "client")));

// Body parser middleware

app.use(bodyParser.json());

app.use(bodyParser.urlencoded({extended: false}));

app.use("/", index);

app.use("/api", students);

app.listen(port, function() {

console.log("Server started on port " + port)

});

Create folder */routes*.

Create files */routes/index.js* and *routes/students.js*

***index.js***

var express = require("express");

var router = express.Router();

router.get("/", function(req, res, next) {

res.send("INDEX page");

});

module.exports = router;

***students.js***

var express = require("express");

var router = express.Router();

router.get("/students", function(req, res, next) {

res.send("STUDENTS API");

});

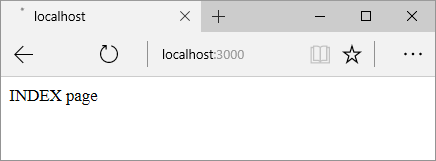
module.exports = router;

Create folder */views*.

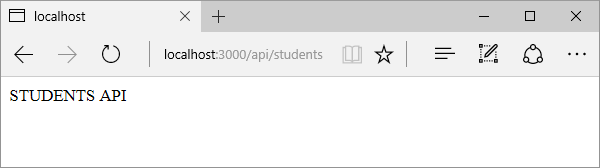
Create empty file /views/index.ejs

node server.js

<http://localhost:3000>



<http://localhost:3000/api/students>



In */routes/index.js* change

res.send("INDEX page");

to

res.render("index", { title: "MEAN BCIT" });

***/views/layout.ejs***

<!doctype html>

<html>

<head><title><%- title %></title></head>

<body>

<div><%- body %></div>

</body>

VS-Code, hit ! followed by tab to scaffold an HTML page

</html>

***/views/index.ejs***

<%- layout('layout.ejs') %>

<h1><%- title %></h1>

<p>This is from: <%- title %></p>

Restart the server then go to <http://localhost:3000>

### nodemon



It is a pain to reload the server. It is suggested to use nodemon instead.

npm install -g nodemon

Instead of *node server.js*, you simply run *nodemon*.

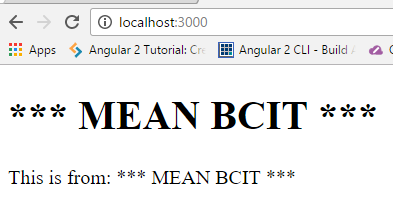
In /*routes/index. js* change:

title: "MEAN BCIT"

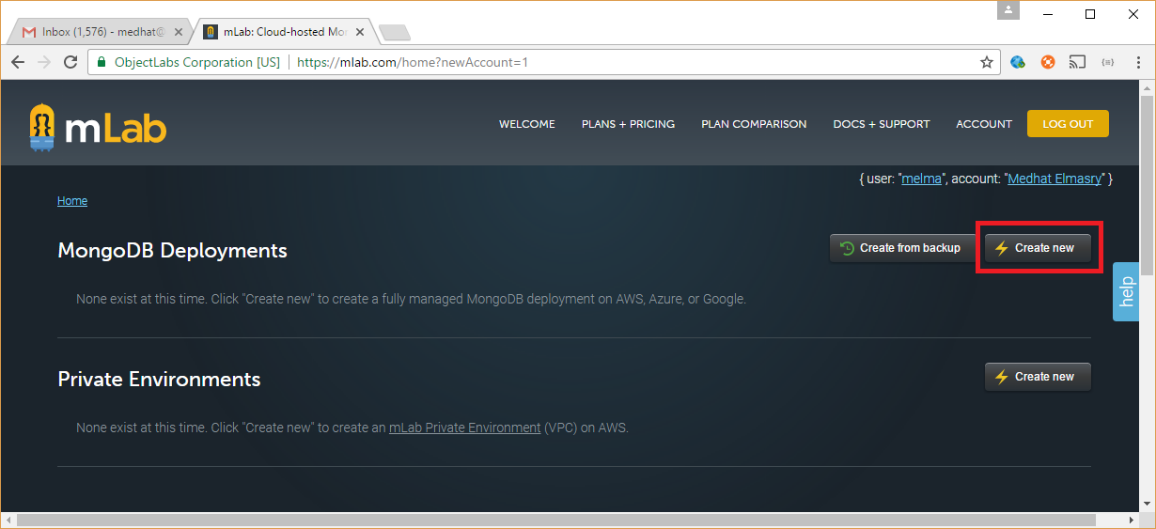
To

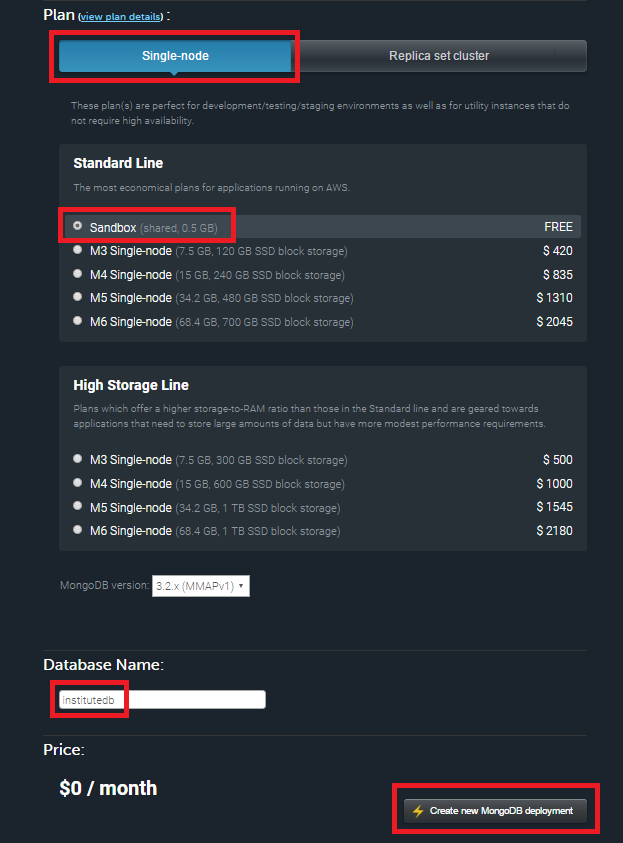
title: "\*\*\* MEAN BCIT \*\*\*"

Refresh page and the browser will update:

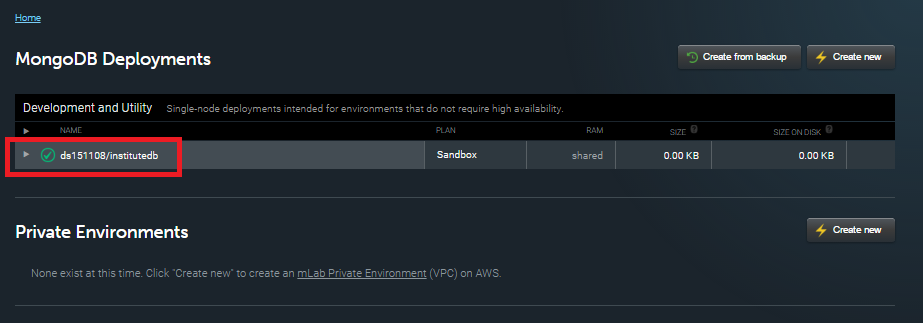


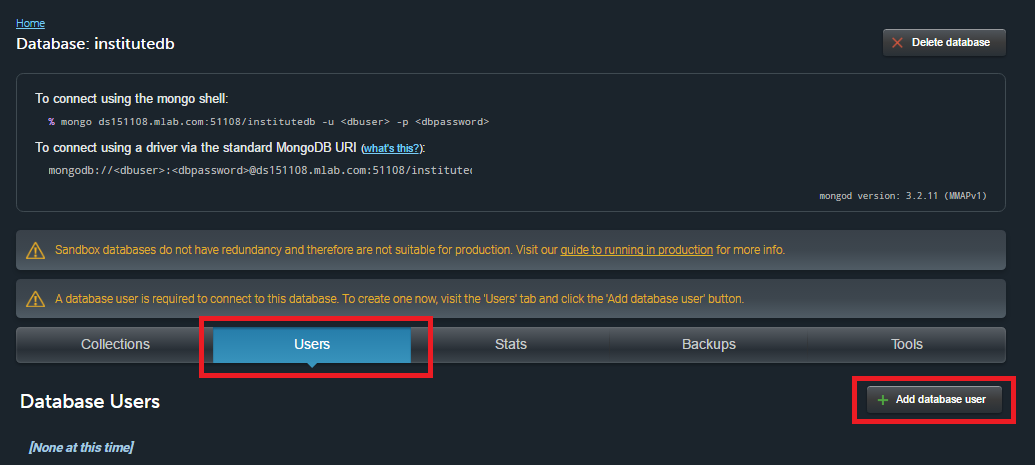
### mLab

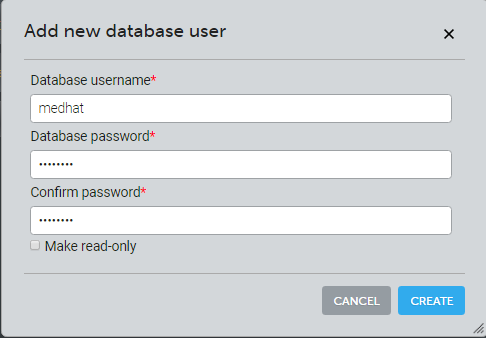


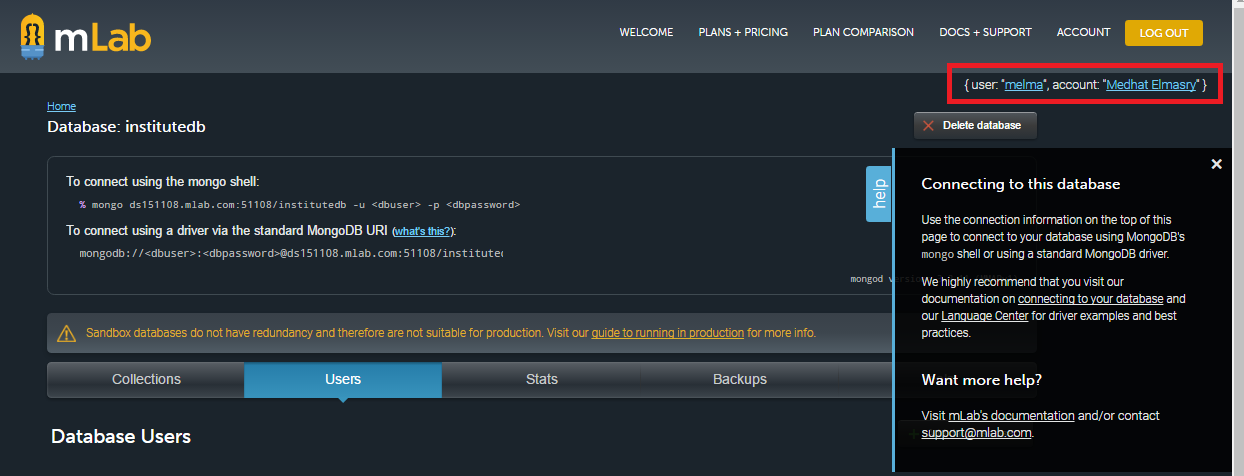


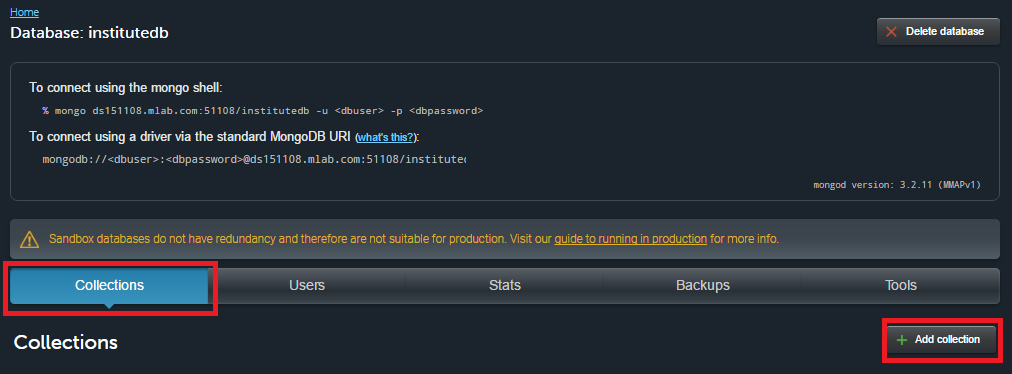
Then click on the “Create new MongoDB deployment” button.

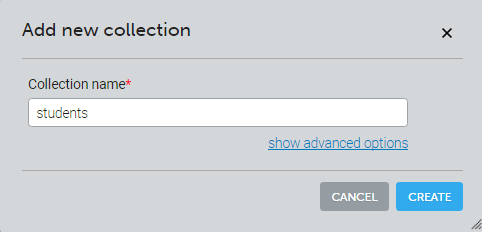


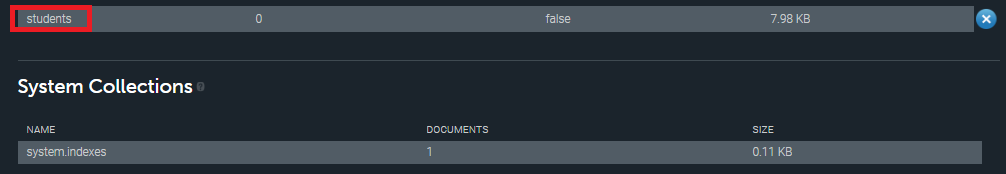


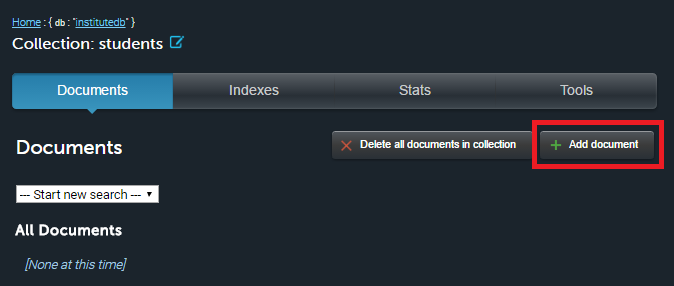












Add the following data one student at a time:

[

{

"FirstName":"Bob",

"LastName":"Doe",

"School":"Engineering",

"StartDate": {"$date": "2015-09-30T08:30:00.000Z"}

}, {

"FirstName":"Ann",

"LastName":"Lee",

"School":"Medicine",

"StartDate": {"$date": "2014-04-15T07:32:00.000Z"}

}, {

"FirstName":"Sue",

"LastName":"Douglas",

"School":"Pharmacy",

"StartDate": {"$date": "2016-01-17T18:30:00.000Z"}

}, {

"FirstName":"Tom",

"LastName":"Brown",

"School":"Business",

"StartDate": {"$date": "2015-07-26T11:22:00.000Z"}

}, {

"FirstName":"Joe",

"LastName":"Mason",

"School":"Health",

"StartDate": {"$date": "2013-08-01T21:30:00.000Z"}

}

]

### Insert Data using mongo client application

Alternatively, you can add data from the mongo.exe client on your computer

./mongo.exe --nodb

// Because we are going to be using a remote connection, be sure

// to start the MongoDB Shell (mongo) with the --nodb flag. Then,

// we can connect and define our own db instance.

// Connect to the MongoLab database.

var connection = new Mongo( "ds151108.mlab.com:51108" );

// Connect to the test database.

var db = connection.getDB( "institutedb" );

// Authorize this connection.

db.auth( "username", "password" );

db.students.insert(

[

{

"FirstName":"Tim",

"LastName":"Burke",

"School":"Tourism",

"StartDate": new Date("2010-02-20T08:30:00")

},{

"FirstName":"Lyn",

"LastName":"Petico",

"School":"Business",

"StartDate": new Date("2016-04-17T17:32:00")

},{

"FirstName":"Harry",

"LastName":"Wilkens",

"School":"Economics",

"StartDate": new Date("2010-02-20T08:30:00")

},{

"FirstName":"Judy",

"LastName":"Roy",

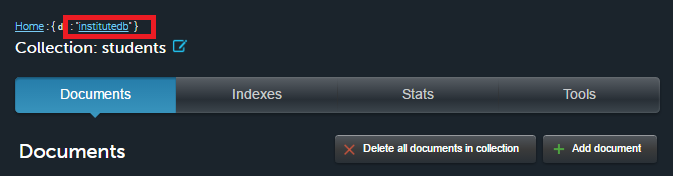
"School":"Marketing",

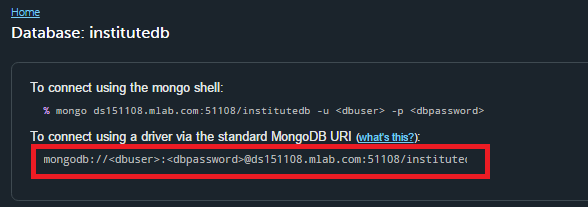
"StartDate": new Date("2014-05-16T17:32:00")

}

]);

### Use mongojs node.js driver to interact with our database





This is the syntax for connecting to the MongoDB database:

var mongojs = require("mongojs");

var db = mongojs("mongodb://<dbuser>:<dbpassword>@ds151108.mlab.com:51108/institutedb");

Replace *<dbuser>* with username and *<dbpassword>* with password. Make sure that this is the user & password for the database and not the admin username and password.

Return to *routes/students.js*. Add:

var mongojs = require("mongojs");

var db = mongojs("mongodb://username:password@ds151108.mlab.com:51108/institutedb", ['students']);

// to connect to local mongodb server

var db = mongojs("institutedb", ["students"]);

Replace:

res.send("STUDENTS API");

With:

db.students.find(function(err, data){

if (err) {

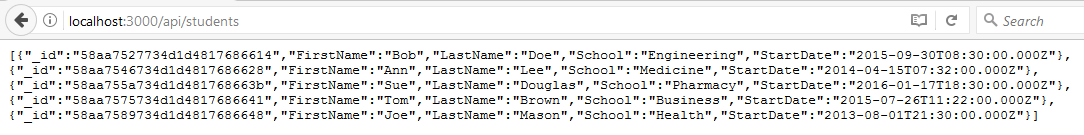
res.send(err)

}

res.json(data);

});

<http://localhost:3000/api/students>



In */routes/students.js* add the following route:

// get single student

router.get("/student/:id", function(req, res, next) {

db.students.findOne({\_id: mongojs.ObjectId(req.params.id)},function(err, data){

if (err) {

res.send(err);

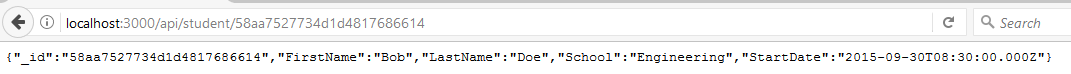
}

res.json(data);

});

});

Get the value of *\_id* and try such a URL: http://localhost:3000/api/student/58aa7527734d1d4817686614



Add these post, delete, and put routes:

// create student

router.post("/student", function(req, res, next) {

var student = req.body;

if (!student.StartDate) {

student.StartDate = new Date();

}

if (!student.FirstName || !student.LastName

|| !student.School) {

res.status(400);

res.json(

{"error": "Bad data, could not be inserted into the database."}

)

} else {

db.students.save(student, function(err, data) {

if (err) {

res.send(err);

}

res.json(data);

})

}

});

// delete student

router.delete("/student/:id", function(req, res, next) {

db.students.remove({\_id: mongojs.ObjectId(req.params.id)},function(err, data){

if (err) {

res.send(err);

}

res.json(data);

});

});

// update student

router.put("/student/:id", function(req, res, next) {

var student = req.body;

var changedStudent = {};

if (student.FirstName) {

changedStudent.FirstName = student.FirstName;

}

if (student.LastName) {

changedStudent.LastName = student.LastName;

}

if (student.School) {

changedStudent.School = student.School;

}

if (student.StartDate) {

changedStudent.StartDate = student.StartDate;

}

if (!changedStudent) {

res.status(400);

res.json(

{"error": "Bad Data"}

)

} else {

db.students.update({\_id: mongojs.ObjectId(req.params.id)}, changedStudent,{},function(err, data){

if (err) {

res.send(err);

}

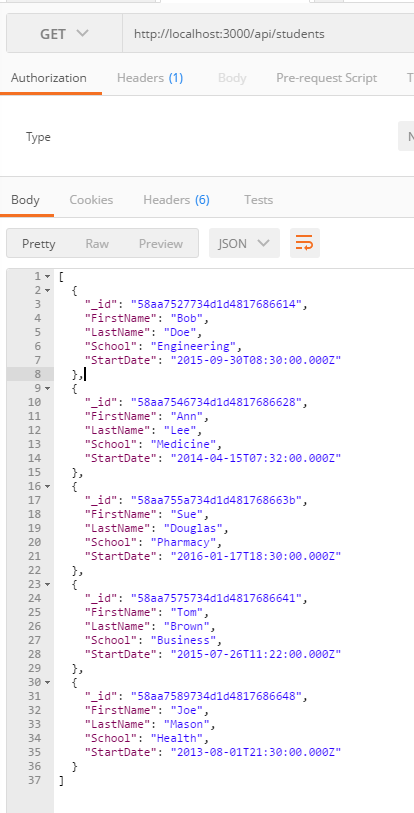
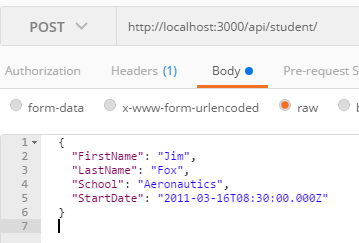
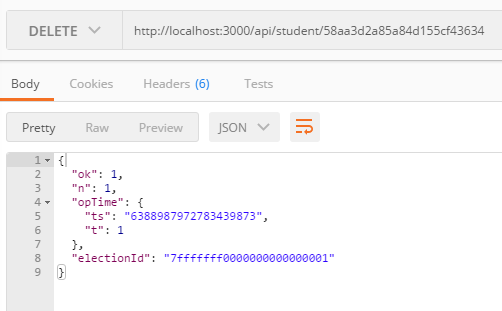
res.json(data);

});

}

});

### Test with Postman:



Content-Type: application/json

# 

Content-Type: application/json