



LẬP TRÌNH SERVER CHO ANDROID

BÀI 7: MONGOOSE

- ① Cài đặt và cấu hình Mongoose
- ① Thao tác MongoDB server với NodeJS qua Mongoose



Phần I: Cài đặt và cấu hình Mongoose

 Tổng quan Mongoose

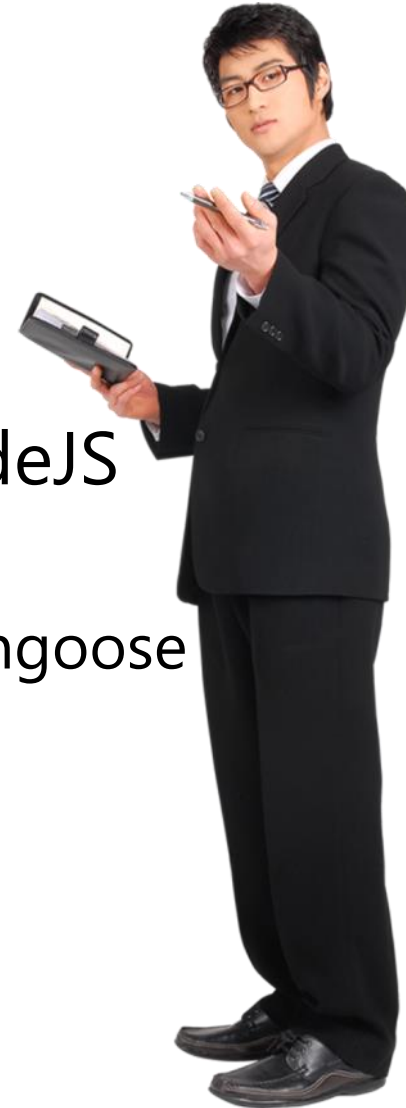
 Cài đặt Mongoose

 Cấu hình Mongoose

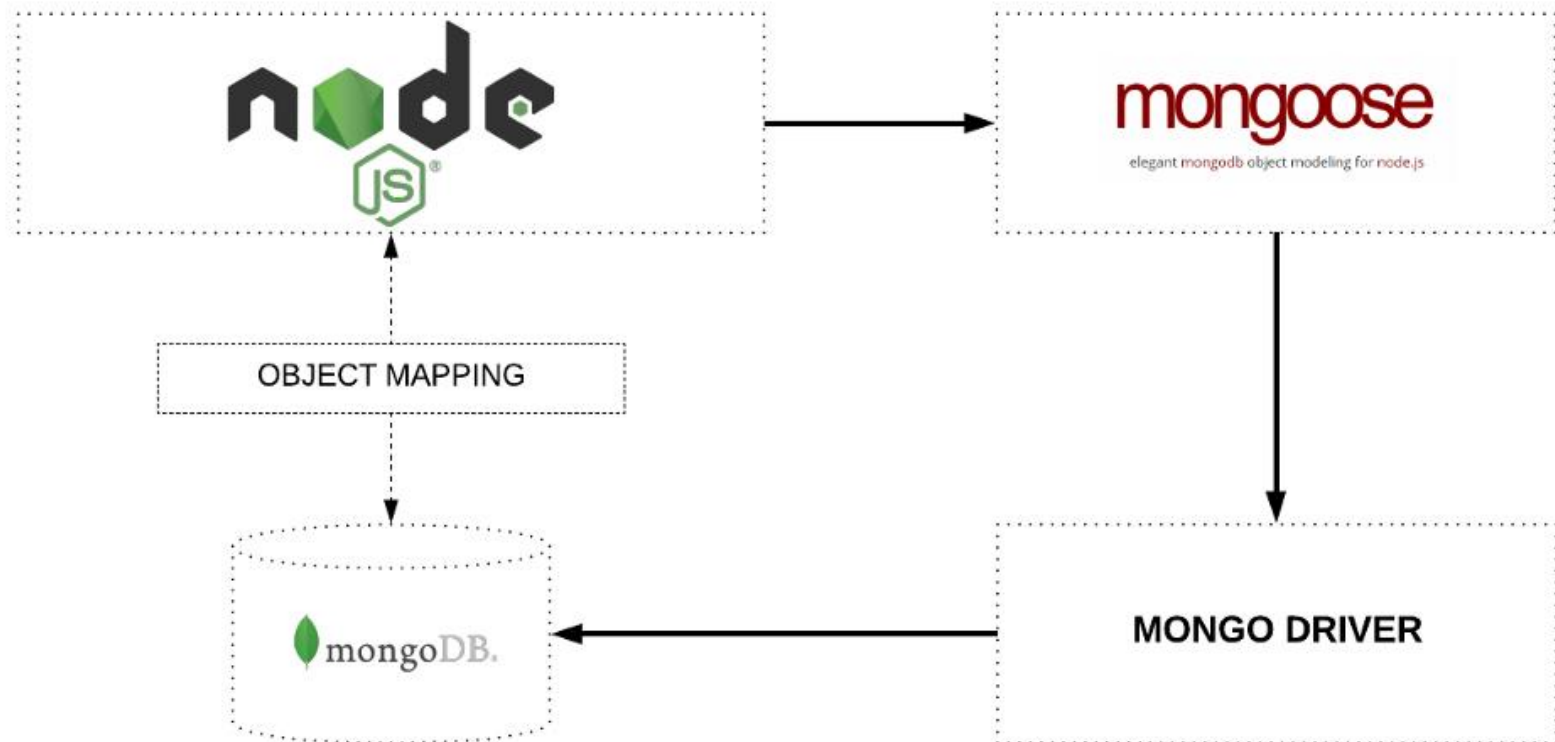
Phần II: Thao tác MongoDB server với NodeJS qua Mongoose

 Kết nối MongoDB server với NodeJS qua Mongoose

 Thao tác dữ liệu



- ❑ Mongoose là một thư viện mô hình hóa đối tượng(Object Data Model - ODM) cho MongoDB và NodeJS.



- ❑ Mongoose cho phép định nghĩa các object (đối tượng) với một schema được định nghĩa rõ ràng, được ánh xạ tới một MongoDB document.
- ❑ Người ta thường dùng thư viện Mongoose để kết nối MongoDB và NodeJS

```
var mongoose = require('mongoose');

module.exports = new mongoose.Schema({
  name: {
    type: String,
    required: true
  },
  email: {
    type: String,
    required: true,
    match: /.+@.+\.+$/,
    lowercase: true
  },
  loggedInCount: {
    type: Number,
    default: 0
  }
});
```

matches X@Y.Z

- ❑ Mongoose cung cấp các chức năng cho việc tạo ra và làm việc với các Schema.
- ❑ Mongoose hiện có 8 SchemaTypes.

1. String
2. Number
3. Date
4. Buffer
5. Boolean
6. Mixed
7. ObjectId
8. Array

```
var PostSchema = new Schema({  
  _id: ObjectId, // implicitly exists  
  title: { type: String, required: true },  
  body: { type: String, required: true },  
  author: { type: ObjectId, required: true, ref: 'User' },  
  tags: [String],  
  date: { type: Date, default: Date.now },  
  is_featured: { type: Boolean, default: false }  
});
```

- ❑ String: Cho phép chỉ định các tùy chọn bổ sung sau:
 - ❖ Chuyển đổi thành chữ thường
 - ❖ Chuyển đổi thành chữ hoa
 - ❖ Cắt dữ liệu trước khi lưu lại
 - ❖ Một biểu thức chính quy có thể giới hạn dữ liệu được phép lưu trong quá trình xác thực
 - ❖ Một enum có thể định nghĩa một danh sách các String(chuỗi) hợp lệ
- ❑ Number và Date: Luôn hỗ trợ chỉ định một giá trị tối thiểu và tối đa cho phép của field đó.
- ❑ Buffer: Cho phép bạn lưu dữ liệu nhị phân

- ❑ Mixed: Biến một thuộc tính thành một field.
- ❑ ObjectId: Thường chỉ định một liên kết đến một tài liệu khác trong Database
- ❑ Array: Cho phép lưu trữ các mảng giống như Javascript

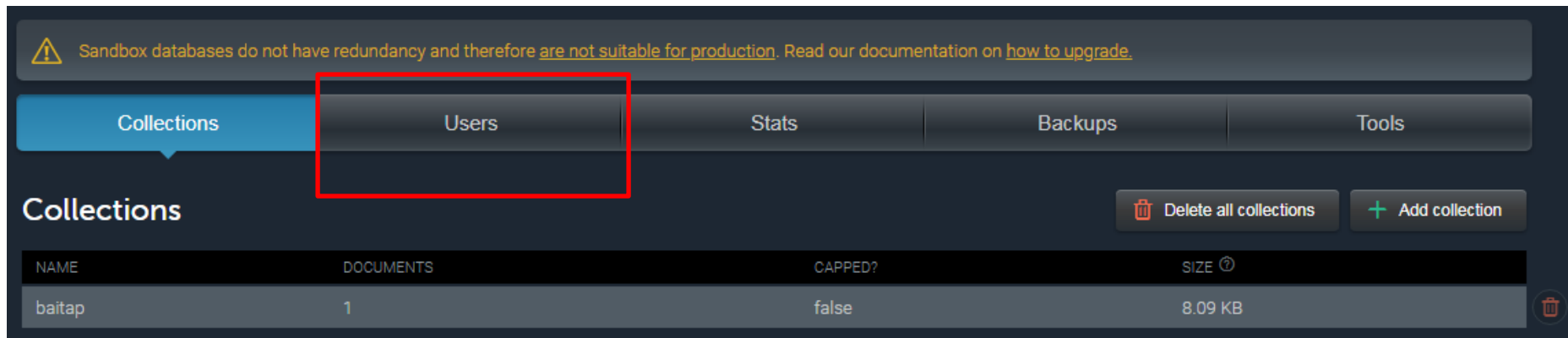
- ❑ Để cài đặt mongoose, các bạn gõ lệnh:
`npm install mongoose --save`

```
Administrator@YZFAXCMZING35CL MINGW64 /e/Hoc Tap/NodeJS/slide1
$ npm install mongoose --save
npm WARN demo@1.0.0 No repository field.

+ mongoose@5.3.0
added 24 packages in 8.772s

Administrator@YZFAXCMZING35CL MINGW64 /e/Hoc Tap/NodeJS/slide1
$ |
```

- ❑ Các bạn vào liên kết sau đăng nhập để tiến hành tạo kết nối trên mlab: <https://mlab.com/home>
- ❑ Tại tab Users các bạn click vào như hình dưới



- ❑ Để lấy được username và password cần vào tab Users tiến hành tạo username và password
- ❑ Lưu ý: Nhớ username và password để thêm vào thư viện mongoose

Add new database user

Database username*

baitapmongo

Database password*

.....

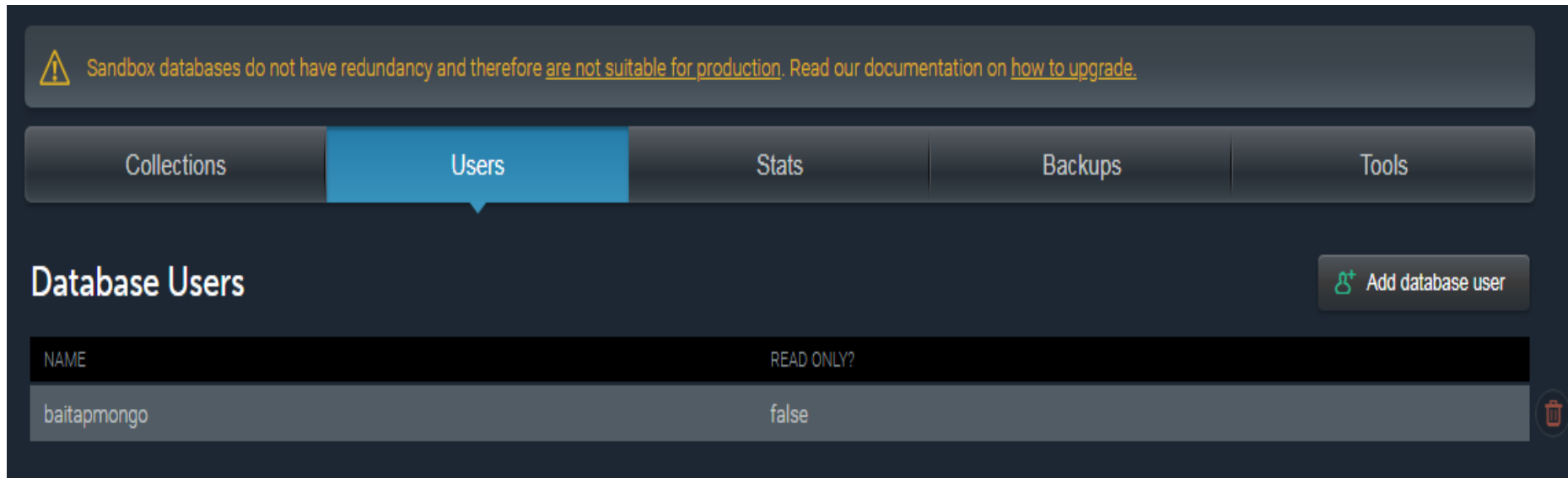
Confirm password*

.....

☐ Make read-only

CANCEL CREATE

- ❑ Sau khi các bạn tạo xong sẽ xuất hiện giao diện như hình dưới



The screenshot shows the MongoDB Compass web interface. At the top, there is a warning message: "Sandbox databases do not have redundancy and therefore are not suitable for production. Read our documentation on [how to upgrade](#)." Below this, there are five tabs: "Collections", "Users", "Stats", "Backups", and "Tools". The "Users" tab is currently selected. The main area is titled "Database Users" and contains a table with two columns: "NAME" and "READ ONLY?". The table has one row with the username "baitapmongo" and the value "false". In the top right corner of the main area, there is a button labeled "Add database user".

NAME	READ ONLY?
baitapmongo	false

- ❑ Lưu ý: Các bạn sẽ sử dụng username và password này trong suốt khóa học.

- ❑ Sau khi tạo xong Username và Password, các bạn tiến hành lấy MongoDB URI.
- ❑ MongoDB URI: sẽ giúp thư viện Mongoose kết nối với MongoDB server cho NodeJS
- ❑ Các bạn vào lại trang quản lí collection như hình dưới:

To connect using the mongo shell:

```
% mongo ds115523.mlab.com:15523/baitap -u <dbuser> -p <dbpassword>
```

To connect using a driver via the standard MongoDB URI ([what's this?](#)):

```
mongodb://<dbuser>:<dbpassword>@ds115523.mlab.com:15523/baitap
```

mongod version: 3.6.8 (MMAPv1)



Sandbox databases do not have redundancy and therefore are not suitable for production. Read our documentation on how to upgrade.

Collections

Users

Stats

Backups

Tools

- ❑ dbuser: là Username các bạn tạo bước trước trong Users
- ❑ dbpassword: là Password các bạn tạo bước trước trong Users

To connect using the mongo shell:

```
% mongo ds115523.mlab.com:15523/baitap -u <dbuser> -p <dbpassword>
```

To connect using a driver via the standard MongoDB URI ([what's this?](#)):

```
mongodb://<dbuser>:<dbpassword>@ds115523.mlab.com:15523/baitap
```

mongod version: 3.6.8 (MMAPv1)



Sandbox databases do not have redundancy and therefore are not suitable for production. Read our documentation on how to upgrade.

Collections

Users

Stats

Backups

Tools

- ❑ Vào File chứa Server của các bạn(index.js) tiến hành thêm dòng lệnh như bên dưới

```
//Gọi EXPRESSJS
var express = require('express');
//Tạo app để cấu hình router,...
var app = express();
//Chạy lên local host với port 3000
app.listen(process.env.PORT || '3232');
```

```
//Gọi thư viện Mongoose
var db = require('mongoose');
var Schema = db.Schema;

//Kết nối đến mlab. Các bạn nhớ thêm tài khoản và mật khẩu lúc này tạo ở Users
db.connect('mongodb://baitapmongo:A123456789@ds115523.mlab.com:15523/baitap');
console.log('Kết nối MongoDB thành công');
```

```
//Cấu hình handlebars
var expressHbs = require('express-handlebars');
app.engine('.hbs', expressHbs());
//Lưu Template vào folder "views"
app.set('view engine', '.hbs');
```

- ❑ Lưu ý: các bạn nhớ bắt buộc phải thêm username và password lúc này tạo ở Users

```
//Gọi thư viện Mongoose
var db = require('mongoose');
var Schema = db.Schema;

//Kết nối đến mlab. Các bạn nhớ thêm tài khoản và mật khẩu lúc này tạo ở Users
db.connect('mongodb://baitapmongo:A123456789@ds115523.mlab.com:15523/baitap');
console.log('Kết nối MongoDB thành công');
```


- ❑ Các bạn vào terminal và gõ dòng lệnh sau để tiến hành bật server: `nodemon index.js`

```
Administrator@YZFAXCMZING35CL MINGW64 /e/Hoc Tap/NodeJS/slide1
$ nodemon index.js
[nodemon] 1.17.5
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node index.js`
Kết nối MongoDB thành công
(node:5424) DeprecationWarning: current URL string parser is deprecated, and will
be removed in a future version. To use the new parser, pass option { useNewUrlParser: true } to MongoClient.connect.
```

- Như vậy là các bạn đã kết nối tới MongoDB thành công



DEMO

- Cài đặt, cấu hình mongoose



- ❑ Sử dụng Model để thêm các mô hình.
- ❑ Thêm dòng lệnh sau để tiến hành thêm collection và document cho MongoDB

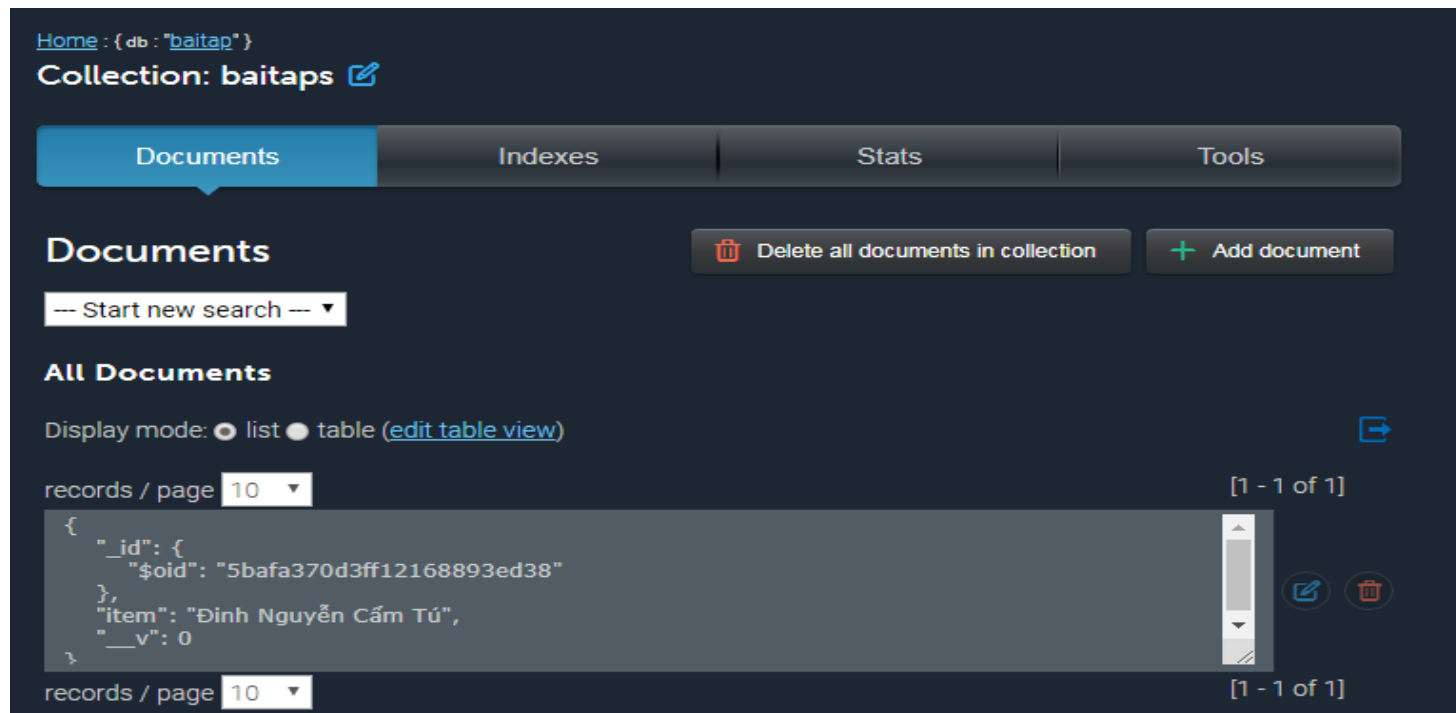
```
//Các bạn thêm các trường vào đây nhé
//Ở đây chỉ thêm một trường là item
var schema = new Schema({
  item: String,
});
//Kết nối tới bảng bài tập mới tạo
var thucHien = db.model('baitap', schema);
//Thêm vào Database
var item = thucHien({item: 'Đinh Nguyễn Cẩm Tú'}).save(function(err) {
  //Nếu bị lỗi sẽ bỏ qua không thông báo
  if(err) throw err;
  //Nếu đúng sẽ thông báo đã lưu
  console.log('Đã lưu');
});
```

... 4 ...

- ❑ Bây giờ tiến hành mở terminal lên chạy:
nodemon index.js

```
[nodemon] restarting due to changes...  
[nodemon] starting 'node index.js'  
Kết nối MongoDB thành công  
(node:5768) DeprecationWarning: current URL string parser is deprecated, and will  
be removed in a future version. To use the new parser, pass option { useNewUrlParser:  
Parser: true } to MongoClient.connect.  
Đã lưu
```

- ❑ Như vậy là xong các bạn có thể bật trang chủ mlab lên xem thử
- ❑ Thư viện Mongoose sẽ tự động thêm trường



UPDATE DOCUMENT TRONG MONGODB THÔNG QUA MONGOOSE

- ❑ Trước khi thực hiện edit các bạn lấy 'id' của nội dung document các bạn muốn sửa như hình dưới

All Documents

Display mode: ☒ list ☐ table ([edit table view](#))

records / page 10 [1 - 1 of 1]

```
{
  "_id": {
    "$oid": "5bb318a4a248690fd058717e"
  },
  "item": "Đinh Nguyễn Cẩm Tú",
  "__v": 0
}
```

records / page 10 [1 - 1 of 1]

UPDATE DOCUMENT TRONG MONGODB THÔNG QUA MONGOOSE

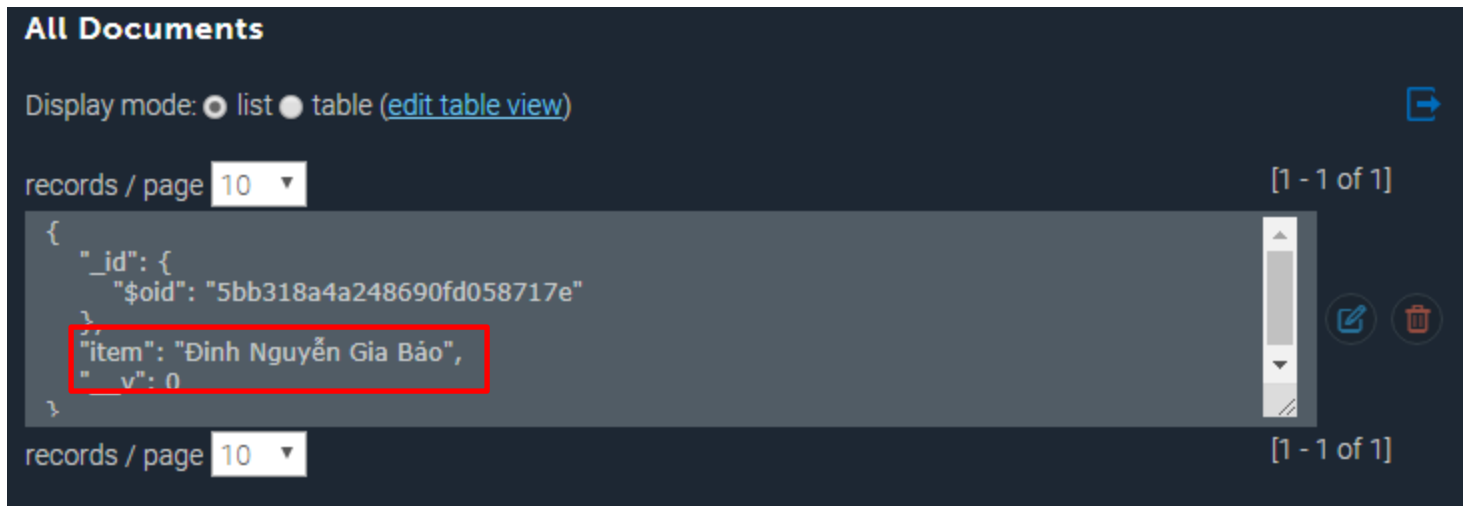
- ❑ Thêm dòng lệnh sau để tiến hành update collection và document cho MongoDB
- ❑ Lưu ý: id ở dạng chuỗi nên các bạn thêm dấu nháy, tránh trường hợp bị lỗi, hàm không xử lý được

```
//Tại phần _id các bạn lấy ở document
var edit = thucHien.update({_id: '5bb318a4a248690fd058717e'},{item: 'Đinh Nguyễn Gia Bảo'}, function(err){
    //Nếu bị lỗi mình sẽ bỏ qua
    if(err) throw err;
    //Nếu thành công mình sẽ thông báo
    console.log('Sửa thành công');
});
```

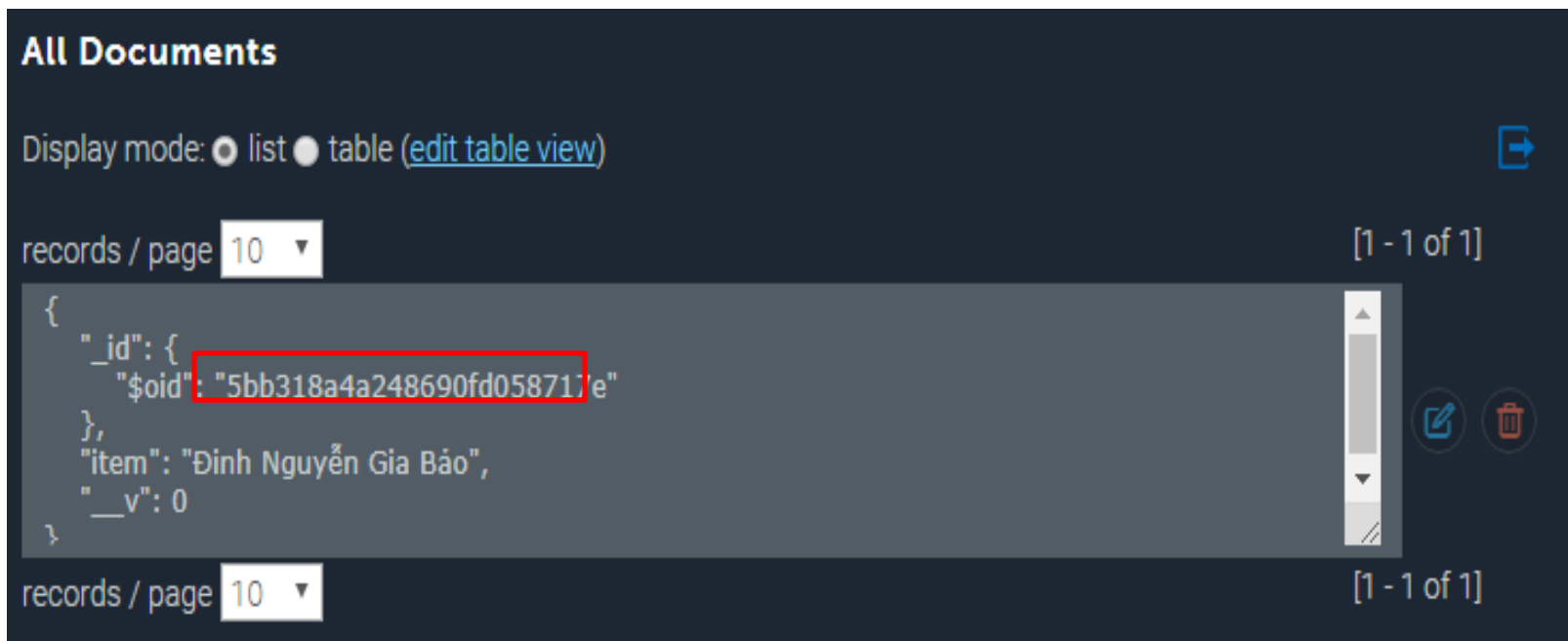
- ❑ Vào terminal gõ: nodemon index.js
- ❑ Kết quả như hình dưới

```
Administrator@YZFAXCMZING35CL MINGW64 /e/Hoc Tap/NodeJS/slide1
$ nodemon index.js
[nodemon] 1.17.5
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching: *.*
[nodemon] starting 'node index.js'
Kết nối MongoDB thành công
(node:4612) DeprecationWarning: current URL string parser is deprecated, and will be removed in a future version. To use the new parser, pass option { useNewUrlParser: true } to MongoClient.connect.
(node:4612) DeprecationWarning: collection.update is deprecated. Use updateOne, updateMany, or bulkWrite instead.
Sửa thành công
```


- ❑ Như vậy là xong các bạn có thể bật trang chủ mlab lên xem thử
- ❑ Document cũ của chúng ta chứa item: 'Đinh Nguyễn Cẩm Tú' giờ đã được sửa thành 'Đinh Nguyễn Gia Bảo'



- ❑ Trước khi thực hiện delete các bạn lấy 'id' của nội dung document các bạn muốn delete như hình dưới



DELETE DOCUMENT TRONG MONGODB THÔNG QUA MONGOOSE

❑ Sau khi đã có 'id' thêm dòng lệnh như bên dưới
(Lưu ý: id ở dạng chuỗi nên các bạn thêm dấu nháy, tránh trường hợp bị lỗi, hàm không xử lý được)

```
//id các bạn lấy ở phần Document
var xoa = thucHien.remove({_id: '5bb318a4a248690fd058717e'},function(err){
  //Nếu bị lỗi sẽ bỏ qua
  if(err) throw err;
  //Nếu thành công sẽ thông báo
  console.log('Đã xóa thành công');
})
```

DELETE DOCUMENT TRONG MONGODB THÔNG QUA MONGOOSE

- ❑ Vào terminal gõ: nodemon index.js
- ❑ Kết quả như hình dưới

```
Administrator@YZFAXCMZING35CL MINGW64 /e/Hoc Tap/NodeJS/slide1
$ nodemon index.js
[nodemon] 1.17.5
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node index.js`
Kết nối MongoDB thành công
(node:5580) DeprecationWarning: current URL string parser is deprecated, and will
be removed in a future version. To use the new parser, pass option { useNewUrlParser: true } to MongoClient.connect.
(node:5580) DeprecationWarning: collection.remove is deprecated. Use deleteOne,
deleteMany, or bulkWrite instead.
Đã xóa thành công
```

DELETE DOCUMENT TRONG MONGODB THÔNG QUA MONGOOSE

- ❑ Như vậy là xong các bạn có thể bật trang chủ mlab lên xem thử
- ❑ Các bạn có thể thấy trên mlab đã xóa document có chứa 'id: **5bb318a4a248690fd058717e**'



Sandbox databases do not have redundancy and therefore are not suitable for production. Read our documentation on [how to upgrade](#).

Collections

Users

Stats

Backups

Tools

Collections



Delete all collections



Add collection

NAME

DOCUMENTS

CAPPED?

SIZE ?

baitaps

0

false

7.98 KB



- ❑ Dựa vào kiến thức có được các bạn thêm trường bằng những thuộc tính sau
 - ❖ Fullname: kiểu String
 - ❖ Email: kiểu String
 - ❖ Phone: kiểu Number
 - ❖ Create_date: kiểu Date, default: Date.now
 - ❖ Status: kiểu String, enum['available', 'unavailable'], default:['available']

- ❖ --> Các bạn tiến hành thao tác create, update, delete document trong MongoDB thông qua Mongoose.




DEMO

- Các thao tác dữ liệu với mongoose



Tổng kết bài học

Phần I: Cài đặt và cấu hình Mongoose

 Tổng quan Mongoose

 Cài đặt Mongoose

 Cấu hình Mongoose

Phần II: Thao tác MongoDB server với NodeJS qua Mongoose

 Kết nối MongoDB server với NodeJS qua Mongoose

 Thao tác dữ liệu





KẾT THÚC