**Initiate mongo db**

cd /

cd data

sudo mongod

**((new tab))**

mongo

**Creating new app in mongoose – ADDING packages to package.json file**

**make a package.json file:**

npm init -y

npm install express --save

npm install ejs --save

npm install body-parser --save

npm install mongoose --save

**Show all DBs**

show dbs

**go into DB**

use “DB NAME:

**show all tables/collections**

show collections

**show all entires**

db.”DB\_NAME”.find({}).pretty()

**create a new table/collection**

db.createCollection(“COLLECTION NAME”)

**destroy a collection**

db.COLLECTION\_NAME.drop()

**Row record = “Document”**

**C**reating a document

**Syntax:**

db.COLLECTION\_NAME.insert({YOUR\_JSON\_DOCUMENT})

// Example:

db.ninjas.insert({name: "Trey", belt: "black", status: "awesome"})

**U**pdate – documents in a collection

db.COLLECTION\_NAME.update({QUERY}, {FIELDS\_TO\_UPDATE}, {OPTIONS})

db.ninjas.update({name: "Trey"}, {location: "Mountain View"})

***making updates to all:***

db.students.update({}, {$set: {number\_of\_belts: 0 / “black”}}, {multi: true});

***to make changes to an existing piece of info, without updating the entire document/row***

db.students.update({home\_state:"Washington"}, **{$set**:{number\_of\_belts: 1}})

***change column header name***

db.students.updateMany({},{$rename:{"number\_of\_belts":"belts\_earned"}})

remove column from document

bb.students.update({} , {$unset: {belts\_earned : 1}} , {multi: true})

**D**estroy/Delete/Remove – documents from collection

db.COLLECTION\_NAME.remove({YOUR\_QUERY\_DOCUMENT}, BOOLEAN)

*if the Boolean is true, it will remove only the first entry it finds.*

db.ninjas.remove({belt: "yellow"})

db.ninjas.remove({belt: "yellow"}, false)

1. Create a database called 'my\_first\_db'.

use my\_first\_db

// 2.Create students collection.

db.createCollection("students")

// 3. Each document you insert into this collection should have the following format: {name: STRING, home\_state: STRING, lucky\_number: NUMBER, birthday: {month: NUMBER, day: NUMBER, year: NUMBER}}

// 4. Create 5 students with the appropriate info.

db.students.insert({name: "Kobe", home\_state:"Pennsylvania", lucky\_number:8, birthday: {month:8, day:23, year:1978}})

db.students.insert({name: "Eminem", home\_state:"Michigan", lucky\_number:3, birthday: {month:10, day:17, year:1973}})

db.students.insert({name: "Michael", home\_state:"New York", lucky\_number:23, birthday: {month:2, day:17, year:1963}})

db.students.insert({name: "Kurt", home\_state:"Washington", lucky\_number:7, birthday: {month:2, day:20, year:1967}})

db.students.insert({name: "Russell", home\_state:"California", lucky\_number:0, birthday: {month:11, day:12, year:1988}})

// 5. Get all students.

db.students.find()

// 6. Retrieve all students who are from California (San Jose Dojo) or Washington (Seattle Dojo).

db.students.find({$or: [{home\_state:"California"}, {home\_state:"Washington"}]})

// 7. Get all students whose lucky number is:

// 7.1 Greater than 3

db.students.find({lucky\_number:{$gt:3}})

// 7.2 Less than or equal to 10

db.students.find({lucky\_number:{$lte:10}})

// 7.3 Between 1 and 9, inclusive

db.students.find({$and:[{lucky\_number:{$lte:9}}, {lucky\_number:{$gte:1}}]})

// 8. Add a field in each student collection called 'interests' that is an ARRAY. It should contain the following entries: 'coding', 'brunch', 'MongoDB'. Do this in ONE operation.

db.students.update({},{$set: {"interests":['coding', 'brunch', 'MongoDB']}},{multi:true})

// 9. Add some unique interests for each particular students into each of their interest arrays.

db.students.update({name: "Eminem"},{$push: {interests:{$each:['rapping', 'Haillie']}}})

db.students.update({name: "Michael"},{$push: {interests:{$each:['basketball', 'being the GOAT']}}})

db.students.update({name: "Kurt"},{$push: {interests:{$each:['Nirvana', 'playing guitar']}}})

db.students.update({name: "Russell"},{$push: {interests:{$each:['basketball', 'slam dunks']}}})

// 10. Add the interest 'taxes' into someone's interest array.

db.students.update({name: "Kobe"},{$push: {interests: "taxes"}})

// 11. Remove the 'taxes' interest you just added.

db.students.update({name: "Kobe"},{$pull: {interests: "taxes"}})

// 12. Remove all students who are from California (or Washington).

db.students.remove({$or: [{home\_state:"California"}, {home\_state:"Washington"}]})

// 13. Remove a user by name.

db.students.remove({name:"Eminem"})

// 14. Remove a student whose lucky number is greater than 5 (JUST ONE)

db.students.remove({lucky\_number: {$gt:5}}, {justOne:true})

// 15. Add a field in each student collection called 'number\_of\_belts' and set it to 0.

db.students.update({},{$set: {"number\_of\_belts":0}},{multi:true})

// 16. Increment this field by 1 for all students in Washington (Seattle Dojo).

db.students.update({home\_state: "Washington"},{$inc: {number\_of\_belts: 1}},{multi:true})

// 17. Rename the 'number\_of\_belts' field to 'belts\_earned'

db.students.update({}, {$rename: {"number\_of\_belts":"belts\_earned"}})

// 18. Remove the 'lucky\_number' field.

db.students.update({}, {$unset:{lucky\_number:""}},{multi:true})

// 19. Add a 'updated\_on' field, and set the value as the current date.

db.students.update({}, {$currentDate:{updated\_on:true}},{multi:true})

***RUNNING MONGOOSEJS COMMANDS IN SERVER.JS (DOCUMENTATION)***

<http://mongoosejs.com/docs/index.html>

***Installing Dependencies***

yourcommandline> npm init -y

yourcommandline> npm install express --save

yourcommandline> npm install ejs --save

yourcommandline> npm install body-parser --save

yourcommandline> npm install mongoose –save

((INSIDE ANGULAR FOLDER))

ng new NEWPROJECT –routing

npm install ng2-cookie

npm install service ‘name’

models and controllers should be a 1 to 1 relationship. Each model must have its own controller