**NodeJS**

**Node Intro**

It’s a JavaScript run time, that’s the Js that run the outside from the browser and run it on our machine, so you can use the same syntax as JS for this

The purpose of Why are we using this because of for creating web server, command line tools (application that run in our command line, when we don’t have UI yet), Video Game, etc.

Node REPL

* REPL = Read, Evaluate, Print, Loop
* The Js console in the browser is a REPL, if we type node in terminal <- That’s also REPL
* REPL = we use it to do somethinf like to code it, it will read our syntax, evaluate it print it also loop it
* So, we use it to debug
* But in Node, it’s like JS world but there is different, it doesn’t have window, document and DOM APIs like in the browser
* But instead of ‘window’ it is ‘global’

A screen shot of a computer

AI-generated content may be incorrect.

* We can run node file through terminal window rather than browser console

Process & Argv

* Process is the object that’s available in global scope to be used to check node
* Process is like nvm (node version management) but it different
  + NVM = to change or do something with node version
  + Process = built in Node that use to do with current node that are using, and give you info
* Process.argv
  + Argv = method for this built in to get argument from the command line

A screen shot of a computer code

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

* + From the Example you can see, I have passed the value thorugh command line after the program name
  + Then it prints the name that I have passed
  + But if we see the entirely array it keep
    - First where the node lives or the path to the Node.js executable
    - Second the path to our script

**Modules & The NPM Universe**

Module,exports

* This is the export function that make for other file to call that function in another file

A screen shot of a computer program

AI-generated content may be incorrect.A screen shot of a computer program

AI-generated content may be incorrect.A black background with white text

AI-generated content may be incorrect.

* This how the app.js call the function or get the value from math.js file
* App.js have to require those functions (I have used extraction)

Introducing to NPM

* It’s a library that have tons of packages. We can use tool or code for free via NPM
* We can use it in command line
* Npm install node\_package\_that\_you\_want\_to\_install
* To require the package, you don’t need ‘./’ for tell its path, just put the name of the package

A computer screen with text on it

AI-generated content may be incorrect.

Adding Global packages

* To install npm library globaly use
* NPM I -g library\_name
* Then use ‘NPM link library\_name’ to link from global to local node project in specific folder

All important packages.json



* Name: the package/project name
* Version: current Version
* Description: short Summary of what this package does
* License: License type
* Author: Who created it
* Type: Module system (CommonJS = require/module.exports)
* Main: Entry point file (when someone runs/uses this package)
* Scripts:

**Server**

Express

* It’s the package but it does a lot for us, so we know it as a framework
  + Framework: to help us to make a web application <- helper
    - Other people have written for us
    - It’s providing the structure, we have to do follow by its rules
  + Library: also, the other people have written it for us
    - But we are in control, we can decide to put it any where in our code
* It helps to start up a server to listen for requests.
* It will parse incoming requests because HTTP request is just a text not JS object, so it turn the request into objects
* Initial structure for Express framework

A screen shot of a computer program

AI-generated content may be incorrect.

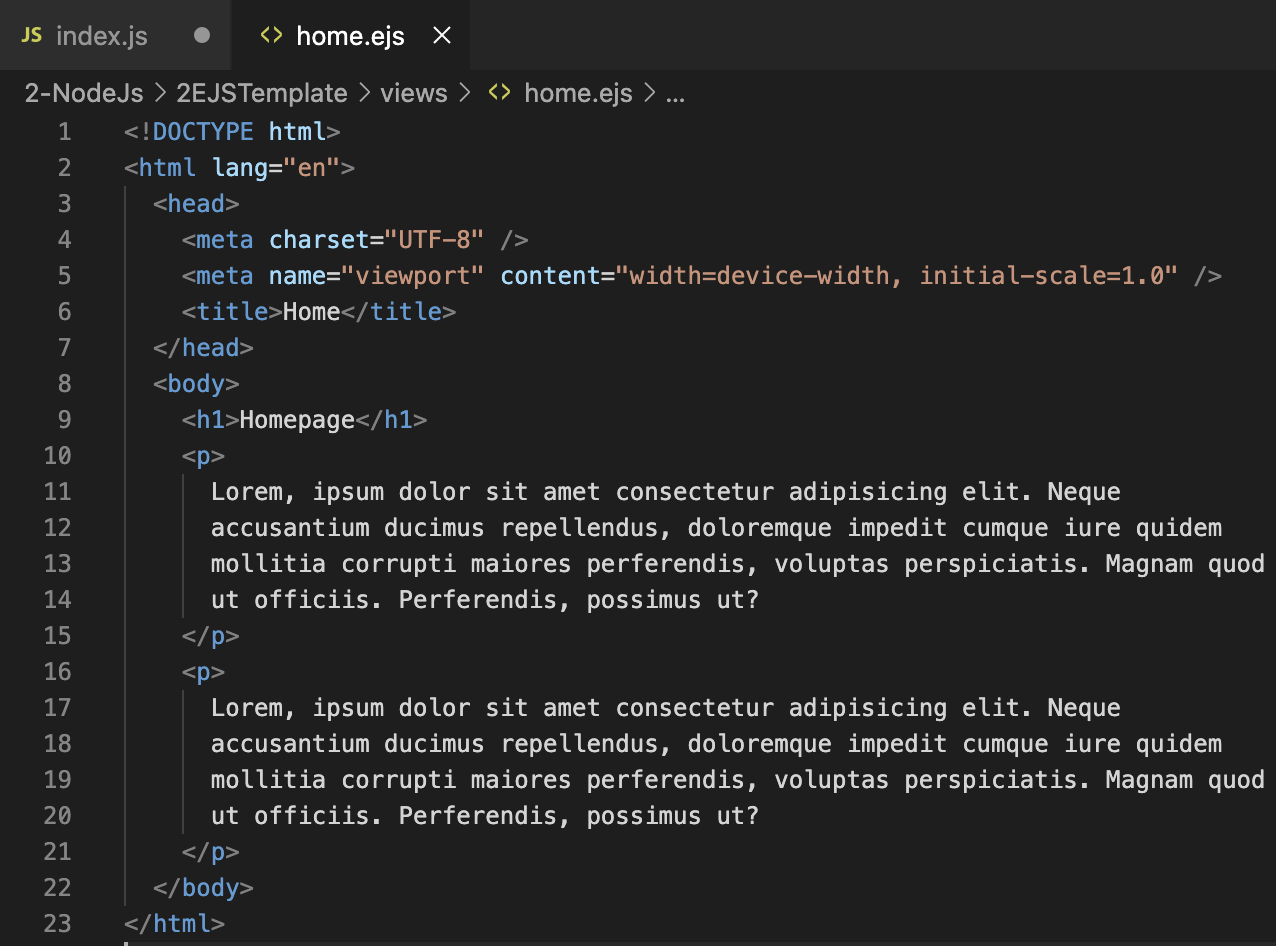
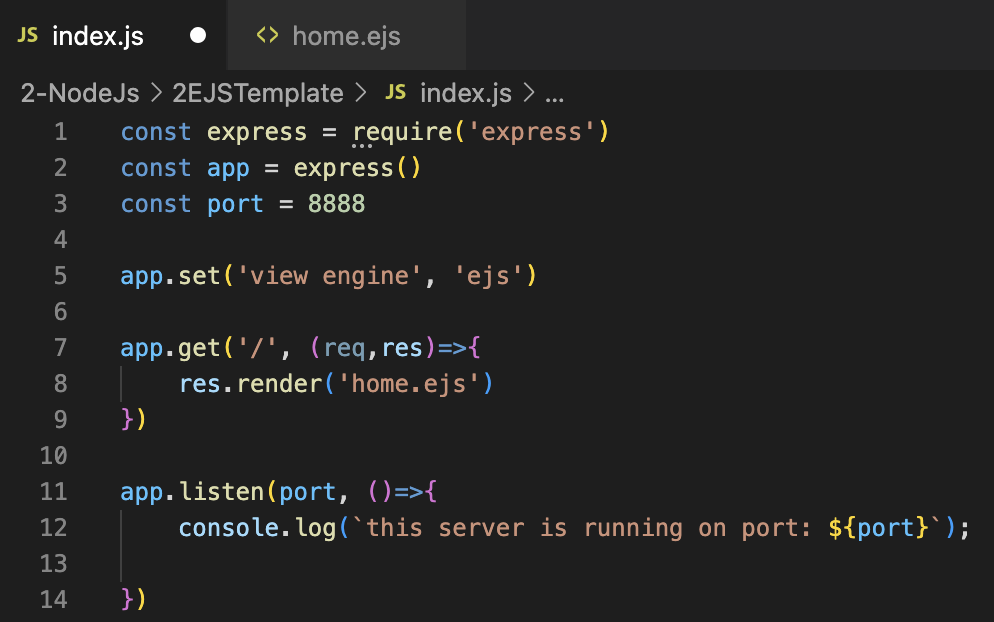
**Creating Dynamic HTML with Templating**

Templating

* it allows us to define a preset ‘pattern’ for a webpage, so that we can dynamically modify
* We can use EJS (embedded java script tempting) to do java template

EJS

* We need to ‘npm i ejs’
* And create folder name ‘views’ then create file .ejs inside that folder to be html file
* In that .ejs file we can write html in there



EJS

* We need to ‘npm i ejs’

Setting the ‘Views’ direction

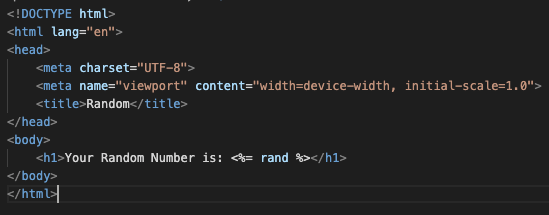
A black screen with text

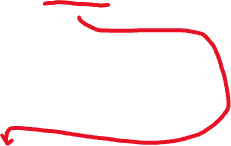
AI-generated content may be incorrect.

* To call where ever

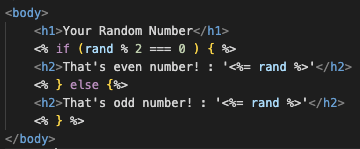
EJS using



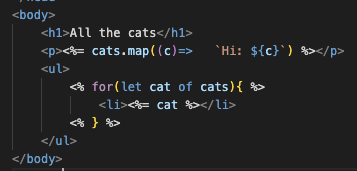




* We can do what ever we want to send parameter from nodejs to html, by ejs tags
* In that tag, you can do loop, condition or what ever, but we prefer to do that part on backend more than do it directly on html
* The tag we can check out more in EJS web
* Condition: by using tag <% condition %>



* Loop



Static Express\*\*\*

* Express.static <- it’s a middleware
* Use it with app.use()
  + Inside the app.use(), everytime we ran the code, it will execute any request, it doesn’t care to post or get, just execute
  + We use this with ‘express.static(folder)’ to serve our assets (css,html,js)

A black screen with blue and white text

AI-generated content may be incorrect.

* To make it more roburst



**RESTful Routes**

Get vs Post request

* Get
  + Get information
  + Sending data via query string
  + Information apear on url
  + Limit amount of data can be sent
* Post
  + Sending information to server, via body
  + For write, update and create
  + Send by any sort(JSON)

Parsing req body

* Post

A screen shot of a computer program

AI-generated content may be incorrect.

* + From this, we send a data via http in www form to backend



* + So, in backend we have to use ‘app.use(express.urlencoded({extended: true}))’ to be middleware to receive any req that in form ‘www-form’
  + App.use: is the way of running some code/function on every single request no matter get or post

Rest

* REST API: a architectural style for system api comunication by following its rules on HTTP protocol like GET, POST, PATCH, DELETE(CRUD)
* Main idea:
  + Use URL to specify “resource” like ‘/users/products’
  + Use HTTP method to tell, what does this api do?
    - GET: sending data from server to client
    - POST: creating data from client to server
    - PUT/PATCH: fixing/updating data from client to server
    - DELTE: deleting data
* It’s like a bridge to communication between server and client to make it flows
* We need to specify the type followby its task

**Database (Mongo DB)**

Mongo

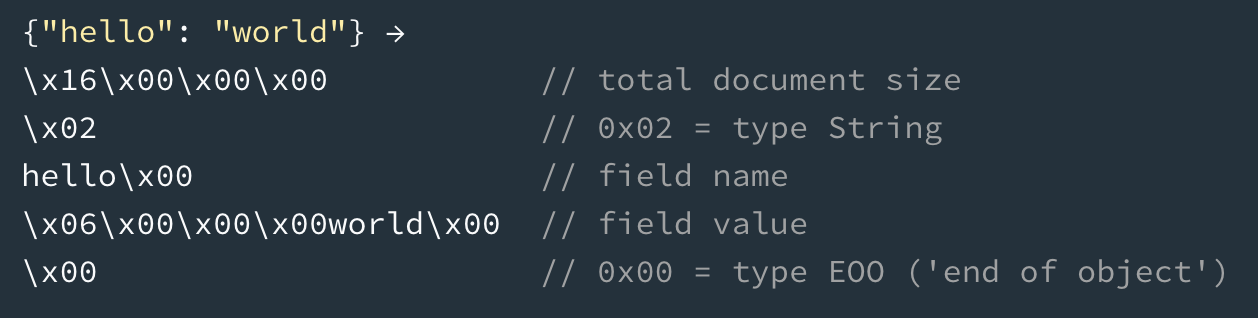
* No need SQL language
* It fits with JS, Express, NodeJs and poppular
* Large community
* It’s NoSQL db
* It’s a document database
* He said it easier to start with(really?)

Mongo Shell

* It’s JS shell, so it can use js command, like .toUpperCase() for string method
* Commands or tips:
  + Mongosh <- to start mongo server
  + Command + k: to clear screen
  + help: to help
  + show databases || show dbs: to show all database in mongo
  + db: show current database
  + use (database name): to use that database
    - If that database doesn’t exist, it will create a new one, but it not actually created. We must insert data into it first, then it will usable database
  + control + c: to quit mongo

BSON

* Mongo expects BSON data format to show or to get
* Like JSON but better
  + JSON is text-based format, slow to parsing (UTF-8 String)
  + BSON stands for ‘Binary JSON’ (Binary)
  + We can write regular JSON, but MONGO will convert it to be BSON by itself
  + In physical file it looks like it takes up more characters, but in memory it takes up a lot less space compares to JSON



INSERT with Mongo

* In mongo, we actuall insert in a collection
  + Collection: a grouping of data in a database
* When we insert, we need to insert it into a collection.
  + Mongo will create that collection for us when we first insert into that collection that doesn’t yet exist
  + There are 2 inserts
    - Db.collection name.insertOne(): insert one item
    - Db.collection name.insertMany(): insert many items
    - Db.collection name.insert(): used to insert in two ways (one, many). Right now it deprecated
  + If you insert data, it will create a unique \_id in every rows automatically
    - \_id you can control it

A screen shot of a computer

AI-generated content may be incorrect.

Finding with Mongo

* It’s a ‘select’ for sql

A screen shot of a computer

AI-generated content may be incorrect.

* Db.collection.find(query,projection) <- full commend
  + Query (Optional) = to filter
  + Projection (Optional) = specifies the fields to return
  + Find (this command) returns cursor, it’s pointer it’s a refference, it’s not return actual data immediately
* Use .find() = ‘select \* from dogs‘ that collection (table)
* Use .find({something in this curly bracket}) = ‘select \* from dogs where something’ <- make criteria
* Use .findOne() = return one item

Update with Mongo

* db.collection.updateOne(<filter>,<update>,<options>)
  + update one thing that matches
* db.collection.updateMany(<filter>,<update>,<options>)
  + update all

A black screen with white text

AI-generated content may be incorrect.





* we must use $set in <update> to set new data
  + $set: {<field>:<new value>}
  + We can not put the field and new value without $set

A screen shot of a computer code

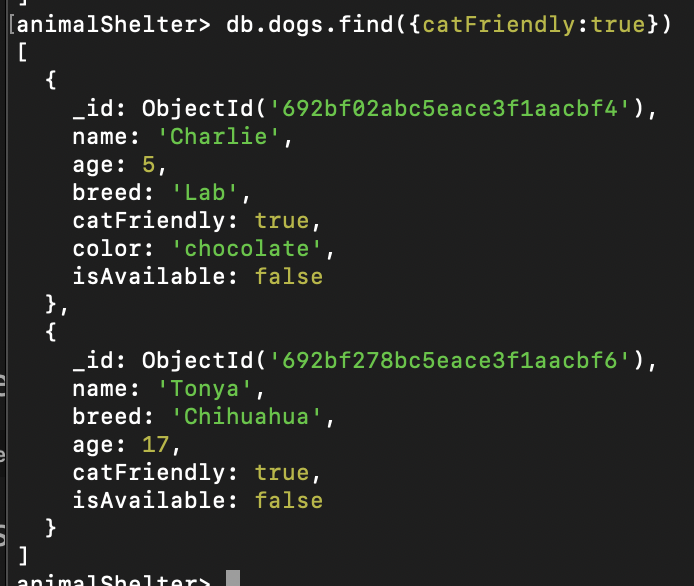
AI-generated content may be incorrect.

* We also can update multiple fields

A computer screen with text and images

AI-generated content may be incorrect.

* We can update/add a new field to that object



* We use updateMany to update all new field to catFriendlt = true



A computer screen with green and yellow text

AI-generated content may be incorrect.

* We can update and use anoter methods to this collection like $currentDate
  + It will create a new field like this situation I use ‘lastChanged’, this method will add a current date in form of timestamp to this field

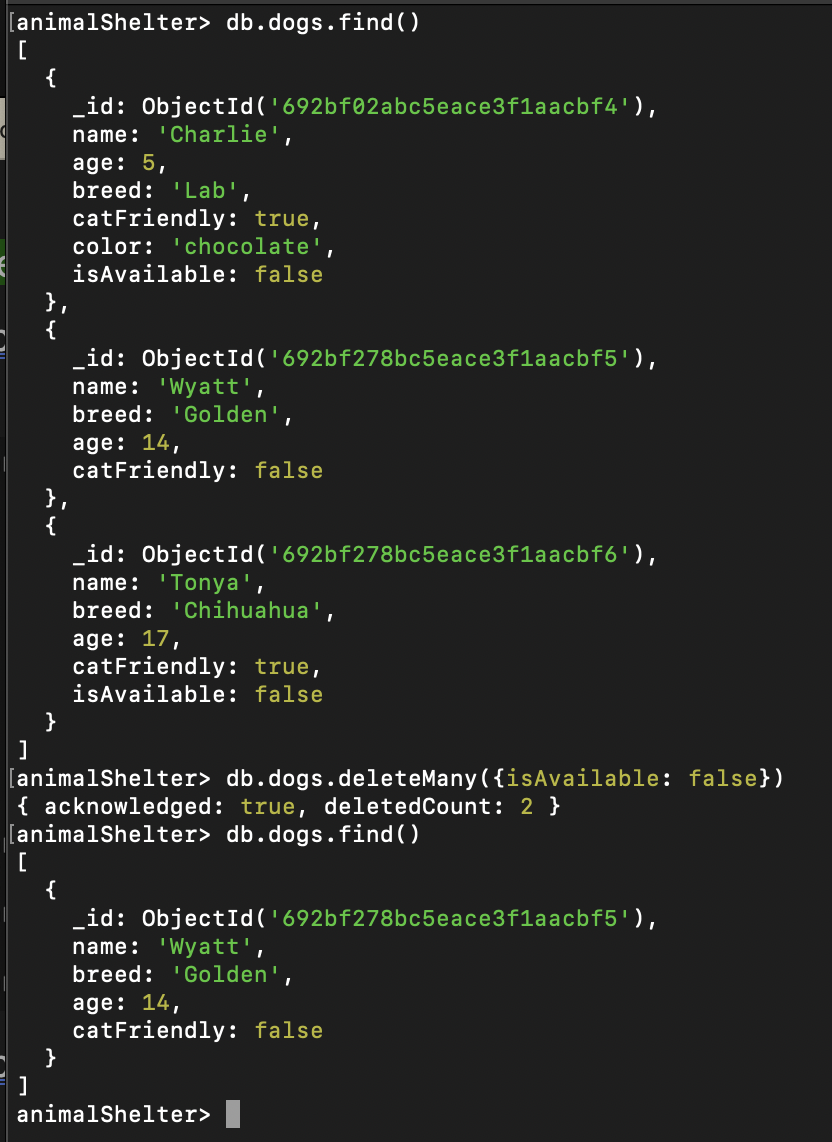
Delete with Mongo

* db.collection.deleteOne()
  + delete one thing that matches

A screen shot of a computer

AI-generated content may be incorrect.

* db.collection.deleteMany()
  + delete many things that match



* + and you can delete all data by not specify any criteria in it by using deleteMany



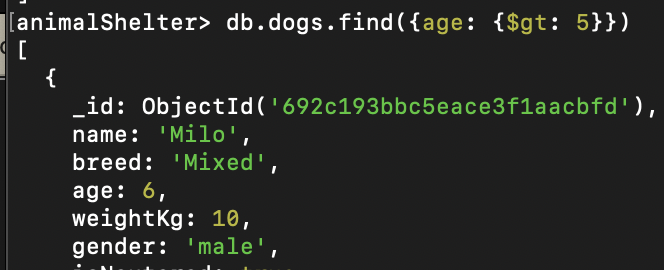
Additional Mongo Operator

A screen shot of a computer

AI-generated content may be incorrect.



* we can use ‘’ in find to specify or go deeper to each layer to specific data to get
* we can do a comparison in Mongo



* + $gt: greater than
  + $gte: greater than or equal to
  + $lt: less than
  + $lte: less than or equal to

A computer screen with green and white text

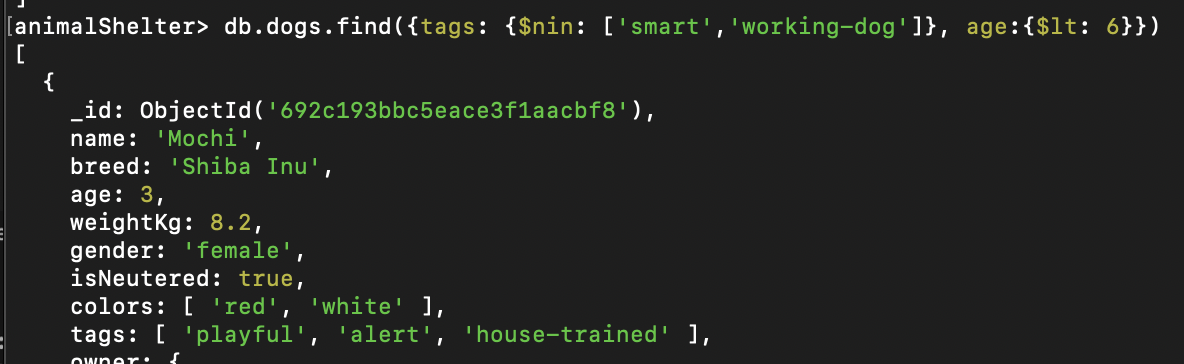
AI-generated content may be incorrect.

* + We can use $in to do with an array like ‘select \* from where tags in (‘’,’’)’
  + Below image is tell how to use ‘and’ like ‘select \* from where tags in ('smart','working-dog') and age < 6’

A screen shot of a computer

AI-generated content may be incorrect.

* + Below image is tell how to use ‘not’ like ‘select \* from where tags not in ('smart','working-dog') and age < 6’



**Mongoose**

Mongoose

* To connect mongo with programming language
* Known as ODM
  + ODM: Object Data Mapper (Document data mapper)
  + It helps to define data structure before do further like defined input sturucture before insert, so insert data must follow those defiend structure.
* Mongoose is ODM for Mongo and NodeJS to connect these two
* It’s a driver from mongo
* Provide a lot of functionality
* It maps data that comes backs from Mongo, and data we want to insert to the Mongo
  + It maps them into uasable JS objects
* Mongoose makes mongo friendlier and more powerful on the JS

Connecting Mongoose to Mongo

* First, you need to ‘npm I mongoose’
* Then after import, you have to use mongoose.connect(‘mongodb:IP/database name’)

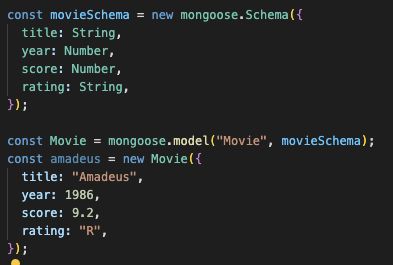
A computer code on a black background

AI-generated content may be incorrect.

* From the picture
  + movieApp: my database name, you can create in mongo or just put it in this. if it does not exist Mongo will create it for you
  + after that line, we can follow by the instrtuction on Mongoose document

Model

* It like a class in JS
  + It’s a class for a collection
* Represent data in Mongo or in collections
  + So, you can CRUD it
* You need to create a model first before you start
  + But before that you need to create a schema
  + schema: object/collection structure



* + From this. You already have model name’s ‘Movie’
  + also create a collection in database movieApp
    - but from the example your collection will not create as ‘Movie’ it’s going to be ‘movies’
    - It’s the rule to create model you must put the capital letter in that name and mongo will create a collection with that name for you, but it will change it for you automatically to small letter with the ’s’
  + And from the amadeus’s object
    - From this we have create it to memory in JS, it will not appear on MongoDb untill you put .save to it
    - Await amadeus.save() <- now it will insert to db

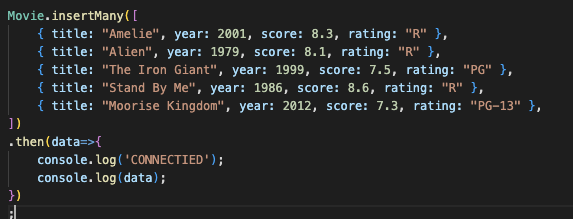
A black screen with text

AI-generated content may be incorrect.

* This call ‘document’ it’s a single record in the database
  + It’s an instance of the Model
  + Amadeus is a document, not a model
  + Documents have:
    - Their own data like amadeus
    - They have instance method
    - The ability to store in mondoDb(collection) by .save() <- model can’t
* Model does, how’s the appreance
* Document is created from model but has its own detail

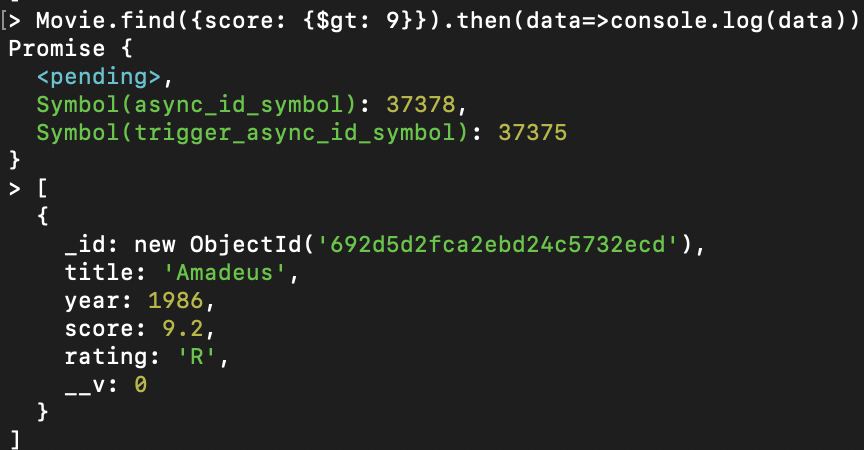
Insert in Mongoose

* From insert command, you don’t need. save anymore. It’s going to insert to the db directly
* Model.insertOne()
* Model,insertMany([])



Finding in Mongoose

* Model.find()



* findById
  + just put an id into this find(). It will gonna find that specify Id to you
  + No need to put an \_id field to it

A computer screen with numbers and symbols

AI-generated content may be incorrect.

Update in Mongoose

* Model.updateOne()
  + Update one item

A screen shot of a computer

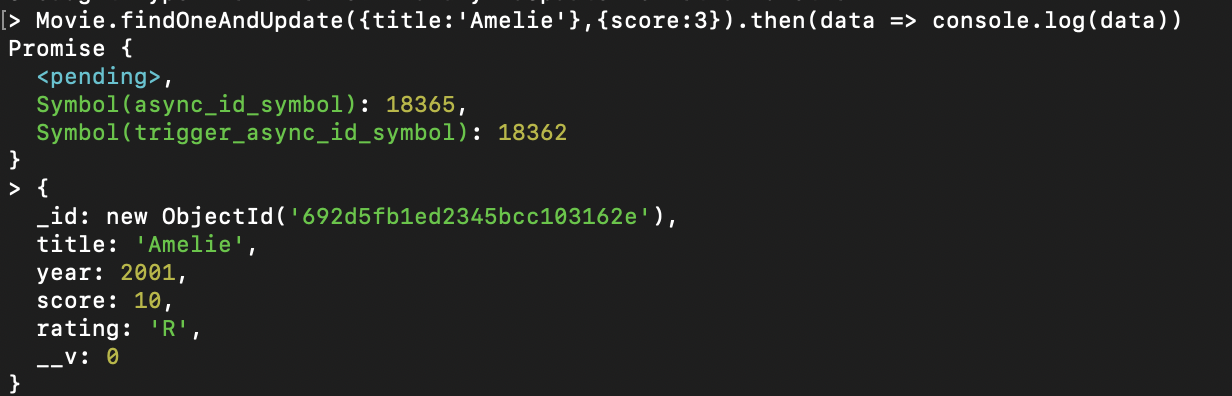
AI-generated content may be incorrect.

* Model.updateMany()
  + Update many items

A black background with white text

AI-generated content may be incorrect.

* You can see from those two updates. They don’t seem to show the data that has updated, so we have another way to done that
* Model.findOneAndUpdate()



* + This can update also showing the data
  + It seems to have only findOne There is no findManyandUpdate

Deleting in Mongoose

* Model.deleteOne()
  + Delete one item in collection

A screen shot of a computer code

AI-generated content may be incorrect.

* Model.deleteMany()
  + Delete Many items in collection

A screen shot of a computer code

AI-generated content may be incorrect.

* Model.findOneAndDelete || Model.findByIdAndDelete()
  + Like update, we can use this method delete to get back what we have deleted

A computer screen with numbers and letters

AI-generated content may be incorrect.

Mongoose Schema Validation

* Validation of each variable should be
* It’s like TypeScript, but do different purpose
  + TypeScript checks type value to variable or before receive input also before using data
  + Mongoose schema to make a direction for creating a Model and collection

A screen shot of a computer program

AI-generated content may be incorrect.

* + Input to this model/collection must follow to this schema that’s declared
  + From the image, required or default is the optional. We can put it or not, up to us
    - Required: can’t be blank
    - Default: the default value for this variable
      * We can let it blank because of its default value

Validation Mongoose Update

* This is how to validate the data before updating by using option

A screen shot of a computer code

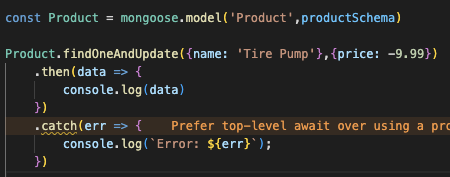
AI-generated content may be incorrect.

* We’ve set the price to has min 0

A screen shot of a computer program

AI-generated content may be incorrect.

* Without option it will update data normally
* Follow by the schema we’ve set, what if I set the price to negative value



A computer screen shot of a computer code

AI-generated content may be incorrect.

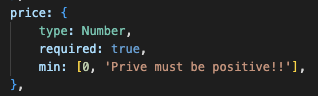
* It can be updated, but why? It doesn’t follow the schema that the value must have more than 0
  + Because we don’t give it options

A screen shot of a computer

AI-generated content may be incorrect.

* So, we add options to it
  + New: this option set to True for get real time update value
  + runValidators: to check value againt the schema before getting an update

Mongoose Validation Errors

A black screen with white text

AI-generated content may be incorrect.

* We can add a message error on its schema
  + The position to put an error message, check again in mongoose document.
  + There is different position error message to put in
* But this method is not common





Model Instance Method

* It’s like a class in JS
* It’s a function you define on schema, and every document created from that model can call them
* Like we say ‘Model.find()’ find() is a method, and it can use with every model <- This is static method

A black background with colorful text

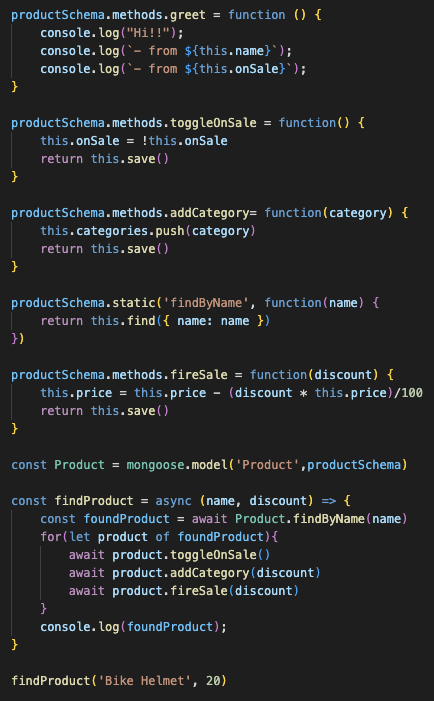
AI-generated content may be incorrect.

* To define instance method, we must use function instead of arrow funtion, because we want to use ‘This’ for accessing



* Like this, we can create a bunch of methods also use ‘this’ for accessing through the schema in object and change something
  + Tip: this.save() is using sometime to finish saving, so we can use ‘return’ then use ‘await’ in front of where we’re calling function

Adding Model Static Method





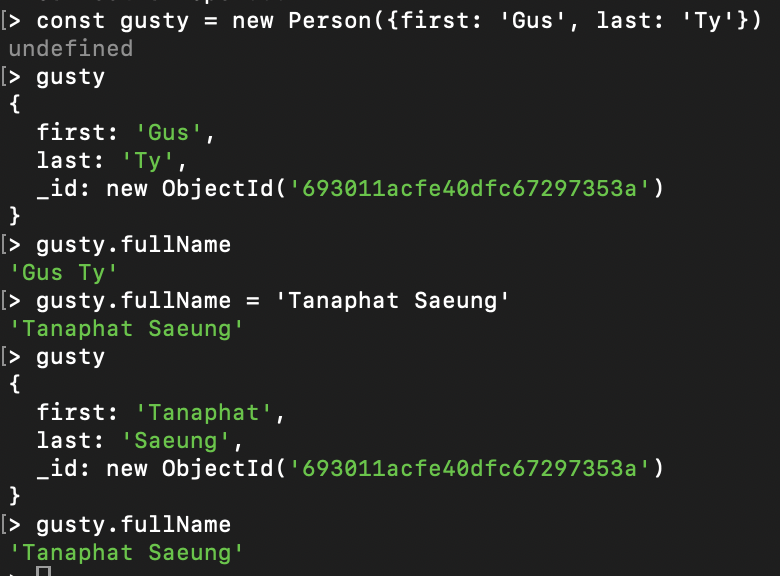
* I make it more dynamic, from this example, I’ve done by using ‘find()’ it returns to object not array, so I must loop through it
* So, this is how to use static method

Mongoose Virtuals

* It gives us ablity to add properties to a schema that dont’t exist in the database itself

A screen shot of a computer code

AI-generated content may be incorrect.



* Virtual is just in JS, not actually in database
* We can do what ever we want before we insert or .save() into database, but that function is just using to look at the data not actually happen I mean data
* From this mongoose virtual, I’ve used ‘get’ and ‘set’ method
  + Get: you can create a function to get a data that you want to see
  + Set: you can set a data input by using the method name then make it equal to something you want to change, and it will happen if you save it

Mongoose Middleware

A computer screen shot of text

AI-generated content may be incorrect.

A computer screen shot of a black screen

AI-generated content may be incorrect.

* Middleware are functions that run before or after a Mongoose operation
* Pre & Post:
  + Pre: run something in middleware first, then do that operation
  + Post: run something in middleware after doing that operation
* From the image, I said before save() it must log message first, after that run other message
* Middleware are functions that are registered on a schema and run before or after a Mongoose operation

**Mongoose + Express**

Setup



**Error - Handling**

Express built in Error Handler

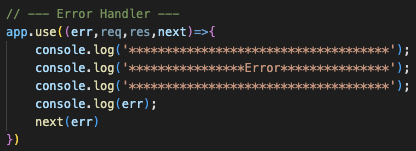


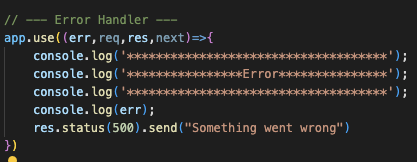
* This is a built-in in Express
  + You can control the error by this Express’s built-in



* + And this will happen

Custom Error Handler





* You can receive that thorwing error to the end of your code with this err middleware
* I prefer res more than next

YELPCAMP Error Handler

* I have splited all the error handler to man file for code cleaner

A computer screen shot of text

AI-generated content may be incorrect.

* From app.all:
  + If user gone to invalid page, this error is going to show up and tell user ‘Page Not Found’ with status 404
* The last middleware is to handle all the error and pass it to another web page
  + If that page has got error, but I don’t put any error message to them. It will go to this middleware with message ‘Something went wrong’ and got status 500, then pass to ‘error’ page



* Here is an error page, the style came from bootstrap

A black background with text

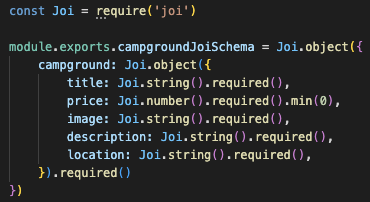
AI-generated content may be incorrect.

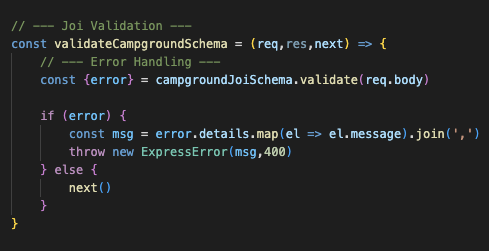
* Here is a comprehensive route that includes the error handling with Async and Schema validation

A computer screen with text and symbols

AI-generated content may be incorrect.

* From this Async error
  + I pass a whole function to it, then it will execute that function, and checking if it causes any error from this function, it will pass next to another middleware, if not it will execute then response something from that function or route





* This is Joi validation, not mongoose validation
* This Joi is going to check schema to all parater that it got
* It’s like Zod from typeScript