Please update this doc daily with what you are learning – along with description.

Also push this doc daily to GitHub

Refer my Shopify repository for the implemented code = https://github.com/PRD24397/Shopify

-> What is Mongo dB and why it is used

1.Store and work with large scale extensive data intensive application

2.NoSql database which is based in JSON and BSON representation for quering and creating databases

3.fast then the relational sql databases

4. Consists of Databases collections and Documents

5. Mongo dB is schema less

6.Schemaless feature of mongo dB gives more flexibility for the application to grow and to change data requirement as well easily over long time

7. Tutorial that I referred to for Mongo DB => <https://www.tutorialspoint.com/mongodb/>

-> Relations in mongodb

1.This can be done by two ways

a. nested documents and

b. References

2. No sql characteristics

1.No predefines data schema and no structure is required.

2.Fewer data relations because of embedding of documents

These are the reasons why mongo dB is important

->Setting up mongo dB

1. For setting up mongo dB locally install mongo dB and mongo dB compass - <https://docs.mongodb.com/v3.2/tutorial/install-mongodb-on-windows/>

2.Configure mongo dB compass GUI

->Install mongo dB driver

1. In terminal npm install –save mongo dB for installing the node mongo dB drivers

2. It is good practice to handle the connection related logic in a single file

In database.js, for connecting to mongo dB.

const mongodb = require('mongodb');

const mongoClient = mongodb.MongoClient;

let url = "mongodb://localhost:27017/mydb";

const mongoConnect = (callback) => {

mongoClient.connect(url).

then(

client => {

console.log('Connected');

callback(client);

}

).

*catch*(err => {

console.log('Error');

console.log(err);

})

}

module.exports = mongoConnect;

->Creating and connecting a database connection

1. What is callback function we use in the mongodbConnect = <https://codeburst.io/javascript-what-the-heck-is-a-callback-aba4da2deced>

For connecting to the database in database.js do

const mongodb = require('mongodb');

const mongoClient = mongodb.MongoClient;

let url = "mongodb://localhost:27017/mydb";

let \_db

const mongoConnect = (callback) => {

mongoClient.connect(url).

then(

client => {

console.log('Connected');

\_db = client.db()

callback(client); *// returning a client object in callback*

}

).

*catch*(err => {

console.log('Error');

console.log(err);

*throw* err;

})

}

const getDb = () => {

*if*(\_db){

*return* \_db;

}

*else*{

*throw* 'No database found';

}

}

exports.mongoConnect = mongoConnect;

exports.getDb = getDb;

Here there is exported method getDB() that helps us to use the database and get database across multiple pages as such in product.js

const getDb = require('../util/database').getDb; *// call this function to get access and use the database*

## Promises in JavaScript == <https://www.youtube.com/watch?v=s6SH72uAn3Q>

## what is .then() and .catch() == <https://www.w3schools.com/nodejs/nodejs_mongodb_createcollection.asp>

* Insert data to the database

In the Product Model of our demo app

const mongodb = require('mongodb');

const getDb = require('../util/database').getDb;

class Product {

constructor(title, price, description, imageUrl) {

*this*.title = title;

*this*.price = price;

*this*.description = description;

*this*.imageUrl = imageUrl;

}

save() {

const db = getDb();

*return* db

.collection('products')

.insertOne(*this*)

.then(result => {

console.log(result);

})

.catch(err => {

console.log(err);

});

}

The save function has db.collections(‘ ’) which makes a new collection if none is present or it enters the javascrit object passes into the insertOne method. To insert more than one record insertMany() is used.

Output

[ Product {

title: 'sss',

price: '12',

description: 'hhshshs',

imageUrl: 'www.google.com',

\_id: 5c2dd380ef05495c841b09bf } ],

insertedCount: 1,

insertedId: 5c2dd380ef05495c841b09bf }

Created Product

* Fetching Products (Retrive entries from database)

The find() can be calles for finding all the products and are used with filters for finding the specific data

find() returns a cursor. Cursor allows us to go through our document step by step. Handle are provided by the mongodb to manually retrive the next document

toarray() converts the retrived entries in form of javascript arrays. Implement pagination if the retrived data from the database is very large