Minhas colas de MongoDB:

**“C\Program Files\MongoDB\Server\5.0\bin\mongod.exe”**

Esse código no CMD serve para ligar o MongoDB

**“C\Program Files\MongoDB\Server\5.0\bin\mongo.exe”**

Esse código em outro CMD, serve para ligar o MongoDB de fato, usa como segundo código.

**Show dbs**

Esse código faz mostrar todos os bancos disponíveis.

**Use nomedodb**

Para usar e criar um banco específico.

**Db.createColletion(‘nomedacolecao’)**

Para criar uma coleção no banco e assim aparecer.

**Db.nomedacolecao.drop**

Para apagar a coleção selecionada do banco que está usando.

**Db.nomedacolecao.insertOne({nome: “Alexandre”, password: “12345”})**

Para adicionar um conteúdo à coleção, simulei sendo uma coleção de usuarios.

**Db.nomedacolecao.insertMany([{nome: “Ale”}, {nome: “Esse”}, {nome: “Leo”}])**

Para adicionar vários conteúdos à coleção, simulei sendo uma coleção de usuarios.

**Db.nomedacolecao.find()**

Esse comando mostra todo o conteúdo da coleção.

**Db.nomedacolecao.findOne()**

Esse comando mostra o primeiro conteúdo da coleção.

**Db.nomedacolecao.find().limit(2)**

Esse comando mostra 2 conteúdos da coleção, podendo ser mais.

**Db.nomedacolecao.find({nome:”Alexandre”})**

Esse comando mostra somente conteudos que tem esse nome, pode ser qualquer parâmetro. Colocando .pretty() no final de cada comando, ele retorna um JSON mais legível.

**Db.nomedacolecao.find({type: “usuario”}).count()**

Esse comando retorna a quantidade das pessoas que tem o tipo == usuário, servindo para qualquer parâmetro, e tirando o parametro entre parenteses (type), ele retorna a quantidade de todo o conteúdo da coleção.

**Db.nomedacolecao.find().sort({name:1})**

Esse comando faz os nomes aparecerem de forma alfabética do menor para o maior, do maior para o menor eu colocaria -1.

**Db.nomedacolecao.updateOne({nome: “Marcela”}, {$set:{tel:”4521402”}})**

Esse comando faz o conteúdo que o nome é Marcela, ser adicionado o numero de telefone igual ao que coloquei.

Podendo colocar qualquer outro parâmetro, adicionar nome e afins.

**Db.nomedacolecao.updateMany({}, {$set:{class:”2001”}})**

Esse comando faz todos os conteúdos da coleção terem a adição de um parâmetro chamado Class.

**Db.nomedacolecao.updateMany({}, {$unset:{class:””}})**

Esse comando faz todos todos os parametros de class serem apagados de todos os conteudos da coleção

**Db.nomedacolecao.update({nome: “Marcela”}, {name: “Marcela”, password: “5555”})**

Esse comando faz o documento que tem o nome Marcela atualizar num todo, se tiver 5 parametros e eu só passar esses 2, nome e senha, ele vai remover tudo e só deixar esses dois atualizando.

Recomendado quase nunca.

**Db.nomedacolecao.update({nome: “MarcelaTais”}, {name: “Marcela”, password: “5555”}, {upsert: true})**

Esse comando se não achar um nome igual MarcelaTais, ele vai criar conteúdo já com os parametros apresentados aqui.

**Db.nomedacolecao.updateMany({}, {$rename:{nome: “nomes”}})**

Esse comando muda o nome dos parâmetros, de nome para nomes, de todos os conteúdos.

**Db.nomedacolecao.updateOne({name: “jonas”}, {$inc:{views:1}})**

Esse comando faz incrementar o numero de views no usuario Jonas

**Agora na IDE, usando Mongoose para linkar ao código.**

**Const mongoose = require(‘mongoose’)**

Esse Código é para chamar o mongoose no código.

**Mongoose.connect(‘mongodb://localhost.com/nomedobanco’, (err, db) => {**

**Console.log(err);**

**Console.log(db);**

**})**

Esse comando é o conector do banco ao sistema back-end, primeiro jeito.

**Mongoose.connect(‘mongodb://localhost.com/nomedobanco’)**

**.then(db=>{Console.log(db)})**

**.catch(err=>{Console.log(err)})**

Esse comando é o conector do banco ao sistema back-end, segundo jeito.

**const linkSchema = new mongoose.Schema({**

**title: String,**

**description: String,**

**url: String,**

**click: Number**

**})**

Esse código é o Schema no banco de dados, tudo que vai ter na coleção, titulo, descrição e tals, podendo substituir por nome, senha, email e coisas do tipo.

Se por exemplo no title eu puser {type: string, required: true} faz o title ser obrigatório.

**const Link = mongoose.model('Link', linkSchema)**

**const link = new Link({**

**title: "Baixar Thumbnail",**

**description: "Link para meu site de miniaturas",**

**url: "https://www.baixarminiatura.com",**

**click: 0**

**})**

**link.save().then(doc=>{**

**console.log(doc)**

**}).catch(err=>{**

**console.log(err)**

**})**

Esse código é o conteúdo que será adicionado na coleção do banco de dados de forma manual. Esse ‘Link’ cria o nome das tebelas, se eu criar “usuario” e colocar os parametros, vai ser criada uma coleção chamada usuários automaticamente, essa then é para retornar um erro caso haja

**mongoose.connect('mongodb://localhost/links', (err, db) =>{**

**console.log(err);**

**console.log(db);**

**app.get('/:title', async (req, res)=>{**

**let title = req.params.title**

**try{**

**let doc = await Link.find({title})**

**res.send(doc)**

**}**

**catch(err){**

**res.send(err)**

**}**

**})**

**})**

Esse código faz o conteudo da coleção aparecer na página ex: localhost.com/3000/instagram em formato JSON.

**app.get('/:title', async (req, res)=>{**

**let title = req.params.title**

**try{**

**let doc = await Link.findOne({title})**

**res.redirect(doc.url)**

**}**

**catch(err){**

**res.send(err)**

**}**

**})**

**})**

Esse código faz ser redirecionado ao link que tá no parametro link apresentado no Mongo.