The “Fruit” MongoDB Project

# NodeJS and MongoDB

* Primeiro, vamos configurar nossa passa dentro do Command Line

**touch app.js**

**npm init**

* Agora, vamos instalar os drives do mongodb

**Npm i mongodb**

* Agora, vamos confirgurar o app.js – não se esqueça de dar o run com *mongod* em uma nova aba do gitbash

**const MongoClient = require('mongodb').MongoClient;**

**const assert = require('assert');**

**// Connection URL**

**const url = 'mongodb://localhost:27017';**

**// Database Name**

**const dbName = 'fruitsDB';**

**// Use connect method to connect to the server**

**MongoClient.connect(url, function(err, client) {**

**assert.equal(null, err);**

**console.log("Connected successfully to server");**

**const db = client.db(dbName);**

**client.close();**

**});**

* ****Agora, vamos adicionar “data” para o app.js****

**const insertDocuments = function(db, callback) {**

**// Get the documents collection**

**const collection = db.collection('documents');**

**// Insert some documents**

**collection.insertMany([**

**{**

**name: "Apple",**

**score: 8,**

**review: "good"**

**},**

**{**

**name: "Orange",**

**score: 6,**

**review: "not bad"**

**},**

**{**

**name: "Banana",**

**score: 10,**

**review: "very good!"**

**}**

**], function(err, result) {**

**assert.equal(err, null);**

O assert verifica que não há erro, e que foram adds 03 itens

**assert.equal(3, result.result.n);**

**assert.equal(3, result.ops.length);**

**console.log("Inserted 3 documents into the collection");**

**callback(result);**

**});**

**}**

* ****Desta maneira, nosso código geral irá ficar assim:****

**//jshint: eversion:6**

**const MongoClient = require('mongodb').MongoClient;**

**const assert = require('assert');**

**// Connection URL**

**const url = 'mongodb://localhost:27017';**

**// Database Name**

**const dbName = 'fruitsDB';**

**// Create a newMongoClient**

**const client = new MongoClient(url, { useUnifiedTopology: true });**

**// Use connect method to connect to the server**

**MongoClient.connect(url, function(err, client) {**

**assert.equal(null, err);**

**console.log("Connected successfully to server");**

**const db = client.db(dbName);**

**insertDocuments(db, function(){**

**client.close();**

**});**

**});**

**const insertDocuments = function(db, callback) {**

**// Get the documents collection**

**const collection = db.collection('documents');**

**// Insert some documents**

**collection.insertMany([**

**{**

**name: "Apple",**

**score: 8,**

**review: "good"**

**},**

**{**

**name: "Orange",**

**score: 6,**

**review: "not bad"**

**},**

**{**

**name: "Banana",**

**score: 10,**

**review: "very good!"**

**}**

**], function(err, result) {**

**assert.equal(err, null); //verifica que não há erro**

**assert.equal(3, result.result.n); // verfica se foram adicionados 03 itens**

**assert.equal(3, result.ops.length);**

**console.log("Inserted 3 documents into the collection");**

**callback(result);**

**});**

**}**

sabe o melhor ? se abrir o ‘mongo’ em uma nova aba, e pedirmos o ‘show dbs’, nós iremos ver o o database

* ****Caso eu queira usar o node para dar o find, basta eu escrever:****

**//insertDocuments(db, function(){ });**

Importante: eu devo cancelar o insert, para não duplicar e chamar a função find()

**findDocuments(db, function(){**

**client.close();**

**})**

**});**

**// finding documents**

**const findDocuments = function(db, callback) {**

**// Get the documents collection**

**const collection = db.collection('fruits');**

**// Find some documents**

**collection.find({}).toArray(function(err, fruits) {**

**assert.equal(err, null);**

**console.log("Found the following records");**

**console.log(fruits)**

**callback(fruits);**

**});**

**}**

node app.js