Personal blog Challange !

# ****Configuração inicial****

* Configuração da pasta:

mongod

// open a new window:

cd desktop

cd toDoList-DB

npm install

npm init

atom .

**Neste momento nós temos os itens iniciais salvos como const, e os adicionais são salvos em uma array.**

* Neste módulo, vamos salvar os itens por mongoose, em uma dataBase

npm install mongod

npm install mongoose

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

****// const items = ["Buy Food", "Cook Food", "Eat Food"];****

****// const workItems = [];****

****mongoose.connect("mongodb://localhost:27017/toDoListDB", {useNewUrlParser: true});****

# Fixing the bug

Sometimes we are redirected before the save() method is done. With the callback method below we just call the “/” after the process is done.

**post.save(function(err){**

**if(!err) {**

**res.redirect("/");**

**}**

**});**

# Going to the post

## 3.1 To access the post, I can do like this:

**<% posts.forEach(function(post){ %>**

**<h1><%=post.title%></h1>**

**<p>**

**<%=post.content.substring(0, 100) + " ..."%>**

**</p>**

**<form class="" action="/postName" method="post">**

**<button type="submit" name="nomeDaPostagem" value="<%= post.\_id %>" >Read More</button>**

**</form>**

**<% }) %>**

**app.post("/postName", function(req, res) {**

**const listNameID = req.body.nomeDaPostagem;**

**console.log(listNameID);**

**Article.findById(listNameID, function(err, result){ //the result should be a DOCUMENT**

**console.log(result);**

**if(!err){**

**if(result){**

**res.render("post", {**

**title: result.title,**

**req.params**  
This property is an object containing properties mapped to the named route “parameters”. For example, if you have the route /user/:name, then the “name” property is available as req.params.name. This object defaults to {}.

source: <http://expressjs.com/en/4x/api.html#req.params>

**req.body**  
Contains key-value pairs of data submitted in the request body. By default, it is undefined, and is populated when you use body-parsing middleware such as body-parser and multer.

**content: result.content**

**});**

**}**

**}**

**else{**

**consle.log(err);**

**}**

**});**

**});**

## ****Or that…****

* In home.ejs you’ll need to change the href of the anchor tag based on the post id instead of the post name.

**<a href="/posts/<%=post.\_id%>">Read More</a>**

* In the app.post() method for the /post route, you should change the express route parameter to postId instead.

**app.get("/posts/:postId", function(req, res){**

**}**

* You’ll need a constant to store the postId parameter value

**const requestedPostId = req.params.postId;**

* You can use the findOne() method to find the post with a matching id in the posts collection.

**Post.findOne({\_id: requestedPostId}, function(err, post){ }**

* Once a matching post is found, you can render its title and content in the post.ejs page.

**Article.findOne({\_id: requestedPostId}, function(err, post){**

**res.render("post", {**

**title: post.title,**

**content: post.content**

**});**

**});**