



Quem se prepara, não para.

Arquitetura de Aplicações Web

5º período

Professora: Michelle Hanne

Sumário

- Criando API Node com Expresss - simples
- Transformando a API em um MVC com arquivo JSON

Instalações Necessárias

NodeJS

<https://nodejs.org/en/download/>

Express

Para instalar acesse o terminal e digite:

npm install -g express body-parser

Criando uma API Node com Express.js

Passo 1: Criando a pasta da API

Ainda no terminal, crie um diretório com o nome da sua aplicação e vá até o diretório criado. *Também poderá ser criada uma pasta no próprio Windows ou VS Code.*

```
C:\> Prompt de Comando

D:\testes_node>mkdir api-node

D:\testes_node>cd api-node

D:\testes_node\api-node>_
```

Passo 2: Criar arquivo app.js

Passo 3: Criar o arquivo package.json

```
npm init -y
```

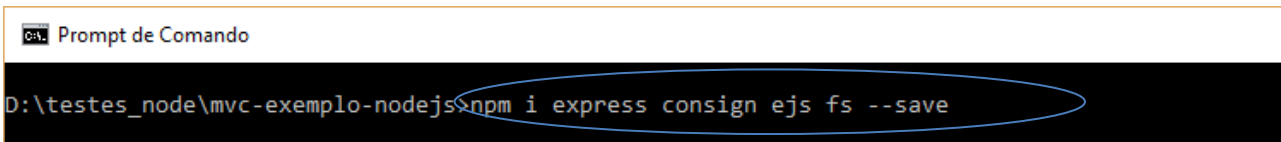
```
D:\testes_node\api-node>npm init -y
Wrote to D:\testes_node\api-node\package.json:

{
  "name": "api-node",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

Criando uma API Node com Express.js

Passo 4: Instalando as dependências que serão utilizadas no projeto

`npm i express consign ejs fs --save`



```
cmd Prompt de Comando
D:\testes_node\mvc-exemplo-nodejs>npm i express consign ejs fs --save
```

Passo 5: Alterando o conteúdo do arquivo `app.js`



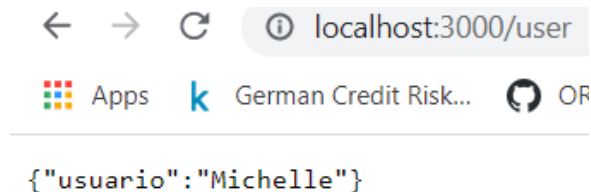
```
1  const express = require('express');
2  const app=express();
3
4  app.get('/user',(req,res) =>{
5    return res.json({usuario: 'Michelle'})
6  });
7
8  const server = app.listen(3000,() => {
9    const port = server.address().port;
10    console.log(`Servidor ativado - porta: ${port}`);
11  });
```

Criando uma API Node com Express.js

Passo 6: Executando o servidor

```
npm app.js
```

No navegador digite: **localhost:3000/user**



Criando uma API Node com Express.js

Passo 7: Instalando o Nodemon para atualizar o servidor automaticamente

```
npm install -g nodemon
```

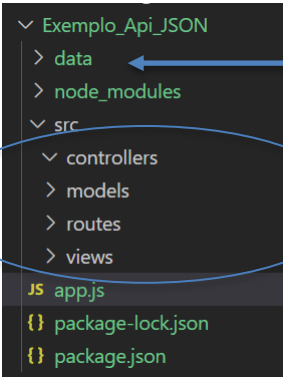
Passo 8: Alterando o arquivo package.json para aceitar o script de start no servidor

```
"start": "nodemon ./app.js"
```

```
1 {
2   "name": "exemplo_api_json",
3   "version": "1.0.0",
4   "description": "",
5   "main": "app.js",
6   "scripts": {
7     "start": "nodemon ./app.js",
8     "test": "echo \"Error: no test specified\" && exit 1"
9   },
10  "keywords": [],
11  "author": "",
12  "license": "ISC",
13  "dependencies": {
14    "consign": "^0.1.6",
15    "ejs": "^3.1.6",
16    "express": "^4.17.3",
17    "fs": "^0.0.1-security"
18  }
19 }
```


Alterando a API para MVC

Passo 9: Criando a estrutura de pastas:



Copie o arquivo JSON para esta pasta

Passo 10: Alterando o conteúdo do arquivo app.js

```
const express = require('express');
const app=express();
var consign = require('consign');

app.set('view engine', 'ejs');
app.set('views', './src/views');

consign()
  .include('src/routes')
  .then('src/models')
  .then('src/controllers')
  .into(app);

const server = app.listen(3000,() => {
  const port = server.address().port;
  console.log(`Servidor ativado - porta: ${port}`);
});
```

Alterando a API para MVC

Passo 11: Criando a rota (routes) - login.js:

```
module.exports = function(application){  
  application.get('/', function(req, res){  
    application.src.controllers.login.index(application, req, res);  
  });  
}
```

Passo 12: Criando o controllers - login.js

```
module.exports.index = function(application, req, res) {  
  var loginModel = new application.src.models.login();  
  
  loginModel.getLogin(function(err, result) {  
    res.render("login/index", {login : result});  
  });  
}
```

Alterando a API para MVC

Passo 13: Criando o model - login.js

```
var fs = require('fs');

function login() {}

login.prototype.getLogin = function(callback) {
  fs.readFile('./data/dados_login.json', 'utf8', function(err, result) {
    var data = [];

    if (!err) {

      var obj = JSON.parse(result);
      var i = 0; //(obj.pessoas.length - 1);

      obj.pessoas.forEach(function(pessoas) {
        if (i >= 0) {
          data[i] = pessoas;
          i++;
        }
      });
      callback(err, data);
    });
  });
};

module.exports = function(){
  return login;
}
```

Alterando a API para MVC

Passo 14: Criando o view (view/login) - index.ejs

```
<!doctype html>
<html lang="pt">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <title>API com MVC NODE</title>
    <link href="https://getbootstrap.com/docs/4.0/dist/css/bootstrap.min.css" rel="stylesheet">
  </head>
  <body>
    <div class="container">
      <br/><br/><br/>
      <h2>Exemplo API Node.JS com MVC</h2>
      <br/>
      <table class="table table-dark">
        <thead>
          <tr>
            <th scope="col">#</th>
            <th scope="col">Nome</th>
            <th scope="col">Email</th>
          </tr>
        </thead>
        <tbody>
          <% if (login.length > 0) { %>
            <% for(var i = 0; i < login.length; i++) {%>
              <tr>
                <th scope="row"><%= login[i].id %></th>
                <td><h4 ><%= login[i].nome %></h4></td>
                <td><%= login[i].email %></td>
              </tr>
            <% } %>
          <% } %>
        </tbody>
      </table>
    </div>
  </body>
</html>
```

Resultado da Aplicação

Passo 15: Executando: node app.js



Exemplo API Node.JS com MVC

#	Nome	Email
1	Layton Serjeantson	lserjeantson0@imageshack.us
2	Rusty Crowley	rcrowley1@phoca.cz
3	Stewart Secrett	ssecret2@privacy.gov.au
4	Oswell Nussey	onussey3@myspace.com
5	Erskine Jerram	ejerram4@gizmodo.com
6	Lillie Stennard	lstennard5@slate.com
7	Elladine Szanto	eszanto6@a8.net
8	Saunders Duffin	sduffin7@facebook.com
9	Cecil De Biasi	cde8@shareasale.com