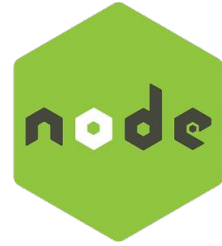


NodeJS



Prof. Celso Henrique Masotti

Você verá nesta aula:

Objetivo: usar o pacote de gerenciamento de servidor HTTP “nodemon”.

Nodemon

Ficar inicializando constantemente o servidor HTTP enquanto trabalhamos é muito cansativo e prejudica a produtividade. Para solucionar este problema fazemos uso de um pacote chamado “nodemon” que tem a finalidade reiniciar o servidor automaticamente sempre que ocorrer modificações de código nos arquivos.

Para instalar, o seu console deve estar no interior da pasta do projeto. Chame o “npm” colocando este código:

```
npm install nodemon -g
```

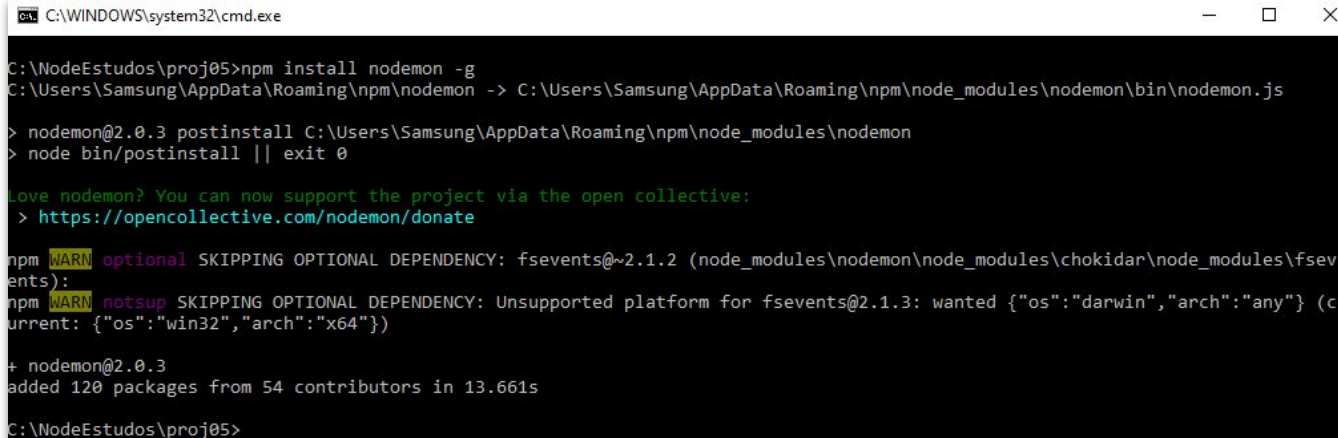
A flag “-g” indica que o pacote ficará ativo globalmente, alcançando todas as pastas do projeto, permitindo que o pacote, quando em atividade, possa perceber mudanças em qualquer código de qualquer arquivo.

Nodemon

Instalando o pacote:

```
C:\NodeEstudos\proj05>npm install nodemon -g
```

quando finalizado:



```
C:\WINDOWS\system32\cmd.exe

C:\NodeEstudos\proj05>npm install nodemon -g
C:\Users\Samsung\AppData\Roaming\npm\nodemon -> C:\Users\Samsung\AppData\Roaming\npm\node_modules\nodemon\bin\nodemon.js

> nodemon@2.0.3 postinstall C:\Users\Samsung\AppData\Roaming\npm\node_modules\nodemon
> node bin/postinstall || exit 0

Love nodemon? You can now support the project via the open collective:
> https://opencollective.com/nodemon/donate

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@~2.1.2 (node_modules\nodemon\node_modules\chokidar\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.3: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

+ nodemon@2.0.3
added 120 packages from 54 contributors in 13.661s

C:\NodeEstudos\proj05>
```

Nodemon

A partir de agora iremos inicializar o servidor HTTP não mais solicitando ao “node” como fazíamos, ou seja “node arquivo.js”, mas, sim, “nodemon arquivo.js”.

Vejamos! Temos o seguinte código em um arquivo chamado “index.js” que trabalha com o servidor HTTP do módulo “express”.

```
proj05 > JS index.js > ...
1  const express = require("express");
2  const app = express();
3
4  app.get("/",function(req,res){
5    res.send("Seja bem vindo ao app!");
6  });
7
8  app.listen(8080,function(){
9    // coloque o texto aqui
10    console.log("Servidor em funcionamento!");
11  });
```

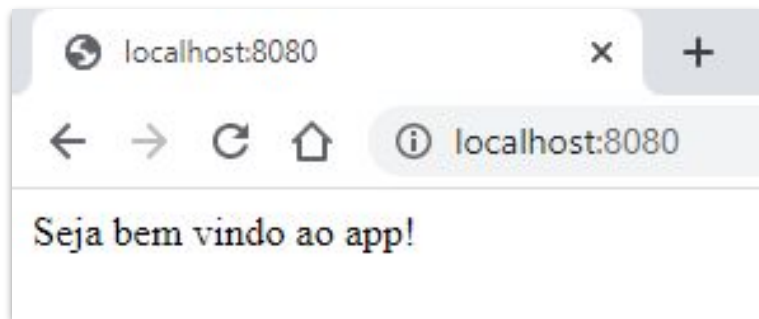
Nodemon

Quando temos o pacote “nodemon” instalado, devemos fazer uso dele para fazer a leitura do arquivo javascript, assim, no lugar de “node”, usaremos “nodemon”:

```
C:\NodeEstudos\proj05>nodemon index.js
[nodemon] 2.0.3
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
Servidor em funcionamento!
```

Nodemon

Veamos no browser o resultado através do servidor local e porta 8080 como está configurado no arquivo “index.js”:



Nodemon

Tudo funcionando perfeitamente.

Vamos alterar o texto que aparece no browser. Ao invés de “Seja bem vindo ao app!” vamos colocar “Hoje é quinta-feira!”.

```
proj05 > JS index.js > ...
1  const express = require("express");
2  const app = express();
3
4  app.get("/", function(req, res){
5    |   res.send("Hoje é quinta-feira!");
6  });
7
8  app.listen(8080, function(){
9    |   // coloque o texto aqui
10   |   console.log("Servidor em funcionamento!");
11  });
```


Nodemon

Observe o que aconteceu no console! O nodemon identificou alterações quando salvamos o arquivo e reiniciou o servidor HTTP automaticamente.

```
C:\NodeEstudos\proj05>nodemon index.js
[nodemon] 2.0.3
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
Servidor em funcionamento!
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
Servidor em funcionamento!
```



Obrigado

Profº Celso Henrique Masotti