

Valderi Leithardt, Dr.

IPW (Class six notes)

Summary

- Review of previous class *
- HTTP
- Pratica
- Exercises

Express é o framework Node mais popular e a biblioteca subjacente para uma série de outros frameworks do Node. O Express oferece soluções para:

- ✓ Gerir requisições de diferentes verbos HTTP em diferentes URLs;

--> Verbos HTTP: GET, POST, DELETE, PUT, PATCH

- ✓ Integrar "view engines" para inserir dados nos templates; Material adaptado conforme descrito [em MDN](#), 2023
- ✓ Definir as configurações comuns da aplicação web, como a porta a ser usada para conexão e a localização dos modelos que são usados para renderizar a resposta;
- ✓ Adicionar novos processos de requisição por meio de "middleware" em qualquer ponto da "fila" de requisições.

Tutoriais de instalação

<https://expressjs.com/en/starter/installing.html>

https://www.tutorialspoint.com/nodejs/nodejs_express_framework.htm

<https://github.com/programadriano/node-express>

<https://github.com/expressjs/express>

O npm é o Gestor de Pacotes do Node (Node Package Manager) que vem junto com ele e que é muito útil no desenvolvimento Node.

- É um repositório amplamente usado para a publicação de projetos Node.js de código aberto (open-source). Isso significa que ele é uma plataforma online onde qualquer pessoa pode publicar e compartilhar ferramentas escritas em JavaScript.
- O npm é uma ferramenta de linha de comando que ajuda a interagir com plataformas online, como navegadores e servidores. Essa utilidade auxilia na instalação e desinstalação de pacotes, gerenciamento da versões e gerenciamento de dependências necessárias para executar um projeto.
- Para utilizar os pacotes, o projeto deve conter um arquivo chamado de package.json. Dentro do pacote encontrará metadados específicos para os projetos.
- Os metadados ajudam a identificar o projeto e agem como uma base para que os usuários obtenham as informações.

Os metadados mostram alguns aspectos do projeto na seguinte ordem:

- 1) O nome do projeto;
- 2) A versão inicial;
- 3) A descrição;
- 4) O ponto de entrada;
- 5) Os comandos de teste;
- 6) O repositório git;
- 7) As palavras-chave;
- 8) A licença;
- 9) As dependências;
- 10) As dependências do desenvolvedor (devDependencies).

NPM

Identificar um projeto por meio de metadados:

- ✓ O nome é x,z,y-npm;
- ✓ A versão é 1.0.0;
- ✓ A descrição informa npm guide for beginner;
- ✓ O ponto de entrada do projeto ou o arquivo principal é beginner-npm.js;
- ✓ As palavras chave ou tags para encontrar o projeto no repositório são npm, example e basic;
- ✓ O autor do projeto é X,Y, X;
- ✓ Este projeto está licenciado sob o MIT;
- ✓ As dependências ou outros módulos que esse módulo usa são express 4.16.4.

```
• {  
•   "name": "xyz-npm",  
•   "version": "1.0.0",  
•   "description": "npm guide for beginner",  
•   "main": "beginner-npm.js",  
•   "scripts": {  
•     "test": "echo \"Error: no test specified\" && exit 1"  
•   },  
•   "keywords": [  
•     "npm",  
•     "example",  
•     "basic"  
•   ],  
•   "author": "X,Y, Z",  
•   "license": "MIT",  
•   "dependencies": {  
•     "express": "^4.16.4"  
•   }  
• }
```

HTTP

```
const express = require('express')

const produtos = [
  'Caneta',
  'lapis',
  'Eraser',
  'Cera',
  'Portatil'
]

const buscaProdutos = (produtoBuscado) => produtos.filter(produto => produtoBuscado === produto)

const server = express() /*servidor express*/

server.get('/produtos', (request, response) => {
  response.send(
    'Lista de Produtos:\n' +
    produtos.join('\n'))
})

server.get('/busca', (request, response) => {
  const produtosEncontrados = buscaProdutos(request.query.nomeProduto)

  response.send(
    'Lista de Produtos:\n' +
    produtosEncontrados.join('\n'))
})

server.listen(4000, () => console.log('Acesse localhost:4000'))
/*
npm install express (no terminal VSC)

No Browser http://localhost:4000/produtos
*/
```

TUTORIAIS

- Módulo fs do Node.js

<https://nodejs.reativa.dev/0046-node-module-fs/index>

- Promises API

<https://nodejs.org/api/fs.html>

- Fetch API

https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch

- Import

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/import>

Exercises

- 1º Trabalho part 2 disponibilizado em:

[https://github.com/isel-leic-ipw/2324i-IPW-LEIC31D/wiki/IPW IP-2324-1-A1](https://github.com/isel-leic-ipw/2324i-IPW-LEIC31D/wiki/IPW_IP-2324-1-A1)

Cap 18 – livro ENG: [https://eloquentjavascript.net/18 http.html](https://eloquentjavascript.net/18_http.html)

* Concluir exercícios das aulas anteriores.

References

- https://eloquentjavascript.net/18_http.html
- <https://www.tutorialspoint.com/expressjs/index.htm>
- <https://expressjs.com/en/starter/installing.html>
- https://www.w3schools.com/nodejs/nodejs_http.asp
- Também foram realizadas adaptações e modificações com base no material disponibilizado por Professor Luís Falcão, acesso online em: <https://github.com/isel-leic-ipw/>
- Aulas gravadas Professor Falcão:
 - <https://videoconf-colibri.zoom.us/rec/share/D9v4LepafqAXK6vaC6vyjL-K4LKWYJpOk-fdgQBbPkk1fcVRAMVjRVATa-AYVsQf.-h8JvXWsiWdq5PXq>
 - https://videoconf-colibri.zoom.us/rec/share/X_ZLYu1AL5xrc48PRcveMKqryq8fd9CdGhzJY7VVXAO4EGOs_VDGobtbTBk-F9TW.vh7Vn1a8dlK4Z59w

Valderi Leithardt, Dr.

Professor IPW

valderi.leithardt@isel.pt