



Estevan Junges



Apresentação

Universidade do Vale do Taquari - UNIVATES

Centro de Ciências Exatas e Tecnológicas - CETEC

12º Congresso de Ciência e Tecnologia do Vale do Taquari – CCTEC





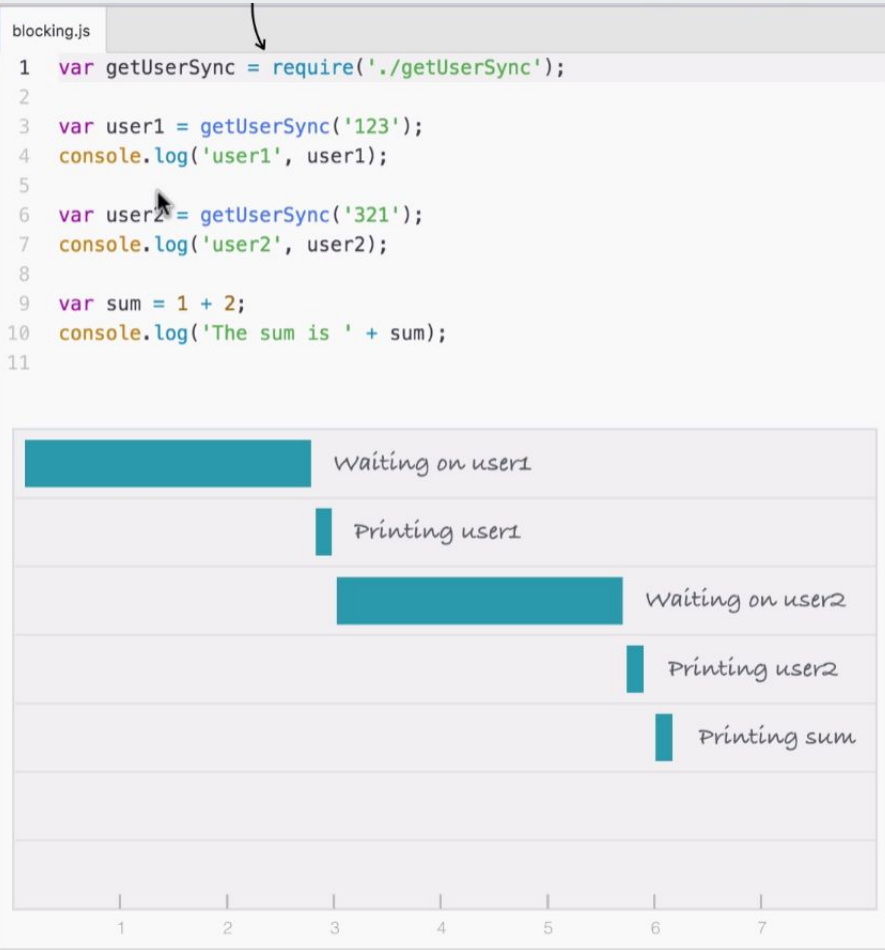
O que é Node.js?

Definição



“Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, npm, is the largest ecosystem of open source libraries in the world.”

<https://nodejs.org/>



non-blocking.js

```
1 var getUser = require('./getUser');
2
3 getUser('123', function (user1) {
4   console.log('user1', user1);
5 });
6
7 getUser('321', function (user2) {
8   console.log('user2', user2);
9 });
10
11 var sum = 1 + 2;
12 console.log('The sum is ' + sum);
13
```

Starting getUser for user1

Starting getUser for user2

Printing sum

Printing user1

Printing user1

1

2

3

4

5

6

7

Histórico do JavaScript



- Criado por Brendan Eich, em maio de 1995, com o objetivo de “embarcar” a linguagem Schema em browsers, a mando da Netscape
- Começou com o nome de Mocha, lançado originalmente com nome de LiveScript (setembro/95) e consolidado como JavaScript (dezembro/95)
- Ainda em 1995, a Netscape lançou a versão server-side. Ficou esquecida e voltou ao mercado em 2009 com o Node.js
- Desde 1996 é especificada pelo TC39 da Ecma (European Computer Manufacturers Association)

Principais características do JavaScript



- Interpretada
- Fracamente Tipada
- Orientada a Objetos
- Estruturada
- Funcional

Introdução ao JavaScript



- Var, let e const
- Controle de fluxo (if, else, for, while)
- Funções
- Data types
- JSON

Funções



```
function somar (a, b) {  
  return a + b;  
}  
  
//Anonymous function  
let somar = function (a, b) {  
  return a + b;  
}  
  
//Arrow function  
let somar = (a, b) => {  
  return a + b;  
}  
  
//Arrow function, uma linha  
let somar = (a,b) => a + b;
```

Data types em JavaScript



Primitivos:

- String
- Number
- Boolean
- Undefined

Não-Primitivos:

- Arrays
- Objetos
- RegExp
- function

typeof, === e !==

JSON



```
{  
  "status": "200",  
  "status_info": "OK",  
  "data": {  
    "users": [  
      {  
        "id": 1,  
        "name": "John"  
      },  
      {  
        "id": 2,  
        "name": "Mary"  
      }  
    ]  
  }  
}
```



**Node.js executa JavaScript no
lado do servidor**

npm - Node Package Manager



- Gerenciador de dependências do Node.js
- Instala pacotes localmente (projeto) e globalmente (máquina/usuário)
- Utiliza o arquivo *package.json* para gerenciar o seu projeto
- Os arquivos ficam no diretório *node_modules*
- Os pacotes são carregados com a função *require()*
- Equivalente a composer, pip, maven, entre outros...
- Yarn

<https://www.npmjs.com/>

Exemplo de *package.json*

```
{
  "name": "minicurso-node",
  "version": "1.0.0",
  "description": "Exemplo construido para o minicurso ministrado no CCTEC 2017",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node index"
  },
  "author": "Estevan Junges",
  "license": "ISC",
  "dependencies": {
    "axios": "^0.16.1",
    "body-parser": "^1.17.2",
    "dotenv": "^4.0.0",
    "express": "^4.15.3",
    "hbs": "^4.0.1",
    "moment": "^2.18.1"
  }
}
```

Atenção!



Nunca esqueça de adicionar a pasta *node_modules* em seu .gitignore!



Atividade Prática

Atividade Prática



O seu cliente pediu para desenvolver um sistema de vendas, que deve ser acessado tanto no navegador quanto em smartphone, e agora?

Vamos fazer uma API!

Legal, mas o que é uma API?

API

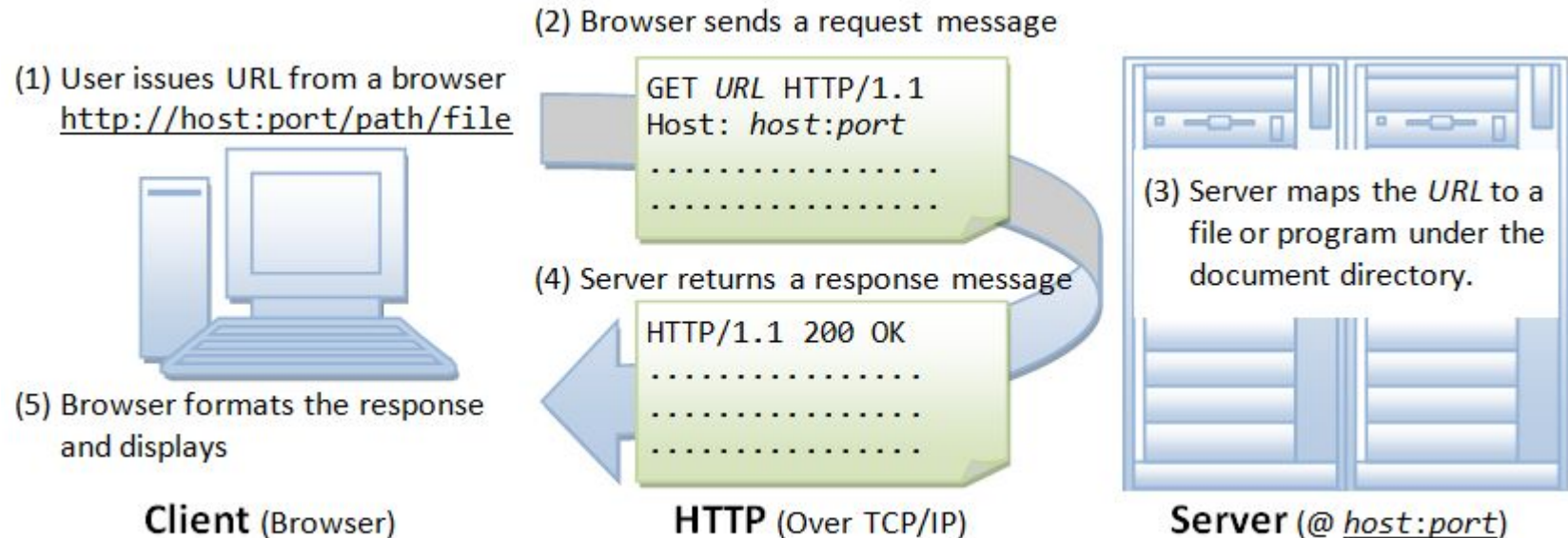


API é o acrônimo para *Application Programming Interface*. É uma interface da sua aplicação que é acessada de forma programática.

Qual é a vantagem? Você se preocupa somente com o *back-end* (regras de negócio e persistência)

Mas como fazer isso? Através do protocolo HTTP.

Protocollo HTTP



Protocolo HTTP



Métodos: GET, POST, PUT, HEAD, OPTIONS...

Status: **200** (OK), **404** (Not Found), **400** (Bad Request), **500** (Server Error) ...

Codificação: form-data, x-www-form-urlencoded, JSON...

Padrão REST



REST é o acrônimo para *Representational State Transfer*. É um padrão de organização dos endpoints de uma API para que ela fique organizada e semântica.

Padrão REST



Exemplo CRUD de usuarios:

- Listar usuarios: GET /usuarios
- Adicionar usuário: POST /usuarios
- Ver usuário: GET /usuarios/{id}
- Deletar usuário: DELETE /usuario/{id}
- Atualizar usuário: PUT (ou PATCH) /usuario/{id}

Postman



<https://www.getpostman.com/>

Descompactar e executar

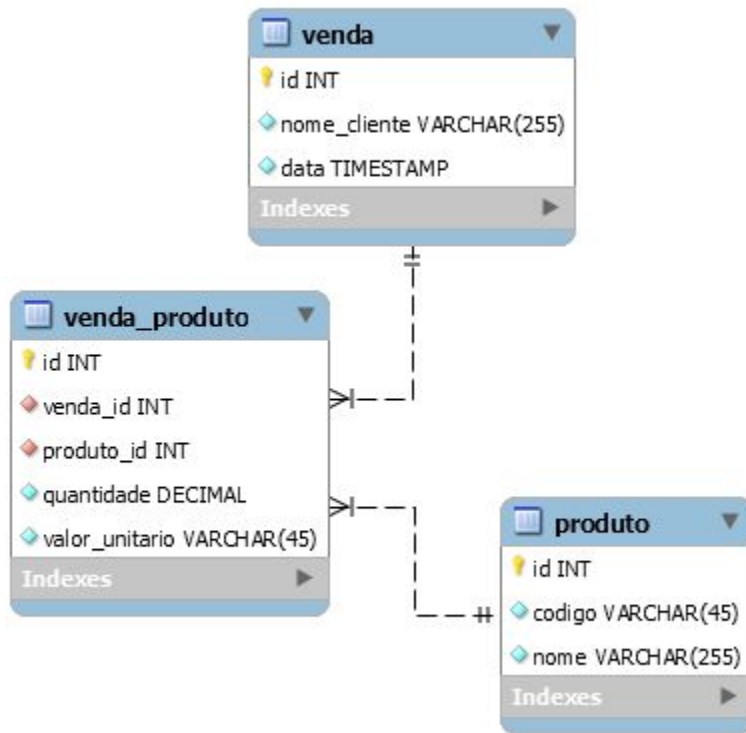
Atividade prática



Download do banco de dados SQLite:

<https://cctec-2018-nodejs.herokuapp.com/produtos>

Atividade prática



Atividade prática



Instalar Nodemon:

```
npm install --save-dev nodemon
```

Executar:

```
node_modules\.bin\nodemon index.js
```

Atividade prática



Instalar express e body-parser:

```
npm install --save express sqlite3 body-parser
```

Instalar SQLite:

```
npm install --save sqlite3
```

Versão final



<https://github.com/EstevanLJ/cctec-2018-nodejs>

Observação: essa versão possui algumas funcionalidades além das desenvolvidas no curso. Os slides do curso estão no repositório.



EXTRAS

Bibliotecas, bibliotecas e mais bibliotecas...



- lodash
- moment
- request
- axios
- body-parser
- dotenv
- ejs
- handlebars
- pug
- mocha
- yargs
- Webpack
- socket.io
- passport
- uuid
- and so on...

<https://www.npmjs.com/browse/depended>

Frameworks

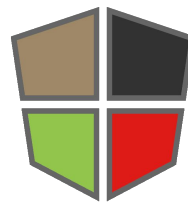


Server-side

- Express
- Koa
- Sails
- LoopBack

Full-stack

- Meteor
- MEAN
- Ember



<http://nodeframework.com/>

U The Complete Node.js D x

Seguro | https://www.udemy.com/the-complete-nodejs-developer-course-2/

Q ☆ 1

udemy

Categorias ▾

Q

Buscar Cursos

Udemy Para Empresas

Torne-se um Instrutor

Fazer login

Cadastre-se

Desenvolvimento

Negócios

TI e Softwares

Produtividade no escritório

Desenvolvimento Pessoal

Design

Marketing

Estilo de Vida

⋮

Use a Udemy no trabalho

Presentear Este Curso

Lista de Desejos

The Complete Node.js Developer Course (2nd Edition)

Learn Node.js by building real-world applications with Node, Express, MongoDB, Mocha, and more!

MAIS VENDIDO

★★★★★ 4,7 (5.814 classificações)

31.421 alunos inscritos

Criado por Andrew Mead, Rob Percival Última atualização em 6/2017 Inglês

Inglês [Auto-generated]

Visualizar este curso

Promoção da curiosidade

R\$20 ~~R\$235~~

91% de desconto

Só mais 3 dias por este preço!

Compre Agora

Adicionar ao carrinho

O que aprenderei?

Build, test, and launch Node apps

Create Express web servers and APIs

Store data with Mongoose and MongoDB

Use cutting-edge ES6/ES7 JavaScript

Deploy your Node apps to production

Create real-time web apps with SocketIO

[Visualizar grade curricular](#)

<https://www.udemy.com/the-complete-nodejs-developer-course-2/>

Curso Completo do Des

Seguro | https://www.udemy.com/curso-completo-do-desenvolvedor-nodejs/

🔍

1

udemy

Categorias

Buscar Cursos

Udemy Para Empresas

Torne-se um Instrutor

Fazer login

Cadastre-se

Desenvolvimento

Negócios

TI e Softwares

Produtividade no escritório

Desenvolvimento Pessoal

Design

Marketing

Estilo de Vida

⋮

Use a Udemy no trabalho

Presentear Este Curso

Lista de Desejos

Curso Completo do Desenvolvedor NodeJS e MongoDB

Aprenda a criar sites e sistemas web utilizando a plataforma NodeJS e o banco de dados MongoDB

★★★★★ 4,5 (1.384 classificações) 5.910 alunos inscritos

Criado por Jorge Sant Ana Última atualização em 8/2017 Português

Visualizar este curso

Promoção da curiosidade

R\$20 ~~R\$160~~

88% de desconto

🕒 Só mais 3 dias por este preço!

Compre Agora

Adicionar ao carrinho

O que aprenderei?

✓ Aprenda a criar aplicações NodeJS integradas com o banco de dados MySQL e MongoDB utilizando Design Pattern MVC.

✓ Aprenda a utilizar o framework Express, o gerenciador de pacotes NPM e o engine

✓ Aprenda a utilizar o JavaScript no back-end como linguagem de programação para páginas dinâmicas.

✓ Aprenda a usar os módulos Consign, Body-parser, Express-validator, Express-

<https://www.udemy.com/curso-completo-do-desenvolvedor-nodejs/>

Dúvidas? Stackoverflow





Referências

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

<https://nodejs.org/en/docs/guides/blocking-vs-non-blocking/>

https://developer.mozilla.org/pt-PT/docs/Web/JavaScript/Guia/Introdu%C3%A7%C3%A3o_ao_JavaScript

<https://www.udemy.com/the-complete-nodejs-developer-course-2/>

<https://becode.com.br/o-que-e-api-rest-e-restful/>