

Programação para Internet

Ambiente


Objetivos

- Compreender o funcionamento do NODEJS
- Entender sobre o gerenciador de pacotes NPM
- Configurar uma pasta com os pacotes (locais) para uso de uma aplicação web simples
- Compreender os elementos de uma aplicação web
- Desenvolver o exemplo

Node.js



Nodejs

Node.js é um ambiente de desenvolvimento (plataforma)  que executa Javascript no lado do servidor. Construída no Javascript (Engine do Google Chrome), de código aberto, multiplataforma, que executa JavaScript. É usado por milhares de desenvolvedores em todo o mundo.

Nodejs



Com ele é possível desenvolver aplicativos:

- Web intensivos;
- aplicativos de página única (Single Page Application - SPA)
- outros aplicativos web

Baixe os programas

Nodejs

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

Vs Code

<https://code.visualstudio.com/Download>

Exemplo – Olá mundo!



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

http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Olá mundo!');
}).listen(8080);
```


- O módulo http é carregado e retorna um objeto Javascript (http), que permite executar as funcionalidades disponíveis no módulo.
- A função createServer() executa um programa servidor que escuta no porto 8080.
- A função callback anónima function(req, res) é invocada sempre que um pedido é recebido pelo servidor. Contém 2 parâmetros:
 - req (request) → objeto que contém toda a informação do pedido
 - res (response) → objeto que devemos preencher com a resposta a enviar.

Exemplo – Aplicação estática!

Carrega páginas estáticas

```
<!DOCTYPE html>
<html>
<body>
  <h2>PPI</h2>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>
  <h2>Multivix</h2>
</body>
</html>
```



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

http.createServer(function (req, res) {
  var q = url.parse(req.url, true);
  var filename = "." + q.pathname;
  fs.readFile(filename, function(err, data) {
    if (err) {
      res.writeHead(404, {'Content-Type': 'text/html'});
      return res.end("404 Not Found");
    }
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.write(data);
    return res.end();
  });
}).listen(8080);
```


Gerenciador de Pacotes



Gerenciador de Pacotes

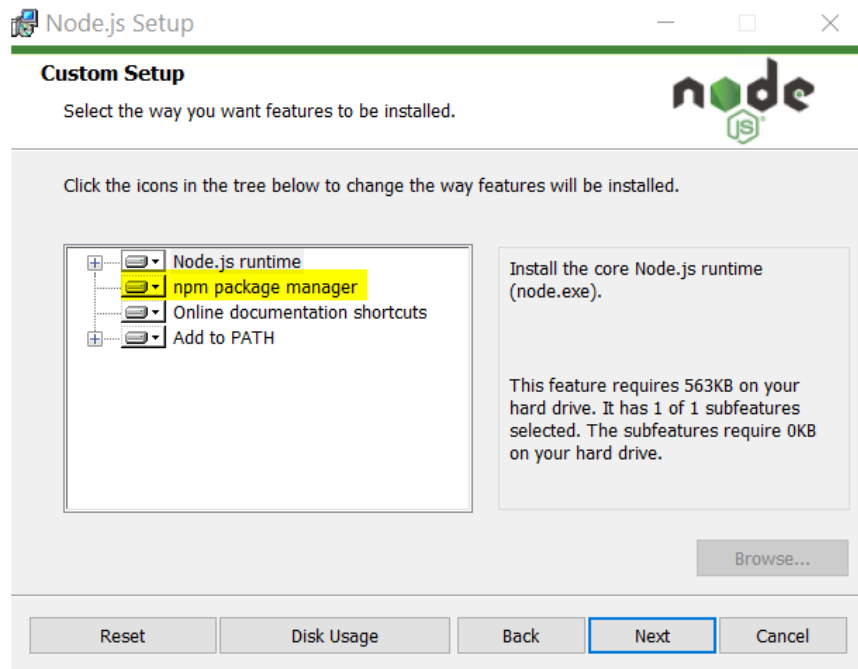


NPM (*Node Package Manager*) é um gerenciador de pacotes do Node.js para administrar as bibliotecas e frameworks utilizados em uma aplicação.

Com ele é possível instalar, desinstalar e atualizar dependências em uma aplicação por meio de uma simples instrução na linha de comando.

Gerenciador de Pacotes

O NPM é distribuído junto com o Node.js



Gerenciador de Pacotes



Instalações de pacotes

Local

Para cada projeto criado é criado (automaticamente ou não) um arquivo chamado `package.json`. Assim é possível que a própria aplicação contenha seus pacotes desacoplado do computador.

Global

As instalações globais podem (e geralmente são) utilizadas por todas as aplicações.

Gerenciador de Pacotes



Comandos

NPM OPCAÇÃO PACOTE --PARAMETRO

Exemplo:

```
npm install nodemon  
npm update nodemon  
npm uninstall nodemon
```

Parâmetros:

--save: Salva o pacote somente para a aplicação

Express



Express

O Express é um framework Node.js para aplicações web. O mais utilizado para aplicações gerais.



A aplicação ficará estruturada desta forma:

>> views

>> views >> layouts

app.js

Express

O Express é um framework Node.js para aplicações web. O mais utilizado para aplicações gerais.



A aplicação ficará estruturada desta forma:

```
>> views →  
>> views >> layouts  
app.js
```

São os templates de página HTML

Express

O Express é um framework Node.js para aplicações web. O mais  utilizado para aplicações gerais.

A aplicação ficará estruturada desta forma:

>> views

>> views >> layouts →
app.js

Parte geral da página

Express

O Express é um framework Node.js para aplicações web. O mais utilizado para aplicações gerais.



A aplicação ficará estruturada desta forma:

>> views

>> views >> layouts

app.js →

- Configurações da aplicação
- Uso e importações de Rotas
- Diretórios estáticos
- Inicialização da aplicação

Exemplo para o protocolo HTTP



Exemplo HTTP

1. Com o vscode aberto, selecione o menu File → Open Folder
2. Selecione uma pasta para criar este projeto
3. Com a pasta aberta pelo vscode, selecione o menu Terminal → New Terminal
4. Instale os pacotes, conforme apresentado abaixo:



O Express é um framework Node.js para aplicações web. O mais utilizado para aplicações gerais.

```
npm install express --save
npm install express-handlebars --save
npm install body-parser --save
npm install express-session --save
```

Exemplo HTTP

5. Crie um novo arquivo chamado appHttp.js

1 de 3

```
const express = require("express");
const handlebars = require("express-handlebars");
const bodyParser = require('body-parser')
const session = require('express-session');

const app = express();

app.engine('handlebars', handlebars() );
app.set('view engine', 'handlebars');
app.use(bodyParser.urlencoded({extended: false}));
app.use(bodyParser.json());
app.use(session({secret: 'supersenha',saveUninitialized: true,resave: true}));

// aqui temos os JS / CSS / HTML estáticos
app.use(express.static('public'));
```



Exemplo HTTP

5. Crie um novo arquivo chamado appHttp.js

2 de 3

```
// rotas da aplicação
app.get('/login', function(request, response) {
  request.session.destroy();
  response.render('login');
});
app.post('/entrar', function(request, response) {
  let usuario = request.body.usuario;
  let senha = request.body.senha;
  if (!(usuario == "bruno" && senha == "multivix")) {

    } else {
      request.session.usuario = usuario;
      response.redirect('/home');
    }
  });
});
```



Exemplo HTTP

5. Crie um novo arquivo chamado appHttp.js

3 de 3

```
app.get('/home', function(request, response) {  
  if (request.session.usuario && request.session.usuario != null)  
    response.render('home', { nomeUsuario: request.session.usuario });  
  else  
    response.status(401).send("Usuário não autenticado");  
});  
  
//----- inicia a aplicação  
app.listen(9977, function() {  
  console.log("Servidor executando")  
});
```



Exemplo HTTP

6. Crie o arquivo views → layouts → main.handlebars



```
<html>
<head>
  <title>Exemplo protocolo HTTP</title>
</head>
<body>
  {{{body}}}
</body>
</html>
```


Exemplo HTTP

7. Crie o arquivo views → home.handlebars



```
<h1>Pagina autenticada</h1>  
<hr>  
<br>  
<h2>Bem vindo {{nomeUsuario}}</h2>  
<br>  
<a href='/login'>Sair</a>
```

Exemplo HTTP

8. Crie o arquivo views → home.handlebars



```
<h2>Entrar no sistema</h2>
<hr>
<form action="/entrar" method="post" role="form">
  <table>
    <tr><td colspan="2" align="center">Informe as credenciais</td></tr>
    <tr>
      <td>Login:</td>
      <td><input type="text" name="usuario"></td>
    </tr>
    <tr>
      <td>Senha:</td>
      <td><input type="password" name="senha"></td>
    </tr>
    <tr>
      <td colspan="2">
        <input type="submit" value="Entrar">
      </td>
    </tr>
  </table>
</form>
```

