



PUC Minas

# Plataforma Node.js

## Módulos internos do Node.js (built-in)

# Principais Módulos Nativos do Node.js

Módulo	Descrição
events	Permite criar e manipular de eventos
http	Permite criar servidores Web
dns	Permite resolver nomes da Internet
path	Permite manipular caminhos de arquivos
os	Provê informações sobre o Sistema Operacional
process	Provê informação sobre o processo Node.js em execução
net	Permite criar servidores e clientes
url	Interpreta strings de URLs
stream	Manipula dados em fluxos

# Módulo events – Classe EventEmitter

Método	Descrição
addListener (event, listener)	Inclui um listener na fila para o evento passado como parâmetro
on (event, listener)	Inclui um listener na fila para o evento passado como parâmetro
once (event, listener)	Inclui um listener de um único disparo na fila para o evento passado como parâmetro
removeListener (event, listener)	Remove o listener da fila do evento passado como parâmetro
removeAllListeners (event)	Remove todos os listeners para o evento passado como parâmetro
listeners (event)	Retorna um array com os listeners ativos para o evento passado como parâmetro
emit (event)	Executa os listeners ativos para o evento passado como parâmetro

# Módulo events

```
// Importa o módulo eventos
import events from 'events'

// Cria um gerenciado de eventos
const eventMgmt = new events.EventEmitter ();
eventMgmt.on('bomdia', (data) => {
    console.log(`Recebi um bom dia de: ${data}`)
})

// Inclui um subscriber ao evento
eventMgmt.addListener ('bomdia', (data) => {
    console.log(`Aconteceu de ${data}`);
})

// Dispara o evento
eventMgmt.emit ('bomdia', 'Rommel')
```

# Módulo process

```
// Apresenta variáveis de ambiente
console.log ('Variáveis de ambiente ----- \n', process.env)

// Apresenta o diretório de trabalho
console.log ('Diretório de trabalho ----- \n', process.cwd())

// Apresenta o diretório de execução
console.log ('Diretório de execução ----- \n', __dirname)

// Apresenta o arquivo (entrypoint) de execução
console.log ('Arquivo de execução ----- \n', __filename)

// Apresenta o processo
console.log ('Processo ----- \n', process)
```

# Módulo http

```
// Importa o módulo do HTTP
const http = require('http');

// Cria um servidor e atribui uma callback de processamento da requisição
const server = http.createServer((req, res) => {
  res.statusCode = 200; // Retorno OK
  res.setHeader('Content-Type', 'text/html');
  res.end('Hello World');
});

// Define parâmetros (hostname e porta) e inicia o servidor
const hostname = '127.0.0.1';
const port = 3000;
server.listen(port, hostname, () => {
  console.log(`Servidor rodando http://${hostname}:${port}/`);
});
```

# Módulo http

```
// Importa o módulo do HTTP
const http = require('http');

// Cria um servidor e atribui uma callback de processamento da requisição
const server = http.createServer((req, res) => {
  res.statusCode = 200; // Retorno OK
  res.setHeader('Content-Type', 'application/json');
  res.end('{ "user": "Rommel Carneiro", "empresa": "PUC Minas" }');
});

// Define parâmetros (hostname e porta) e inicia o servidor
const hostname = '127.0.0.1';
const port = 3000;
server.listen(port, hostname, () => {
  console.log(`Servidor rodando: http://${hostname}:${port}/`);
});
```

# Módulo http

```
import http from 'http';

const server = http.createServer((req, res) => {
  if (req.method === "GET") {
    res.statusCode = 200;
    res.setHeader('Content-Type', 'text/html');
    res.end(`<form method="POST">Nome: <input type="text" name="nome"></form>`);
  }
  else {
    let str = ''
    req.on('data', function (chunk) { str += chunk; });
    req.on('end', function () {
      let nome = str.split('=')[1];
      res.statusCode = 200;
      res.setHeader('Content-Type', 'application/json');
      res.end(`{ "username": "${nome}" }`);
    });
  }
});

server.listen (3000, () => { console.log(` Servidor rodando`); });
```



# Módulo dns

```
const { Resolver } = require('dns');
const resolver = new Resolver();

// Define o servidor a ser utilizado
resolver.setServers(['8.8.8.8']);

// Realiza a tradução de um nome de domínio para um endereço IP
resolver.resolve4('pucminas.br', (err, addresses) => {
  if (err)
    console.log(`Erro ao traduzir: ${err.message}`);
  else
    console.log('Endereço IP: ' + addresses[0]);
});
```