

Aula 03: Desenvolvimento Web

Sumário

1. Introdução ao módulo HTTP
2. Introdução ao express
3. Exercícios

Introdução ao Módulo HTTP do Node.js

O que é o Módulo HTTP?

O módulo `http` do Node.js permite criar servidores web simples e manipular requisições e respostas HTTP. Ele é um dos módulos principais do Node.js e não requer instalação adicional.

Como o Módulo HTTP Funciona?

O módulo `http` cria servidores que escutam requisições em uma porta específica e respondem a essas requisições com base na lógica definida pelo desenvolvedor. Ele permite manipular diretamente as requisições HTTP (GET, POST, PUT, DELETE, etc.), cabeçalhos HTTP, e corpo das mensagens.

Exemplo Prático 1: Criando um Servidor HTTP Simples

Este exemplo cria um servidor HTTP básico que responde "Hello, World!" para qualquer requisição.

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

// Criar um servidor
const server = http.createServer((req, res) => {
  res.statusCode = 200; // Status de sucesso
```

```

    res.setHeader('Content-Type', 'text/plain'); // Tipo de c
    res.end('Hello, World!\n'); // Resposta
  });

  // Servidor escutando na porta 3000
  server.listen(3000, '127.0.0.1', () => {
    console.log('Servidor rodando em http://127.0.0.1:3000/')
  });

```

Como funciona:

- `http.createServer()`: Cria o servidor.
- `req` (request) contém as informações sobre a requisição.
- `res` (response) é usado para enviar a resposta ao cliente.
- O servidor é configurado para escutar na porta 3000.

Exemplo Prático 2: Manipulando Rotas no Servidor HTTP

Este exemplo expande o servidor anterior para responder de maneira diferente dependendo da URL acessada.

```

const http = require('http');

const server = http.createServer((req, res) => {
  if (req.url === '/') {
    res.statusCode = 200;
    res.setHeader('Content-Type', 'text/plain');
    res.end('Você está na página inicial!\n');
  } else if (req.url === '/sobre') {
    res.statusCode = 200;
    res.setHeader('Content-Type', 'text/plain');
    res.end('Esta é a página sobre.\n');
  } else {

```

```
        res.statusCode = 404;
        res.setHeader('Content-Type', 'text/plain');
        res.end('Página não encontrada.\n');
    }
});

server.listen(3000, '127.0.0.1', () => {
    console.log('Servidor rodando em http://127.0.0.1:3000/')
});
```

Como funciona:

- `req.url` é usado para identificar a rota acessada pelo usuário.
- O servidor responde com diferentes mensagens dependendo da URL.
- Uma mensagem de erro 404 é retornada para URLs que não são reconhecidas.

Introdução ao Express.js

O que é o Express?

Express é um framework web minimalista para Node.js que facilita a criação de servidores web e APIs. Ele fornece uma camada de abstração sobre o módulo `http` do Node.js, tornando o código mais simples e fácil de ler.

Por que Usar o Express?

- **Simplicidade:** Torna a criação de rotas, middleware e resposta a requisições mais intuitiva.
- **Flexibilidade:** Pode ser usado para construir aplicações desde pequenas APIs até grandes sistemas web.
- **Comunidade e Ecossistema:** Possui uma grande comunidade e uma vasta coleção de middlewares e plugins.

Exemplo Prático 1: Criando um Servidor com Express

Este exemplo cria um servidor simples usando Express que responde "Hello, World!".

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

// Rota para a página inicial
app.get('/', (req, res) => {
  res.send('Hello, World!');
});

// Servidor escutando na porta 3000
app.listen(3000, () => {
  console.log('Servidor Express rodando em http://localhost');
});
```

Como funciona:

- `express()` cria uma aplicação Express.
- `app.get('/', ...)` define uma rota para a página inicial.
- `res.send()` envia uma resposta ao cliente.

Exemplo Prático 2: Criando Rotas Dinâmicas com Express

Este exemplo mostra como criar rotas dinâmicas que respondem a diferentes URLs.

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

// Rota para a página inicial
app.get('/', (req, res) => {
  res.send('Você está na página inicial!');
});
```

```
// Rota para a página sobre
app.get('/sobre', (req, res) => {
  res.send('Esta é a página sobre.');
```



```
// Rota dinâmica que responde com o nome do usuário
app.get('/usuario/:nome', (req, res) => {
  res.send(`Olá, ${req.params.nome}!`);
});
```



```
// Servidor escutando na porta 3000
app.listen(3000, () => {
  console.log('Servidor Express rodando em http://localhost');
```

Exercícios

Exercício 1: Criando um Servidor HTTP com Rotas (Demonstração)

Objetivo: Criar um servidor HTTP simples que responda a diferentes URLs com mensagens específicas.

Instruções:

1. Crie um servidor HTTP usando o módulo `http`.
2. O servidor deve responder com as seguintes mensagens:
 - **Página Inicial (/):** "Bem-vindo à Página Inicial!"
 - **Página Sobre (/sobre):** "Esta é a Página Sobre."
 - **Página Contato (/contato):** "Entre em contato conosco!"
 - **Qualquer outra rota:** "404 - Página não encontrada"

Código Esperado:

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

const server = http.createServer((req, res) => {
  if (req.url === '/') {
    res.statusCode = 200;
    res.setHeader('Content-Type', 'text/plain');
    res.end('Bem-vindo à Página Inicial!\n');
  } else if (req.url === '/sobre') {
    res.statusCode = 200;
    res.setHeader('Content-Type', 'text/plain');
    res.end('Esta é a Página Sobre.\n');
  } else if (req.url === '/contato') {
    res.statusCode = 200;
    res.setHeader('Content-Type', 'text/plain');
    res.end('Entre em contato conosco!\n');
  } else {
    res.statusCode = 404;
    res.setHeader('Content-Type', 'text/plain');
    res.end('404 - Página não encontrada\n');
  }
});

server.listen(3000, '127.0.0.1', () => {
  console.log('Servidor rodando em http://127.0.0.1:3000/');
});
```

Exercício 2: Criando um Servidor Express com Rotas Dinâmicas (Prática)

Objetivo: Criar um servidor Express que responda a diferentes URLs, incluindo uma rota dinâmica que retorna a idade do usuário.

Instruções:

1. Crie um servidor Express.
2. O servidor deve ter as seguintes rotas:

- **Página Inicial (/)**: "Bem-vindo ao Express!"
- **Página de Produtos (/produtos)**: "Aqui estão nossos produtos."
- **Rota Dinâmica para Usuário (/usuario/:nome)**: "Olá, [nome do usuário]!"
- **Rota Dinâmica para Idade (/idade/:ano)**: Responda com a idade do usuário calculada com base no ano de nascimento informado.
(Exemplo: Se o usuário acessar /idade/1990, o servidor deve responder "Sua idade é 34 anos!")

Dicas:

- Utilize `req.params` para capturar os parâmetros dinâmicos.
- Use a função `new Date().getFullYear()` para obter o ano atual.

Código Esperado:

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

app.get('/', (req, res) => {
  res.send('Bem-vindo ao Express!');
});

app.get('/produtos', (req, res) => {
  res.send('Aqui estão nossos produtos.');
```

```
});

app.get('/usuario/:nome', (req, res) => {
  res.send(`Olá, ${req.params.nome}!`);
});

app.get('/idade/:ano', (req, res) => {
  const anoAtual = new Date().getFullYear();
  const idade = anoAtual - parseInt(req.params.ano);
  res.send(`Sua idade é ${idade} anos!`);
});

app.listen(3000, () => {
```

```
console.log('Servidor Express rodando em http://localhost  
');
```