

Introdução ao Express

Express.js é um framework de aplicação Web minimalista e flexível para Node.js, que fornece um conjunto robusto de recursos para desenvolver aplicações Web e móveis, facilitando a criação e o gerenciamento de servidores HTTP e simplificando o processo de desenvolvimento comparado ao uso de Node.js puro.

Vantagens do Express

1. **Facilidade de Uso:** Simplifica a criação de servidores e rotas.
2. **Middleware:** Permite adicionar funcionalidades extras, como autenticação e manipulação de requisições/respostas.
3. **Roteamento:** Fornece uma maneira clara e eficiente de definir rotas.
4. **Ecossistema:** Grande quantidade de bibliotecas e plugins disponíveis.
5. **Performance:** Leve e rápido, ideal para aplicações de alto desempenho.

Uso do Express

```
npm install express
```

Criando um Servidor Básico

```
const express = require('express');
const app = express();
const porta = 3000;

app.get('/', (req, res) => {
  res.send('Hello World, Express!');
});

app.listen(porta, () => {
  console.log(`Servidor rodando em http://localhost:${porta}`);
});
```

Para rodar o servidor:

```
node app.js
```

Roteamento com Express

Express facilita a definição de rotas.

```
const express = require('express');
const app = express();
const port = 3000;

app.get('/', (req, res) => {
  res.send('Hello World from Express!');
});

app.get('/about', (req, res) => {
  res.send('About Page');
});

app.post('/submit', (req, res) => {
  res.send('Data submitted');
});

app.put('/update', (req, res) => {
  res.send('Data updated');
});

app.delete('/delete', (req, res) => {
  res.send('Data deleted');
});

app.listen(port, () => {
  console.log(`Servidor rodando em http://localhost:${port}`);
});
```

Middleware

Middleware são funções que têm acesso ao objeto de solicitação (**req**), ao objeto de resposta (**res**) e à próxima função middleware no ciclo de solicitação/resposta.

```

const express = require('express');
const app = express();
const port = 3000;

// Middleware de logging
app.use((req, res, next) => {
  console.log(`${req.method} ${req.url}`);
  next(); // Chama a próxima função middleware no stack
});

app.get('/', (req, res) => {
  res.send('Hello World with Middleware!');
});

app.listen(port, () => {
  console.log(`Servidor rodando em http://localhost:\${port}`);
});

```

Trabalhando com JSON

Express facilita o trabalho com JSON, tanto na leitura quanto na escrita.

```

const express = require('express');
const app = express();
const port = 3000;

app.use(express.json()); // Middleware para parsear JSON

app.post('/data', (req, res) => {
  const data = req.body;
  res.json({
    message: 'Data received',
    data: data
  });
});

app.listen(port, () => {
  console.log(`Servidor rodando em http://localhost:\${port}`);
});

```