

Daniel R. Medeiros



## **A Jornada do Jedi Developer: Node.js e Express**

# 1. Instalando o Express

Para começar, certifique-se de que o Node.js e o npm estão instalados no seu sistema. Você pode verificar com os comandos:

```
bash
```

[Copiar código](#)

```
node -v
```

```
npm -v
```

Se estiver tudo certo, crie uma nova pasta para o projeto e inicie o npm:

```
bash
```

[Copiar código](#)

```
mkdir meu-projeto
```

```
cd meu-projeto
```

```
npm init -y
```

Agora, instale o Express:

```
bash
```


[Copiar código](#)

```
npm install express
```

## 2. Configurando o Servidor com Express

Crie um arquivo chamado `server.js` e configure um servidor básico com o Express:

javascript

 Copiar código

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


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

const PORT = 3000;
app.listen(PORT, () => {
  console.log(`Servidor rodando em http://localhost:${PORT}`);
});
```

### Como executar o servidor

No terminal, use o comando:

bash

 Copiar código

```
node server.js
```

Abra o navegador e acesse **`http://localhost:3000`**. Você verá a mensagem “Olá, mundo! Bem-vindo ao Express!”

# 3. Criando uma API Simples com Express

Vamos criar uma API que gerencia uma lista de tarefas (To-Do List).

## Habilitando o JSON no Express:

1 - Adicione o middleware para manipular requisições JSON:

javascript

 Copiar código

```
app.use(express.json());
```

## Roteando as requisições:

2 -Adicione rotas para listar, adicionar e deletar tarefas:

javascript

 Copiar código

```
let tarefas = [];  
  
// Listar tarefas  
app.get('/tarefas', (req, res) => {  
  res.json(tarefas);  
});  
  
// Adicionar uma nova tarefa  
app.post('/tarefas', (req, res) => {  
  const { titulo } = req.body;  
  if (!titulo) {  
    return res.status(400).json({ erro: 'Título é obrigatório!' });  
  }  
  const novaTarefa = { id: Date.now(), titulo };  
  tarefas.push(novaTarefa);  
  res.status(201).json(novaTarefa);  
});  
  
// Deletar uma tarefa  
app.delete('/tarefas/:id', (req, res) => {  
  const { id } = req.params;  
  tarefas = tarefas.filter((tarefa) => tarefa.id !== parseInt(id));  
  res.json({ mensagem: 'Tarefa removida com sucesso!' });  
});
```

### Testando a API:

Use ferramentas como [Postman](#) ou [Insomnia](#) para testar os endpoints:

- GET /tarefas: Lista todas as tarefas.
- POST /tarefas: Adiciona uma nova tarefa (envie um JSON com { "titulo": "Minha tarefa" }).
- DELETE /tarefas/:id: Deleta uma tarefa pelo ID.

## 4. Conclusão

Com o Express, você pode criar APIs rapidamente e com simplicidade. Este exemplo básico já cobre os fundamentos para manipular dados e criar rotas.

Agora é sua vez: expanda essa API, adicionando novos recursos como autenticação, validação de dados ou até mesmo integração com um banco de dados como MongoDB.

Happy coding!