

Axios, JWT e Nodemon

Axios

Biblioteca baseada em Promises para fazer requisições HTTP no navegador e no Node.js. Utilizada para realizar operações como GET, POST, PUT, DELETE, entre outras, facilitando a comunicação entre o Front-End e o Back-End.

- **Facilidade de uso:** Simplifica o código para fazer requisições HTTP comparado com a API Fetch nativa do JavaScript.
- **Interceptores:** Permite adicionar interceptores para tratar requisições ou respostas antes que elas sejam manipuladas pelo `.then` ou `.catch`.
- **Configurações globais:** Permite configurar valores padrão para todas as requisições, como cabeçalhos e a URL base.
- **Cancelamento de requisições:** Suporta o cancelamento de requisições usando tokens de cancelamento.

Instalação: `npm install axios`

```
import axios from 'axios';

// Fazendo uma requisição GET
axios.get('https://jsonplaceholder.typicode.com/posts/1')
  .then(response => {
    console.log(response.data);
  })
  .catch(error => {
    console.error('Erro ao fazer a requisição GET:', error);
  });
```

JWT (JSON Web Token)

JWT é um padrão aberto (RFC 7519) para criar *tokens* de acesso que permitem a verificação segura entre duas partes: o **cliente** e o **servidor**. É amplamente utilizado para autenticação e troca de informações.

- **Autenticação:** JWTs são frequentemente usados para autenticar usuários em sistemas Web. O servidor gera um token JWT que é enviado ao cliente, o qual o armazena (normalmente em `localStorage` ou `cookies`). O cliente inclui este token em futuras requisições ao servidor para provar sua identidade.

- **Troca de Informações:** JWTs podem transportar informações entre partes de forma segura, pois são assinados digitalmente usando um segredo (com HMAC) ou um par de chaves pública/privada (com RSA).

Instalação: `npm install jsonwebtoken`

```
import jwt from 'jsonwebtoken';

const secret = 'Geração Tech'; // Palavra secreta

function gerarJWT() {
  const usuario = { id: 1, username: 'Pedro' };
  const options = { expiresIn: '1h' };

  // Gera um token
  return jwt.sign(usuario, secret, options);
}

function verificarToken(token) {
  jwt.verify(token, secret, (erro, decoded) => {
    if (erro) {
      console.error('Token inválido:', erro);
    } else {
      console.log('Token decodificado:', decoded);
    }
  });
}

// ===== Chamada das Functions ===== //
const token = gerarJWT();
console.log('Token:', token);

verificarToken(token);
```

Nodemon

Nodemon é uma ferramenta que ajuda a desenvolver aplicações baseadas em Node.js **reiniciando automaticamente o servidor** sempre que **mudanças** são **detectadas** nos arquivos do projeto. É especialmente útil durante o desenvolvimento, pois elimina a necessidade de reiniciar manualmente o servidor a cada alteração no código.

- **Monitoramento de arquivos:** Nodemon monitora os arquivos do projeto e reinicia automaticamente o servidor sempre que uma mudança é detectada.
- **Configuração simples:** Pode ser configurado facilmente através de um arquivo de configuração ou linha de comando.
- **Compatibilidade:** Funciona com qualquer aplicativo Node.js sem necessidade de alterações no código existente.

Instalação

Para instalar globalmente, de modo que possa ser utilizado em qualquer projeto:

```
npm install -g nodemon
```

Para instalar globalmente, de modo que possa ser utilizado em qualquer projeto:

```
npm install --save-dev nodemon
```

Iniciar uma aplicação com Nodemon

Ao invés de usar node para iniciar a aplicação, use nodemon:

```
nodemon app.js
```

Configuração Personalizada

- Nodemon pode ser configurado utilizando um arquivo **nodemon.json** no diretório raiz do projeto. Este arquivo permite especificar extensões de arquivos para monitorar, diretórios a serem ignorados, entre outros.

```
{  
  "watch": ["src"],  
  "ext": "js,json",  
  "ignore": ["node_modules"],  
  "exec": "node src/app.js"  
}
```

- É possível usar nodemon em conjunto com npm scripts. No arquivo package.json, configure um script para iniciar o servidor com nodemon:

```
{  
  "name": "meu-projeto",  
  "version": "1.0.0",  
  "main": "app.js",  
  "scripts": {  
    "start": "node app.js",  
    "dev": "nodemon app.js"  
  },  
  "dependencies": {  
    "express": "^4.17.1"  
  },  
  "devDependencies": {  
    "nodemon": "^2.0.7"  
  }  
}
```