CONSUMINDO UMA API DE CONVERSÃO DE MOEDA

Objetivo

I. ExchangeRate-API;
II. Estrutura do Projeto;
III. Inicialização do Projeto;
IV. Configuração da base do projeto;
V. Configurar o arquivo tsconfig.json
VI. Configurar o arquivo package.json
VII. Backend com TypeScript (src/server.ts)
VIII. Frontend em HTML (views/index.html)
IX. Execução do Projeto

I. ExchangeRate-API

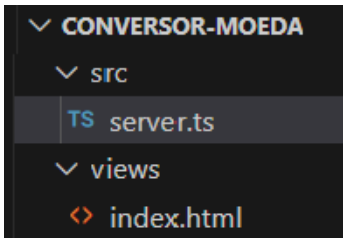
A ExchangeRate-API é uma interface de programação de aplicações (API) que fornece dados de taxas de câmbio para diversas moedas globais. Esta API busca dados de várias fontes, incluindo bancos centrais e provedores de dados financeiros, para oferecer taxas de câmbio precisas e atualizadas. É utilizada por empresas de todos os portes, desde pequenas e médias empresas até grandes corporações, em diversos setores. Ela oferece taxas de câmbio em tempo real e dados históricos além de permitir conversões entre diversas moedas.

Para obter os dados desta api, deve-se cadastrar gratuitamente no site: https://www.exchangerate-api.com/

Ao se cadastrar, anote a chave API do seu cadastro. Você irá precisar dela.

II. Estrutura do Projeto

Para o nosso projeto devemos ter a seguinte estrutura do projeto:



III. Inicialização do Projeto

Vamos iniciar o nosso projeto com a instalação das dependências:

```
npm init -y
npm install dotenv axios
npm install express@4 @types/express@4
npm install -D typescript ts-node-dev
npx tsc --init
```

IV. Configuração da Base do Projeto

Nosso projeto TypeScript consome uma API externa, o arquivo .env é essencial para armazenar informações sensíveis e configurar o ambiente da sua aplicação de forma segura e organizada.

Na raiz do projeto, vamos criar um arquivo .env que irá definir uma variável de ambiente chamada API_KEY com o valor da chave recebida no seu e-mail. O conteúdo desse arquivo será:

```
API_KEY=Sua_Chave_Aqui
```

V. Configurar o arquivo tsconfig.json

```json
{
  "compilerOptions": {
    "target": "es6",
    "module": "commonjs",
    "lib": ["es6"],
    "outDir": "./dist",
    "rootDir": "./src",
    "strict": true,
    "esModuleInterop": true,
    "moduleResolution": "node",
    "resolveJsonModule": true,
    "skipLibCheck": true
  },
  "include": ["src/**/*"]
}
```

VI. Configurar o arquivo package.json

```json
{
  "name": "conversor-moeda",
  "version": "1.0.0",
  "main": "index.js",
  ▷Debug
  "scripts": {
    "dev": "ts-node-dev --respawn --transpile-only src/server.ts",
    "build": "tsc",
    "start": "node dist/server.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": "",
  "dependencies": {
    "@types/express": "^4.17.21",
    "axios": "^1.8.4",
    "dotenv": "^16.4.7",
    "express": "^4.21.2"
  },
  "devDependencies": {
    "@types/node": "^22.14.0",
    "ts-node-dev": "^2.0.0",
    "typescript": "^5.8.3"
  }
}
```

VII. Backend com TypeScript (src/server.ts)

```typescript
import express, { Request, Response } from "express";
import dotenv from "dotenv";
import axios from "axios";
import path from "path";

dotenv.config();

const app = express();
const PORT = 3000;

app.use(express.static(path.join(__dirname, "../views")));

app.get("/", (req: Request, res: Response) => {
  res.sendFile(path.join(__dirname, "../views/index.html"));
});

app.get("/convert", async (req: Request, res: Response) => {
  const { amount, from, to } = req.query;

  if (!amount || !from || !to) {
    return res.status(400).json({ error: "Parâmetros inválidos" });
  }

  try {
    const apiKey = process.env.API_KEY;
    const url = `https://v6.exchangerate-api.com/v6/${apiKey}/pair/${from}/${to}/${amount}`;
    const response = await axios.get(url);
    const converted = response.data.conversion_result;
    res.json({ result: `${amount} ${from} = ${converted.toFixed(2)} ${to}` });
  } catch (error) {
    res.status(500).json({ error: "Erro ao converter moeda." });
  }
});

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

VIII. Frontend em HTML (views/index.html)

```html
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <title>Conversor de Moeda</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      text-align: center;
      margin-top: 50px;
    }
    input, select, button {
      margin: 10px;
      padding: 8px;
    }
  </style>
</head>
<body>
  <h1>Conversor de Moeda</h1>

  <form id="convertForm">
    <input type="number" step="0.01" name="amount" placeholder="Valor" required />
```

```
    <select name="from" required>
      <option value="">De:</option>
      <option value="USD">USD - Dólar Americano</option>
      <option value="EUR">EUR - Euro</option>
      <option value="BRL">BRL - Real Brasileiro</option>
      <option value="GBP">GBP - Libra Esterlina</option>
      <option value="JPY">JPY - Iene Japonês</option>
      <option value="CAD">CAD - Dólar Canadense</option>
      <option value="AUD">AUD - Dólar Australiano</option>
      <option value="CHF">CHF - Franco Suíço</option>
      <option value="CNY">CNY - Yuan Chinês</option>
    </select>

    <select name="to" required>
      <option value="">Para:</option>
      <option value="USD">USD - Dólar Americano</option>
      <option value="EUR">EUR - Euro</option>
      <option value="BRL">BRL - Real Brasileiro</option>
      <option value="GBP">GBP - Libra Esterlina</option>
      <option value="JPY">JPY - Iene Japonês</option>
      <option value="CAD">CAD - Dólar Canadense</option>
      <option value="AUD">AUD - Dólar Australiano</option>
      <option value="CHF">CHF - Franco Suíço</option>
      <option value="CNY">CNY - Yuan Chinês</option>
    </select>
```

```
    <button type="submit">Converter</button>
  </form>

  <h2 id="result"></h2>

  <script>
    document.getElementById("convertForm").addEventListener("submit", async function (e) {
      e.preventDefault();
      const form = e.target;
      const amount = form.amount.value;
      const from = form.from.value;
      const to = form.to.value;

      const response = await fetch(`/convert?amount=${amount}&from=${from}&to=${to}`);
      const data = await response.json();

      const result = document.getElementById("result");
      result.textContent = data.result || data.error;
    });
  </script>
</body>
</html>
```

IX. Execução do Projeto

Utilize os comandos abaixo para compilar e executar o projeto.

```
npm run build
npm run dev
```

Execute em localhost:3000