Universidade de Brasília UnB Gama

Métodos de Desenvolvimento de Software

Dashboard Integração MDS

Professora: Carla Rocha

Alunos:

Nathan Abreu 221022696

Maria Helena Carvalho 222006982

Marlon 222025914

Thales Henrique - 222006178

Victor Hugo 222021924

Otavio 211043692

Marcelo Adrian Ribeiro de Araújo 202016909

22 de maio de 2024

Brasília - DF

Apresentação: Integrando Dados JSON em um Dashboard React Usando Plotly

Introdução

Com o crescimento da demanda por dashboards interativos e intuitivos, a combinação de React e Plotly se tornou uma escolha popular. React oferece uma poderosa biblioteca de interface de usuário, enquanto a Plotly fornece gráficos interativos e personalizáveis. Neste guia, mostraremos como integrar dados JSON diretamente em um projeto React para criar gráficos dinâmicos com Plotly.

Argumentação

Optar por integrar dados JSON diretamente em um projeto React apresenta várias vantagens:

1. Simplicidade: Evita a necessidade de um backend adicional, como Flask, para servir os dados.
2. Eficiência: Reduz a complexidade da arquitetura, facilitando a manutenção e o desenvolvimento.
3. Velocidade: Carregar dados diretamente do diretório público é rápido e direto, ideal para dados estáticos ou que não mudam com frequência.

Vamos explorar como configurar um projeto React que lê dados de arquivos JSON e usa Plotly para renderizar gráficos.

Passos Detalhados

1. Estrutura do Projeto

Organização do projeto em front-end e back-end:

```
my-project
|
├── backend
|   └── data
|       └── data.json
|
├── frontend
|   ├── public
|   ├── src
|   |   ├── components
|   |   |   └── Dashboard.js
|   |   └── App.js
|   ├── package.json
|   └── README.md
└── README.md
```

Instalar Plotly e React-Plotly.js no Frontend

No diretório do frontend instalei Plotly e a integração do React.

```
cd frontend
npm install plotly.js react-plotly.js
```

Configurar o Acesso ao Json

Para acessar os arquivos JSON que estão fora do diretório público do frontend, configurei um script de build para copiar os arquivos JSON do backend para o diretório público do frontend.

Instalei o pacote copyfiles para ajudar com isso:

```
npm install --save-dev copyfiles
```

Adicionei um script no Package.json do frontend para copiar os arquivos JSON:

```
{
  "scripts": {
    "copy-json": "copyfiles -u 2 '../backend/data/*.json' 'public/data'"
  }
}
```

Agora, sempre que eu rodar npm run copy-json, ele copiará os arquivos JSON do backend para o diretório público do frontend.