

Atividade de Fixação: Desenvolvendo um Conversor de Moedas com React

Objetivo:

O objetivo desta atividade é reforçar os conceitos de React, especialmente o uso de componentes, estados, e chamadas a APIs externas para criar uma aplicação funcional que converte valores entre diferentes moedas.

Descrição do Projeto:

Você vai criar um conversor de moedas simples utilizando React. O projeto permitirá ao usuário inserir um valor em uma moeda específica e convertê-lo para outra moeda à escolha. Os valores de câmbio serão obtidos de uma API externa. Esse projeto ajudará a consolidar o entendimento sobre como gerenciar estados, lidar com formulários e realizar requisições HTTP em React.

Passo a Passo para Desenvolver o Projeto:

Passo 1: Configuração Inicial do Projeto

1. Criar um Novo Projeto React:

- Abra o terminal e execute o comando para criar um novo projeto:

```
npx create-react-app conversor-moedas
```

- Navegue até o diretório do projeto:

```
cd conversor-moedas
```

2. Preparar o Arquivo Principal:

- No diretório `src`, abra o arquivo `App.js` e remova o conteúdo padrão:

```
import React from 'react';
import './App.css';

function App() {
  return (
    <div className="App">
      <h1>Conversor de Moedas</h1>
    </div>
  );
}

export default App;
```

Passo 2: Criar o Formulário para Conversão

1. Criar o Formulário de Entrada:

- Dentro do arquivo `App.js`, adicione um formulário para o usuário inserir o valor e escolher as moedas:

```
import React, { useState } from 'react';
import './App.css';

function App() {
  const [amount, setAmount] = useState('');
  const [fromCurrency, setFromCurrency] = useState('USD');
  const [toCurrency, setToCurrency] = useState('EUR');
  const [convertedAmount, setConvertedAmount] =
    useState(null);

  const handleAmountChange = (e) =>
    setAmount(e.target.value);
  const handleFromCurrencyChange = (e) =>
    setFromCurrency(e.target.value);
  const handleToCurrencyChange = (e) =>
    setToCurrency(e.target.value);

  return (
    <div className="App">
      <h1>Conversor de Moedas</h1>
      <form>
        <input
          type="number"
          value={amount}
          onChange={handleAmountChange}
          placeholder="Digite o valor"
        />
        <select value={fromCurrency}
          onChange={handleFromCurrencyChange}>
          <option value="USD">USD</option>
          <option value="EUR">EUR</option>
          <option value="BRL">BRL</option>
          { /* Adicione mais opções conforme necessário */ }
        </select>
        <span>para</span>
        <select value={toCurrency}
          onChange={handleToCurrencyChange}>
          <option value="EUR">EUR</option>
          <option value="USD">USD</option>
          <option value="BRL">BRL</option>
          { /* Adicione mais opções conforme necessário */ }
        </select>
        <button type="button">Converter</button>
      </form>
      {convertedAmount && (
        <h2>
          {amount} {fromCurrency} é igual a
          {convertedAmount} {toCurrency}
        </h2>
      )}
    </div>
  );
}

export default App;
```

- Esse formulário básico permite ao usuário inserir o valor e selecionar as moedas para conversão. O botão de conversão ainda não está funcional.

Passo 3: Fazer a Requisição à API de Conversão

1. Adicionar a Lógica de Conversão:

- Utilize a função `fetch` para obter os valores de câmbio de uma API externa (exemplo: [ExchangeRate-API](#)):

```
import React, { useState } from 'react';
import './App.css';

function App() {
  const [amount, setAmount] = useState('');
  const [fromCurrency, setFromCurrency] = useState('USD');
  const [toCurrency, setToCurrency] = useState('EUR');
  const [convertedAmount, setConvertedAmount] =
    useState(null);

  const handleAmountChange = (e) =>
    setAmount(e.target.value);
  const handleFromCurrencyChange = (e) =>
    setFromCurrency(e.target.value);
  const handleToCurrencyChange = (e) =>
    setToCurrency(e.target.value);

  const handleConvert = async () => {
    const response = await fetch(
      `https://v6.exchangerate-api.com/v6/YOUR-API-
      KEY/latest/${fromCurrency}`
    );
    const data = await response.json();
    const rate = data.conversion_rates[toCurrency];
    setConvertedAmount((amount * rate).toFixed(2));
  };

  return (
    <div className="App">
      <h1>Conversor de Moedas</h1>
      <form>
        <input
          type="number"
          value={amount}
          onChange={handleAmountChange}
          placeholder="Digite o valor"
        />
        <select value={fromCurrency}
          onChange={handleFromCurrencyChange}>
          <option value="USD">USD</option>
          <option value="EUR">EUR</option>
          <option value="BRL">BRL</option>
          { /* Adicione mais opções conforme necessário */ }
        </select>
        <span>para</span>
        <select value={toCurrency}
          onChange={handleToCurrencyChange}>
          <option value="EUR">EUR</option>
          <option value="USD">USD</option>
          <option value="BRL">BRL</option>
```

```
        { /* Adicione mais opções conforme necessário */ }
      </select>
      <button type="button"
onClick={handleConvert}>Converter</button>
    </form>
    {convertedAmount && (
      <h2>
        {amount} {fromCurrency} é igual a
        {convertedAmount} {toCurrency}
      </h2>
    )}
  </div>
);
}

export default App;
```

- Este código adiciona a lógica para converter o valor usando a taxa de câmbio obtida da API. O valor convertido é então exibido abaixo do formulário.

Passo 4: Estilizar a Aplicação

1. Adicionar Estilos ao Projeto:

- No arquivo `App.css`, adicione o seguinte para estilizar a aplicação:

```
.App {
  text-align: center;
  margin-top: 50px;
}

input, select, button {
  margin: 10px;
  padding: 10px;
  font-size: 16px;
}

h2 {
  margin-top: 20px;
  color: #4CAF50;
}
```

- Isso dará um visual limpo e organizado ao formulário e ao resultado da conversão.

Passo 5: Testar a Aplicação

1. Executar o Projeto:

- No terminal, execute o comando para iniciar o projeto:

```
npm start
```

- Acesse `http://localhost:3000` no navegador para interagir com a aplicação.

2. Verificar o Funcionamento:

- Insira um valor e selecione as moedas para verificar se a conversão está funcionando corretamente.

Passo 6: Melhorias e Desafios

1. Adicionar Validações:

- Implemente validações para garantir que o valor inserido seja um número positivo antes de realizar a conversão.

2. Histórico de Conversões:

- Como um desafio adicional, adicione uma funcionalidade que mostre o histórico das últimas cinco conversões realizadas.

Conclusão

Com esta atividade, você vai praticar o uso de estados, componentes e chamadas a APIs externas, além de desenvolver uma aplicação prática que pode ser expandida com diversas funcionalidades adicionais.