Material de Apoio

Context API

O que é?

A Context API é uma funcionalidade do React introduzida na versão 16.3, que permite compartilhar dados entre componentes sem a necessidade de passar props manualmente em cada nível da árvore de componentes.

Isso é particularmente útil para compartilhar dados que são consumidos por muitos componentes em diferentes níveis da aplicação, como informações de autenticação do usuário, temas, ou preferências do usuário.

Como funciona?

A Context API é composta por três partes principais:

1. Criar o Contexto: Você cria um contexto utilizando a função React.createContext().
2. Provider (Provedor): O provedor é um componente que envolve outros componentes e permite que o contexto esteja disponível para todos os componentes filhos. Ele usa a propriedade value para fornecer o valor do contexto.
3. Consumer (Consumidor): Os consumidores são os componentes que acessam o valor do contexto. Eles utilizam a função useContext para obter o valor do contexto.

Tipos de API's

Criar Contexto

```
import React from 'react';

const MeuContexto = React.createContext();
```

Criar Provider

```
import React, { useState } from 'react';

const MeuContextoProvider = ({ children }) => {
  const [estado, setEstado] = useState('valor inicial');

  return (
    <MeuContexto.Provider value={{ estado, setEstado }}>
      {children}
    </MeuContexto.Provider>
  );
};

export { MeuContexto, MeuContextoProvider };
```

Tipos de API's

Utilizar o Consumer

```
import React, { useContext } from 'react';
import { MeuContexto } from './MeuContexto';

const MeuComponente = () => {
  const { estado, setEstado } = useContext(MeuContexto);

  return (
    <div>
      <p>O valor do estado é: {estado}</p>
      <button onClick={() => setEstado('novo valor')}>Mudar valor</button>
    </div>
  );
};

export default MeuComponente;
```

Exemplos de Uso

Tema da Aplicação

Compartilhar o tema da aplicação (claro/escuro) entre diversos componentes.

Autenticação do Usuário

Manter o estado de autenticação do usuário disponível em toda a aplicação.

Configurações de Idioma

Compartilhar a linguagem selecionada pelo usuário em componentes que exibem texto.

Exemplos de Uso de APIs

Vantagens:

- Reduz a necessidade de passar props manualmente.
- Facilita a manutenção e a leitura do código.
- Melhora a organização do estado global da aplicação.

Desvantagens:

- Pode introduzir complexidade se não for usado corretamente.
- Rerenders podem ocorrer em componentes que não são necessários se o valor do contexto mudar.

Dicas e Boas Práticas

- Use a Context API para dados que realmente precisam ser compartilhados entre muitos componentes.
- Combine a Context API com hooks personalizados para organizar melhor o código.
- Evite usar a Context API para estados locais de componentes, prefira useState ou useReducer.

E aí, curtiu?

Esperamos que esse resumo tenha enriquecido sua perspectiva estratégica para enfrentar os desafios.

Salve esse PDF para consultar sempre que precisar.

DNC