

React: useContext

НАШИ ПРАВИЛА




Включенная камера




Вопросы по поднятой руке



Не перебиваем друг друга



Все вопросы, не связанные с тематикой курса (орг-вопросы и т. д.), должны быть направлены куратору



Подготовьте свое рабочее окружение для возможной демонстрации экрана (закройте лишние соцсети и прочие приложения)

Повторим;)

■ В чем разница SPA и MPA?

■ Какие типы ссылок есть в react router?

■ Какой вспомогательный хук для навигации в react router вы знаете?

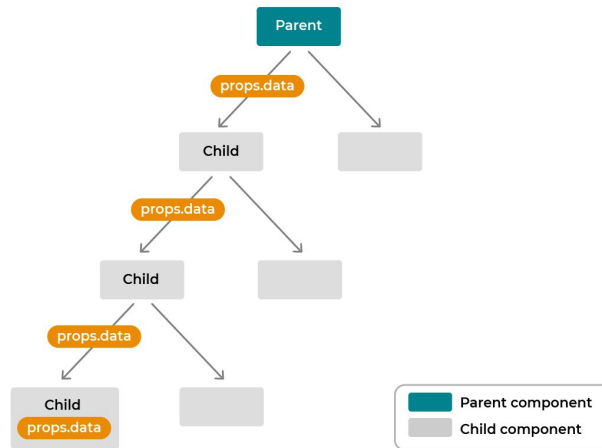
ЦЕЛЬ

Изучить способ передачи данных через большую вложенность компонентов

ПЛАН ЗАНЯТИЯ

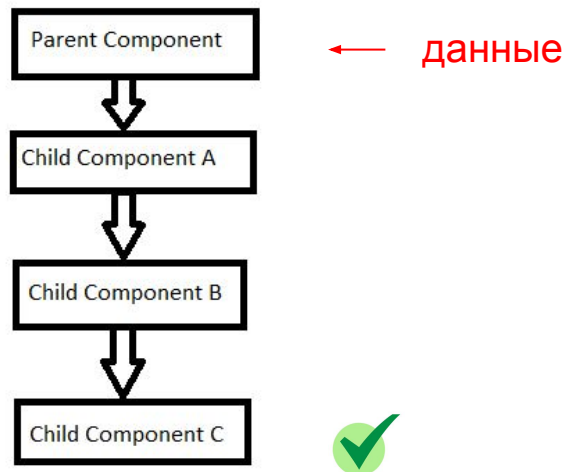
- 
- 1. Props drilling
 - 2. useContext

useContext



useContext - это React Hook, предоставляющий способ получения значения из контекста в функциональных компонентах React.

Контекст в React используется для передачи данных глубоко вложенным компонентам без явной передачи пропс через каждый уровень.



Шаг 1. Создаём контекст в компоненте, который будет передавать значение

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

// Создаем контекст с типом React.Context и значением по умолчанию
const MyContext = createContext<string>("");

// Компонент-поставщик контекста
const MyContextProvider = ({ children }) => {
  const [sharedValue, setSharedValue] = useState('Значение из контекста')

  const updateValue = (newValue) => {
    setSharedValue(newValue);
  };

  return (
    <MyContext.Provider value={{ sharedValue, updateValue }}>
      {children}
    </MyContext.Provider>
  );
};

export { MyContext, MyContextProvider };
```

создаём контекст

делаем компонент
поставщиком
контекста

Шаг 2. Соответствующий `<Context.Provider>` должен находиться выше компонента, которому нужны данные из контекста. В данном случае **MyContextProvider** находится на самом верхнем уровне, но это необязательно приложения

```
import React from 'react';
import { MyContextProvider } from './MyContextProvider';
import MyComponent from './MyComponent';

const App = () => {
  return (
    <MyContextProvider>
      <div>
        <h1>Мое React Приложение</h1>
        <MyComponent />
        {/* Другие компоненты в приложении */}
      </div>
    </MyContextProvider>
  );
};

export default App;
```

импортируем
компонент с
контекстом

Оборачиваем другие
компоненты

Шаг 3. Передаём значение из компонента поставщика (провайдера) в другой компонент

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

// Компонент, использующий значение из контекста с помощью useContext
const MyComponent = () => {
  const { sharedValue, updateValue } = useContext(MyContext);

  const handleClick = () => {
    // Обновляем значение в контексте
    updateValue('Новое значение из компонента');
  };

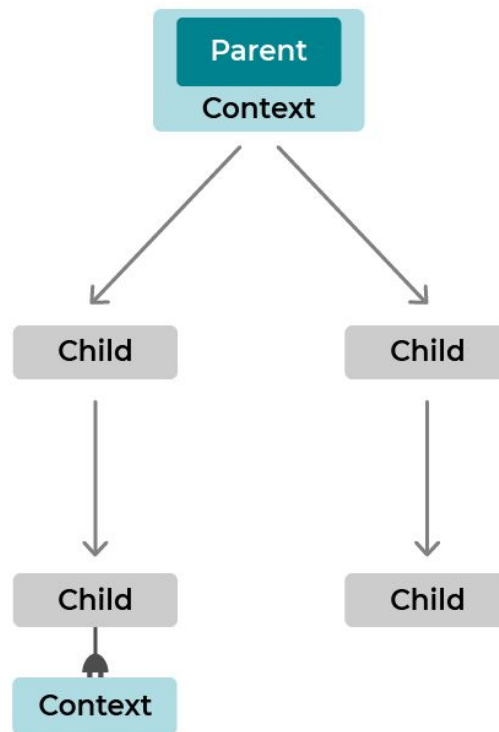
  return (
    <div>
      <p>Значение из контекста: {sharedValue}</p>
      <button onClick={handleButtonClick}>Обновить значение</button>
    </div>
  );
};

export default MyComponent;
```

импортируем
контекст

С помощью useContext
получаем доступ к
значению и функции
для его изменения из
компонента
MyContextProvider

Когда значение в контексте обновляется в `MyContextProvider`, все компоненты, использующие `useContext(MyContext)`, автоматически получают новое значение, и их перерисовывается.





Ваша новая IT-профессия – Ваш новый уровень жизни

Программирование с нуля в
немецкой школе AIT TR GmbH