

Расширенные хуки и базовая оптимизация

Junior Frontend Developer

Цель урока:

1. Изучим хуки **useRef**, **useMemo**, **useCallback**
2. Поймем, что такое **компоненты высшего порядка** (Higher-Order Components, HOC)
3. Поймем, зачем нужен **React.memo** (компонент высшего порядка) и чем он отличается от **useMemo**
4. Изучим **React.cloneElement** и **React.Children**
5. Закрепим полученные знания на создании универсального **компонента формы**

Содержание урока:

1. Хуки **useRef**, **useMemo**, **useCallback**, их особенности и способы применения
2. HOC (компоненты высшего порядка) - что это такое и чем могут быть полезны
3. **React.memo** (компонент высшего порядка) - повышение производительности
4. **React.cloneElement**, **React.Children** - особенности и способы применения
5. Разработка переиспользуемого компонента формы
6. Автоматическая установка фокуса на поле ввода при открытии страницы
7. Переключение полей для ввода при нажатии на Enter

Обратите внимание!

Данный урок затрагивает более глубокие темы React. Сейчас вы можете просто ознакомиться с материалом, а в дальнейшем при прохождении курса возвращаться и погружаться более глубоко.

Также обратите внимание, что в уроке используется **не основной проект "fast-company"**, а его модифицированная версия. В следующих уроках мы вернемся к нашему основному проекту.

useRef — в нем можно **сохранять любое значение и хранить/изменять его** в течение всей жизни компонента. При изменении значения **не произойдет повторный рендер**.

useMemo — С помощью него можно **избежать дорогостоящих вычислений при каждом рендере**.

useCallback — С помощью него можно **предотвращать ненужные рендереры дочерних компонентов, которым мы передаем callback функцию**.

HOC (компонент высшего порядка) — С помощью них можно **переиспользовать один и тот же функционал для нескольких компонентов**.

React.memo — С помощью него можно **оптимизировать количество ререндеров целого компонента**.

`React.cloneElement` — С помощью него можно **клонировать React элемент, добавляя или изменяя его параметры.**

`React.Children` — С помощью него можно **использовать полезные функции для работы с `props.children`.**