

# Управління ефектами компонентів

## Можливості методу useEffect

**№ уроку:** 5 **Курс:** React Базовий

**Засоби навчання:** Visual Studio Code, розширення React Developer Tools, пакет-менеджер node.

### Огляд, ціль та призначення уроку

Визначити DOM-елементи в React. ознайомитися з хуком прямого доступу до DOM-елементів useRef. Визначити, що таке рендер в React та як він працює. Ознайомитися з хуком обробки сторонніх ефектів useEffect.

### Вивчивши матеріал даного заняття, студент зможе:

- Застосовувати хук useRef.
- Розуміти процес рендерингу функціональних компонентів.
- Вміти застосовувати хук useEffect.

### Зміст уроку

1. Ознайомлення з DOM-елементами в React.
2. Ознайомлення з хуком useRef.
3. Огляд рендеру функціональних компонентів.
4. Ознайомлення з хуком useEffect.

### Резюме

- Кінцевою реалізацією будь-якого вебінтерфейсу є HTML-документи. Основою HTML-документів є теги, кожен HTML-тег можна представити об'єктом DOM-дерева. В React об'єктами DOM можна вважати елементи, які ми розробляємо за допомогою синтаксису JSX.
- Щоб «прив'язатись» до якогось DOM-елементу всередині функціонального компонента React, використовується хук useRef.
- Метод useRef в функціональних компонентах дозволяє отримувати доступ до елементів DOM-дерева напряму. Даний хук відкриває доступ до отримання значення value, статусу, стилів та інших атрибутів JSX-елементів.
- Процес рендерингу - це внутрішні дії реакту пов'язані з формуванням віртуального дерева компонентів. Після того як віртуальний DOM був сформований, він порівнюється з поточним DOM-деревом. Якщо в процесі були виявлені якісь відмінності, відбувається оновлення поточного DOM-дерева та інтерфейс перерисовується.
- Процес рендерингу в React не перерисовує DOM дерево, його задача сформувати віртуальне DOM дерево, яке потім порівнюється з попереднім та якщо вони нічим не відрізняються, оновлення не відбувається.

- Щоб обробляти сторонні ефекти життєвого циклу функціонального компонента в React, використовується хук `useEffect`. Хук `useEffect` - це метод який дозволяє синхронізувати компонент із зовнішньою системою. В реальній розробці даний метод дозволяє встановити слухач, що буде виконувати функцію, яку ми помістимо в нього, кожен раз при спрацюванні відповідних залежностей.

### Закріплення матеріалу

- За допомогою чого створюються об'єкти DOM в React компонентах?
- За допомогою якого хука можна отримати значення, введене в елемент `input type='text'` ?
- Чи відбувається рендер батьківського компонента при рендері дочірнього?
- Чи відбувається рендер дочірнього компонента при рендері батьківського?
- За допомогою якого хука можна викликати функцію після рендеру компоненту?

### Додаткове завдання

#### Завдання

Реалізуйте декілька компонентів, які будуть відображати пости та коментарі. Дані отримайте через API з порталу <https://jsonplaceholder.typicode.com> - /posts та /comments.

### Самостійна діяльність студента

#### Завдання 1

Створити 3 елементи `input` для імені, прізвища та емейл адреси. До кожного елементу `input` застосувати хук `useRef`. Створити елемент `button` та функцію, яка буде збирати введенні в `input` дані та додавати їх в `state`. Порада: використовуйте масив об'єктів, як тип даних для `state`.

#### Завдання 2

Створити хук `useEffect`, який при кожному рендері компоненту буде виводити останнє значення `стейту` в консоль.

#### Завдання 3

Створити дочірній компонент, який буде приймати `props`, та передайте в нього якийсь об'єкт. Відобразіть даний `props` в інтерфейсі. Що відбудеться, якщо `props` зміниться?

### Рекомендовані ресурси

#### Рендер в React

<https://react.dev/reference/react-dom/render>

#### Хук `useRef` в React

[https://www.w3schools.com/react/react\\_useref.asp](https://www.w3schools.com/react/react_useref.asp)

#### Хук `useEffect` в React

<https://react.dev/reference/react/useEffect>

#### Метод `useEffect` простою мовою

<https://dmitripavlutin.com/react-useeffect-explanation/>