

# Урок 5: useEffect, ContextAPI

Temirlan  
Akhanov



# Что такое Side Effect (побочный эффект) ?

В программировании термин "побочный эффект" относится к ситуациям, когда функция или операция, кроме возвращения значения, также вносит изменения в состояние системы или имеет наблюдаемое взаимодействие с внешним миром.

Изменение  
глобальных  
переменных

Изменение  
входных  
данных

Взаимодействие с  
внешним миром

Ошибки и  
исключения

Модификация  
баз данных

# Хендлим побочные эффекты с `useEffect()` хуком

```
useEffect(() => { ... }, [ зависимости ]);
```



Функция которая  
отработает ПОСЛЕ  
рендера компонента  
ЕСЛИ изменились  
зависимости

Ваш побочный эффект  
кладем в функцию



Зависимости этого  
эффекта – функция  
отработает только тогда  
если одна из них  
изменится

Укажите зависимости тут

# Что не нужно добавлять в массив зависимостей `useEffect()`?

Функции  
обновления  
состояния (Set...)

Встроенные API  
функции

Переменные,  
функции ВНЕ  
компонента

- Вам НЕ нужно добавлять функции обновления состояния (как мы делали на последней лекции с `setFormIsValid`): React гарантирует, что эти функции не меняются, поэтому вам не нужно добавлять их в зависимости (хотя вы могли бы).
- Вам также НЕ нужно добавлять "встроенные" API или функции, такие как `fetch()`, `localStorage` и т. д. (функции и возможности, встроенные в браузер и, следовательно, доступные глобально): Эти API браузера / глобальные функции не связаны с циклом рендеринга компонента React и они также не меняются.
- Вам также НЕ нужно добавлять переменные или функции, которые вы могли бы определить ВНЕ своих компонентов (например, если вы создаете новую вспомогательную функцию в отдельном файле): такие функции или переменные также не создаются внутри функции компонента, и следовательно, изменение их не повлияет на ваши компоненты (компоненты не будут переоцениваться, если такие переменные или функции меняются и наоборот).

Таким образом, коротко говоря: вы должны добавить все "вещи", которые вы используете в вашей функции эффекта, если эти "вещи" могут измениться из-за того, что ваш компонент (или какой-либо родительский компонент) был перерисован. Вот почему переменные или состояние, определенное в функциях компонента, свойства или функции, определенные в функциях компонента, должны быть добавлены в качестве зависимостей!