**Практическая работа №20.**

**Тема:** Работа с Axios и JSONPlaceholder

**Цель:** Научиться работать с API с использованием библиотеки axios, выполнять запросы GET , POST , PUT и DELETE, управлять состояниями с помощью useState .

**Задания:** Создайте приложение "Калькулятор", где пользователь может:

1. Загрузить список постов.
2. Просмотреть детали поста.
3. Добавить новый пост.
4. Редактировать пост.
5. Удалить пост.

Ход работы

Чтобы загрузить список постов необходимо создать функцию, которая будет получать, и обновлять данные о постах.

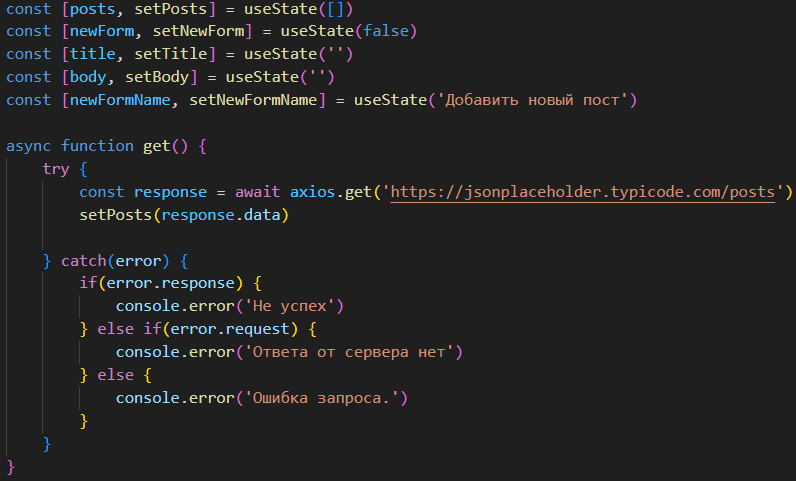


Рисунок 1 – функция получения данных

Дальше мы должны создать кнопку, при нажатии на которую будет срабатывать функция.



Рисунок 2 – кнопка обновления данных

Далее после написания функций, мы должны создать элемент, который будет отображать наш список полностью. Вместе с последующими функциями.



Рисунок 3 – список постов

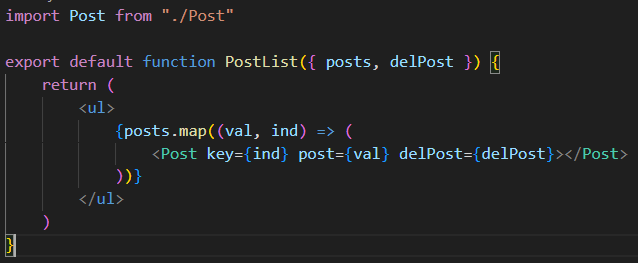


Рисунок 4 – функция вывода списка с постами

Чтобы просмотреть детали поста можно создать кнопку, которая будет переключать переменную и получать данные по посту. Отображение будет ниже.

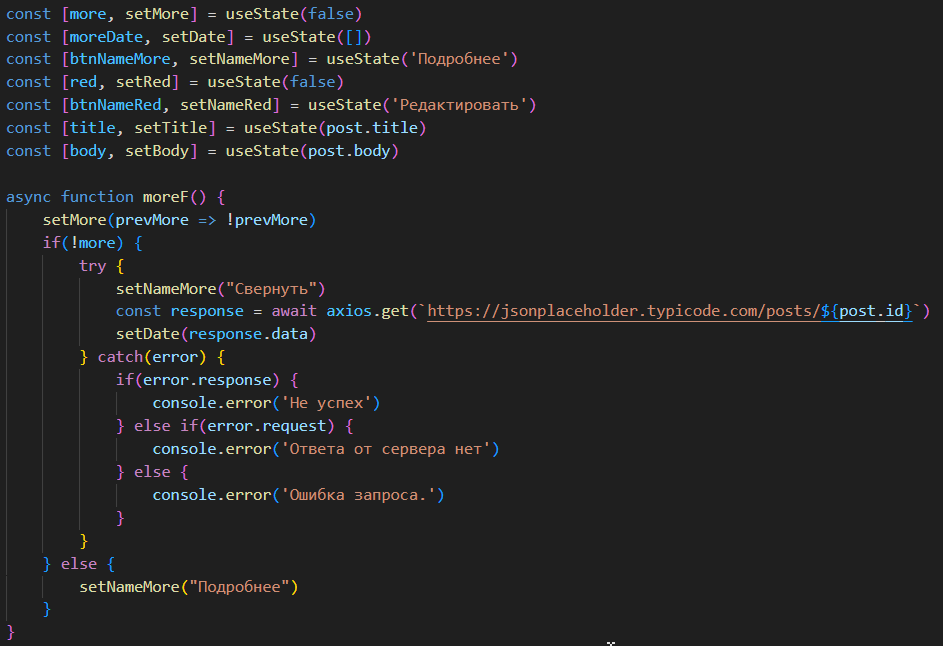


Рисунок 5 – функция раскрытия подробностей

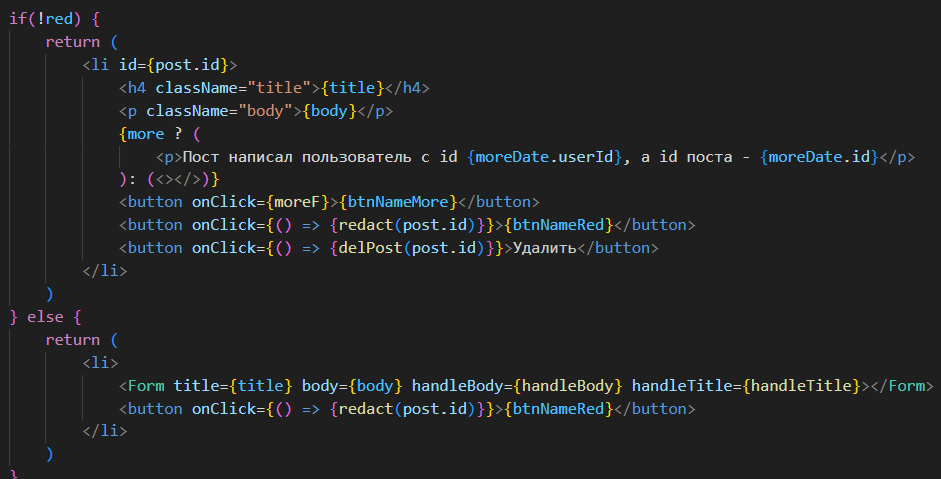


Рисунок 6 – отображение поста

Для добавления нового поста стоит создать форму общего вида, для отображения после и в постах.

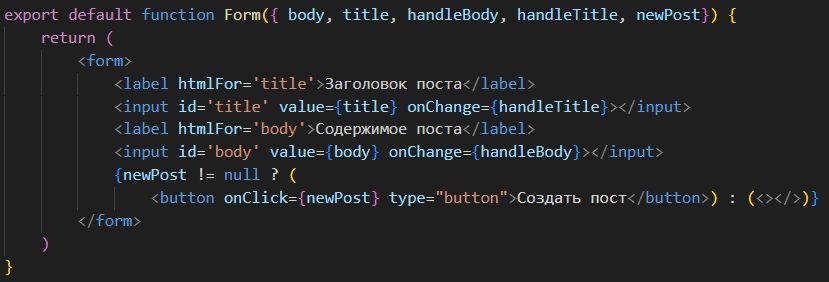


Рисунок 7 – форма ввода

По похожей схеме добавляем развёртывание формы и обрабатываем всё с помощью функций.

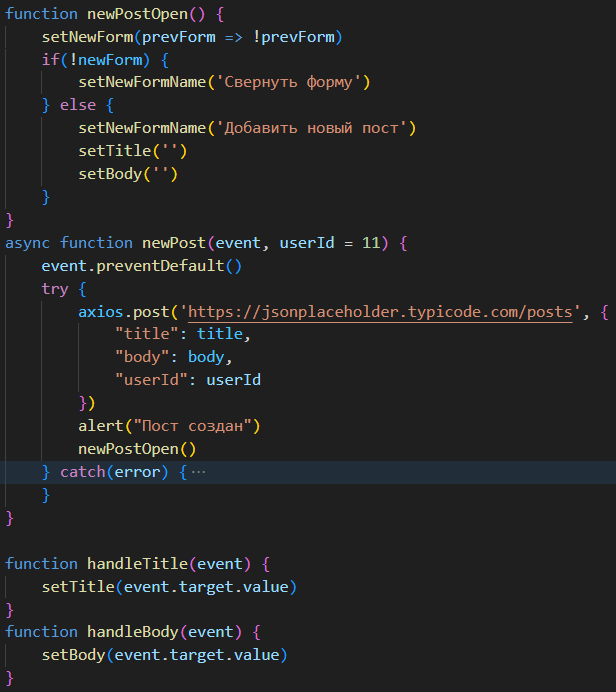


Рисунок 8 – создание поста

Для добавления функции редактирования поста по уже также знакомой схеме будет делать функцию. Загрузка на страницу уже показывалась выше.

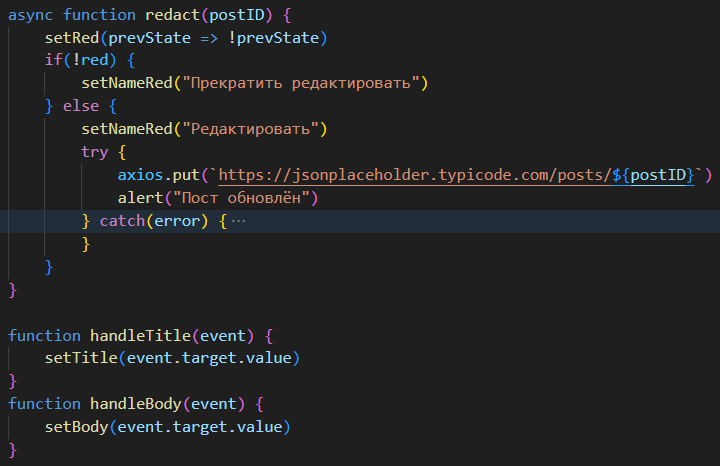


Рисунок 9 – функция редактирования

Прям удаление поста не получилось, но покажу что получилось, потому что оно работает, но немного. :<

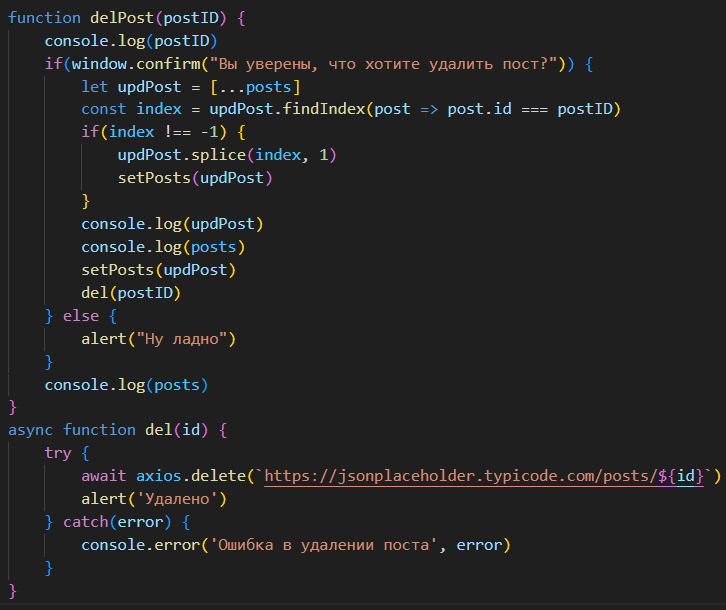


Рисунок 10 – попытка удаления поста

Ответы на вопросы:

1. Axios помогает нам упрощенно отправлять запросы на сервер.
2. В GET нет необходимости в теле запроса.