

АНАЛІЗ МЕТОДІВ КЛІЄНТ-СЕРВЕРНОЇ ВЗАЄМОДІЇ ПРИ РОЗРОБЦІ ПАНЕЛІ АДМІНІСТРАТОРА ЗА ДОПОМОГОЮ РЕАКТИВНОГО ФРЕЙМВОРКА REACT

Вінницький національний технічний університет

Анотація

В роботі проведено аналіз методів клієнт-серверної взаємодії зі сторони клієнта, що використовуються при розробці панелі адміністратора, яка дозволяє підтримувати сайт структурного підрозділу та оновлювати інформацію за допомогою візуального інтерфейсу. Використаний frontend фреймворк React для зручної розробки та підтримки сайту.

Ключові слова: React, GraphQL, клієнт-сервер, Apollo Client, MobX.

Abstract

The paper analyzes the methods of client-server interaction on the part of the client, used in the development of the admin panel, which allows you to maintain the site of the structural unit and update the information using the visual interface. Used frontend framework React for easy site development and maintenance.

Keywords: React, GraphQL, client-server, Apollo Client, MobX.

Вступ

Неможливо представити сучасні веб-додатки без використання клієнт-серверної взаємодії. Розглядаючи саме клієнтську частину взаємодії, можна виділити такі технології, як програмний інтерфейс REST API або бібліотека Apollo Client, що використовується для створення користувацького інтерфейсу. Використовуючи спільно ці можливості, можна реалізувати комбіновану та зручну архітектуру проекту.

Аналізуючи найпопулярніші глобальні сховища — MobX та Redux — які можна поєднати з React, вибір пав на другий варіант. Порівняно з бібліотекою Redux перша ефективніша для використання в невеликих проектах, продуктивніша та дозволяє створювати відразу декілька сховищ, що допоможе зручніше розділити стан адміністративної панелі і основного сайту.

Вибір методу для клієнт-серверної взаємодії

Apollo Client — це бібліотека керування станом для JavaScript, яка дозволяє керувати локальними та віддаленими даними за допомогою GraphQL. Вона використовується для отримання, кешування та зміни даних програми, автоматичного оновлення інтерфейсу користувача. Також Apollo Client допомагає структурувати код без ручного відстеження станів додатку (декларативний підхід), що відповідає сучасним методам розробки програмного забезпечення. Основна бібліотека забезпечує вбудовану інтеграцію з React [1]. Щодо використання безпосередньо в коді, в основному використовуються хуки useQuery та useMutation. В деяких випадках, наприклад для сповіщення клієнта про зміну даних на сервері, використовують хук useSubscription.

Керування даними не повинне бути складним та незрозумілим. Саме декларативний підхід та кешування Apollo до отримання даних забезпечує написання меншої кількості коду з кращим результатом.

Схематично зв'язок між сервером і клієнтом, використовуючи Apollo, подано на рис. 1. Можна побачити, що взаємодія з базою даних, робота мікросервісів та REST API за допомогою цієї бібліотеки дозволяє використання GraphQL на додатках для IOS чи Android, або ж, в нашому випадку — веб-додатку.

В сукупності з React така архітектура забезпечить високу швидкодію сайту структурного підрозділу, його масштабованість під різні доповнення для майбутніх розробників, які будуть підтримувати чи покращувати додаток.



Рисунок 1 — Візуальне представлення клієнт-серверної взаємодії з допомогою Apollo

При цьому, при додатковому використанні бібліотеки MobX з'являється контроль глобального стану додатку та зручне використання всіх даних сховища [2]. Окрім продуктивності, дана бібліотека не потребує ручного керування зміною стану компонента чи даних. Це відбувається автоматично, на відміну від Redux.

Висновки

Масштабованість, простота та швидкодія — одні з найважливіших параметрів сучасних веб-додатків. Вибрані технології та реалізація їх взаємодії дозволяють досягти найкращих значень вказаних параметрів.

Apollo Client для роботи з GraphQL являється популярним та перспективним вибором для сайту, допомагає використати просту та водночас ефективну архітектуру додатку. Ще однією з переваг Apollo є доступність потрібних даних для серверного рендерингу відразу «із коробки».

Керування глобальним станом реалізовано з допомогою бібліотеки MobX, яка є відразу зручним, в деяких випадках гнучким та простим рішенням.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Introduction to Apollo Client [Електронний ресурс] – Режим доступу до ресурсу: <https://www.apollographql.com/docs/react/>

2. MobX introduction [Електронний ресурс] – Режим доступу до ресурсу: <https://mobx.js.org/README.html>

Войцеховська Олена Валеріївна – кандидат технічних наук, доцент кафедри обчислювальної техніки, Вінницький національний технічний університет, Вінниця

Нукитюк В'ячеслав Олександрович – студент групи KI-20МСз, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: slavik.nukutuk@gmail.com

Voytsekhovska Olena V. — PhD, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University

Nykytiuk Viacheslav O. – students, KI-20MS, Faculty of information Technologies and Computer Engineering, Vinnytsia National Technical University, email: slavik.nukutuk@gmail.com