

DOI: <https://doi.org/10.36910/6775-2524-0560-2023-51-04>

УДК 004.94:656.02

Багнюк Наталія Володимирівна, к.т.н., доцент

<https://orcid.org/0000-0002-7120-5455>

Бортник Катерина Яківна, к.т.н., доцент

<https://orcid.org/0000-0001-5282-099X>

Лавренчук Світлана Василівна

<https://orcid.org/0000-0002-5453-3924>

Домарацький Ілля Віталійович, магістр

Луцький національний технічний університет, м.Луцьк, Україна

ПРИНЦИП РОБОТИ КРОСПЛАТФОРМЕННИХ ДОДАТКІВ З ВЕЛИКИМИ ОБ'ЄМАМИ ДАНИХ З ВИКОРИСТАННЯМ БІБЛІОТЕКИ REDUX

Багнюк Н.В., Бортник К.Я., Лавренчук С.В., Домарацький І.В. Принцип роботи кросплатформених додатків з великими об'ємами даних з використанням бібліотеки Redux. У статті було розглянуто проблеми розробки кросплатформених додатків з великими обсягами даних та запропоновано способи їх вирішення. Досліджено принцип роботи кросплатформених додатків з великими об'ємами даних з використанням бібліотеки Redux та порівняно його з Redux toolkit.

Ключові слова: React Native, Redux, кросплатформенні додатки, Redux toolkit.

Bahniuk N., Bortnyk K., Lavrenchuk S., Domaratskyi I. The principle of operation of cross-platform applications with large volumes of data using the Redux library. This article considered the problems of developing cross-platform applications with large amounts of data and suggested ways to solve them. The principle of operation of cross-platform applications with large amounts of data using the Redux library was studied and compared with the Redux toolkit.

Keywords: React Native, Redux, cross-platform applications, Redux toolkit.

Постановка проблеми та аналіз досліджень. Постійний ріст обсягів даних, що генеруються та оброблюються в сучасному програмному забезпеченні, ставить перед розробниками складні завдання щодо оптимізації та ефективного управління цими даними. Одним із найважливіших аспектів є підтримка різних платформ, оскільки користувачі використовують різні пристрої з різними операційними системами. Крім того, необхідно забезпечити швидкий та ефективний доступ до великих обсягів даних без зайвого навантаження на пристрої користувача. Проведено дослідження щодо використання бібліотеки Redux для керування станом додатка та оптимізації роботи з великими обсягами даних, але до цього часу не проведено аналізу та порівняння різних підходів до роботи з даними в кросплатформених додатках. Отже, в статті розглядається аналіз досліджень та досвіду використання Redux для роботи з великими обсягами даних в кросплатформених додатках з метою з'ясування переваг та недоліків цього підходу та визначення оптимальних практик для розробки ефективних додатків з великими обсягами даних.

Виділення невирішених раніше частин загальної проблеми

Хоча кросплатформенність додатків і може розв'язати деякі проблеми з їх сумісністю на різних платформах, збільшення обсягу даних, що обробляються, може призвести до проблем з продуктивністю та взаємодією з користувачем. Використання бібліотеки Redux може допомогти зберегти ефективність додатків з великими об'ємами даних, однак дослідження в цій області все ще не вирішують всіх питань, пов'язаних з оптимізацією продуктивності та взаємодією з користувачем в кросплатформених додатках з великими об'ємами даних. Тому дослідження зосереджені на вивченні цих частин загальної проблеми та пошуку способів їх вирішення [4].

Метою даного дослідження є вивчення принципу роботи кросплатформених додатків з великими об'ємами даних з використанням бібліотеки Redux. Крім того, метою є дослідження різних підходів до розробки кросплатформених додатків з використанням Redux, що дозволить знайти оптимальний підхід до реалізації таких додатків з великими об'ємами даних.

Результати дослідження можуть бути корисними для розробників кросплатформених додатків та інших зацікавлених осіб у галузі розробки програмного забезпечення.

Виклад основного матеріалу дослідження

Розглянуто принцип роботи кросплатформених додатків з великими об'ємами даних з використанням бібліотеки Redux. Бібліотека Redux використовується для керування станом додатку, що дозволяє зменшити кількість коду та зробити додаток більш простим для розуміння

та розширення [2].

Кросплатформність (або багатоплатформність, мультиплатформність) – властивість програмного забезпечення працювати більш ніж на одній програмній (в тому числі операційній системі) або апаратній платформі, та технології, що дозволяють досягти цієї властивості.

Кросплатформність дозволяє суттєво скоротити витрати на розробку нового або адаптацію існуючого програмного забезпечення. Залежно від засобів реалізації, кросплатформність поділяється на

- кросплатформність на рівні мов програмування (а також інструментів таких мов: компіляторів та редакторів зв'язків);
- кросплатформність на рівні середовища виконання;
- кросплатформність на рівні операційної системи;
- кросплатформність на рівні апаратної платформи.

Кросплатформні додатки стали дуже популярними за останні роки, оскільки дозволяють розробляти додатки, що можуть працювати на різних операційних системах. Однак, великі об'єми даних можуть стати проблемою для кросплатформних додатків, оскільки вони можуть призвести до зниження продуктивності та затримок у роботі додатку. У даному дослідженні проаналізовані проблеми, що виникають при роботі з великими об'ємами даних в кросплатформних додатках та запропоновані методи використання бібліотеки Redux для їх вирішення (рис. 1).

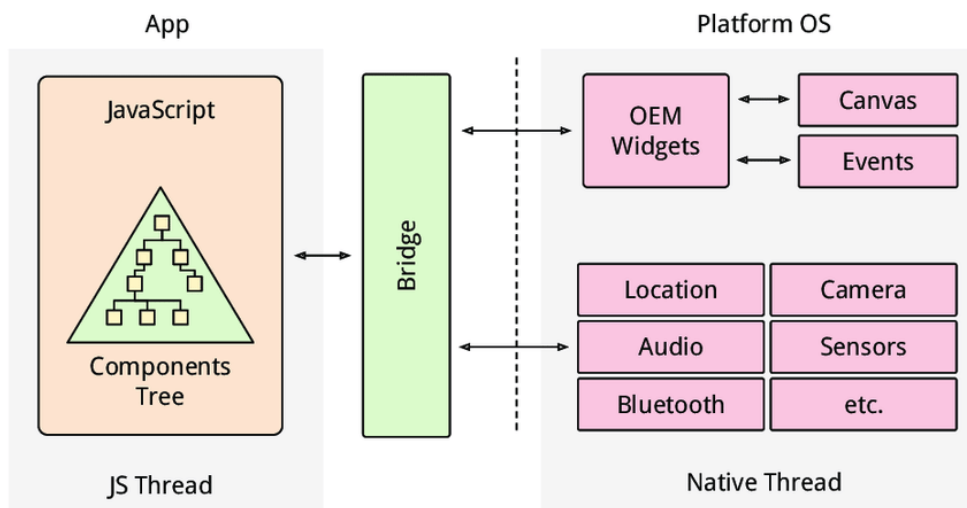


Рис. 1. – Архітектура кросплатформного додатку React Native

Крім того, подальші дослідження зосереджені на дослідженні підходів для оптимізації роботи додатків з великими об'ємами даних, таких як використання пагінації, ліниве завантаження та кешування даних. Також розглядаються можливості використання інших бібліотек та фреймворків, які можуть допомогти вирішити проблему великих об'ємів даних. Дослідження також включає порівняння різних підходів до роботи з великими об'ємами даних у кросплатформних додатках, щоб визначити найбільш оптимальний та ефективний підхід для вирішення даної проблеми [1].

Redux є бібліотекою управління станом додатку, яка забезпечує просту та прозору роботу зі станом застосунку. Вона базується на парадигмі функціонального програмування та є відкритим джерелом. Redux дозволяє відокремити стан застосунку від користувацького інтерфейсу, зберігаючи його в одному місці.

Для роботи з Redux необхідно дотримуватись певної структури додатку та використовувати певні методи, такі як «actions» та «reducers», для зміни стану. Також важливо використовувати інші бібліотеки, такі як «react-redux» та «redux-thunk», для інтеграції з фронтенд-фреймворками та забезпечення асинхронного потоку даних.

Redux toolkit – це набір інструментів для розробки Redux додатків з метою спрощення розробки та зменшення кількості написаного коду. Хоча Redux є потужним інструментом для керування станом додатків, він може бути складним для вивчення та використання, особливо для

новачків. Redux toolkit намагається усунути цю проблему, забезпечуючи набір готових шаблонів та вбудованої логіки для роботи з Redux.

Розробка додатку на Redux без використання Redux toolkit може бути досить складною та вимагати багато часу. Необхідно вручну створювати дії, редуктори та створювати зв'язки між ними. З іншого боку, Redux toolkit має багато готових функцій, таких як createSlice, яка дозволяє автоматично створювати дії та редуктори для конкретного сегмента стану додатку. Це значно зменшує кількість написаного коду та спрощує розробку [3].

Однак, якщо розробник має досвід у роботі з Redux та знає, як правильно налаштувати додаток, то він може отримати більшу гнучкість та контроль над процесом розробки за рахунок безпосереднього використання Redux. Вибір між Redux та Redux toolkit залежить від досвіду та вимог розробника та конкретних вимог проекту.

При створенні кросплатформенного додатку з великими об'ємами даних, використання Redux може значно полегшити розробку та підтримку такого додатку на різних платформах. Зокрема, Redux дозволяє забезпечити єдиний стан додатку на всіх платформах та дозволяє легко зберігати, змінювати та передавати великі об'єми даних між компонентами додатку [5].

Однак, використання Redux може мати свої недоліки, зокрема, велику кількість коду та складність відлагодження. Крім того, великі об'єми даних можуть призводити до зниження продуктивності та збільшення часу відгуку додатку. Тому важливо розглянути й інші підходи до управління станом, такі як контекст React або MobX.

Додаток зберігає всі дані в об'єктах стану, що є невід'ємною частиною бібліотеки Redux. Це дозволяє ефективно керувати даними та забезпечує зручну можливість їх відстеження та маніпулювання. Також Redux пропонує зручний інтерфейс для зміни стану додатку, що забезпечує більшу безпеку та надійність роботи програмного забезпечення.

Крім того, Redux забезпечує можливість розширення функціоналу додатку шляхом додавання нових модулів та зберігання стану у відповідних редюсерах. Це дозволяє розробникам зосередитися на окремих аспектах додатку та забезпечує зручну можливість масштабування та розвитку проекту.

Окрім того, Redux дозволяє забезпечувати більшу безпеку даних та зменшити ризик помилок при роботі з ними, що особливо важливо для додатків з великими об'ємами даних. Це досягається завдяки використанню незмінюваних об'єктів стану, що дозволяє уникнути неочікуваних змін даних та забезпечити їх консистентність (рис. 2).

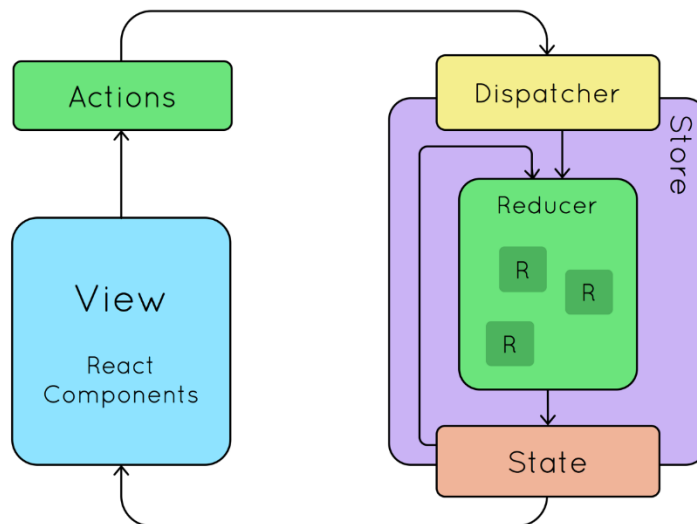


Рис. 2. – Структура Redux фреймворка

Таким чином, використання бібліотеки Redux у кросплатформенних додатках з великими об'ємами даних є важливим кроком у забезпеченні ефективної та безпечної роботи програмного забезпечення. Описані вище переваги та можливості бібліотеки дозволяють забезпечити необхідну масштабованість, надійність та зручність управління даними в додатках з великими об'ємами даних. Крім того, бібліотека Redux забезпечує зручний інтерфейс для роботи зі станом додатку, що спрощує розробку і підтримку коду. Це дає можливість розробникам ефективно

© Багнюк Н.В., Бортник К.Я., Лавренчук С.В., Домарацький І.В.

працювати з великими об'ємами даних без збоїв у роботі програми [8].

Кросплатформенні додатки з використанням Redux можуть бути запущені на різних платформах, таких як веб, мобільні пристрої та настільні комп'ютери. Це забезпечує широку аудиторію користувачів та розвиток бізнесу.

Однак, при дослідженні також виявлено, що при роботі з великими об'ємами даних, може виникати проблема зі швидкістю роботи додатку. Ця проблема може бути вирішена за допомогою оптимізації коду та використання спеціальних бібліотек для роботи з великими об'ємами даних. Крім того, бібліотека Redux дозволяє використовувати механізм middleware для здійснення додаткової обробки даних перед їх збереженням у store. Це дає можливість розширювати функціональність додатка та дозволяє використовувати додаткові бібліотеки та інструменти для обробки даних.

Загальна концепція Redux дозволяє ефективно працювати з великими об'ємами даних та забезпечує їх надійне збереження та маніпулювання. Використання цієї бібліотеки у кросплатформенних додатках дозволяє створювати високоефективні програми, які забезпечують швидку та стабільну роботу незалежно від операційної системи, на якій вони запущені [6].

Одним з основних плюсів використання Redux є можливість декомпозиції додатку на незалежні компоненти та взаємодії між ними шляхом передачі стану через store. Це дозволяє зберігати та керувати станом додатку в одному місці, зробити його більш прогнозованим та контрольованим.

Також Redux має вбудовану підтримку асинхронних дій та можливість використання middleware для обробки дій, що збільшує гнучкість та розширюваність додатку.

Важливим аспектом є також можливість тестування коду, що використовує Redux, завдяки його простоті та однорідності. Тестування допомагає забезпечити якість додатку та виявити можливі помилки в ранній стадії розробки.

Загалом, використання бібліотеки Redux дозволяє зменшити складність додатку, зберігати та управляти станом додатку в одному місці, забезпечити більш прогнозований та контрольований код, підвищити його гнучкість та розширюваність та забезпечити якість за допомогою тестування (рис. 3).

```
export const counterSlice = createSlice({
  name: 'counter',

  initialState: {
    count: 0,
  },

  reducers: {
    up: state => {
      state.count += 1
    },
    down: state => {
      state.count -= 1
    }
  }
})
```

Reducers are functions that take the current state and change it

Functions in this reducers object are auto-exported as "actions" that can be used later

This LOOKS like a mutating change but since redux-toolkit uses immer, it's actually a non-mutating change

Рис. 3 – Структура Redux slice

Для кращого розуміння принципу роботи кросплатформенних додатків з використанням бібліотеки Redux, розглянемо приклад. Нехай розробляється додаток для онлайн-магазину з великою кількістю продуктів та складів. За допомогою бібліотеки Redux можна зберігати всю інформацію про продукти та склади в глобальному стані нашого додатку. Щоразу, коли користувач робить запит на сайті, додаток буде завантажувати дані з цього глобального стану

замість того, щоб завантажувати їх з сервера знову та знову. Це значно зменшує навантаження на сервер та забезпечує швидшу роботу додатку [7].

Крім того, якщо користувач змінює якусь інформацію, наприклад, кількість одиниць товару на складі, ця зміна буде зберігатися в глобальному стані та оновлюватиметься на всіх сторінках додатка. Це забезпечує єдину точку доступу до даних та запобігає можливим проблемам синхронізації.

Висновки та перспективи подальших досліджень

Досліджено принцип роботи кроссплатформених додатків з великими об'ємами даних з використанням бібліотеки Redux та порівняно його з Redux Toolkit. В результаті аналізу можна зробити висновок, що Redux є потужним інструментом для керування станом додатку та дозволяє ефективно працювати з великими об'ємами даних. Однак, Redux Toolkit надає більш простіший та зрозуміліший підхід для розробки додатків з використанням Redux.

Подальші дослідження можуть бути спрямовані на вдосконалення підходів розробки додатків на Redux та на дослідження використання Redux Toolkit в більш складних додатках з великими об'ємами даних. Також можна дослідити використання Redux в поєднанні з іншими бібліотеками та фреймворками, що дозволяє більш гнучко працювати зі станом додатку та збільшує його швидкодію. За результатами дослідження можна стверджувати, що Redux та Redux Toolkit є потужними інструментами для розробки кроссплатформених додатків з великими об'ємами даних та дозволяють значно спростити та прискорити розробку таких додатків.

Список бібліографічного опису

1. Redux. (2022). Отримано з <https://redux.js.org/>
2. Abramov, D. (2015). Початок роботи з Redux. Отримано з <https://egghead.io/courses/getting-started-with-redux>
3. Kang, J., Kim, Y., & Kim, D. (2017). Дослідження впровадження кроссплатформеної мобільної програми з використанням react-native. Журнал Товариства корейського академічного та промислового співробітництва, 18(4), 155-161.
4. Lee, J., & Yoo, C. (2018). Потік даних у реальному часі на основі Redux для веб-додатків із Firebase і React. Журнал Корейського товариства академічної та промислової співпраці, 19(11), 45-53.
5. Gallo, A., Hester, B., & Treude, C. (2020). Вплив Redux на якість коду: емпіричне дослідження додатків React Native. Емпірична інженерія програмного забезпечення, 25(6), 3942-3976.
6. Cheng, P., Yang, L., Liu, X., Chen, Q., & Wang, Z. (2019). Дослідження оптимізації кроссплатформної системи розробки додатків на основі продуктивності. У 2019 році 3-я Міжнародна конференція з комп'ютерних наук та штучного інтелекту (CSAI 2019) (pp. 507-511). Atlantis Press..
7. Li, Y., & Lin, K. (2019). Дослідження оптимізації кроссплатформної розробки додатків на основі React Native. У 2019 році Міжнародна конференція з комп'ютерних наук та програмної інженерії (CSSE) (pp. 109-112). IEEE..
8. Liu, H., Wang, Y., Ma, Z., & Liu, Q. (2020). Дослідження кроссплатформної мобільної розробки на основі Weex. Journal of Physics: Серія конференцій, 1704(1), 012078.

References

1. Redux. (2022). Retrieved from <https://redux.js.org/>.
2. Abramov, D. (2015). Getting Started with Redux. Retrieved from <https://egghead.io/courses/getting-started-with-redux>
3. Kang, J., Kim, Y., & Kim, D. (2017). A study on the implementation of cross-platform mobile application using react-native. Journal of the Korea Academia-Industrial cooperation Society, 18(4), 155-161.
4. Lee, J., & Yoo, C. (2018). Redux-based Real-time Data Flow for Web Application with Firebase and React. Journal of the Korea Academia-Industrial cooperation Society, 19(11), 45-53.
5. Gallo, A., Hester, B., & Treude, C. (2020). Redux's impact on code quality: an empirical study on React Native applications. Empirical Software Engineering, 25(6), 3942-3976.
6. Cheng, P., Yang, L., Liu, X., Chen, Q., & Wang, Z. (2019). Research on the Optimization of Cross-platform Application Development Framework Based on Performance. In 2019 3rd International Conference on Computer Science and Artificial Intelligence (CSAI 2019) (pp. 507-511). Atlantis Press..
7. Li, Y., & Lin, K. (2019). A Study on the Optimization of Cross-platform App Development Based on React Native. In 2019 International Conference on Computer Science and Software Engineering (CSSE) (pp. 109-112). IEEE..
8. Liu, H., Wang, Y., Ma, Z., & Liu, Q. (2020). Research on Cross-platform Mobile Development Based on Weex. Journal of Physics: Conference Series, 1704(1), 012078.