

ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ СТЕЙТ-МЕНЕДЖЕРОВ

Петрова А.В. (Университет ИТМО)

Научный руководитель – преподаватель, кандидат педагогических наук Государев И.Б.
(Университет ИТМО)

Введение. Современные одностраничные веб-приложения (SPA) требуют эффективных инструментов управления состоянием, так как от этого зависит производительность приложения, удобство разработки и поддержки кода. Существуют различные стейт-менеджеры, такие как Redux, MobX, Zustand, Vuex и Effector [1-12], каждый из которых имеет свои особенности. Однако разработчики часто сталкиваются с трудностями при выборе оптимального инструмента [2]. Для решения этой проблемы в данной работе был проведен экспериментальный анализ, направленный на оценку эффективности различных стейт-менеджеров по метрикам производительности и использования памяти.

Основная часть. В ходе исследования были разработаны тестовые приложения с интеграцией пяти стейт-менеджеров. Тесты включали рендеринг списков и матриц, выполнение операций сортировки и фильтрации данных, обработку асинхронных операций и множественные подписки на одно состояние. Измерения проводились в стандартизированной среде с использованием Chrome DevTools, а для объективности результатов каждый тест выполнялся по 5 раз. Полученные данные позволили выявить различия в скорости выполнения операций, времени рендеринга пользовательского интерфейса и объеме используемой памяти для каждого стейт-менеджера.

Выводы. Результаты показали, что Redux остаётся самым популярным инструментом, но имеет высокие накладные расходы. MobX обеспечивает высокую скорость обработки состояния, но потребляет больше памяти. Effector продемонстрировал баланс производительности и простоты использования. Zustand оказался наиболее эффективным по потреблению памяти, но уступает в обработке сложных сценариев. Vuex хорошо интегрирован с Vue, но менее универсален. Таким образом, выбор стейт-менеджера должен основываться на требованиях проекта: Redux подойдет для сложных и долгосрочных приложений, MobX и Zustand — для динамических интерфейсов, Vuex — для Vue-приложений, а Effector — для задач, где важна минимальная сложность и высокая производительность.

Список использованных источников:

Научная и методическая литература

1. Le T. Comparing State Management Between Redux And Zustand In React // International Journal of Scientific Research in Computer Science Engineering and Information Technology – p. 211-218 – 2023.
2. Evergreen P. Selecting a state management strategy for modern web frontend applications // Tampere University – 2023.
3. Luu T. D. Comparison of State Management Solutions for Angular Application // Metropolia University of Applied Sciences – 2019.
4. Kumpulainen T. Web application development with Vue. js. // Information and Communication Technology – 2021.
5. Tran N. Applying vue. js framework in developing web applications // Lahti university of applied science LDT – 2020.

6. Caspers M. K. React and redux // Rich Internet Applications w/HTML and Javascript. – 2017. – Т. 11.

7. Krutika Patil. Redux State Management System - A Comprehensive Review // International Journal of Trend in Scientific Research and Development (IJTSRD). – 2022. –Volume 6 Issue 7 – С. 1021 – 1021.

Электронные ресурсы

8. Redux Documentation. redux.js.org [Электронный ресурс]. – Режим доступа: <https://redux.js.org/>

9. Vuex Documentation. vuex.vuejs.org [Электронный ресурс]. – Режим доступа: <https://vuex.vuejs.org/>

10. Zustand Documentation. zustand-demo.pmnd.rs [Электронный ресурс]. – Режим доступа: <https://zustand-demo.pmnd.rs/>

11. Effector Documentation. effector.dev [Электронный ресурс]. – Режим доступа: <https://effector.dev/>

12. MobX Documentation. [mobx.js.org](https://mobx.js.org/README.html) [Электронный ресурс]. – Режим доступа: <https://mobx.js.org/README.html>