УДК 004.056.53

ПІДХОДИ УПРАВЛІННЯ СТАНОМ У МІКРОФРОНТЕНД КОМУНІКАЦІЯХ

STATE MANAGEMENT APPROACHES IN MICRO-FRONTEND COMMUNICATIONS

Oleksandr Stepanov, Halyna Klym

Lviv Polytechnic National University, 12, Bandera Str, Lviv, 79013, Ukraine

State management involves organizing and updating the data and state of a software application, ensuring smooth functionality and a good user experience. It includes techniques like local and global state management, and state containers. In web applications, state management enables dynamic, interactive pages by managing data on the client-side (using cookies, local storage, session storage) and server-side (using session management, caching, databases). Client-side management offers fast data retrieval but has storage limits and security concerns, while server-side provides robust storage. Popular state management tools include Redux, React Context API, MobX for client-side, and Node.js, Django, Flask for server-side.

Using state management between micro-frontends can undermine the benefits of micro-frontend architecture by making independent management and development difficult. While state management is crucial for managing shared state across components or pages in a web application, extending it to micro-frontends can create dependencies, complicating maintenance and updates.

To handle state management within micro-frontends effectively, several strategies can be implemented. Local storage allows web applications to store data in the browser, enabling each micro-frontend to access and update shared state variables. This method is simple and efficient but has storage limitations and potential security concerns. A shared API utility micro-frontend caches all fetch/XHR requests and their responses, reducing network requests and improving performance. This centralizes data fetching and ensures consistency across micro-frontends. Using the Single SPA framework, state can be shared between micro-frontends by exporting functions from one application and importing them in another. This method allows direct access to and modification of shared state. Custom browser events enable state management by dispatching and listening for events between micro-frontends. This approach allows communication without direct coupling but can become complex in larger applications. Libraries like Zustand and redux-micro-frontend provide structured solutions for managing state across micro-frontends, offering features for synchronization and state sharing, enhancing consistency and scalability.

Suggested combine mixed approaches for effective state management in micro-frontends, maintain local state within each micro-frontend to ensure modularity and independence; implement an event-driven approach for managing shared state and facilitating inter-frontend communication; and utilize Module Federation to share common utilities and state management logic across micro-frontends, thereby combining the strengths of each method to create a scalable, efficient, and maintainable micro-frontend architecture.

Bibliography

1. Lee H., Kim J. Adapting Frontend Frameworks for Micro Frontend Architectures. *Journal of Modern Web Development*. 2023. Vol. 31(1). P. 55-74.
2. Micro Frontends Architecture. Plain Concepts. URL: https://www.plainconcepts.com/microfrontends/ (accessed 16.11.2023).