

РОЗРОБКА АРХІТЕКТУРИ ІНТЕРФЕЙСУ КОРИСТУВАЧА УНІВЕРСАЛЬНОЇ СИСТЕМИ НА ПРИКЛАДІ ФАЙЛОВОГО СХОВИЩА

Лавренів Д.В., Філатова Г.Є.

*Національний технічний університет
«Харківський політехнічний інститут», м. Харків*

В сучасній ІТ-індустрії при розробці програмних продуктів велика кількість роботи виконується так, що архітектура кожного продукту реалізована повністю під потреби конкретних вимог. Але такі розробки дуже складно, а іноді і не можливо розширювати або переносити архітектурні рішення між продуктами. Через це розробники виконують багато зайвої роботи, створюючи архітектуру повторно кожний раз.

Для рішення даної проблеми було розроблено систему на прикладі файлового сховища. В даному прикладі система використовує мову JavaScript для функціонування інтерфейсу користувача, що дозволяє легко розробити її та використовувати у багатьох сучасних браузерях.

Для коректної роботи, можливості розширення та легкої розробки інтерфейс створений на основі архітектурних патернів MVP та REST, налагоджено роутінг та створено систему обміну повідомлень між компонентами сторінки. Це дозволяє легко створювати нові сторінки та контролювати дані, розробляти модулі програми та сервісів, які не залежать один від одного, що дає можливість легкого розширення програми, зміни функціоналу чи навіть видалення зайвих модулів без втрати працездатності системи. Presenter (Представник) відображає тільки одну сторінку і реалізує всю бізнес-логіку для неї завдяки патерну MVP. Кожний такий представник діє незалежно від інших, кожна сторінка не залежить від інших сторінок. Якщо розробник хоче розширити інтерфейс користувача, він повинен лише створити сторінку та представника для надання їй функціональності, а також додати його у схему роутінгу для коректної маршрутизації між сторінками.

Інтерфейс користувача зв'язаний з серверною частиною за допомогою декількох REST Endpoint (кінцевих точок). Кожна така кінцева точка відповідає за конкретну дію серверної частини. Завдяки цьому вдалося досягти атомарності та цілісності операцій при роботі з базою даних. Кожна окрема операція інтерфейсу користувача зв'язана тільки з однією кінцевою точкою, тому неможливо щоб невірні дані відправлялися на точки, які не призначені для цих даних. Це все призводить до того, що система також стає безпечною для даних користувача, відмовостійкою через атомарність операцій та архітектуру, яка реагує однаково коректно як на правильні, так і на неправильні дії користувача.

Таким чином була запропонована архітектура інтерфейсу користувача універсальної системи на прикладі файлового сховища, яка легко розширюється, має незалежні модулі, що дозволяє створювати різні програмні продукти на основі розробленої архітектури.