

РОЗРОБКА АРХІТЕКТУРИ СЕРВЕРНОЇ ЧАСТИНИ УНІВЕРСАЛЬНОЇ СИСТЕМИ НА ПРИКЛАДІ ФАЙЛОВОГО СХОВИЩА

Фомічов А.І., Філатова Г.Є.

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

Сфера ІТ-індустрії зараз розвивається великими кроками, з'являються нові технічні рішення, які стають частиною нашого життя та допомагають нам жити, проводити час та комфортно себе почувати у сучасному світі. Існує багато компаній та стартапів, які розробляють програмне забезпечення для різних сайтів, сервісів чи мобільних девайсів. Але велика кількість роботи виконується так, що архітектура кожного продукту реалізована повністю під потреби конкретних вимог. Дуже мало розробників створюють свою архітектуру такою, щоб її можна було легко розширювати та доповнювати. Через це люди 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 Java для функціонування серверної частини, що дозволяє легко розробити її та використовувати у багатьох випадках. Для коректної роботи, можливості розширення та легкої розробки серверна частина створена на основі архітектурного патерну REST, методології DDD та принципів об'єктно-орієнтованого програмування. Серверна частина реалізована таким чином, що REST-контролери відповідають за зв'язок інтерфейсу користувача з серверною частиною та являють собою сервісний рівень. Кожний REST контролер отримує дані з інтерфейсу користувача, формує на їх основі команди, запускає процес виконання даної команди, а результат виконання команди повертає на інтерфейс користувача. Якщо виникає помилка при виконанні, процес передає відповідне повідомлення до контролеру, і контролер передає на інтерфейс користувача відгук, який містить код помилки та її текст. Інтерфейс користувача зв'язаний з серверною частиною за допомогою відповідних REST Endpoint (кінцевих точок). Кожна така кінцева точка відповідає за конкретну дію серверної частини. Архітектура розроблена таким чином, що вона має вигляд команди, або процесу, або результату, що повертається. Завдяки цьому вдалося реалізувати принцип транзакцій для різних операцій при роботі з базою даних. Це означає, якщо на якому-небудь із етапів виявлена помилка, то проводиться повний відкат, тому серверна частина однаково коректно реагує як на помилкові, так і на правильні дії користувача. Система стійка до відмов, оскільки усі варіанти помилок передбачені.

Таким чином, вдалося розробити серверну частину системи такою, що різні модулі програми ніяк не пов'язані між собою, є можливість легко розширювати функціонал, прибирати непотрібні модулі або міняти існуючі. 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, користувачеві буде повідомлено про неправильну дію з його сторони.