

ПОРІВНЯННЯ BACKEND ТА FRONTEND ТЕСТУВАННЯ У ВЕБ-ЗАСТОСУНКАХ

Прокоп'єв С.А.

Науковий керівник – к.н.т., доц. Тітова О.В.

Харківський національний університет радіоелектроніки, каф. ІНФ,
м. Харків, Україна,

тел.: (057) 702-14-19, e-mail: stepan.prokopiev@nure.ua

This work is devoted to comparing backend and frontend testing, analysing their strong and weak sides, their use in ensuring the comprehensive test coverage of the product. The focus will be on both manual and automation aspects of testing and how we can make sure that the critical functionality works as expected to provide the best user experience and not harm firm's reputation and business operations. As an example, the checkout charging process will be discussed as a part of this work.

Для перевірки того, що програмний продукт працює згідно з очікуваннями та є доступним до усіх користувачів існує багато різних видів тестування. Прикладами можуть слугувати: functional, API, smoke, regression, accessibility, security, performance, compatibility види тестування, тощо [1, 2]. В роботі розглядається два основні види тестування щодо автоматизації тестових сценаріїв:

- Тестування Frontend, або End-to-End (E2E) тести.
- Тестування Backend, або Application Programming Interface (API) тести.

Тестування – це процес, який складно зробити універсальним, різні проекти мають свої специфічні потреби, різні власники продуктів (product owners) мають своє бачення та побажання щодо автоматизації, саме тому часто можуть повставати проблеми з імплементацією авто-тестів.

Також іноді складно обрати який саме тип тестів обрати для покриття певного функціоналу. Кожен зі згаданих вище видів має свої переваги та недоліки [3].

Головною перевагою написання E2E тестів є те, що вони симулюють реальні user-flows, які будуть відповідати потенційним діям користувачів. Щодо недоліку, то такий тип тестування займає більше часу, адже відбувається інтеракція з усіма UI елементами.

В свою чергу перевагою API тестів є час та швидкість виконання тестових сценаріїв. Недоліком, як зазначено вище є те, що це не відповідає діям реальних користувачів, адже відсутня взаємодія з Frontend частиною.

Щодо технічного аспекту, то можна зазначити, що для написання E2E з використанням best practises потрібно імплементувати Page Object Model, а для цього потрібні знання з об'єктно-орієнтованого програмування. Також потрібно розуміти, що таке Document Object Model, які типи селекто-

рів існують та які є найбільш стабільними. А задля написання API тестів знадобляться знання з How The Web Works, які існують HTTP методи, та для яких цілей вони використовуються [4]. Також існує велика ймовірність зіткнення з асинхронною поведінкою виконання коду [5].

Тож який тип авто-тестів є кращим, та який слід використовувати? Насправді, однозначної відповіді на це запитання нема, адже це залежить від великої сукупності різноманітних факторів. Здебільше на проектах поєднують ці два типи тестування. Особливо це важливо для критично важливого функціоналу компанії з точки зору бізнесу, наприклад, коли справа стосується фінансової складової.

Уявімо, що користувач хоче купити певний товар на сайті і має промокод, який дає певну знижку. Після введення промокоду користувач бачить оновлену ціну, яка відображається на сторінці, покупає товар і помічає, що з картки було стягнуто початкову вартість товару без застосованого промокоду.

Тобто в той час, коли на Frontend'і було відображене актуальне значення, то Backend "забув" змінити вартість та надіслав запит на покупку й передав разом з ним початкову ціну. Відповідно саме таке значення й було списано з картки. В такому разі гарно покрити даний тестовий сценарій й з боку Frontend, й з боку Backend, щоб переконатись в тому, що функціонал працює належним чином. Адже якщо в Backend запиті надсилається вірна ціна зі знижкою, а на Frontend відображається початкова ціна, то це призведе до недовіри користувачів і негативно вплине на репутацію компанії. Таким чином перевірка лише Frontend'у або лише Backend'у в даному разі не гарантує якісного тестового покриття.

Такі випадки все ще зустрічаються й тестувальникам потрібно докласти зусиль, щоб запевнитись, що в першу чергу критичний функціонал працює згідно до вимог.

Список використаних джерел:

1. Всі відомі типи тестування. URL: <https://dou.ua/forums/topic/40666/> (дата звернення 23.02.2024).
2. Тітова О.В. Веб-системи в освіті: основні методи та принципи проведення юзабіліті-тестування / С. В. Тітов, О. В. Тітова // Експертні оцінки елементів навчального процесу: XVII межвузівська науч.-практ. конференція. – Х. : ХГУ «НУА», 2015. – С. 90–92.
3. Tvoroshenko, I. S., & Maksimenko, H. (2021). To the question of analysis of existing mechanisms of web application testing.
4. End-to-End тестування: процес та переваги. URL: <https://foxminded.ua/end-to-end-testuvannia/> (дата звернення 24.02.2024).
5. Asynchronous Programming in JavaScript. URL: <https://www.freecodecamp.org/news/asynchronous-programming-in-javascript/> (дата звернення 25.02.2024).