

ТЕСТУВАННЯ WEB-ЗАСТОСУНКІВ ІЗ ВИКОРИСТАННЯМ JHIPSTER НА ПРИКЛАДІ СИСТЕМИ ДЛЯ УПРАВЛІННЯ РЕСУРСАМИ КІНОТЕАТРУ

Вінницький національний технічний університет

Анотація

У роботі розглянуто підходи до тестування веб-застосунків, створених із використанням платформи JHipster, на прикладі системи для управління ресурсами кінотеатру. Проведено аналіз засобів, що застосовуються для перевірки працездатності бекенд- та фронтенд-компонентів: JUnit і MockMvc – для модульного та інтеграційного тестування REST API, Testcontainers – для перевірки взаємодії з базою даних PostgreSQL, Cypress – для наскрізного тестування інтерфейсу користувача. Особливу увагу приділено перевірці логіки авторизації, бронювання квитків, обробки оплат, обліку товарів і обробки повідомлень через Kafka. Запропоновано структуру автоматизованого тестового покриття, що забезпечує надійність і контроль якості програмного забезпечення на всіх рівнях.

Ключові слова: JHipster, тестування, REST API, JUnit, Cypress, кінотеатр.

Abstract

The paper examines testing approaches for web applications developed using the JHipster platform, based on the example of a cinema resource management system. The study highlights methods for validating both backend and frontend components: JUnit and MockMvc for unit and integration testing of REST APIs, Testcontainers for testing PostgreSQL interactions, and Cypress for end-to-end user interface testing. Particular focus is given to verifying core processes such as authentication, ticket booking, payment handling, inventory tracking, and Kafka-based message processing. A structured automated testing strategy is proposed to ensure software reliability and quality across all system layers.

Keywords: JHipster, testing, REST API, JUnit, Cypress, cinema.

Вступ

У сучасних умовах цифрової трансформації забезпечення якості програмного забезпечення є критичним чинником для успішного впровадження інформаційних систем. Особливо це стосується веб-застосунків, які функціонують у багатокористувацькому середовищі, оперують великою кількістю даних у реальному часі та взаємодіють із зовнішніми сервісами. Системи для управління ресурсами кінотеатрів є прикладом таких застосунків, де важливою є коректність виконання бізнес-логіки, надійність обробки замовлень, стійкість до збоїв та захищеність персональних даних клієнтів.

Використання платформи JHipster [1] для розробки веб-застосунку на основі технологій Java та Spring Boot [2] забезпечує гнучку архітектуру, що дозволяє застосовувати сучасні підходи до тестування на всіх рівнях – від модульних і інтеграційних перевірок до наскрізного тестування користувацьких сценаріїв. У даній роботі розглянуто практичні аспекти тестування такого застосунку з акцентом на перевірку основної функціональності системи управління кінотеатром, зокрема бронювання квитків, обліку ресурсів, обробки оплат та управління персоналом.

Результати дослідження

Під час дослідження було проведено тестування веб-застосунку для управління ресурсами кінотеатру, створеного на основі технологій JHipster, Spring Boot [2] та React. Було реалізовано комплексний підхід, що включає кілька рівнів перевірок: модульне (JUnit, Jest), інтеграційне (Spring Test), а також наскрізне тестування інтерфейсів користувача (Cypress) [3].

Модульні тести дозволили оцінити працездатність окремих компонентів системи, таких як сервіси обробки бронювань, обчислення вартості квитків, оновлення даних про залишки сніків та

Під час інтеграційних тестів було перевірено взаємодію REST API застосунку з базою даних PostgreSQL [4], зовнішніми сервісами Kafka для асинхронних повідомлень, а також перевірено роботу механізмів JWT-автентифікації [5]. Використання Spring Test дозволило переконатися в надійності реалізації взаємодії між компонентами застосунку, виявити потенційні проблеми із транзакціями та перевірити стабільність роботи системи під навантаженням.

У рамках дослідження виконано тестування серверної частини веб-застосунку для управління ресурсами кінотеатру, реалізованого за допомогою JHipster, Spring Boot та React. Тестування охоплювало кілька аспектів, зокрема: генерацію документації Javadoc, перевірку стилю коду за допомогою Checkstyle [6], модульні та інтеграційні тести із застосуванням JUnit, а також вимірювання покриття коду за допомогою JaCoCo [7].

Проведено модульні тестування основних компонентів бекенду, зокрема сутностей, DTO-об'єктів та маперів (MapStruct) [8]. Всього виконано 87 модульних тестів, які пройшли успішно без помилок та зауважень. Це підтверджує коректність реалізації основних сутностей, таких як фільми, зали, місця, квитки, співробітники тощо.

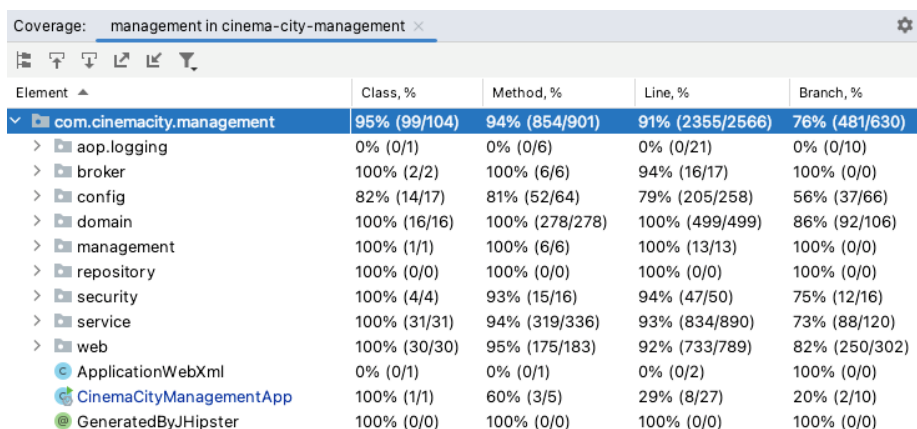
Додатково були виявлені незначні попередження MapStruct щодо неповних мапінгів полів, які не впливають критично на роботу системи, але є підставою для подальшого вдосконалення проєкту.

Результат комплексного тестування, що зображено на рисунку 1, серверної частини підтвердило належну якість реалізації веб-застосунку та готовність до подальшого впровадження у виробниче середовище.

[illegible]

Рис. 1. Результати тестування серверної частини системи управління ресурсами кінотеатру

Зокрема було проведено комплексне модульне та інтеграційне тестування серверної частини програмного продукту із використанням інструментарію, який підтримує обрана технологія розробки. Було виконано перевірку покриття тестами, яка включає виконання інформаційних, стилістичних, модульних, документаційних і інтеграційних тестів за допомогою Maven та JUnit. Результати покриття тестами програмного коду серверної частини наведено на рисунку 2.



Element	Class, %	Method, %	Line, %	Branch, %
com.cinemacity.management	95% (99/104)	94% (854/901)	91% (2355/2566)	76% (481/630)
> aop.logging	0% (0/1)	0% (0/6)	0% (0/21)	0% (0/10)
> broker	100% (2/2)	100% (6/6)	94% (16/17)	100% (0/0)
> config	82% (14/17)	81% (52/64)	79% (205/258)	56% (37/66)
> domain	100% (16/16)	100% (278/278)	100% (499/499)	86% (92/106)
> management	100% (1/1)	100% (6/6)	100% (13/13)	100% (0/0)
> repository	100% (0/0)	100% (0/0)	100% (0/0)	100% (0/0)
> security	100% (4/4)	93% (15/16)	94% (47/50)	75% (12/16)
> service	100% (31/31)	94% (319/336)	93% (834/890)	73% (88/120)
> web	100% (30/30)	95% (175/183)	92% (733/789)	82% (250/302)
ApplicationWebXml	0% (0/1)	0% (0/1)	0% (0/2)	100% (0/0)
CinemaCityManagementApp	100% (1/1)	60% (3/5)	29% (8/27)	20% (2/10)
GeneratedByJHipster	100% (0/0)	100% (0/0)	100% (0/0)	100% (0/0)

Рис. 2. Результати покриття тестами серверної частини CI/CD системи управління ресурсами кінотеатру

В результаті виконання тестів усі перевірки завершилися успішно, без виявлених помилок. Загалом було виконано 305 тестів, з яких жоден не завершився помилкою чи винятком. Це свідчить про стабільність реалізованої бізнес-логіки, належне функціонування взаємозв'язків між сутностями та коректну роботу REST-контролерів. Було протестовано класи, які охоплюють ключові аспекти бізнес-домену, зокрема Ticket, Movie, Schedule, Sale, Customer, Inventory, а також сервіси, мапери та DTO-моделі.

Для перевірки покриття коду тестами використовувався модуль jасосо, який забезпечив генерацію звітів щодо охоплення. Підсумкове значення покриття для всіх класів склало 81% за рядками коду, 88,9% за методами та 56,1% за гілками умов. Це свідчить про те, що більшість логіки системи піддано автоматичному тестуванню. Особливо високого покриття досягнуто в пакетах web.rest, service, domain, security, де охоплення перевищує 90%.

Деякі пакети, зокрема aop.logging та service.api.dto, ще не мають реалізованих тестів, що відкриває можливість для подальшого підвищення надійності проекту шляхом покращення тестового покриття.

Загалом результати тестування підтверджують відповідність серверної частини вимогам до якості, забезпечують основу для впровадження CI/CD процесів та дозволяють із впевненістю перейти до розгортання і супроводу програмного продукту.

Таким чином, застосування комплексного підходу до тестування веб-застосунку на базі JHipster підтвердило його ефективність і забезпечило високий рівень якості та надійності роботи системи для управління ресурсами кінотеатру.

Висновки

У ході дослідження підтверджено ефективність застосування комплексного тестування для веб-застосунку системи управління ресурсами кінотеатру, створеного з використанням JHipster. Модульні, інтеграційні та наскрізні тести забезпечили перевірку основної бізнес-логіки, взаємодії з базою даних, механізмів автентифікації та користувацьких сценаріїв.

Результати показали високу стабільність системи: виконано понад 300 успішних тестів, а покриття коду склало понад 80%, що свідчить про надійність реалізації. Виявлені незначні зони без тестового покриття окреслюють потенціал для подальшого удосконалення.

Таким чином, обрані підходи до тестування підтвердили якість програмного продукту та його готовність до впровадження.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. JHipster Documentation. URL: <https://www.jhipster.tech/documentation-archive/v8.0.0/> (Дата звернення: 05.06.2025)
2. Spring Boot Reference Documentation. URL: <https://docs.spring.io/spring-boot/docs/current/reference/html/> – (Дата звернення: 05.06.2025)
3. Cypress Documentation. URL: <https://docs.cypress.io/> (Дата звернення: 05.06.2025)
4. PostgreSQL Official Documentation. URL: <https://www.postgresql.org/docs/> (Дата звернення: 05.06.2025)
5. Auth0. JSON Web Token Introduction. URL: <https://auth0.com/docs/secure/tokens/json-web-tokens> (Дата звернення: 05.06.2025)
6. Checkstyle – Check Java Code Style. URL: <https://checkstyle.sourceforge.io/> (Дата звернення: 05.06.2025)
7. JaCoCo Java Code Coverage Library. URL: <https://www.eclemma.org/jacoco/> (Дата звернення: 05.06.2025)
8. MapStruct Documentation. URL: <https://mapstruct.org/documentation/stable/reference/html/> (Дата звернення: 05.06.2025)

Богач Ілона Віталіївна – к.т.н., доцент кафедри автоматизації та інтелектуальних інформаційних технологій, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: ilona.bogach@gmail.com

Слободиський Владислав Віталійович – здобувач вищої освіти групи ІІСТ-21б, кафедра автоматизації та інтелектуальних інформаційних технологій, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: sloboda55@gmail.com

Bogach Ilona Vitaliivna – Associate Professor of Automation and Intelligent Information Technologies Department, Faculty of Intelligent Information Technology and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: ilona.bogach@gmail.com

Slobodyskyi Vladyslav Vitaliiovych – student of ІІСТ-21b group, Department of Automation and Intelligent Information Technologies, Faculty of Intelligent Information Technology and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: sloboda55@gmail.com