

ВИКОРИСТАННЯ CYPRESS ДЛЯ E2E ТЕСТУВАННЯ

Вінницький національний технічний університет

Анотація

У роботі розглянуто використання інструменту Cypress для автоматизованого end-to-end (E2E) тестування веб-застосунків. Актуальність теми зумовлена зростаючими вимогами до якості програмного забезпечення та необхідністю скорочення часу на його перевірку. Cypress позиціонується як сучасний, зручний та швидкий інструмент, що дозволяє ефективно реалізовувати тестування з мінімальними витратами на конфігурацію[1]. У дослідженні проаналізовано архітектуру Cypress, його ключові можливості, а також проведено порівняння з традиційними фреймворками, зокрема з Selenium. Було здійснено практичну реалізацію тестів на Cypress, що дозволило оцінити його переваги, недоліки та доцільність використання у різних типах проектів.

Ключові слова: автоматизоване тестування, теорія тестування, js, cypress, npm.

Abstract

This paper explores the use of the Cypress tool for automated end-to-end (E2E) testing of web applications. The relevance of the topic is driven by increasing demands for software quality and the need to reduce the time required for its validation. Cypress is positioned as a modern, user-friendly, and fast tool that enables efficient test implementation with minimal configuration effort. The study analyzes the architecture of Cypress, its key capabilities, and includes a comparison with traditional frameworks, particularly Selenium. A practical implementation of tests using Cypress was carried out, allowing an assessment of its advantages, limitations, and applicability across different types of projects.

Keywords: automated testing, testing theory, JavaScript, Cypress, npm.

Вступ

У сучасному світі програмне забезпечення відіграє ключову роль у багатьох сферах діяльності, включаючи бізнес, медицину, фінанси та розваги. Надійність, безпека та якість програмних продуктів мають вирішальне значення для забезпечення безперебійної роботи систем і задоволення потреб користувачів.

Зі збільшенням складності програмного забезпечення виникає необхідність у більш ефективних підходах до тестування, які дозволяють виявляти дефекти на ранніх стадіях розробки. Традиційні методи тестування, такі як мануальне тестування та класичні автоматизовані підходи, хоча й ефективні, вимагають значних ресурсів і часу. Тому автоматизація тестування, зокрема E2E тестування, стає дедалі важливішою для прискорення процесу розробки та покращення якості продукту.[3]

Одним із популярних інструментів для автоматизації E2E тестування є Cypress – сучасний фреймворк, що поєднує простоту використання, швидкість виконання тестів і потужні можливості для тестування веб-застосунків.

Метою даного дослідження є аналіз можливостей використання Cypress для автоматизованого E2E тестування, вивчення його переваг і недоліків порівняно з іншими інструментами тестування, а також оцінка його ефективності в реальних проектах.

У рамках дослідження буде проведений аналіз існуючих підходів до автоматизованого E2E тестування, розглянуті ключові особливості Cypress, а також виконані експериментальні тести для оцінки його продуктивності та функціональних можливостей.[2]

Результати дослідження

Використання Cypress для автоматизації енд-ту-енд (E2E) тестування веб-застосунків демонструє високу ефективність у забезпеченні якості програмного забезпечення. Завдяки своїм особливостям, таким як швидке виконання тестів, вбудовані інструменти для налагодження та простий синтаксис,

Однією з головних переваг Cypress є його архітектура, яка відрізняється від традиційних фреймворків для тестування, таких як Selenium. Cypress виконує тести безпосередньо у браузері, що дозволяє отримувати швидкий зворотний зв'язок, ефективно працювати з DOM-елементами та усувати проблеми, пов'язані з очікуванням завантаження сторінки.

Таблиця 1 Порівняння Cypress та Selenium

Параметр		
Архітектура	Виконує тести безпосередньо в браузері	Виконує тести через WebDriver
Швидкість	Вища, оскільки працює напяму з браузером	Нижча через взаємодію з браузером через WebDriver
Синтаксис	Простий, зрозумілий, нагадує jQuery	Гнучкий, але складніший
Підтримка браузерів	Chromium-браузери (Chrome, Edge) та Firefox	Усі основні браузери, включаючи Safari та Internet Explorer
Мобільне тестування	Відсутнє	Є підтримка мобільного тестування
Робота з багатовіконними тестами	Не підтримується	Підтримується
Автоматичне очікування елементів	Так	Ні (необхідно явно вказувати очікування)
Інтеграція з CI/CD	Добре інтегрується	Також добре інтегрується
Гнучкість для різних типів тестування	Підходить для UI та API тестування	Підходить для UI, API, мобільного та кросбраузерного тестування

Таким чином, Cypress більше підходить для швидкого UI-тестування, у той час як Selenium залишається гнучкішим рішенням для складних сценаріїв, таких як кросбраузерне та мобільне тестування.

Використання Cypress у тестуванні:

Перед початком роботи з Cypress необхідно встановити інструмент та налаштувати тестове середовище рисунк 1.

```
npm install cypress --save-dev
```

Рис. 1. Налаштування тестового середовища

Запуск Cypress у графічному режимі рисунок 2

```
npx cypress open
```

Рис. 2. Запуск Cypress у графічному режимі

Приклад простого тесту для перевірки логіну на веб-застосунку на рисунку 3.

```
describe('Перевірка логіну', () => {
  it('Логін з коректними даними', () => {
    cy.visit('https://example.com/login') // Відкриття сторінки логіну
    cy.get('input[name="username"]').type('testuser') // Введення логіну
    cy.get('input[name="password"]').type('password123') // Введення пароля
    cy.get('button[type="submit"]').click() // Натискання кнопки входу
    cy.url().should('include', '/dashboard') // Перевірка, що користувач переш
  })
})
```

Рис. 3. Приклад перевірки логіну

Тести можна запускати у двох режимах:

- GUI-режим (інтерактивний режим з візуалізацією виконання тестів).
- CLI-режим (для інтеграції з CI/CD).

Запуск тестів через командний рядок рисунок 4.

```
npx cypress run
```

Рис. 4. Запуск тестів

Cypress має вбудовану підтримку запису відео та скріншотів під час тестування. Це допомагає швидко ідентифікувати причини невдалих тестів. Також існує Cypress Dashboard, який дозволяє відстежувати результати тестування у хмарі та отримувати аналітику.[5]Cypress легко інтегрується з популярними CI/CD-системами, такими як Jenkins, GitHub Actions, GitLab CI/CD. Приклад конфігурації для GitHub Actions рисунок 5.

```
name: Cypress Tests
on: [push]
jobs:
  test:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout repository
        uses: actions/checkout@v2
      - name: Install dependencies
        run: npm install
      - name: Run Cypress tests
        run: npx cypress run
```

Рис. 5. Конфігурація для GitHub Actions

Висновки

Використання Cypress для автоматизації E2E тестування забезпечує швидке виконання тестів, простий синтаксис та ефективний механізм роботи з DOM-елементами. У порівнянні з Selenium, Cypress є зручнішим для тестування веб-інтерфейсів, але має деякі обмеження, зокрема щодо підтримки мобільного тестування та роботи з кількома вікнами.

Загалом, Cypress є відмінним вибором для команд, які працюють з React, Angular, Vue та іншими сучасними фреймворками, а також прагнуть автоматизувати тестування з мінімальними затратами на налаштування та підтримку.

СПИСОК ЛІТЕРАТУРИ

1. Introduction to Software Testing. Paul Ammann & Jeff Offutt. [Електронний ресурс] – Режим доступу: <https://cs.gmu.edu/~offutt/softwaretest/> (дата звернення: 20.03.2025).
2. Software Testing Tutorial – Guru99. [Електронний ресурс] – Режим доступу: <https://www.guru99.com/software-testing.html> (дата звернення: 20.03.2025).
3. Cypress Documentation. [Електронний ресурс] – Режим доступу: <https://docs.cypress.io/> (дата звернення: 20.03.2025).
4. Як і чому змінюється тестування і що з цим робити [Електронний ресурс] – Режим доступу: <https://blog.ithillel.ua/articles/how-and-why-testing-is-changing> (дата звернення: 20.03.2025).
5. Software Testing Introduction [Електронний ресурс] – Режим доступу: <https://learn.epam.com/detailsPage?id=744c27ca-8961-409c-9f8b-5a906b552c5b> (дата звернення: 20.03.2025).

Гуральник Фредерік Борисович – аспірант кафедри автоматизації та інтелектуальних інформаційних технологій, факультет інтелектуальних технологій та автоматизації, Вінницький національний технічний університет, м. Вінниця, email: frederikguralnik@gmail.com

Науковий керівник: **Кветний Роман Наумович** - доктор технічних наук, професор, професор кафедри автоматизації та інтелектуальних інформаційних технологій, Вінницький національний технічний університет, м. Вінниця

Guralnyk Frederic Borysovych – Postgraduate student at the Department of Automation and Intelligent Information Technologies, Faculty of Intelligent Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, email: frederikguralnik@gmail.com

Scientific advisor: **Kvetnyi Roman Naumovych** – Doctor of Technical Sciences, Professor, Professor of the Department of Automation and Intelligent Information Technologies, Vinnytsia National Technical University, Vinnytsia