



Example Project
[View all projects](#)

Latest runs

Branches

Analytics

Run status

Run duration

Test suite size

Top failures

Slowest tests

Most common errors

Flaky tests

Project settings

FILTER BY

develop

Tag

Last 3 Months

Weekly

Passed Failed Timed out Errored Canceled

Total runs over time



Огляд фреймворку Cypress

Cypress – це сучасний тестовий фреймворк, який використовується для написання, виконання та автоматизації тестів для веб-додатків. Він надає зручний інтерфейс для розробки тестів, високу продуктивність, зручну систему візуалізації, а також вбудовану підтримку для відлагодження тестів.

Total runs

359

↓ 18%

Average per day

6.30

↓ 26%

Passed runs

101

Failed runs

181



WEEK BEGINNING

PASSED

FAILED

TIMED OUT

ERRORED

CANCELED

TOTAL

FAILURE RATE

Основні особливості Cypress



Дружній інтерфейс

Простий та зрозумілий синтаксис для написання тестів



Автоматичне очікування

Cypress автоматично очікує дій користувача та подій



Відлагодження

Вбудований інструмент для зручного відлагодження тестів

Візуальне тестування

Виконання тестів у реальному часі з візуалізацією

Асинхронні операції

Гарантована відповідність виконання тестів

Робота з DOM

Простий доступ та маніпулювання елементами

Мультипристроїв

Емуляція різних пристройів для тестування

Основні складові Cypress



Mocha фреймворк

Використовує Mocha для написання тестових сценаріїв



jQuery селектори

Синтаксис селекторів, схожий на jQuery, для взаємодії з DOM-елементами

Цей тестовий фреймворк надає розробникам можливість швидко та ефективно писати тести для веб-додатків, а також легко відлагоджувати та аналізувати результати тестування.

```
it('should multiply two numbers together', () => {
    expect(appController.multiply(2, 2)).toBe(4);
    expect(appController.multiply(3, 2)).toBe(6);
    expect(appController.multiply(2, 3)).toBe(6);
    expect(appController.multiply(0, 1)).toBe(0);
});

it('should divide the first number with the second', () => {
    expect(appController.divide(2, 2)).toBe(1);
    expect(appController.divide(3, 4)).toBe(0.75);

    expect(appController.divide(4, 3)).toBe(1.3333333333333333);
    expect(appController.divide(4, 3)).not.toBe(1.3);

    expect(appController.divide(1, 0)).toBe(Infinity);
});
```

Chai бібліотека

Зручна перевірка тестових умов



Cypress Dashboard

Платформа для збереження результатів тестів та їх аналізу

Основні аспекти Cypress



Простота використання

Інтуїтивний API, зрозумілі команди

Автоматичне очікування

Не потрібно вставляти чеки в тестах

Висока продуктивність

Вбудована паралельність тестів

Відлагодження

Точки зупинки, спостереження змінних

Візуальне тестування

Перевірка візуальних змін у реальному часі під час виконання тестів

Робота з асинхронністю

Правильна послідовність виконання тестів з асинхронними операціями

Області застосування Cypress

Функціональне тестування

Перевірка роботи всіх функцій та інтерфейсу додатку

Тестування юзабіліті

Перевірка узгодженості інтерфейсу та користувацького досвіду

Регресійне тестування

Збереження надійності під час розширення додатків

Універсальність

- Невеликі веб-сайти
- Складні веб-додатки
- Проекти будь-якого масштабу
- Різні рівні складності

- Інтеграційне тестування:** Використовуючи Cypress, ви можете виконувати тестування взаємодії між різними компонентами або сервісами у вашому веб-додатку.



**Types of tests
that can be automated**

Рівні тестування в Cypress



End-to-End тести

Повне відтворення взаємодії користувача з веб-додатком від початку до кінця



Компонентні тести

Ізольоване тестування окремих компонентів та їх функціоналу

Cypress надає можливість проведення тестів на різних рівнях, що дозволяє розробникам обирати оптимальний підхід залежно від потреб проекту. Кожен рівень має свої особливості та переваги для забезпечення якості веб-додатків.

End-to-End

10^N of 1m Web Tests

Integration

10^{N+1} of 1s Service Tests

Unit

10^{N+2} of 1ms Code Tests

End-to-End (E2E) тести

Взаємодія користувача

Точне відтворення дій користувача у веб-додатку: навігація, взаємодія з елементами, перевірка стану

Візуальне тестування

Проведення візуальних тестів у реальному часі



Автоматичне очікування

Вбудована синхронізація очікує стану елементів перед виконанням дій

Збереження стану

Підтримка стабільності між кроками тестування

Компонентні тести



Ізольоване тестування

Перевірка функціоналу окремих компонентів незалежно від інших частин додатку



Маніпулювання DOM

Зручний спосіб роботи з DOM-елементами у компонентних тестах



Структура компонентів

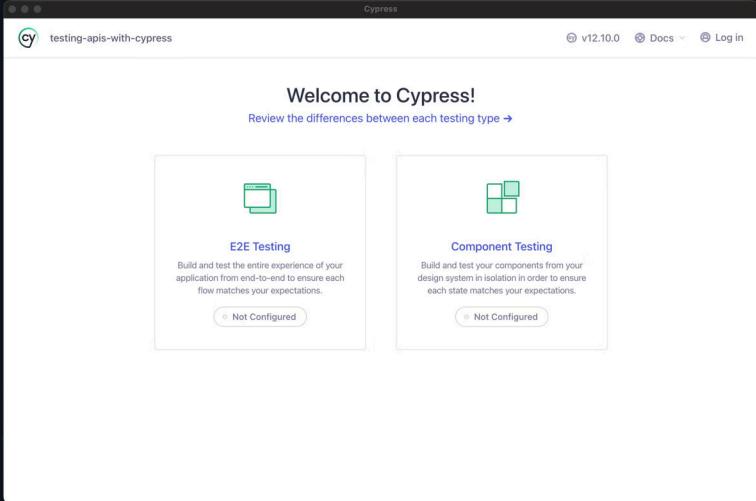
Перевірка структури, взаємодії та правильності відображення даних



Асинхронність та ізоляція

Робота з асинхронними операціями та забезпечення ізоляції коду

Висновок



Універсальність Cypress

Cypress надає можливості для тестування різних аспектів вашого веб-додатку як на рівні цілісних функціональностей, так і на рівні окремих компонентів.

E2E тести

Повне тестування реальної взаємодії користувача з додатком

Компонентні тести

Перевірка окремих частин функціоналу в ізольованому середовищі

Cypress забезпечує зручний і потужний інструментарій для розробників тестів, що робить його одним з найпопулярніших фреймворків для тестування сучасних веб-додатків.