

Фільтрація елементів у Cypress

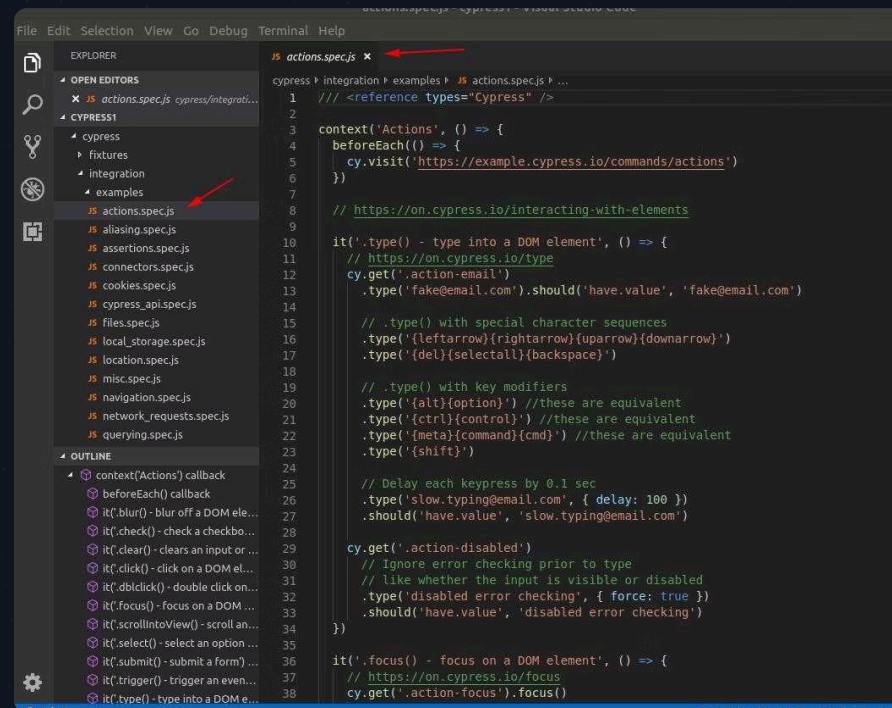
Фільтрація елементів у Cypress дозволяє вибирати певні елементи на сторінці за допомогою різних умов або критеріїв. В Cypress використовуються методи, такі як `filter()`, `find()`, та інші, щоб вибрати конкретні елементи для подальших взаємодій або перевірок.

Використання filter()

Приклад коду

```
// Вибрати всі елементи з класом 'my-class'  
cy.get('.my-class')  
  .filter(':visible')  
  .click();
```

Метод `filter()` дозволяє відфільтрувати набір елементів за певною умовою, наприклад видимістю.



Фільтрація вкладених елементів



Знайти батьківський елемент

Використовуйте
`cy.get('.parent-element')`



Фільтрувати вкладені

Метод `find('.child-element')`
знаходить дочірні елементи



Перевірити результат

Використовуйте
`should('have.length', 3)`

```
cy.get('.parent-element')  
  .find('.child-element')  
  .should('have.length', 3);
```

Фільтрація за текстом та індексом

За текстом елемента

```
// Вибрати елемент за  
текстом  
cy.contains('Кнопка').click();
```

Команда `contains()` знаходить елемент, який містить вказаний текст.

За індексом елемента

```
// Вибрати елемент за  
індексом  
cy.get('ul li')  
  .eq(2)  
  .should('have.text', 'Третій  
елемент');
```

Метод `eq()` вибирає елемент за його позицією в наборі.

ts

Export

Most popular

ject ^

Type

Inve

ner Ruby On Rails

Course

25

s new in Laravel 5.4

Episode

∞

WordPress Multi-Site

Course

30

ering UX Methodologies

Course

45

nt Oriented JavaScript

Course

24

JS Fundamentals

Course

100

typing With FramerJS

Episode

∞

Фільтрація за атрибутом та станом



Фільтрація за атрибутом

```
cy.get('[data-testid="submit-button"]').click();
```

Використовуйте селектори атрибутів для точного вибору елементів.



Фільтрація за станом

```
cy.get('input')  
  .filter(':disabled')  
  .should('have.length', 1);
```

Фільтруйте елементи за їх станом: `:disabled`, `:checked`, `:visible`.

Використання методу not()

Виключення класів

```
cy.get('.item')  
  .not('.hidden')  
  .should('exist');
```

Метод not() виключає елементи, що відповідають умові.

Невибрані чекбокси

```
cy.get('input[type="checkbox"]')  
  .not(':checked')  
  .check();
```

Вибір всіх чекбоксів, які не є активними.



Важливо: Cypress робить автоматичне очікування, тому вам не потрібно вручну додавати затримки або чекати на завантаження елементів.

Аліаси в Cypress



Аліаси (aliases) — потужний інструмент для покращення читабельності та підтримки коду тестів. Вони дозволяють надавати імена елементам і секціям тесту для легкого посилання.

01

Створення аліасу

```
cy.get('.login-button').as('loginButton');
```

02

Використання аліасу

```
cy.get('@loginButton').click();
```

03

Перевірка з аліасом

```
cy.get('@loginButton').should('be.visible');
```

04

Багаторазове використання

```
cy.get('@usernameInput').type('john_doe');  
cy.get('@usernameInput').should('have.value',  
'john_doe');
```


Robert C. Martin Series

Clean Architecture

A Craftsman's Guide to
Software Structure and Design

Robert C. Martin

With contributions by James Grenning and Simon Brown

Foreword by Kevlin Henney

Afterword by Jason Gorman

Переваги використання аліасів

Читабельність коду

Аліаси роблять код більш зрозумілим і легким для читання.

Покращена підтримка

Зміна селектора в одному місці оновлює всі посилання.

Менша залежність

Тести стають менш залежними від конкретної реалізації сторінки.

Використання аліасів в тестах Cypress значно покращує якість вашого коду, робить його більш професійним та легким у підтримці. Це особливо корисно у великих проектах з багатьма тестами.