

Підтримка CSS та XPATH у Cypress

Фреймворк Cypress має розширений набір можливостей для вибору елементів за допомогою CSS селекторів. Ці можливості дозволяють зручно та ефективно здійснювати вибір елементів на веб-сторінці для подальших дій (наприклад, клік, введення тексту, перевірка стану тощо).

Переваги CSS селекторів

- Висока швидкість виконання
- Нативна підтримка в Cypress
- Інтуїтивний синтаксис
- Широкі можливості комбінування

Основні методи

- `cy.get()` для прямого вибору
- `cy.contains()` для пошуку за текстом
- Комбінування селекторів
- Використання атрибутів `data-testid`

Ось деякі з основних методів вибору елементів за допомогою CSS селекторів у Cypress, які дозволяють створювати надійні та підтримувані автоматизовані тести для вашого веб-додатку.

Web Automation

Метод `cy.get()` для вибору елементів



Вибір за класом

```
cy.get('.my-class');
```

Використовується для отримання всіх елементів з певним CSS класом



Вибір за ідентифікатором

```
cy.get('#myId');
```

Швидкий спосіб отримати унікальний елемент на сторінці



Вибір за тегом

```
cy.get('button');
```

Вибирає всі елементи певного типу HTML-тегу

Цей метод є основним інструментом для отримання елементів на сторінці. Він приймає CSS селектор як параметр і повертає об'єкт Cypress, який містить знайдені елементи. Метод `cy.get()` підтримує всі стандартні CSS селектори, що робить його надзвичайно гнучким інструментом для тестування.

Вибір за атрибутом

```
cy.get('[data-testid=myTestId]');
```

Рекомендований підхід для створення стабільних тестів, оскільки `data-testid` не залежить від візуальних змін інтерфейсу

Вибір вкладеного елемента

```
cy.get('ul li');
```

Дозволяє знайти елементи, які знаходяться всередині інших елементів, використовуючи ієрархію DOM

Метод `cy.contains()` для пошуку за текстом

Цей метод шукає елемент з текстовим вмістом, що відповідає вказаному шаблону. Це особливо корисно, коли ви хочете знайти елемент на основі його видимого тексту, а не структури DOM або CSS класів. Метод `cy.contains()` може комбінуватися з CSS селекторами для більш точного пошуку.

01

Елемент з текстом та класом

```
cy.contains('.my-class', 'Text');
```

Знаходить елемент, який має певний клас і містить вказаний текст

02

Посилання з текстом

```
cy.contains('a', 'Link Text');
```

Шукає посилання (тег `<a>`), яке містить певний текст всередині

03

Елемент за атрибутом та текстом

```
cy.contains('[data-testid=myTestId]', 'Content');
```

Комбінує пошук за data-атрибутом і текстовим вмістом для максимальної точності



Порада: Метод `cy.contains()` особливо корисний для тестування кнопок, посилань та інших інтерактивних елементів, коли їх текст є стабільним ідентифікатором. Він робить тести більш читабельними та зрозумілими.

Коли використовувати `cy.contains()`

- Для пошуку елементів за видимим текстом
- Коли текст є стабільним ідентифікатором
- Для тестування локалізованого контенту
- При роботі з динамічними класами або ID

Переваги методу

- Читабельність тестів
- Незалежність від структури
- Природний підхід

Комбінування CSS селекторів

За допомогою комбінації CSS селекторів ви можете використовувати більш складні вирази для вибору конкретних елементів. Це дозволяє створювати точні та надійні селектори, які можуть знаходити елементи навіть у складній структурі DOM. Комбінування селекторів є потужним інструментом для написання ефективних тестів.

Вкладені елементи

```
cy.get('.parent-class .child-class');
```

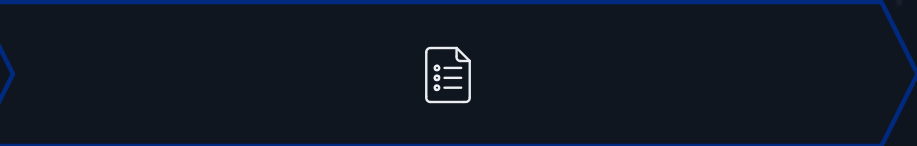
Вибір елемента, який знаходиться на будь-якому рівні всередині батьківського



Прямі дочірні елементи

```
cy.get('.parent-class > .child-class');
```

Вибір елемента, який є безпосереднім нащадком батьківського елемента



Множинний вибір

```
cy.get('.my-class, #myId');
```

Вибір елементів за декількома різними селекторами одночасно

Псевдо-класи

```
cy.get('ul li:nth-child(2)');
```

Використання CSS псевдо-класів для точного вибору елементів у списку

Важливо: Ці методи дозволяють ефективно вибирати елементи за допомогою CSS селекторів у Cypress, надаючи можливість шукати та взаємодіяти з елементами на сторінці для тестування вашого веб-додатку. Використовуйте найбільш специфічні селектори для забезпечення стабільності тестів.

Тип селектора	Приклад використання
Комбінація класів	cy.get('.class1.class2')
Атрибут з значенням	cy.get('[type="submit"]')
Сусідній елемент	cy.get('.class1 + .class2')
Перший дочірній	cy.get('ul li:first-child')

Підтримка XPath селекторів у Cypress



❏ **Зауваження:** Фреймворк Cypress не надає вбудованої підтримки для використання XPath-селекторів напряду для вибору елементів на веб-сторінці. Однак існує альтернативне рішення.

Багато засобів вибору елементів, доступних у CSS, також можуть бути використані в Cypress. Враховуючи це, найбільш рекомендованим методом для вибору елементів в Cypress є використання CSS-селекторів, оскільки вони є основним інструментом для вибору елементів в цьому фреймворку.

Плагін cypress-xpath

Якщо ви хочете використовувати XPath-селектори, ви можете скористатися сторонніми бібліотеками для цього

Статус підтримки

Плагін є deprecated, тобто більше не підтримується розробниками активно

Рекомендація

Використовуйте CSS-селектори як основний інструмент для стабільних та ефективних тестів

Встановлення cypress-xpath

Щоб використати cypress-xpath, встановіть його у ваш проект Cypress, використовуючи npm:

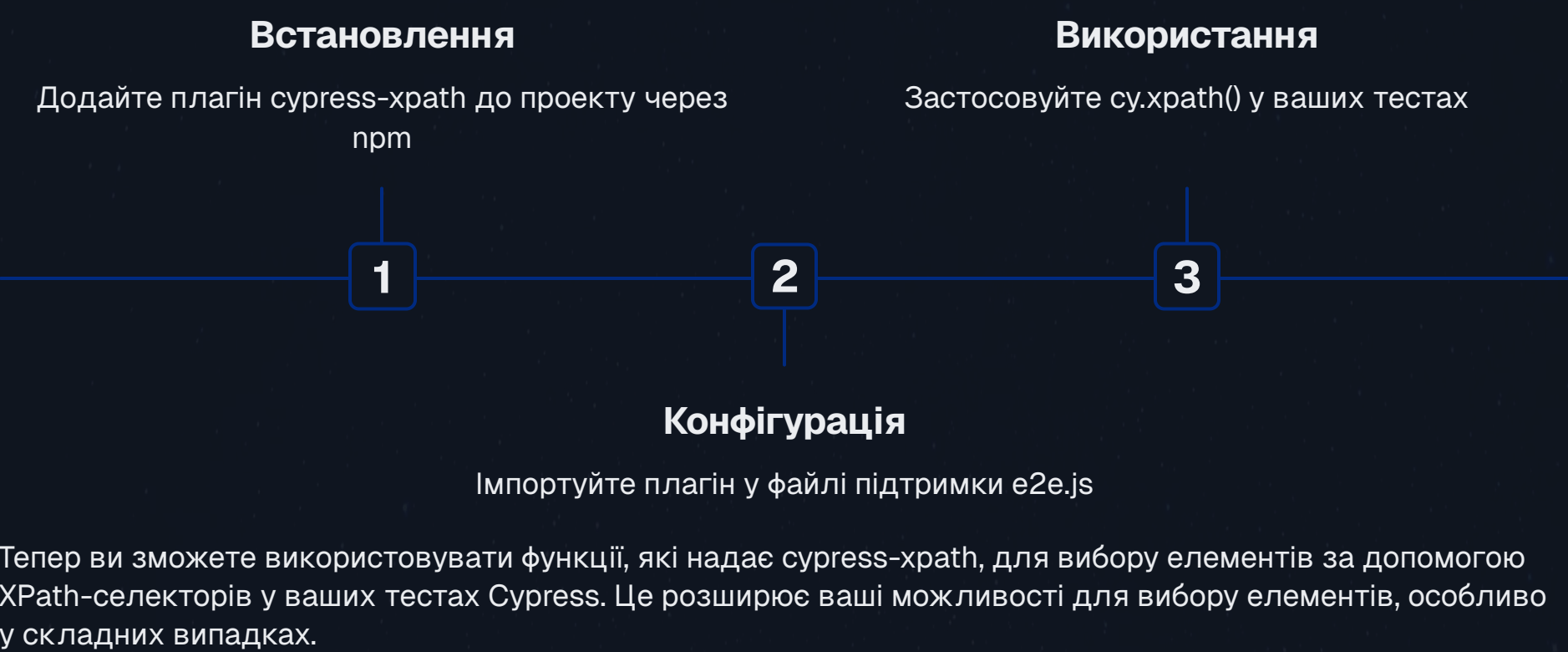
```
npm install -D cypress-xpath
```

Після встановлення додайте наступний код у вашому `cypress/support/e2e.js` файлі для налаштування плагіну:

```
import 'cypress-xpath';
```

Документація: <https://docs.cypress.io/app/core-concepts/writing-and-organizing-tests#Support-file>

Використання XPath у Cypress тестах



Приклад використання XPath

```
cy.xpath('//button[contains(text(),"Submit")]').click();
```

Коли використовувати XPath

- Коли CSS селектори не можуть досягти потрібного елемента
- При роботі зі складними ієрархіями DOM
- Для вибору елементів за текстом у складних структурах
- При міграції тестів з інших фреймворків

Обмеження XPath

- Нижча продуктивність порівняно з CSS селекторами
- Відсутність нативної підтримки в Cypress
- Плагін є deprecated
- Складніший синтаксис для підтримки

Важливе зауваження: Використання XPath може бути менш ефективним порівняно з CSS-селекторами у Cypress, тому рекомендується використовувати XPath лише у випадках, коли ви не можете досягти потрібного елемента за допомогою CSS-селекторів. CSS залишається основним та найбільш оптимальним способом вибору елементів у Cypress.

Переваги CSS

- Швидша робота
- Нативна підтримка
- Простіший синтаксис
- Краща документація

Випадки для XPath

- Складні умови пошуку
- Пошук по тексту
- Навігація по осях
- Спеціальні сценарії