**Data attributes in Cypress**

**Selecting Elements**

**Anti-Pattern:**Using highly brittle selectors that are subject to change.

**Best Practice:**Use data-\* attributes to provide context to your selectors and isolate them from CSS or JS changes.

Every test you write will include selectors for elements. To save yourself a lot of headaches, you should write selectors that are resilient to changes.

Oftentimes we see users run into problems targeting their elements because:

* Your application may use dynamic classes or ID's that change
* Your selectors break from development changes to CSS styles or JS behavior

Luckily, it is possible to avoid both of these problems.

1. Don't target elements based on CSS attributes such as: id, class, tag
2. Don't target elements that may change their textContent
3. Add data-\* attributes to make it easier to target elements

How It Works:

Given a button that we want to interact with:

<button

id="main"

class="btn btn-large"

name="submission"

role="button"

data-cy="submit"

>

Submit

</button>

Let's investigate how we could target it:

| **Selector** | **Recommended** | **Notes** |
| --- | --- | --- |
| cy.get('button').click() | Never | Worst - too generic, no context. |
| cy.get('.btn.btn-large').click() | Never | Bad. Coupled to styling. Highly subject to change. |
| cy.get('#main').click() | Sparingly | Better. But still coupled to styling or JS event listeners. |
| cy.get('[name=submission]').click() | Sparingly | Coupled to the name attribute which has HTML semantics. |
| cy.contains('Submit').click() | Depends | Much better. But still coupled to text content that may change. |
| cy.get('[data-cy=submit]').click() | Always | Best. Isolated from all changes. |

Targeting the element above by tag, class or id is very volatile and highly subject to change. You may swap out the element, you may refactor CSS and update ID's, or you may add or remove classes that affect the style of the element.

Instead, adding the data-cy attribute to the element gives us a targeted selector that's only used for testing.

The data-cy attribute will not change from CSS style or JS behavioral changes, meaning it's not coupled to the **behavior** or **styling** of an element.

Additionally, it makes it clear to everyone that this element is used directly by test code.

**Did you know?**

The [Selector Playground](https://docs.cypress.io/guides/core-concepts/test-runner#Selector-Playground) automatically follows these best practices.

When determining an unique selector it will automatically prefer elements with:

* data-cy
* data-test
* data-testid

#### **Real World Example**

        cy.visit('http://localhost:3000/signin')

        cy.get('#username').type("Allie2")

        cy.get('#password').type("s3cret")

        cy.get('.MuiButton-label').click()

        cy.get('[data-test="nav-top-new-transaction"]').click()

        cy.get('[data-test="user-list-search-input"]').type("Ibrahim")