

# Cypress – Page Object Pattern i trudne przypadki

Adam Stasiak – Automatyzacja Testowania w  
Praktyce 2023





# Page Object Pattern

Test A

```
cy.get('[data-test="product_icon"]')
  .eq(0)
  .click();
cy.get('[data-test="product_quantity"]')
  .eq(0)
  .type('1');
cy.get('[data-test="add_to_shopping_cart"]')
  .eq(0)
  .click();
cy.get('[data-test="checkout"]').click();
cy.contains('Your Shopping Cart').should('be.visible');
```

Test B

```
cy.get('[data-test="product_icon"]')
  .eq(0)
  .click();
cy.get('[data-test="product_quantity"]')
  .eq(0)
  .type('1');
cy.get('[data-test="add_to_shopping_cart"]')
  .eq(0)
  .click();
cy.get('[data-test="checkout"]').click();
cy.contains('Quantity: 1').should('be.visible');
```

Test C

```
cy.get('[data-test="product_icon"]')
  .eq(0)
  .click();
cy.get('[data-test="product_quantity"]')
  .eq(0)
  .type('1');
cy.get('[data-test="add_to_shopping_cart"]')
  .eq(0)
  .click();
cy.get('[data-test="checkout"]').click();
cy.contains('Payment method').should('be.visible');
```



# Page Object Pattern

Test A

```
listingPage.addItemToCartWithIndex(0);  
listingPage.goToCart();  
  
cart.headerIsDisplayed();
```

Test B

```
listingPage.addItemToCartWithIndex(0);  
listingPage.goToCart();  
  
cart.itemQuantityIsEqualTo(1);
```

Test C

```
listingPage.addItemToCartWithIndex(0);  
listingPage.goToCart();  
  
cart.paymentMethodIsVisible();
```

# Podjęcie 1.

Selektory

```
class HomePage {  
  elements = {  
    header: () => cy.get('#navbar'),  
    utilitiesSectionButton: () => cy.contains('Utilities'),  
  };  
  
  navigate() {  
    cy.visit('https://example.cypress.io/');  
  }  
}  
  
export default HomePage;
```

Metody dostępne w  
ramach danej strony /  
komponentu



## Podejście 1.

```
describe('Basic Page Object', () => {  
  let homePage: HomePage = new HomePage();  
  let utilitiesPage: UtilitiesPage = new UtilitiesPage();  
  it('home vs utilities page', () => {  
    homePage.navigate();  
    homePage.elements.header().should('be.visible');  
    homePage.elements.utilitiesSectionButton().click();  
    utilitiesPage.elements.sectionHeader('#').should('be.visible');  
  });  
});
```



## Podejście 2.

Definicja interfejsu

```
interface AppPage {  
  elements(): any;  
  visit(): void;  
  getHeader(): Cypress.Chainable<jQuery<HTMLElement>>;  
}
```

```
export default AppPage;
```

Metody zwracające  
ten sam typ co  
wszystkie inne  
selectory Cypressa  
operujące na  
elementach



## Podjęcie 2.

```
import AppPage from './appPage';

class HomePage implements AppPage {
  elements() {
    return {
      header: () => cy.get('#navbar'),
    };
  }
  visit(): void {
    cy.visit('https://example.cypress.io/');
  }
  getHeader(): Cypress.Chainable<JQuery<HTMLElement>> {
    return this.elements().header();
  }
}

export default HomePage;
```

Definicja klasy  
implementującej  
interfejs – edytor  
kodu sam podpowie  
nam szablony klasy

Metody właściwe dla  
danego widoku



## Podejście 2.

```
describe('Interface Page Object', () => {  
  let homePage: HomePage = new HomePage();  
  it('Header should be visible.', () => {  
    homePage.visit();  
    homePage.getHeader().should('be.visible');  
  });  
});
```

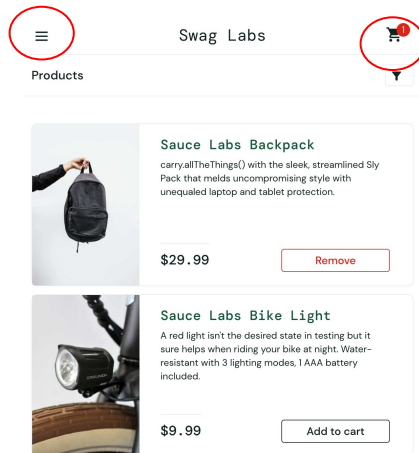


## Podejście 3.

definicja klasy  
bazowej

```
class Page {  
  elements = {  
    burgerIcon: () => cy.get('.bm-burger-button'),  
    basketIcon: () => cy.get('.shopping_cart_link'),  
  };  
  loadHomePage() {  
    cy.visit('https://www.saucedemo.com/');  
  }  
  openBurgerMenu() {  
    this.elements.burgerIcon().click();  
  }  
  openBasket() {  
    this.elements.basketIcon().click();  
  }  
}  
export default Page;
```

definicja bazowych  
elementów i metod  
(dostępnej w ramach  
całej aplikacji)



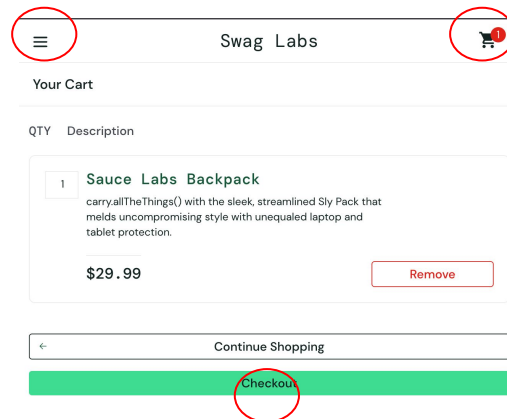


## Podejście 3.

definicja klasy  
rozszerzającej bazową

elementy właściwe  
dla danego widoku

```
class BasketPage extends Page {  
  constructor() {  
    super();  
  }  
  basketElements = {  
    checkoutButton: () => cy.get('[data-test="checkout"]'),  
  };  
  clickCheckout() {  
    this.basketElements.checkoutButton().click();  
  }  
}  
export default BasketPage;
```



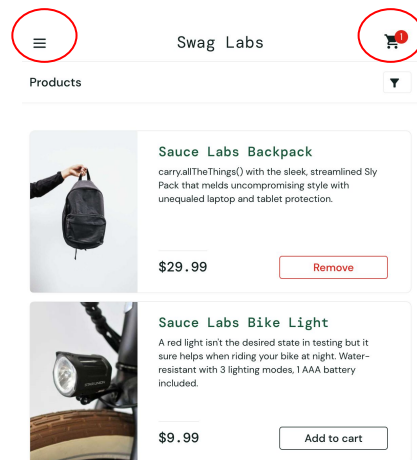


## Podójście 3.

definicja klasy  
rozszerzającej bazową

```
class ListingPage extends Page {  
  constructor() {  
    super();  
  }  
}  
  
export default ListingPage;
```

wywołanie  
konstruktora bazowej  
klasy



## Podejście 3.



**div.inventory\_item** 679 × 240

Sauce Lak

carry.allTheThings() with the sleek, streamlined Sly Pack that melds uncompromising style with unequaled laptop and tablet

```
▼ <div class="inventory_item"> flex
  ► <div class="inventory_item_img"> ... </div>
  ▼ <div class="inventory_item_description"> fl
```

\$29.99

Remove

```
class Item {
  title: string;
  constructor(title: string) {
    this.title = title;
  }
  elements = {
    container: () =>
      cy
        .contains(this.title)
        .parentsUntil('.inventory_item')
        .parent(),
    title: () => this.elements.container().find('.inventory_item_name'),
    summary: () => this.elements.container().find('.inventory_item_desc'),
  };
  bar = () => new ItemBar(this.elements.container());
}
export default Item;
```



## Podejście 3.

```
class BasketItem extends Item {  
  constructor(title: string) {  
    super(title);  
  }  
  basketSpecificElements = {  
    quantity: () => this.elements.container().find('.cart_quantity'),  
  };  
}  
  
export default BasketItem;
```

### Your Cart

 **div.cart\_item** 694 x 174

1

### Sauce Labs Backpack

carry.allTheThings() with the sleek, streamlined Sly Pack that melds uncompromising style with unequaled laptop and tablet protection.

\$29.99

Remove



Continue Shopping

Checkout

## Podejście 3.

```
class ItemBar {  
  parent: Cypress.Chainable<jQuery<HTMLElement>>;  
  constructor(parent: Cypress.Chainable<jQuery<HTMLElement>>) {  
    this.parent = parent;  
  }  
  elements = {  
    price: () => this.parent.find('.inventory_item_price'),  
    addToBasketButton: () =>  
      this.parent.find('[data-test^="add-to-cart-sauce-labs-"]'),  
    removeFromBasketButton: () =>  
      this.parent.find('[data-test^="remove-sauce-labs-"]'),  
  };  
  
  addToBasket() {  
    this.elements.addToBasketButton().click();  
  }  
}  
  
export default ItemBar;
```



**div.pricebar** 415.34 × 42

\$29.99

Remove

## Podejście 3.

```
class ListingItem extends Item {
  constructor(title: string) {
    super(title);
  }

  listingSpecificElements = {
    picture: () => this.elements.container().find('.inventory_item_img'),
  };
}

export default ListingItem;
```

 `div.inventory_item` 679 x 240




### Sauce Labs Backpack

carry.allTheThings() with the sleek, streamlined Sly Pack that melds uncompromising style with unequaled laptop and tablet protection.

\$29.99

Remove



### Sauce Labs Bike Light

A red light isn't the desired state in testing but it sure helps when riding your bike at night. Water-resistant with 3 lighting modes, 1 AAA battery included.

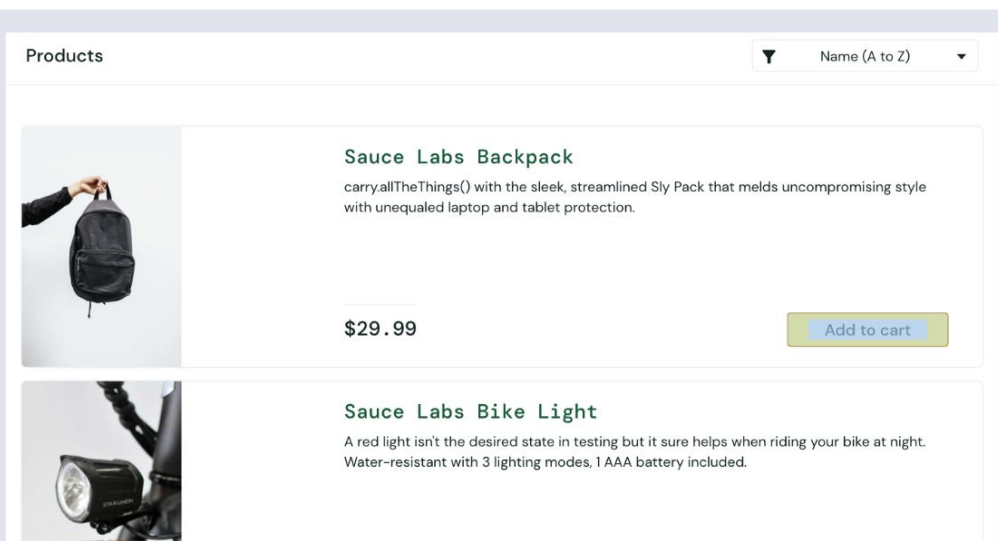
\$9.99

Add to cart



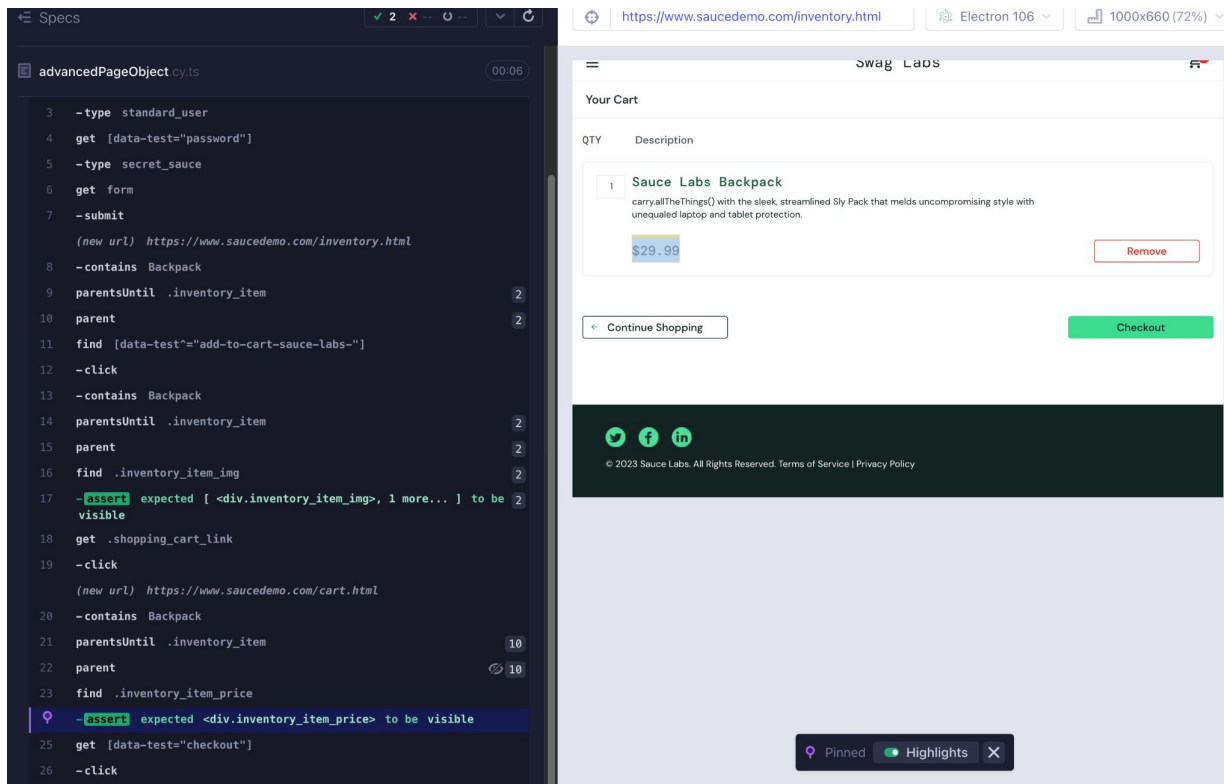
## Podejście 3.

```
advancedPageObject.cy.ts 00:06
3   -type standard_user
4   get [data-test="password"]
5   -type secret_sauce
6   get form
7   -submit
   (new url) https://www.saucedemo.com/inventory.html
8   -contains Backpack
9   parentsUntil .inventory_item 2
10  parent 2
11  ♀ find [data-test^="add-to-cart-sauce-labs-"]
12  -click
13  -contains Backpack
```





# Podejście 3.



The image shows a Cypress test runner interface on the left and a browser window on the right. The browser window displays the 'Your Cart' page of the Sauce Labs demo application, showing a single item 'Sauce Labs Backpack' with a price of \$29.99 and a 'Remove' button. The Cypress test runner shows a test suite 'advancedPageObject.cy.ts' with a series of commands for navigating to the inventory page, adding an item to the cart, and checking out. The test runner interface includes a command log on the right side of the code editor, showing the execution of each command and its duration. The browser window has a dark theme and a 'Swag Labs' header. The footer of the browser window contains social media links and a copyright notice.

```
Specs
advancedPageObject.cy.ts 00:06

3 -type standard_user
4 get [data-test="password"]
5 -type secret_sauce
6 get form
7 -submit
  (new url) https://www.saucedemo.com/inventory.html
8 -contains Backpack
9 parentsUntil .inventory_item 2
10 parent 2
11 find [data-test="add-to-cart-sauce-labs-"]
12 -click
13 -contains Backpack
14 parentsUntil .inventory_item 2
15 parent 2
16 find .inventory_item_img 2
17 -assert expected [ <div.inventory_item_img>, 1 more... ] to be visible 2
18 get .shopping_cart_link
19 -click
  (new url) https://www.saucedemo.com/cart.html
20 -contains Backpack
21 parentsUntil .inventory_item 10
22 parent 10
23 find .inventory_item_price
24 -assert expected <div.inventory_item_price> to be visible
25 get [data-test="checkout"]
26 -click
```

Browser window: https://www.saucedemo.com/inventory.html

Swag Labs

Your Cart

QTY	Description
1	<b>Sauce Labs Backpack</b> carry.allTheThings() with the sleek, streamlined Sly Pack that melds uncompromising style with unequalled laptop and tablet protection. \$29.99 <a href="#">Remove</a>

[Continue Shopping](#) [Checkout](#)

© 2023 Sauce Labs. All Rights Reserved. [Terms of Service](#) | [Privacy Policy](#)

Pinned Highlights



## Podjęcie 3.

```
describe('Advanced Page Object', () => {  
  let listingPage: ListingPage = new ListingPage();  
  let basketPage: BasketPage = new BasketPage();  
  let loginPage: LoginPage = new LoginPage();  
  
  it('should go to checkout with Backpack', () => {  
    let listingItem: ListingItem = new ListingItem('Backpack');  
    let basketItem: BasketItem = new BasketItem('Backpack');  
    loginPage.login();  
    listingItem.bar().addToBasket();  
    listingItem.listingSpecificElements.picture().should('be.visible');  
    listingPage.openBasket();  
    basketItem  
      .bar()  
      .elements.price()  
      .should('be.visible');  
    basketPage.clickCheckout();  
  });  
});
```



# Iframe

```
export const getIframeDocument = () => {  
  return cy  
    .get('iframe')  
    .its('0.contentDocument')  
    .should('exist');  
};  
  
export const getIframeBody = () => {  
  // get the document  
  return getIframeDocument()  
    .its('body')  
    .should('not.be.undefined')  
    .then(cy.wrap);  
};
```

```
getIframeBody().within(() => {  
  // your logic  
});
```



## Nowe zakładki

```
cy.get('.example > a')  
  .invoke('removeAttr', 'target')  
  .click();
```



## Nowe okna

```
it('Handling New Window — using stub', function() {
  cy.visit('https://alapanme.github.io/testing-cypress.html');
  const newUrl = 'https://the-internet.herokuapp.com/';

  cy.window().then((win) => {
    cy.stub(win, 'open').as('windowOpen');
  });

  cy.get('button').click();
  cy.get('@windowOpen').then((out) => {
    cy.wrap(out.args[0][0]).as('newRealUrl');
  });

  cy.get('@windowOpen').should('be.calledWith', newUrl);
  cy.window().then((win) => {
    cy.get('@newRealUrl').then((nu) => {
      win.location.href = `${nu}`;
    });
  });
  cy.get('h2').should('contain', 'Available Examples');
});
```



# Testy snapshot

```
cy.get('...').toMatchImageSnapshot();  
cy.get('...').toMatchSnapshot();
```

```
cypress_block > TS cypress.config.ts > ...  
1  import { defineConfig } from 'cypress';  
2  const { initPlugin } = require('cypress-plugin-snapshots/plugin');  
3  
4  export default defineConfig({  
5    e2e: {  
6      chromeWebSecurity: false,  
7      setupNodeEvents(on, config) {  
8        initPlugin(on, config);  
9        // implement node event listeners here  
10     },  
11   },  
12 });  
13  
14
```

▼ Snapshots

✗ Image & Data snapshots

▼ TEST BODY

```
1  visit https://example.cypress.io/  
2  get .banner > .container  
3  - snapshot ✓ SNAPSHOTS MATCH  
4  get .banner > .container  
5  - snapshot <⚡ COMPARE SNAPSHOT
```

! Error

Snapshots do not match:

--- Snapshots > Image & Data snapshots #0

+++ Snapshots > Image & Data snapshots #0