

Aplikacja webowa kursy walut NBP

Autor: Patryk Figas

Aplikacja umożliwia użytkownikowi pobranie kursów walut z API Narodowego Banku Polskiego (NBP). Użytkownik wybiera walutę oraz zakres dat, a aplikacja wyświetla odpowiednie dane w postaci tabeli.

Technologie:

- **Framework:** Angular (standalone components)
- **Język:** TypeScript
- **HTTP klient:** Angular HttpClient
- **Testy:** Jasmine + Karma

Kluczowe komponenty i funkcjonalności:

CurrencyComponent

- Wyświetla formularz z wyborem waluty oraz daty początkowej i końcowej.
- Umożliwia pobranie kursów walut z API NBP dla wybranego zakresu.
- Prezentuje dane w formie tabeli.
- Obsługuje błędy (np. brak danych, błędne daty).

CurrencyService

- Metoda `getTodayRates()` – pobiera listę dostępnych walut i kursów z tabeli A.
- Metoda `getRatesForPeriod(code, start, end)` – pobiera dane dla wybranej waluty i zakresu dat.

Testy jednostkowe:

Testowane scenariusze:

1. **Tworzenie komponentu** – sprawdzana jest poprawność inicjalizacji CurrencyComponent.
2. **Wywołanie ngOnInit()** – sprawdza, czy dane z serwisu są poprawnie wczytywane.
3. **Pobieranie kursów walut (fetchRates)** – symuluje wybór daty i waluty, sprawdzając:
 - a. czy wywoływana jest metoda serwisu,
 - b. czy wynik jest przypisany do rates,
 - c. czy przypisana jest poprawna nazwa waluty (currencyName).

Technologie testowe:

- jasmine.createSpyObj – tworzenie atrap serwisu.
- TestBed – konfiguracja środowiska testowego Angular.
- expect() – asercje do sprawdzania działania.

Przykład użycia:

1. Użytkownik otwiera aplikację.
2. Wybiera np. walutę: EUR, zakres: 2024-05-01 – 2024-05-10.
3. Kliknięcie „Pobierz dane” powoduje zapytanie do API.
4. Dane są wyświetlane jako tabela z datą i kursem średnim (NBP).

Obsługa błędów:

- Jeśli API zwróci błąd (np. zły kod waluty, brak danych), wyświetlany jest komunikat o błędzie.
- Walidacja: nie można wysłać pustego formularza – aplikacja wymaga wszystkich pól.

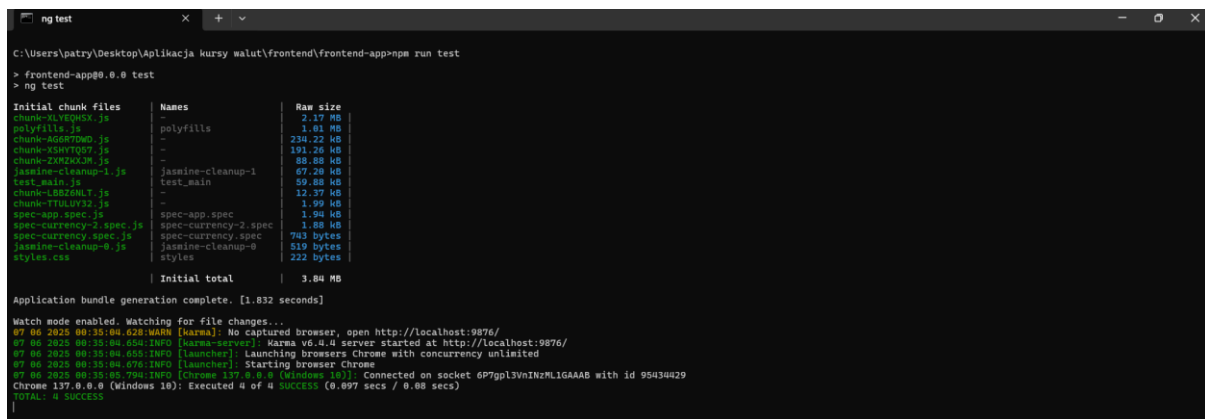
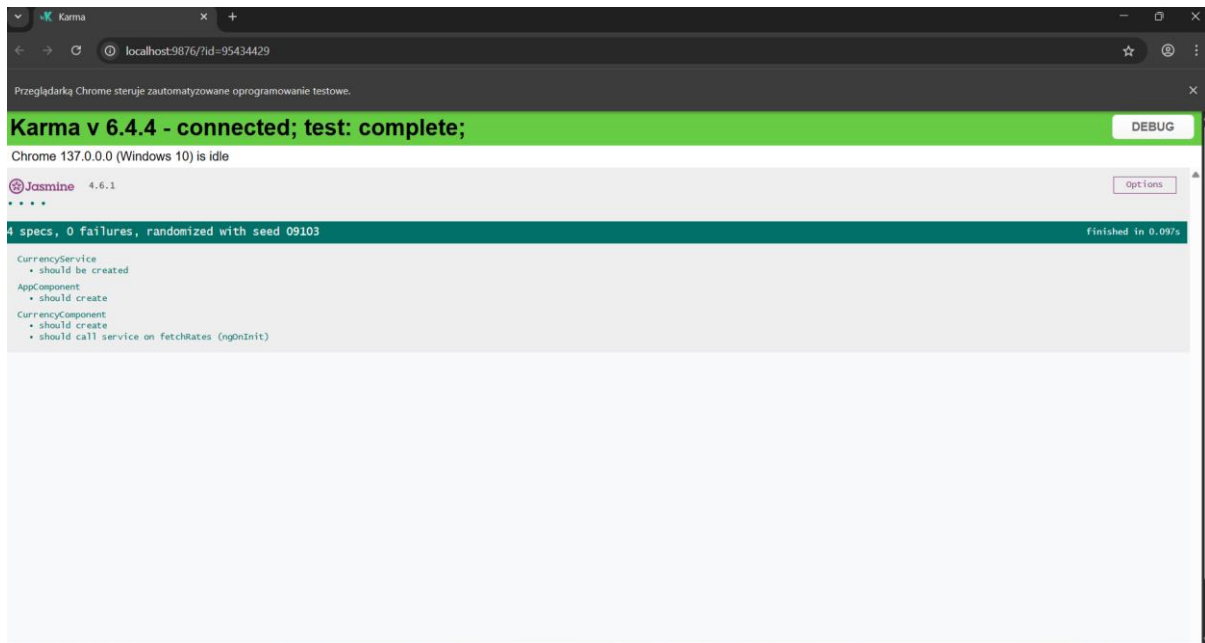
Podsumowanie:

Aplikacja jest prostym, lecz w pełni funkcjonalnym narzędziem do śledzenia kursów walut z oficjalnego źródła (NBP). Umożliwia podstawowe interakcje, posiada testy jednostkowe i opiera się na nowoczesnym podejściu standalone components w Angularze.

Testy frontend:

Test nr 1 sprawdza **inicjalizację komponentu**, czyli czy po jego uruchomieniu:

- wywołuje się serwis,
- dane są poprawnie zapisane,
- i czy ustawiono domyślną wartość waluty.



```
currency.spec.ts
Plik  Edytor  Wyświetl

import { ComponentFixture, TestBed } from '@angular/core/testing';
import { CurrencyComponent } from './currency.component';
import { CurrencyService } from '../services/currency.service';
import { of } from 'rxjs';
import { FormsModule } from '@angular/forms';

describe('CurrencyComponent', () => {
  let component: CurrencyComponent;
  let fixture: ComponentFixture<CurrencyComponent>;
  let mockService: jasmine.SpyObj<CurrencyService>;

  beforeEach(async () => {
    mockService = jasmine.createSpyObj('CurrencyService', ['getTodayRates', 'getRatesForPeriod']);

    // Ustawienie domyślnej wartości zwracanej przez getTodayRates
    mockService.getTodayRates.and.returnValue(of({
      rates: [{ currency: 'Euro', code: 'EUR', mid: 4.5 }],
      effectiveDate: '2025-06-06'
    }));

    await TestBed.configureTestingModule({
      imports: [FormsModule, CurrencyComponent],
      providers: [{ provide: CurrencyService, useValue: mockService }]
    }).compileComponents();

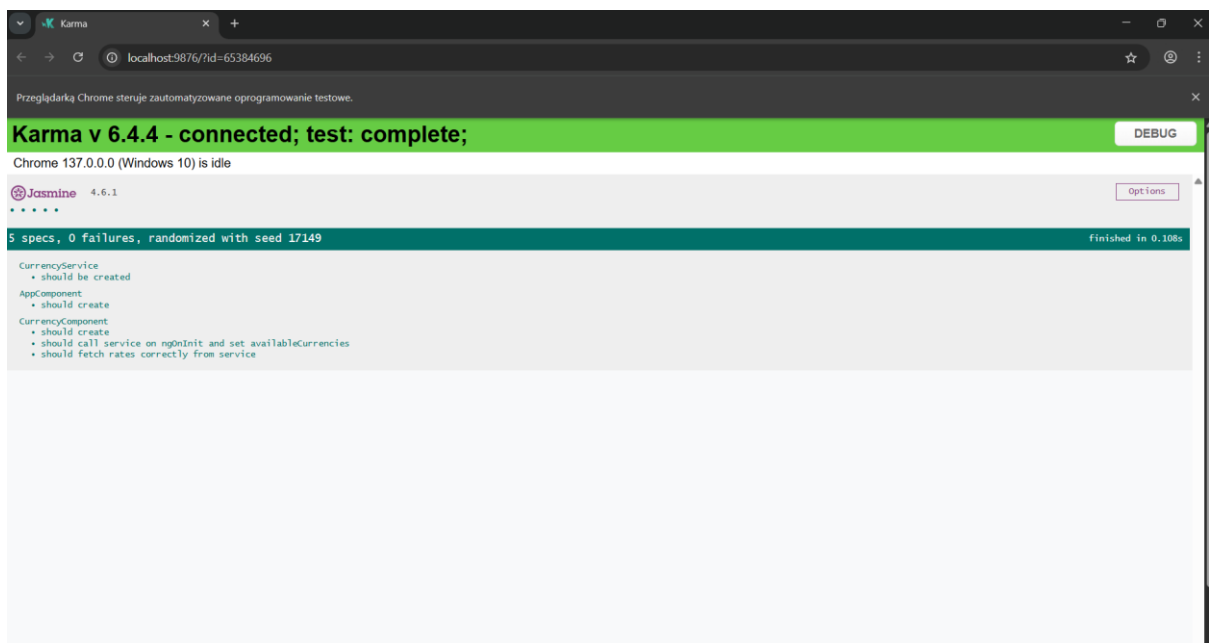
    fixture = TestBed.createComponent(CurrencyComponent);
    component = fixture.componentInstance;
    fixture.detectChanges(); // + tu już działa ngOnInit
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });

  it('should call service on fetchRates (ngOnInit)', () => {
    expect(mockService.getTodayRates).toHaveBeenCalled();
    expect(component.availableCurrencies.length).toBeGreaterThan(0);
    expect(component.currencyCode).toBe('EUR');
  });
});
```

Test nr 2:

- **symuluje działanie serwisu (mockService)** — nie odpala prawdziwych zapytań HTTP,
- **sprawdza logikę komponentu** — czy dane są pobierane i przypisywane prawidłowo,
- pomagają upewnić się, że komponent działa stabilnie i nie łamie się przy zmianach.



```
ng test

C:\Users\patry\Desktop\Aplikacja kursy walut\frontend\frontend-app>npm run test

> frontend-app@0.0.0 test
> ng test

Initial chunk files      Names      Raw size
chunk-XLYEQH5X.js      -          2.17 MB
polyfills.js           polyfills  1.01 MB
chunk-AG6R7Dw0.js      -          234.22 kB
chunk-XSHYTQ57.js      -          191.26 kB
chunk-ZMXKOJ3M.js      -          88.88 kB
jasmine-cleanup-1.js   jasmine-cleanup-1  57.20 kB
test_main.js           test_main  59.88 kB
chunk-LBBZ6MLT.js      -          12.37 kB
spec-currency-2.spec.js spec-currency-2.spec  2.81 kB
chunk-TTULUJ22.js      -          1.99 kB
spec-app.spec.js       spec-app.spec  1.94 kB
spec-currency.spec.js   spec-currency.spec  743 bytes
jasmine-cleanup-0.js   jasmine-cleanup-0  519 bytes
styles.css              styles      222 bytes

Initial total          3.85 MB

Application bundle generation complete. [1.860 seconds]

Watch mode enabled. Watching for file changes...
07 06 2025 08:47:23.689:WARN [karma]: No captured browser, open http://localhost:9876/
07 06 2025 08:47:23.723:INFO [karma-server]: Karma v6.4.4 server started at http://localhost:9876/
07 06 2025 08:47:23.724:INFO [launcher]: Launching browsers Chrome with concurrency unlimited
07 06 2025 08:47:23.737:INFO [launcher]: Starting browser Chrome
07 06 2025 08:47:25.054:INFO [Chrome 137.0.0.0 (Windows 10)]: Connected on socket s-egnTCTz5laAZDIAAAB with id 65384696
Chrome 137.0.0.0 (Windows 10): Executed 5 of 5 SUCCESS (0.108 secs / 0.091 secs)
TOTAL: 5 SUCCESS
```

```
currency.spec.ts

Plik  Edytuj  Wyświetl

import { ComponentFixture, TestBed } from '@angular/core/testing';
import { CurrencyComponent } from './currency.component';
import { CurrencyService } from '../services/currency.service';
import { of } from 'rxjs';
import { FormsModule } from '@angular/forms';
import { CommonModule } from '@angular/common';

describe('CurrencyComponent', () => {
  let component: CurrencyComponent;
  let fixture: ComponentFixture<CurrencyComponent>;
  let mockService: jasmine.SpyObj<CurrencyService>;

  beforeEach(() => {
    mockService = jasmine.createSpyObj('CurrencyService', ['getTodayRates', 'getRatesForPeriod']);

    await TestBed.configureTestingModule({
      imports: [FormsModule, CommonModule, CurrencyComponent],
      providers: [{ provide: CurrencyService, useValue: mockService }]
    }).compileComponents();

    fixture = TestBed.createComponent(CurrencyComponent);
    component = fixture.componentInstance;
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });

  it('should call service on ngOnInit and set availableCurrencies', () => {
    const mockData = {
      rates: { currency: 'EUR', code: 'EUR', mid: 4.5 },
      effectiveDate: '2025-06-06'
    };

    mockService.getTodayRates.and.returnValue(of(mockData));
    component.ngOnInit();

    expect(mockService.getTodayRates).toHaveBeenCalled();
    expect(component.availableCurrencies.length).toBe(1);
    expect(component.currencyCode).toBe('EUR');
  });

  it('should fetch rates correctly from service', () => {
    const mockResponse = {
      rates: [
        { effectiveDate: '2025-06-06', mid: 4.5 },
        { effectiveDate: '2025-06-06', mid: 4.55 }
      ],
      currency: 'EUR'
    };

    component.currencyCode = 'EUR';
    component.startDate = '2025-06-01';
    component.endDate = '2025-06-10';

    mockService.getRatesForPeriod.and.returnValue(of(mockResponse));
    component.fetchRates();

    expect(mockService.getRatesForPeriod).toHaveBeenCalled();
    expect(component.rates.length).toBe(2);
    expect(component.rates[0].rate).toBe(4.5);
    expect(component.rates[1].rate).toBe(4.55);
    expect(component.currencyName).toBe('EUR');
  });
});
```

Okno z wyborem dat i waluty:

← → ↻ ⓘ localhost:4200

Kursy walut NBP (zakres dat)

Waluta

bat (Tajlandia) (THB) ▾

Data od

dd.mm.rrrr 📅

Data do

dd.mm.rrrr 📅

Pobierz

Okno z wynikiem:

← → ↻ ⓘ localhost:4200

Kursy walut NBP (zakres dat)

Waluta

bat (Tajlandia) (THB) ▾

Data od

04.05.2025 📅

Data do

06.06.2025 📅

Pobierz

bat (Tajlandia) (THB)

DATA	KURS
2025-05-05	0.1143 zł
2025-05-06	0.1157 zł
2025-05-07	0.1152 zł
2025-05-08	0.1152 zł
2025-05-09	0.1142 zł
2025-05-12	0.1139 zł
2025-05-13	0.1152 zł
2025-05-14	0.1136 zł
2025-05-15	0.1135 zł
2025-05-16	0.1138 zł
2025-05-19	0.1150 zł

Repozytorium GITHUB z kodem źródłowym:

<https://github.com/PatrykFigas/Aplikacja-kursy-walut.git>