

# **Dokumentacja aplikacji do wyświetlania kursów walut. PIPR**

**Autor: Artem Kukushkin, 317140**

## **1. Wprowadzenie**

Aplikacja do wyświetlania kursów walut to narzędzie napisane w języku Python, które umożliwia pobieranie i wyświetlanie kursów walutowych z dwóch źródeł: Narodowego Banku (NBP) oraz Europejskiego Banku Centralnego (EBC) za pośrednictwem API frankfurter.app i exchangerate.host. Program oferuje interfejs graficzny (GUI) oparty na bibliotece Tkinter, umożliwiając wybór waluty, źródła danych, trybu pobierania (aktualny lub archiwalny) oraz wizualizację kursów w formie tekstowej i graficznej (wykresu). Aplikacja zawiera również zestaw testów jednostkowych weryfikujących poprawność kluczowych funkcji.

## **2. Zasada działania programu**

Aplikacja działa w następujący sposób:

### **1. Wybór parametrów:**

Użytkownik wybiera walutę (np. USD, EUR) i źródło danych (NBP lub EBC).

Użytkownik może włączyć tryb danych archiwalnych, określając zakres dat, lub korzystać z danych aktualnych (domyślnie z poprzedniego dnia dla NBP).

### **2. Pobieranie danych:**

Dla źródła NBP:

W trybie aktualnym aplikacja pomija bieżący dzień i pobiera dane z poprzedniego dnia, z możliwością cofania się do 5 dni wstecz w przypadku braku danych (kod HTTP 404).

W trybie archiwalnym pobiera dane dla podanego zakresu dat.

Obsługuje tabele A (kurs średni) i C (kursy kupna/sprzedaży).

Dla źródła EBC:

Wykorzystuje API frankfurter.app jako główne, z zapasowym API exchangerate.host w przypadku błędu.

Pobiera kursy względem EUR dla trybu aktualnego lub archiwalnego.

### **3. Wyświetlanie wyników:**

Wyniki tekstowe (kursy średnie, kupna, sprzedaży) są wyświetlane w polu tekstowym.

W trybie archiwalnym generowany jest wykres kursów średnich (dla NBP) lub kursów względem EUR (dla EBC) za pomocą Matplotlib.

#### 4. Walidacja danych:

Aplikacja sprawdza poprawność wybranej waluty, źródła i formatu dat (RRRR-MM-DD).

W przypadku błędów wyświetla komunikaty w oknie dialogowym.

#### 5. Czyszczenie wyników:

Użytkownik może wyczyścić pole tekstowe i ukryć wykres za pomocą przycisku „Wyczyść wyniki”.

### 3. Komunikacja z API

Aplikacja komunikuje się z dwoma zewnętrznymi API:

#### 3.1. API NBP

**Adres bazowy:** <https://api.nbp.pl/api/exchangerates/rates/{table}/{currency}>

**Parametry:**

table: Tabela NBP („A” dla kursu średniego, „C” dla kursów kupna/sprzedaży).

currency: Kod waluty (np. „USD”).

Tryb aktualny: {currency}/{date}/?format=json (np. 2025-06-10).

Tryb archiwalny: {currency}/{start\_date}/{end\_date}/?format=json (np. 2025-06-01/2025-06-05).

#### 3.2. API EBC

**Główne API:** <https://api.frankfurter.app>

Tryb aktualny: /latest?to={currency}.

Tryb archiwalny: /{start\_date}..{end\_date}?to={currency}.

**Zapasowe API:** <https://api.exchangerate.host>

Tryb aktualny: /latest?base=EUR&symbols={currency}.

Tryb archiwalny:

/timeseries?start\_date={start\_date}&end\_date={end\_date}&base=EUR  
&symbols={currency}.

**Logika:**

Próbuje najpierw frankfurter.app; w przypadku błędu przechodzi do exchangerate.host.

Zwraca komunikat błędu, jeśli oba API zawiodą.

#### 4. Budowa GUI

Interfejs graficzny oparty jest na bibliotece Tkinter i składa się z następujących elementów:

1. **Ramka wejściowa** (input\_frame):

**Wybór waluty:** Lista rozwijana (Combobox) z walutami: USD, EUR, GBP, CHF, JPY, AUD, CAD.

**Wybór źródła:** Lista rozwijana z opcjami: NBP, EBC.

**Dane archiwalne:** Pole wyboru (Checkbox), które pokazuje/ukrywa pola dat.

**Pola dat** (date\_frame): Dwa pola tekstowe (Entry) dla dat początkowej i końcowej (domyślnie: dziś i tydzień wstecz).

2. **Ramka przycisków** (button\_frame):

Przycisk „Pobierz kursy”: Wywołuje metodę fetch\_rates.

Przycisk „Wyczyść wyniki”: Wywołuje metodę clear\_results.

3. **Pole wyników tekstowych** (result\_text):

Pole tekstowe (Text) wyświetlające kursy w formacie:

NBP: Kurs średni (tabela A), kursy kupna/sprzedaży (tabela C), z adnotacją o cofaniu daty w trybie aktualnym.

EBC: Kurs względem EUR.

4. **Wykres** (canvas\_widget):

Wykres generowany przez Matplotlib, osadzony za pomocą FigureCanvasTkAgg.

Wyświetlany tylko w trybie archiwalnym poniżej pola tekstowego.

Pokazuje kursy średnie (NBP) lub kursy względem EUR (EBC) z formatowaniem dat na osi X.

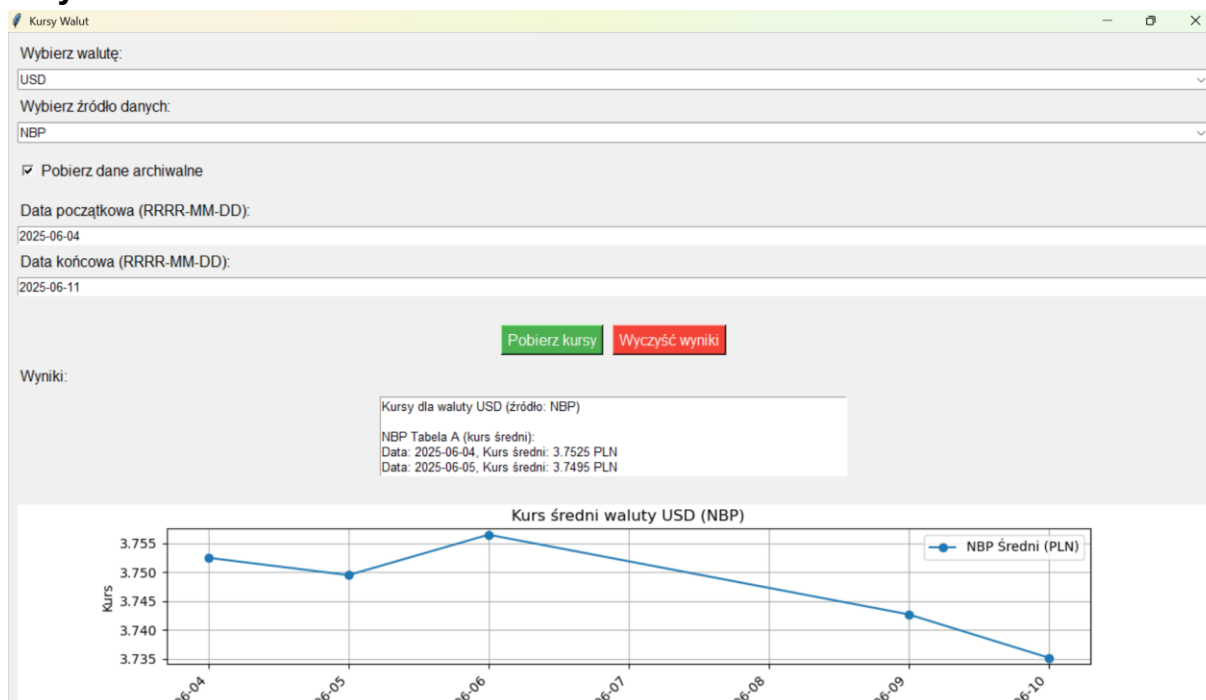
5. **Dynamika:**

Pola dat są ukrywane, gdy tryb archiwalny jest wyłączony (toggle\_date\_fields).

Wykres jest ukryty w trybie aktualnym lub po czyszczeniu wyników.

Walidacja danych wejściowych wyświetla komunikaty błędów w oknach dialogowych (messagebox.showerror).

## 6. Przykład GUI



## 5. Instrukcja obsługi

### 1. Wybór waluty i źródła:

Z listy rozwijanej „Wybierz walutę” wybierz walutę (np. USD).

Z listy „Wybierz źródło danych” wybierz NBP lub EBC.

### 2. Tryb danych:

Dla danych aktualnych: Pozostaw pole „Pobierz dane archiwalne” odznaczone. Program pobierze kursy z poprzedniego dnia dla NBP lub bieżące dla EBC.

Dla danych archiwalnych: Zaznacz „Pobierz dane archiwalne”, wprowadź daty w formacie RRRR-MM-DD (np. 2025-06-01, 2025-06-05).

### 3. Pobieranie kursów:

Kliknij „Pobierz kursy”. Wyniki pojawią się w polu tekstowym.

W trybie archiwalnym dodatkowo pojawi się wykres poniżej pola tekstowego.

### 4. Czyszczenie wyników:

Kliknij „Wyczyść wyniki”, aby usunąć tekst i ukryć wykres.

### 5. Obsługa błędów:

W przypadku błędnych danych (np. zły format daty, brak waluty) pojawi się okno dialogowe z komunikatem.

## 6. Testy jednostkowe

Testy jednostkowe znajdują się w pliku `test_project.py` i wykorzystują biblioteki `unittest` oraz `unittest.mock` do symulacji zapytań API i interakcji GUI. Testy pokrywają kluczowe funkcjonalności aplikacji.

### 6.1. Struktura testów

**Klasa testowa:** `TestCurrencyApp`.

**Metody przygotowania i sprzątania:**

`setUp`: Tworzy instancję Tkinter (root) i aplikacji (`CurrencyApp`), ustawia przykładowe dane (waluta, źródło, daty).

`tearDown`: Zamyka i niszczy okno Tkinter, z obsługą błędów związanych z GUI.

**Liczba testów:** 18, pokrywających metody `get_nbp_rates`, `get_ecb_rates`, `display_rates`, `plot_rates`, `toggle_date_fields`, `fetch_rates`.

### 6.2. Lista testów

#### 1. Testy dla `get_nbp_rates`:

- `test_get_nbp_rates_success_archival`: Pobieranie danych archiwalnych (sukces).
- `test_get_nbp_rates_current_skips_today`: Pomijanie bieżącego dnia w trybie aktualnym.
- `test_get_nbp_rates_current_retry_further`: Cofanie do wcześniejszego dnia przy błędzie 404.
- `test_get_nbp_rates_404_after_retries`: Brak danych po 5 próbach cofania.
- `test_get_nbp_rates_network_error`: Błąd sieciowy.

#### 2. Testy dla `get_ecb_rates`:

- `test_get_ecb_rates_success`: Pobieranie danych z `frankfurter.app` (sukces).
- `test_get_ecb_rates_fallback_success`: Pobieranie z `exchangerate.host` po błędzie `frankfurter.app`.
- `test_get_ecb_rates_both_fail`: Błąd obu API.

#### 3. Testy dla `display_rates`:

- `test_display_rates_nbp_success_current`: Wyświetlanie kursów NBP w trybie aktualnym.
- `test_display_rates_nbp_success_archival`: Wyświetlanie kursów NBP w trybie archiwalnym.
- `test_display_rates_ecb_success`: Wyświetlanie kursów EBC w trybie archiwalnym.

#### 4. **Testy dla** `plot_rates`:

- `test_plot_rates_nbp_success`: Rysowanie wykresu NBP (sukces).
- `test_plot_rates_nbp_error`: Rysowanie wykresu NBP przy błędzie.

#### 5. **Testy dla** `toggle_date_fields`:

- `test_toggle_date_fields_show`: Pokazywanie pól dat w trybie archiwalnym.
- `test_toggle_date_fields_hide`: Ukrywanie pól dat w trybie aktualnym.

#### 6. **Testy dla** `fetch_rates`:

- `test_fetch_rates_invalid_currency`: Walidacja nieprawidłowej waluty.
- `test_fetch_rates_invalid_date_format`: Walidacja nieprawidłowego formatu daty.
- `test_fetch_rates_success`: Poprawne pobieranie kursów w trybie archiwalnym.

### 7. **Możliwe rozszerzenia**

#### 1. **Pomijanie weekendów:**

Dodanie logiki w `get_nbp_rates`, aby pomijać soboty i niedziele podczas cofania daty (np. za pomocą `datetime.weekday()`).

#### 2. **Dodatkowe waluty:**

Rozszerzenie listy walut na podstawie dostępnych danych w API.

#### 3. **Eksport wyników:**

Możliwość zapisania wyników tekstowych lub wykresu do pliku (np. CSV, PNG).