### **Scenariusze testowe dla aktualizacji danych**

#### **Cel:**

Zweryfikować, czy dane kursów walut są automatycznie aktualizowane w określonych odstępach czasu oraz czy użytkownicy bota zawsze widzą aktualne informacje bez opóźnień.

### **1. Testy funkcjonalne aktualizacji danych**

#### **Scenariusz 1: Automatyczna aktualizacja danych co 30 minut**

* **Cel:** Sprawdzić, czy bot automatycznie aktualizuje dane kursów walut zgodnie z ustawionym interwałem (30 minut).
* **Kroki:**
  + Uruchom bota.
  + Sprawdź w logach (lub w konsoli), czy funkcja update\_all\_kantors() uruchamia się dokładnie co 30 minut.
  + Porównaj czas aktualizacji w logach z czasem wyświetlanym w polu "Ostatnia aktualizacja" przez bota.
* **Oczekiwany wynik:**

Funkcja update\_all\_kantors() jest uruchamiana co 30 minut bez błędów. Pole "Ostatnia aktualizacja" w danych kantoru jest aktualizowane zgodnie z harmonogramem.

#### **Scenariusz 2: Ręczne wywołanie aktualizacji**

* **Cel:** Zweryfikować, czy ręczne wywołanie aktualizacji (np. przez administratora) działa poprawnie i natychmiast nadpisuje dane w bazie.
* **Kroki:**
  + Wstrzymaj automatyczną aktualizację (np. zatrzymaj scheduler).
  + Ręcznie wywołaj funkcję update\_all\_kantors() w środowisku testowym.
  + Sprawdź w logach, czy dane zostały pobrane i zapisane w bazie.
  + Wyświetl dane kantoru przez bota i porównaj je z rzeczywistymi danymi ze strony kantoru.
* **Oczekiwany wynik:**

Ręczne wywołanie aktualizacji natychmiast pobiera najnowsze dane i wyświetla je poprawnie w aplikacji.

### **2. Testy zgodności danych z harmonogramem**

#### **Scenariusz 3: Aktualizacja w okresie braku zmian na stronach**

* **Cel:** Zweryfikować, czy bot poprawnie aktualizuje dane nawet wtedy, gdy kursy walut na stronach kantorów się nie zmieniają.
* **Kroki:**
  + Porównaj dane kursów walut przed i po automatycznej aktualizacji.
  + Sprawdź w bazie danych, czy czas "ostatniej aktualizacji" został zmieniony, mimo braku zmian w wartościach kursów.
* **Oczekiwany wynik:**

Dane kursów w bazie pozostają niezmienione, jeśli wartości na stronach są takie same, ale czas "ostatniej aktualizacji" jest zaktualizowany.

#### **Scenariusz 4: Aktualizacja danych w okresie dużego ruchu**

* **Cel:** Sprawdzić, czy bot aktualizuje dane bez opóźnień w okresie dużego obciążenia (np. gdy wielu użytkowników korzysta z bota jednocześnie).
* **Kroki:**
  + Uruchom wielu klientów bota (np. za pomocą symulatora użytkowników).
  + W momencie dużego ruchu sprawdź, czy dane kursów są aktualizowane zgodnie z harmonogramem.
* **Oczekiwany wynik:**

Aktualizacja danych odbywa się zgodnie z harmonogramem bez zauważalnych opóźnień i bez wpływu na działanie bota dla użytkowników.

### **3. Testy wydajnościowe aktualizacji danych**

#### **Scenariusz 5: Czas aktualizacji dla wielu kantorów**

* **Cel:** Sprawdzić, ile czasu zajmuje pełna aktualizacja danych kursów walut dla wszystkich kantorów w zmiennej urls.
* **Kroki:**
  + Uruchom funkcję update\_all\_kantors() i zmierz czas jej wykonania (np. przy użyciu modułu time lub logowania).
  + Sprawdź, czy czas wykonania mieści się w dopuszczalnych granicach (np. poniżej 30 sekund).
* **Oczekiwany wynik:**

Funkcja update\_all\_kantors() przetwarza dane wszystkich kantorów w rozsądnym czasie, nie przekraczając 30 sekund.

#### **Scenariusz 6: Obsługa dużej liczby kantorów**

* **Cel:** Sprawdzić, jak bot działa przy zwiększeniu liczby kantorów (np. dodaniu dodatkowych 50 URL-ów do zmiennej urls).
* **Kroki:**
  + Dodaj więcej kantorów do zmiennej urls (symulując rzeczywiste kantory).
  + Uruchom funkcję update\_all\_kantors() i zmierz czas wykonania.
  + Sprawdź, czy aktualizacja działa poprawnie dla wszystkich kantorów i nie powoduje przeciążenia.
* **Oczekiwany wynik:**

Bot skutecznie aktualizuje dane dla wszystkich kantorów, a czas aktualizacji proporcjonalnie rośnie bez błędów.

### **4. Testy graniczne aktualizacji**

#### **Scenariusz 7: Aktualizacja danych w nieregularnych odstępach czasu**

* **Cel:** Sprawdzić, czy bot obsługuje scenariusze, w których harmonogram aktualizacji jest zmieniany (np. aktualizacja co 10 minut zamiast 30).
* **Kroki:**
  + Zmień interwał w schedulerze z 30 minut na 10 minut.
  + Uruchom funkcję update\_all\_kantors() z nowym harmonogramem.
  + Sprawdź, czy aktualizacje odbywają się zgodnie z nowym interwałem.
* **Oczekiwany wynik:**

Bot poprawnie aktualizuje dane zgodnie z nowym harmonogramem.

#### **Scenariusz 8: Aktualizacja przy braku połączenia z internetem**

* **Cel:** Zweryfikować, jak bot reaguje na brak dostępu do internetu podczas próby aktualizacji danych.
* **Kroki:**
  + Tymczasowo odłącz internet na serwerze bota.
  + Uruchom funkcję update\_all\_kantors().
  + Sprawdź logi bota oraz odpowiedź w aplikacji.
* **Oczekiwany wynik:**

W logach pojawia się komunikat ostrzegawczy o braku połączenia, a bot nie ulega awarii. Odpowiedź dla użytkownika: „Nie udało się zaktualizować danych. Spróbuj ponownie później.”

### **5. Testy negatywne aktualizacji danych**

#### **Scenariusz 9: Błędne dane na stronie kantoru podczas aktualizacji**

* **Cel:** Sprawdzić, jak bot radzi sobie z błędnymi danymi na stronie kantoru (np. brak tabeli z kursami).
* **Kroki:**
  + Zmień strukturę strony jednego z kantorów (np. usuń tabelę lub wstaw błędne wartości).
  + Uruchom funkcję update\_all\_kantors().
* **Oczekiwany wynik:**

Bot pomija błędne dane, wyświetla ostrzeżenie w logach i kontynuuje aktualizację dla pozostałych kantorów.

#### **Scenariusz 10: Wywołanie aktualizacji dla pustej listy kantorów**

* **Cel:** Sprawdzić, jak bot reaguje na brak kantorów w zmiennej urls.
* **Kroki:**
  + Wyczyść listę urls (ustaw ją na pustą).
  + Uruchom funkcję update\_all\_kantors().
* **Oczekiwany wynik:**

Bot kończy działanie bez błędów, a w logach pojawia się komunikat: „Brak kantorów do aktualizacji.”

### **Podsumowanie:**

Powyższe scenariusze zapewnią kompleksowe przetestowanie mechanizmu aktualizacji danych w bocie, zarówno w standardowych warunkach, jak i w sytuacjach wyjątkowych.