

Prezentacja bieżących danych z systemów kolejkowych jednostek Urzędu m.st. Warszawy i Dzielnic

Dokumentacja projektu

Dawid Sygocki
nr indeksu: 304108
27 stycznia 2020

Implementacja funkcjonalności

Specyfikacja projektu zakładała implementację następującej funkcjonalności:

- wyświetlanie listy dostępnych urzędów,
- wyświetlanie listy spraw urzędowych dostępnych w danym urzędzie,
- wyświetlenie aktualnego stanu kolejki dla danych spraw urzędowych,
- wizualizacja stanu monitorowanych kolejek na wykresie (liczba osób od czasu),
- lokalne cache'owanie pobieranych danych.

Ze względu na domniemaną docelową grupę odbiorczą i dla wygody użytkownika zdecydowałem się na stworzenie aplikacji okienkowej z graficznym interfejsem użytkownika.

Zawartość okna aplikacji

Wizualnie aplikację można podzielić na cztery główne części:

- rozwijaną listę dostępnych urzędów,
- wykres liniowy z seriami prezentujący liczebność kolejek na przestrzeni ostatniej godziny,
- tabelę z listą spraw urzędowych dostępnych w wybranym urzędzie,
- pasek stanu aplikacji z konfiguracyjnymi polami wyboru.

Ze względu na tymczasowość danych udostępnianych przez API systemów kolejkowych jednostek Urzędu m.st. Warszawy i Dzielnic, aplikacja nie może na wykresie zaprezentować danych z okresu, gdy nie była uruchomiona.

Wykres prezentuje wyłącznie wszystkie sprawy z danego urzędu: nie ma możliwości ograniczenia ich liczby albo łączenia spraw z różnych urzędów.

Budowa kodu aplikacji

Program podzielony został na trzy moduły odpowiedzialne kolejno za komunikację z internetowym API, przechowywanie pobranych danych lokalnie oraz budowę i obsługę graficznego interfejsu użytkownika.

Komunikacja z API (api.py)

Moduł kierujący zapytania HTTP do systemów Urzędu m.st. Warszawy i Dzielnic. Jako że Urząd nie udostępnia API zwracającego listę dostępnych urzędów, moduł przeszukuje pod tym kątem dokument HTML (przeznaczony raczej dla użytkownika niż dla maszyny). Dane dotyczące spraw w poszczególnych urzędach dostępne są natomiast w formacie JSON. Moduł używa wbudowanego parsera tego formatu w celu przetworzenia danych.

Obsługa lokalnej bazy danych (database.py)

Ta część programu rozwija funkcjonalność poprzedniego modułu o zapisywanie danych w lokalnej bazie danych. Więcej informacji na ten temat znajduje się w niżej położonej sekcji „Dane przechowywanie lokalnie”. Użyty system zarządzania bazą danych to SQLite wybrany ze względu na prostotę użycia, powszechność i architekturę rozwiązania (inna niż klient-serwer).

Graficzny interfejs użytkownika (gui.py)

Moduł ten korzysta z rozbudowanej, wieloplatformowej biblioteki PyQt, konstruuje z jej użyciem graficzny interfejs programu oraz zarządzając działaniem aplikacji.

Dodatkowo obecny jest również moduł główny (main.py) uruchamiający aplikację oraz skompilowany moduł przechowujący klucz użytkownika API wykorzystywany przez aplikację do autoryzacji (apikey.pyc).

Dodatkowy, szczegółowy opis zawarty jest w komentarzach rozszaniach po kodzie aplikacji.

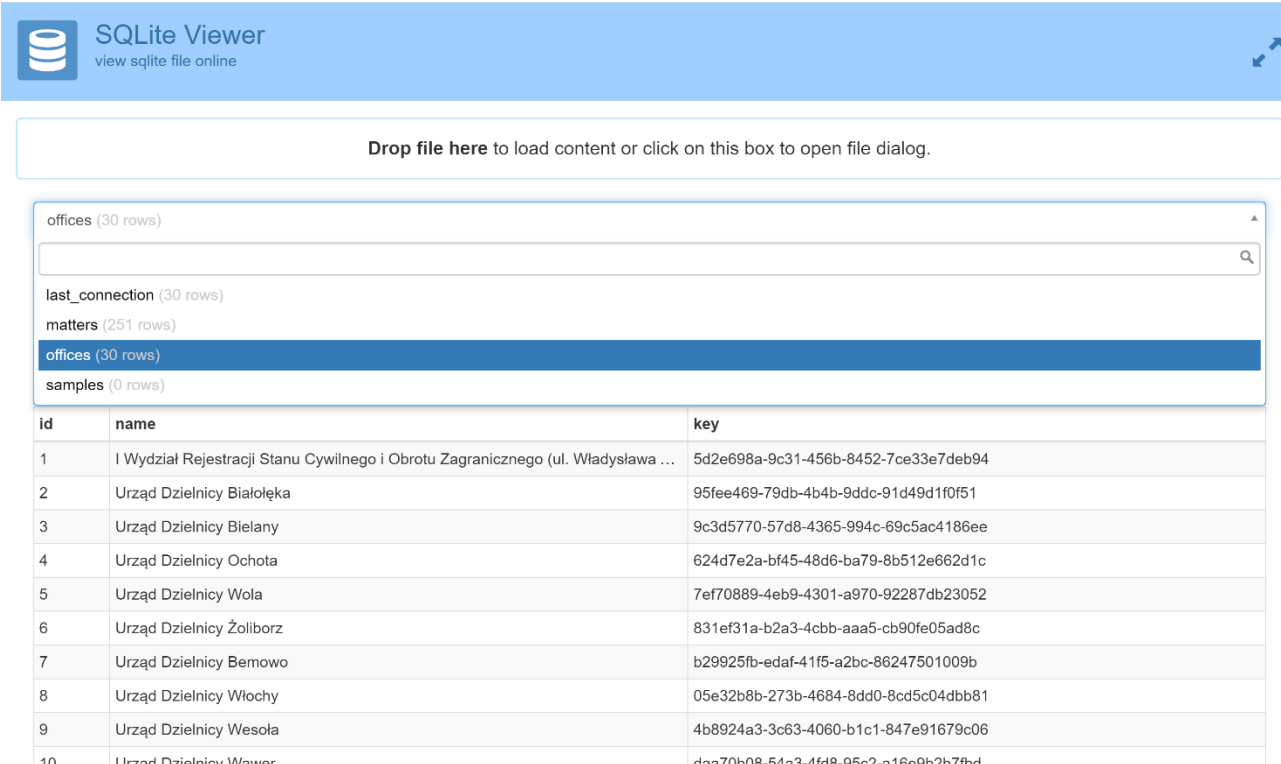
Dane przechowywane lokalnie

Aplikacja przechowuje lokalnie (w swoim folderze) dwa rodzaje danych: pobrane dane dotyczące systemów kolejkowych oraz ogólną konfigurację programu.

Baza danych systemów kolejkowych (cache.db)

Lokalna baza danych systemu SQLite przechowuje listę urzędów, dostępnych w nich spraw oraz próbek czasowych stanu kolejek. Dodatkowo baza zawiera informację o czasie ostatniego dostępu do API (w celu ograniczenia liczby zapytań).

Baza danych została wybrana do tego celu, ponieważ jej zastosowaniem jest z definicji przechowywanie i zarządzanie dużą ilością danych. Dodatkowo, system zarządzania danymi pozwala na wygodne i szybkie filtrowanie czy sortowanie rekordów.



The screenshot shows the SQLite Viewer application interface. At the top, there's a header with the SQLite logo and the text 'SQLite Viewer view sqlite file online'. Below this is a large box with the text 'Drop file here to load content or click on this box to open file dialog.' Underneath, there's a list of tables: 'offices (30 rows)', 'last_connection (30 rows)', 'matters (251 rows)', 'offices (30 rows)' (highlighted), and 'samples (0 rows)'. Below the table list, there's a preview of the 'offices' table with columns 'id', 'name', and 'key'. The table contains 10 rows of data.

id	name	key
1	I Wydział Rejestracji Stanu Cywilnego i Obrotu Zagranicznego (ul. Władysława ...	5d2e698a-9c31-456b-8452-7ce33e7deb94
2	Urząd Dzielnicy Białołęka	95fee469-79db-4b4b-9ddc-91d49d1f0f51
3	Urząd Dzielnicy Bielany	9c3d5770-57d8-4365-994c-69c5ac4186ee
4	Urząd Dzielnicy Ochota	624d7e2a-bf45-48d6-ba79-8b512e662d1c
5	Urząd Dzielnicy Wola	7ef70889-4eb9-4301-a970-92287db23052
6	Urząd Dzielnicy Żoliborz	831ef31a-b2a3-4cbb-aaa5-cb90fe05ad8c
7	Urząd Dzielnicy Bemowo	b29925fb-edaf-41f5-a2bc-86247501009b
8	Urząd Dzielnicy Włochy	05e32b8b-273b-4684-8dd0-8cd5c04dbb81
9	Urząd Dzielnicy Wesoła	4b8924a3-3c63-4060-b1c1-847e91679c06
10	Urząd Dzielnicy Wawer	daa70b08-54a3-4fd8-95c2-a16e9b2b7fbd

Rysunek 1. Podgląd przykładowych danych zapisanych w bazie

Konfiguracja aplikacji (settings.ini)

Plik konfiguracyjny zawiera proste ustawienia dotyczące wizualnej (pozycja i rozmiar okna, ostatnio przeglądany urząd) oraz funkcjonalnej części aplikacji (zakres przeprowadzanej okresowo aktualizacji danych: wyświetlany urząd lub wszystkie dostępne). Ustawienia te ładowane są przy uruchamianiu programu i zapisywane podczas jego pracy.

Wybrany został do tego celu format INI: równie prosty, co przechowywane ustawienia.

```
[window]
size=@Size(1068 820)
position=@Point(539 12)

[check_box]
restore_last_closed=false
update_only_current=true

[combo_box]
index=0
```

Rysunek 2. Przykładowa zawartość pliku INI

Podsumowanie

W wynikowej aplikacji zawarłem pełną zadaną funkcjonalność. Nie obyło się to oczywiście bez trudności. Przykładem może być zawieszanie się interfejsu graficznego przy aktualizacji danych, co spowodowane było blokadą głównej pętli programu przez synchroniczny dostęp do zasobów dyskowych i internetowych. Problem rozwiązałem, korzystając z dostępnego w bibliotece PyQt mechanizmu wielowątkowości.

Czasem planowane ulepszenia funkcjonalności okazywały się zbyt skomplikowane w implementacji, na przykład pierwotny projekt wizualny aplikacji zakładał pojawianie się okienka informacyjnego po najechaniu kursorem na punkt serii danych. Ze względu na zawartość lub brak dostępnego modułu wykresów biblioteki Qt, pomysł został porzucony. Najczęściej pojawiały się natomiast dylematy dotyczące implementacji: jaką bibliotekę lub algorytm wykorzystać, jak zaprojektować struktury danych.

Uważam jednak, że podjąłem właściwe decyzje, czego rezultatem jest spójna aplikacja przyjazna użytkownikowi i spełniająca swoje zadanie. Oprócz tego, sporządziłem i przeprowadziłem testy oraz udokumentowałem kod, co pomogło mi dopracować kod oraz uczyniło go zrozumialszym dla czytającego.

Instrukcja obsługi

Aplikacja służy do monitorowania danych systemów kolejkowych jednostek Urzędu m.st. Warszawy i Dzielnic.



Rysunek 3. Budowa okna aplikacji

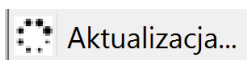
Aby rozpocząć zbieranie danych, wybierz urząd z listy rozwijanej. Spowoduje to pobranie bieżących danych na temat stanu systemu kolejkowego w tym urzędzie. W tabeli spraw zostaną wylistowane sprawy możliwe do załatwienia w tym urzędzie i dotyczące ich dane kolejkowe.

Wykres pozostanie pusty, dopóki program nie pobierze nowych danych (co robi co minutę). Wtedy pojawią się pierwsze linie wizualizujące zmieniającą się liczebność kolejek. Z biegiem czasu wykres będzie coraz pełniejszy. Aplikacja musi być włączona, by rejestrować dane na temat urzędu. Dane starsze niż godzina (a więc niemieszczące się na wykresie) będą automatycznie usuwane. Pobrane dane nie znikają jednak wraz z wyłączeniem programu – po włączeniu (w ciągu godziny) wykres dalej zawierał będzie zebrane informacje lub ich część.

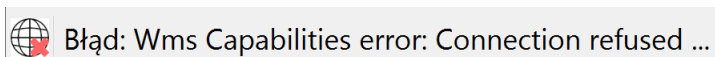
Po lewej stronie paska stanu prezentowane są dane na temat stanu aplikacji. Początkowo obszar jest pusty, ale po wybraniu urzędu pojawi się tam informacja graficzna oraz tekstowa. Program informuje użytkownika o stanie gotowości, przeprowadzanej aktualizacji danych oraz napotkanych błędach:



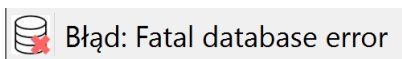
Rysunek 4. Pasek stanu: gotowość aplikacji



Rysunek 5. Pasek stanu: trwająca aktualizacja danych



Rysunek 6. Pasek stanu: przykładowy błąd sieciowy



Rysunek 7. Pasek stanu: przykładowy błąd przechowywania danych

Prawa część paska stanu zawiera dwa pola wyboru, pozwalające na konfigurację pracy programu. Zaznaczenie opcji „Przywracaj ostatnio zamknięty urząd” powoduje otwieranie aplikacji z wybranym ostatnio używanym urzędem. W takim przypadku przy uruchomieniu na liście rozwijanej nie jest wyświetlany tekst zachęty „Wybierz urząd...”, a aktualizacja danych przeprowadzana jest od razu (bez czekania na wybór urzędu przez użytkownika).

Opcja „Aktualizuj tylko bieżący urząd” zmienia zasady aktualizacji danych. Jeśli jest zaznaczona, pobierane są wyłącznie dane dotyczące obecnie wybranego urzędu. Po zmianie urzędu, wykres nie będzie zawierał danych z okresu, gdy wybrany był poprzedni urząd. Jeśli nie jest zaznaczona, aktualizowane są dane wszystkich urzędów jednocześnie. Pozwala to na równoczesne monitorowanie kilku urzędów.

Kliknięcie na sprawę z tabeli spraw zaznaczy ją i pogrubie oraz przeniesie na wierzch powiązaną z nią serię danych na wykresie. Zaznaczyć można na raz tylko jedną sprawę. By wyczyścić zaznaczenie, naciśnij i przytrzymaj klawisz Ctrl, jednocześnie klikając zaznaczoną sprawę.