**Struktura bazy danych**

Projekt aplikacji webowej

Łukasz Duda IO1

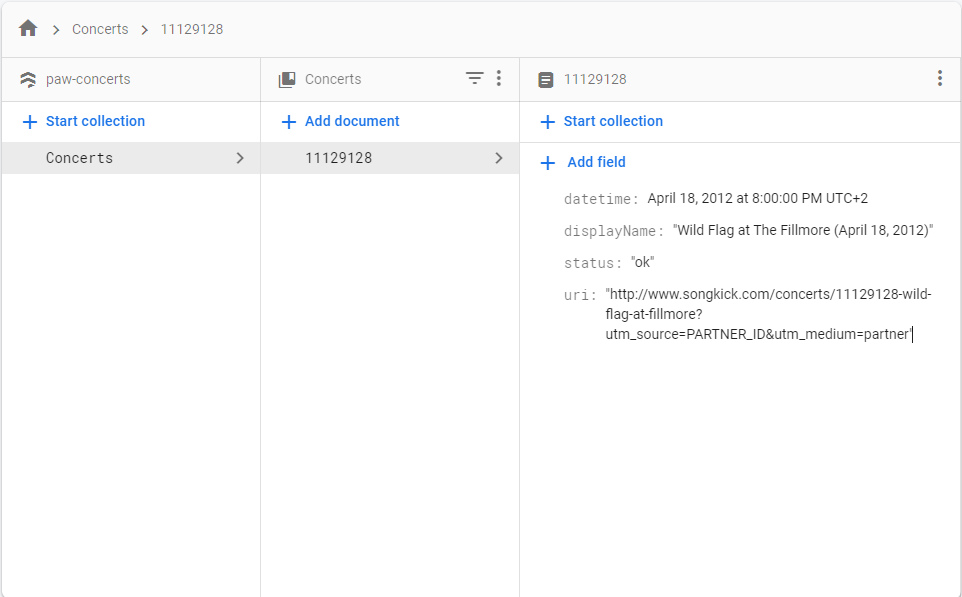
1. Struktura przykładowego pliku .json pobieranego z API. Jest to zbiór informacji na temat jednego wydarzenia.
2. {
3. "id":11129128,
4. "type":"Concert",
5. "uri":"http://www.songkick.com/concerts/11129128-wild-flag-at-fillmore?utm\_source=PARTNER\_ID&utm\_medium=partner",
6. "displayName":"Wild Flag at The Fillmore (April 18, 2012)",
7. "start": {
8. "time":"20:00:00",
9. "date":"2012-04-18",
10. "datetime":"2012-04-18T20:00:00-0800"
11. },
12. "performance": [
13. {
14. "artist":{
15. "uri":"http://www.songkick.com/artists/29835-wild-flag?utm\_source=PARTNER\_ID&utm\_medium=partner",
16. "displayName":"Wild Flag",
17. "id":29835,
18. "identifier":[]
19. },
20. "id":21579303,
21. "displayName":"Wild Flag",
22. "billingIndex":1,
23. "billing":"headline"
24. }
25. ],
26. "location": {
27. "city":"San Francisco, CA, US",
28. "lng":-122.4332937,
29. "lat":37.7842398
30. },
31. "venue": {
32. "id":6239,
33. "displayName":"The Fillmore",
34. "uri":"http://www.songkick.com/venues/6239-fillmore?utm\_source=PARTNER\_ID&utm\_medium=partner",
35. "lng":-122.4332937,
36. "lat":37.7842398,
37. "metroArea": {
38. "uri":"http://www.songkick.com/metro-areas/26330-us-sf-bay-area?utm\_source=PARTNER\_ID&utm\_medium=partner",
39. "displayName":"SF Bay Area",
40. "country": { "displayName":"US" },
41. "id":26330,
42. "state": { "displayName":"CA" }
43. }
44. },
45. "status":"ok",
46. "popularity":0.012763
47. }

Na podstawie przytoczonego kodu mogę zaplanować strukturę oraz hierarchię kolekcji dokumentów w mojej bazie danych.

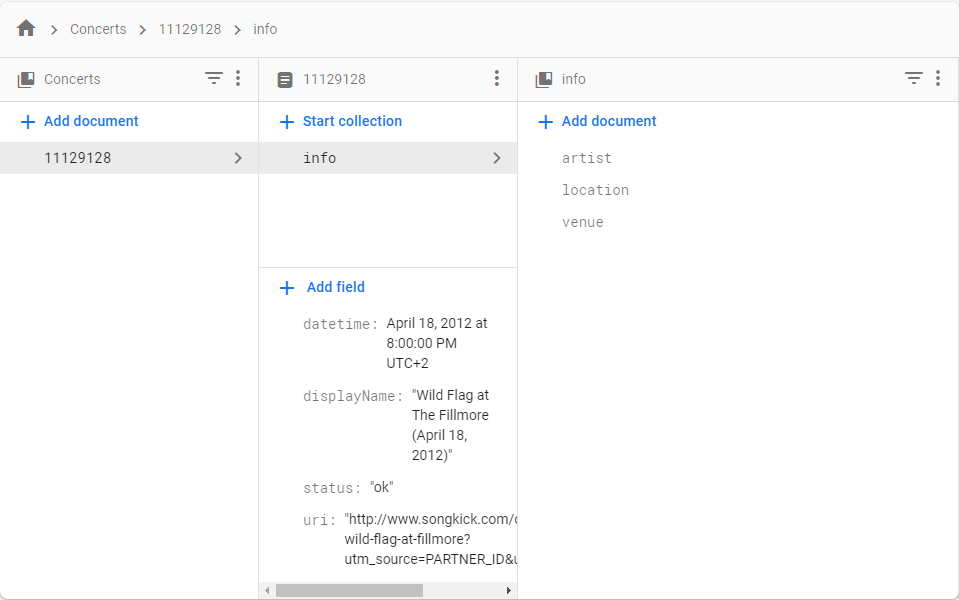
2. Struktura bazy danych.

Główna kolekcja nazywa się „Concerts”, w niej znajduje się zbiór dokumentów których nazwa odpowiada numerowi ID pobranemu z API.  
Podstawowe pola w tym dokumencie to:

* datetime – jest to data oraz czas rozpoczęcia wydarzenia,
* displayName – ogólna nazwa eventu,
* status – jest to informacja o stanie wydarzenia, „ok”(bez zmian), „canceled”(odwołany), „postponed”(przesunięty),
* uri – odnośnik do strony koncertu w aplikacji internetowej, która udostępnia API.

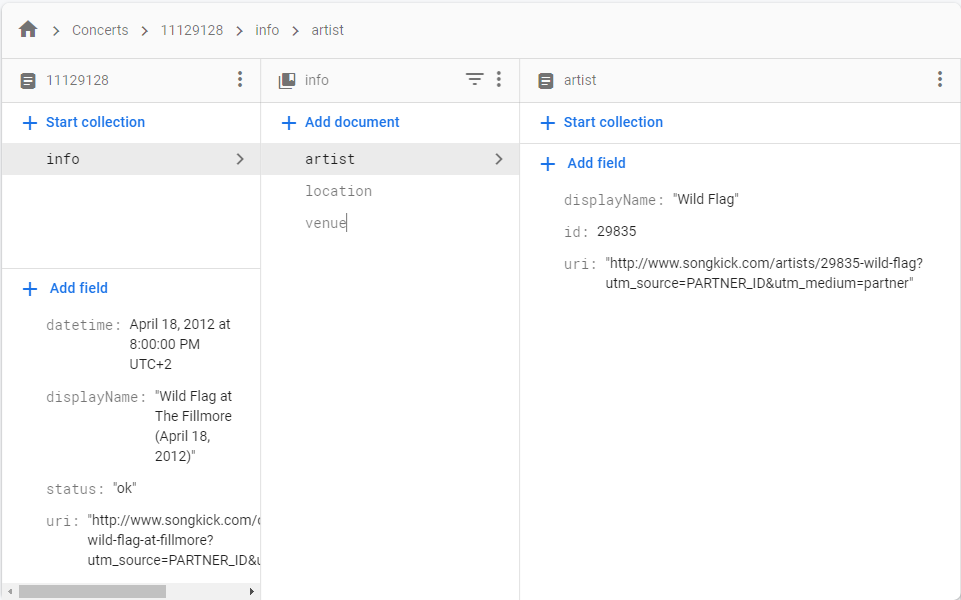


Dalej w dokumencie zagnieżdżona będzie kolekcja „info”, w której znajdować się będą dokumenty zawierające bardziej szczegółowe informacje na temat koncertu, tj. artysta, lokacja, lokal.



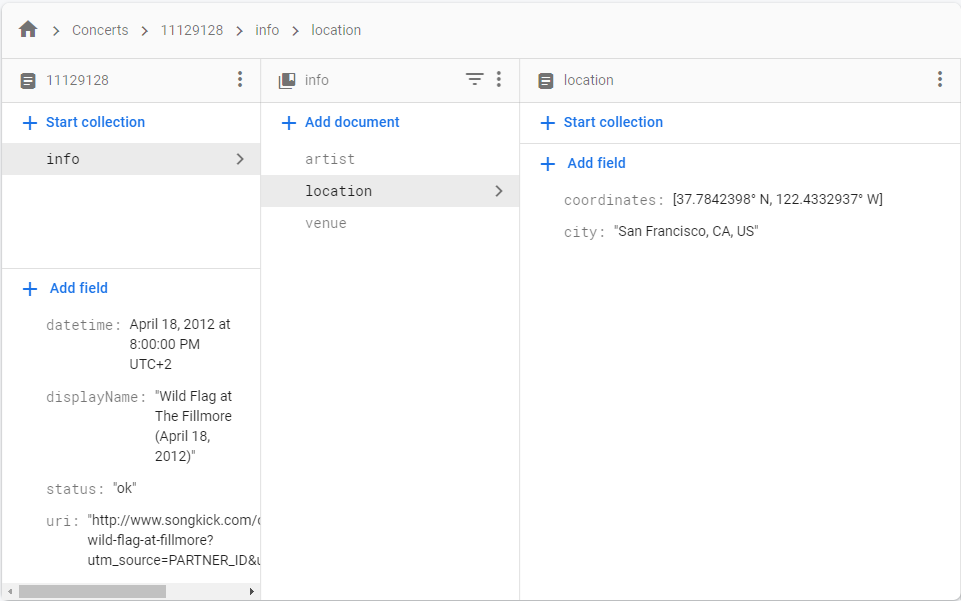
W dokumencie „artist” znajdować się będą podstawowe informacje na temat wykonawcy koncertu:

* displayName – nazwa lub imię i nazwisko artysty,
* id,
* uri – odnośnik do strony artysty w aplikacji internetowej, która udostępnia API.



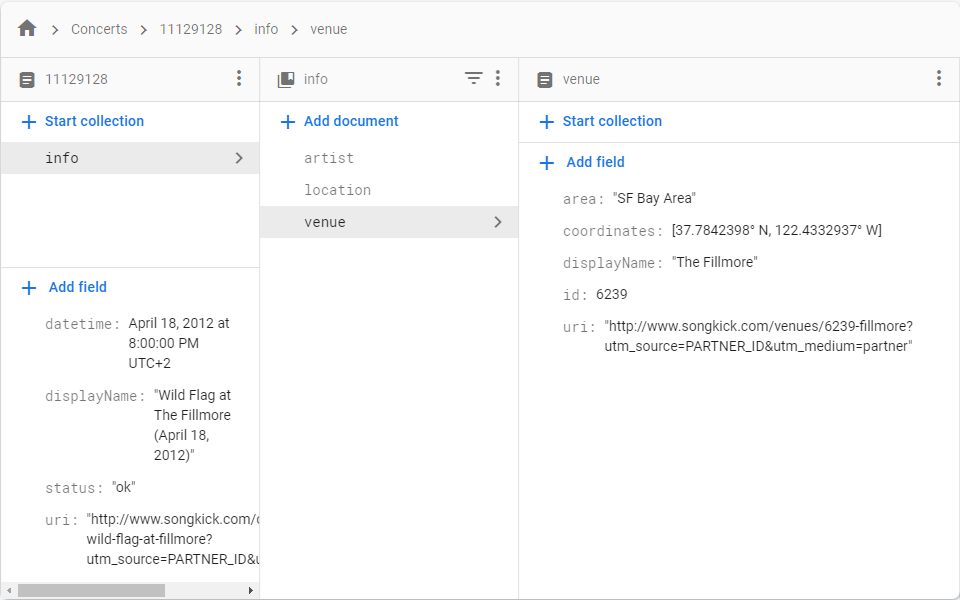
Dokument „location” zawiera krótką informację na temat miasta w którym odbywać będzie się koncert:

* coordinates – są to współrzędne miasta w którym odbędzie się wydarzenie,
* city – pełna nazwa miasta.



Ostatni dokument pt. „venue” to zbiór informacji o dokładnym miejscu koncertu – lokalu, hali, miejsca:

* area – nazwa okolicy w danym mieście(np. Warszawa – Mokotów),
* coordinates – dokładne współrzędne miejsca,
* displayName – nazwa miejsca,
* id,
* uri - odnośnik do strony „miejsca” w aplikacji internetowej, która udostępnia API.



Głównym założeniem aplikacji jest wyświetlanie użytkownikowi proponowanych lub docelowych koncertów, w zależności jakimi kategoriami wyszukuję pożądanych informacji. (artysta, data, miejsce). Niestety w API nie znajduje się informacja o kategorii muzyki danego artysty, jest to swego rodzaju ograniczenie, ponieważ nie da się zaproponować w ten sposób koncertów które również mogłyby się spodobać użytkownikowi. Jednak mając informację o dokładnych lokalizacjach (współrzędne), można spróbować proponować koncerty w okolicach zamieszkania użytkownika.