Dokumentacja
Drzewo Genealogiczne - Proof of concept

Mariusz Biegański

1 Wstęp

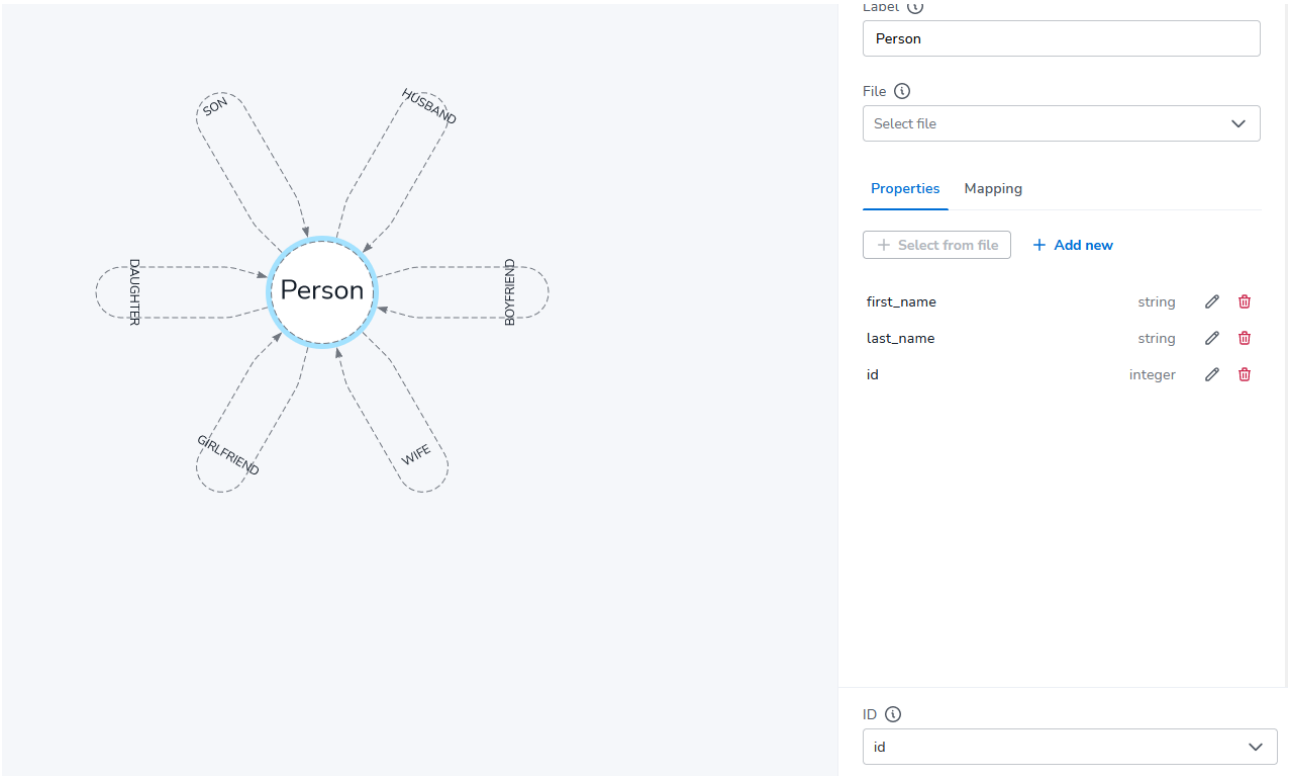
Celem projektu jest wykorzystanie bazy danych Neo4j do zweryfikowania możliwości jej zastosowania do rozwiązania określonego problemu lub realizacji konkretnego celu. Implementacja pozwala zauważyć zarówno korzyści, jak i ewentualne ograniczenia związane z wykorzystaniem tej technologii w danym projekcie. Projek Proof of Concept ogranicza koszty przed wyborem technologii do projektu biznesowego.

2 Funkcjonalność

Aplikacja umożliwia następujące funkcjonalności:

- Przeglądanie listy osób zapisanych w bazie danych Neo4j oraz ich relacji.
- Dodawanie nowych osób do bazy danych.
- Aktualizowanie danych osoby zapisanej w bazie danych.
- Usuwanie osób zapisanych w bazie danych.
- Tworzenie nowych relacji między osobami zapisanymi w bazie danych.
- Wyszukiwanie relacji między dwoma osobami zapisanymi w bazie danych.
- Wizualizację relacji pomiędzy osobami

3 Diagram UML



Rysunek 1: Diagram przedstawiający węzeł Person wraz z własnościami i relacjami

4 Technologie

Aplikacja wykorzystuje następujące technologie:

- **Neo4j**: grafowa baza danych służąca do przechowywania i przetwarzania danych w formie grafu.
- **AuraDB**: darmowa usługa hostingowa dla bazy danych Neo4j
- **Express.js**: framework sieciowy służący do obsługi żądań HTTP i tworzenia aplikacji sieciowych.
- **EJS**: silnik szablonów umożliwiający wstawienie dynamicznych danych do szablonów stron internetowych.
- **Vis.js**: biblioteka JavaScript do tworzenia interaktywnych wizualizacji danych i diagramów.
- **Render**: darmowa usługa hostingowa umożliwiająca hostowanie aplikacji Node.js. Aplikacja jest dostępna pod adresem https://family-tree-rq7c.onrender.com/.

5 Wdrożenie

Do połączenia z bazą danych Neo4j został użyty moduł neo4j-driver. Adres bazy danych oraz dane do logowania są odczytywane z pliku credentials.env za pomocą modułu dotenv.

Aplikacja obsługuje różne żądania HTTP, takie jak GET i POST, które służą do pobierania danych lub przesyłania danych do bazy. Na przykład, żądanie GET '/' renderuje stronę główną aplikacji, która wyświetla przyciski nawigacyjne, listę osób i ich relacji z pobranymi danymi z bazy danych oraz wizualizuje dane. Żądanie POST '/new-person' dodaje nową osobę do bazy danych za pomocą przesłanych danych z formularza.

Aby pobrać dane z bazy danych, aplikacja wywołuje asynchroniczne funkcje, takie jak `get_persons`, `get_relations` i `get_data`. Funkcje te wysyłają zapytania w języku Cypher do bazy danych, a następnie przetwarzają otrzymane dane, aby wyświetlić je na stronie.

Przy każdym żądaniu aplikacja pobiera nowe dane z bazy danych, aby zapewnić aktualność widoku. Dzięki temu użytkownik ma zawsze dostęp do najnowszych danych z bazy.

6 Wnioski

Baza danych Neo4j posiada następujące korzyści:

- Łatwe modelowanie i przechowywanie danych opartych na relacjach między elementami.
- Możliwość wykonywania złożonych zapytań z wykorzystaniem języka Cypher.
- Dobra integracja z innymi narzędziami oraz możliwość łatwego rozszerzenia o nowe funkcjonalności.
- Otwarta, rozwijająca się społeczność oraz szeroki wybór narzędzi i bibliotek wspierających pracę z Neo4j.
- Baza grafowa pozwala na lepsze zrozumienie powiązań między danymi, co może pomóc np. w ujawnieniu przestępstw czy innych nieprawidłowości