

AKADEMIA GÓRNICZO-HUTNICZA

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
KIERUNEK INFORMATYKA STOSOWANA



PRZETWARZANIE DANYCH W CHMURACH OBLICZENIOWYCH

Dokumentacja projektu

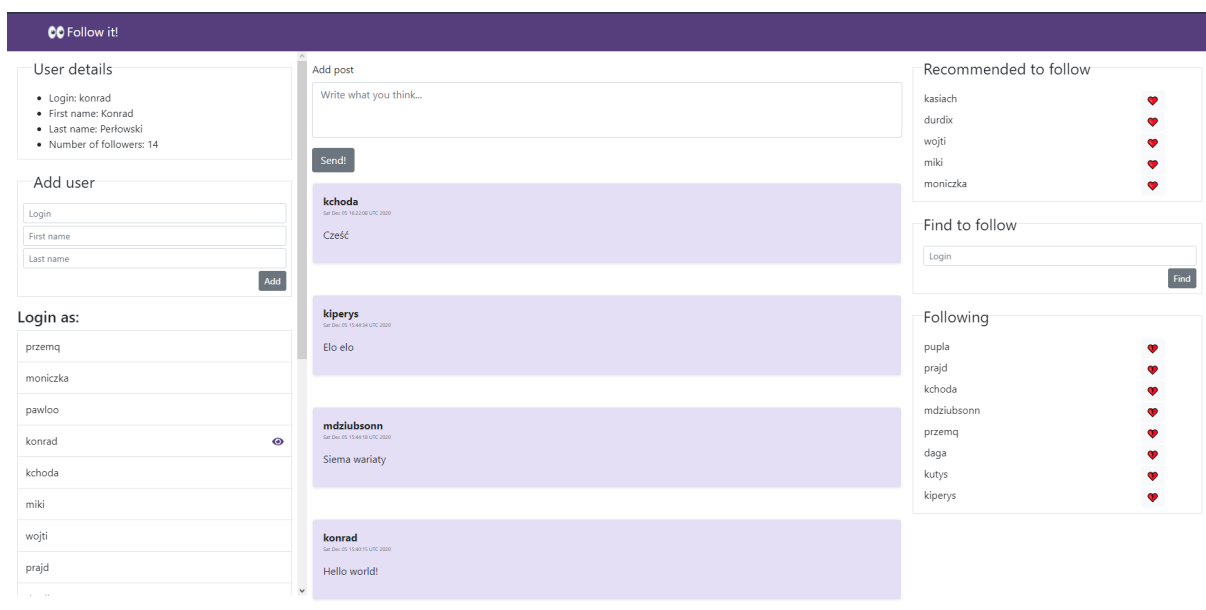
Konrad Perłowski

Kraków, 12 grudnia 2020

1 Wstęp

Dokumentacja dotyczy projektu *follow-it* stworzonego na przedmiot Przetwarzanie danych w chmurach obliczeniowych. Działający projekt znajduje się pod adresem: <https://follow-it.herokuapp.com>, kod źródłowy można znaleźć na repozytorium w serwisie github pod adresem: <https://github.com/Silnari/follow-it>. Celem projektu było stworzenie aplikacji internetowej korzystającej z grafowej bazy danych Neo4j.

Aplikacja *follow-it* pozwala na dodawanie nowych użytkowników, wyświetlanie widoku dla wybranego użytkownika, obserwowaniu innych użytkowników oraz dodawania własnych postów. Na stronie wyświetlane są posty obserwowanych przez aktualnie zalogowanego użytkownika obserwatorów, lista obserwowanych osób oraz rekomendacje użytkowników do obserwacji. Wygląd aplikacji został przedstawiony na rysunku 1.

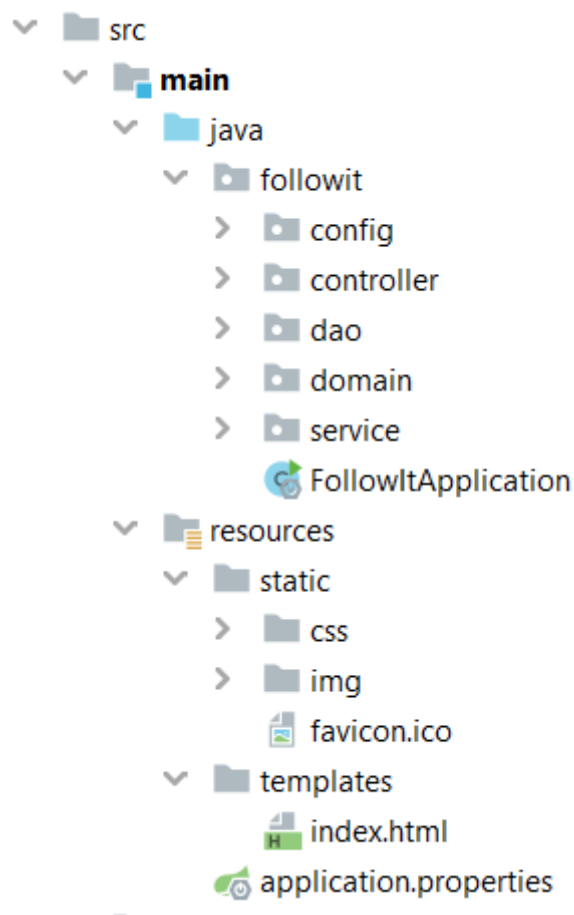


Rysunek 1: Strona główna aplikacji

2 Struktura aplikacji

Aplikacja została napisana w języku programowania Java z wykorzystaniem frameworku Spring w oparciu o model MVC (Model-View-Controller).

Struktura katalogów aplikacji została przedstawiona na rysunku 2.



Rysunek 2: Struktura aplikacji

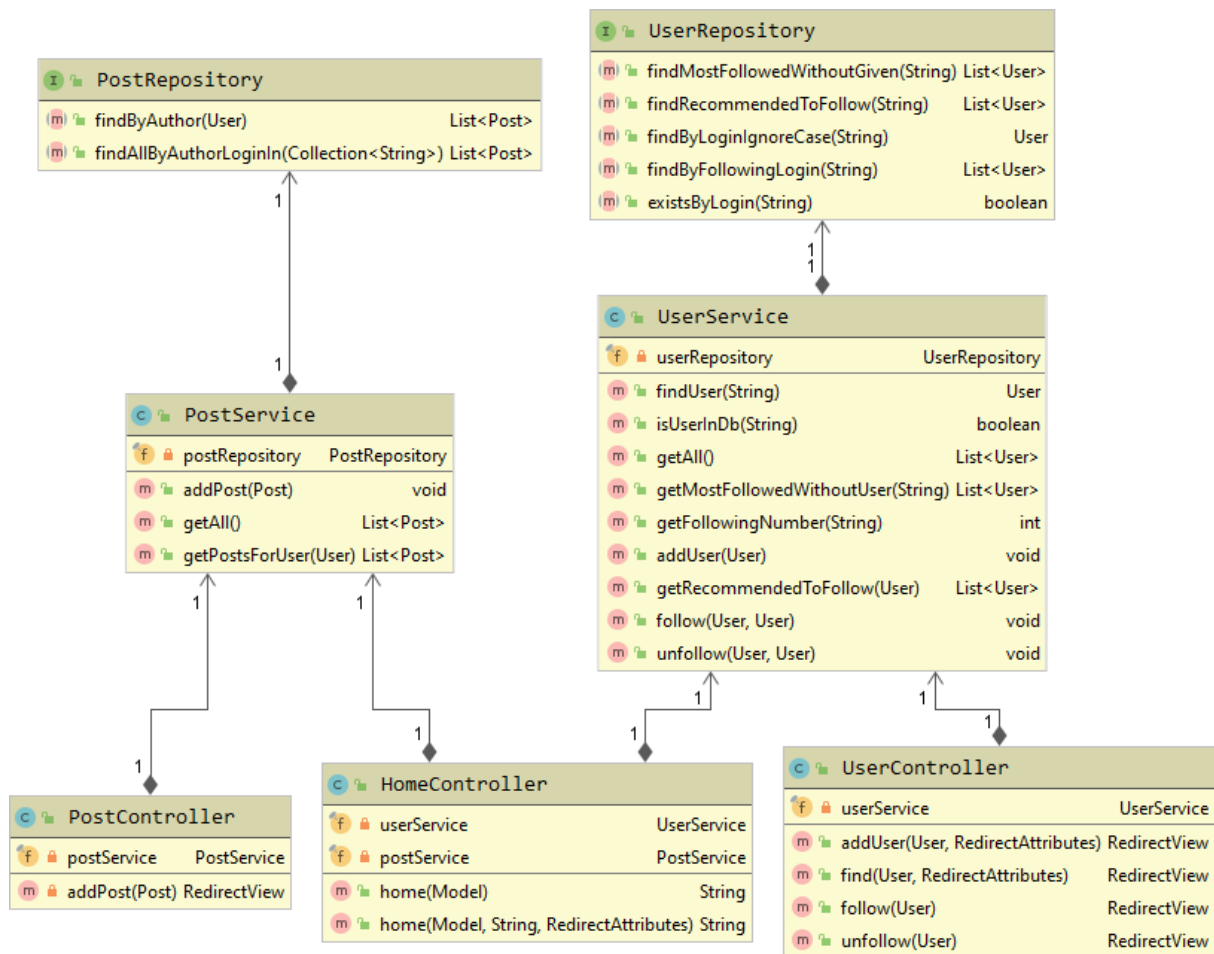
Na część front-endu aplikacji składa się plik *index.html* opisujący strukturę aplikacji, oraz pliki css umieszczone w folderze *css* opisujące wygląd poszczególnych elementów.

Część back-endu aplikacji została podzielona na następujące paczki:

- paczka *config* - zawiera klasy odpowiedzialne za niezbędną konfigurację, w tym przypadku zapewniają one konfigurację z bazą neo4j;
- paczka *controller* - zawiera klasy będące kontrolerami, udostępniają metody komunikacji warstwy widoku z warstwą modelu;
- paczka *dao* - zawiera repozytoria stanowiące warstwę dostępu do bazy danych;
- paczka *domain* - zawiera klasy definiujące węzły w bazie danych;
- paczka *service* - zawiera klasy udostępniające metody służące do pobierania danych z repozytoriów i zwracania ich w odpowiednim formacie.

Klasa *FollowitApplication* stanowi główną klasę aplikacji posiadającą metodę *main*.

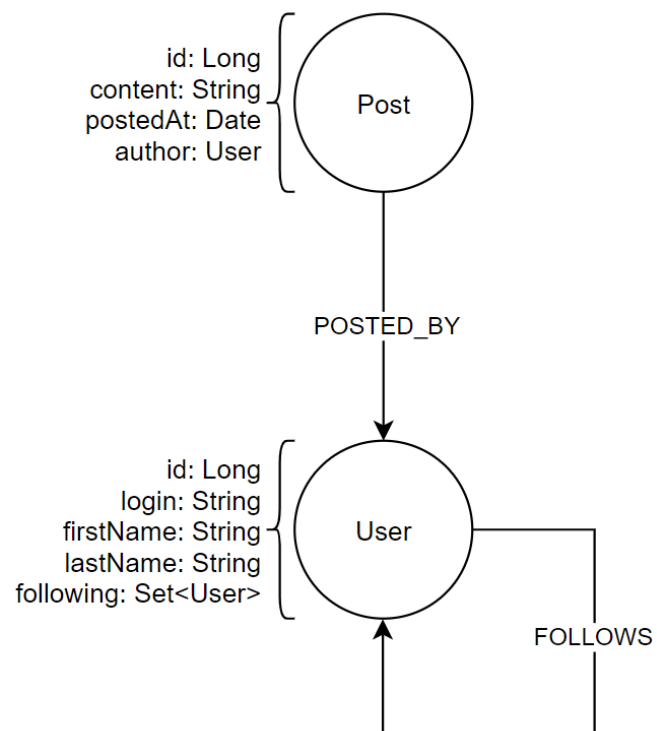
Struktura klas aplikacji została przedstawiona na rysunku 3.



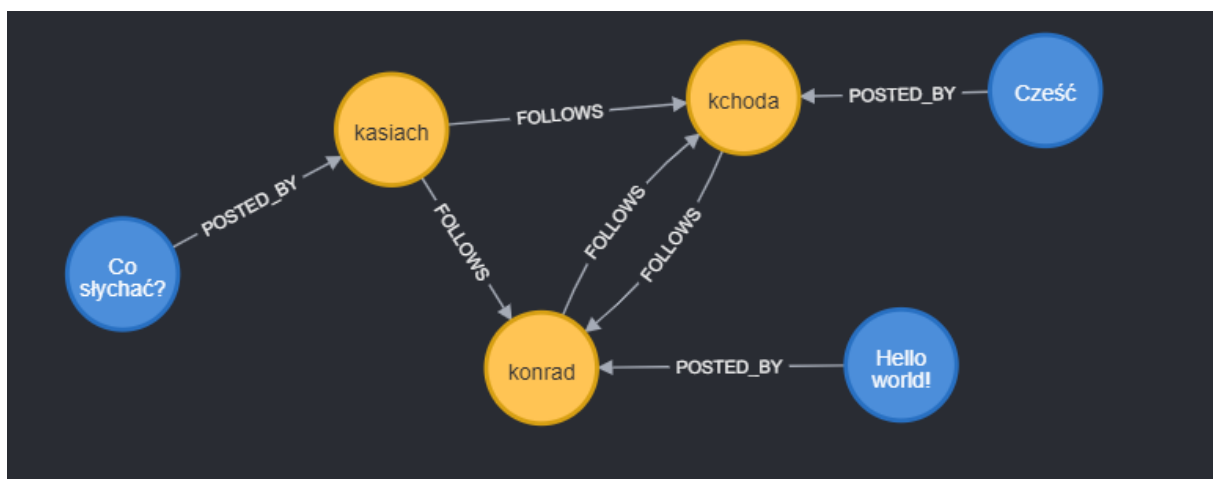
Rysunek 3: Diagram UML klas aplikacji

Diagram pokazujący strukturę bazy został przedstawiony na rysunku 4.

Przykładowe dane w bazie przedstawione zostały na rysunku 5.



Rysunek 4: Diagram bazy danych



Rysunek 5: Przykładowe dane w bazie

3 Wdrożenie aplikacji

Aplikacja została wdrożona w serwisie Heroku ze względu na prostotę procesu wdrażania. Wysłanie aplikacji wymaga zalogowania się poprzez terminal.

```
heroku login
```

Po poprawnym zalogowaniu niezbędna jest inicjalizacja nowego repozytorium git oraz zatwierdzenie wszystkich zmian.

```
git init
git add .
git commit -m "commit message"
```

Następnie tworzona jest aplikacja Heroku.

```
heroku create APP_NAME
```

Po stworzeniu aplikacji wraz z repozytorium można wysłać zatwierdzone zmiany do heroku co rozpocznie proces budowy projektu.

```
git push heroku master
```

Po poprawnym wysłaniu zmian otrzymamy link do naszej aplikacji.

Dodanie nowych zmian do aplikacji polega na otwarciu aplikacji

```
heroku open
```

oraz wysłaniu zmian wykorzystując system kontroli wersji GIT

```
git add .
git commit -m "commit message"
git push heroku master
```