

Projekt zaliczeniowy realizowany w ramach przedmiotu „Zaawansowane technologie internetowe”

Autor: Gabriel Naleźnik

Data: 29.06.2022 r.

1. Założenia projektu

Projekt nosi nazwę „Football Matches Collector”, przyjmując również polski tytuł: „Kolekcjoner meczy piłkarskich”. Jest to aplikacja internetowa umożliwiająca przechowywanie i przeglądanie zapisów z meczy piłkarskich. Ponadto, użytkownik ma możliwość utworzenia konta w serwisie. Aplikacja zrealizowana została zgodnie z wzorcem RESTful i podzielona na odrębne serwisy części serwerowej oraz części klienta.

Poza zapisywaniem danych ze spotkania, użytkownik może również dodać dane piłkarzy oraz klubów piłkarskich, a następnie wyszukiwać je z bazy po podanych słowach kluczowych.

2. Opis funkcjonalności serwerowej

Aplikacja serwerowa została zrealizowana przy pomocy technologii Java Enterprise Edition z wykorzystaniem frameworku Spring Boot.

Dostęp do danych został wykonany w oparciu o odwzorowanie obiektowo-relacyjne Spring Data. Na bazę danych została wybrana baza Postgres udostępniona w rozwiązaniu chmurowym DBaaS w serwisie <https://www.elephantsql.com/>

Serwer udostępnia następujące endpointy umożliwiające interakcję z zapisanymi danymi oraz dodanie nowych zapisów:

- GET /api/footballers
- GET /api/footballers/{id}
- POST /api/footballers
- PUT /api/footballers/{id}
- DELETE /api/footballers/{id}
- DELETE /api/footballers

- GET /api/clubs
- GET /api/clubs/{id}
- POST /api/clubs
- PUT /api/clubs/{id}
- DELETE /api/clubs/{id}
- DELETE /api/clubs

- GET /api/matches
- GET /api/matches/{id}
- POST /api/matches

- PUT /api/matches/{id}
- DELETE /api/matches/{id}
- DELETE /api/matches

- POST /api/auth/signin
- POST /api/auth/signup

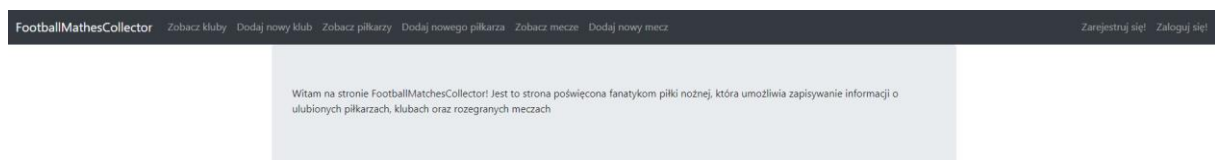
Serwis backendowy został wdrożony pod postacią rozwiązania chmurowego przy pomocy chmury Heroku.

3. Opis funkcjonalności klienta

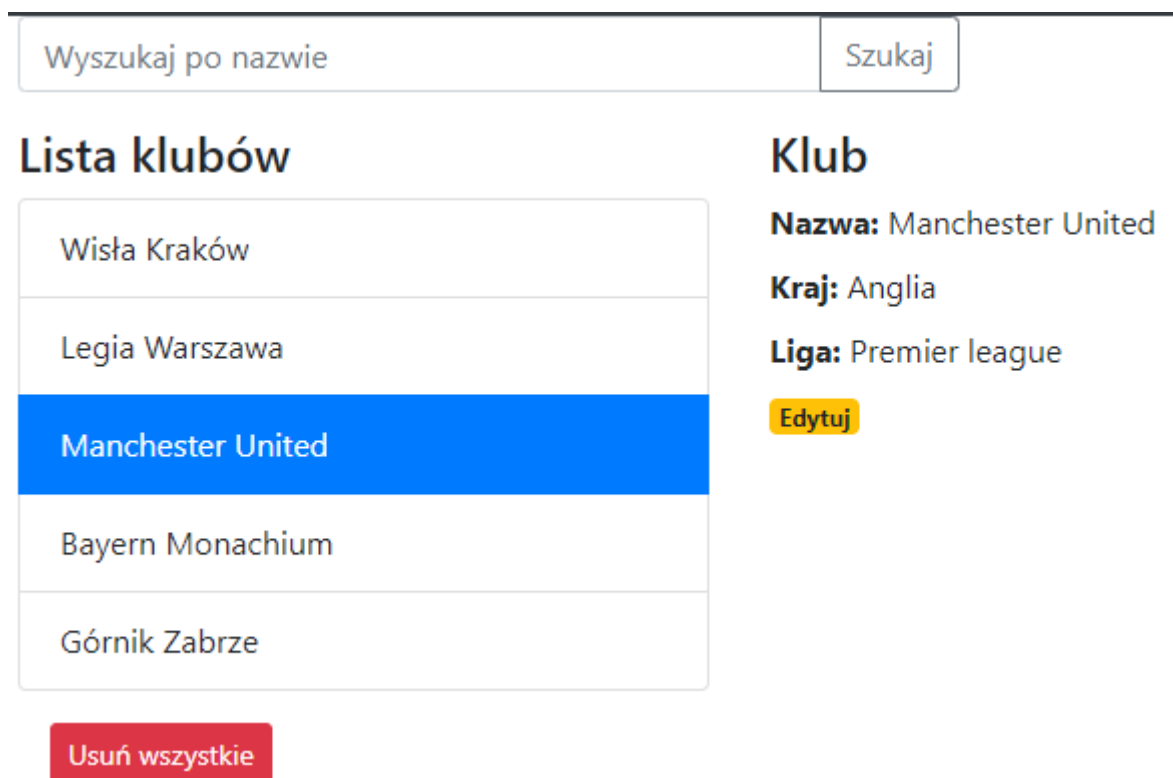
Aplikacja kliencka została opracowana z wykorzystaniem serwisu WWW i języka TypeScript przy pomocy frameworku Angular w wersji 13.

Serwis wykorzystuje protokoły http do komunikacji z częścią serwerową w celu pobrania zapisanych danych i dodania nowych w bazie postgres.

Do wprowadzania danych od użytkownika, wykorzystane zostały wbudowane moduły technologii Angular, udostępniające formularze „@angular/forms”.



Rys 1. Strona główna aplikacji



Rys 2. Strona wyszukiwania klubów piłkarskich

Nazwa klubu

Kraj

Liga

Dodaj

Rys 3. Formularz dodania nowego klubu

Szukaj po nazwisku

Szukaj

Szukaj po klubie

Szukaj

Lista piłkarzy

Jakub Błaszczykowski

Jerzy Brzeczek

Jakub Kosecki

Cristiano Ronaldo

Leo Messi

Robert Lewandowski

Kevin De Bruyne

Usuń wszystkich

Piłkarz

Imię: Cristiano

Nazwisko: Ronaldo

Wiek: 38

Klub: Manchester United

Edytuj

Rys 4. Strona wyszukiwania piłkarzy

Piłkarz

Imię

Nazwisko

Wiek

Klub

Rys 5. Formularz edycji danych piłkarza

Imię

Nazwisko

Wiek

Klub

Rys 6. Formularz dodania nowego piłkarza

Szukaj po klubie

Szukaj

Lista meczy piłkarskich

Wisła Kraków 2-2 Legia Warszawa
(05.10.2000)

Manchester United 2-1 Bayern Monachium
(26.05.1999)

Polska 2-0 Niemcy (11.10.2014)

Usuń wszystkie

Mecz piłkarski

Data: 11.10.2014

Wynik: 2-0

Gospodarz: Polska

Goście: Niemcy

Bramkarz gości: Manuel Neuer

Lewy obrońca gości: Erik Durm

Lewy środkowy obrońca gości: Jérôme Boateng

Prawy środkowy obrońca gości: Mats Hummels

Prawy obrońca gości: Antonio Rüdiger

Rys 7. Strona wyszukiwania meczy piłkarskich

Data

Wynik

Gospodarz

Goście

Bramkarz gości

Rys 8. Formularz dodania nowego meczu

Nazwa użytkownika

Email

Hasło

Zarejestruj się

Rys 9. Formularz rejestracji

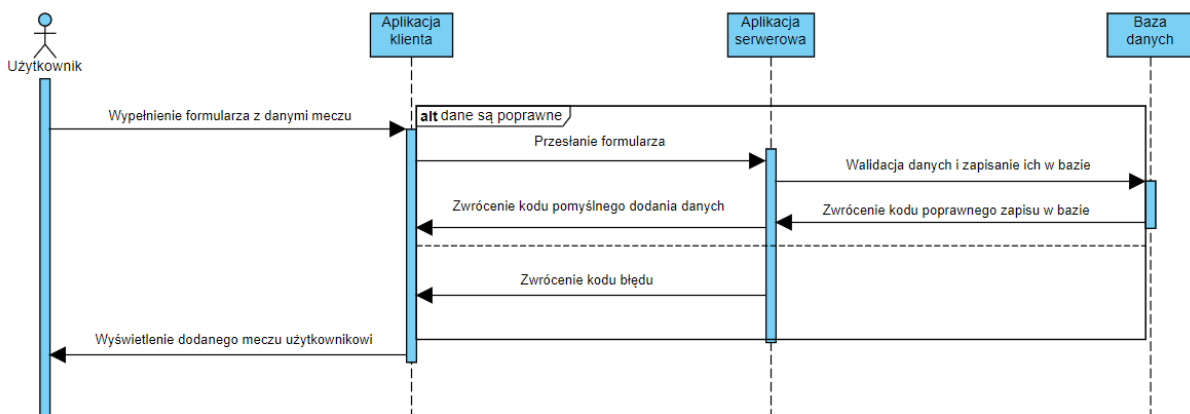
Nazwa użytkownika

Hasło

Zaloguj

Rys 10. Formularz logowania

4. Projekt diagramów UML



Rys 11. Diagram sekwencji dodania nowego meczu w serwisie

5. Instrukcja uruchomieniowa i wdrożeniowa

Aplikacja klienta i serwerowa została umieszczona w chmurze obliczeniowej Heroku.

Serwis klienta jest dostępny pod adresem: <https://football-matches-collector.herokuapp.com/>.

Adresem URL API części serwerowej jest: <https://infinite-island-13820.herokuapp.com/>. Przykładowe zapytanie wyświetlające zapisane mecze: <https://infinite-island-13820.herokuapp.com/api/matches>

Aby umieścić projekt na serwisie Heroku, należy posiadać aktywne konto Heroku oraz wykonać w głównym katalogu następujące polecenia:

- \$ heroku login
- \$ heroku create
- \$ git push heroku master

Aplikacja serwerowa może zostać również uruchomiona lokalnie w środowisku Idea IntelliJ przy pomocy serwera Apache Tomcat. W tym celu należy stworzyć spakowany folder zip w środowisku Idea IntelliJ i uruchomić lokalny serwer (domyślnie pod adresem localhost:8080).

Aby uruchomić lokalnie aplikację klienta, należy pobrać środowisko angular CLI za pomocą menedżera pakietów npm (\$ npm install -g @angular/cli) oraz serwer Node.js dostępny do pobrania na oficjalnej stronie: <https://nodejs.org/en/about/releases/>. Aplikacja zostaje uruchomiona poleceniem: **node server.js**

6. Dokumentacja kodu źródłowego

Część serwerowa została udokumentowana w kodzie źródłowym przy pomocy Javadoc.

```
/**
 * Klasa MatchController obsługująca endpointy dotyczące meczy piłkarskich
 */
@CrossOrigin(origins = "*", maxAge = 3600)
@RestController
@RequestMapping("/api")
public class MatchController {
    ...

    /**
     * Zwraca wszystkie mecze lub takie w których wystąpił podany klub
     * @param club
     * @return
     */
    @GetMapping("/matches")
    public ResponseEntity<List<Match>> getAllMatches(@RequestParam(required = false) String club) {
        ...
    }
}
```

Rys 12. Przykład udokumentowanego kodu części serwerowej

Pliki źródłowe zostały pogrupowane w następujące katalogi:

- Config – pliki konfiguracyjne
- Controller – Kontrolery opisujące endpointy REST
- Model – Struktury danych zapisywane w bazie danych
- Payload – Struktury przekazywane z aplikacji klienta
- Repository – Interfejs komunikacji z bazą danych
- Security – Pliki zabezpieczeń i autentykacji

Aplikacja klienta została podzielona na katalogi:

- _helpers – Funkcje pomocnicze do zarządzania nagłówkami autentykacji
- _services – Funkcje zawierające komunikację autentykacji z serwerem
- Board-admin – Komponent panelu admina
- Board-moderator – Komponent panelu moderatora
- Board-user – Komponent panelu użytkownika
- Components – Pozostałe komponenty aplikacji
- Home – Komponent strony domowej
- Login – Komponent strony logowania
- Models – Modele wysyłane do części serwerowej
- Profile – Komponent profilu użytkownika
- Register – Komponent rejestracji
- Services – Serwisy komunikacji z częścią serwerową

7. Podstawowy podręcznik użytkownika

Użytkownik może poruszać się po aplikacji za pomocą paska nawigacyjnego znajdującego się na górze strony. Z jego poziomu udostępnione zostały następujące opcje:

- FootballMatchesCollector – powrót na stronę główną
- Zobacz kluby – Wyświetlenie listy klubów zapisanych w bazie
- Dodaj nowy klub – Wyświetlenie formularza dodania nowego klubu
- Zobacz piłkarzy - Wyświetlenie listy piłkarzy zapisanych w bazie
- Dodaj nowego piłkarza – Wyświetlenie formularza dodania nowego piłkarza
- Zobacz mecze - Wyświetlenie listy meczy zapisanych w bazie
- Dodaj nowy mecz – Wyświetlenie formularza dodania nowego meczu
- Zarejestruj się – Wyświetlenie formularza rejestracyjnego
- Zaloguj się – Wyświetlenie formularza logowania do serwisu

Po zalogowaniu, wyświetlane są dodatkowo następujące opcje:

- Wyloguj się – wylogowanie aktualnie zalogowanego użytkownika
- Panel użytkownika – wyświetlenie panelu użytkownika
- <nazwa użytkownika> - wyświetlenie danych profilu użytkownika