

Łączą nas Scala — sprawozdanie

1.Opis projektu

Projekt składa się aplikacji internetowej umożliwiającej organizację turniejów koszykówki. API udostępnia narzędzia do tworzenia, edycji oraz usuwania zespołów, zawodników oraz samych turniejów i rozgrywanych na nich spotkań. Serwis ma również zaimplementowane uwierzytelnianie i autoryzację użytkowników .

Początkowo część frontendowa planowana była do wykonania za pomocą frameworka Spring, jednakże nie okazał się on szczególnie użyteczny w przypadku aplikacji internetowej. Użyliśmy zatem frameworka Scala Play, korzystającego z dobrodziejstw paradygmatu funkcyjnego. Baza danych została utworzona na klastrze dokumentowej bazy MongoDB. Aplikacja łączy się z bazą za pomocą serwera napisanego w środowisku Node.js za pomocą frameworka Express.

W ramach udogodnienia używania projektu zarówno aplikacja webowa jak i serwer mają przygotowane pliki Dockerfile, najlepiej je uruchomić na osobnych terminalach:

```
$ cd ScalaProject-main/server
$ docker build -t server .
$ docker run -p 3001:3001 server
```

oraz

```
$ cd ../webapp/target/docker/stage
$ docker build -t webapp .
$ docker run --network=host -p 9000:9000 webapp
```

Konieczne jest uruchomienie serwera w pierwszej kolejności, gdyż aplikacja webowa posiada autoryzację łączącą się od razu z bazą.

2.Struktura bazy danych

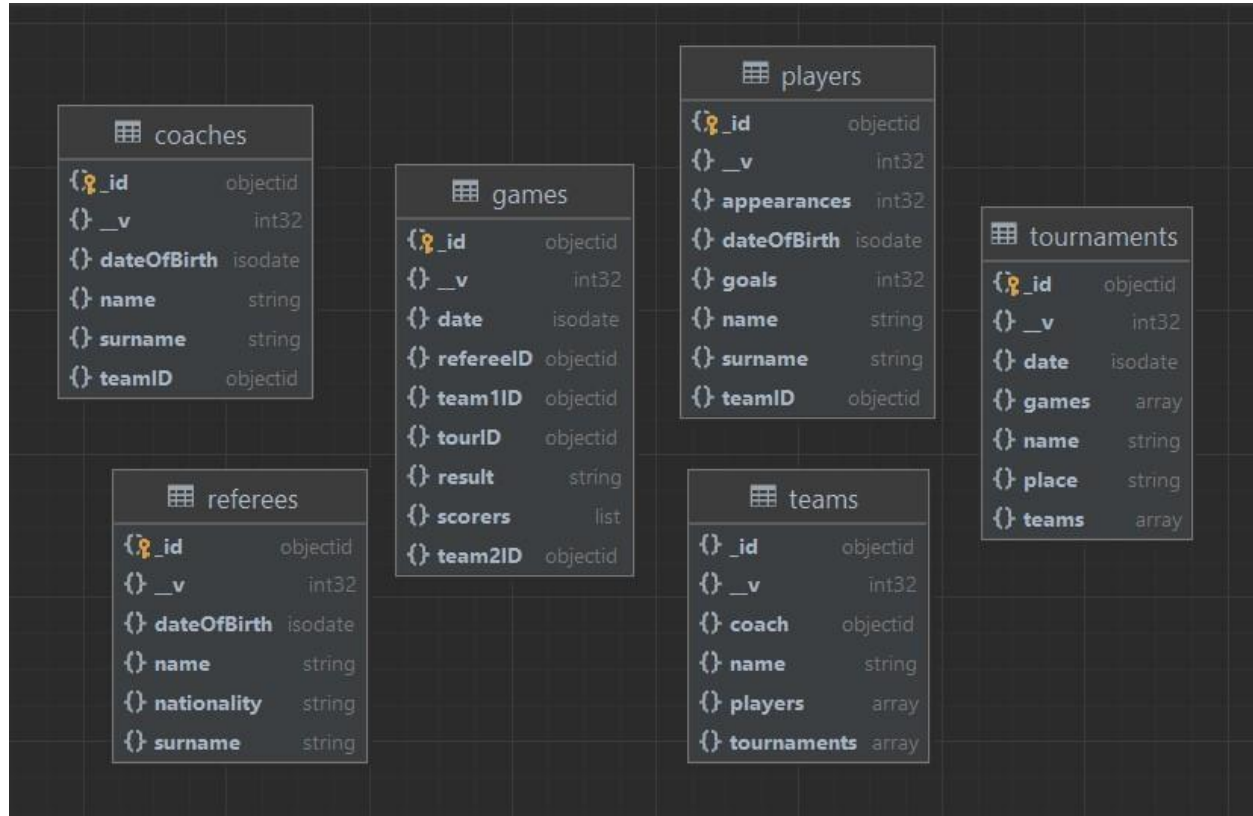
Ze względu na potrzebę zapewnienia wszystkich informacji i statystyk dotyczących rozgrywek zdecydowaliśmy, że w bazie powinny znaleźć się następujące dane:

- turniej - kto w nim uczestniczył, jakie mecze zostały rozegrane, gdzie i kiedy się odbył
- mecz - kto z kim grał, o której godzinie dokładnie, jaki był wynik, kto sędziował i kto zdobywał punkty (zrezygnowaliśmy z kartek i asyst, na naszych turniejach wszyscy grają przepisowo :))))
- drużyna - w jakich turniejach uczestniczyła, jakich posiada zawodników, jakiego trenera
- trener - czy jest zatrudniony; jeśli tak to w jakim klubie
- zawodnik - w jakim klubie gra, ile ma występów a ile punktów zdobył

- sędzia - jakiej jest narodowości (organizatorzy turniejów stawiają na multikulturowość)

Oczywiście stosowne dokumenty powinny mieć podstawowe informacje typu imię i nazwisko, data urodzenia lub nazwa ogólna.

Początkowo wytypowaliśmy poniższą strukturę bazy dokumentowej do naszego projektu:



Dokładny opis kolekcji z przykładami zawieramy poniżej:

1. **players** — kolekcja zawierająca dane oraz statystyki zawodników (występy i bramki), każdy zawodnik może należeć do maksymalnie 1 klubu, może też być wolnym zawodnikiem
 - a. *name* - imię zawodnika (String)
 - b. *surname* - nazwisko zawodnika (String)
 - c. *dateOfBirth* - data urodzenia (Date)
 - d. *appearances* - liczba występów (Int)
 - e. *goals* - liczba zdobytych punktów (Int)
 - f. *teamID* - identyfikator klubu zawodnika (ObjectId)

```
{
  "_id": {"$oid": "628198719f6938c26998f69d"},
  "__v": 0,
  "appearances": 0,
  "dateOfBirth": {"$date": "2000-02-28T23:00:00.000Z"},
  "goals": 0,
  "name": "Bartosz",
  "surname": "Wiśniewski",
  "teamID": {"$oid": "62829c5030c158c42a071d3c"}
},
```

Zawodnicy (ze względu na dodanie informacji o zdobywcach punktów w poszczególnych meczach) nie mogli być wyłącznie zagnieżdżonymi dokumentami wewnątrz zespołów.

2. **coaches** - kolekcja zawierająca informacje o trenerach

- name* - imię trenera (String)
- surname* - nazwisko trenera (String)
- dateOfBirth* - data urodzenia (String)
- teamID* - identyfikator prowadzonej drużyny (ObjectId)

```
{
  "_id": {"$oid": "6282ea5396509a3a077f4eac"},
  "__v": 0,
  "dateOfBirth": {"$date": "1972-03-14T23:00:00.000Z"},
  "name": "Józef",
  "surname": "Lustgarten",
  "teamID": {"$oid": "6284ee916498af0b0f7e36c9"}
},
```

Wstępnie rozważaliśmy trenerów jako zagnieżdżone dokumenty w drużynach, jednakże w trakcie pracy wyszło, że przydałaby się grupa trenerów bezrobotnych, których można przydzielać i wymieniać między klubami w związku z czym uczyniliśmy z ich osobną kolekcją.

3. **teams** — kolekcja zawierająca dane o klubach, w tym identyfikatory ich zawodników oraz turnieje, w których uczestniczył zespół

- name* - nazwa klubu (String)
- coachID* - identyfikator trenera (ObjectId)
- players* - tablica identyfikatorów zawodników (Array[ObjectId])
- tournaments* - tablica identyfikatorów turniejów (Array[ObjectId])

```
{
  "_id": {"$oid": "62829a662a6012913c319c66"},
  "__v": 0,
  "coach": {"$oid": "62828bf50533c8dcc9473016"},
  "name": "Real Madryt",
  "players": [
    {"$oid": "628198319f6938c26998f699"},
    {"$oid": "628198609f6938c26998f69b"},
    {"$oid": "628198719f6938c26998f69d"},
    {"$oid": "628198f19f6938c26998f69f"}
  ],
  "tournaments": [
    {"$oid": "62848e28c644bcc7aa5b89bf"}
  ]
},
```

4. **referees** - kolekcja zawierająca dane sędziów

- name* - imię sędziego (String)
- surname* - nazwisko sędziego (String)
- dateOfBirth* - data urodzenia sędziego (Date)
- nationality* - narodowość sędziego (String)

```
{
  "_id": {"$oid": "627e2e67210dfa6ba18f0893"},
  "__v": 0,
  "dateOfBirth": {"$date": "1980-03-31T00:00:00.000Z"},
  "name": "Leszek",
  "nationality": "Polska",
  "surname": "Siwik"
},
```

Sędziowie musieli zostać wyeksportowani w związku z możliwością prowadzenia wielu spotkań.

5. **tournaments** - kolekcja zawierająca informacje o rozgrywkach

- name* - nazwa turnieju (String)
- place* - miejsce rozgrywek (String)
- date* - termin rozpoczęcia rozgrywek (Date)
- teams* - tablica identyfikatorów zespołów biorących udział w turnieju (Array[ObjectId])
- games* - tablica identyfikatorów spotkań rozegranych w ramach turnieju (Array[ObjectId])

```
{
  "_id": {"$oid": "62848e28c644bcc7aa5b89bf"},
  "__v": 0,
  "date": {"$date": "2022-05-12T22:00:00.000Z"},
  "games": [
    {"$oid": "62848e28c644bcc7aa5b89c8"},
    {"$oid": "62848e28c644bcc7aa5b89cb"},
    {"$oid": "62848e28c644bcc7aa5b89ce"}
  ],
  "name": "Liga Mistrzów",
  "place": "Wilamowice",
  "teams": [
    {"$oid": "62829a662a6012913c319c66"},
    {"$oid": "62829b412a6012913c319c87"},
    {"$oid": "62829c5030c158c42a071d3c"}
  ]
},
```

6. **games** - kolekcja zawierająca dane dotyczące meczów między zespołami
- tournamentID* - identyfikator turnieju w ramach którego rozgrywany był mecz (ObjectId)
 - team1ID* - identyfikator pierwszej drużyny (ObjectId)
 - team2ID* - identyfikator drugiej drużyny (ObjectId)
 - refereeID* - identyfikator sędziego (ObjectId)
 - result* - końcowy wynik meczu (String)
 - date* - termin spotkania (DateTime)
 - scorers* - tablica identyfikatorów strzelców bramek (Array[ObjectId])

```
{
  "_id": {"$oid": "62848e28c644bcc7aa5b89c8"},
  "__v": 0,
  "date": {"$date": "2022-05-20T15:15:00.000Z"},
  "refereeID": {"$oid": "6282e9e296509a3a077f4eab"},
  "result": "3-4",
  "scorers": [
    {"$oid": "628199089f6938c26998f6a1"},
    {"$oid": "628198319f6938c26998f699"},
    {"$oid": "628198719f6938c26998f69d"}
  ],
  "team1ID": {"$oid": "62829a662a6012913c319c66"},
  "team2ID": {"$oid": "62829b412a6012913c319c87"},
  "tourID": {"$oid": "62848e28c644bcc7aa5b89bf"}
},
```

Po analizie modelu uznaliśmy, że kolekcja gier niekoniecznie jest potrzebna osobno (sprawia tylko dodatkowe problemy z związane z pipeline'ami po stronie frontendu), w związku z czym wzięliśmy pod uwagę zagnieżdżenie dokumentów z grami wewnątrz kolekcji turniejów.

Tak wyglądałoby to po zmianie:

```
{
  "_id": {"$oid": "62848e28c644bcc7aa5b89bf"},
  "__v": 0,
  "date": {"$date": "2022-05-12T22:00:00.000Z"},
  "games": [
    {
      "date": {"$date": "2022-05-20T15:15:00.000Z"},
      "refereeID": {"$oid": "6282e9e296509a3a077f4eab"},
      "result": "3-4",
      "scorers": [
        {"$oid": "628199089f6938c26998f6a1"},
        {"$oid": "628198319f6938c26998f699"},
        {"$oid": "628198719f6938c26998f69d"}
      ],
      "team1ID": {"$oid": "62829a662a6012913c319c66"},
      "team2ID": {"$oid": "62829b412a6012913c319c87"}
    },
    {
      "date": {"$date": "2022-05-17T15:30:00.000Z"},
      "refereeID": {"$oid": "6282e9c396509a3a077f4eaa"},
      "result": "0-0",
      "scorers": [],
      "team1ID": {"$oid": "62829b412a6012913c319c87"},
      "team2ID": {"$oid": "62829c5030c158c42a071d3c"}
    }
  ],
  "name": "Liga Mistrzów",
  "place": "Wilamowice",
  "teams": [
    {"$oid": "62829a662a6012913c319c66"},
    {"$oid": "62829b412a6012913c319c87"},
    {"$oid": "62829c5030c158c42a071d3c"} ]
}
```

Jednakże, okazało się, że dostęp do wszystkich gier jest potrzebny w widoku głównym, gdzie wyświetlają się zbliżające się mecze. Zrezygnowaliśmy zatem z tego pomysłu.

Na głównej stronie aplikacji wyświetlamy dokumenty uszeregowane według pewnych pól. Nie ma chyba lepszej możliwości do skorzystania z indeksów:

a.) indeks porządkujący graczy według strzelonych bramek i rozegranych spotkań


```
db.players.getIndexes()
```

Output Result 3

2 rows

	key	name	v
1	{"_id": new NumberInt("1")}	_id_	2
2	{"goals": new NumberInt("1"), "appearances": new NumberInt("-1")}	goals_1_appearances_-1	2

b.) indeks porządkujący turnieje według daty rozpoczęcia

```
db.tournaments.getIndexes()
```

Output Result 4

2 rows

	key	name	v
1	{"_id": new NumberInt("1")}	_id_	2
2	{"date": new NumberInt("1")}	date_1	2

c.) indeks porządkujący mecze według daty rozegrania

```
db.games.getIndexes()
```

Output Result 5

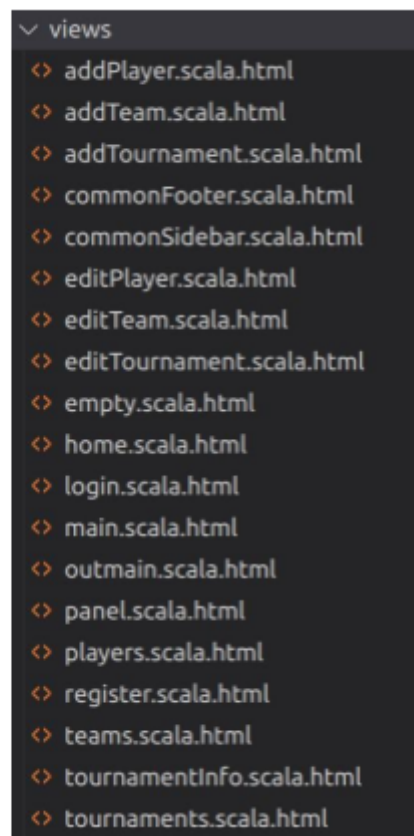
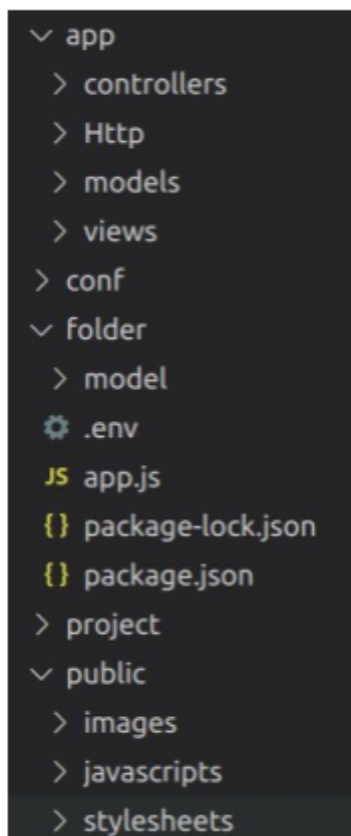
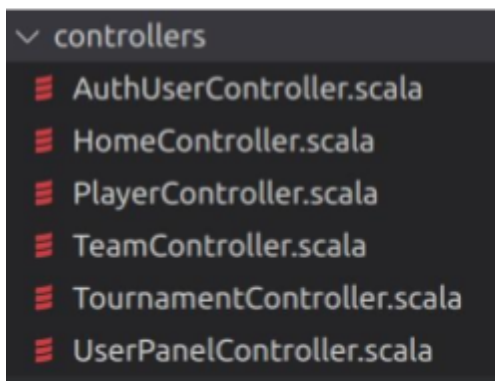
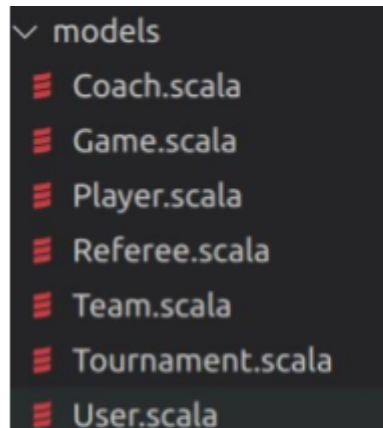
2 rows

	key	name	v
1	{"_id": new NumberInt("1")}	_id_	2
2	{"date": new NumberInt("1")}	date_1	2

3. Aplikacja Frontendowa

Głównym problem jaki napotkaliśmy w pracy z frameworkiem Scala Play była bardzo uboga dokumentacja oraz brak możliwości odwoływania się do globalnych serwisów lub zmiennych.

Struktura projektu (uwzględniając pliki, w które istotnie zawierały efekt naszej pracy) wyglądała następująco:



Każdy widok aplikacji odpowiadający kolekcji z bazy danych (Tournaments, Players, Teams) musiał mieć osobno wstrzykiwane kolekcje obiektów

```
Ok((views.html.teams("addTeam"))(views.html.addTeam(teamForm)(players)(availableCoaches))(teams)(coaches))
```

Użyliśmy do tego mechanizmów case classes w celu mapowania pobranych z bazy kolekcji dokumentów na kolekcję konkretnych obiektów oraz pipeline'ów dzięki którym bardzo łatwe okazało się lawirowanie między danymi i wyszukiwanie tych pożądaných.

```
case class Player (val _id: String, var name : String,
                  var surname : String, val dateOfBirth : Date,
                  var goals : Int, var appearances : Int, var teamID : String){
  def getData(): String = {
    name + " " + surname
  }
}
```

```
val tournaments = Http.HttpRequestHandler.getTournaments.filter(el => el.teams.contains(index))
val players = Http.HttpRequestHandler.getPlayers.filter(el => el.teamID == index)
```

Ciekawym mechanizmem okazały się argumenty typu implicit, które w przypadku nie określenia jawnie argumentów są automatycznie dostarczane przez kompilator choćby przy obsłudze requestów i wyświetlaniu wiadomości:

```
@(playerForm: Form[playerData])(teams:Seq[Team])
|(implicit request: RequestHeader, messagesProvider: MessagesProvider)
```


Po uruchomieniu aplikacji powita nas widok zalogowania (można również się zarejestrować):

LOGIN

Username:

michal

Password:

.....

Submit

[REGISTER](#)

Na głównej stronie aplikacji wyświetlają się najlepsi zawodnicy (liczba punktów) oraz zbliżające się spotkania i turnieje:

Top scorers

Min number of goals

1

Show

Leo Messi	Borussia Dortmund	7
Jakub Blaszczykowski	Wisła Kraków	5

Upcoming games

Real Madryt	Borussia Dortmund	2022-06-20 15:40
Real Madryt	Bayern Monachium	2022-06-23 15:40
Borussia Dortmund	Bayern Monachium	2022-07-01 20:45

Upcoming tournaments

Liga Mistrzów	2022-05-30	Poznań
---------------	------------	--------

W kolejnych zakładkach znajdują się narzędzia do dodawania, edytowania i usuwania dokumentów z konkretnych kolekcji:

a.) Turnieje

Name

Place

Data

dd . mm . rrrr

Teams

Real Madryt
Borussia Dortmund
Bayern Monachium
Wisła Kraków
Puszcza Niepołomice
Chrobry Głogów

Add tournament

Tournament name	Place	Date		
Liga Mistrzów	Poznań	2022-05-30	show	Edit
1 Liga	Kraków	2022-05-20	show	Edit

b.) Szczegółowy widok wewnątrz turnieju

Liga Mistrzów

Game

Real Madryt --- Borussia Dortmund

Result

Date

dd . mm . rrrr , -- : --

Referee

Leszek Siwik

Scorers

Bartłomiej Kowalczyk
Leo Messi
Adam Nowakowski
Łukasz Piszczek

Update game info

Team nr.1	Result	Team nr.2	Referee	Date of game	Scorers of goals
Real Madryt	---	Borussia Dortmund	Leszek Siwik	2022-06-20 15:40	
Real Madryt	---	Bayern Monachium	Leszek Siwik	2022-06-23 15:40	
Borussia Dortmund	---	Bayern Monachium	Leszek Siwik	2022-07-01 20:45	

c.) Drużyny

Name

Coach

Franciszek Smuda

Players

Gerwazy Grabowski
Krystian Zając

Add team

d.) Zawodnicy

Name

Surname

Date of birth

Team

Real Madryt

Goals

Appearances

Add player

Name and surname	Team	Date of birth	Appearances	Goals		
Adam Nowakowski	Borussia Dortmund	1999-12-12	0	0	Delete	Edit
Bartosz Wiśniewski	Real Madryt	2000-02-28	0	0	Delete	Edit
Michał Wójcik	Bayern Monachium	2000-01-08	0	0	Delete	Edit
Bartłomiej Kowalczyk	Bayern Monachium	2000-01-09	0	0	Delete	Edit
Cyprian Kamiński	Chrobry Głogów	2000-01-08	0	0	Delete	Edit
Dariusz Lewandowski	Puszcza Niepołomice	2000-01-06	0	0	Delete	Edit
Wojciech Kwiatkowski	Puszcza Niepołomice	2000-01-16	0	0	Delete	Edit
Krzysztof Wojciechowski	Puszcza Niepołomice	2000-01-20	0	0	Delete	Edit

e.) Panel z danymi dotyczącymi konta oraz możliwością dodania sędziów

Hello michał!

Your roles:

User

Your account is OK!

Name

Surname

Date of birth

Nationality

England

Add referee

Name and surname	Date of birth	Nationality	
Leszek Siwik	1980-03-31	Polska	Delete
Tomasz Mikulski	1980-12-31	Polska	Delete
Maciej Wierzbowski	1981-01-02	Polska	Delete
Robert Małek	1981-01-04	Polska	Delete
Marcin Szulc	1981-01-06	Polska	Delete
Paweł Gil	1981-01-08	Polska	Delete
Jose Minera	1983-06-09	Brazylia	Delete
Robert Marcjan	1974-05-15	France	Delete

Kliknięcie Edit w dowolnej kolekcji uaktywnia ukryty formularz edycji istniejących dokumentów w bazie, natomiast Delete usuwa takowe.

Zbieżność imion i nazwisk wśród sędziów i zawodników zupełnie przypadkowa ;)

Mateusz Skowron
Kamil Miśkowiec