

Werkwijze Club Brugge Player Stats app

Live demo: <https://master.dxxhdfgeumfxwc.amplifyapp.com/>

Github link: <https://github.com/mablomma/club-stats>

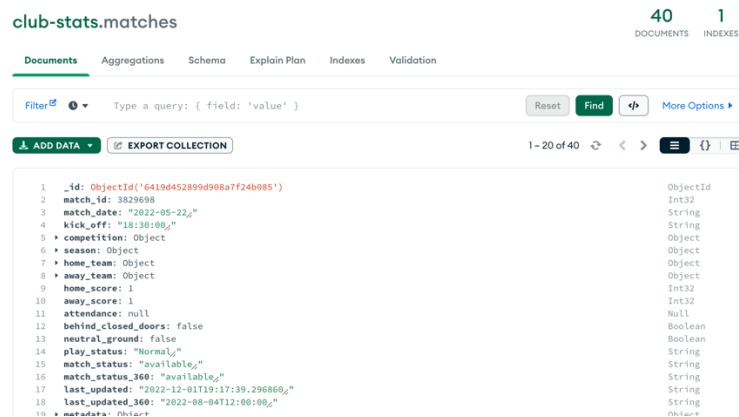
Stap 1: Data inspectie en cleaning

- Gebruik van **Jupyter notebook** voor cleaning (**Python 3**)
- Inlezen van de aparte **'match'**-json-files en vervolgens **samenvoegen tot 1 json-array**, ter voorbereiding van het uploaden naar een mongodb database (collection 'matches').
- Inlezen van de aparte **'info'**-json-files en vervolgens **samenvoegen tot 1 json-array**, ter voorbereiding van het uploaden naar een mongodb database (collection 'stats').
- De attributen onder **home_team** en **away_team** heb ik aangepast: de prefixen van de attributen (**home_team_name**, **home_team_id**, ...) heb ik verwijderd zodat het object onder **home_team** en **away_team** **eenzelfde layout krijgt van type Team**.
- Oplijsten van alle unieke teams en unieke spelers om vervolgens voor deze **teams/spelers clubiconen en afbeeldingen** te verzamelen.

id		name		
0	333	KV Oostende		
1	534	KRC Genk		
2	977	Club Brugge		
3	986	RSC Anderlecht	player_id	player_name
4	991	Standard Liège	0	2937
5	1038	Zulte-Waregem	1	3279
6	1039	Kortrijk	2	3539
7	1040	Antwerp	3	4872
8	1044	Gent	4	6405
9	1045	Sporting Charleroi	5	6844
10	1046	Cercle Brugge	6	9125
11	1048	Sint-Truiden	7	10594
12	1050	AS Eupen	8	11116
13	1352	Mechelen	9	12259
14	2390	Beerschot-Wilrijk	10	16281
15	2927	OH Leuven	11	16282
16	5328	RFC Seraing		

Stap 2: Backend met graphql

- Maken van MongoDB **model** voor zowel de 'matches' collectie als de 'stats' collectie, gebruik makende van package **mongoose**.
- Opzetten van een **apollo graphql server**
 - Aanmaken nodige **typeddefinitions**
 - Aanmaken nodige **resolvers**
- Bij het ophalen van een match object (by ID), kan de frontend het veld **player_stats** toevoegen. Enkel indien de query dit veld bevat wordt niet alleen de 'matches' collectie gelezen met mongoose maar ook de 'stats' collectie. Alle stat records die betrekking hebben op de opgevraagde match worden dan toegevoegd in een array onder het veld 'player_stats'.



Stap 3: Design in Adobe XD

- Design heb ik gemaakt in Adobe XD. Heb verschillende designs herleid tot finaal design. (Initieel design was minder geschikt voor brede schermen en groter aantal matchen).



Stap 4: Frontend met React

- Creatie van volgende componenten
 - **Header** met branding
 - **Footer**
 - **Matches** component: deze component bevat een oplistijng van alle matchen van Club Brugge van het seizoen 2021-2022. Elke match heeft zijn eigen kaart:
 - **MatchCard** component: deze component geeft voor elke match de datum, teams, score en snellink naar stats weer in een vaste layout.
 - **Match** component: deze component bevat de details van een bepaalde match. Bovenop de data die we ook al terugvinden in de MatchCard component zijn er nog enkel details toegevoegd: stadion, scheidsrechter en spelerstatistieken. Voor deze statistieken heb ik een aparte component aangemaakt:
 - **PlayerStats** component: deze component bevat een tabel met verschillende statistieken voor elke speler. Ik heb een **PieChart** component gemaakt om de data visueel te ondersteunen.
- Voor **routing** heb ik gebruik gemaakt van de package 'react-router-dom'. Ik heb een **scrollToTop** component toegevoegd zodat bij het veranderen van route de scroll positie telkens hersteld wordt.

- Voor het inlezen van de **graphql** data heb ik opnieuw gebruik gemaakt van **Apollo**.