

Ward Gauderis  
20183431  
10/01/2021

Gedistribueerde Systemen  
Informatica  
UAntwerpen

# Verslag

## Project Gedistribueerde Systemen

# Inhoudsopgave

<b>1</b>	<b>Praktisch</b>	<b>3</b>
<b>2</b>	<b>Tools</b>	<b>4</b>
<b>3</b>	<b>Design</b>	<b>5</b>
3.1	Nginx . . . . .	5
3.2	Auth . . . . .	5
3.3	Stats . . . . .	6
3.4	CRUD . . . . .	6
3.5	Database . . . . .	6
3.6	Frontend . . . . .	6
<b>4</b>	<b>API</b>	<b>7</b>
4.1	Auth . . . . .	9
4.2	Stats . . . . .	9
4.3	CRUD . . . . .	12
4.3.1	Club . . . . .	12
4.3.2	Team . . . . .	14
4.3.3	Division . . . . .	16
4.3.4	Match . . . . .	18
4.3.5	Referee . . . . .	21
4.3.6	User . . . . .	22

# 1 Praktisch

De webapplicatie is opgesplitst in verschillende Docker containers die beheerd worden met Docker-Compose. Enkel het docker-compose commando, een draaiende Docker daemon en Docker gebruikersrechten zijn noodzakelijk om de applicatie op te starten. Hiervoor zijn de scriptjes **install.sh** en **run.sh** voorzien. Binnen de containers zelf worden automatisch de verschillende dependencies geïnstalleerd.

Zowel de API als de bijhorende site zijn beschikbaar op poort 80 van **localhost**. De applicatie is geïnitieerd met de data die werd voorzien via Blackboard. Gebruikers van de applicatie vallen in 5 categorieën (combinaties zijn mogelijk) met verschillende rechten. Voor elk van deze gebruikers is er een account aangemaakt waarmee de applicatie kan getest worden.

Gebruikersrol	Username	Password
Anonieme gebruiker		
Geregistreerde gebruiker	Voetbalfan	Voetbalfan
Geregistreerde gebruiker gelinkt aan een team	Voetballer	Voetballer
Administator	Admin	Admin
Superadministrator	Super	Super

## 2 Tools

Wat betreft de tools heb ik me gehouden aan de lijst van de opdracht, met enkele aanpassingen en toevoegingen.

In plaats van Bootstrap, gebruik ik Bulma als CSS framework. In het Programming Project Databases van vorig jaar gebruikten we exact dezelfde technologieën als in dit project en toen hadden we voornamelijk spijt van het feit dat we Bootstrap hadden gebruikt omwille van de verborgen complexiteit. Bulma is een meer light-weight variant hierop zonder javascript.

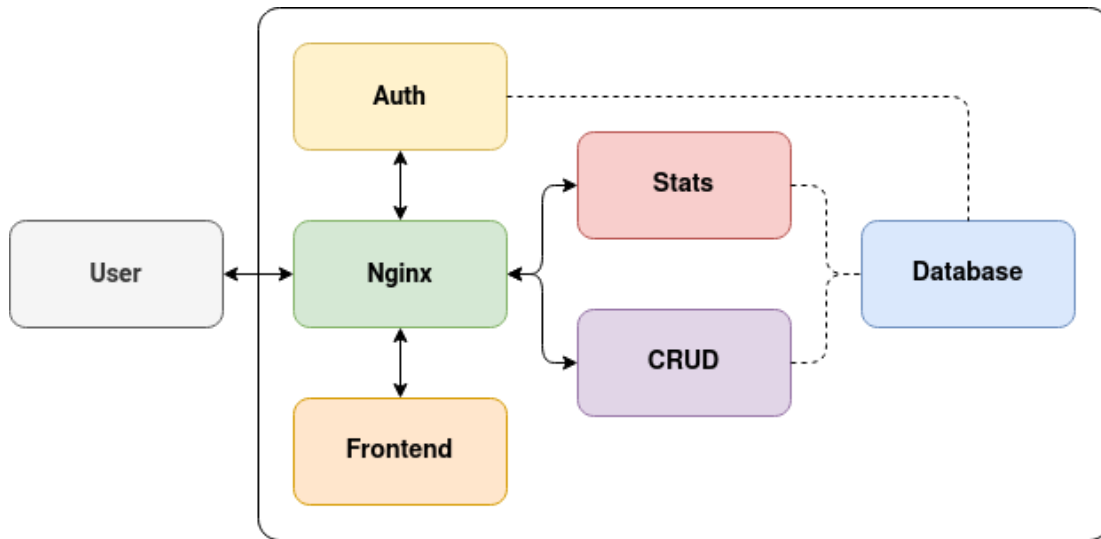
Eveneens gebruik ik voor de geocoding niet de Google Geocoding API maar Positionstack omdat deze service minder persoonlijke data van mij opvroeg om een gratis account aan te maken.

Verder maak ik gebruik van JSON Web Tokens voor de token-based authentication bij de API, van Flask-Login voor de session based authentication bij de website frontend, van WTForms om enkele aspecten van de HTML-forms te vereenvoudigen, van de webserver Nginx als reverse proxy die het verkeer tussen de verschillende Docker containers en de gebruiker orkestreert en van Unicorn als WSGI application server om de communicatie tussen Nginx en Flask te overzien.

### 3 Design

De applicatie heb ik opgesplitst in 6 verschillende containers op basis van functionaliteit die voorzien wordt en de rechten die nodig zijn om deze functionaliteit te mogen gebruiken.

Het is altijd mogelijk om de backend (vooral de API bestaande uit **Stats** en **CRUD**) verder op te splitsen in meer microservices, maar dit leek me vooral meer werk met zich mee te brengen en niet veel directe voordelen.



Hier volgt een beschrijving per container.

#### 3.1 Nginx

De **Nginx** container regelt het verkeer tussen de verschillende containers en zorgt ervoor dat de applicatie voor de gebruiker er als één geheel uitziet. De gebruiker heeft enkel toegang tot poort 80 van localhost terwijl alle interne HTTP-verkeer via poorten 5000 gaat. **Nginx** bepaalt welke requests naar welke microservice moeten worden doorgestuurd. Eveneens worden voor de API requests die autorisatie nodig hebben autorisatierequests gestuurd naar **Auth** vooraleer deze requests te forwarden.

#### 3.2 Auth

De **Auth** service is een Flask applicatie die instaat voor de token-based authenticatie en de autorisatie van de API. Deze maakt gebruik van JSON web tokens die de gebruiker van de API moet voorzien in de autorisatieheader van zijn request. De service genereert JWT tokens op aanvraag van gebruikers met correcte HTTP basic access authentication. Daarnaast autoriseert de **Auth** service ook gebruikers voor toegang tot bepaalde delen van de API door te antwoorden op autorisatierequests van **Nginx**.

### 3.3 Stats

De **Stats** service vormt samen met **CRUD** en **Auth** de van buitenaf toegankelijke API. Deze Flask applicatie staat in voor alle functionaliteit die toegankelijk is voor alle gebruikers en die geen autorisatie vereist. Deze container communiceert met de **Database** om alle vereiste statistieken uit de opdracht te voorzien, waaronder het genereren van de league tables, het oplijsten van bepaalde fixtures, het berekenen van de beste teams en het berekenen van statistieken omtrent een team of fixture. Deze service is eveneens verantwoordelijk voor het voorzien van weerdata voor toekomstige wedstrijden en interageert dus met de OpenWeatherMap en Positionstack API.

### 3.4 CRUD

De **CRUD** service staat in voor alle functionaliteit van de API die niet voor alle gebruikers toegankelijk is. Requests hiernaar moeten geautoriseerd worden door de **Auth** service. Zoals de naam aangeeft, is deze service dan ook voornamelijk verantwoordelijk voor de create, read, update en delete operaties op de verschillende entiteiten in de **Database**. Het grootste deel van deze operaties is enkel toegestaan voor admins en superadmins, maar zoals de opdracht aangeeft, moeten teamleden bijvoorbeeld ook toegang krijgen tot hun clubinformatie en scores. Eveneens zijn sommige readoperaties ook toegestaan voor alle gebruikers indien deze enkel publiek toegankelijke data opvragen. Net zoals bij alle API functionaliteit worden deze operaties sterk gecontroleerd op inconsistenties om interne databasefouten te vermijden.

### 3.5 Database

De **Database** container is een PostgreSQL server waarmee andere containers kunnen communiceren via poort 5432. Dit gebeurt langs hun kant via SQLAlchemy. De database is geïnitialiseerd met alle voorziene data plus vier gebruikers om te testen. Een groot deel van de design van de database werd bepaald door de structuur van deze voorziene data. Buiten de 5 entiteiten uit de opdracht, bevat de database ook nog een tabel voor de divisies. Seizoenen werden hierin niet opgenomen maar worden afgeleid uit de de matchdata om inconsistenties te vermijden. Verschillende constraints zijn aanwezig op de tabellen om de integriteit van de database te behouden.

### 3.6 Frontend

De **Frontend** service is een Flask applicatie die voor de gebruiker ervan communiceert met de achterliggende API. Deze service heeft dus geen speciale interne connecties met bv. de database en zou evengoed extern kunnen opereren of geschreven zijn door een gebruiker zelf. De **Frontend** voorziet een (Bulma) GUI voor alle API operaties, gecombineerd in een overzichtelijke website. Zo wordt de token-based login vertaald naar een session-based login en worden create, update en delete requests omgezet in HTML forms. Meldingen over incorrecte operaties uit de backend worden eveneens doorgepropageerd naar de frontend zodat de gebruiker weet wat hij verkeerd doet. Voor alle vereiste functionaliteit uit de opdracht is een pagina voorzien. Pagina's die enkel toegankelijk zijn voor gebruikers met bepaalde rechten worden verborgen voor andere gebruikers om verwarring te vermijden.

## 4 API

De van buitenaf aanspreekbare API is verspreid over de **Auth**, **Stats** en **CRUD** microservices en is beschikbaar via poort 80 op localhost/api. Als RESTful API wordt er gebruik gemaakt van de verschillende HTTP request methods en het json-formaat om duidelijker te zijn over het doel van de request. Hier volgt een beschrijving van alle aanspreekbare API endpoints.

Indien de status code van de response niet gelijk is aan 200, zal het bericht steeds de volgende vorm hebben:

```
{
  "code" : <status code>,
  "description" : <description>,
  "error" : <status message>
}
```

Velden die optioneel zijn, kunnen zowel in de body van de request als de response worden weggelaten en defaulten naar null. Indien vereiste velden missen in requests, wordt dit gemeld. Om de beschrijvingen van de API endpoints wat in te korten, geef ik hier al de JSON representatie van de verschillende entiteiten in de database.

Club

```
{
  "stam_number": <club stam number>,
  "name": <club name>,
  "address": <club address>,
  "zip_code": <club zip code>,
  "city": <club city>,
  "website": <optional club website>
}
```

Team

```
{
  "id" : <team id>,
  "stam_number": <club stam number>,
  "suffix": <optional team suffix>,
  "colors": <team colors>,
  "name": <full team name>
}
```

Division

```
{
  "id" : <division id>,
  "name" : <division name>
}
```

Match

```

{
  "id" : <match id>,
  "division_id" : <division id>,
  "matchweek" : <match week>,
  "date" : <match date>,
  "time" : <match time>,
  "home_team_id" : <team id>,
  "away_team_id" : <team id>,
  "goals_home_team" : <optional home team goals>,
  "goals_away_team" : <optional away team goals>,
  "status" : <optional status (Postponed, Canceled, Forfait)>,
  "referee_id" : <optional referee id>,
  "home_team_name" : <team name>,
  "away_team_name" : <team name>,
  "division_name" : <division name>,
  "referee_name" : <optional referee name>,
  "season" : <match season (bv. 2020 voor 2020-2021)>
}

```

#### Referee

```

{
  "id" : <referee id>,
  "first_name" : <referee first name>,
  "last_name" : <referee last name>,
  "address" : <referee address>,
  "zip_code" : <referee zip_code>,
  "city" : <referee city>,
  "phone_number" : <referee phone number>,
  "email" : <referee email>,
  "date_of_birth" : <referee date of birth>
}

```

#### User

```

{
  "id" : <user id>,
  "username" : <username>,
  "email" : <email>,
  "team_id" : <optional team id>,
  "is_admin" : <bool>,
  "is_super_admin" : <bool>
  "team_name": <optional team name>,
  "stam_number": <optional club stam number>
}

```



## 4.1 Auth

<b>post</b>	<b>/api/auth</b> <i>Vraag JWT authenticatie token aan met HTTP basic access authentication.</i>
Authorization: Basic <username:password in b64>	
<b>Response</b> <span>application/json</span>	
<b>200</b> ok	<pre>{   "token" : &lt;token&gt; }</pre>

## 4.2 Stats

Deze endpoints vereisen geen authenticatie.

<b>get</b>	<b>/api/stats/league_tables/&lt;div&gt;/&lt;season&gt;</b> <i>Vraag de league table op voor een bepaalde divisie en seizoen.</i>
<b>Parameter</b>	
div	divisie id of 0 voor alle
season	seizoen (bv. 2020 voor seizoen 2020-2021) of 0 voor altijd
<b>Response</b> <span>application/json</span>	
<b>200</b> ok	<pre>[{   "GA": &lt;goals against&gt;,   "GD": &lt;goal difference&gt;,   "GF": &lt;goals for&gt;,   "drawn": &lt;matches drawn&gt;,   "lost": &lt;matches lost&gt;,   "played": &lt;matches played&gt;,   "points": &lt;points&gt;,   "pos": &lt;position&gt;,   "team": &lt;TEAM&gt;   "won": &lt;matches won&gt; }, ...]</pre>

<b>get</b>	<b>/api/stats/fixtures/&lt;div&gt;/&lt;team&gt;/&lt;season&gt;</b> <i>Vraag de fixtures op voor een bepaalde divisie, team en seizoen.</i>
<b>Parameter</b>	
div	divisie id of 0 voor alle
team	team id of 0 voor alle
season	seizoen (bv. 2020 voor seizoen 2020-2021) of 0 voor toekomstig
<b>Response</b> <span style="float: right;">application/json</span>	
<b>200</b>	ok
<pre>[   &lt;MATCH&gt;, ... ]</pre>	

<b>get</b>	<b>/api/stats/top/&lt;season&gt;</b> <i>Vraag de teams met de beste attack, defense en clean sheets op per divisie voor een bepaald seizoen.</i>
<b>Parameter</b>	
season	seizoen (bv. 2020 vor seizoen 2020-2021) of 0 voor alle
<b>Response</b> <span style="float: right;">application/json</span>	
<b>200</b>	ok
<pre>[{   "best_attack": {     "GF": &lt;goals for&gt;,     &lt;TEAM&gt;   },   "best_defence": {     "GA": &lt;goals against&gt;,     &lt;TEAM&gt;   },   "most_clean_sheets": {     "count": &lt;clean sheet count&gt;,     &lt;TEAM&gt;   },   &lt;DIVISION&gt; }, ...]</pre>	

get	/api/stats/fixtures/<id> Vraag informatie op omtrent een fixture.
<b>Parameter</b>	
id	match id
<b>Response</b>	
200	ok
<pre> {   &lt;MATCH&gt;   "location": &lt;home team club location&gt;,    # enkel bij toekomstig fixtures   "previous_games": &lt;amount of games against each other&gt;,   "home_team_wins": &lt;amount of games home team won against     each other&gt;,   "away_team_wins": &lt;amount of games away team won against     each other&gt;,   "recent_matches": [      &lt;MATCH&gt;, ...   ],   "recent_matches_home_team": [{     "id": &lt;match id&gt;,     "result": &lt;W/L/D&gt;   }, ...],   "recent_matches_away_team": [{     "id": &lt;match id&gt;,     "result": &lt;W/L/D&gt;   }, ...]    # enkel voor matches komende week   # is een errormelding als de weer/geolocation services niet   correct reageren   "weather": {     "temp_max": &lt;maximum temperature&gt;,     "temp_min": &lt;minimum temperature&gt;,     "description": &lt;description&gt;,     "icon": &lt;icon id&gt;,     "humidity": &lt;humidity percentage&gt;,     "feels_like": &lt;feels like temperature&gt;   } } </pre>	

<b>get</b>	<b>/api/stats/teams/id</b> <i>Vraag informatie op omtrent een team.</i>
<b>Parameter</b>	
id	team id
<b>Response</b> <span>application/json</span>	
200	ok
<pre>[{   &lt;TEAM&gt;,   "recent_matches": [     &lt;MATCH&gt;, ...   ],   "future_matches": [     &lt;MATCH&gt;, ...   ],   "club": &lt;CLUB&gt; }, ...]</pre>	

## 4.3 CRUD

Bij alle PUT operaties zijn de velden optioneel, deze velden worden niet aangepast in de database.

### 4.3.1 Club

<b>get</b>	<b>/api/crud/clubs</b> <i>Verkrijg alle clubs.</i>
<b>Response</b> <span>application/json</span>	
200	ok
<pre>[   &lt;CLUB&gt;, ... ]</pre>	

<b>post</b>	<b>/api/crud/clubs</b> <i>Creëer club.</i>
Authorization: Bearer <token van admin of superadmin>	
<b>Body</b>	application/json
<pre>&lt;CLUB zonder stam_number&gt;</pre>	
<b>Response</b>	application/json
<b>200</b> ok	<pre>{   "stam_number": &lt;club stam number&gt; }</pre>

<b>get</b>	<b>/api/crud/clubs/&lt;stam_number&gt;</b> <i>Verkrijg club informatie.</i>
<b>Parameter</b>	
stam_number club stam number	
<b>Response</b>	application/json
<b>200</b> ok	<pre>&lt;CLUB&gt;</pre>

<b>delete</b>	<b>/api/crud/clubs/&lt;stam_number&gt;</b> <i>Verwijder club.</i>
Authorization: Bearer <token van admin of superadmin>	
<b>Parameter</b>	
stam_number club stam number	
<b>Response</b>	application/json
<b>200</b> ok	

<b>put</b>	<b>/api/crud/clubs/&lt;stam_number&gt;</b> <i>Verander club informatie.</i>
Authorization: Bearer <token van admin of superadmin of van een user gelinkt aan een team van deze club>	
<b>Parameter</b>	
stam_number	club stam number
<b>Body</b>	application/json
<pre>&lt;CLUB zonder stam_number&gt;</pre>	
<b>Response</b>	application/json
200	ok

#### 4.3.2 Team

<b>get</b>	<b>/api/crud/teams</b> <i>Verkrijg alle teams.</i>
<b>Response</b>	
200	ok
<pre>[   &lt;TEAM&gt;, ... ]</pre>	

<b>post</b>	<b>/api/crud/teams</b> <i>Creëer team.</i>
Authorization: Bearer <token van admin of superadmin>	
<b>Body</b>	application/json
<pre>&lt;TEAM zonder id en name&gt;</pre>	
<b>Response</b>	application/json
<b>200</b> ok	<pre>{   "id": &lt;team id&gt; }</pre>

<b>get</b>	<b>/api/crud/teams/&lt;id&gt;</b> <i>Verkrijg team informatie.</i>
<b>Parameter</b>	
id	team id
<b>Response</b>	application/json
<b>200</b> ok	<pre>&lt;TEAM&gt;</pre>

<b>delete</b>	<b>/api/crud/teams/&lt;id&gt;</b> <i>Verwijder team.</i>
Authorization: Bearer <token van admin of superadmin>	
<b>Parameter</b>	
id	team id
<b>Response</b>	application/json
<b>200</b> ok	

<b>put</b>	<b>/api/crud/teams/&lt;id&gt;</b> <i>Verander team informatie.</i>
Authorization: Bearer <token van admin of superadmin>	
<b>Parameter</b>	
id	team id
<b>Body</b>	application/json
<pre>&lt;TEAM zonder id en name&gt;</pre>	
<b>Response</b>	application/json
200	ok

#### 4.3.3 Division

<b>get</b>	<b>/api/crud/divisions</b> <i>Verkrijg alle divisions.</i>
<b>Response</b>	
200	ok
<pre>[   &lt;DIVISION&gt;, ... ]</pre>	



<b>post</b>	<b>/api/crud/divisions</b> <i>Creëer division.</i>
Authorization: Bearer <token van admin of superadmin>	
<b>Body</b>	application/json
<pre>&lt;DIVISION zonder id&gt;</pre>	
<b>Response</b>	application/json
<b>200</b> ok	<pre>{   "id": &lt;division id&gt; }</pre>

<b>get</b>	<b>/api/crud/divisions/&lt;id&gt;</b> <i>Verkrijg division informatie.</i>
<b>Parameter</b>	
id division id	
<b>Response</b>	application/json
<b>200</b> ok	<pre>&lt;DIVISION&gt;</pre>

<b>delete</b>	<b>/api/crud/divisions/&lt;id&gt;</b> <i>Verwijder division.</i>
Authorization: Bearer <token van admin of superadmin>	
<b>Parameter</b>	
id division id	
<b>Response</b>	application/json
<b>200</b> ok	

<b>put</b>	<b>/api/crud/divisions/&lt;id&gt;</b> <i>Verander division informatie.</i>
Authorization: Bearer <token van admin of superadmin>	
<b>Parameter</b>	
id	division id
<b>Body</b>	
	application/json
<pre>&lt;DIVISION zonder id&gt;</pre>	
<b>Response</b>	
	application/json
<b>200</b>	ok

#### 4.3.4 Match

<b>get</b>	<b>/api/crud/matches</b> <i>Verkrijg alle matches.</i>
<b>Parameter</b>	
season	Filter by season
division_id	Filter by division
team_id	Filter by team
<b>Response</b>	
	application/json
<b>200</b>	ok
<pre>[   &lt;MATCH&gt;, ... ]</pre>	

<b>post</b>	<b>/api/crud/matches</b> <i>Creëer match.</i>
Authorization: Bearer <token van admin of superadmin>	
<b>Body</b>	application/json
<pre>&lt;MATCH zonder id, home_team_name, away_team_name, division_name , referee_name en season&gt;</pre>	
<b>Response</b>	application/json
<b>200</b> ok	<pre>{   "id": &lt;match id&gt; }</pre>

<b>get</b>	<b>/api/crud/matches/&lt;id&gt;</b> <i>Verkrijg match informatie.</i>
<b>Parameter</b>	
id match id	
<b>Response</b>	application/json
<b>200</b> ok	<pre>&lt;MATCH&gt;</pre>

<b>delete</b>	<b>/api/crud/matches/&lt;id&gt;</b> <i>Verwijder match.</i>
Authorization: Bearer <token van admin of superadmin>	
<b>Parameter</b>	
id match id	
<b>Response</b>	application/json
<b>200</b> ok	

<b>put</b>	<b>/api/crud/matches/&lt;id&gt;</b> <i>Verander match informatie.</i>
Authorization: Bearer <token van admin of superadmin>	
<b>Parameter</b>	
id	match id
<b>Body</b>	application/json
<pre>&lt;MATCH zonder id, home_team_name, away_team_name, division_name , referee_name en season&gt;</pre>	
<b>Response</b>	application/json
<b>200</b>	ok

<b>patch</b>	<b>/api/crud/matches/&lt;id&gt;</b> <i>Verander match score.</i>
Authorization: Bearer <token van admin of superadmin of een user gelinkt aan het home team>	
<b>Parameter</b>	
id	match id
<b>Body</b>	application/json
<pre>{   "goals_home_team": &lt;optional match home team goals&gt;,   "goals_away_team": &lt;optional match away team goals&gt; }</pre>	
<b>Response</b>	application/json
<b>200</b>	ok

#### 4.3.5 Referee

<b>get</b>	<b>/api/crud/referees</b> <i>Verkrijg alle referees.</i>
Authorization: Bearer <token van admin of superadmin>	
<b>Response</b>	application/json
200 ok	<pre>[   &lt;REFEREE&gt;, ... ]</pre>

<b>post</b>	<b>/api/crud/referees</b> <i>Creëer referee.</i>
Authorization: Bearer <token van admin of superadmin>	
<b>Body</b>	application/json
	<pre>&lt;REFEREE zonder id&gt;</pre>
<b>Response</b>	application/json
200 ok	<pre>{   "id": &lt;referee id&gt; }</pre>

<b>get</b>	<b>/api/crud/referees/&lt;id&gt;</b> <i>Verkrijg referee informatie.</i>
Authorization: Bearer <token van admin of superadmin>	
<b>Parameter</b>	
id	referee id
<b>Response</b>	application/json
200 ok	<pre>&lt;REFEREE&gt;</pre>

<b>delete</b>	<b>/api/crud/referees/&lt;id&gt;</b> <i>Verwijder referee.</i>
Authorization: Bearer <token van admin of superadmin>	
<b>Parameter</b>	
id	referee id
<b>Response</b>	application/json
<b>200</b>	ok

<b>put</b>	<b>/api/crud/referees/&lt;id&gt;</b> <i>Verander referee informatie.</i>
Authorization: Bearer <token van admin of superadmin>	
<b>Parameter</b>	
id	match id
<b>Body</b>	application/json
<pre>&lt;REFEREE zonder id&gt;</pre>	
<b>Response</b>	application/json
<b>200</b>	ok

#### 4.3.6 User

<b>get</b>	<b>/api/crud/users</b> <i>Verkrijg alle users.</i>
Authorization: Bearer <token van admin of superadmin>	
<b>Response</b>	application/json
<b>200</b>	ok
<pre>[   &lt;USER&gt;, ... ]</pre>	

<b>post</b>	<b>/api/crud/users</b> <i>Creëer user.</i>
Authorization: Bearer <token van admin of superadmin>	
<b>Body</b>	application/json
<pre>&lt;USER zonder id, is_admin, is_super_admin, team_name en   stam_number maar met password&gt;</pre>	
<b>Response</b>	application/json
<b>200</b> ok	<pre>{   "id": &lt;user id&gt; }</pre>

<b>get</b>	<b>/api/crud/users/&lt;id&gt;</b> <i>Verkrijg user informatie.</i>
Authorization: Bearer <token van admin of superadmin of de user zelf>	
<b>Parameter</b>	
id	user id
<b>Response</b>	application/json
<b>200</b> ok	<pre>&lt;USER&gt;</pre>

<b>delete</b>	<b>/api/crud/users/&lt;id&gt;</b> <i>Verwijder user.</i>
Authorization: Bearer <token van admin of superadmin>	
<b>Parameter</b>	
id	user id
<b>Response</b>	application/json
<b>200</b> ok	

<b>put</b>	<b>/api/crud/users/&lt;id&gt;</b> <i>Verander user informatie.</i>
Authorization: Bearer <token van admin of superadmin>	
<b>Parameter</b>	
id	user id
<b>Body</b>	application/json
<pre>&lt;USER zonder id, is_admin, is_super_admin, team_name en   stam_number maar met password&gt;</pre>	
<b>Response</b>	application/json
<b>200</b>	ok

<b>post</b>	<b>/api/crud/admins/&lt;id&gt;</b> <i>Maak user admin.</i>
Authorization: Bearer <token van superadmin>	
<b>Parameter</b>	
id	user id
<b>Response</b>	application/json
<b>200</b>	ok

<b>delete</b>	<b>/api/crud/admins/&lt;id&gt;</b> <i>Verwijder user als admin.</i>
Authorization: Bearer <token van superadmin>	
<b>Parameter</b>	
id	admin id
<b>Response</b>	application/json
<b>200</b>	ok



get	/api/crud/seasons	Verkrijg alle seasons.
Response		application/json
200	ok	
		[ <season (bv. 2020 voor 2020-2021)>, ... ]