

# Floris Buyse (080000fb)

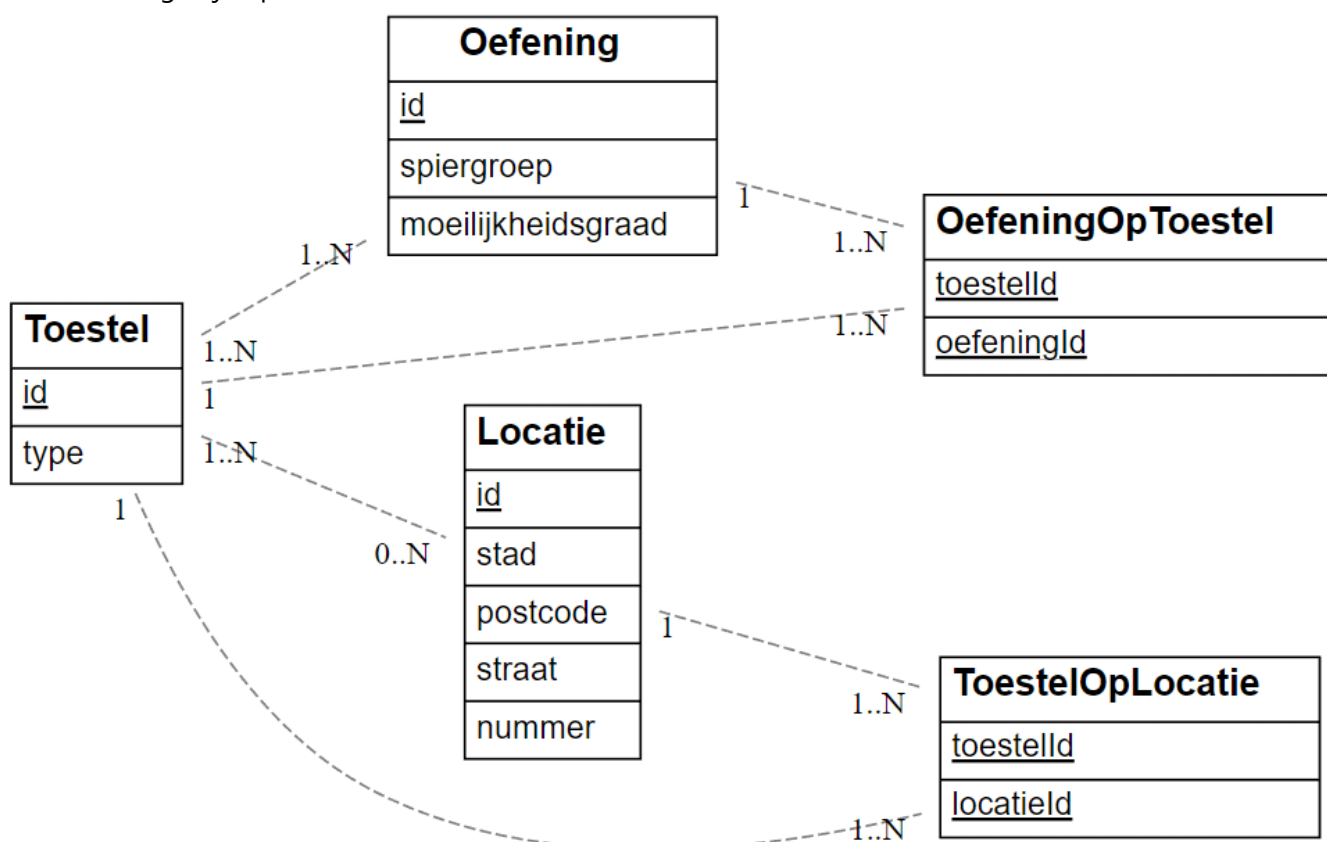
- ☐ Web Services: GITHUB URL
  - [GitHub repository](#)

## Logingegevens

- Gebruikersnaam/e-mailadres: e2e-testing@openluchtfitness.be
- Wachtwoord: OpenluchtIsKool12345

## Projectbeschrijving

Met mijn API kan je locaties van openlucht fitness ruimtes krijgen. Je kan ook zien welke toestellen er staan en welke oefeningen je op deze toestellen kan uitvoeren.



## Screenshots



GET <https://two223-webservices-floris-buyse-fb.onrender.com/api/locaties>

Params **Authorization** Headers (7) Body Pre-request Script Tests Settings

Type **Bearer Token**

The authorization header will be automatically generated when you send the request. Learn more about [authorization](#)

Heads up! These param variables

Token

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON

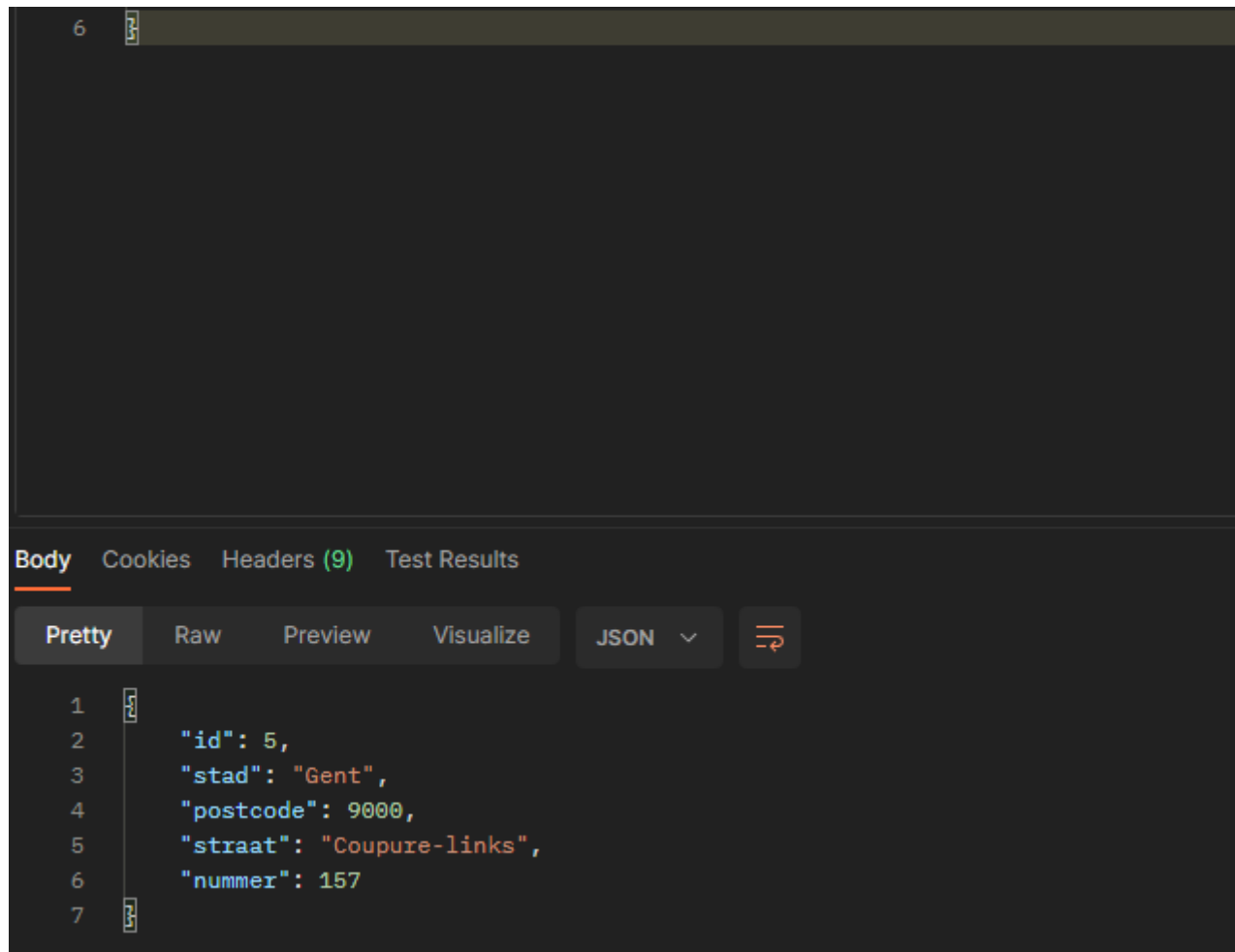
```
1  [
2    "items": [
3      {
4        "id": 3,
5        "stad": "Gent",
6        "postcode": 9000,
7        "straat": "Bijloke site",
8        "nummer": 10
9      },
10     {
11       "id": 1,
12       "stad": "Gent",
13       "postcode": 9000,
14       "straat": "Blaarmeersen",
15       "nummer": 1
16     },
17     {
18       "id": 2,
19       "stad": "Gent",
20       "postcode": 9000,
21       "straat": "Rabotpark",
22       "nummer": 95
23     }
24   ]
25 ]
```

POST <https://two223-webservices-floris-buyse-fb.onrender.com/api/locaties>

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1  [
2    "stad": "Gent",
3    "postcode": 9000,
4    "straat": "Coupure-links",
5    "nummer": 157
```




PUT ▼ <https://two223-webservices-floris-buyse-fb.onrender.com/api/locaties/5>

Params Authorization ● Headers (9) **Body** ● Pre-request Script Tests Settings

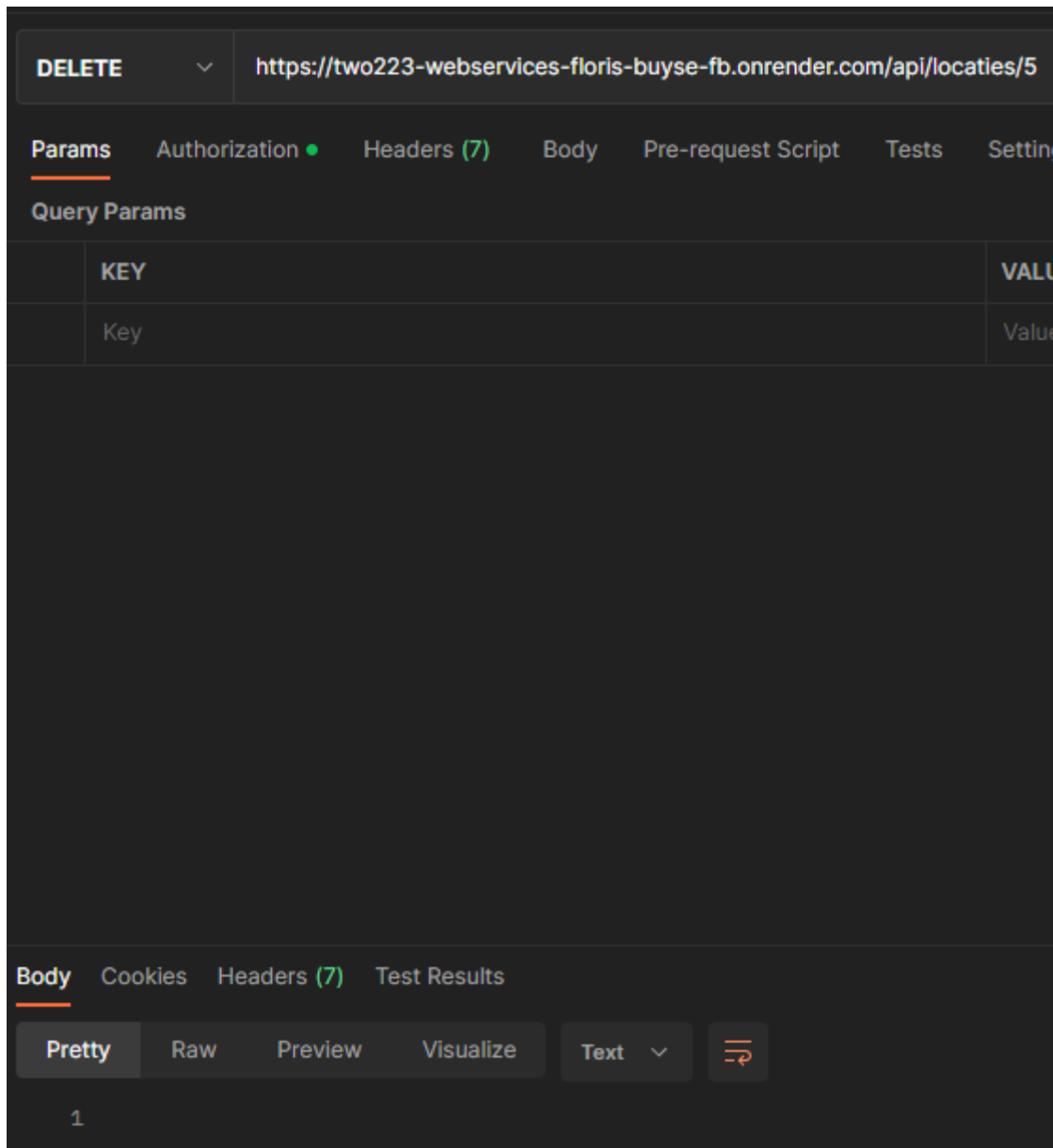
☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼

```
1 {
2   ...."stad": "Gent",
3   ...."postcode": 9000,
4   ...."straat": "Coupure-rechts",
5   ...."nummer": 99
6 }
```

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON ▼ 

```
1 {
2   "id": 5,
3   "stad": "Gent",
4   "postcode": 9000,
5   "straat": "Coupure-rechts",
6   "nummer": 99
7 }
```



## Behaalde minimumvereisten

### Web Services

- **data laag**
  - [ X ] voldoende complex (meer dan één tabel)
  - [ X ] één module beheert de connectie + connectie wordt gesloten bij sluiten server
  - [ X ] heeft migraties
  - [ X ] heeft seeds
- **repository laag**
  - [ X ] definieert één repository per entiteit (niet voor tussentabellen) - indien van toepassing
  - [ X ] mapt OO-rijke data naar relationele tabellen en vice versa
- **servicelaag met een zekere complexiteit**
  - [ X ] bevat alle domeinlogica
  - [ X ] bevat geen SQL-queries of databank-gerelateerde code

- **REST-laag**

- [ X ] meerdere routes met invoervalidatie
- [ X ] degelijke foutboodschappen
- [ X ] volgt de conventies van een RESTful API
- [ X ] bevat geen domeinlogica
- [ X ] degelijke autorisatie/authenticatie op alle routes

- **varia**

- [ X ] een aantal niet-triviale testen (min. 1 controller  $\geq 80\%$  coverage)
- [ X ] minstens één extra technologie
- [ X ] duidelijke en volledige **README .md**
- [ X ] maakt gebruik van de laatste ES6-features (object destructuring, spread operator...)
- [ X ] volledig en tijdig ingediend dossier

## Projectstructuur

### Web Services

Er is een src, een screenshots\_dossier map, een docs map, een coverage map, een config map en een tests map.

- In de src map kan je de verschillende lagen van de applicatie vinden (REST, data, service, repository).
- In de screenshots\_dossier map kan je de gebruikte screenshots vinden van het dossier.
- In de docs map kan je een html file vinden, die wanneer je deze in de liveserver beschouwd, je de api documentatie geeft.
- In de coverage map kan je een html file vinden, die wanneer je deze in de liveserver beschouwd, je de test coverage toont.
- In de config map kan je de environmental variables vinden.
- In de tests map kan je alle test suites vinden.

## Extra technologie

### Web Services

Ik heb ervoor gekozen om een andere package voor API documentatie te gebruiken. Ik gebruik namelijk apiDoc in plaats van Swagger.

[LINK NPM](#)

## Testresultaten

### Web Services

Ik heb een test voor locaties, toestellen en oefeningen. Hierbij test ik de CRUD-operaties, met authenticatie en validatie.

__tests__	<div><div></div></div>	100%	16/16	100%	0/0	100%	4/4	100%	16/16
config	<div><div></div></div>	100%	2/2	100%	0/0	100%	0/0	100%	2/2
src	<div><div></div></div>	56%	42/75	16.66%	5/30	75%	6/8	57.74%	41/71
src/core	<div><div></div></div>	67.79%	40/59	68.75%	11/16	52.63%	10/19	68.42%	39/57
src/data	<div><div></div></div>	80%	40/50	50%	3/6	100%	3/3	80%	40/50
src/data/migrations	<div><div></div></div>	88.46%	23/26	100%	0/0	66.66%	6/9	88.46%	23/26
src/repository	<div><div></div></div>	44.11%	45/102	100%	0/0	33.33%	6/18	44.11%	45/102
src/rest	<div><div></div></div>	82.99%	122/147	61.53%	16/26	46.15%	12/26	83.56%	122/146
src/service	<div><div></div></div>	60.31%	38/63	16.66%	1/6	29.41%	5/17	62.29%	38/61
_health.js	<div><div></div></div>	81.81%	9/11	100%	0/0	33.33%	1/3	81.81%	9/11
_locaties.js	<div><div></div></div>	100%	30/30	100%	0/0	100%	6/6	100%	30/30
_oefeningen.js	<div><div></div></div>	79.31%	23/29	100%	0/0	16.66%	1/6	79.31%	23/29
_toestellen.js	<div><div></div></div>	79.31%	23/29	100%	0/0	16.66%	1/6	79.31%	23/29
_validation.js	<div><div></div></div>	69.44%	25/36	61.53%	16/26	50%	2/4	71.42%	25/35
index.js	<div><div></div></div>	100%	12/12	100%	0/0	100%	1/1	100%	12/12

## Gekende bugs

### Web Services

Ik heb momenteel alleen een `locaties.spec.js` staan. Ik had ook `toestellen` en `oefeningen` toegevoegd maar als ik deze ook had toegevoegd kreeg ik een error namelijk: "knex is not a function". Hierdoor failen de migrations en bijgevolg falen de testen ook. Ik heb hiervoor nog geen oplossing gevonden.