Inleiding Datastandaard Fietsparkeerdata

Inhoud

Inleiding

Omdat de technische handleiding van de datastandaard fietsparkeerdata op https://github.com/Stichting-CROW/datastandaard-fietsparkeren niet voor iedereen even makkelijk leesbaar is, is mij verzocht deze standaard ook inzichtelijk te maken voor eenieder die zich minder op zijn gemak voelt bij tabellen, json en geneste objecten.

Ik raad de lezer echter aan na lezing van dit document ook een blik te werpen op genoemde technische documentatie. De json-notatie met de vele accolades en vierkante haken ziet er enger uit dan dat ze is. Een korte blik op een van de vele json-tutorials die het internet rijk is, bijvoorbeeld https://beginnersbook.com/2015/04/json-tutorial/, kan daarbij van pas komen.

Terug naar de datastandaard fietstellingen. Deze is zo ontworpen dat hij zowel geschikt is voor realtime bezettingsmetingen, die reeds in diverse stations- en bewaakte stallingen plaatsvinden, als voor incidentele straattellingen.

De data valt uiteen in drie hoofdonderdelen:

Data over een onderzoek: met gegevens over opdrachtgever, uitvoerder, tijdstip van onderzoek, etc. Het id van een onderzoek kan gebruikt worden om diverse metingen te koppelen aan dit bepaalde onderzoek.

Statische data: gegevens over stallingen en straten, zoals id's, namen en geografische afbakening.

Dynamische data: dit is uiteraard de data waar het allemaal om te doen is. Hoeveel tweewielers staan er gestald op een bepaalde plek? Wat voor soort tweewielers zijn dit? Fieten, bromfiets, scooters, …? Welke stallingsvoorzieningen staan er in het telgebied? Met welke capaciteit?

De datastandaard laat het toe tellingen tot op het kleinste detail op te slaan. Zo kun je bijvoorbeeld aangeven in welke staat een fiets verkeert of voor welke voertuigen een bepaalde voorziening geschikt is. Echter, dit hoeft niet en zal in de praktijk ook niet vaak gebeuren. Een dataleverancier kan ermee volstaan alleen door te geven hoeveel plekken er bezet zijn en verder niets.

Statische en dynamische data zijn gekoppeld aan meetgebieden, oftewel secties. Elke sectie heeft een unieke id, die door de dataleverancier bepaald kan worden. Het is daarom van essentieel belang er zeker van te zijn dat deze id's alleen voor de gegeven sectie gebruikt zullen worden. ID 'Dorpsstraat' is dus niet goed. 'Dorpsstraat_Ons_Dorp' wel.

In de statische data wordt algemene, nauwelijks aan veranderingen onderhevige data opgeslagen, zoals een straatnaam, een postcode en de geo-locatie.

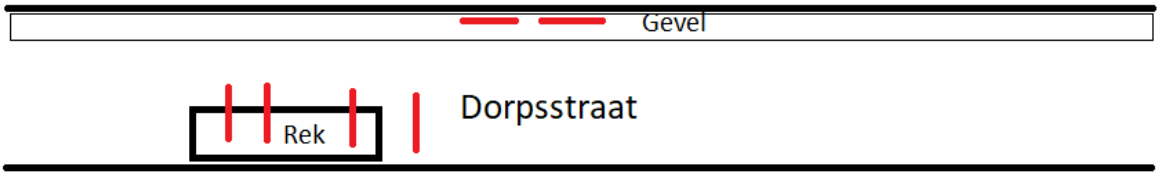
In de dynamische data staan de daadwerkelijk tellingen die door de sectieId worden gekoppeld aan de statische data.

Een paar voorbeelden van dynamische data

Afbeeldingen zeggen vaak meer dan woorden. Daarom een tweetal voorbeelden met afbeeldingen, die de essentie van de datastandaard direct duidelijk maken.

Voorbeeld 1: een straattelling in een doorsnee straat: Dorpsstraat, Ons Dorp

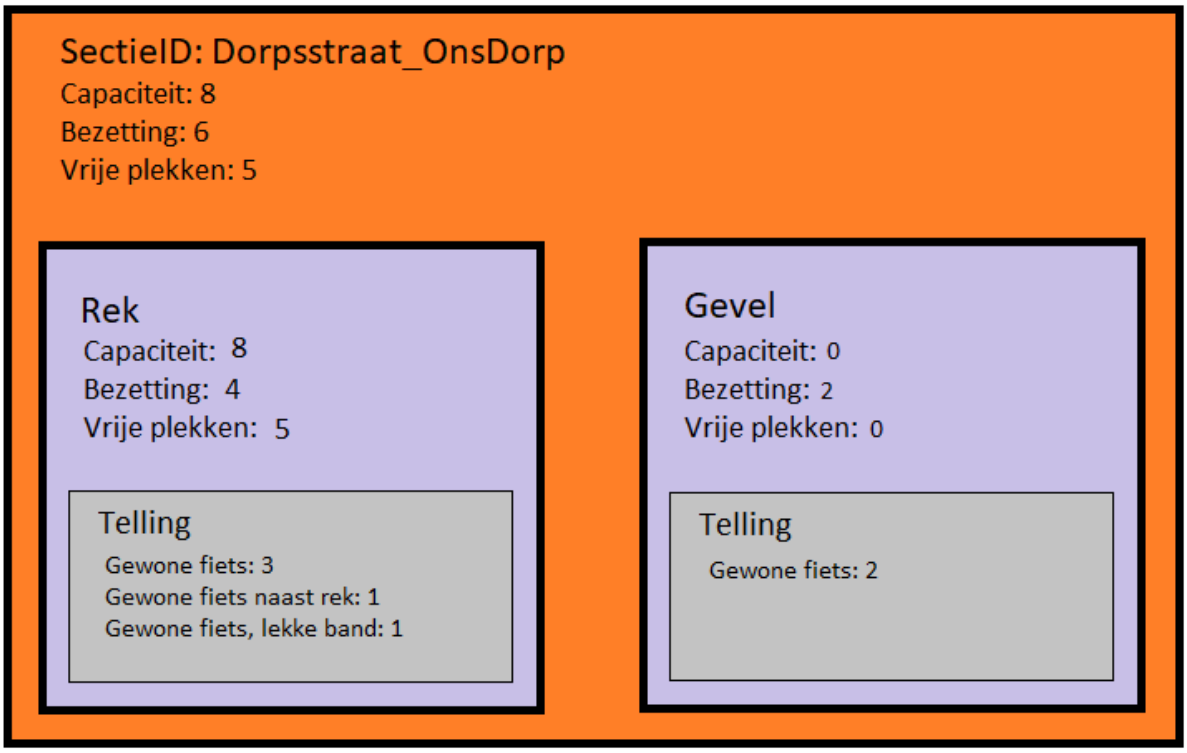
Een straattelling van een straat met een fietsenrek met 8 plekken, waarin 3 fietsen staan. 1 fiets staat naast het rek en twee fietsen staan tegen de gevel van een gebouw.



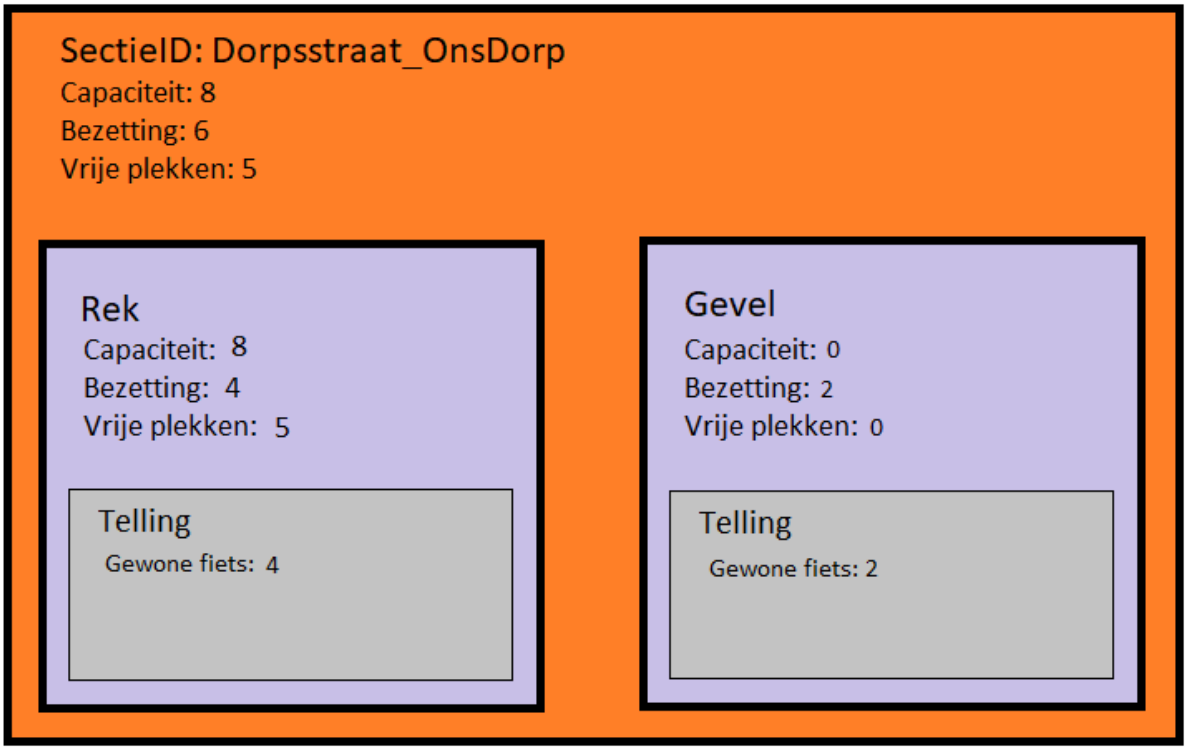
De straat is een sectie. Deze exacte locatie daarvan wordt vastgelegd in de statische data, bijvoorbeeld:

ID: Dorpsstraat_OnsDorp
Naam: Dorpsstraat
Geo-locatie: Polygon[hier een reeks coördinaten]

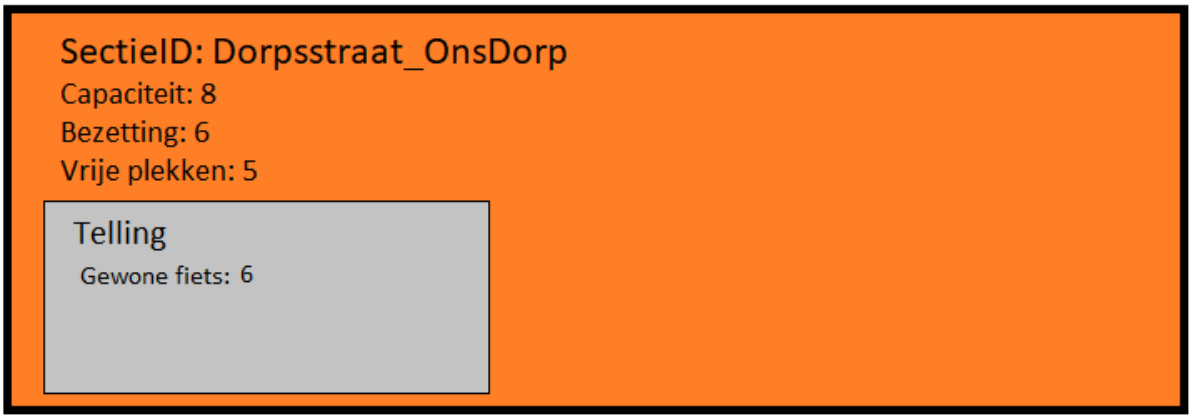
De dynamische data kan er visueel zo uit zien, als de tellers zeer gedetailleerd te werk gaan. Zo kan bijvoorbeeld zelfs worden aangegeven dat één van de fietsen in het rek een lekke band heeft.



Echter, in de praktijk zal een telling er misschien eerder zo uit zien:



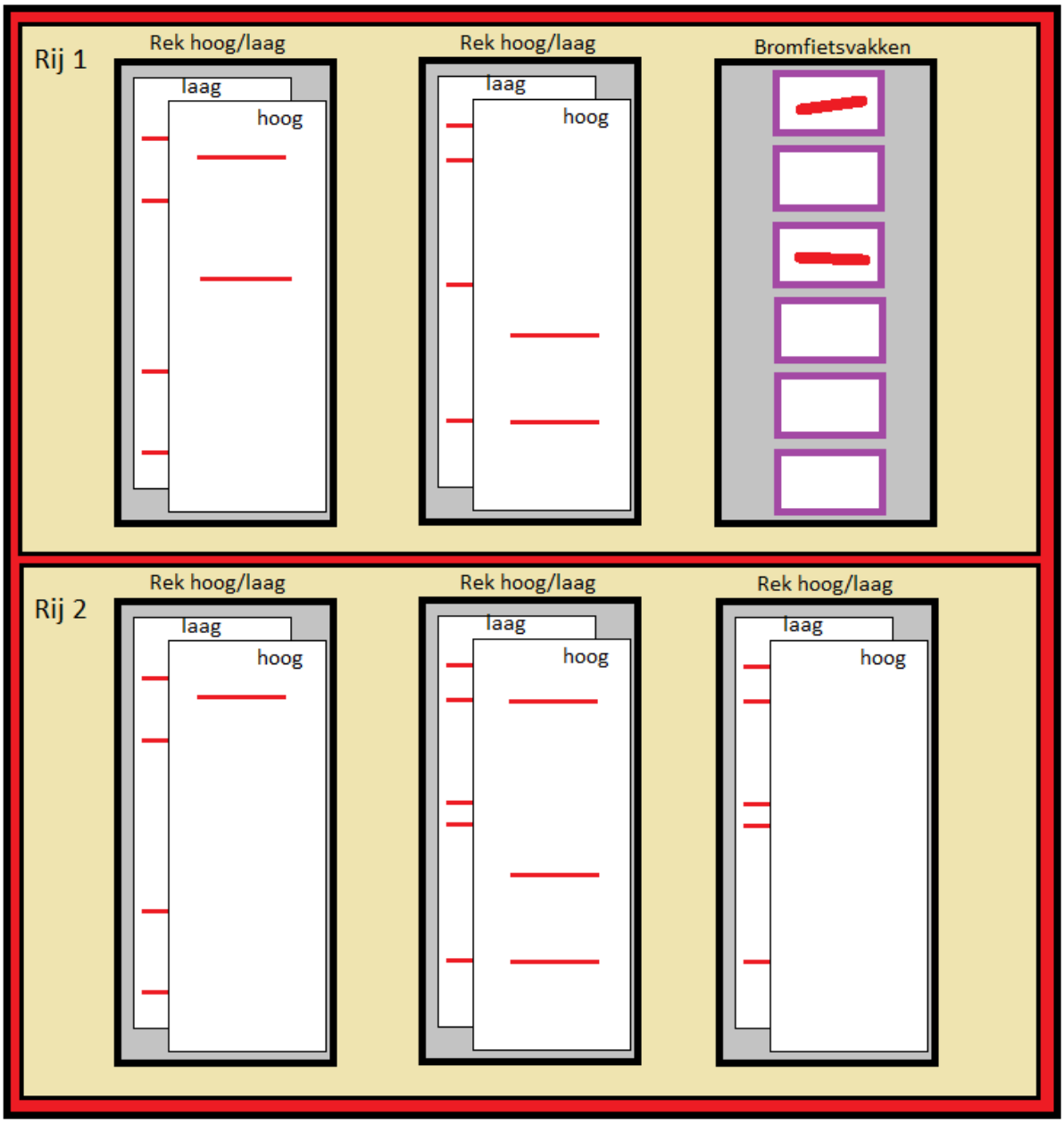
Of zelfs nog eenvoudiger, in geval van een weinig gedetailleerde telling:



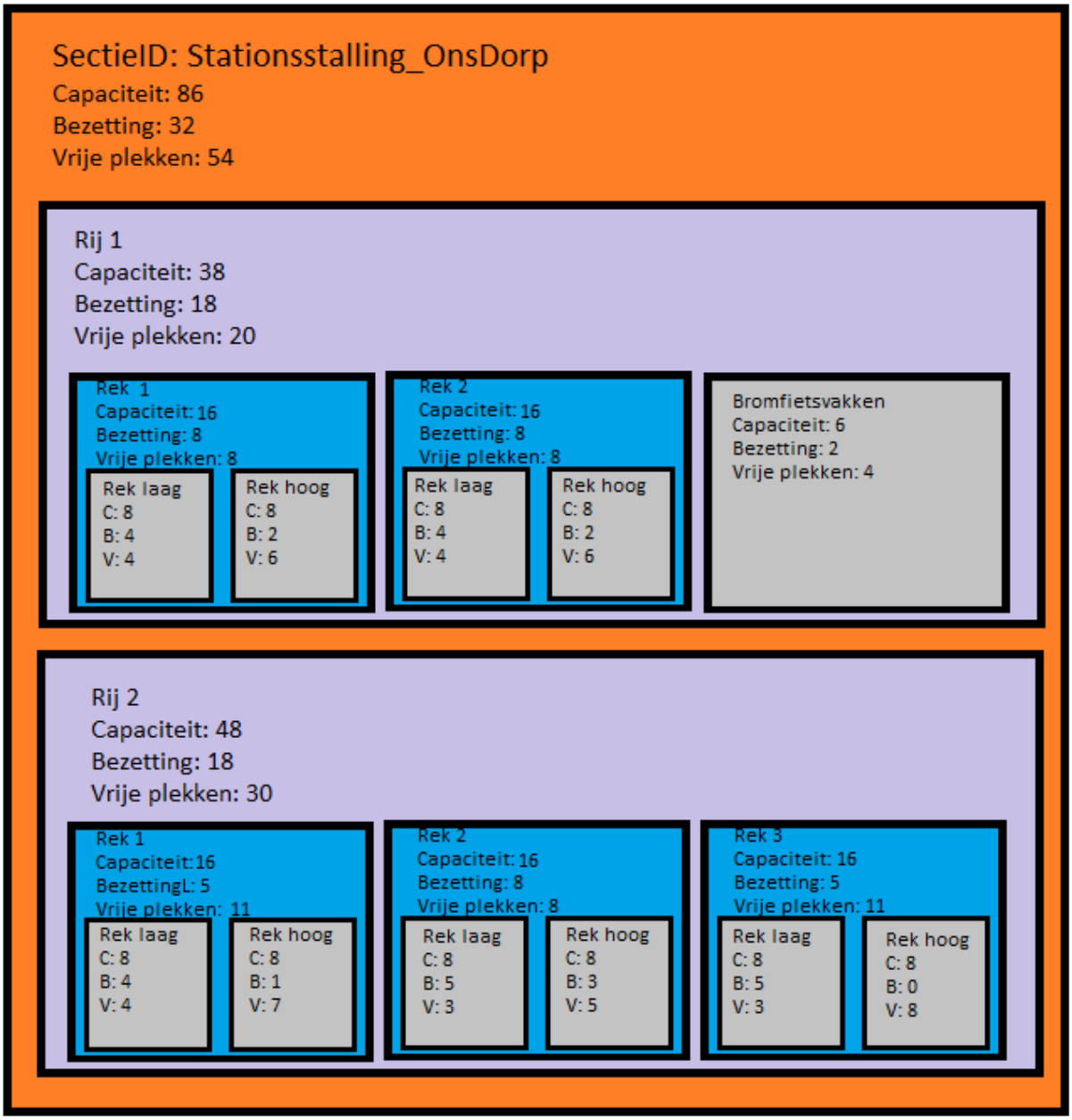
Voor bepaalde analysedoeleinden is het grijze blokje telling zelfs niet eens zo interessant. Met de juiste zoekopdrachten in de API kan een respons als deze volstaan, zelfs als de meest gedetailleerde telling is opgeslagen:

Voorbeeld 2: een doorsnee stationsstalling

Prorail beheert een groot aantal stationsstallingen in Nederland en laat deze automatisch tellen door sensoren in de rekken. Een typische Prorailstalling is opgedeeld in rijen. Iedere rij heeft een aantal voorzieningen met lage en hoge rekken. In dit voorbeeld heb ik er een sectie met bromfietsvlakken bijgetekend, iets wat Prorail-stallingen ongetwijfeld zullen hebben, maar waarop vooralsnog geen geautomatiseerde tellingen op plaatsvinden.



Dit ziet er in de datastandaard zo uit:



Data opvragen

Je kunt talloze manieren bedenken waarop een analist de data wil doorzoeken. De vraag daarbij is in hoeverre deze functionaliteit de dataportal moet worden ondersteund. De analist kan na opvraag van de complete dataset uiteraard ook zelf de data uitpluizen.

Als er in de dataportal metingen tot op het grootste detailniveau zijn opgeslagen, zoals bovenstaande voorbeeld, dan kunnen datasets door de tijd heen en enorm groot worden. Dat is zeker het geval als het gaat om automatische tellingen die meerdere keren per uur worden ingestuurd, Enige filtermogelijkheden in de dataportal is dan geen slechte benadering.

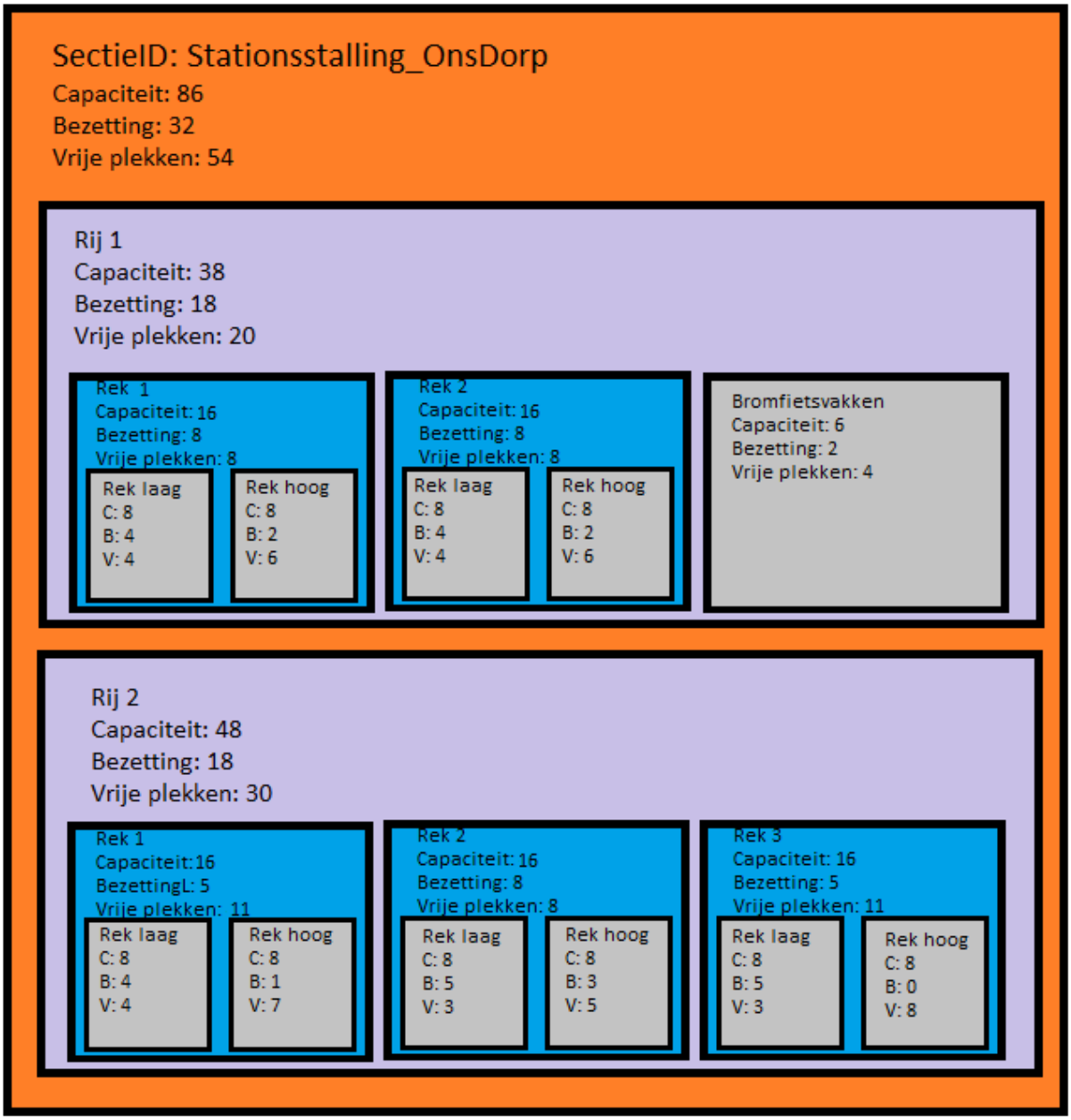
Daarom stelt de dataportal enkele zoekfunctionaliteiten verplicht voor de dataportals.

Filter op diepte

Gedetailleerde informatie over de bezetting geeft behoorlijk grote bomen, die niet voor iedere analist van deze data even interessant is. Als deze analist de waarnemingen van een stationsstalling door de tijd opvraagt, kunnen de responsen enorm groot worden. Als 90% van de data voor de analist niet relevant is, is dit natuurlijk onzinnig.

Daarom stelt de datastandaard een beperkt aantal zoekfunctionaliteiten verplicht. De belangrijkste van deze is dat je de mate van detail, de diepte (*depth*), bij een zoekopdracht kunt aangeven aan de API.

Stel dat in het dataportal de meest gedetailleerde boom is opgeslagen:



Als een analist alleen geïnteresseerd is in de bezetting van de totale stalling, vraagt hij data op op diepte 1. De API geeft hem dan dit resultaat:

Als een stalling iedere 2 minuten data opslaat en een analist vraagt de data op van een hele maand, scheelt dat uiteraard enorm in de datastroom en daarmee in de performance van de API.

Filter op inhoud

Om data-analisten te helpen hun weg te vinden in de gestaag groeiende data van een dataportal, stelt de datastandaard een aantal zoekfunctionaliteiten verplicht. Zo moet een er minimaal gefilterd kunnen worden op:

1. Data binnen een bepaalde tijdspanne
2. Data van een bepaalde sectie
3. Data van een bepaald onderzoek
4. Data van een bepaalde opdrachtgever
5. Data van een bepaalde dataleverancier
6. Statische data binnen een bepaalde geo-polygoon of geo-punt + radius

De praktijk moet uitwijzen of deze lijst voldoende is om aan alle wensen van de data-analisten te voldoen. Indien nodig zullen er meer zoekfuncties aan deze lijst worden toegevoegd.

Filter op tijdspanne

Gebruik bij het zoeken de url-parameters startDate en endData. Tijdstippen worden altijd doorgegeven in ISO8601 timestamp formaat.

Dus:

<url_api_dataportal>?startDate=2020-01-01T0:00:00&endDate=2020-02-01T0:00:00

Filter op sectie

Als een analist alleen data wil van een bepaald fietstype, bijvoorbeeld gewone fietsen:

<url_api_dataportal>?sectionID=<sectionID>

Of meerdere secties:

<url_api_dataportal>?sectionID=<sectionID1>,<sectionID2>,<sectionID3>

Filter op onderzoek

Data die ingestuurd wordt door een dataleverancier kan worden voorzien van een onderzoeksID (surveyID). Als deze ID gegeven is, is het uiteraard mogelijk hierop te filteren:

<url_api_dataportal>?surveyID=TellingGemeenteAmsterdam2020

Filter op opdrachtgever

Hetzelfde idee al filteren op onderzoek: als de data voorzien is van de ID van een opdrachtgever, kan hierop gefilterd worden:

<url_api_dataportal>?clientID=TellingGemeenteAmsterdam

Filter op dataleverancier

De dataportal kan de data die wordt ingestuurd voorzien van de ID van de leverancier van deze data. Als dat gebeurt, kan er uiteraard gezocht worden op dit ID:

<url_api_dataportal>?dataProviderID=DeFietstellersBV

Filter op geo-data

Als de statische data van secties is voorzien van exacte geo-coördinaten, moeten deze secties gevonden kunnen worden aan de van een geo-zoekopdracht.

Zoek op een punt + straal van dit punt in meters:

<url_api_dataportal>?geopoint=52.370216,4.895168&r=1000

Zoeken naar de sectie binnen een polygoon

<url_api_dataportal>?geopolygon=52.370216,4.895168, 53.370216,4.895168, 53.370216,5.895168, 52.370216,4.895168