

Bijlage 22B - Versie 26 12-12-2024

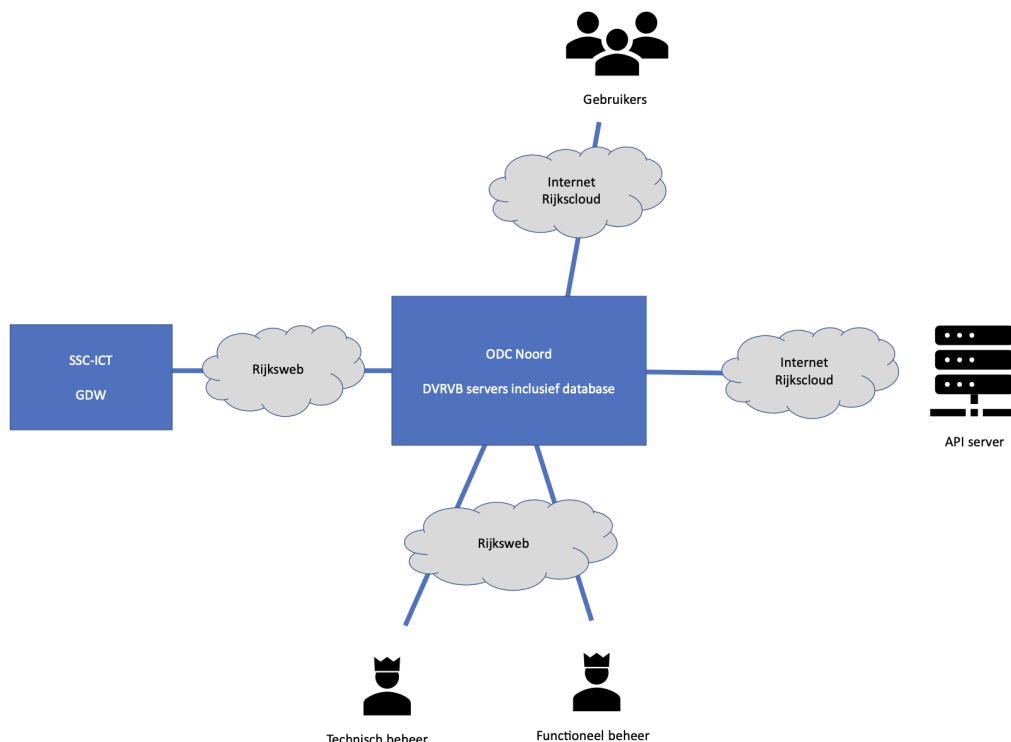
DVRVB API

- Beschrijving van DVRVB
 - Globale functionaliteit
- Aansluiten API accounts
 - Verkrijgen API account
 - Testen API account
- Gegevens
 - JSON
 - Uploaden Gegevens
 - Testen van de upload
 - Wijzigen en verwijderen van gegevens
- Locaties
 - Uploaden
 - POST versus PUT
 - Downloaden
- Documenten
 - Aanmaken nieuw document
 - JSON en Base64
 - Bestandstypen en SVG
 - Wijzigen bestaand document
- Aanmaken van relaties
 - JSON
 - API
- Raadpleeg API endpoints
 - Complexen
 - Complex
 - Gegevens
 - Documenten

Beschrijving van DVRVB

Globale functionaliteit

Het volgend geeft een overzicht van de omgeving van DVRVB



Hierover valt het volgende op te merken

1. Gebruikers komen bij de applicatie via de Rijkscloud. Deze is via het internet benaderbaar. Er wordt onderscheid gemaakt tussen:
 - a. Gewone gebruikers. Deze hebben alleen kijkrechten;
 - b. Gegevensbeheerders. Door gebruikers aan de juiste autorisatiegroepen te koppelen kunnen gebruikers rechten krijgen om gegevens te muteren;
 - c. Gebruikersbeheerders. Deze kunnen gebruikers aanmaken (in Keycloak) en gebruikers koppelen aan autorisatiegroepen.
2. DVRVB ondersteunt koppeling met systemen binnen en buiten de Rijksoverheid via een API;
3. RVB gegevens worden door DVRVB dagelijks opgevraagd bij interne systemen.

In dit document wordt de API koppeling beschreven.

Aansluiten API accounts

DVRVB is bedoeld als systeem waarbij externe partijen binnen en buiten de Rijksoverheid vastgoedgegevens kunnen uploaden. Dat uploaden kan via de front-end handmatig worden gedaan. Gegevens kunnen ook via een systeemkoppeling worden geüpload via een REST-API. Hiervoor is een API account nodig.

Verkrijgen API account

Het aansluiten van een API account is een samenspel van meerdere afdelingen binnen het RVB. Aan RVB-zijde is RVI (Regie Vastgoed Informatie) het single point of contact. RVI betreft zo nodig andere afdelingen van het RVB.

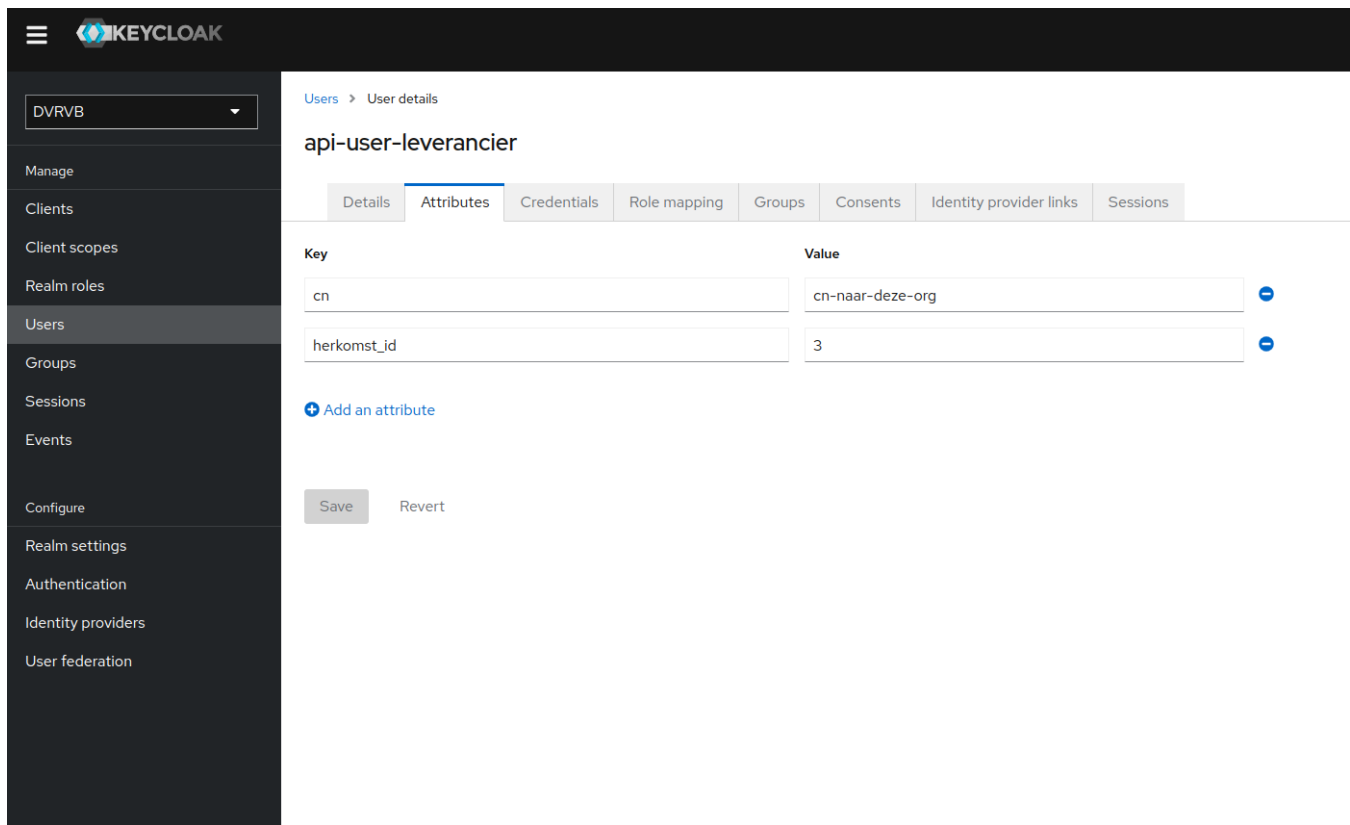
De API is beveiligd met een combinatie van API gebruiker in keycloak en een IP allow-list. De allow-list wordt door DVRVB beheerd. De te zetten stappen zijn:

Stap 1: Firewall / Proxy IP filter laten aanpassen.

Stap 2: Aanmaken API account in Keycloak:

The screenshot shows the Keycloak user management interface. On the left is a dark sidebar with a menu. The top of the sidebar has a hamburger menu icon and the 'KEYCLOAK' logo. Below this is a dropdown menu currently showing 'DVRVB'. The menu items are: Manage, Clients, Client scopes, Realm roles, Users (highlighted), Groups, Sessions, Events, Configure, Realm settings, Authentication, Identity providers, and User federation. The main content area has a breadcrumb 'Users > User details' and the title 'api-user-leverancier'. Below the title are tabs: Details (active), Attributes, Credentials, Role mapping, Groups, Consents, Identity provider links, and Sessions. The 'Details' tab contains the following fields: ID (8cf7fd27-066c-47b7-acf9-9e067e87595b), Created at (9/30/2024, 5:44:14 PM), Required user actions (a dropdown menu with 'Select action' and a help icon), Username (api-user-leverancier), Email (api@leverancier.com), Email verified (a toggle switch set to 'Yes'), First name (api-user), and Last name (leverancier). At the bottom of the form are two buttons: 'Save' and 'Revert'.

1. Voer onder het menu **Users** in de **hoofdrealm van DVRVB** een nieuwe user op (Dus niet onder de subrealm voor een leverancier)
2. Voer een wachtwoord in onder het tabblad van Credentials
3. En geef de attributen van de CommonName (CN) en herkomst_id (Het ID uit DVRVB van de gewenste herkomst) op



Na het aanmaken van een API account in Keycloak zal de DVRVB applicatie, **na het eerste request van dit systeem**, automatisch een API-gebruiker aanmaken, met de opgegeven herkomst (CN wordt nog niet gebruikt, maar is al wel verplicht).

De API URL's zijn:

Acceptatie: <https://dvrvb-api-acc.rvb.rijksweb.nl/>
\$API_URL = <https://dvrvb-api-acc.rvb.rijksweb.nl/api/1>

Productie: <https://dvrvb-api.rvb.rijksweb.nl/>
\$API_URL = <https://dvrvb-api.rvb.rijksweb.nl/api/1>

Testen API account

Testen van de werking van de API kan via een Linux command line als volgt worden uitgevoerd. Met een willekeurig token als voorbeeld:

```
# Login
curl -i -X POST \
  -H "Content-Type:application/x-www-form-urlencoded" \
  -d "client_id=dvrvb-systemapi" \
  -d "username=api-user-leverancier" \
  -d "password=een-secret-voor-de-specifieke-afnemer" \
  -d "grant_type=password" \
  -d "client_secret=z0ZFwiFO0gDX6W395JklzrhB6xzBsEbw" \
  'https://login.rvb.rijkscloud.nl/auth/realms/DVRVB/protocol/openid-connect/token'

export ACCESS_TOKEN="...JWT-uit-response..."

# Testen of de verbinding en token goed zijn ingesteld:
curl -H "Authorization:Bearer $ACCESS_TOKEN" "https://dvrvb-api.rvb.rijkscloud.nl/api/1/complexen"
```

Dit levert een JSON bestand met de geautoriseerde complexen.

Gegevens

JSON

Alle gegevens worden in JSON formaat met de DVRVB server uitgewisseld. De enige uitzondering hierop is het opvragen van documenten. Een document wordt als binary data verstrekt. Een upload van een document gaat echter wel via een JSON. Dit wordt verderop gedocumenteerd.

Via de front-end kunnen JSON's worde geüpload van Locaties en Gegevens. Dezelfde JSON's kunnen ook via de API worden gebruikt. Deze opzet is mede gekozen om de ontwikkeling van de API koppeling te vergemakkelijken.

Uploaden Gegevens

Gegevens kunnen in JSON formaat bij DVRVB worden geüpload. Een voorbeeld JSON bestand is:

```
[
{
  "complexuvid": "WTH-KRB-Y8VT-5XE", "name": "100570G01_490120_8-8",
  "classificatie": 100, "categorie": "Onderhoud",
  "omschrijving": "Bu. dekkend oplosmiddelhoudend", "data1": "49-12",
  "data2": "",
  "data3": "8",
  "data4": "2010-06-23 13:50:48",
  "toelichting": "Kozijnen; Tuinentree"
},
{
  "complexuvid": "WTH-KRB-Y8VT-5XE", "name": "100570G01_490120_10-10",
  "classificatie": 100, "categorie": "Onderhoud",
  "omschrijving": "Bu. dekkend oplosmiddelhoudend", "data1": "49-12",
  "data2": "",
  "data3": "10",
  "data4": "2010-06-23 13:50:48",
  "toelichting": "Gestukte voorgevel vzw dekkend verfsysteem. (overige gestukte
gevels zijn niet afgewerkt met een verfsysteem)"
}
]
```

De velden complexuvid, name en classificatie (geel gemarkeerd) zijn verplicht. Een JSON record zonder deze velden worden geweigerd. Het complexuvid en name is nodig om het record uniek te identificeren. De velden data1 tot en met data8 zijn optioneel maar hebben een bijzondere betekenis: deze velden worden in de front-end getoond in de kolommen Tekst 1 -8 in het overzichtstabel. Alleen op deze kolommen en de name (name) kan worden gefilterd.

Voor elk record is het noodzakelijk om het vertrouwelijkheidsniveau aan te geven. In de REST-API is de classificatie een getal met de volgende waarden:

- Openbaar = 0. Dit is publieke informatie (bijvoorbeeld BAG pand-id)
- Intern Algemeen = 100. Iedereen die bij dit complex kan mag deze informatie zien.
- Intern Vertrouwelijk = 200. Deze informatie mag alleen worden verstrekt aan medewerkers die dit voor hun werk nodig hebben.
- Departementaal Vertrouwelijk = 300. Dit is als Dep. V. gerubriceerde bijzondere informatie conform het VIR-BI.

Testen van de upload

Nadat is vastgesteld dat het API account werkt (zie paragraaf 4.1.1) kan een testbestand worden geüpload. Vanaf de Linux command line kan als test een testdata.json bestand worden geüpload via:

```
# Commando om gegevens in testdata.json te uploaden:
curl https://dvrvb-api.rvb.rijkscloud.nl/api/1/eenheden -H 'Content-Type: application/json' -H "Authorization: Bearer $ACCESS_TOKEN" -X POST -d @testdata.json
```

Mits succesvol levert dit een response met HTTP statuscode 204 op.

Wijzigen en verwijderen van gegevens

Het wijzigen van eerder aangeleverde gegevens werkt met dezelfde REST call. De upload is overschrijvend. Er bestaat geen PUT method. Deze wordt door een POST ingevuld.

Het verwijderen van gegevens gaat het beste via de Front-end. Selecteer de te verwijderen gegevens en kies de optie "verwijderen"

Locaties

Uploaden

Data uitwisseling van locaties gaat via het GeoJSON formaat. Via de front-end is het mogelijk om locaties te selecteren en deze selectie te exporteren. Dat levert een GeoJSON bestand dat kan worden geopend met een GIS tool zoals QGIS. Na bewerking in QGIS kan het resulterende GeoJSON bestand worden geupload.

Een voorbeeld GeoJSON bestand van de Poelendaelesingel 18 Middelburg is

```
{
  "name": "dv_ZM3-E2Q-YZRB-YYF_TR",
  "type": "FeatureCollection", "features": [{
    "type": "Feature", "geometry": {
      "type": "MultiPolygon",
      "coordinates": [[[[[3.6114097508540195, 51.49382353161147],
[3.611774667975467, 51.49400624846896], [3.6124209928975057, 51.494329871036456],
[3.612276806626252, 51.494442082823156], [3.6106998717799765, 51.49365247569153],
[3.6108440593911997, 51.49354026585286], [3.6114097508540195, 51.49382353161147]]]]],
    },
    "properties": { "classificatie": 100,
      "name": "B96-7Z0-8KGP-K71",
      "type": "Bouwwerk",
      "subtype": "", "peilmaat": 0,
      "hoogte": 40000
    }
  }
]
```

De groen gemarkeerde "name" geeft aan om welke bouwlaag van welk complex het gaat. De opmaak is

```
dv_<complexuvid>_<bouwlaag>
```

Hierin is bouwlaag een getal of "TR" voor het terrein.

Bij de API is dat de enige manier om die informatie te delen. Er is voor gekozen om de upload via de front-end en de API identiek te houden. Als gevolg daarvan is de groen gemarkeerde "name" een verplicht veld.

Voor elk feature is het verplicht om een (unieke) "name" aan te geven en de classificatie van het betreffende feature. Deze items zijn in het voorbeeld geel gemarkeerd.

De geometrie moet in WGS84 coördinaten worden opgegeven. DVRVB voert geen coördinaten conversie uit. Let hierop bij het opslaan vanuit een GIS applicatie.

POST versus PUT

Het API endpoint voor het uploaden van locaties is /api/1/dried. Er zijn twee mogelijkheden:

1. Met POST worden eerst de huidige locaties op de bouwlaag gemarkeerd als verwijderd. Vervolgens worden de gegevens ingeladen. Eventueel aangebrachte relaties worden opnieuw aangebracht.
2. Voor het behouden van de huidige locaties op de aangegeven bouwlaag: gebruik PUT

Om een POST (huidige bouwlaag van mijn herkomst verwijderen) te testen:

```
curl $API_URL/dried -H 'Content-Type: application/json' -H "Authorization:Bearer $ACCESS_TOKEN" -X POST -d @muntstraat7.json
```

Voor de PUT (items op bouwlaag behouden):

```
curl $API_URL/dried -H 'Content-Type: application/json' -H "Authorization:Bearer $ACCESS_TOKEN" -X PUT -d @muntstraat7.json
```

Dit onderscheid tussen POST en PUT wordt alleen voor locaties gemaakt.

Downloaden

De tabel van locaties kan met het algemene complex endpoint van paragraaf 7.2 worden opgevraagd. Daarnaast is er de mogelijkheid om locaties per bouwlaag als GeoJSON te downloaden.

Het API endpoint is /verdieping/<complexuvid><bouwlaag>. Voor het terrein dient bouwlaag gelijk te zijn aan 'TR'. Bijvoorbeeld:

```
curl $API_URL/verdieping/WTH-KRB-Y8VT-5XE/TR -H 'Content-Type: application/json' -H "Authorization:Bearer $ACCESS_TOKEN"
```

Gebruik het bouwlaagnummer voor de interne opbouw van de bouwwerken van een complex. Voor bijvoorbeeld bouwlaag 0:

```
curl $API_URL/verdieping/WTH-KRB-Y8VT-5XE/0 -H 'Content-Type: application/json' -H "Authorization:Bearer $ACCESS_TOKEN"
```

Het bestand is een GeoJSON die in een GIS applicatie als QGIS kan worden bekeken. De opmaak van het gedownloadte is zodanig dat deze ook weer kan worden geüpload.

Het is niet mogelijk om GeoJSON's per bouwwerk te downloaden. Dit gaat altijd per complex.

Documenten

Aanmaken nieuw document

JSON en Base64

Het volgende is voorbeeld van een werkend JSON bestand:

```
{
  "name": "rood1x1.png", "classificatie": "100", "content":
  "iVBORw0KGgoAAAANSUHEUgAAAAEAAAABCAIAAACQd1PeAAAAAXNSR0I
  Ars4c6QAAARnQU1BAACxjwv8YQUAAAJcEhZcwAAGdYAAbnWARjRyu0 AAAAMSURVBhXY3growIAAycBLhVrvukAAAAASUVORK5CYII=",
  "data1": "Demo bestand"
}
```

De geel gemarkeerde velden zijn verplicht. In dit voorbeeld is alleen het veld "data1" optioneel. Het documenttype wordt afgeleid uit de content
Bij het uploaden moet de complexcode worden opgegeven in de URL
/document/<complexuvid>. Bijvoorbeeld:

```
curl $API_URL/document/WTH-KRB-Y8VT-5XE -H 'Content-Type: application/json' -H "Authorization:Bearer $ACCESS_TOKEN" -X POST -d @testdoc.json
```

De content in dit JSON bericht is de base64 encoding van het te uploaden bestand. Om dit na te gaan: download het resultaat van de bovenstaande JSON upload. Dat is een 1x1 PNG bestand van een rode pixel met de naam rood1x1.png. Het linux / bash commando

base64 rood1x1.png

geeft:

```
iVBORw0KGgoAAAANSUHEUgAAAAEAAAABCAIAAACQd1PeAAAAAXNSR0IArs4c6QAAARnQU1BAACx
jwv8YQUAAAJcEhZcwAAGdYAAbnWARjRyu0AAAAMSURVBhXY3growIAAycBLhVrvukAAAAASUVO RK5CYII=
```

Dit komt overeen met het content veld van de JSON.

Bestandstypen en SVG

DVRVB ondersteunt alleen het uploaden van PDF documenten en afbeeldingen (Mime type image/*). Het bestandstype wordt afgeleid uit de content door libmagic (Linux).

SVG documenten, mits herkend als zodanig, worden omgezet naar PNG type. Dit is noodzakelijk omdat in een SVG bestand een Javascript script kan worden opgenomen. Dit zou gebruikt kunnen worden om de beveiliging van DVRVB te doorbreken.

Wijzigen bestaand document

Om de inhoud, classificatie, of metadata van een bestaand document te wijzigen kan de PATCH methode worden gebruikt. Hierbij kan dezelfde payload worden meegegeven als aan bovenstaande POST methode. De unieke identifier is hierbij het 'name' attribuut. Deze dient dus gelijk te zijn aan het bestaande document.

```
curl $API_URL/document/WTH-KRB-Y8VT-5XE -H 'Content-Type: application/json' -H "Authorization:Bearer $ACCESS_TOKEN" -X PATCH -d @testdoc.json
```

Aanmaken van relaties

JSON

Via de API kunnen relaties worden opgegeven. Zoals alle andere functies gaat dit ook via een JSON. Bijvoorbeeld:


```
[{
  "item": ["object", "WTH-KRB-Y8VT-5XE", "6QC-AC7-VV7E-9GA"],
  "eenheid": [{"CVTE", "Gebruik", "R0-01"}],
  "document": [{"RVB", "KRV", "rood1x1.png"}],
  "object": [{"RVB", "KRV", "K2A-Y5P-42E5-048"}]
}]
```

In dit voorbeeld worden drie relaties aangemaakt. Alle relatie worden gemaakt vanuit de locatie (object) met naam 6QC-AC7-VV7E-9GA. Deze locatie moet eerder zijn aangeleverd vanuit de bron van de uploader. Het is alleen mogelijk om "eigen" items te relateren aan ander items. De naam in combinatie met het complexuid (WTH-KRB-Y8VT-5XE in dit voorbeeld) identificeert dan het item uniek.

De relatie kan worden gelegd naar

1. Locaties. Gebruik hiervoor "object"
2. Gegevens. Gebruik hiervoor "eenheid"
3. Documenten. Gebruik hiervoor "document"

Bij alle drie doelen is de complexuid al bekend. Het is niet mogelijk om relaties over complexen heen te leggen; de complexuid van de oorsprong is ook die van het doel.

Aangezien een object uniek geïdentificeerd wordt door het viertal

- a. Complex(uuid)
- b. Organisatie
- c. Bron binnen deze organisatie
- d. Naam

Is het nodig om de velden b t/m d mee te geven in het JSON. Zie het voorbeeld waar bijvoorbeeld een relatie wordt aangemaakt met het gegeven (eenheid) van CVTE, bron Gebruik en naam R0-01.

API

Voor relaties is er één endpoint: /api/1/relaties Om te testen:

```
curl $API_URL/relaties -H 'Content-Type: application/json' -H "Authorization:Bearer $ACCESS_TOKEN" -X POST -d @koppel.json
```

Raadpleeg API endpoints

De REST-API wordt gebruikt voor het uploaden van gegevens maar ook door de front-end. De ontwerpkeuze is om dezelfde API te gebruiken voor beide doeleinden. Mocht dit op een gegeven moment beperkingen opleveren dan moet dit via Functioneel Beheer DVRVB kenbaar worden gemaakt. Voor de volledigheid worden hieronder alle ondersteunde endpoints van de API genoemd. De front-end nog meer endpoints maar dat betreft op gebruikers gerichte functionaliteit als het opvragen en opslaan van filters, gebruikersbeheer en dergelijke. Deze functies worden benadrukt via de prefix /dvapi en worden niet ondersteund via de REST-API.

In het vervolg worden de raadpleegfuncties beschreven (HTTP GET method). Voor mutatiefuncties wordt verwezen naar de voorafgaande hoofdstukken.

Complexen

Via de API kan de lijst complex worden opgevraagd:

```
curl $API_URL/complexen -H 'Content-Type: application/json' -H "Authorization:Bearer $ACCESS_TOKEN"
```

Dit levert een JSON met gegevens over de complexen waarop lees-rechten zijn toegekend. Bijvoorbeeld:

```
{
  "complexen": [{
    "code": "3512M7",
    "uvid": "WTH-KRB-Y8VT-5XE", "name": "MUNTSTRAAT 7 UTRECHT",
    "adres": "Muntstraat 7", "postcode": "3512ET", "plaatsnaam": "Utrecht", "land": "Nederland",
    "complexfunctie": "Kantoorlocatie",
    "departement": "Ministerie van Binnenlandse Zaken en Koninkrijksrelaties", "hoofdgebruiker": "BZK DG Overheidsorganisatie",
    "favoriet": false, "favoriet_sinds": null
  }]
}
```

Complex

Van een specifiek complex kunnen de tabellen worden opgevraagd in een call:

```
curl $API_URL/complex/WTH-KRB-Y8VT-5XE -H 'Content-Type: application/json' -H "Authorization:Bearer $ACCESS_TOKEN"
```

Dat levert de drie tabellen van het complexen scherm samen met gegevens die nodig zijn om het complex goed te tonen in de front-end. Bijvoorbeeld:

```
{
  "presentatie": { "code": "3512M7",
    "uvid": "WTH-KRB-Y8VT-5XE", "naam": "MUNTSTRAAT 7 UTRECHT",
    "adres": "Muntstraat 7", "postcode": "3512ET", "plaatsnaam": "Utrecht",
    "wgs_longitude": 5.125358449568122,
    "wgs_latitude": 52.0927340371063,
    "zoom": 0.0005046301168487588,
    "min": -1,
    "max": 2,
    "last_modified": "2023-03-15T12:21:57.905926+00:00",
    "favoriet": false, "favoriet_sinds": null
  },
  "uploadpermissie": { "eenheid": 0,
    "document": 0,
    "object": 0, "relatie": false
  },
  "eenheden": [[7, 300, "RVB", "KRV", "WTH-KRB-Y8VT-5XE", [], [], [82], "", "", "", "", "", "", "", "", "2023-03-15T12:15:51.345397+00:00"]],
  "documenten": [[1, 100, "RVB", "Techniek", "DVRVB API.pdf", [], [], [], "", "", "", "", "", "", "", "", "2023-05-01T08:12:27.878699+00:00"]],
  "dried": [[82, 100, "RVB", "KRV", "6QC-AC7-VV7E-9GA", [], [], [], "Bouwwerk", "", null, 0.0, 2700.0, "2023-03-15T12:21:57.901808+00:00"]]
}
```

Bij de lijsten komen de kolommen van de front-end terug. Het eerste item per eenheid/document/dried (groen gemarkeerd) is de DVRVB database index. Deze index is nodig voor het vragen van meer gegevens over het betreffende item. In geel staan de relaties van het betreffende item. Indien aanwezig dan bestaat dit uit een lijst met indices.

Gegevens

Van een specifiek gegevensrecord wordt in de tabel een deel van de geüploade gegevens getoond. Het volledig record kan per stuk worden opgevraagd. Bijvoorbeeld

```
curl $API_URL/eenheid/WTH-KRB-Y8VT-5XE/7 -H 'Content-Type: application/json' -H "Authorization:Bearer $ACCESS_TOKEN"
```

Dit levert een JSON.

Documenten

Een specifiek document kan in binaire vorm worden opgevraagd. Bijvoorbeeld

```
curl $API_URL/document/WTH-KRB-Y8VT-5XE/1 -H 'Content-Type: application/json' -H "Authorization:Bearer $ACCESS_TOKEN" --output file.pdf
```

Dit is overigens de enige endpoint die binaire gegevens levert. Alle andere endpoints werken met JSONs.