

Handleiding Open Water Data API van HDSR

- Voor ontwikkelaars –

Opgesteld door Roger de Crook & Robbert van de Ven (Adviseurs monitoring waterkwantiteit)
8-8-2017

Publiek dataportaal

De Open water Data API heeft 2 verschijningsvormen:

- 1) De GUI waarin meetlocaties, parameters en tijdreeksen visueel in kaart en grafiek raadpleegbaar zijn.

<http://hdsr.lizard.net>

- 2) De API waarmee meetlocaties, parameters en tijdreeksen als data opvraagbaar zijn.

<http://hdsr.lizard.net/api>

Databeschrijving

De Open Water Data API bevat tijdreeksen van oppervlakte waterkwantiteit, dwz: op meetlocaties waar waterhoogte (mNAP) en debiet (m³/s) wordt bepaald. Deze tijdreeksen kunnen een lange historie hebben. Indien de meetlocatie nog operationeel is, is de vertraging tussen de meting in het veld en de beschikbaarheid in de OWD-API maximaal een half uur. Het debiet kan gemeten zijn op een meetpunt of berekend zijn op basis van een debietformule bij een gemaal, stuw of inlaat.

De volgende data, met een tijdsinterval van 15 minuten, is beschikbaar in de API:

ID	Eigenschap	Type object	Inlog vereist?
Q.G.15	Debiet [m ³ /s]	Debietmeter en kunstwerk	Nee
DD.15	Draaiduur [s]	Pompvijzel	Ja
DDH.15	Draaiduur hoogtoeren [s]	Pompvijzel	Ja
DDL.15	Draaiduur laagtoeren [s]	Pompvijzel	Ja
Hh.15	Hefhoogte [mNAP]	Schuif	Ja
Hh2.15	Hefhoogte 2 [mNAP]	Schuif	Ja
IB.15	In bedrijf [-] (aan/uit)	Pompvijzel	Ja
IBH.15	In bedrijf hoog toeren [-]	Pompvijzel	Ja
IBL.15	In bedrijf laag toeren [-]	Pompvijzel	Ja
Hk.15	Kruinhoogte [mNAP]	Stuw	Ja
POS.15	Openingspercentage schuif [%]	Schuif	Ja
POS2.15	Openingspercentage schuif 2 [%]	Schuif	Ja
F.15	Pompfrequentie [Hz]	Pompvijzel	Ja
SS.15	Stroomsnelheid [m/s]	Debietmeter	Ja
TT.15	Toerental [rpm]	Pompvijzel	Ja
H.G.15	Waterhoogte [mNAP]	Waterstandsmeter	Nee

Rechten

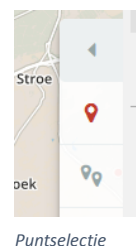
Zonder inlog kunnen de parameters debiet en waterhoogte geraadpleegd worden door een gebruiker. Indien een gebruiker beschikt over een inlogcode zijn de overige parameters ook beschikbaar. Een inlogcode is opvraagbaar bij Roger de Crook (Roger.de.crook@hdsr.nl) of Robbert van de Ven (Robbert.van.de.Ven@hdsr.nl).

Gebruik van de GUI

De HDSR-objecten in de grafische user interface (GUI) betreffen 3 soorten iconen: meetpunten (waterstandsmeters of debietmeters), gemalen (pompen of vijzels) en stuwen (inclusief inlaten). Ook meetobjecten van andere waterbeherende organisaties worden getoond in de GUI. Helaas kan er niet gefilterd worden op organisatie. De meetobjecten van HDSR zijn doorgaans herkenbaar als rode stippen (na inzoomen: rode kleine dichte druppelvorm). De parameters debiet in m³/s (*Q.G.15*) en waterhoogte in mNAP (*H.G.15*) kunnen weergegeven worden in de GUI.

Hieronder 3 tips ten aanzien van het selecteren van meetlocaties in bijzondere situaties:

- 1) Selecteer in het rechtergedeelte van het scherm Puntselectie:
- 2) Aangezien een stuw die naast een pomp staat momenteel nog dezelfde XY-coördinaten kan hebben, zie je soms iconen over elkaar heen liggen en dan is het moeilijk om de onderste meetlocatie te selecteren. Gelukkig beginnen deze over elkaar liggende objecten met dezelfde naam, dus door deze naam in het zoekvenster links bovenin te typen, kan dat object toch geselecteerd worden.
- 3) Op een beperkt aantal locaties zijn debietmeters aanwezig. Om deze objecten gemakkelijk te vinden en te kunnen selecteren kan eveneens het zoekvenster gebruikt worden door de term "debiet" of "ADM" in te voeren.
- 4) Soms zijn er meerdere pompen bij eenemaal en dan wordt een keuzemenu getoond om de meetlocatie van desbetreffende pomp te selecteren.



Nadat een icoon geselecteerd is, verschijnt de tijdsreeks ervan onder de beschrijving van de meetlocatie. Door de tijdbalk rechts onderin uit te zoomen verschijnt een langere periode in beeld.

In plaats van de kaart kan ook het dashboard weergegeven worden. Hier kunnen meerdere tijdreeksen boven elkaar getoond worden door een nieuwe zoekterm te gebruiken. Na selectie van een zoekresultaat wordt die tijdreeks toegevoegd. **Let op:** zodra je meerdere tijdreeksen in het dashboard hebt staan, zal de kaartweergave in de multi-select modus komen.

Als in de kaartweergave meerdere locaties geselecteerd zijn, zullen er geen tijdreeksen worden getoond onder de beschrijving van elke locatie. Dit blijft zo, totdat slechts één locatie in de selectie over is, bv. door andere te deselecteren (X) of door de single-select modus weer te activeren.

Gebruik van de API

De API kan gebruikt worden om data te downloaden. De basis url is <https://hdsr.lizard.net/api/v3/>. Aan de basis url kan een parameter worden toegevoegd, zodat bepaalde data gedownload kan worden. Om alle resultaten van een query te kunnen opslaan wordt aanbevolen achter elke query het volgende toe te voegen: "&format=json&page_size=9999".

Om locaties binnen HDSR op te vragen, dient altijd het volgende toegevoegd te worden aan de url: "&organisation_name=HDSR".

Om de locaties van gemalen, stuwen, inlaten en debietmeters op te halen kunnen in de browser verschillende queries uitgevoerd worden, waarbij gefilterd wordt op HDSR en de locatie-prefix "KW".

Voor de waterstandslocaties dient de locatie-prefix "OW" gebruikt te worden. Het is altijd mogelijk meerdere parameters of locaties op te vragen door een "&" tussen de url te plaatsen.

a) alle kunstwerklocaties =

https://hdsr.lizard.net/api/v3/locations/?organisation_code_startswith=KW&organisation_name=HDSR&format=json&page_size=9999

1. gemalen =

https://hdsr.lizard.net/api/v3/locations/?organisation_code_startswith=KW&name_icontains=pompvijzel&organisation_name=HDSR&format=json&page_size=9999

2. stuwen =

https://hdsr.lizard.net/api/v3/locations/?organisation_code_startswith=KW&name_icontains=stuw&organisation_name=HDSR&format=json&page_size=9999

3. inlaten =

https://hdsr.lizard.net/api/v3/locations/?organisation_code_startswith=KW&name_icontains=stuw&organisation_name=HDSR&format=json&page_size=9999

4. debietmeters =

https://hdsr.lizard.net/api/v3/locations/?code_startswith=KW&name_icontains=debiet_meting&organisation_name=HDSR&format=json&page_size=9999

5. waterstandlocaties =

https://hdsr.lizard.net/api/v3/locations/?organisation_code_startswith=OW&organisation_name=HDSR&format=json&page_size=9999

Tijdseries

Om een tijdreeks van een locatie op te vragen kan in de browser een query worden uitgevoerd zoals in onderstaand voorbeeld, waarbij de locatiecode (beginnend met "KW" of "OW") bekend moet zijn en de parameter opgegeven worden. De parameter voor kwartiergemiddelde waterhoogte in mNAP is "H.G.15" en voor kwartiergemiddeld debiet in m³/s is "Q.G.15". Het onderstaande is een voorbeeld voor het ophalen van de kwartiergemiddelde debiet van een bepaalde locatie.

https://hdsr.lizard.net/api/v3/timeseries/?name=Q.G.15&location_code=KW323711&location_organisation_name=HDSR&format=json&page_size=9999

De reactie van de API op de request, is een antwoord met daarin een nieuwe URL van de locatie. In deze URL bevat de tijdreeks gegevens van de locatie en opgegeven parameter. Om te zorgen dat de tijdreeks beperkt blijft tot een overzichtelijke hoeveelheid waarden, dient een start- en eindmoment te worden opgegeven in UNIX formaat. Dit is het aantal milliseconden sinds 1 januari 1970 om 1 uur 's nachts. Oudjaarsdag 2016: "&end=1483225200000&start=1483138800000". Om de datumtijd in het UNIX formaat te krijgen, volgt hieronder een voorbeeld in Python:

```
import datetime, time

old_dt = datetime.datetime.now() - datetime.timedelta(days=30)

time.mktime(old_dt.timetuple())*1e3 + old_dt.microsecond/1e3

new_dt = datetime.datetime(2016,11,19,11,58,57,0)

time.mktime(new_dt.timetuple())*1e3 + new_dt.microsecond/1e3
```

Het blok code op de volgende pagina is een voorbeeld voor het automatische opvragen van een tijdserie in python voor een opgegeven locatie en parameter.

```

import requests
import datetime, time

#Parameter, locationcode, start and end date of the location of interest
parameter = "Q.G.15"
#location = "OW439002"
location = "KW323711"
sd = 2017,1,1,12,0,0,0 #start date
ed = 2017,8,10,12,0,0,0 #end date

#url to retrieve the data from the location (kunstwerk) for the specified parameter
url = "https://hdr.lizard.net/api/v3/timeseries/?name=" + parameter + "&location__code=" + location +
"&location__organisation__name=HDSR&format=json&page_size=9999"

#convert the start and end date to UNIX code:
start_dt = datetime.datetime(sd[0],sd[1],sd[2],sd[3],sd[4],sd[5],sd[6])
strt = int(time.mktime(start_dt.timetuple())*1e3 + start_dt.microsecond/1e3)
end_dt = datetime.datetime(ed[0],ed[1],ed[2],ed[3],ed[4],ed[5],ed[6])
nd = int(time.mktime(end_dt.timetuple())*1e3 + end_dt.microsecond/1e3)

#Check if the url is valid. status_code 200 is a valid, working url.
response = requests.get(url)
if response.status_code != 200:
    print('Response status:', response.status_code, 'Problem with the request. The url could be invalid')

#Fetch url containing the timeseries data, from the specified location and parameter url.
data = response.json()
data_results = data["results"]
ts = data_results[0]["url"]

if response.status_code == 200:
    print('Name of the location:')
    print(str(data_results[0]["location"]["name"]))

#Insert the specified start and end date
response_series = requests.get((ts[: -11] + "start=" + str(strt) + "&end=" + str(nd) + "&format=json"))
data_series = response_series.json()

#Fetch the values and timestamps
value = [None] * (len(data_series["events"]))
timestamp = [None] * (len(data_series["events"]))

for i in range(0,(len(data_series["events"]),1):
    value[i] = (data_series["events"][i]["max"])
    timestamp[i] = (data_series["events"][i]["timestamp"])

#combine the list of timestamps and values
dataset = zip (timestamp, value)

```

Grote datasets

Als u veel data wenst te downloaden van een bepaalde parameter of locatietype, is het aan te raden om hier een asynchrone taak van te maken met de parameter "async=true". Als de data gedownload is, verandert de "task_status" van "PENDING" naar "SUCCESS" en kunt u de "result_url" volgen voor de data. Voorbeeld:

https://hdsr.lizard.net/api/v3/weirs/?async=true&format=json&page_size=100&page=1

Uit de aanvraag volgt een nieuwe URL:

<https://hdsr.lizard.net/api/v3/tasks/39ac2109-7fa3-4588-b572-3dd0e8eba74d/>

De nieuwe URL bevat de data en zal de status 'succes' bevatten, nadat deze is afgerond.

Ruimtelijk filteren

Het is mogelijk data op te vragen op basis van een ruimtelijke locatie binnen een "bounding box" of op basis van de afstand ten opzichte van een punt.

Voor een bounding box dient u het volgende aan de url toe te voegen "?in_bbox=min_lat,min_lon,max_lat,max_lon". Hierbij staan min_lat en max_lat voor de minimale en maximale latitude van de bounding box en min_lon en max_lon staan voor de minimale en maximale longitude van de bounding box. Voorbeeld:

https://hdsr.lizard.net/api/v3/locations/?organisation_code_startswith=KW&organisation_name=HDSR&format=json&page_size=9999&in_bbox=4.9895994542,51.9778979539,5.34117091607,52.100802442

Een andere manier voor ruimtelijke filtering is een punt aangeven met een bepaalde radius eromheen, met de parameters 'distance=x&point=lat,lon'. Hierbij is x het afstand vanaf het punt en staan lat en lon voor de latitude en longitude van het punt. Bijvoorbeeld:

https://hdsr.lizard.net/api/v3/pumpstations/?organisation_code_startswith=KW&organisation_name=HDSR&format=json&page_size=9999/?distance=400000&point=5.09539,52.03774

Filtering met Python

Binnen de API zijn er geen mogelijkheden om direct bij de aanvraag de gegevens te filteren op eigenschappen van de objecten, of alleen een deel van de eigenschappen van de data te downloaden. Het 'filteren' van de gegevens gebeurt eigenlijk op basis van de request naar de API server. Indien hierin alle gemalen, waterstanden, stuwen etc. wordt opgegeven wordt deze data ook aangeleverd. Om de gegevens na het downloaden te filteren is een voorbeeld uitgewerkt op de volgende pagina met Python:

- Vraag een lijst van alle locaties op met een waterstandmeter.

Contact

Bij vragen en/of opmerkingen kunt u contact opnemen met Roger de Crook (roger.de.crook@hdsr.nl) of Robbert van de Ven (Robbert.van.de.Ven@hdsr.nl).

```

# Import necessary libraries
import requests

# All pumps (gemalen):
url =
'https://hdsr.lizard.net/api/v3/locations/?organisation_code__startswith=KW&name__icontains=pompvijzel&organisation__name=HDSR&format=json&page_size=9999'

# All waterlevel measurement points
#url =
'https://hdsr.lizard.net/api/v3/locations/?organisation_code__startswith=OW&organisation_name=HDSR&format=json&page_size=9999'

#Check if the url is valid. status_code 200 is a valid, working url.
response = requests.get(url)
if response.status_code != 200:
    print('Response status:', response.status_code, 'Problem with the request. The url could be invalid')

data = response.json()
data_results = data["results"] # lijst met resultaten

location_id = []
location_name = []
location_x = []
location_y = []

for i in range(0,(len(data_results)),1):
    if data_results[i]['id']== None:
        location_id.append(0)
    else:
        location_id.append(data_results[i]['code'])
    if data_results[i]['name']== None:
        location_name.append(0)
    else:
        location_name.append(data_results[i]['name'])
    if data_results[i]['geometry']== None:
        location_x.append(0)
    else:
        location_x.append(data_results[i]['geometry']['coordinates'][0])
    if data_results[i]['geometry']== None:
        location_y.append(0)
    else:
        location_y.append(data_results[i]['geometry']['coordinates'][1])

#combine the lists
dataset = zip (location_id, location_name, location_x, location_y)

```