

testRWS_api

Deltares

October 30, 2018

Opvragen waterkwaliteitsdata door middel van RWS DDL API

Ik gebruik eigen R functies. Als package op github: wstolte/rwsapi. Dit package is nog in ontwikkeling.

```
if(!require("rwsapi", character.only = TRUE)){  
  devtools::install_github("wstolte/rwsapi")  
  library("rwsapi", character.only = TRUE)  
}  
require(rwsapi)
```

Aantal parameters

Opvragen metadata catalogus:

```
metadata <- rwsapi::rws_metadata()
```

Dit zijn metadata van alle datasets in de DDL. Bijvoorbeeld, Nitraat, of Chlorophyll zit er in

```
metadata$content$AquoMetadataLijst$Parameter_Wat_Omschrijving[  
  grep(x = metadata$content$AquoMetadataLijst$Parameter_Wat_Omschrijving, pattern = "nitraat")]
```

```
## [1] "(massa)Concentratie nitraat in Oppervlaktewater uitgedrukt in stikstof / opgeloste fractie in mg/l"  
## [2] "(massa)Concentratie som nitraat en nitriet in Oppervlaktewater uitgedrukt in stikstof / opgeloste fractie in mg/l"
```

```
metadata$content$AquoMetadataLijst$Parameter_Wat_Omschrijving[  
  grep(x = metadata$content$AquoMetadataLijst$Parameter_Wat_Omschrijving, pattern = "chlorofy")]
```

```
## [1] "(massa)Concentratie chlorofyl-a in Oppervlaktewater ug/l"
```

Maar Silikaat:

```
metadata$content$AquoMetadataLijst$Parameter_Wat_Omschrijving[  
  grep(x = metadata$content$AquoMetadataLijst$Parameter_Wat_Omschrijving, pattern = "sili")]
```

```
## character(0)
```

kan ik niet vinden.

Functie voor ophalen van data

R list structure wordt gebruikt als input in de functie hierboven.

```
# create list for request  
requestlist <- structure(list(AquoPlusWaarnemingMetadata = structure(list(AquoMetadata = structure(list(  
  Parameter = structure(list(Code = "CHLFA"), .Names = "Code"),  
  Grootheid = structure(list(Code = "CONCTTE"), .Names = "Code")), .Names = c("Parameter",  
"Grootheid"))), .Names = "AquoMetadata"), Locatie = structure(list(  
  X = "558558.371921023", Y = "5732781.94848472", Code = "SCHARDKDPPT"), .Names = c("X",  
"Y", "Code")), Periode = structure(list(Begindatumtijd = "1970-11-27T09:00:00.000+01:00",  
  Einddatumtijd = "2018-01-28T09:01:00.000+01:00"), .Names = c("Begindatumtijd",
```

```
"Einddatumtijd"))), .Names = c("AquoPlusWaarnemingMetadata",
"Locatie", "Periode"))
```

Als JSON ziet het er zo uit:

```
print(toJSON(requestlist, auto_unbox = T, digits = NA))
```

```
{ "AquoPlusWaarnemingMetadata": { "AquoMetadata": { "Parameter": { "Code": "CHLFa" }, "Grootheid": { "Code": "CONCTTE" } },
  "11-27T09:00:00.000+01:00", "Einddatumtijd": "2018-01-28T09:01:00.000+01:00" } }
```

Uitvoeren request. De response is in dit geval een r list object

```
response <- rws_observations(bodylist = requestlist)
```

Response omzetten naar dataframe

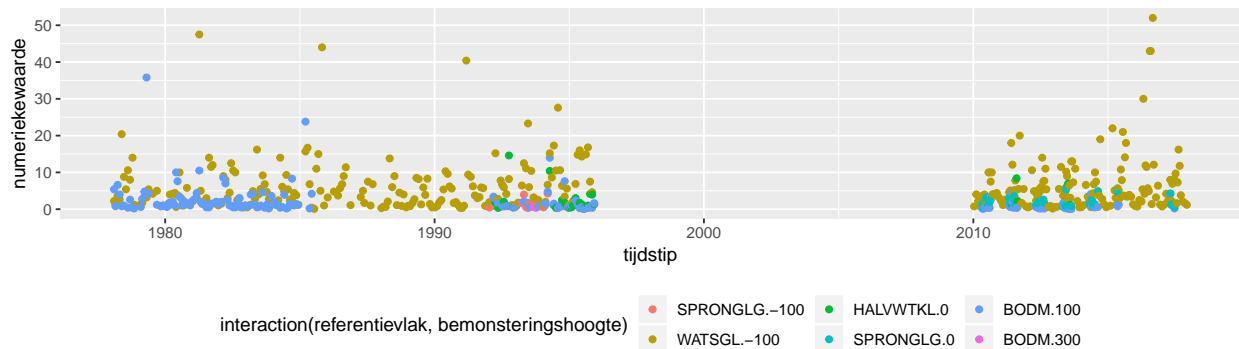
```
for(ii in seq(1:length(response$content$WaarnemingenLijst))) {
  temp.df = data.frame(
    locatie.code = response$content$WaarnemingenLijst[[ii]]$Locatie$Code,
    EPSG = response$content$WaarnemingenLijst[[ii]]$Locatie$Coordinatenstelsel,
    X = response$content$WaarnemingenLijst[[ii]]$Locatie$X,
    Y = response$content$WaarnemingenLijst[[ii]]$Locatie$Y,
    # locationname = ,
    tijdstip = lubridate::as_datetime(response$content$WaarnemingenLijst[[ii]]$MetingenLijst %>% map_chr(
    referentievlak = response$content$WaarnemingenLijst[[ii]]$MetingenLijst %>% map_chr(list(3,3,1),
    bemonsteringshoogte = response$content$WaarnemingenLijst[[ii]]$MetingenLijst %>% map_chr(list(3,2,
    kwaliteitswaardecode = response$content$WaarnemingenLijst[[ii]]$MetingenLijst %>% map_chr(list(3,
    bemonsteringsapparaat = response$content$WaarnemingenLijst[[ii]]$AquoMetadata$BemonsteringsApparaat$Code,
    bemonsteringsSoort = response$content$WaarnemingenLijst[[ii]]$AquoMetadata$BemonsteringsSoort$Code,
    biotaxoncode = response$content$WaarnemingenLijst[[ii]]$AquoMetadata$BioTaxon$Code,
    biotaxoncompartimentcode = response$content$WaarnemingenLijst[[ii]]$AquoMetadata$BioTaxon_Compartiment$Code,
    compartimentcode = response$content$WaarnemingenLijst[[ii]]$AquoMetadata$Compartiment$Code,
    eenheidcode = response$content$WaarnemingenLijst[[ii]]$AquoMetadata$Eenheid$Code,
    grootheidcode = response$content$WaarnemingenLijst[[ii]]$AquoMetadata$Grootheid$Code,
    hoedanigheidcode = response$content$WaarnemingenLijst[[ii]]$AquoMetadata$Hoedanigheid$Code,
    meetapparaatcode = response$content$WaarnemingenLijst[[ii]]$AquoMetadata$MeetApparaat$Code,
    monsterbewerkingsmethodecode = response$content$WaarnemingenLijst[[ii]]$AquoMetadata$MonsterBewerking$Code,
    orgaancode = response$content$WaarnemingenLijst[[ii]]$AquoMetadata$Orgaan$Code,
    parametercode = response$content$WaarnemingenLijst[[ii]]$AquoMetadata$Parameter$Code,
    plaatsbepalingsapparaatcode = response$content$WaarnemingenLijst[[ii]]$AquoMetadata$PlaatsBepaling$Code,
    typeringcode = response$content$WaarnemingenLijst[[ii]]$AquoMetadata$Typering$Code,
    waardebepalingsstechniekcode = response$content$WaarnemingenLijst[[ii]]$AquoMetadata$WaardeBepaling$Code,
    waardebepalingsmethodecode = response$content$WaarnemingenLijst[[ii]]$AquoMetadata$WaardeBepaling$Code,
    waardebewerkingsmethodecode = response$content$WaarnemingenLijst[[ii]]$AquoMetadata$WaardeBewerking$Code,
    numeriekewaarde = response$content$WaarnemingenLijst[[ii]]$MetingenLijst %>% map_dbl(list("Meetwaarde",
  )
  if(ii != 1){
    df = rbind(df, temp.df)
  }else df = temp.df
}
df[df$numeriekewaarde>1e10,] <- NA
```

```
df <- df[!is.na(df$numeriekewaarde),]
knitr::kable(head(df))
```

locatie.code	EPSG	X	Y	tijdstip	referentievlak	bemonstering
SCHARDKDPPT	25831	558558.371921023	5732781.9484847197	1991-01-08 23:00:00	WATSGL	-100
SCHARDKDPPT	25831	558558.371921023	5732781.9484847197	1991-02-03 23:00:00	WATSGL	-100
SCHARDKDPPT	25831	558558.371921023	5732781.9484847197	1991-03-06 23:00:00	WATSGL	-100
SCHARDKDPPT	25831	558558.371921023	5732781.9484847197	1991-04-01 22:00:00	WATSGL	-100
SCHARDKDPPT	25831	558558.371921023	5732781.9484847197	1991-05-06 22:00:00	WATSGL	-100
SCHARDKDPPT	25831	558558.371921023	5732781.9484847197	1991-06-03 22:00:00	WATSGL	-100

TijdsSerie

```
ggplot(df, aes(tijdstip, numeriekewaarde)) + geom_point(aes(color = interaction(referentievlak, bemonstering)))
theme(legend.position="bottom")
```



Dit ziet er goed uit. In Waterbase waren alleen oppervlaktewaarnemingen te zien, hier zijn ook diepere monsters meegenomen.

Aantal parameters

In de DDL webservice zijn

```
metadata <- rwsapi::rws_metadata()

grep(x = metadata$content$AquoMetadataLijst$Parameter_Wat_Omschrijving, pattern = "sili")

## integer(0)
```

Dit zijn metadata van alle datasets in de DDL. Bijvoorbeeld, Nitraat, of Chlorophyll zit er in

```
metadata$content$AquoMetadataLijst$Parameter_Wat_Omschrijving[
  grep(x = metadata$content$AquoMetadataLijst$Parameter_Wat_Omschrijving, pattern = "nitraat")]

## [1] "(massa)Concentratie nitraat in Oppervlaktewater uitgedrukt in stikstof / opgeloste fractie in mg/l"
## [2] "(massa)Concentratie som nitraat en nitriet in Oppervlaktewater uitgedrukt in stikstof / opgeloste fractie in mg/l"

metadata$content$AquoMetadataLijst$Parameter_Wat_Omschrijving[
  grep(x = metadata$content$AquoMetadataLijst$Parameter_Wat_Omschrijving, pattern = "chlorofy")]

## [1] "(massa)Concentratie chlorofyl-a in Oppervlaktewater ug/l"
```

Maar Silikaat:

```
metadata$content$AquoMetadataLijst$Parameter_Wat_Omschrijving[  
  grep(x = metadata$content$AquoMetadataLijst$Parameter_Wat_Omschrijving, pattern = "sili")]
```

```
## character(0)
```

kan ik niet vinden.