

Download RWS water data from data distribution layer

```
# devtools::install_github("wstolte/rwsapi", force = TRUE)
require(rwsapi)

# retrieve metadata
metadata <- rwsapi::rws_metadata()

# look what's there, e.g. nitrate
metadata$content$AquoMetadataLijst$Parameter_Wat_Omschrijving[
  grep(x = metadata$content$AquoMetadataLijst$Parameter_Wat_Omschrijving, pattern = "nitraat", ignore.case = T)]

locatiecodes <- metadata$content$LocatieLijst$Code
grootheidcodes <- metadata$content$AquoMetadataLijst$Grootheid$Code
parametercodes <- metadata$content$AquoMetadataLijst$Parameter$Code
omschrijvingcodes <- metadata$content$AquoMetadataLijst$Parameter_Wat_Omschrijving

myparameter = "NO3"
myquantity = "CONCTTE"

mylocation <- "SCHAARVODDL"
locindex <- which(metadata$content$LocatieLijst$Code == mylocation)
x = metadata$content$LocatieLijst$X[locindex]
y = metadata$content$LocatieLijst$Y[locindex]

# create list for request
requestlist <- structure(list(
  AquoPlusWaarnemingMetadata = structure(list(
    AquoMetadata = structure(list(
      Parameter = structure(list(
        Code = myparameter,
        .Names = "Code"),
      Grootheid = structure(list(
        Code = myquantity,
        .Names = "Code")),
      .Names = c("Parameter", "Grootheid"))),
    .Names = "AquoMetadata"),
  Locatie = structure(list(
    X = x,
    Y = y,
    Code = mylocation,
    .Names = c("X", "Y", "Code")),
  Periode = structure(list(
    Begindatumtijd = "2013-11-27T09:00:00.000+01:00",
    Einddatumtijd = "2019-10-01T09:01:00.000+01:00",
    .Names = c("Begindatumtijd", "Einddatumtijd")),
  .Names = c("AquoPlusWaarnemingMetadata", "Locatie", "Periode"))

# request data
response <- rwsapi::rws_observations(bodylist = requestlist)

# make dataframe from json response

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(list
(1), .default = NA)),
```

```

referentievlak = response$content$WaarnemingenLijst[[ii]]$MetingenLijst %>% map_chr(list(3,3,1), .default
= NA),
bemonsteringshoogte = response$content$WaarnemingenLijst[[ii]]$MetingenLijst %>% map_chr(list(3,2,1), .
default = NA),
kwaliteitswaardecode = response$content$WaarnemingenLijst[[ii]]$MetingenLijst %>% map_chr(list(3,5,1), .
default = NA),
bemonsteringsapparaat = response$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]]$AquoMetadata$MonsterBewerkingsMethode$Code,
orgaancode = response$content$WaarnemingenLijst[[ii]]$AquoMetadata$Orgaan$Code,
parametercode = response$content$WaarnemingenLijst[[ii]]$AquoMetadata$Parameter$Code,
plaatsbepalingsapparaatcode = response$content$WaarnemingenLijst
[[ii]]$AquoMetadata$PlaatsBepalingsApparaat$Code,
typeringcode = response$content$WaarnemingenLijst[[ii]]$AquoMetadata$Typering$Code,
waardebepalingstechniekcode = response$content$WaarnemingenLijst
[[ii]]$AquoMetadata$WaardeBepalingstechniek$Code,
waardebepalingsmethodecode = response$content$WaarnemingenLijst
[[ii]]$AquoMetadata$WaardeBepalingsmethode$Code,
waardebewerkingsmethodecode = response$content$WaarnemingenLijst
[[ii]]$AquoMetadata$WaardeBewerkingsmethode$Code,
numeriekewaarde = response$content$WaarnemingenLijst[[ii]]$MetingenLijst %>% map_dbl(list("Meetwaarde",
"Waarde_Numeriek"), .default = NA)
)
if(ii != 1){
df = rbind(df, temp.df)
}else df = temp.df
}
df[df$numeriekewaarde>1e10,] <- NA
df <- df[!is.na(df$numeriekewaarde),]

# if you wish to retrieve data from many locations at once:
# select locations based on waterbody
wslocations <- rwsapi::select_locations_in_waterbody(metadata, "westerschelde", 2000)

```

To be continued.....