

Download Rijkswaterstaatdata from Waterbase

Waterbase is an online subset of DONAR, the water-related database of Rijkswaterstaat. Data can be downloaded using HTML commands. This is an R script that compiles a list with substances and stations to HTML commands, checks whether there is any data for that particular combination and combines the data in one table.

Two files are needed:

WATERBASE_locations

WATERBASE_parameters

```
#####
#Update Waterbase files
#Author: M.P. Weeber
#Company: Deltares
#####

rm(list = objects())

library("RCurl")
library("stringr")
library("reshape2")
library("tcltk")
library("downloader")
library("ncdf")
library("chron")

mainDir = getwd()
subDir = "DATA"
subDir2 = "dump"

setwd(mainDir)
destination_data = file.path(mainDir, subDir)
destination_dump = file.path(mainDir, subDir2)

#Create Cache folder + dump folder
# dir.create(file.path(mainDir,subDir))
# dir.create(file.path(mainDir,subDir2))

# open required locations HEADER???
WATERBASE_locations = read.csv("YOURLOCATIONS.csv", sep = ";", stringsAsFactor = FALSE)

# open required parameters
WATERBASE_parameters = read.csv("YOURPARAMETERS.csv", sep = ";", stringsAsFactor = FALSE)

#Parts URL
WATERBASE_1 = "http://live.waterbase.nl/wboutput.cfm?loc="
WATERBASE_2
=
"&byear=1700&bmonth=01&bday=01&eyear=2014&emonth=12&eday=31&output=Tekst&whichform=1"

log = c("")

# loop for locations
for(i in 1:length(WATERBASE_locations[,1])){
  for(j in 1:length(WATERBASE_parameters[,1])){
    # Clear old
    if(!(i == 1 & j == 1)){
      rm(list = c("WATERBASE_data", "file", "file2", "REAL_WATERBASE_URL"))
    }
    # Naming for file name
    substantie_char = gsub("/", "_", gsub(" ", "_", WATERBASE_parameters[j,3]))
    get_id = gsub("%7C", "", gsub("&wbwns=", "", WATERBASE_parameters[j,4]))
    file_location = file.path(destination_data, paste("id", get_id, "-", WATERBASE_locations[i,],
                                                    "-170001010000-201406140000.txt",
sep = ""))

    file_location_dump = file.path(destination_dump, paste(WATERBASE_locations[i,],
                                                         "_", substantie_char, ".
txt", sep = ""))
```

```

#Naming for URL
locatie = WATERBASE_locations[i,4]
substantie_code = WATERBASE_parameters[j,4]
substantie = gsub("%","%25",gsub("/","%2F",gsub(" ","+",WATERBASE_parameters[j,3])))

#Download
#Get link to the files
WATERBASE_data <- paste(WATERBASE_1, locatie, substantie_code, substantie, WATERBASE_2, sep = "")

#Connect to repos to get substances
file = getURI(WATERBASE_data)
file2 = unlist(str_split(file, "window.location ="))
REAL_WATERBASE_URL = unlist(str_split(file2[3], "'')[2])

# Check if data exists
if(is.na(REAL_WATERBASE_URL)){
  #Report to Log
  log = c(log, paste("The combination ", locatie, " : ", substantie_char, " does not exist!", sep = ""))
}else{
  #Download the data

  #manuele download
  #shell.exec(WATERBASE_data)
  #n <- readline(prompt="Enter anything to continue or Q to quit: ")
  #if(n == "Q"){
  #  stop(print("Quited execution"))
  #}
  download.file(REAL_WATERBASE_URL, destfile = file_location, mode = "w")
}
}
}

#check if file contains measurements else remove
setwd(destination_data)
files_to_check = list.files(destination_data)

for(k in 1:length(files_to_check)){
  file = readLines(files_to_check[k])
  if(length(file) == 5){
    # save files without data in log
    log = c(log, paste(files_to_check[k], " does not contain data!", sep = ""))
    # remove files without data
    file.remove(files_to_check[k])
  }else{}
}

files_to_bind = list.files(destination_data)
setwd(destination_data)

#test
m = 10; data = read.csv(files_to_bind[m], sep = ";", na.strings = "NA", skip = 3)

for(m in 1:length(files_to_bind)){ ##length(files_to_bind)
  data = read.csv(files_to_bind[m], sep = ";", na.strings = "NA", skip = 3)
  if((m == 1)){
    collected <- data
  }
  collected = rbind(data, collected)
}

write.csv2(collected, "collected-data.csv", row.names = F)

#Evaluate script
log
warnings()
print("Done.")

```

