

# Data Extraction: Energinet Data Service

Kasper Rosenkrands

2/12/2020

## Available Datasets

To see the available datasets from Energinet Data Service the following function have been created. The function takes a URL and a path as input and the provides the available datasets, note that this is specific to Energinet Data Service.

```
available.datasets.energinet <- function(url = "https://api.energidataservice.dk",
                                         path = "/package_list") {
  # store the response from the Energinet Data Service API, this
  # variable is of the class "response"
  raw.result <- GET(url = url, path = path)
  # the raw.result will have a list element of name "content"
  # of the class "raw", we will need to convert this to something
  # that we can interpret. This is done using the rawToChar function
  this.raw.content <- rawToChar(raw.result$content)
  # it now becomes clear that we have a JSON format, which we can convert
  # using the fromJSON function. This will give us a list with "result" as one
  # of the list entries
  this.content <- fromJSON(this.raw.content)
  # we are now able to store the result in a dataframe
  return(as.data.frame(this.content$result))
}
(datasets <- available.datasets.energinet())
```

```
##               this.content$result
## 1               afrrreservesdk1
## 2                   brineswap
## 3               calorificvalues
## 4               capacityauctiondk1
## 5               capacityauctiondk2
## 6               capacitypermunicipality
## 7                   co2emis
## 8                   co2emisprog
## 9               communityconsumption
## 10              communityproduction
## 11              consumptionpergridarea
## 12              consumptionpermunicipalityde35
## 13              datahubpricelist
## 14              declarationemissionhour
## 15              declarationproductiontypeshour
## 16                  electricitybalance
## 17              electricitybalancenonv
## 18              electricityprodex5minrealtime
```

```

## 19      electricitysupplierspergridarea
## 20              elspotprices
## 21              entryexitgasquality
## 22              fcrreservesdk1
## 23              fcrreservesdk2
## 24      fixedresidualconsumption
## 25              forecasts_5min
## 26              forecasts_hour
## 27              gascomposition
## 28      gascompositionmonthly
## 29      gascompositionyearly
## 30              gasflow
## 31      gassystemrightnow
## 32 herfindahlhirschmanindexhhipergridarea
## 33              industrycodes_de35
## 34      inertianordicsyncharea
## 35              mfrrreservesdk1
## 36              mfrrreservesdk2
## 37      mrstationsgasquality
## 38              nordpoolmarket
## 39      powersystemrightnow
## 40              realtimemarket
## 41      storageavailablepct
## 42              storagecapacity
## 43              storagenomination
## 44              storageutilization
## 45      transmissionlines

```

## Extract a Dataset

To extract a certain dataset one could use the function below.

```

data.extract <- function(resource_id) {
  # first we need to determine total row count
  url  <- "https://api.energidataservice.dk"
  # we combine a string with the certain resource_id (table name)
  # that we want to extract
  path <- paste("datastore_search?resource_id=",
                resource_id,
                "&limit=5",
                sep = ' '
  )
  raw.result <- GET(url = url, path = path)
  this.raw.content <- rawToChar(raw.result$content)
  this.content <- fromJSON(this.raw.content)
  # the number of rows in the dataset is stored in the following variable
  row.limit <- this.content$result$total
  # as we now know how many rows we need to extract we are ready to pull
  # the actual data.
  # we need to provide a rowlimit to the path which we can now do
  path <- paste("datastore_search?resource_id=",
                resource_id,
                "&limit=",
                as.character(row.limit),

```

```

        sep = ''
    )
    raw.result <- GET(url = url, path = path)
    this.raw.content <- rawToChar(raw.result$content)
    this.content <- fromJSON(this.raw.content)
    # the actual data from the given table is stored in the following variable
    data.extract.data.frame <- as.data.frame(this.content$result$records)
    return(data.extract.data.frame)
}

```

Below is a bit simpler formulation of the above function, however performance wise they seem to be equivalent.

```

data.extract.2 <- function(resource_id) {
  # Determine total row count
  url <- "https://api.energidataservice.dk"
  path <- paste("datastore_search?resource_id=",
               resource_id,
               "&limit=5",
               sep = '')
  )
  this.content <- fromJSON(
    rawToChar(
      GET(url = url, path = path)$content
    )
  )
  row.limit <- this.content$result$total
  # Extract all the observations to a data.frame
  path <- paste("datastore_search?resource_id=",
               resource_id,
               "&limit=",
               as.character(row.limit),
               sep = '')
  )
  data.extract.data.frame <- as.data.frame(
    fromJSON(
      rawToChar(
        GET(url = url, path = path)$content
      )
    )$result$records
  )
  return(data.extract.data.frame)
}

```

## Extract All

Suppose you would like to have a local copy of all the available datasets. This is a possible using the following function. Note however that running this function will take about an hour. I don't know if it's possible to optimize the code or if it is the data origin that is the limiting factor.

```

data.extract.all <- function() {
  # specify where we would like to extract the data to
  setwd("Z:/")
  # create a timestamped folder in the given directory
  dir.create(paste("Extracted_data_from_", Sys.Date(), sep = ''))
}

```

```

# change directory into this folder
setwd(paste("Z:", "/Extracted_data_from_", Sys.Date(), sep = ''))
# how many datasets do we need to extract
total <- length(data.set.list)
# count the iteration
iter <- 0
# record the starting time
start <- Sys.time()
for (i in 1:length(data.set.list)) {
  # record the beginning of the iteration
  begin <- Sys.time()
  iter <- iter + 1
  # extract the data to the variable data.set
  data.set <- data.extract.2(as.character(data.set.list[i]))
  # save the data.set variable in a file named after the data set
  save(data.set, file = paste(as.character(data.set.list[i]), ".Rdata", sep = ''))
  # record the end of the iteration
  end <- Sys.time()
  # let the user know the duration of the iteration etc.
  cat(iter, "of", total, "done | Iteration time:", difftime(end, begin, units = "secs"),
      "seconds | Total time spent:", difftime(end, start, units = "mins") , "minutes", "\n")
}
# record the time for which the extraction concluded
finished <- Sys.time()
# let the user know the total duration
cat("Total time spent:", difftime(finished, start, units = "mins"), "minutes", "\n")
}
#data.extract.all()

```