Package: NHSRtools (via r-universe)

September 3, 2024

Title NHS-R Tools

Version 0.0.0.9001

Description Provides tools commonly used by Analysts within the NHS.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Depends R (>= 3.5.0)

BugReports <https://github.com/nhs-r-community/NHSRtools>

Imports assertthat, dplyr, httr, janitor, rlang, stringr, readr,
    purrr, sf, lubridate

Suggests covr, devtools, magrittr, mockery, purrr, roxygen2, testthat,
    tibble, tidyr, usethis

Repository https://nhs-r-community.r-universe.dev

RemoteUrl https://github.com/nhs-r-community/NHSRtools

RemoteRef main

RemoteSha 070b7d031396fba06532edb60d48e97ae2fd60c3

Contents

---

nhspd_as_sfc *Convert NHSPD table to SF*

Description

Converts NHSPD to a SF object by turning the OS easting and northing values to a st_point

Usage

```
nhspd_as_sfc(nhspd)
```

Arguments

nhspd NHS Postcode Database object (loaded by nhspd_load())

---

nhspd_load *NHS Postcode Directory*

Description

The NHS Postcode Directory is a database of all of the Postcodes within the United Kingdom, along with the codes for things like CCG, STP, Local Authority for which that postcode belongs to. Unfortunately, the raw csv's do not have any column headings, so it's pretty difficult to use.

Usage

```
nhspd_load(
  file = dir(path, "^nhg.*\\.csv$", full.names = TRUE),
  path = "./Data"
)
```

Arguments

file the file to load, defaults to a file that begins with "nhg" in the folder "path"

path the path where the file is stored. Defaults to "Data" in the current directory. Change this to where you saved the downloaded files, or set "." for the current directory

Details

This script will handle loading of the csv by adding in the correct column headings and datatypes for those columns.

Instructions for use ——–

Download from https://geoportal.statistics.gov.uk the latest NHSPD file. This is found under the "Postcodes"->"NHS Postcode Directory (NHSPD)" menu.

Download the "NHS Postcode Directory UK Full" version, and not the extract version.

Unzip the contents of the download and update the "path" variable to point to the correct location (alternatively, extract the zip contents to your current R working directory: you can view this folder easily by clicking the "More" button in the R-Studio Files pane and selecting "Show folder in new window").

Inside the folder there are a number of directories. If you look at the word document inside of the "Contents" folder this will explain the structure of the NHSPD. The "User Guide" is also very useful as it explains what each of the columns contains.

NOTE: loading the complete csv will use up a large amount of RAM, (the Nov-19 version used 2.3 GB of RAM on my machine). You may wish to use one of the regional files (see the Contents folder to figure out which to use), or you may wish to consider altering the column types to skip columns that you are not interested in. To do this, change the col type for that column to be col_skip().

ODS_API_ENDPOINT *ODS API Endpoint*

Description

The API endpoint for the NHS Organisation Data Services: all calls to get data should start from this URI.

Usage

```
ODS_API_ENDPOINT
```

Format

An object of class character of length 1.

`ods_api_results_limit`   *ODS API Results Limit*

Description

Get's, or set's, the current limit of results to be returned by the ODS API. If a value is provided for the limit argument then the value is updated. If a value is not provided then the current limit value is returned.

Usage

```
ods_api_results_limit(limit)
```

Arguments

limit   an integer between 1 and 1000

Value

the currently set limit, or the updated limit

Examples

```
# get the current limit
ods_api_results_limit() # defaults to 1000

# we can update this limit to a value between 1 and 1000
ods_api_results_limit(50) # should return 50
ods_api_results_limit() # should now be 50
```

`ods_get_organisations`   *ODS Organisations*

Description

This function queries the ODS Search API.

Usage

```
ods_get_organisations(
  name = as.character(NA),
  post_code = as.character(NA),
  last_change_date = as.Date(NA),
  status = c(NA, "active", "inactive"),
  primary_role_id = as.character(NA),
  non_primary_role_id = as.character(NA),
```

```
  org_record_class = as.character(NA)
)

ods_get_trusts(
  name = as.character(NA),
  post_code = as.character(NA),
  last_change_date = as.Date(NA),
  status = c(NA, "active", "inactive"),
  non_primary_role_id = as.character(NA),
  org_record_class = as.character(NA)
)

ods_get_trust_sites(
  name = as.character(NA),
  post_code = as.character(NA),
  last_change_date = as.Date(NA),
  status = c(NA, "active", "inactive"),
  non_primary_role_id = as.character(NA),
  org_record_class = as.character(NA)
)

ods_get_ccgs(
  name = as.character(NA),
  post_code = as.character(NA),
  last_change_date = as.Date(NA),
  status = c(NA, "active", "inactive"),
  non_primary_role_id = as.character(NA),
  org_record_class = as.character(NA)
)
```

Arguments

`name`	a character: queries organisations that contain that word in the name of the organisation
`post_code`	a character: queries organisations based on their postcode. You can search by either the full postcode, or a partial postcode. If you use a partial postcode then it must be at least the Outcode portion of the postcode (e.g. B1, WV10).
`last_change_date`	a date: queries all organisations that have been changed after that date
`status`	a character (either active or inactive): searches for organisations based on their status
`primary_role_id`	a character: queries all organisations that have the specified primary role (see `ods_get_roles`)
`non_primary_role_id`	a character: queries all organisations that have the role as a secondary role (see `ods_get_roles`)

```
org_record_class
```
                    a character: queries all organisations based on their record class TODO: imple-
                    ment a function that queries this API

Details

One or more arguments should be specified, if not it will default (with a warning to using the
primary role of "NHS Trust")

Value

a tibble of organisations

Examples

```
## Not run:
# this will default to getting organisations with a primary role of "NHS
# TRUST", but it will throw a warning:
ods_get_organisations()

# you could be more specific and specify that you want to use a primary role
# of "NHS TRUST" using the code from ods_get_roles()
primary_role_id <- ods_get_roles() %>%
  filter(displayName == "NHS TRUST") %>%
  pull(id)
ods_get_organisations(primary_role_id = primary_role_id)

# or, you could use the specific functions for the primary role id's
ods_get_trusts()
ods_get_trust_sites()
ods_get_ccgs()

# you can use as many of the arguments as you like
# get current active CCG's
ods_get_ccgs(status = "active")

# get NHS Trust Sites with "Royal" in their name
ods_get_trust_sites(name = "Royal")

## End(Not run)
```

```
ods_get_roles                *ODS Roles*
```

Description

This function queries the ODS Roles API.

Usage

```
ods_get_roles()
```

Details

Each organisation that is in the ODS organisations API will have a primary role (e.g. NHS TRUST) and can have secondary roles. These role ids can be used to query the Organisations api.

Value

A tibble containing the roles that can be used to query the ODS organisations api (with `ods_get_organisations`)

Examples

```
## Not run:
ods_get_roles()

## End(Not run)
```

---

ods_get_successors   *ODS Successor Org Lookup*

Description

Builds a lookup table for organisations that have merged over time

Usage

```
ods_get_successors()
```

Index