Package ‘NHSDataDictionary’

Type Package
Title NHS Data Dictionary Toolset for NHS Lookups
Version 1.2.5
Maintainer Gary Hutson <hutsons-hacks@outlook.com>
Description Providing a common set of simplified web scraping tools for working with the NHS Data Dictionary <https://datadictionary.nhs.uk/data_elements_overview.html>. The intended usage is to access the data elements section of the NHS Data Dictionary to access key lookups. The benefits of having it in this package are that the lookups are the live lookups on the website and will not need to be maintained.
This package was commissioned by the NHS-R community <https://nhsrcommunity.com/> to provide this consistency of lookups. The OpenSafely lookups have now been added <https://www.opencodelists.org/docs/>.
License MIT + file LICENSE
Encoding UTF-8
LazyData false
RoxygenNote 7.1.1
Imports xml2, dplyr, magrittr, rvest, stringr, purrr, tibble, httr
Collate 'left_xl.R' 'len_xl.R' 'linkScrapeR.R' 'mid_xl.R'
  'nhs_data_elements.R' 'scrapeR.R' 'tableR.R'
  'nhs_table_findR.R' 'right_xl.R' 'openSafely_listR.R'
  'xpathTextR.R'
Suggests knitr, markdown, spelling
VignetteBuilder knitr
Language en-US
NeedsCompilation no
Author Gary Hutson [aut, cre] (<https://orcid.org/0000-0003-3534-6143>), Calum Polwart [aut], Tom Jemmett [aut] (<https://orcid.org/0000-0002-6943-2990>)
Repository CRAN
Date/Publication 2021-07-09 13:10:05 UTC
R topics documented:

- left_xl
- len_xl
- linkScrapeR
- mid_xl
- nhs_data_elements
- nhs_table_findeR
- openSafely_listR
- right_xl
- scrapeR
- tableR
- xpathTextR

Index

left_xl

left_xl function

This function replicates the LEFT function in Excel and is utilised for left trimming of character strings

Description

left_xl function

This function replicates the LEFT function in Excel and is utilised for left trimming of character strings

Usage

left_xl(text, num_char = 0)

Arguments

- text: The text you want to LEFT trim
- num_char: The number of characters you want to trim by

Value

Trims the text entered by the number of character parameter and returns the trimmed string

Examples

left_xl(text = "This is some example text", num_char = 4)
### len_xl

**Description**

This function replicates the LEN function in Excel and is utilised for finding the length of character strings.

**Usage**

\[
\text{len}_x_l(\text{text}, \ldots)
\]

**Arguments**

- **text**: The text you want to calculate the length
- **...**: Function forwarding to work with the base nchar method

**Value**

An integer value calculating the length of the text passed

**Examples**

\[
\text{len}_x_l(\text{"Guess the length of me!"})
\]

### linkScrapeR

**Description**

This is used to scrape all hyperlinks from a specific web page.

**Usage**

\[
\text{linkScrapeR}(\text{url}, \text{SSL}_\text{needed} = \text{FALSE})
\]

**Arguments**

- **url**: The website URL to detect active anchor hyperlink tags and extract them into a tibble
- **SSL_needed**: Default - FALSE - Boolean to indicate whether to need a SSL certificate
Details
Once the links have been scraped they will be outputted into a tibble for exploration. This can be used on any website to pull back the hyperlink content of a web page.

Value
A tibble (class data.frame) with all active hyperlinks on the website for the URL (uniform resource locator) passed to the function.

- result - the extracted html table from url and xpath passed
- link_name - the name of the link
- url - the full url of the active href tag from HTML

Examples
linkScrapeR("https://www.datadictionary.nhs.uk/", FALSE)

mid_xl function

Description
This function replicates the MID function in Excel and is utilised for left trimming of character strings.

Usage
mid_xl(text, start_num = 1, num_char = 0)

Arguments
text The text you want to MID trim
start_num The start number to start the trim. This needs to be numeric.
num_char The number of characters you want to trim by. This field needs to be numeric.

Details
This has been included as a convenience function for working with text and string data.

Value
The extracted text between the start_num and the num_char to produce a sub string result.
nhs_data_elements

Examples

mid_xl(text = "This is some example text", start_num = 6, num_char = 10)

mid_xl(text = "This is some example text", start_num = 6, num_char = 10)

description

Searches all the data elements in the data element index of the NHS data dictionary and returns the links.

Usage

nhs_data_elements()

details

This function has no input parameters and returns the

Value

A tibble (class data frame) with the results of scraping the NHS Data Dictionary website for the data elements look ups, if no return this will produce an appropriate informational message.

- link_name - the name of the scraped link. This relates to the actual name of the data element from the NHS Data Dictionary.
- url - the url passed to the parameter
- full_url - the full url of where the data element is on the NHS Data Dictionary website
- xpath_nat_code - utilises the element in the website and appends the link_short - to pull back only national codes from the dictionary site. NOTE: not all of the returns will have national code tables.
- xpath_default_codes - pulls back the data dictionary default codes - these can be then used with the national codes
- xpath_also_known - pulls back the data dictionary elements alias table - this will be available for all data elements

Examples

nhs_data_lookup <- nhs_data_elements()
head(nhs_data_lookup, 10)
Description

This function uses the `tableR` parent function to return a table of elements, specifically from the NHS Data Dictionary.

Usage

```r
nhs_table_findeR(data_element_name, ...)
```

Arguments

- `data_element_name`: The data element name from NHS Data Dictionary i.e. ACCOMMODATION STATUS CODE
- `...`: Function forwarding to parent function to pass additional arguments to function (e.g. title, add_zero_prefix)

Value

A tibble (class data.frame) output from the results of the web scrape

- `result`: the extracted national HTML code table from the element page of the NHS Data Dictionary
- `DictType`: defaults to Not Specified if nothing passed, however allows for custom dictionary / data frame tags to be created
- `DttmExtracted`: a date and time stamp

Examples

```r
# Returns a tibble from tableR parent function
nhs_table_findeR("ACCOMMODATION STATUS CODE", title="ACCOM_STATUS")
nhs_table_findeR("accommodation status code") # Changes case to match
```
openSafely_listR  

**Description**

This function uses the `tableR` parent function to return a table of elements, specifically from the OpenSafely Code List [https://www.opencodelists.org/](https://www.opencodelists.org/)

**Usage**

```r
openSafely_listR(list_name, version = "", ...)
```

**Arguments**

- `list_name` The code list ID from [https://www.opencodelists.org/](https://www.opencodelists.org/) for which to return the National table of elements, for example "opensafely/ace-inhibitor-medications"
- `version` The version of the code list if not the most recent
- `...` Function forwarding to parent function to pass additional arguments to function (e.g. title, add_zero_prefix)

**Value**

A tibble (class data.frame) output from the results of the web scrape

- `type` - the OpenSafely type
- `id` - the id for the OpenSafely element
- `bnf_code` - British National Formulary - NICE guidelines code
- `nm` - medicine type, dosage and manufacturer
- `Dict_type` - title specified for dictionary
- `DttmExtracted` - the date and time the code set was extracted

**Examples**

```r
openSafely_listR("opensafely/ace-inhibitor-medications")
# Pull back current list
openSafely_listR("opensafely/ace-inhibitor-medications", "2020-05-19")
# Pull back list with date
```
### right_xl function

**Description**

This function replicates the RIGHT function in Excel and is utilised for right trimming of character strings.

**Usage**

```r
right_xl(text, num_char = 0)
```

**Arguments**

- `text`: The text you want to RIGHT trim
- `num_char`: The number of characters you want to trim by. This field needs to be numeric.

**Details**

This has been included as a convenience function for working with text and string data.

**Value**

The trimmed string from the text parameter and trimming by the number of characters `num_char` passed to the parameter.

**Examples**

```r
right_xl(text= "This is some example text", num_char = 10)
right_xl(text= "This is some example text", num_char = 10)
```

---

### scrapeR

**Description**

Takes the url and xpath and scrapes HTML table elements from a website.

**Usage**

```r
scrapeR(url, xpath, ...)
```
tableR

Arguments

- url: Website address to connect to
- xpath: Xpath obtained through inspecting the individual HTML elements
- ...: Function to pass additional function forwarding options

Details

This function is specifically designed to work with HTML tables and xpath links through to direct HTML elements. The function is versatile and can be used on any URL where an xpath can be obtained through the URL and HTML inspection process.

Value

Returns the results of the scraping operation and the relevant fields from the html table - the xpath should make reference to an html table, otherwise an error is returned advising the user to check the xpath and url are correct.

tableR function

Description

This function uses the scapeR parent function to return a table of elements

Usage

tableR(url, xpath, title = "Not Specified", add_zero_prefix = FALSE, ...)

Arguments

- url: The URL of the website to scrape the table element from
- xpath: The unique xpath of the HTML element to be scraped
- title: Unique name for the relevant HTML table that has been scraped
- add_zero_prefix: Adds zero prefixes to certain codes that get converted by native functions
- ...: Function forwarding to parent function to pass additional arguments to function

Value

A tibble (class data.frame) output from the results of the web scrape

- result - the extracted html table from url and xpath passed
- DictType - defaults to Not Specified if nothing passed, however allows for custom dictionary / data frame tags to be created
- DttmExtracted - a date and time stamp
**xpathTextR**

**xpathTextR function**

**Description**

Returns xpath text from websites and can be used to access specific HTML nodes.

**Usage**

`xpathTextR(url, xpath, ssl_needed = FALSE)`

**Arguments**

- **url**
  The link for the website
- **xpath**
  The xpath string derived by using the Inspect functionality in a web browser.
- **ssl_needed**
  Default - FALSE - Boolean to indicate whether to need a SSL certificate

**Value**

A list with the results of scraping the specific xpath element

- **result** - the extracted text from the website element that has been scraped
- **website_passed** - a copy of the input url for the website
- **html_node_result** - returns the extracted html node result
- **datetime_access** - returns a timestamp of when the results of the scraping operation have been completed
- **person_accessed** - retrieves the system environment stored username and domain - this is concatenated together to form a mixed character string
Index

left_xl, 2
len_xl, 3
linkScrapeR, 3

mid_xl, 4

nhs_data_elements, 5
nhs_table_findeR, 6
openSafely_listR, 7

right_xl, 8
scrapeR, 8
tableR, 6, 7, 9
xpathTextR, 10