

# Package ‘NHSDataDictionary’

May 5, 2021

**Type** Package

**Title** NHS Data Dictionary Toolset for NHS Lookups

**Version** 1.2.3

**Maintainer** Gary Hutson <g.hutson@nhs.net>

**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](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 to provide this consistency of lookups. The OpenSafely lookups have now been added <<https://codelists.opensafely.org/>>.

**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' 'right\_xl.R' 'scrapeR.R' 'tableR.R' 'openSafely\_listR.R' 'xpathTextR.R'

**Suggests** knitr, rmarkdown, spelling

**VignetteBuilder** knitr

**Language** en-US

**NeedsCompilation** no

**Author** Gary Hutson [aut, cre],  
Calum Polwart [aut]

**Repository** CRAN

**Date/Publication** 2021-05-05 06:50:02 UTC

## R topics documented:

left_xl . . . . .	2
len_xl . . . . .	3
linkScrapeR . . . . .	3
mid_xl . . . . .	4
nhs_data_elements . . . . .	5
openSafely_listR . . . . .	6
right_xl . . . . .	7
scrapeR . . . . .	7
tableR . . . . .	8
xpathTextR . . . . .	9

<b>Index</b>	<b>10</b>
--------------	-----------

---

left_xl	<i>left_xl function This function replicates the LEFT function in Excel and is utilised for left trimming of character strings</i>
---------	--

---

### 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 your 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	<i>len_xl function</i>
--------	------------------------

---

**Description**

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

**Usage**

```
len_xl(text, ...)
```

**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**

```
len_xl("Guess the length of me!")
```

---

linkScrapeR	<i>linkScrapeR</i>
-------------	--------------------

---

**Description**

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

**Usage**

```
linkScrapeR(url, SSL_needed = 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://nhsrcommunity.com/")
```

---

mid_xl	<i>mid_xl function</i>
--------	------------------------

---

**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 your 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.

**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)
```

---

nhs\_data\_elements      *NHS data elements method*

---

**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)
```

---

openSafely\_listR      *openSafely\_listR function*

---

### Description

This function uses the `tableR` parent function to return a table of elements, specifically from the OpenSafely Code List <https://codelists.opensafely.org/>

### Usage

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

### Arguments

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

### 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

```
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	<i>right_xl function</i>
----------	--------------------------

---

**Description**

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

**Usage**

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

**Arguments**

text	The text you want to RIGHT trim
num_char	The number of characters your 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**

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

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

---

scrapeR	<i>ScrapeR - scrape web information with scrapeR</i>
---------	--

---

**Description**

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

**Usage**

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

**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 x path 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	<i>tableR function</i>
--------	------------------------

---

**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**

<code>url</code>	The link for the website
<code>xpath</code>	The xpath string derived by using the Inspect functionality in a web browser.
<code>ssl_needed</code>	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

openSafely\_listR, 6

right\_xl, 7

scrapeR, 7

tableR, 6, 8

xpathTextR, 9