Package ‘uklr’

October 12, 2022

Title  Client to United Kingdom Land Registry
Version  1.0.2
Description  Access data from Land Registry Open Data
<http://landregistry.data.gov.uk/> through 'SPARQL' queries. 'uklr'
supports the house price index, transaction and price paid data.
License  GPL-3
URL  https://kvasilopoulos.github.io/uklr/,
https://github.com/kvasilopoulos/uklr/
BugReports  https://github.com/kvasilopoulos/uklr/issues
Depends  R (>= 3.0.2)
Imports  curl (>= 4.3), httr (>= 1.4.1), jsonlite (>= 1.6), tibble (>= 2.1.3)
Suggests  markdown (>= 1.1), covr (>= 3.5.0), spelling (>= 2.1),
testthat (>= 2.1.0), knitr (>= 1.29)
Encoding  UTF-8
Language  en-US
LazyData  true
RoxygenNote  7.1.1
VignetteBuilder  knitr
NeedsCompilation  no
Author  Kostas Vasilopoulos [aut, cre]
  (<https://orcid.org/0000-0002-9769-6395>)
Maintainer  Kostas Vasilopoulos <k.vasilopoulos@gmail.com>
Repository  CRAN
Date/Publication  2021-05-25 00:10:02 UTC
R topics documented:

- ons_countries .................................................. 2
- ons_lookup ...................................................... 3
- ons_pc .......................................................... 3
- pc .............................................................. 4
- retrieve_query .................................................. 4
- sparql .......................................................... 5
- ukhp_avail_items ................................................ 5
- ukhp_get ....................................................... 6
- uklr_browse .................................................... 7
- uklr_def ........................................................ 8
- ukppd_avail_items ............................................. 9
- ukppd_get ..................................................... 9
- uktrans_avail_items ......................................... 10
- uktrans_get ................................................... 11

Index 13

ons_countries | Office of National Statistic Location Classification

Description

Functions `ons_countries`, `ons_regions`, `ons_eng_counties` and `ons_la` provide a vector of countries, regions, English counties and local authorities respectively.

Usage

- `ons_countries(modify = TRUE)`
- `ons_eng_regions(modify = TRUE)`
- `ons_regions(modify = TRUE)`
- `ons_eng_counties(modify = TRUE)`
- `ons_la(modify = FALSE)`

Arguments

- `modify` Modifies the vector to conform with the queries.

Value

A character vector with the location names.
**ons_lookup**

**Examples**

```r
ons_countries()
```

---

**ons_lookup**  
*Office of National Statistic Location Classification*

**Description**

`ons_lookup` provides a table that combines local authority districts (lad), local administrative units (lau), nomenclature of territorial units for statistics (NUTS) 1, 2, and 3. The other functions can extract the mentioned variables.

**Usage**

```r
ons_lookup()
ons_nuts1()
ons_nuts2()
ons_nuts3()
ons_lad()
```

**Value**

A character vector with the location names.

**Examples**

```r
ons_lookup()
```

---

**ons_pc**  
*UK Postcodes*

**Description**

Vector of UK postcodes. Search for matches with the ‘pattern’ argument.

**Usage**

```r
ons_pc(pattern = NULL)
```
Arguments

pattern partial matching

Details

ons_pc("EH21 8A")

---

pc UK Postcodes and NUTS3 Codes

---

Description

UK Postcodes and NUTS3 Codes

Usage

pc

Format

An object of class spec_tbl_df (inherits from tbl_df, tbl, data.frame) with 1759911 rows and 2 columns.

Examples

pc

---

retrieve_query Get the sparql query performed with sparql.

---

Description

Get the sparql query performed with sparql.

Usage

retrieve_query(x)

Arguments

x the result of query.

Value

Returns the a character vector with the query.
______sparql_________Tools to create a custom SPARQL query__________

**Description**

Function to create custom queries with the ‘/landregistry/query endpoint’. All necessary prefixes have to be included and output will be parsed from json to dataframe. The most prefixes from land registry can be include with prefix_query.

**Usage**

```
sparql(
   query,
   endpoint = "http://landregistry.data.gov.uk/landregistry/query",
   ...
)
```

```
prefix_query(query)
```

**Arguments**

- `query` custom query.
- `endpoint` land registry’s web service endpoint.
- `...` further arguments passed to `httr::GET`.

**Value**

Returns a tibble that has been parsed from json.

**Examples**

```
custom_query <- "select * where {
}"

sparql(prefix_query(custom_query))
```

---

---

**ukhp_avail_items**

Display Land Registry available categories

**Description**

`ukhp_avail_regions()` displays available regions, `ukhp_avail_regions()` displays available items, `ukhp_avail_date_span()` displays available date sample, and `ukhp_avail_date_last` displays the last available date.
ukhp_get

Usage

ukhp_avail_items()
ukhp_avail_regions()
ukhp_avail_date_span()
ukhp_avail_date_last()

Value

Returns a character vector.

Examples

ukhp_avail_regions()[1:10]
ukhp_avail_items()
ukhp_avail_date_last()

ukhp_get

Get House Price Data

Description

The UK House Price Index (UK HPI) captures changes in the value of residential properties. The UK HPI uses sales data collected on residential housing transactions, whether for cash or with a mortgage. Data is available at a national and regional level, as well as counties, local authorities and London boroughs.

Usage

ukhp_get(
  region = "england",
  item = "housePriceIndex",
  regexp = FALSE,
  start_date = NULL,
  end_date = NULL,
  ...
)

Arguments

region the region to select. If regexp is set to FALSE then the matching should be exact, see ukhp_avail_regions for available regions. If regexp is set to TRUE then partial matching is possible. Furthermore if it is set to NULL then selects all available regions.
uklr_browse

    item     the item to select. See ukhp_avail_items for the available categories.
    regexp   use regular expression in sparql to search for regions.
    start_date the start date as YYYY-MM-DD.
    end_date  the end date as YYYY-MM-DD.
    ...      query modifiers passed through rdf_modifiers.

Details

Properties have been included:

- in England and Wales since January 1995
- in Scotland since January 2004
- in Northern Ireland since January 2005

Value

Returns a tibble in long format.

See Also

rdf_modifiers

Examples

# This is case sensitive
ukhp_get("england")

# However you can use regular expression instead of exact match
ukhp_get("england", regexp = TRUE)

# For all available items
ukhp_avail_items()

ukhp_get(c("england", "wales"), item = c("salesVolume", "housePriceIndexDetached"))

uklr_browse

Quickly browse to Land Registry's webpage

Description

These functions take you to land registry’s webpages and return the URL invisibly.
Usage
uklr_browse()
ukhp_browse()
ukppd_browse()
uktrans_browse()

Examples
uklr_browse()

---

Linked Data Definitions

Description
These functions return the definitions for the linked data construction in a tibble.

Usage
uklr_def()
ukhp_def()
ukppd_def()
uktrans_def()

Value
Returns a tibble.

Examples
uklr_def()
ukhp_def()
**ukppd_avail_items**

Display Price Paid Date available categories

**Description**

ukppd_avail_items() displays all available items for search, ukppd_avail_optional_items() displays optional items that refer to the location of the transaction and ukppd_avail_postcodes displays the available postcodes. /

**Usage**

ukppd_avail_items()

ukppd_avail_optional_items()

ukppd_avail_postcodes()

**Value**

Returns a character vector.

**Examples**

ukppd_avail_items()

ukppd_avail_optional_items()

ukppd_avail_postcodes()[1:10]

**ukppd_get**

Get Price Paid Data

**Description**

Price Paid Data tracks property sales in England and Wales submitted to HM Land Registry for registration. Price Paid Data is based on the raw data released each month.

**Usage**

ukppd_get(
    postcode = "PL6 8RU",
    item = NULL,
    optional_item = NULL,
    start_date = NULL,
)
Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>postcode</td>
<td>Postcode to select, see <code>ukppd_avail_postcode()</code> for available postcodes.</td>
</tr>
<tr>
<td>item</td>
<td>Item to select, see <code>ukppd_avail_items()</code> for available items.</td>
</tr>
<tr>
<td>optional_item</td>
<td>Optional item to select that describes the location of the transaction, see <code>ukppd_avail_optional_items()</code> for available items.</td>
</tr>
<tr>
<td>start_date</td>
<td>The start date as YYYY-MM-DD.</td>
</tr>
<tr>
<td>end_date</td>
<td>The end date as YYYY-MM-DD.</td>
</tr>
<tr>
<td>...</td>
<td>Query modifiers passed through <code>rdf_modifiers</code>.</td>
</tr>
</tbody>
</table>

Value

Returns a tibble in long format.

Examples

```r
ukppd_get("PL6 8RU")
ukppd_get("PL6 8RU", start_date = "2001-01-01")
ukppd_get("PL6 8RU", item = "newBuild", optional_item = "street")
```

---

## uktrans_avail_items

*Display Land Registry Transaction available categories*

### Description

`uktrans_avail_items()` and `uktrans_avail_regions()` display available items and regions for search respectively.

### Usage

```r
uktrans_avail_items()
uktrans_avail_regions()
```

### Value

Returns a character vector.
**uktrans_get**

**Examples**

```r
uktrans_avail_items()
uktrans_avail_regions()
```

---

**uktrans_get**  
*Get Transaction Data*

**Description**

Transaction Data is a dataset that shows how many customer applications we completed, in the preceding month for: first registrations, leases, transfers of part, dealings, official copies and searches. This is based on customer and location.

**Usage**

```r
uktrans_get(
  item = "totalApplicationCountByRegion",
  region = NULL,
  regexp = TRUE,
  start_date = NULL,
  end_date = NULL,
  ...
)
```

**Arguments**

- `item`: item to select, see `uktrans_avail_items()` for available items.
- `region`: region to select, see `uktrans_avail_regions()` for available regions.
- `regexp`: use regular expression in sparql to search for regions.
- `start_date`: the start date as YYYY-MM-DD.
- `end_date`: the end date as YYYY-MM-DD.
- `...`: query modifiers passed through `rdf_modifiers`.

**Value**

Returns a tibble in long format.

**Examples**

```r
uktrans_get(item = "totalApplicationCountByRegion", region = "East Anglia")
```

# If `region` is left as NULL then it returns all available regions
```r
test <- uktrans_get(item = "totalApplicationCountByRegion")
```
# Querying all available transaction data

```python
uktrans_get(item = uktrans_avail_items())
```
Index

* datasets
  pc, 4
  ons_countries, 2
  onsEng_countries (ons_countries), 2
  onsEng_regions (ons_countries), 2
  ons_la (ons_countries), 2
  ons_lad (ons_lookup), 3
  ons_lookup, 3
  ons_nuts1 (ons_lookup), 3
  ons_nuts2 (ons_lookup), 3
  ons_nuts3 (ons_lookup), 3
  ons_pc, 3
  ons_regions (ons_countries), 2

  pc, 4
  prefix_query (sparql), 5

  retrieve_query, 4

  sparql, 5

  ukhp_avail_date_last
    (ukhp_avail_items), 5
  ukhp_avail_date_span
    (ukhp_avail_items), 5
  ukhp_avail_items, 5
  ukhp_avail_regions (ukhp_avail_items), 5
  ukhp_browse (uklr_browse), 7
  ukhp_def (uklr_def), 8
  ukhp_get, 6
  uklr_browse, 7
  uklr_def, 8
  ukppd_avail_items, 9
  ukppd_avail_optional_items
    (ukppd_avail_items), 9
  ukppd_avail_postcodes
    (ukppd_avail_items), 9
  ukppd_browse (uklr_browse), 7
  ukppd_def (uklr_def), 8
  ukppd_get, 9

  uktrans_avail_items, 10
  uktrans_avail_regions
    (uktrans_avail_items), 10
  uktrans_browse (uklr_browse), 7
  uktrans_def (uklr_def), 8
  uktrans_get, 11