

# Package: uklr (via r-universe)

September 10, 2024

**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

**Repository** <https://kvasilopoulos.r-universe.dev>

**RemoteUrl** <https://github.com/kvasilopoulos/uklr>

**RemoteRef** HEAD

**RemoteSha** 6bce20a76eca369d8cde853e54c238da09e3e211

## Contents

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

<b>Index</b>	<b>12</b>
--------------	-----------

---

ons_countries	<i>Office of National Statistic Location Classification</i>
---------------	---

---

## 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.

## Examples

```
ons_countries()
```

---

ons_lookup	<i>Office of National Statistic Location Classification</i>
------------	---

---

**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**

```
ons_lookup()
```

```
ons_nuts1()
```

```
ons_nuts2()
```

```
ons_nuts3()
```

```
ons_lad()
```

**Value**

A character vector with the location names.

**Examples**

```
ons_lookup()
```

---

ons_pc	<i>UK Postcodes</i>
--------	---------------------

---

**Description**

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

**Usage**

```
ons_pc(pattern = NULL)
```

**Arguments**

pattern           partial matching

**Details**

```
ons_pc("EH21 8A")
```

---

pc	<i>UK Postcodes and NUTS3 Codes</i>
----	-------------------------------------

---

**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	<i>Get the sparql query performed with sparql.</i>
----------------	--

---

**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	<i>Tools to create a custom SPARQL query</i>
--------	--

---

### 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 {
  <http://landregistry.data.gov.uk/data/ukhpi/region/newport/month/2013-10> ?property ?value
}"
sparql(prefix_query(custom_query))
```

---

ukhp_avail_items	<i>Display Land Registry available categories</i>
------------------	---

---

### Description

ukhp\_avail\_regions() displays available regions, ukhp\_avail\_items() displays available items, ukhp\_avail\_date\_span() displays available date sample, and ukhp\_avail\_date\_last displays the last available date.

**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.

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()
```

---

uklr\_def

*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	<i>Display Price Paid Date available categories</i>
-------------------	---

---

**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	<i>Get Price Paid Data</i>
-----------	----------------------------

---

**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,  
  end_date = NULL,  
  ...  
)
```

**Arguments**

postcode	postcode to select, see ukppd_avail_postcode() for available postcodes.
item	item to select, see ukppd_avail_items() for available items.
optional_item	optional item to select that describes the location of the transaction, , see ukppd_avail_optional_items() for available items.
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**

```
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**

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

**Value**

Returns a character vector.

**Examples**

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

---

uktrans_get	<i>Get Transaction Data</i>
-------------	-----------------------------

---

### 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

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

### Arguments

item	item to select, see <code>uktrans_avail_items()</code> for available items.
region	region to select, see <code>uktrans_avail_regions()</code> 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 <code>rdf_modifiers</code> .

### Value

Returns a tibble in long format.

### Examples

```
uktrans_get(item = "totalApplicationCountByRegion", region = "East Anglia")  
  
# If `region` is left as NULL then it returns all available regions  
uktrans_get(item = "totalApplicationCountByRegion")  
  
# Querying all available transaction data  
uktrans_get(item = uktrans_avail_items())
```

# Index

## \* datasets

- pc, 4
- ons\_countries, 2
- ons\_eng\_counties (ons\_countries), 2
- ons\_eng\_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