

Package ‘helsinki’

September 29, 2021

Type Package

Title R Tools for Helsinki Open Data

Version 1.0.5

Description Tools for accessing various open data sources in the Helsinki region in Finland. Current data sources include the Real Estate Department (<<http://ptp.hel.fi/avoindata/>>), Service Map API (<<http://api.hel.fi/servicemap/v2/>>), Linked Events API (<<http://api.hel.fi/linkedevents/v1/>>), Helsinki Region Infoshare statistics API (<<https://dev.hel.fi/stats/>>).

License BSD_2_clause + file LICENSE

Depends R (>= 3.1.0)

Imports jsonlite, httr, purrr, xml2, methods, dplyr, sf, httpcache, curl

Suggests knitr, rmarkdown, remotes, ggplot2, covr, testthat (>= 3.0.0)

URL <http://ropengov.github.io/helsinki/>,
<https://github.com/rOpenGov/helsinki>

BugReports <https://github.com/ropengov/helsinki/issues>

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Encoding UTF-8

RoxygenNote 7.1.2

Config/testthat/edition 3

Author Juuso Parkkinen [aut, cre],
Joonas Lehtomäki [aut],
Pyry Kantanen [aut] (<<https://orcid.org/0000-0003-2853-2765>>),
Leo Lahti [aut] (<<https://orcid.org/0000-0001-5537-637X>>)

Maintainer Juuso Parkkinen <juuso.parkkinen@iki.fi>

Date/Publication 2021-09-29 13:40:02 UTC

R topics documented:

helsinki-package	2
get_city_map	3
get_feature	4
get_feature_list	4
get_hri_stats	5
get_linkedevents	6
get_rakennustietoruudukko	7
get_servicemap	8
get_vaestotietoruudukko	10
select_feature	11
to_sf	12
wfs_api	12

Index	14
--------------	-----------

helsinki-package	<i>Helsinki open data R tools</i>
------------------	-----------------------------------

Description

Brief summary of the helsinki package

Details

Package:	helsinki
Type:	Package
Version:	See sessionInfo() or DESCRIPTION file
Date:	2013-2021
License:	BSD-2-Clause
LazyLoad:	yes

Tools for accessing various open data sources in the Helsinki region in Finland.

Author(s)

Juuso Parkkinen, Leo Lahti, Joonas Lehtomaki and Pyry Kantanen <louhos@googlegroups.com>

References

See citation("helsinki") <https://github.com/rOpenGov/helsinki>

Examples

```
# library(helsinki)
```

`get_city_map`*Get city administrative regions*

Description

Sf object of city districts

Usage

```
get_city_map(city = NULL, level = NULL)
```

Arguments

<code>city</code>	The desired city. Options: Helsinki, Espoo, Vantaa, Kauniainen
<code>level</code>	The desired administrative level. Options are: "suurpiiri", "tilastoalue", "pienalue" and "aanestysalue"

Details

See `list_features()` for a list of all available features

Value

sf object

Author(s)

Pyry Kantanen <pyry.kantanen@gmail.com>

Source

Metropolitan area in districts: <<https://hri.fi/data/fi/dataset/paakaupunkiseudun-aluejakokartat>>

Examples

```
## Not run:  
map <- get_city_map(city = "helsinki", level = "suuralue")  
  
## End(Not run)
```

get_feature	<i>Produce an SF object</i>
-------------	-----------------------------

Description

Produces an sf object for easy visualization

Usage

```
get_feature(  
  base.url = "https://kartta.hsy.fi/geoserver/wfs",  
  typename = "asuminen_ja_maankaytto:Vaestotietoruudukko_2015",  
  CRS = 3879  
)
```

Arguments

base.url	WFS url, for example "https://kartta.hsy.fi/geoserver/wfs"
typename	accepts feature names, e.g. "asuminen_ja_maankaytto:1000m_verkostobufferi" No short form titles here, e.g. "1000m_verkostobufferi"!
CRS	Default CRS is 3879 (or EPSG:3879), see ?sf::st_crs for other input options

Value

sf object

Author(s)

Pyry Kantanen <pyry.kantanen@gmail.com>

See Also

Use [get_feature_list](#) to list all available features for a given WFS, [select_feature](#) for listing and selecting a feature

get_feature_list	<i>Print all available Features</i>
------------------	-------------------------------------

Description

Basically a neat wrapper for "request=GetCapabilities".

Usage

```
get_feature_list(base.url = NULL, queries = "request=GetCapabilities")
```

Arguments

base.url WFS url, for example "https://kartta.hsy.fi/geoserver/wfs"
queries desired query for acquiring the list of features, default is "request=GetCapabilities"

Details

Lists all <FeatureType> nodes.

Value

data frame

Author(s)

Pyry Kantanen <pyry.kantanen@gmail.com>

See Also

Use [get_feature](#) to download feature, [select_feature](#) for menu-driven listing and downloading

Examples

```
## Not run:  
dat <- get_feature_list(base.url = "https://kartta.hsy.fi/geoserver/wfs")  
  
## End(Not run)
```

get_hri_stats

Helsinki Region Infoshare statistics API

Description

Retrieves data from the Helsinki Region Infoshare (HRI) statistics API: <http://dev.hel.fi/stats/>. Currently provides access to the 'aluesarja't data: <http://www.aluesarjat.fi/>.

Usage

```
get_hri_stats(query = "", verbose = TRUE)
```

Arguments

query A string, specifying the dataset to query
verbose logical. Should R report extra information on progress? Default is TRUE

Details

Current implementation is very simple. You can either get the list of resources with query="", or query for a specific resources and retrieve it in a three-dimensional array form.

Value

multi-dimensional array

Author(s)

Juuso Parkkinen <louhos@googlegroups.com>

References

See `citation("helsinki")`

See Also

See <https://dev.hel.fi/apis/statistics/dev.hel.fi> website and <http://dev.hel.fi/stats/API> documentation (in Finnish)

Examples

```
stats_array <- get_hri_stats("aluesarjat_a03s_hki_vakiluku_aidinkieli")
```

get_linkedevents

Access Helsinki Linked Events API

Description

Easy access to Helsinki Linked Events API

Usage

```
get_linkedevents(query, ...)
```

Arguments

query	The API query as a string, for example "event", "category", "language", "place" or "keyword".
...	Additional parameters that narrow down the output (optional).

Details

Complete list of possible query input: "keyword", "keyword_set", "place", "language", "organization", "image", "event", "search", "user".

With "..." the user can pass on additional parameters that depend on the chosen query input. For example, when performing a search (query = "search"), the search can be narrowed down with parameters such as:

- q="konsertti" complete search, returns events that have the word "konsertti"

- input="konse" partial search, returns events with words that contain "konse"
- type="event" returns only "events"
- start="2017-01-01" events starting on 2017-01-01 or after
- end="2017-01-10" events ending on 2017-01-10 or before

For more detailed explanation, see <http://api.hel.fi/linkedevents/v1/>.

Value

Data frame or a list

Author(s)

Juuso Parkkinen <louhos@googlegroups.com>, Pyry Kantanen

Source

Helsinki Linked Events API v1. Developed by the City of Helsinki Open Software Development team. Event data from Helsinki Marketing, Helsinki Cultural Centres, Helmet metropolitan area public libraries and City of Helsinki registry of service unit. CC BY 4.0. <<https://creativecommons.org/licenses/by/4.0/>>

For more API documentation and license information see the API link: <http://api.hel.fi/linkedevents/v1/>

Examples

```
events <- get_linkedevents(query="search", q="teatteri", start="2020-01-01")
```

```
get_rakennustietoruudukko
```

Produce an SF object: Rakennustietoruudukko

Description

Produces an sf object for Rakennustietoruudukko (building information grid)

Usage

```
get_rakennustietoruudukko(year = NULL)
```

Arguments

year year as numeric from range 2015:2019

Details

Additional data not available here can be manually downloaded from HRI website: <https://hri.fi/data/fi/dataset/rakennustietoruudukko>

Years 2015-2019 are tested to work at the time of development. Additional years may be added in the future and older datasets may be removed from the API.

The current datasets can be listed with `get_feature_list()` or `select_feature()`.

Value

sf object

Author(s)

Pyry Kantanen <pyry.kantanen@gmail.com>

Examples

```
## Not run:
building_grid <- get_rakennustietoruudukko(year = 2018)

## End(Not run)
```

get_servicemap

Access Helsinki region Service Map API

Description

Access the new Helsinki region Service Map (Paakaupunkiseudun Palvelukartta) <https://palvelukartta.hel.fi/fi/> data through the API: <http://api.hel.fi/servicemap/v2/>. For more API documentation and license information see the API link.

Usage

```
get_servicemap(query, ...)
```

Arguments

query	The API query as a string, for example "search", "service", or "unit". For full list of available options and details, see https://dev.hel.fi/apis/service-map-backend-api/ .
...	Additional parameters to the API (optional). For additional details, see https://dev.hel.fi/apis/service-map-backend-api/ .

Details

Complete list of possible query input:

- "unit" unit, or service point
- "service" category of service provided by a unit
- "organization" organization providing services
- "search" full text search for units, services and street addresses
- "accessibility" rule database for calculating accessibility scores
- "geography" spatial information, where services are located

With "...", the user can pass on additional parameters that depend on the chosen query input. For example, when performing a search (query = "search"), the search can be narrowed down with parameters such as:

- "q" complete search
- "input" partial search
- "type" valid types: service_node, service, unit, address
- "language" as two-character ISO-639-1 code: fi, sv, en
- "municipality" comma-separated list of municipalities, lower-case, in Finnish
- "service" comma-separated list of service IDs
- "include" include the complete content from certain fields with a comma-separated list of field names with a valid type prefix
- "only" restricts the results with a comma-separated list of field names with a valid type prefix
- "page" request a certain page number
- "page_size" determine number of entries in one page

For more detailed explanation, see <https://dev.hel.fi/apis/service-map-backend-api/>.

Value

Data frame or a list

Author(s)

Juuso Parkkinen <louhos@googlegroups.com>, Pyry Kantanen

Source

API contents: All content is available under CC BY 4.0, except where otherwise stated. The City of Helsinki logo is a registered trademark. The Helsinki Grotesk Typeface is a proprietary typeface licensed by Camelot Typefaces. <<https://creativecommons.org/licenses/by/4.0/>>

API Location: <https://api.hel.fi/servicemap/v2/>

API documentation: <https://dev.hel.fi/apis/service-map-backend-api/>

Examples

```
# A data.frame with 47 variables
search_puisto <- get_servicemap(query="search", q="puisto")
# A data.frame with 7 variables
search_padel <- get_servicemap(query="search", input="padel",
  only="unit.name, unit.location.coordinates, unit.street_address",
  municipality="helsinki")
```

`get_vaestotietoruudukko`*Produce an SF object: Vaestotietoruudukko*

Description

Produces an sf object for Väestötietoruudukko (population grid)

Usage

```
get_vaestotietoruudukko(year = NULL)
```

Arguments

year year as numeric from range 2015:2019

Details

Additional data not available here can be manually downloaded from HRI website: <https://hri.fi/data/fi/dataset/vaestotietoruudukko>

Years 2015-2019 are tested to work at the time of development. Additional years may be added in the future and older datasets may be removed from the API.

The current datasets can be listed with `get_feature_list()` or `select_feature()`.

Value

sf object

Author(s)

Pyry Kantanen <pyry.kantanen@gmail.com>

Examples

```
## Not run:  
pop_grid <- get_vaestotietoruudukko(year = 2017)  
  
## End(Not run)
```

select_feature	<i>Interactively browse and select features</i>
----------------	---

Description

Use an interactive menu to select and download a feature for use in other functions

Usage

```
select_feature(base.url = NULL, get = FALSE)
```

Arguments

base.url	WFS url, for example "https://kartta.hsy.fi/geoserver/wfs"
get	Should the selected feature be downloaded? Default is FALSE

Value

feature Title (character) or feature object (sf), if get parameter is TRUE

Author(s)

Pyry Kantanen <pyry.kantanen@gmail.com>

See Also

[get_feature](#), [get_feature_list](#)

Examples

```
## Not run:
selection <- select_feature(base.url = "https://kartta.hsy.fi/geoserver/wfs")
feature <- get_feature(base.url = "https://kartta.hsy.fi/geoserver/wfs", type_name = selected)
ggplot(feature) +
  geom_sf()

## End(Not run)
```

to_sf	<i>Transform to sf-object</i>
-------	-------------------------------

Description

Transform a wfs_api object into a sf object.

Usage

```
to_sf(api_obj)
```

Arguments

api_obj	wfs api object
---------	----------------

Details

FMI API response object's XML (GML) content is temporarily wrtitten on disk and then immediately read back in into a sf object.

Value

sf object

Note

For internal use, not exported.

Author(s)

Joona Lehtomäki <joona.lehtomaki@iki.fi>

wfs_api	<i>WFS API</i>
---------	----------------

Description

Requests to various WFS API.

Usage

```
wfs_api(base.url = NULL, queries)
```

Arguments

base.url	WFS url, for example "https://kartta.hsy.fi/geoserver/wfs"
queries	List of query parameters

Details

Make a request to the specific WFS API. The base url is `https://kartta.hsy.fi/geoserver/wfs` to which other components defined by the arguments are appended.

This is a low-level function intended to be used by other higher level functions in the package.

Note that GET requests are used using 'httpcache' meaning that requests are cached. If you want clear cache, use `[httpcache::clearCache()]`. To turn the cache off completely, use `[httpcache::cacheOff()]`

Value

wfs_api (S3) object with the following attributes:

content XML payload.

path path provided to get the response.

response the original response object.

Author(s)

Joona Lehtomäki <joona.lehtomaki@iki.fi>, Kostas Vasilopoulos, Pyry Kantanen

Source

Gracefully failing HTTP request code (slightly adapted by Pyry Kantanen) from RStudio community member kvasilopoulos. Many thanks!

Source of the original RStudio community discussion: <https://community.rstudio.com/t/internet-resources-should-fail-gracefully/49199>

Examples

```
wfs_api(base.url = "https://kartta.hsy.fi/geoserver/wfs",
        queries = append(list("service" = "WFS", "version" = "1.0.0"),
                          list(request = "getFeature",
                                layer = "tilastointialueet:kunta4500k_2017")))
```

Index

* **package**

helsinki-package, [2](#)

* **utilities**

to_sf, [12](#)

get_city_map, [3](#)

get_feature, [4](#), [5](#), [11](#)

get_feature_list, [4](#), [4](#), [11](#)

get_hri_stats, [5](#)

get_linkedevents, [6](#)

get_rakennustietoruudukko, [7](#)

get_servicemap, [8](#)

get_vaestotietoruudukko, [10](#)

helsinki (helsinki-package), [2](#)

helsinki-package, [2](#)

select_feature, [4](#), [5](#), [11](#)

to_sf, [12](#)

wfs_api, [12](#)