Package: kwb.fisbroker (via r-universe)

October 17, 2024

Title R Package for Getting Spatial Data from FIS-Broker

Version 0.0.0.9000

Description R Package for getting spatial data from FIS-Broker (https://www.stadtentwicklung.berlin.de/geoinformation/fis-broker/).

License MIT + file LICENSE

URL https://github.com/KWB-R/kwb.fisbroker

BugReports https://github.com/KWB-R/kwb.fisbroker/issues

Depends R (>= 2.10)

Imports dplyr, fs, httr, jsonlite, kwb.utils, magrittr, readr, rlang, rvest, sf, tibble, tidyr, xml2

Suggests covr, DT, ggplot2, knitr, leaflet, rmarkdown

VignetteBuilder knitr

Remotes github::kwb-r/kwb.utils

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.1

Repository https://kwb-r.r-universe.dev

RemoteUrl https://github.com/KWB-R/kwb.fisbroker

RemoteRef HEAD

RemoteSha 853cbcd475cf005ae52ab9921dceaab493419e7f

Contents

`compose_fis_broker_url` *Compose FIS-Broker URL*

Description

Compose FIS-Broker URL

Usage

```
compose_fis_broker_url(
  cmd = "user_login",
  session_id = NULL,
  type = NULL,
  id = NULL
)
```

Arguments

`cmd`	command (default: "user login")
`session_id`	as retrieved by `login_to_fis_broker`, (default: NULL)
`type`	dataset type, i.e. "ATOM", "WFS" or "WMS" (default: NULL)
`id`	dataset id (default: NULL)

Value

composed FIS-Broker URL

`create_info_page_hrefs` *Create Full URLs to Info Pages about Datasets in Overview Table*

Description

Create Full URLs to Info Pages about Datasets in Overview Table

Usage

```
create_info_page_hrefs(overview, dbg = TRUE)
```

Arguments

`overview`	overview tibble as retrieved by `get_dataset_overview`, (default: `get_dataset_overview`)
`dbg`	whether or not to show debug messages (default: TRUE)

Value

vector with hrefs

`get_dataset_overview` *Get Dataset Overview*

Description

Get Dataset Overview

Usage

```
get_dataset_overview(dbg = TRUE)
```

Arguments

`dbg`	whether or not to show debug messages

Value

tibble with 7 columns and rows equal to number of datasets (one for each dataset type, e.g WMS, WFS, ATOM)

category_id Category id of dataset (artifically generated)

category_name Category name of dataset)

dataset_id Dataset id, artifically generated!!!

dataset_name Dataset name as listed on table

dataset_name_href Link of dataset name. JS session Id was replaced with a placeholder

dataset_type Dataset type as listed on table one of `ATOM`, `WFS` or `WMS`

dataset_type_href Link of dataset type. JS session Id was replaced with a placeholder

Examples

```
fb_dataset_overview <- kwb.fisbroker::get_dataset_overview()
# View(fb_dataset_overview)
```

`get_urls` *Get Urls*

Description

Get Urls

Usage

```
get_urls(..., key. = NULL)
```

Arguments

`...`	further arguments passed to `resolve`
`key.`	optional. Name of element to be returned from the URL dictionary. If `NULL` (the default), the whole dictionary (a list) is returned.

Value

return `base` and `wfs` urls of FIS-Broker

Examples

```
# List of all defined (partial) URls
get_urls()

# One specific URL with placeholders replaced as given
get_urls(
  key. = "href_type", sid = "my-session-id", type = "WFS", id = "my-id"
)
```

`httr_get_or_fail` *GET Request with Check for Error*

Description

The function stops with error if the GET request returns status != 200

Usage

```
httr_get_or_fail(url, handle = NULL)
```

Arguments

url	URL to which to send a GET request
handle	passed to GET

Value

If the status was not 200, the function returns what GET returned

---

login_to_fis_broker *Login to FIS-Broker*

Description

Login to FIS-Broker

Usage

```
login_to_fis_broker(dbg = TRUE)
```

Arguments

dbg	print debug messages (default: TRUE)

Value

session id

Examples

```
login_to_fis_broker()
```

---

lookup_dataset_id *Lookup Dataset ID Required for Download*

Description

Lookup Dataset ID Required for Download

Usage

```
lookup_dataset_id(overview, dataset_id, type)
```

Arguments

overview	overview tibble as retrieved by get_dataset_overview
dataset_id	one dataset_id from column "dataset_id" of provided overview
type	dataset type, i.e. "ATOM", "WFS" or "WMS"

Value

dataset id required for read_wfs

---

read_all_metadata *Read Metadata based on Overview on FIS Broker Datasets*

Description

Read Metadata based on Overview on FIS Broker Datasets

Usage

```
read_all_metadata(
  overview = get_dataset_overview(),
  preserve_handle = TRUE,
  dbg = TRUE
)
```

Arguments

overview	overview tibble as retrieved by get_dataset_overview, (default: get_dataset_overview)
preserve_handle	logical. If TRUE (the default is FALSE), the Curl handle is created in advance and reused for all GET requests.
dbg	whether or not to show debug messages (default: TRUE)

Value

tibble with all metdata information

Examples

```
## Not run:
overview <- kwb.fisbroker::get_dataset_overview()
overview_atom <- overview[overview$type == "ATOM",]
metadata_atom <- kwb.fisbroker::read_all_metadata(overview_atom)
overview_wfs <- overview[overview$type == "WFS",]
metadata_wfs <- kwb.fisbroker::read_all_metadata(overview_wfs)
overview_wms <- overview[overview$type == "WMS",]
metadata_wms <- kwb.fisbroker::read_all_metadata(overview_wms)

## End(Not run)
```

`read_metadata` *Read metadata*

Description

Read metadata

Usage

```
read_metadata(
  dataset_id = "s_wfs_alkis_bezirk",
  service_type = "WFS",
  encoding = "Windows-1252",
  dbg = TRUE,
  url = NULL,
  handle = NULL
)
```

Arguments

`dataset_id`	id of wfs dataset (default: "s_wfs_alkis_bezirk"). Needs to checked out manually as described below: 1. Go to: https://fbinter.stadt-berlin.de/fb/ 2. Click on a "WFS" or "WMS" Dataset 3. Read the content of `Rechneraddresse` in the opened tab. The basename of the url is the required dataset id!
`service_type`	either "WFS" or "WMS" (default: "WFS")
`encoding`	encoding of metadata FIS-Broker site (default: "Windows-1252")
`dbg`	print debug messages (default: TRUE)
`url`	optional. URL to web resource from which to load metadata. by default generated by the function
`handle`	passed to `GET`

Value

tibble with metadata for provided dataset_id

See Also

https://fbinter.stadt-berlin.de/fb/berlin/service_intern.jsp?id=s_wfs_alkis_bezirk@senstadt&type=WFS

Examples

```
### One Dataset
berlin_bezirke_metadata <- read_metadata(dataset_id = "s_wfs_alkis_bezirk")
berlin_bezirke_metadata

### Multiple
ids_wfs <- readLines(system.file("extdata/ids_wfs.txt",
package = "kwb.fisbroker"))

wfs_meta_list <- setNames(lapply(ids_wfs, function(id) {
kwb.fisbroker::read_metadata(id)}),
ids_wfs)

wfs_meta <- dplyr::bind_rows(wfs_meta_list, .id = "id_wfs")

wfs_meta
```

---

read_wfs *Read WFS Dataset from FIS-Broker*

Description

Read WFS Dataset from FIS-Broker

Usage

```
read_wfs(
  dataset_id = "s_wfs_alkis_bezirk",
  service_version = "2.0.0",
  srs = "EPSG:25833",
  encoding = "UTF-8",
  dbg = TRUE
)
```

Arguments

dataset_id	id of wfs dataset (default: "s_wfs_alkis_bezirk"). Needs to checked out manually as described below: 1. Go to: https://fbinter.stadt-berlin.de/fb/ 2. Click on a WFS Dataset 3. Read the content of Rechneraddresse in the opened tab. The basename of the url is the required dataset id!
service_version	one of "1.0.0", "1.1.0" or "2.0.0" (default: "2.0.0")
srs	one of "EPSG:4258" or "EPSG:25833" (default: "EPSG:25833")
encoding	default: UTF-8
dbg	prints debug messages if TRUE (default: TRUE)

Value

imports selected WFS dataset into R

See Also

https://fbinter.stadt-berlin.de/fb/berlin/service_intern.jsp?id=s_wfs_alkis_bezirk@senstadt&type=WFS

Examples

```
berlin_bezirke <- kwb.fisbroker::read_wfs(dataset_id = "s_wfs_alkis_bezirk")
```

---

write_to *Write to JSON or CSV*

Description

Write to JSON or CSV

Usage

```
write_to(format, df)
```

Arguments

format	either 'json' or 'csv'
df	data frame

Value

writes provided df to working directory with same name

Index