Package: bodsr (via r-universe)

October 17, 2024

Type Package

Title Call the Bus Open Data Service ('BODS') API Through R

Version 0.1.0

Description A wrapper to allow users to download Bus Open Data Service 'BODS' transport information from the API (<https://www.bus-data.dft.gov.uk/>). This includes timetable and fare metadata (including links for full datasets), timetable data at line level, and real-time location data.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.1

Imports httr, jsonlite, xml2, dplyr, purrr, tibble, rlang

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Francesca Bryden [aut, cre]

Maintainer Francesca Bryden <francesca.bryden@dft.gov.uk>

Date/Publication 2023-02-11 15:20:09 UTC

Repository https://dipad-fran-bryden.r-universe.dev

RemoteUrl https://github.com/cran/bodsr

RemoteRef HEAD

RemoteSha b9a110ae5d2803eeee89377b8cce713ca97e0393

Contents

`count_nodes` *Search an xml file for a specific named node and count the number of instances*

Description

Search an xml file for a specific named node and count the number of instances

Usage

```
count_nodes(x, xpath)
```

Arguments

`x`	An xml object
`xpath`	string. The node name to search for within the xpath.

Value

Returns a numeric count value

`extract_line_level_data` *Open data from a single line metadata table where it's zip or xml format*

Description

Open data from a single line metadata table where it's zip or xml format

Usage

```
extract_line_level_data(file)
```

Arguments

file	A single row of table metadata extracted using get_timetable_metadata()

Value

returns a dataframe of information extracted from the given xml or zip url

---

find_node_value *Search an xml file for a specific named mode and return the value(s) stored in it*

Description

Search an xml file for a specific named mode and return the value(s) stored in it

Usage

```
find_node_value(x, xpath)
```

Arguments

x	An xml object
xpath	string. The node name to search for within the xpath.

Value

Returns a single string of values from the specified node

---

get_fares_metadata *Return fares metadata from the 'BODS' API*

Description

Return fares metadata from the 'BODS' API

Usage

```
get_fares_metadata(
  api_key = Sys.getenv("BODS_KEY"),
  limit = 25,
  noc = NULL,
  status = NULL,
  bounding_box = NULL
)
```

Arguments

api_key	API key for the 'BODS' dataset passed as a string. Can be obtained from the 'BODS' API login
limit	integer. Maximum number of records to return for a query. Defaults to 25
noc	string or vector of strings. Limit results to fares data sets for specified National Operator Codes. A full lookup of NOC to bus operator names can be seen using noc_lookup(). Defaults to NULL.
status	string. Limit results to fares data sets for specified status, accepted values are "published" or "inactive". Defaults to NULL.
bounding_box	vector of four numerics. Limit results to fares data sets that contain information for the area within the rectangular boundingBox you set using co-ordinates [minLatitude, maxLatitude, minLongitude, maxLongitude]. Defaults to NULL.

Value

Returns a data frame of fares metadata including links to data from the 'BODS' API.

Examples

```
## Not run:
#Before running these examples, ensure you have an API key saved
#Return the first 25 results of fares metadata with no filters
get_fares_metadata()

#Return fares metadata for National Express
get_fares_metadata(noc = "NATX")

#Return only published fares metadata for Go Ahead
get_fares_metadata(noc = "BHBC", status = "published")

#Return fares metadata for the specified bounding box
get_fares_metadata(bounding_box = c(51.401, 51.509, 0.01, 0.201))

## End(Not run)
```

---

get_location_gtfs *Return GTFS-RT location data from the 'BODS' API*

Description

Return GTFS-RT location data from the 'BODS' API

Usage

```
get_location_gtfs(
  api_key = Sys.getenv("BODS_KEY"),
  bounding_box = NULL,
  route_id = NULL,
  start_time_after = NULL,
  start_time_before = NULL
)
```

Arguments

api_key	API key for the 'BODS' dataset passed as a string. Can be obtained from the 'BODS' API login
bounding_box	vector of four numerics. Limit results to location data for vehicles within the rectangular boundingBox you set using co-ordinates [minLatitude, maxLatitude, minLongitude, maxLongitude]. Defaults to NULL.
route_id	string or vector of strings. Limit results to bus location data with the specified routeId. Defaults to NULL.
start_time_after	integer. Limit results to bus location data with a start time after the specified Unix timestamp. Defaults to NULL.
start_time_before	integer. Limit results to bus location data with a start time before the specified Unix timestamp. Defaults to NULL.

Value

Returns bus location data in GTFS-RT format. More detail on this format can be found the 'BODS' data formats documentation

Examples

```
## Not run:
#Before running these examples, ensure you have an API key saved


#Return data for specified route ID
get_location_gtfs(route_id = "45")

#Return data within a specified bounding box
get_location_gtfs(bounding_box = c(51.401, 51.509, 0.01, 0.201))


## End(Not run)
```

`get_location_xml` *Return XML vehicle location data from the 'BODS' API*

Description

Return XML vehicle location data from the 'BODS' API

Usage

```
get_location_xml(
  api_key = Sys.getenv("BODS_KEY"),
  bounding_box = NULL,
  noc = NULL,
  vehicle_ref = NULL,
  line_ref = NULL,
  producer_ref = NULL,
  origin_ref = NULL,
  destination_ref = NULL
)
```

Arguments

`api_key`	API key for the 'BODS' dataset passed as a string. Can be obtained from the 'BODS' API login
`bounding_box`	vector of four numerics. Limit results to location data for vehicles within the rectangular boundingBox you set using co-ordinates [minLatitude, maxLatitude, minLongitude, maxLongitude]. Defaults to NULL.
`noc`	string or vector of strings. Limit results to fares data sets for specified National Operator Codes. A full lookup of NOC to bus operator names can be seen using noc_lookup(). Defaults to NULL.
`vehicle_ref`	string. Limit results to bus location data with the specified vehicle_ref. This is a unique reference for the vehicle that is consistent and is generated by the vehicle equipment. Defaults to NULL.
`line_ref`	string. Limit results to bus location data with the specified line_ref. Defaults to NULL.
`producer_ref`	string. Limit results to bus location data with the specified producer_ref. Defaults to NULL.
`origin_ref`	string. Limit results to bus location data with the specified origin reference. Accepts any National Public Transport Access Nodes (NaPTAN) value, which can be found the NaPTAN access nodes dataset. Defaults to NULL.
`destination_ref`	string. Limit results to bus location data with the specified destination reference. Accepts any National Public Transport Access Nodes (NaPTAN) value, which can be found the NaPTAN access nodes dataset. Defaults to NULL.

Value

Returns bus location data in XML SIRI-VM format. More detail on this format can be found the 'BODS' data formats documentation

Examples

```
## Not run:
#Before running these examples, ensure you have an API key saved

##Return unfiltered data from XML API
get_location_xml()

#Return data for vehicle reference "BUSC" only
get_location_xml(vehicle_ref = "BUSC")

#Return data for specified origin
get_location_xml(origin_ref = "21024515")


## End(Not run)
```

`get_timetable_data` *Extract line-level timetable data from all rows of the provided metadata table*

Description

Extract line-level timetable data from all rows of the provided metadata table

Usage

```
get_timetable_data(timetable_metadata)
```

Arguments

`timetable_metadata`
: A single row of table metadata extracted using get_timetable_metadata()

Value

returns list of timetable dataframes, with each dataframe corresponding to a row on the provided timetable metadata

Examples

```
## Not run:
#Before running these examples, ensure you have an API key saved
#Return the first 5 results of timetable metadata with no filters
metadata <- get_timetable_metadata(limit = 5)



## End(Not run)
```

`get_timetable_metadata` *Return timetable metadata from the 'BODS' API*

Description

Return timetable metadata from the 'BODS' API

Usage

```
get_timetable_metadata(
  api_key = Sys.getenv("BODS_KEY"),
  limit = 25,
  search = NULL,
  noc = NULL,
  admin_area = NULL,
  status = NULL,
  end_date_start = NULL,
  end_date_end = NULL,
  modified_date = NULL,
  start_date_start = NULL,
  start_date_end = NULL,
  dq_rag = NULL,
  bods_compliance = NULL
)
```

Arguments

`api_key`	API key for the 'BODS' dataset passed as a string. Can be obtained from the BODS API login
`limit`	integer. Maximum number of records to return for a query. Defaults to 25
`search`	string to search records on; can be a value or partial value to match the data set name, data set description, organisation name, or admin area name. Defaults to NULL.

noc	string or vector of strings. Limit results to fares data sets for specified National Operator Codes. A full lookup of NOC to bus operator names can be seen using noc_lookup(). Defaults to NULL.
admin_area	string or vector of strings. Limit results to datasets with services that stop within the specified area(s). 'ATCO' Area Codes are as specified in the NPTG area codes Defaults to NULL.
status	string. Limit results to data sets with the specified status, accepted values are "published" or "inactive". Defaults to NULL.
end_date_start	datetime. Limit results to data sets with services with end dates after this date. Defaults to NULL.
end_date_end	datetime. Limit results to data sets with services with end dates before this date. Defaults to NULL.
modified_date	datetime. Limit results to data sets that have been created or updated since the specified date. Defaults to NULL.
start_date_start	datetime. Limit results to data sets with services with start dates after this date. Defaults to NULL.
start_date_end	datetime. Limit results to data sets with services with start dates before this date. Defaults to NULL.
dq_rag	string. Limit results to data sets with the specified RAG status. Accepted options are "red", "amber" and "green". Defaults to NULL.
bods_compliance	logical. Limit results to datasets with the specified BODS compliance status. Defaults to NULL.

Value

Returns a data frame of timetable metadata including links to data from the 'BODS' API.

Examples

```
## Not run:
#Before running these examples, ensure you have an API key saved
#Return the first 25 results of timetable metadata with no filters
get_timetable_metadata()

#Return timetable metadata for National Express
get_timetable_metadata(noc = "NATX")

#Return only published timetable metadata for Go Ahead with a green RAG status
get_timetable_metadata(noc = "BHBC", status = "published", dq_rag = "green")

#Return timetable metadata for the Devon admin area and search string
get_timetable_metadata(admin_area = "110", search = "Plymouth Citybus")

## End(Not run)
```

line_level_xml *Pull a table of relevant values from specified nodes in the xml*

Description

Pull a table of relevant values from specified nodes in the xml

Usage

```
line_level_xml(x, count = 1, total_count = 1)
```

Arguments

x	An xml object
count	numeric. Where the xml file is taken from a zip collection, the number file it is. Defaults to 1.
total_count	numeric. Where the xml file is taken from a zip collection, the total number of files in the zip. Defaults to 1.

Value

Returns a table of values extracted from specified nodes of an xml document

noc_lookup *Lookup between operator names and national operator code ('NOC') lookup*

Description

Lookup between operator names and national operator code ('NOC') lookup

Usage

```
noc_lookup()
```

Value

Returns a data frame of operator names and their corresponding national operator code from the 'BODS' API.

Examples

```
##Check operator name lookup
## Not run:
noc_lookup()

## End(Not run)
```

not_null — *Join together a value and an associated API string if the value is not NULL*

Description

Join together a value and an associated API string if the value is not NULL

Usage

```
not_null(obj, string)
```

Arguments

obj	R object to check whether it is NULL or not
string	API string to append to R object

not_null_date — *Join together a date value and an associated API string if the value is not NULL*

Description

Join together a date value and an associated API string if the value is not NULL

Usage

```
not_null_date(date, string)
```

Arguments

date	R object to check whether it is NULL or not
string	API string to append to R object

`open_all_xml` *Open every xml file within a zip object and extract data of interest from it using a given function*

Description

Open every xml file within a zip object and extract data of interest from it using a given function

Usage

```
open_all_xml(url, fun)
```

Arguments

`url`	A url pointing towards a zip object
`fun`	name of a data extracting function to apply to the zip folder

Value

returns a dataframe of information extracted from xml documents

`poss_xml` *Try to read an xml file using read_xml; where this fails, quietly return a NULL value*

Description

Try to read an xml file using read_xml; where this fails, quietly return a NULL value

Usage

```
poss_xml(...)
```

Arguments

`...`	arguments to pass to the read_xml function

`xml_file_counter` *Count the number of xml files included within a provided metadata dataframe, whether the provided file links are xml or zip*

Description

Count the number of xml files included within a provided metadata dataframe, whether the provided file links are xml or zip

Usage

```
xml_file_counter(timetable_metadata)
```

Arguments

`timetable_metadata`
: A table of metadata extracted using get_timetable_metadata()

Value

returns a numeric vector of number of xml files by row of the provided metadata

Index