Package 'gbfs'

April 3, 2021

Type Package

Title Interface with Live Bikeshare Data

Version 1.3.7

Description Supplies a set of functions to interface with bikeshare data following the General Bikeshare Feed Specification, allowing users to query and accumulate tidy datasets for specified cities/bikeshare programs.

License CC0

Imports dplyr, readr, stringr, jsonlite, lubridate, httr, purrr, curl

Encoding UTF-8

RoxygenNote 7.1.1

Suggests knitr, rmarkdown, testthat (>= 2.1.0), covr

URL https://github.com/simonpcouch/gbfs

BugReports https://github.com/simonpcouch/gbfs/issues

NeedsCompilation no

Author Simon P. Couch [aut, cre],
Kaelyn Rosenberg [aut],
Mark Padgham [ctb]

Maintainer Simon P. Couch <simonpatrickcouch@gmail.com>

Repository CRAN

Date/Publication 2021-04-03 04:50:02 UTC

R topics documented:

`gbfs` *Package: gbfs*

Description

The `gbfs` package allows users to query tidy datasets about bikeshare programs around the world by supplying a set of functions to interface with .json feeds following the General Bikeshare Feed Specification, a standard data release format developed by the North American Bikeshare Association.

Details

The main function exported by this package is `get_gbfs()`, which grabs every feed released by a city. Alternatively, the user can just grab information on specific feeds (or groups of feeds).

Each of the feeds described below can be queried with the `get_suffix()` function, where `suffix` is replaced with the name of the relevant feed.

Although all of the feeds are livestreamed, only a few of the datasets change often:

- `station_status`: Supplies the number of available bikes and docks at each station as well as station availability
- `free_bike_status`: Gives the coordinates and metadata on available bikes that are parked, but not at a station.

In this package, these two datasets are considered "dynamic", and can be specified as desired datasets by setting `'feeds = "dynamic"` in the main wrapper function in the package, `get_gbfs`.

Much of the data supplied in this specification can be considered static. If you want to grab all of these for a given city, set `feeds = "static"` when calling `get_gbfs`. Static feeds include:

- `system_information`: Basic metadata about the bikeshare program
- `station_information`: Information on the capacity and coordinates of stations
- Several optional feeds: `system_hours`, `system_calendar`, `system_regions`, `system_pricing_plans`, and `system_alerts`

Author(s)

Maintainer: Simon P. Couch <simonpatrickcouch@gmail.com>

Authors:

- Kaelyn Rosenberg <kaerosenberg@gmail.com>

Other contributors:

- Mark Padgham <mark.padgham@email.com> [contributor]

See Also

Useful links:

- https://github.com/simonpcouch/gbfs
- Report bugs at https://github.com/simonpcouch/gbfs/issues

`get_free_bike_status` *Grab the free_bike_status feed.*

Description

Grab a dataframe giving the geographic location and other metadata of bikeshare bikes not parked at bikeshare stations. Metadata for this dataset can be found at: https://github.com/NABSA/gbfs/blob/master/gbfs.md

Usage

```
get_free_bike_status(
  city,
  directory = NULL,
  file = "free_bike_status.rds",
  output = NULL
)
```

Arguments

`city`	A character string that can be matched to a gbfs feed. The recommended argument is a system ID supplied in the output of [get_gbfs_cities()], but will also attempt to match to the URL of an active .json feed or city name.
`directory`	Optional. Path to a folder (or folder to be created) where the feed will be saved.
`file`	Optional. The name of the file to be saved (if `output` is set to `"save"` or `"both"`), as a character string. Must end in `".rds"`.
`output`	Optional. The type of output method. If left as default, this argument is inferred from the `directory` argument. If `output = "save"`, the object will be saved as an .rds object at # the given path. If `output = "return"`, the output will be returned as a dataframe object. Setting `output = "both"` will do both.

Value

The output of this function depends on the argument to `output` and `directory`. Either a saved .rds object generated from the current station_information feed, a dataframe object, or both. If a saved feed of the same type already exists at the filepath, the feed will be appended to rather than overwritten.

See Also

[get_gbfs()] for a wrapper to call each of the `get_feed` functions, [get_gbfs_cities()] for a dataframe of cities releasing gbfs functions, and [get_which_gbfs_feeds()] for a dataframe of which feeds are released by a given city.

Examples

```
# grab the free bike status feed for portland, oregon's bikeshare program
get_free_bike_status(city =
"https://gbfs.biketownpdx.com/gbfs/en/free_bike_status.json",
output = "return")
```

`get_gbfs` *Grab bikeshare data*

Description

`get_gbfs` grabs bikeshare data supplied in the General Bikeshare Feed Specification format for a given city. By default, the function returns the results as a named list of dataframes, but to make accumulation of datasets over time straightforward, the user can also save the results as .Rds files that will be automatically row-binded. Metadata for each dataset can be found at: https://github.com/NABSA/gbfs/blob/master/gbfs.md

Usage

```
get_gbfs(city, feeds = "all", directory = NULL, output = NULL)
```

Arguments

`city`	A character string that can be matched to a city or a url to an active gbfs.json feed. See `get_gbfs_cities` for a current list of available cities.
`feeds`	Optional. A character string specifying which feeds should be saved. Options are `"all"`, `"static"`, and `"dynamic"`.
`directory`	Optional. Path to a folder (or folder to be created) where the feed will will be saved.
`output`	Optional. The type of output method. By default, output method will be inferred from the `directory` argument. If `output = "save"`, the dataframes will be saved as .rds objects in the given folder. If `output = "return"`, the results will be returned as a named list of dataframes. Setting `output = "both"` will do both. If both are left as NULL, the result will be returned and not saved to file.

Value

The output of this function depends on the arguments supplied to `output` and `directory`. Either a folder of .rds dataframes saved at the given path, a returned named list of dataframes, or both. The function will raise an error if the `directory` and `output` arguments seem to conflict.

Examples

```
# grab all of the feeds released by portland's
# bikeshare program and return them as a
# named list of dataframes
get_gbfs(city = "biketown_pdx")

# if, rather than returning the data, we wanted to save it:
get_gbfs(city = "biketown_pdx", directory = tempdir())

# note that, usually, we'd supply a character string
# (like "pdx", maybe,) to the directory argument
# instead of `tempdir()`.

# if we're having trouble specifying the correct feed,
# we can also supply the actual URL to the feed
get_gbfs(city = "https://gbfs.biketownpdx.com/gbfs/gbfs.json")

# the examples above grab every feed that portland releases.
# if, instead, we just wanted the dynamic feeds
get_gbfs(city = "biketown_pdx", feeds = "dynamic")
```

---

get_gbfs_cities *Get table of all cities releasing GBFS feeds*

Description

Get table of all cities releasing GBFS feeds

Usage

```
get_gbfs_cities()
```

Value

A `data.frame` of all cities issuing GBFS feeds. The 'Auto-Discovery URL' column supplies the relevant .json feeds, while the entries in the 'URL' column take the user to the public-facing webpage of the programs.

Source

North American Bikeshare Association, General Bikeshare Feed Specification https://raw.githubusercontent.com/NABSA/gbfs/master/systems.csv

`get_station_information` *Grab the station_information feed.*

Description

`get_station_information` grabs and tidies the station_information feed for a given city. This dataset contains locations, capacity, and other information about bikeshare stations. Metadata for this dataset can be found at: https://github.com/NABSA/gbfs/blob/master/gbfs.md

Usage

```
get_station_information(
  city,
  directory = NULL,
  file = "station_information.rds",
  output = NULL
)
```

Arguments

`city`	A character string that can be matched to a gbfs feed. The recommended argument is a system ID supplied in the output of [get_gbfs_cities()], but will also attempt to match to the URL of an active .json feed or city name.
`directory`	Optional. Path to a folder (or folder to be created) where the feed will be saved.
`file`	Optional. The name of the file to be saved (if `output` is set to `"save"` or `"both"`), as a character string. Must end in `".rds"`.
`output`	Optional. The type of output method. If left as default, this argument is inferred from the `directory` argument. If `output = "save"`, the object will be saved as an .rds object at # the given path. If `output = "return"`, the output will be returned as a dataframe object. Setting `output = "both"` will do both.

Value

The output of this function depends on argument to `output` and `directory`. Either a saved .rds object generated from the current feed, a dataframe object, or both.

See Also

[get_gbfs()] for a wrapper to call each of the `get_feed` functions, [get_gbfs_cities()] for a dataframe of cities releasing gbfs functions, and [get_which_gbfs_feeds()] for a dataframe of which feeds are released by a given city.

Examples

```
# grab the free bike status feed for portland, oreoon's bikeshare program
get_station_information(city =
"https://gbfs.biketownpdx.com/gbfs/en/station_information.json",
                        output = "return")
```

`get_station_status` *Grab the station_status feed.*

Description

Grab a dataframe giving the geographic location and other metadata of bikeshare bikes parked at bikeshare stations. Metadata for this dataset can be found at: https://github.com/NABSA/gbfs/blob/master/gbfs.md

Usage

```
get_station_status(
  city,
  directory = NULL,
  file = "station_status.rds",
  output = NULL
)
```

Arguments

`city`	A character string that can be matched to a gbfs feed. The recommended argument is a system ID supplied in the output of [get_gbfs_cities()], but will also attempt to match to the URL of an active .json feed or city name.
`directory`	Optional. Path to a folder (or folder to be created) where the feed will be saved.
`file`	Optional. The name of the file to be saved (if `output` is set to `"save"` or `"both"`), as a character string. Must end in `".rds"`.
`output`	Optional. The type of output method. If left as default, this argument is inferred from the `directory` argument. If `output = "save"`, the object will be saved as an .rds object at # the given path. If `output = "return"`, the output will be returned as a dataframe object. Setting `output = "both"` will do both.

Value

The output of this function depends on the argument to `output` and `directory`. Either a saved .rds object generated from the current station_information feed, a dataframe object, or both. If a saved feed of the same type already exists at the filepath, the feed will be appended to rather than overwritten.

See Also

[get_gbfs()] for a wrapper to call each of the `get_feed` functions, [get_gbfs_cities()] for a dataframe of cities releasing gbfs functions, and [get_which_gbfs_feeds()] for a dataframe of which feeds are released by a given city.

Examples

```
# we can grab the free bike status feed for portland,
# oregon's bikeshare program in several ways! the most
# straightforward way is just to supply the `city` argument
# as a string:
get_station_status(city = "biketown_pdx")

# the `city` argument can also be supplied as an
# actual URL to an active .json feed
get_station_status(city =
"https://gbfs.biketownpdx.com/gbfs/en/station_status.json")
```

---

get_system_alerts *Grab the system_alerts feed.*

Description

`get_system_alerts` grabs and tidies the system_alerts feed for a given city. This feed informs users about changes to normal operation. Metadata for this dataset can be found at: https://github.com/NABSA/gbfs/blob/master/gbfs.md

Usage

```
get_system_alerts(
  city,
  directory = NULL,
  file = "system_alerts.rds",
  output = NULL
)
```

Arguments

`city`	A character string that can be matched to a gbfs feed. The recommended argument is a system ID supplied in the output of [get_gbfs_cities()], but will also attempt to match to the URL of an active .json feed or city name.
`directory`	Optional. Path to a folder (or folder to be created) where the feed will be saved.
`file`	Optional. The name of the file to be saved (if `output` is set to `"save"` or `"both"`), as a character string. Must end in `".rds"`.
`output`	Optional. The type of output method. If left as default, this argument is inferred from the `directory` argument. If `output = "save"`, the object will be saved as an .rds object at # the given path. If `output = "return"`, the output will be returned as a dataframe object. Setting `output = "both"` will do both.

Value

The output of this function depends on argument to `output` and `directory`. Either a saved .rds object generated from the current feed, a dataframe object, or both.

See Also

[get_gbfs()] for a wrapper to call each of the `get_feed` functions, [get_gbfs_cities()] for a dataframe of cities releasing gbfs functions, and [get_which_gbfs_feeds()] for a dataframe of which feeds are released by a given city.

Examples

```
# grab the system alerts feed for portland, oregon
get_system_alerts(city =
"https://gbfs.biketownpdx.com/gbfs/en/system_alerts.json",
                  output = "return")
```

`get_system_calendar` *Grab the system_calendar feed.*

Description

`get_system_calendar` grabs and tidies the system_calendar feed for a given city. Metadata for this dataset can be found at: https://github.com/NABSA/gbfs/blob/master/gbfs.md

Usage

```
get_system_calendar(
  city,
  directory = NULL,
  file = "system_calendar.rds",
  output = NULL
)
```

Arguments

`city`	A character string that can be matched to a gbfs feed. The recommended argument is a system ID supplied in the output of [get_gbfs_cities()], but will also attempt to match to the URL of an active .json feed or city name.
`directory`	Optional. Path to a folder (or folder to be created) where the feed will be saved.
`file`	Optional. The name of the file to be saved (if `output` is set to `"save"` or `"both"`), as a character string. Must end in `".rds"`.
`output`	Optional. The type of output method. If left as default, this argument is inferred from the `directory` argument. If `output = "save"`, the object will be saved as an .rds object at # the given path. If `output = "return"`, the output will be returned as a dataframe object. Setting `output = "both"` will do both.

Value

The output of this function depends on argument to `output` and `directory`. Either a saved .rds object generated from the current feed, a dataframe object, or both.

See Also

[get_gbfs()] for a wrapper to call each of the `get_feed` functions, [get_gbfs_cities()] for a dataframe of cities releasing gbfs functions, and [get_which_gbfs_feeds()] for a dataframe of which feeds are released by a given city.

Examples

```
# grab the system calendar feed for portland, oregon
get_system_calendar(city =
"https://gbfs.biketownpdx.com/gbfs/en/system_calendar.json",
                    output = "return")
```

`get_system_hours` *Grab the system_hours feed.*

Description

`get_system_hours` grabs and tidies the system_hours feed for a given city. Metadata for this dataset can be found at: https://github.com/NABSA/gbfs/blob/master/gbfs.md

Usage

```
get_system_hours(
  city,
  directory = NULL,
  file = "system_hours.rds",
  output = NULL
)
```

Arguments

`city`	A character string that can be matched to a gbfs feed. The recommended argument is a system ID supplied in the output of [get_gbfs_cities()], but will also attempt to match to the URL of an active .json feed or city name.
`directory`	Optional. Path to a folder (or folder to be created) where the feed will be saved.
`file`	Optional. The name of the file to be saved (if `output` is set to `"save"` or `"both"`), as a character string. Must end in `".rds"`.
`output`	Optional. The type of output method. If left as default, this argument is inferred from the `directory` argument. If `output = "save"`, the object will be saved as an .rds object at # the given path. If `output = "return"`, the output will be returned as a dataframe object. Setting `output = "both"` will do both.

Value

The output of this function depends on argument to `output` and `directory`. Either a saved .rds object generated from the current feed, a dataframe object, or both.

See Also

[get_gbfs()] for a wrapper to call each of the `get_feed` functions, [get_gbfs_cities()] for a dataframe of cities releasing gbfs functions, and [get_which_gbfs_feeds()] for a dataframe of which feeds are released by a given city.

Examples

```
# grab the system hours feed for portland, oregon
get_system_hours(city =
"https://gbfs.biketownpdx.com/gbfs/en/system_hours.json",
                 output = "return")
```

---

get_system_information

Grab the system_information feed.

Description

`get_system_information` grabs and tidies the system_information feed for a given city. Metadata for this dataset can be found at: https://github.com/NABSA/gbfs/blob/master/gbfs.md

Usage

```
get_system_information(
  city,
  directory = NULL,
  file = "system_information.rds",
  output = NULL
)
```

Arguments

`city`	A character string that can be matched to a gbfs feed. The recommended argument is a system ID supplied in the output of [get_gbfs_cities()], but will also attempt to match to the URL of an active .json feed or city name.
`directory`	Optional. Path to a folder (or folder to be created) where the feed will be saved.
`file`	Optional. The name of the file to be saved (if `output` is set to `"save"` or `"both"`), as a character string. Must end in `".rds"`.
`output`	Optional. The type of output method. If left as default, this argument is inferred from the `directory` argument. If `output = "save"`, the object will be saved as an .rds object at # the given path. If `output = "return"`, the output will be returned as a dataframe object. Setting `output = "both"` will do both.

Value

The output of this function depends on argument to `output` and `directory`. Either a saved .rds object generated from the current feed, a dataframe object, or both.

See Also

[get_gbfs()] for a wrapper to call each of the `get_feed` functions, [get_gbfs_cities()] for a dataframe of cities releasing gbfs functions, and [get_which_gbfs_feeds()] for a dataframe of which feeds are released by a given city.

Examples

```
# we can grab the free bike status feed for portland,
# oregon's bikeshare program in  several ways! first, supply the `city`
# argument as a URL, and save to file by leaving output
# set to it's default. usually, we would supply a character
# string (like "pdx", maybe,) for the `directory` argument
# instead of `tempdir`.
get_system_information(city =
"https://gbfs.biketownpdx.com/gbfs/en/system_information.json",
                       directory = tempdir())

# or, instead, just supply the name of
# the city as a string and return the output as a dataframe
get_system_information(city = "biketown_pdx",
                       output = "return")
```

`get_system_pricing_plans` *Grab the system_pricing_plans feed.*

Description

`get_system_pricing_plans` grabs and tidies the system_pricing_plans feed for a given city. Metadata for this dataset can be found at: https://github.com/NABSA/gbfs/blob/master/gbfs.md

Usage

```
get_system_pricing_plans(
  city,
  directory = NULL,
  file = "system_pricing_plans.rds",
  output = NULL
)
```

Arguments

city	A character string that can be matched to a gbfs feed. The recommended argument is a system ID supplied in the output of [get_gbfs_cities()], but will also attempt to match to the URL of an active .json feed or city name.
directory	Optional. Path to a folder (or folder to be created) where the feed will be saved.
file	Optional. The name of the file to be saved (if output is set to "save" or "both"), as a character string. Must end in ".rds".
output	Optional. The type of output method. If left as default, this argument is inferred from the directory argument. If output = "save", the object will be saved as an .rds object at # the given path. If output = "return", the output will be returned as a dataframe object. Setting output = "both" will do both.

Value

The output of this function depends on argument to output and directory. Either a saved .rds object generated from the current feed, a dataframe object, or both.

See Also

[get_gbfs()] for a wrapper to call each of the get_feed functions, [get_gbfs_cities()] for a dataframe of cities releasing gbfs functions, and [get_which_gbfs_feeds()] for a dataframe of which feeds are released by a given city.

---

get_system_regions *Grab the system_regions feed.*

Description

get_system_regions grabs and tidies the system_regions feed for a given city. Metadata for this dataset can be found at: https://github.com/NABSA/gbfs/blob/master/gbfs.md

Usage

```
get_system_regions(
  city,
  directory = NULL,
  file = "system_regions.rds",
  output = NULL
)
```

Arguments

city	A character string that can be matched to a gbfs feed. The recommended argument is a system ID supplied in the output of [get_gbfs_cities()], but will also attempt to match to the URL of an active .json feed or city name.
directory	Optional. Path to a folder (or folder to be created) where the feed will be saved.

file	Optional. The name of the file to be saved (if output is set to "save" or "both"), as a character string. Must end in ".rds".
output	Optional. The type of output method. If left as default, this argument is inferred from the directory argument. If output = "save", the object will be saved as an .rds object at # the given path. If output = "return", the output will be returned as a dataframe object. Setting output = "both" will do both.

Value

The output of this function depends on argument to output and directory. Either a saved .rds object generated from the current feed, a dataframe object, or both.

See Also

[get_gbfs()] for a wrapper to call each of the get_feed functions, [get_gbfs_cities()] for a dataframe of cities releasing gbfs functions, and [get_which_gbfs_feeds()] for a dataframe of which feeds are released by a given city.

Examples

```
# we can grab the system regions feed for portland,
# oregon in one of several ways! first, supply the `city`
# argument as a URL, and save to file by leaving output
# set to it's default. usually, we would supply a character
# string (like "pdx", maybe,) for the `directory` argument
# instead of `tempdir`.
get_system_regions(city =
"https://gbfs.biketownpdx.com/gbfs/en/system_regions.json",
                   directory = tempdir())

# or, instead, just supply the name of
# the city as a string and return the output
# as a dataframe
get_system_regions(city = "biketown_pdx",
                   output = "return")
```

---

get_which_gbfs_feeds *Get dataframe of bikeshare feeds released by a city*

Description

Of the different types of feeds supplied by the gbfs, some are required, some are conditionally required, and some are optional. This function grabs a list of each of the feeds supplied by a given city, as well as the URLs to access them.

Usage

```
get_which_gbfs_feeds(city)
```

Arguments

city	A character string that can be matched to a gbfs feed. The recommended argument is a system ID supplied in the output of [get_gbfs_cities()], but will also attempt to match to the URL of an active .json feed or city name.

Value

A data.frame containing the feeds supplied by a city. . The ‘feed‘ column supplies the name of the relevant .json feeds, while the entries in the ‘URL‘ column supply the feeds themselves.

Source

North American Bikeshare Association, General Bikeshare Feed Specification https://github.com/NABSA/gbfs/blob/master/gbfs.md

Examples

```
# grab all of the feeds released by portland
get_which_gbfs_feeds(city = "biketown_pdx")
```

Index