Package 'seeclickfixr'

October 14, 2022

Type Package

Title Access Data from the SeeClickFix Web API

Version 1.1.0

BugReports https://github.com/justindbk/seeclickfixr/issues

Depends R (>= 3.1.2)

LazyData true

Description Provides a wrapper to access data from the SeeClickFix web API for R. SeeClickFix is a central platform employed by many cities that allows citizens to request their city's services. This package creates several functions to work with all the built-in calls to the SeeClickFix API. Allows users to download service request data from numerous locations in easy-to-use dataframe format manipulable in standard R functions.

License GPL-3

Imports jsonlite, RCurl

NeedsCompilation no

Author Justin de Benedictis-Kessner [aut, cre], Christian Lemp [ctb]

Maintainer Justin de Benedictis-Kessner <justindbk@gmail.com>

Repository CRAN

Date/Publication 2016-12-07 18:05:33

R topics documented:

`get_city_issues` *Query all issues in a given location*

Description

Returns all issues in the specific location, specified either by coordinates or by name/address.

Usage

```
get_city_issues(city=NULL, lat=NULL,long=NULL, status =
"open,acknowledged,closed,archived", limit = 100)
```

Arguments

`city`	A written description of the location for which issue types should be returned. If city is specified, lat/long should not be.
`lat`	Latitude of coordinates, specified instead of city.
`long`	Longitude of coordinates, specified instead of city.
`status`	Which status types of issues should be returned. Separate statuses should be separated by commas without spaces. Available options are open, acknowledged, closed, and archived. Default is all.
`limit`	Number of items to return. Defaults to 100.

Value

`issue_id`	Unique ID number for the given issue
`status`	Status of the issue (open/closed)
`summary`	Summary of the status for the issue
`description`	Text description of the issue as reported
`rating`	Importance rating of the issue
`lat`	Latitude of the issue as reported
`lng`	Longitude of the issue as reported
`issue_address`	Address of the issue as reported
`created_at`	Date and time when issue report was created
`acknowledged_at`	Date and time when issue report was acknowledged by city
`closed_at`	Date and time when issue report was closed by city
`reopened_at`	Date and time when issue report was reopened, if it was
`updated_at`	Date and time when issue report was last updated
`shortened_url`	Shortened URL of the issue report
`video_url`	URL for the video of the issue, if provided

image_full	Image of the issue as reported
image_square_100x100	Square version of the image of the issue
representative_image_url	A representative image of the issue, if no actual image was submitted
issue_type	Type of issue
url	URL to the report of the issue
html_url	URL to the report of the issue in HTML format
comment_url	URL to the comments on the issue
flag_url	URL to the flag for the issue
close_url	URL to the closing report of the issue
open_url	URL to the opening report of the issue
reporter_id	Issue reporter's unique ID number
reporter_name	Name of the issue reporter
reporter_wittytitle	Username/witty name of the issue reporter
reporter_role	Issue reporter's role in the city, if any
reporter_civicpoints	Number of civic points the issue reporter has, if any
reporter_avatar_full	Chosen avatar of the issue reporter
reporter_avatar_square	Square version of the avatar

See Also

get_specific_issue

`get_issues_by_date` *Query information about issues within a date-time range.*

Description

Returns all issues within a date-time window in the specified location, specified either by coordinates or by name/address.

Usage

```
 get_issues_by_date(city, after =
Sys.time() - 86400, before = Sys.time(), status =
"open,acknowledged,closed,archived", limit = 100)
```

Arguments

city	A written description of the location for which issue types should be returned. If city is specified, lat/long should not be.
after	Beginning of time window from which issues should be returned. Specified in POSIX date-time format. Defaults to 24 hours prior to system time.
before	End of time window from which issues should be returned. Specified in POSIX date-time format. Defaults to system time.
status	Which status types of issues should be returned. Separate statuses should be separated by commas without spaces. Available options are open, acknowledged, closed, and archived. Default is all.
limit	Number of items to return. Defaults to 100.

Value

issue_id	Unique ID number for the given issue
status	Status of the issue (open/acknowledged/closed)
summary	Summary of the status for the issue
description	Text description of the issue as reported
rating	Importance rating of the issue
lat	Latitude of the issue as reported
lng	Longitude of the issue as reported
issue_address	Address of the issue as reported
created_at	Date and time when issue report was created
acknowledged_at	Date and time when issue report was acknowledged by city
closed_at	Date and time when issue report was closed by city
reopened_at	Date and time when issue report was reopened, if it was
updated_at	Date and time when issue report was last updated
shortened_url	Shortened URL of the issue report
video_url	URL for the video of the issue, if provided
image_full	Image of the issue as reported
image_square_100x100	Square version of the image of the issue
representative_image_url	A representative image of the issue, if no actual image was submitted
issue_type	Type of issue
url	URL to the report of the issue
html_url	URL to the report of the issue in HTML format
comment_url	URL to the comments on the issue
flag_url	URL to the flag for the issue

close_url URL to the closing report of the issue

open_url URL to the opening report of the issue

reporter_id Issue reporter's unique ID number

reporter_name Name of the issue reporter

reporter_wittytitle
Username/witty name of the issue reporter

reporter_role Issue reporter's role in the city, if any

reporter_civicpoints
Number of civic points the issue reporter has, if any

reporter_avatar_full
Chosen avatar of the issue reporter

reporter_avatar_square
Square version of the avatar

See Also

get_issues_by_type get_city_issues

---

get_issues_by_type *Query information about issues of a certain type or types.*

Description

Returns all issues of a given type or types in the specified location, specified either by coordinates or by name/address.

Usage

```
 get_issues_by_type(city, issue_type,
status = "open,acknowledged,closed,archived", limit = 100)
```

Arguments

city A written description of the location for which issue types should be returned. If city is specified, lat/long should not be. Make sure to use same spelling/capitalization of city as it appears in the 'url_name' field returned by list_places()

issue_type Type of issues which should be returned. If multiple types, they should be separated by commas and without spaces.

status Which status types of issues should be returned. Separate statuses should be separated by commas without spaces. Available options are open, acknowledged, closed, and archived. Default is all.

limit Number of items to return. Defaults to 100.

Value

issue_id	Unique ID number for the given issue
status	Status of the issue (open/acknowledged/closed)
summary	Summary of the status for the issue
description	Text description of the issue as reported
rating	Importance rating of the issue
lat	Latitude of the issue as reported
lng	Longitude of the issue as reported
issue_address	Address of the issue as reported
created_at	Date and time when issue report was created
acknowledged_at	Date and time when issue report was acknowledged by city
closed_at	Date and time when issue report was closed by city
reopened_at	Date and time when issue report was reopened, if it was
updated_at	Date and time when issue report was last updated
shortened_url	Shortened URL of the issue report
video_url	URL for the video of the issue, if provided
image_full	Image of the issue as reported
image_square_100x100	Square version of the image of the issue
representative_image_url	A representative image of the issue, if no actual image was submitted
issue_type	Type of issue
url	URL to the report of the issue
html_url	URL to the report of the issue in HTML format
comment_url	URL to the comments on the issue
flag_url	URL to the flag for the issue
close_url	URL to the closing report of the issue
open_url	URL to the opening report of the issue
reporter_id	Issue reporter's unique ID number
reporter_name	Name of the issue reporter
reporter_wittytitle	Username/witty name of the issue reporter
reporter_role	Issue reporter's role in the city, if any
reporter_civicpoints	Number of civic points the issue reporter has, if any
reporter_avatar_full	Chosen avatar of the issue reporter
reporter_avatar_square	Square version of the avatar

See Also

list_issue_types get_issues_by_date get_city_issues

---

get_specific_issue *Query information about a given issue*

Description

Returns all information about a specific issue, specified by its ID number.

Usage

```
get_specific_issue(issue_id)
```

Arguments

issue_id	The unique ID number of the requested issue.

Value

issue_id	Unique ID number for the given issue
status	Status of the issue (open/acknowledged/closed)
summary	Summary of the status for the issue
description	Text description of the issue as reported
rating	Importance rating of the issue
lat	Latitude of the issue as reported
lng	Longitude of the issue as reported
issue_address	Address of the issue as reported
created_at	Date and time when issue report was created
acknowledged_at	Date and time when issue report was acknowledged by city
closed_at	Date and time when issue report was closed by city
reopened_at	Date and time when issue report was reopened, if it was
updated_at	Date and time when issue report was last updated
shortened_url	Shortened URL of the issue report
video_url	URL for the video of the issue, if provided
image_full	Image of the issue as reported
image_square_100x100	Square version of the image of the issue
representative_image_url	A representative image of the issue, if no actual image was submitted
issue_type	Type of issue

`url`	URL to the report of the issue
`html_url`	URL to the report of the issue in HTML format
`comment_url`	URL to the comments on the issue
`flag_url`	URL to the flag for the issue
`close_url`	URL to the closing report of the issue
`open_url`	URL to the opening report of the issue
`reporter_id`	Issue reporter's unique ID number
`reporter_name`	Name of the issue reporter
`reporter_wittytitle`	Username/witty name of the issue reporter
`reporter_role`	Issue reporter's role in the city, if any
`reporter_civicpoints`	Number of civic points the issue reporter has, if any
`reporter_avatar_full`	Chosen avatar of the issue reporter
`reporter_avatar_square`	Square version of the avatar

See Also

`get_city_issues`

`list_issue_types` *Query issue types in a given location*

Description

Returns all issue types in use in the specific location, specified either by coordinates or by name/address.

Usage

```
list_issue_types(city = NULL, lat = NULL, long = NULL, limit = 100)
```

Arguments

`city`	A written description of the location for which issue types should be returned. If city is specified, lat/long should not be.
`lat`	Latitude of coordinates, specified instead of city.
`long`	Longitude of coordinates, specified instead of city.
`limit`	Number of items to return. Defaults to 100.

Value

`title`	Category of issue
`organization`	Organization handling this type of requests in this location
`url`	URL to this category of issues
`potential_duplicate_issues_url`	URL to possible duplicates in this category of issues

See Also

`list_places`

`list_places` *Query sub-city neighborhoods in which issues can be reported.*

Description

Returns a list of sub-city neighborhoods where citizens can report issues.

Usage

```
list_places(startingpoint, limit = 100)
```

Arguments

`startingpoint`	Starting point around which the names of sub-city locations will be returned.
`limit`	The maximum number of location names to be returned.

Value

`id`	Unique place id
`name`	Name of sub-city neighborhood/location
`url_name`	URL for
`county`	County where the location is
`state`	State where the location is
`place_type`	Type of location (city, county, neighborhood)
`url`	URL for
`html_url`	
`html_report_url`	
`type`	
`lat`	Latitude of coordinates for location
`lng`	Longitude of coordinates for location

See Also

list_issue_types

Examples

```
list_places("Boston, MA",limit = 5)

## Returns:
# id           name        url_name    county         state   place_type
# 1 28632        Central central_suffolk   Suffolk            MA Neighborhood
# 2 72870       Downtown downtown_boston  Downtown            MA Neighborhood
# 3 72869      North End northend_boston North End            MA Neighborhood
# 4 72838    Beacon Hill     beacon_hill           Massachusetts Neighborhood
# 5 28631 East Cambridge  east-cambridge Middlesex            MA Neighborhood
# url                                html_url
# 1 https://seeclickfix.com/api/v2/places/28632 https://seeclickfix.com/central_suffolk
# 2 https://seeclickfix.com/api/v2/places/72870 https://seeclickfix.com/downtown_boston
# 3 https://seeclickfix.com/api/v2/places/72869 https://seeclickfix.com/northend_boston
# 4 https://seeclickfix.com/api/v2/places/72838     https://seeclickfix.com/beacon_hill
# 5 https://seeclickfix.com/api/v2/places/28631  https://seeclickfix.com/east-cambridge
# html_report_url  type      lat       lng
# 1 https://seeclickfix.com/central_suffolk/report Point 42.35854 -71.05931
# 2 https://seeclickfix.com/downtown_boston/report Point 42.35630 -71.05707
# 3 https://seeclickfix.com/northend_boston/report Point 42.36534 -71.05325
# 4     https://seeclickfix.com/beacon_hill/report Point 42.35716 -71.06791
# 5  https://seeclickfix.com/east-cambridge/report Point 42.36833 -71.07928
```

Index