Package ‘seeclci8kfixr’

October 14, 2022

Type Package
Title Access Data from the SeeClickFix Web API
Version 1.1.0
BugReports https://github.com/justindbk/seeclci8kfixr/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 ....................................................... 2
get_issues_by_date .................................................. 3
get_issues_by_type .................................................. 5
get_specific_issue ................................................... 7
list_issue_types ...................................................... 8
list_places ........................................................... 9

Index 11
get_city_issues  \hspace{1cm} \textbf{Query all issues in a given location}

\textbf{Description}

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

\textbf{Usage}

\begin{verbatim}
get_city_issues(city=NULL, lat=NULL, long=NULL, status =
"open, acknowledged, closed, archived", limit = 100)
\end{verbatim}

\textbf{Arguments}

- \texttt{city} \hspace{1cm} A written description of the location for which issue types should be returned. If city is specified, lat/long should not be.
- \texttt{lat} \hspace{1cm} Latitude of coordinates, specified instead of city.
- \texttt{long} \hspace{1cm} Longitude of coordinates, specified instead of city.
- \texttt{status} \hspace{1cm} 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.
- \texttt{limit} \hspace{1cm} Number of items to return. Defaults to 100.

\textbf{Value}

- \texttt{issue_id} \hspace{1cm} Unique ID number for the given issue
- \texttt{status} \hspace{1cm} Status of the issue (open/closed)
- \texttt{summary} \hspace{1cm} Summary of the status for the issue
- \texttt{description} \hspace{1cm} Text description of the issue as reported
- \texttt{rating} \hspace{1cm} Importance rating of the issue
- \texttt{lat} \hspace{1cm} Latitude of the issue as reported
- \texttt{lng} \hspace{1cm} Longitude of the issue as reported
- \texttt{issue_address} \hspace{1cm} Address of the issue as reported
- \texttt{created_at} \hspace{1cm} Date and time when issue report was created
- \texttt{acknowledged_at} \hspace{1cm} Date and time when issue report was acknowledged by city
- \texttt{closed_at} \hspace{1cm} Date and time when issue report was closed by city
- \texttt{reopened_at} \hspace{1cm} Date and time when issue report was reopened, if it was
- \texttt{updated_at} \hspace{1cm} Date and time when issue report was last updated
- \texttt{shortened_url} \hspace{1cm} Shortened URL of the issue report
- \texttt{video_url} \hspace{1cm} URL for the video of the issue, if provided
get_issues_by_date

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

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

**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
get_issues_by_type

- 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

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

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

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

**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

gerget_city_issues

---

### Description

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

### Usage

```r
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.
list_places

Value

<table>
<thead>
<tr>
<th>title</th>
<th>Category of issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>organization</td>
<td>Organization handling this type of requests in this location</td>
</tr>
<tr>
<td>url</td>
<td>URL to this category of issues</td>
</tr>
<tr>
<td>potential_duplicate_issues_url</td>
<td>URL to possible duplicates in this category of issues</td>
</tr>
</tbody>
</table>

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

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

```java
## Returns:

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>url_name</th>
<th>county</th>
<th>state</th>
<th>place_type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Central</td>
<td>central_suffolk</td>
<td>Suffolk</td>
<td>MA</td>
<td>Neighborhood</td>
</tr>
<tr>
<td>2</td>
<td>Downtown</td>
<td>downtown_boston</td>
<td>Downtown</td>
<td>MA</td>
<td>Neighborhood</td>
</tr>
<tr>
<td>3</td>
<td>North End</td>
<td>northend_boston</td>
<td>North End</td>
<td>MA</td>
<td>Neighborhood</td>
</tr>
<tr>
<td>4</td>
<td>Beacon Hill</td>
<td>beacon_hill</td>
<td>Middlesex</td>
<td>Massachusetts</td>
<td>Neighborhood</td>
</tr>
<tr>
<td>5</td>
<td>East Cambridge</td>
<td>east-cambridge</td>
<td>Middlesex</td>
<td>MA</td>
<td>Neighborhood</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>url</th>
<th>html_url</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>html_report_url</th>
<th>type</th>
<th>lat</th>
<th>lng</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://seeclickfix.com/central_suffolk/report">https://seeclickfix.com/central_suffolk/report</a></td>
<td>Point</td>
<td>42.35854</td>
<td>-71.05931</td>
</tr>
<tr>
<td><a href="https://seeclickfix.com/downtown_boston/report">https://seeclickfix.com/downtown_boston/report</a></td>
<td>Point</td>
<td>42.35630</td>
<td>-71.05707</td>
</tr>
<tr>
<td><a href="https://seeclickfix.com/northend_boston/report">https://seeclickfix.com/northend_boston/report</a></td>
<td>Point</td>
<td>42.36534</td>
<td>-71.05325</td>
</tr>
<tr>
<td><a href="https://seeclickfix.com/beacon_hill/report">https://seeclickfix.com/beacon_hill/report</a></td>
<td>Point</td>
<td>42.35716</td>
<td>-71.06791</td>
</tr>
<tr>
<td><a href="https://seeclickfix.com/east-cambridge/report">https://seeclickfix.com/east-cambridge/report</a></td>
<td>Point</td>
<td>42.36833</td>
<td>-71.07928</td>
</tr>
</tbody>
</table>
```
Index

get_city_issues, 2, 5, 7, 8
get_issues_by_date, 3, 7
get_issues_by_type, 5, 5
get_specific_issue, 3, 7

list_issue_types, 7, 8, 10
list_places, 9, 9