Package ‘walkscoreAPI’

February 20, 2015

Type Package
Title Walk Score and Transit Score API
Version 1.2
Date 2012-01-04
Author John Whalen
Maintainer John Whalen <whalenjf@gmail.com>
Description A collection of functions to perform the Application Programming Interface (API) calls associated with the Walk Score website (www.walkscore.com) within the R environment. These functions can be used to query the Walk Score and Transit Score database for a wide variety of information using R scripts. This package includes the simple Walk Score and Transit Score API calls, which return the scores associated with an input location, as well as calls which return some data used to calculate the scores. These functions are especially useful for mass data collection and gathering Walk Score and Transit Score values for large lists of locations.
License GPL-2
LazyLoad yes
Repository CRAN
Date/Publication 2012-10-29 08:59:59
NeedsCompilation no

R topics documented:

walkscoreAPI-package .................. 2
checkTSsupport ........................ 3
geoloc .................................. 4
getTS .................................. 5
getTScities ............................. 6
getWS .................................. 7
networkSearch .......................... 8
Description

A collection of functions to perform the Application Programming Interface (API) calls associated with the Walk Score website (www.walkscore.com) within the R environment. These functions can be used to query the Walk Score and Transit Score database for a wide variety of information using R scripts. This package includes the simple Walk Score and Transit Score API calls, which return the scores associated with an input location, as well as calls which return some data used to calculate the scores. These functions are especially useful for mass data collection and gathering Walk Score and Transit Score values for large lists of locations.

Details

```
Package: walkscoreAPI
Type: Package
Version: 1.2
Date: 2012-01-04
License: GPL-2
LazyLoad: yes
```

Every function in this package requires the use of a Walk Score API key number, entered as a parameter. The key is free to obtain with limited use, and can be requested here: http://www.walkscore.com/professional/api.php.

The easiest way to enter the key is to store the string as a variable and entering that variable as a parameter for the function calls.

Author(s)

John Whalen
Maintainer: <whalenjf@gmail.com>

References

Visit www.walkscore.com for information on Walk Score and to obtain an API key

Examples

```
## Not run:
```
checkTSsupport

Description
A function to check if a city of interest is among the cities supported by the Transit Score.

Usage
checkTSsupport(city, state, key)

Arguments
  city         name of city of interest (string)
  state        postal abbreviation of city’s state (string)
  key          your Walk Score API key (string)

Details
Transit Score only works in select cities.

Value
TRUE if city is supported, FALSE otherwise

Note
Visit www.walkscore.com for information on Walk Score and to obtain an API key

Author(s)
John Whalen

References

See Also
getTScities
geoloc

Examples

## Not run:
checkTSsupport("Buffalo","NY","your key")
## End(Not run)

---

### geoloc

**Google Geolocation API Call**

---

**Description**

A function to perform the Google Geolocation API call, and return the longitude and latitude coordinates of the query location.

**Usage**

geoloc(address, apikey)

**Arguments**

- **address**: query address (string). Do not use commas, and zip codes are not required, e.g. "1600 Pennsylvania Ave Washington DC".
- **apikey**: your Google API key (string)

**Details**

Use of this function requires a Google API key, which is different from the Walk Score API key. Get one here: [http://code.google.com/apis/maps/signup.html](http://code.google.com/apis/maps/signup.html)

**Value**

Returns an object of class `googlegeoloc`, basically a list of the following elements:

- **coordinates**: A vector of two numbers, the first representing the longitude and the second representing the latitude.
- **accuracy**: Accuracy rating of geolocation.
- **city**: City containing the requested address
- **state**: State containing the requested address
- **country**: Country containing the requested address

**Note**

For description of Google Geolocation see here: [http://code.google.com/apis/gears/api_geolocation.html](http://code.google.com/apis/gears/api_geolocation.html)

**Author(s)**

John Whalen
getTS

References

http://code.google.com/apis/gears/api_geolocation.html

Examples

```r
## Not run:
geoloc("350 5th Avenue New York NY", "your Google API key")
## End(Not run)
```

---

**getTS**  
**Transit Score API Call**

Description

A function to perform the basic Transit Score API call.

Usage

```r
getTS(x, y, city, state, key)
```

Arguments

- `x` longitude of query location (numeric)
- `y` latitude of query location (numeric)
- `city` name of core city where the query location is located (string)
- `state` postal abbreviation of query location’s state (string)
- `key` your Walk Score API key (string)

Details

The Transit Score API call only works in supported cities. Use the functions "checkTSsupport" or "getTScities" to check for support in the city of interest. Also note that calls should use the core city name, even when the query location is technically in a suburb of the core city.

Value

Returns an object of class `TransitScore`, basically a list of the following elements:

- `transitscore` Transit Score of query location.
- `url` Link to Walk Score page associated with your query.
- `description` Qualitative description of query location regarding transit.
- `summary` Summary of nearby routes and stops.

Note

Visit www.walkscore.com for information on Walk Score and to obtain an API key.
getTScities

Author(s)
John Whalen

References

See Also
getWS

Examples

## Not run:
getTS(-73.98496,40.74807,"New York","NY","your key")

## End(Not run)

---

getTScities  List Cities With Transit Score Support

---

Description
A function to list cities supported by Transit Score.

Usage
getTScities(key)

Arguments

key  Your Walk Score API key (string)

Value
Prints a list of all cities currently supported by Transit Score.

Note
Visit www.walkscore.com for information on Walk Score and to obtain an API key

Author(s)
John Whalen

References
getWS

See Also
checkTSsupport

Examples

## Not run:
getTScities("your key")

## End(Not run)

---

**getWS**  
*Walk Score API Call*

**Description**
A function to perform the basic Walk Score API call.

**Usage**

getWS(x, y, key)

**Arguments**

- **x**: longitude of query location (numeric)
- **y**: latitude of query location (numeric)
- **key**: your Walk Score API key (string), see Details below

**Details**
Note that the call uses longitude and latitude coordinates and not addresses like the website interface. It is strongly recommended that Google Geolocation is used to convert addresses to coordinates because this is the method used by the Walk Score website, and will result in the same Walk Score as entering the address into the website interface. The function "geoloc" in this package is a tool for using the Google Geolocation API.

**Value**
Otherwise Returns an object of class `walkScore`, basically a list of the following elements:

- **status**: Status code of the request. Status of 1 indicates a successful call. See the Walk Score API page for interpretation of other codes.
- **walkscore**: Walk Score of query location.
- **description**: Qualitative description of location.
- **updated**: Date and time of most recent update to this location’s Walk Score.
- **snappedLong**: grid point longitude to which the input was snapped to.
- **snappedLat**: grid point latitude to which the input was snapped to.
networkSearch

Note

Visit www.walkscore.com for information on Walk Score and to obtain an API key

Author(s)

John Whalen

References

http://www.walkscore.com/professional/api.php

See Also

geloc

Examples

```r
## Not run:
getWS(-73.98496,40.74807,"your key")
## End(Not run)
```

networkSearch                Network Search

Description

A function to perform the Network Search API call.

Usage

```r
networkSearch(x, y, key)
```

Arguments

- `x` longitude of query location (numeric)
- `y` latitude of query location (numeric)
- `key` your Walk Score API key (string)

Details

gives information about all routes and all stops within a mile radius of a query location. This function returns a very data-rich object with details on every stop and every route included in this radius.
Value

Returns an object of class `networkSearch`, which has two elements: first is `$routelist`, which is a list of objects of class `route`, and second is `$stoplist`, which is a list of objects of class `stop`.

- **routelist**: List of routeID, route name, route category, maintaining agency, agency website, and stops served by the route.
- **stoplist**: List of stopID, stop name, stop latitude, stop longitude, and a list of routes which use the stop.

Note

Visit www.walkscore.com for information on Walk Score and to obtain an API key.

Author(s)

John Whalen

References


See Also

`stopDetails`, `routeDetails`

Examples

```r
## Not run:
networkSearch(-73.98496,40.74807,"your key")
## End(Not run)
```

routeDetails  

**Route Details**

Description

A function to perform the Route Details API call, which provides additional information about a particular route.

Usage

```r
routeDetails(routeid, key)
```

Arguments

- **routeid**: Route ID number for the route of interest (string)
- **key**: your Walk Score API key (string)
**Details**

Route ID is a unique string of characters used to identify routes. They can be obtained through the search functions provided in this library.

**Value**

Returns an object of class `routeDetails`, basically a list of the following elements:

- `routeID`: The route ID of the route of interest (same as input ID)
- `routeName`: Name of the route of interest
- `routeCategory`: Mode of transportation associated with this route
- `agency`: Transit agency associated with this route
- `agencyURL`: Agency website
- `routeGeometry`: Coordinates of route linestring, used for mapping the route.
- `stopList`: List of stop ID’s served by this route.

**Note**

Visit www.walkscore.com for information on Walk Score and to obtain an API key

**Author(s)**

John Whalen

**References**


**See Also**

`networkSearch`, `stopDetails`

**Examples**

```r
## Not run:
routeDetails("r415","your key")

## End(Not run)
```
**Description**

A function to perform the Stop Details API call, which provides additional information about a transit stop of interest.

**Usage**

\[
\text{stopDetails(stopid, key)}
\]

**Arguments**

- **stopid**: The stop ID of the stop of interest. (string)
- **key**: your Walk Score API key. (string)

**Details**

Stop ID is a unique string of characters used to identify stops. They can be obtained through the search functions provided in this library.

**Value**

Returns an object of class StopDetails, basically a list of the following elements:

- **stopID**: ID of the stop of interest (same as input ID)
- **stopName**: Name of stop of interest.
- **stopLong**: Longitude of stop of interest.
- **stopLat**: Latitude of stop of interest.
- **routeList**: List of route ID’s which serve the stop of interest.

**Note**

Visit www.walkscore.com for information on Walk Score and to obtain an API key

**Author(s)**

John Whalen

**References**


**See Also**

stopSearch, networkSearch
Examples

```r
## Not run:
stopDetails("s13993", "your key")

## End(Not run)
```

**stopSearch**

**Stop Search**

**Description**

A function to preform the Stop Search API Call.

**Usage**

```r
stopSearch(x, y, key)
```

**Arguments**

- **x**: longitude of query location (numeric)
- **y**: latitude of query location (numeric)
- **key**: your Walk Score API key (string)

**Details**

The Stop Search API Call returns the sixteen closest stops to a query location which service unique routes.

**Value**

Returns a list of objects of class "Stop2", which are basically lists with the following elements:

- **stopID**: A unique ID for the particular stop, which can be used to get additional info about that stop.
- **stopName**: Name of the particular stop.
- **stopDistance**: Distance to the query location, in miles.
- **stopLong**: Longitude of the particular stop.
- **stopLat**: Latitude of the particular stop.
- **routeDetails**: Object of class "RouteDetails", which lists the route id, name, catagory, and agency of the orute served at the stop.

**Note**

Visit www.walkscore.com for information on Walk Score and to obtain an API key.
walkshed

Author(s)

John Whalen

References


See Also

stopDetails, networkSearch

Examples

```r
## Not run:
stopSearch(~73.98496, 40.74807, "your key")

## End(Not run)
```

walkshed | Return Walk Shed

Description

A function to preform the "Walking Distance" API Call.

Usage

```r
walkshed(x, y, key)
```

Arguments

- `x` longitude of query location (numeric)
- `y` latitude of query location (numeric)
- `key` your Walk Score API key (string)

Details

The "Walking Distance" API call returns the geometry of a polygon which bounds network walking distance from the given origin point.
Value

Returns an object of class "Walkshed", which is basically a list with the following elements:

- **status**: A code which tells the status of the request. The table of status codes can be found on the Walk Score API site. A status of 1 means a successful request.
- **origin**: Latitude and longitude of the walkshed center.
- **geometry**: Geometry type of walkshed.
- **coordinates**: List of walkshed polygon vertexes, returned as a data frame with two columns.
- **radius**: Network distance in miles.
- **snappedLong**: Longitude to which the input was snapped to.
- **snappedLat**: Latitude to which the input was snapped to.

Note

Visit www.walkscore.com for information on Walk Score and to obtain an API key.

Author(s)

John Whalen

References


Examples

```r
## Not run:
walkshed(-122.335, 47.5815, "your key")

## End(Not run)
```
Index

*Topic **package**
  - walkscoreAPI-package, 2

checkTSsupport, 3, 7
geoloc, 4, 8
getTS, 5
getTScities, 3, 6
getWS, 6, 7

networkSearch, 8, 10, 11, 13
routeDetails, 9, 9
stopDetails, 9, 10, 11, 13
stopSearch, 11, 12

walkscoreAPI (walkscoreAPI-package), 2
walkscoreAPI-package, 2
walkshed, 13