walkscore frontend Documentation

Release 0.2

James Malone

April 13, 2015

Contents

Note: This module is under **active development** so things will change.

CHAPTER 1

Status

CHAPTER 2

Contents

2.1 Installation

To install this module, use the source on GitHub or install via pip with PyPi.

Note: The GitHub method may have newer features and functionality.

2.1.1 PyPi (pip) method

You can install this module via pip:

```
pip install walkscore-api-binding
```

2.1.2 Via GitHub

Alternatively, you can install the latest from this repository:

```
git clone git@github.com:evilsoapbox/walkscore_frontend.git
python setup.py install
```

2.2 Modules & Methods

2.2.1 Primary methods for walkscore frontend

Walkscore frontend to interact with the Walkscore website without using the limited API. Returns data about cities and neighborhoods from Walkscore.

`walkscore_frontend.`**`data_for_city`**(*city*, *state*)

Get the Walkscore data for the given city.

Parameters

- **city** – Name of the city (such as *Seattle*)
- **state** – Two-letter code of the state (such as *WA*)

Returns Walkscore data about the specified neighborhood

Return type dict

walkscore_frontend.**data_for_neighborhood**(*neighborhood*, *city*, *state*)
Get the Walkscore data for the given neighborhood.

Parameters

- **neighborhood** – Nane of the neighborhood (such as *Denny Triangle*)
- **city** – Name of the city (such as *Seattle*)
- **state** – Two-letter code of the state (such as *WA*)

Returns Walkscore data about the specified neighborhood

Return type dict

walkscore_frontend.**get_city**(*city*, *state*)
Return a City object with data for this city.

Parameters

- **city** – Name of the city (such as *Seattle*)
- **state** – Two-letter code of the state (such as *WA*)

Returns City object with data about the specified city

Return type City

walkscore_frontend.**get_neighborhood**(*neighborhood*, *city*, *state*)
Return a Neighborhood object with data for this neighborhood.

Parameters

- **neighborhood** – Nane of the neighborhood (such as *Denny Triangle*)
- **city** – Name of the city (such as *Seattle*)
- **state** – Two-letter code of the state (such as *WA*)

Returns Neighborhood object with data about the specified neighborhood

Return type Neighborhood

2.2.2 Submodules

2.2.3 walkscore_frontend.http module

Performs HTTP work for the frontend. Uses the Requests library to communicate via HTTP to WalkScore.

walkscore_frontend.http.**get_json_data**(*url*)
Return JSON formatted output of the content from the given URL.

Parameters **url** – url to access

Returns JSON-formatted content from given url

Return type string (json)

Raises Exception if the request did not return HTTP code 200

walkscore_frontend.http.**get_page_data**(*url*)
Return string formatted output of the content from the given URL.

Parameters **url** – url to access

Returns content from given url

Return type string (content)

Raises Exception if the request did not return HTTP code 200

walkscore_frontend.http.**walkscore_city_url**(*city*, *state*, *json=False*)
Return a well-formatted Walkscore URL a city.

Parameters

- **city** – Name of the city (such as *Seattle*)
- **state** – Two-letter code of the state (such as *WA*)
- **json** – Whether to return the the JSON data URL (for some metadata)

Returns Walkscore URL for given city/state

Return type string

walkscore_frontend.http.**walkscore_neighborhood_url**(*neighborhood*, *city*, *state*, *json=False*)
Return a well-formatted Walkscore URL a neighborhoods.

Parameters

- **neighborhood** – Nane of the neighborhood (such as *Denny Triangle*)
- **city** – Name of the city (such as *Seattle*)
- **state** – Two-letter code of the state (such as *WA*)
- **json** – Whether to return the the JSON data URL (for some metadata)

Returns Walkscore URL for given neighborhood/city/state

Return type string

2.2.4 walkscore_frontend.regex module

Handles regular expression (regex) parsing for the walkscore_frontend module.

walkscore_frontend.regex.**parse_data_points**(*content*)
Parse the page data and look for expected contents based on regular expressions.

Parameters **content** – content to search

Returns parsed data based on the built-in regex searches

Return type dict

walkscore_frontend.regex.**regex_page_data**(*pattern*, *content*, *rtype=None*)
Get a value from page data based on a regex pattern.

Parameters

- **pattern** – regex pattern to match against
- **content** – content to search

Returns first result for the given match

Return type object

walkscore_frontend.regex.**regex_page_data_table**(*pattern*, *content*)
Extract data from a table on the text based on an id and value.

Parameters

- **pattern** – id and value of element to parse
- **content** – content to search

Returns array of data from the table

Return type string array

2.2.5 walkscore_frontend.utils module

walkscore_frontend.utils.**merge_dicts**(**dicts*)
Merge dictionaries together.

walkscore_frontend.utils.**remove_unneeded_elements**(*dict_to_clean*, *attrs_to_remove*)
Pop unneeded elements from the given dictionary.

2.2.6 walkscore_frontend.wslocation module

class walkscore_frontend.wslocation.**City**(**init_data*, ***kwargs*)
Bases: walkscore_frontend.wslocation.WsLocation

Represents a city from the Walkscore website.

neighborhoods
Get a list of neighborhoods for the city.

class walkscore_frontend.wslocation.**Neighborhood**(**init_data*, ***kwargs*)
Bases: walkscore_frontend.wslocation.WsLocation

Represents a neighborhood from the Walkscore website.

class walkscore_frontend.wslocation.**WsLocation**(**init_data*, ***kwargs*)
Represents a location helf in the Walkscore website.

You can find this module on GitHub or PyPi.

CHAPTER 3

Indices and tables

- *genindex*
- *modindex*
- *search*

W

C

D

G

M

N

P

R

W