python-lametro-api Documentation

Release 0.2

Los Angeles Times Data Desk

March 16, 2014

A simple Python wrapper for L.A. Metro's Realtime API for bus stops, routes and vehicles

CHAPTER 1

Features

- Retrieve the current location of Metro buses.
- Retrieve the location of Metro bus stops, and predictions for when buses will next arrive.
- Retrieve all Metro bus routes and the stops they connect with.

CHAPTER 2

Documentation

2.1 Getting started

This tutorial will walk you through the process of installing python-lametro-api and making your first requests.

2.1.1 Installation

Provided that you have pip installed, you can install the library like so:

```
$ pip install python-lametro-api
```

2.1.2 Creating a client

Before you can interact with Metro's data, you first must import the library and initialize a client to talk with the site on your behalf.

```
>>> from la_metro import LAMetro
>>> client = LAMetro()
```

2.1.3 Retrieve a bus stop and get predictions for incoming buses

```
>>> obj = client.bus.stops.get(6033)
>>> obj
<BusStop: Santa Monica / Vermont>
>>> obj.predictions
[<BusPrediction: Santa Monica / Vermont (4)>, <BusPrediction: Santa Monica / Vermont (4)>]
```

2.1.4 Retrieve a bus route and get the location of all stops and vehicles

```
>>> obj = client.bus.routes.get(704)
>>> obj
<BusRoute: 704>
>>> obj.stops
[<BusStop: 2nd / Santa Monica>, <BusStop: Ocean / Santa Monica>, <BusStop: Santa Monica / 4th>, <Bus
>>> obj.vehicles
[<BusVehicle: 9364>, <BusVehicle: 9376>, <BusVehicle: 9391>, <BusVehicle: 9380>, <BusVehicle: 9390>,
```

2.1.5 Get the location of vehicles

Here's how you can get all vehicles:

```
>>> obj_list = client.bus.vehicles.all()
>>> len(obj_list)
392
>>> obj_list[0]
<BusVehicle: 7433>
```

And here's how to get a single one:

```
>>> obj = client.bus.vehicles.get(7433)
>>> obj.latitude, obj.longitude
(34.047089, -118.282776)
# Also available with some other mappable attributes
>>> obj.y, obj.x
(34.047089, -118.282776)
>>> obj.wkt
POINT(-118.282776 34.047089)
>>> obj.geojson
{"type": "Point", "coordinates": [-118.282776, 34.047089]}
```

2.2 Bus data

Methods for retrieving data about buses, stops and routes in the L.A. Metro system.

2.2.1 Stops

`client.bus.stops.`**`get`**(*id*)
: Return the stop with the provided Metro identifer.

  ```
  >>> from la_metro import LAMetro
  >>> client = LAMetro()
  >>> client.bus.stops.get(6033)
  <BusStop: Santa Monica / Vermont>
  ```

`stop_obj.`**`id`**
: The identifier in the Metro system

`stop_obj.`**`name`**
: The name of the bus stop

`stop_obj.`**`latitute`**
: The y coordinate of the stop's location

`stop_obj.`**`longitude`**
: The x coordinate of the stop's location

`stop_obj.`**`y`**
: Alias to the latitude of the stop's location

`stop_obj.`**`x`**
: Alias to the longitude of the stop's location

`stop_obj.`**`wkt`**
: The stop's location in Well-Known Text format

`stop_obj.`**`geojson`**
The stop's location in GeoJSON format

`stop_obj.`**`messages`**
Returns an messages Metro has left for users of this bus stop. This can contain information about service problems and delays.

`stop_obj.`**`predictions`**
Returns a list of predictions that guess when busses will next arrive at this stop.

`stop_obj.`**`routes`**
Returns a list of the routes that connect with this bus route.

2.2.2 Routes

`client.bus.routes.`**`all`**`()`
Return all routes in the Metro system

```
>>> from la_metro import LAMetro
>>> client = LAMetro()
>>> client.bus.routes.all()
[<BusRoute: 2>, <BusRoute: 4>, <BusRoute: 10>, <BusRoute: 14>, <BusRoute: 16>, <BusRoute: 18>, <
```

`client.bus.routes.`**`get`**`(`*`id`*`)`
Return the route with the provided Metro identifer.

```
>>> from la_metro import LAMetro
>>> client = LAMetro()
>>> client.bus.routes.get(704)
<BusRoute: 704>
```

`route_obj.`**`id`**
The identifier in the Metro system

`route_obj.`**`name`**
The name of the bus route

`route_obj.`**`runs`**
Returns a list of the runs on this bus route.

`route_obj.`**`stops`**
Returns a list of the stops on this bus route in their proper order.

`route_obj.`**`vehicles`**
Returns a list of the vehicles on this bus route with their latest positions.

2.2.3 Vehicles

`client.bus.vehicles.`**`all`**`()`
Return all vehicles out in the Metro system

```
>>> from la_metro import LAMetro
>>> client = LAMetro()
[<BusVehicle: 3129>, <BusVehicle: 6735>, <BusVehicle: 7433>, <BusVehicle: 6729>, <BusVehicle: 92
```

`client.bus.vehicles.`**`get`**`(`*`id`*`)`
Return the vehicle with the provided Metro identifer.

```
>>> from la_metro import LAMetro
>>> client = LAMetro()
>>> client.bus.vehicles.get(7433)
<BusVehicle: 7433>
```

`vehicle_obj.`**`id`**
The identifier in the Metro system

`vehicle_obj.`**`seconds_since_report`**
The time since the data on this vehicle was last updated

`vehicle_obj.`**`is_predictable`**
The boolean indicator related to whether or not the busses arrival time can be predicted that I do not understand

`vehicle_obj.`**`id`**
The identifier in the Metro system

`vehicle_obj.`**`latitute`**
The y coordinate of the vehicle's location

`vehicle_obj.`**`longitude`**
The x coordinate of the vehicle's location

`vehicle_obj.`**`y`**
Alias to the latitude of the vehicle's location

`vehicle_obj.`**`x`**
Alias to the longitude of the vehicle's location

`vehicle_obj.`**`wkt`**
The vehicle's location in Well-Known Text format

`vehicle_obj.`**`geojson`**
The vehicle's location in GeoJSON format

`vehicle_obj.`**`heading`**

`vehicle_obj.`**`route`**
The route the vehicle is on.

`vehicle_obj.`**`run`**
The run the vehicle is on.

2.2.4 Runs

`run_obj.`**`id`**
The identifier in the Metro system

`run_obj.`**`name`**
The name of the bus run

`run_obj.`**`direction`**
The direction the run is going along the route

`run_obj.`**`route`**
The route the run is on.

2.2.5 Predictions

`prediction_obj.`**`stop`**
: The stop this prediction is estimating an arrival for

`prediction_obj.`**`route`**
: The route the prediction is estimating an arrival for

`prediction_obj.`**`run`**
: The run the prediction is estimating an arrival for

`prediction_obj.`**`minutes`**
: The estimated arrival time in minutes

`prediction_obj.`**`seconds`**
: The estimated arrival time in seconds

`prediction_obj.`**`is_departing`**
: A boolean indicator I do not understand

2.3 Changelog

2.3.1 0.2

- Improved test coverage
- Python 3 fixes
- Automated coverage testing with coveralls.io
- PEP8 and PyFlakes compilance and testing

2.3.2 0.1.4

- Alternative GIS formats for longitude and latitude
- Python 3.3 support
- Removed unneeded dependencies
- Travis CI integration

2.3.3 0.1

- A rough client that pulls bus data on stops, routes and vehicles.

CHAPTER 3

Contributing

- Code repository: https://github.com/datadesk/python-lametro-api
- Issues: https://github.com/datadesk/python-lametro-api/issues
- Packaging: https://pypi.python.org/pypi/python-lametro-api
- Testing: https://travis-ci.org/datadesk/python-lametro-api
- Coverage: https://coveralls.io/r/datadesk/python-lametro-api