landtransportsg

Release 1.3.0

Yuhui

May 31, 2020

CONTENTS

Release v1.3.0.

This is an unofficial Python package for interacting with APIs available at LTA DataMall.

CHAPTER

ONE

INSTALLING THE PACKAGE

Install the package using `pip`:

```
pip install landtransportsg
```

CHAPTER

TWO

USING THE PACKAGE

Pre-requisite:

- API key. Request for API access.

The main steps are:

1. Import a class.
2. Instantiate an object from the class.
3. Call a function on that object.

Refer to any of the modules' documentation for specific examples.

CHAPTER

THREE

CONTENTS

3.1 Package Overview

Interacting with LTA DataMall's API is done through one of three clients, where each client corresponds with a set of endpoints.

The four clients are: `ActiveMobility`, `Geospatial`, `PublicTransport` and `Traffic`.

Each client contains several public functions, one function per endpoint. A function's name is the same as its corresponding endpoint's ending path.

Some functions accept named arguments, where an argument corresponds with a parameter that the endpoint accepts.

> *Why have separate clients instead of one single client?*
>
> Without knowing how LTA DataMall's API will evolve, and noticing that the endpoints were themselves already partitioned into sets, it seemed like a good idea to keep each set of endpoints in its own contextual client. This allows for each set of endpoints to be customised on their own, e.g. the `PublicTransport` passenger volume-related endpoints allow for a string to be returned, whereas the other endpoints return a list.

3.2 Package Reference

3.2.1 landtransportsg

Module contents

`landtransportsg.`**`ActiveMobility`**
alias of *landtransportsg.active_mobility.client.Client*

`landtransportsg.`**`Geospatial`**
alias of *landtransportsg.geospatial.client.Client*

`landtransportsg.`**`PublicTransport`**
alias of *landtransportsg.public_transport.client.Client*

`landtransportsg.`**`Traffic`**
alias of *landtransportsg.traffic.client.Client*

landtransportsg.exceptions

Exceptions that could occur when interacting with any API endpoint.

exception landtransportsg.exceptions.**APIError**(*message*, *errors=None*)
Bases: Exception

Error when the API returns an error.

message
The general error message to display when the error is raised.

Type str

errors
(optional) Other messages that were part of the raised error.

Type list of str

3.2.2 landtransportsg.active_mobility

Client for interacting with the Active Mobility API endpoints.

Example usage:

```
# get the bicycle parking locations
from landtransportsg import ActiveMobility
client = ActiveMobility(API_KEY)
bicycle_parking_locations = client.bicycle_parking(1.364897, 103.766094)
```

Methods

class landtransportsg.active_mobility.client.**Client**(*account_key*)
Bases: landtransportsg.client.__Client

Interact with the active mobility-related endpoints.

References

https://www.mytransport.sg/content/dam/datamall/datasets/LTA_DataMall_API_User_Guide.pdf

bicycle_parking(*latitude*, *longitude*, *distance=0.5*)
Get bicycle parking locations within a radius.

Parameters

- **latitude** (*float*) – Latitude map coordinates of a location.
- **longitude** (*float*) – Longitude map coordinates of a location.
- **distance** (*float*) – (optional) Radius in kilometres from the latitude-longitude location to retrieve bicycle parking locations. Default: 0.5.

Returns (list) Available bicycle parking locations at the specified location.

Raises **ValueError** – If latitude, longitude or distance are not floats.

3.2.3 landtransportsg.geospatial

Client for interacting with the Geospatial API endpoints.

Example usage:

```
# get a geospatial whole island layer
from landtransportsg import Geospatial
client = Geospatial(API_KEY)
geospatial_whole_island = client.geospatial_whole_island('ArrowMarking')
```

Methods

class landtransportsg.geospatial.client.**Client**(*account_key*)

Bases: landtransportsg.client.__Client

Interact with the geospatial-related endpoints.

References

https://www.mytransport.sg/content/dam/datamall/datasets/LTA_DataMall_API_User_Guide.pdf

geospatial_whole_island(*geospatial_layer_id*)

Get the SHP files of the requested geospatial layer.

Parameters **geospatial_layer_id** (*str*) – Name of geospatial layer. Refer to the GEOSPATIAL_WHOLE_ISLAND_LAYER_IDS constant for the list of valid names.

Returns (str) Link for downloading the requested SHP file.

Raises

- **ValueError** – Raised if geospatial_layer_id is not specified.
- **ValueError** – Raised if geospatial_layer_id is not a string.
- **ValueError** – Raised if geospatial_layer_id is not a valid ID.

3.2.4 landtransportsg.public_transport

Client for interacting with the Public Transport API endpoints.

Example usage:

```
# get the bus arrival information at a bus stop
from landtransportsg import PublicTransport
client = PublicTransport(API_KEY)
bus_arrival = client.bus_arrival('83139')
```

Methods

class landtransportsg.public_transport.client.**Client**(*account_key*)
Bases: landtransportsg.client.__Client

Interact with the public transport-related endpoints.

References

https://www.mytransport.sg/content/dam/datamall/datasets/LTA_DataMall_API_User_Guide.pdf

bus_arrival(*bus_stop_code*, *service_number=None*)
Get real-time Bus Arrival information of Bus Services at a queried Bus Stop, including Est. Arrival Time, Est. Current Location, Est. Current Load.

Parameters

- **bus_stop_code** (*str*) – 5-digit bus stop reference code.
- **service_number** (*str*) – (optional) Bus service number. If omitted, then all bus services at the bus stop code are returned.

Returns (list) Information about bus arrival at the specified bus stop.

Raises

- **ValueError** – Raised if bus_stop_code or service_number are not strings.
- **ValueError** – Raised if bus_stop_code is not exactly 5 characters long.
- **ValueError** – Raised if bus_stop_code is not a number-like string.

bus_routes()
Get detailed route information for all services currently in operation, including: all bus stops along each route, first/last bus timings for each stop.

Returns (list) Information about bus routes currently in operation.

bus_services()
Get detailed service information for all buses currently in operation, including: first stop, last stop, peak / offpeak frequency of dispatch.

Returns (list) Information about bus services currently in operation.

bus_stops()
Get detailed information for all bus stops currently being serviced by buses, including: Bus Stop Code, location coordinate.

Returns (list) Location coordinaties of bus stops with active services.

facilities_maintenance(*station_code*)
Get the pre-signed links to JSON file containing facilities maintenance schedules of the particular station.

Parameters **station_code** (*str*) – Station Code of train station. Refer to the STATION_CODES_REGEX_PATTERN constant for the expected regex pattern that this code has to match.

Returns (str) Link for downloading the requested JSON file.

Raises

- **ValueError** – Raised if station_code is not specified.
- **ValueError** – Raised if station_code is not a string.

- **ValueError** – Raised if station_code does not match the expected regex pattern.

passenger_volume_by_bus_stops(*dt=None*)
Get tap in and tap out passenger volume by weekdays and weekends for individual bus stop.

Parameters **dt** (*date*) – (optional) Date of a specific month to get passenger volume. This must be a valid date object, e.g. *date(2019, 7, 2)*. But only the year and month will be used since that is what the endpoint accepts. Must be within the last 3 months of the current month.

Returns (str) Download link of file containing passenger volume data.

passenger_volume_by_origin_destination_bus_stops(*dt=None*)
Get number of trips by weekdays and weekends from origin to destination bus stops.

Parameters **dt** (*date*) – (optional) Date of a specific month to get passenger volume. This must be a valid date object, e.g. *date(2019, 7, 2)*. But only the year and month will be used since that is what the endpoint accepts. Must be within the last 3 months of the current month.

Returns (str) Download link of file containing passenger volume data.

passenger_volume_by_origin_destination_train_stations(*dt=None*)
Get number of trips by weekdays and weekends from origin to destination train stations.

Parameters **dt** (*date*) – (optional) Date of a specific month to get passenger volume. This must be a valid date object, e.g. *date(2019, 7, 2)*. But only the year and month will be used since that is what the endpoint accepts. Must be within the last 3 months of the current month. Default: None, i.e. today.

Returns (str) Download link of file containing passenger volume data.

passenger_volume_by_train_stations(*dt=None*)
Get tap in and tap out passenger volume by weekdays and weekends for individual train station.

Parameters **dt** (*date*) – (optional) Date of a specific month to get passenger volume. This must be a valid date object, e.g. *date(2019, 7, 2)*. But only the year and month will be used since that is what the endpoint accepts. Must be within the last 3 months of the current month.

Returns (str) Download link of file containing passenger volume data.

taxi_availability()
Get location coordinates of all Taxis that are currently available for hire. Does not include "Hired" or "Busy" Taxis.

Returns (list) Location coordinaties of available taxis.

taxi_stands()
Get detailed information of Taxi stands, such as location and whether is it barrier free.

Returns (list) Detailed information of taxi stands .

train_service_alerts()
Get detailed information on train service unavailability during scheduled operating hours, such as affected line and stations etc.

Returns (list) Information about train service unavailability.

3.2.5 landtransportsg.traffic

Client for interacting with the Traffic API endpoints.

Example usage:

```
# get the list of available car park spaces
from landtransportsg import Traffic
client = Traffic(API_KEY)
patents = client.carpark_availability()
```

Methods

class landtransportsg.traffic.client.**Client**(*account_key*)

Bases: landtransportsg.client.__Client

Interact with the traffic-related endpoints.

References

https://www.mytransport.sg/content/dam/datamall/datasets/LTA_DataMall_API_User_Guide.pdf

carpark_availability()

Get number of available lots from HDB, LTA and URA carpark data.

Returns (list) Available carpark lots.

erp_rates()

Get ERP rates of all vehicle types across all timings for each zone.

Returns (list) ERP rates per vehicle type by zones.

estimated_travel_times()

Get estimated travel times of expressways (in segments).

Returns (list) Expressway estimated travel times by segments.

faulty_traffic_lights()

Get alerts of traffic lights that are currently faulty, or currently undergoing scheduled maintenance.

Returns (list) Traffic light alerts and their status.

road_openings()

Get all planned road openings.

Returns (list) Road openings for road works.

road_works()

Get all road works being / to be carried out.

Returns (list) Road works that are being / to be carried out.

traffic_images()

Get links to images of live traffic conditions along expressways and Woodlands & Tuas Checkpoints.

Returns (list) Traffic images at expressways and checkpoints.

traffic_incidents()

Get incidents currently happening on the roads, such as Accidents, Vehicle Breakdowns, Road Blocks, Traffic Diversions etc.

Returns (list) Traffic incidents currently happening.

traffic_speed_bands()
Get current traffic speeds on expressways and arterial roads, expressed in speed bands.

Returns (list) Traffic speed bands on expressways and arterial roads.

vms()
Get traffic advisories (via variable message services) concerning current traffic conditions that are displayed on EMAS signboards along expressways and arterial roads.

Returns (list) Traffic advisories for expressways and arterial roads.

3.3 External References

LTA DataMall's Developer Guide

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

CHAPTER

FIVE

LICENSE

This project is licensed under the GNU General Public License v3.0.

CHAPTER

SIX

LOGO

The logo is copyright by the Land Transport Authority, Singapore. It is used for reference only. This project's owner does not have any copyright claim over it.

PYTHON MODULE INDEX

l

INDEX

R

T

V