VGN

Feb 17, 2020

Contents:

CHAPTER 1

Functions

1.1 API Version

vgn.functions.**api_version**() → str
Version info from the VGN REST-API.

1.2 Stations

vgn.functions.**stations**(*station_name: str*) → List[vgn.data_classes.Station]
List of stations for the specified station name.

Parameters **station_name** – Name of a station.

Returns List of station objects for the given stop_name.

Return type list

vgn.functions.**all_stations**() → List[vgn.data_classes.Station]
List of all stations.

Returns List of stations for the VGN transport association.

Return type list

vgn.functions.**nearby_stations**(*location: vgn.data_classes.Coordinates*, *radius: int = 1000*) → List[vgn.data_classes.Station]
List stops close to a given location.

Parameters

- **location** – Search for stations close to this location.
- **radius** (*optional*) – Radius for search in meter

Returns List of station objects in radius of the given location.

Return type list

vgn.functions.**station_additional_information**(*stop_id: int*) → List[str]
List of information text strings for a given stop.

Parameters **stop_id** (*optional*) – The VGN stop identifier number.

Returns List of strings containing additional information for the given station.

Return type list

1.3 Departures

vgn.functions.**departure_schedule**(*stop_id: int, transport_type: List[vgn.data_classes.TransportType] = [TransportType(), TransportType(), TransportType()], timespan: int = 10, timedelay: int = 5, limit_result: int = 0*) → List[vgn.data_classes.Departure]
Departures for a specific stop.

Parameters

- **stop_id** – The VGN stop identifier number.
- **transport_type** – Information shall only be given for the defined transport means of transportation.
- **limit_result** (*optional*) – Limit amount of returned results. Zero means no limit.
- **timedelay** (*optional*) – Time delay for the request in minutes.
- **timespan** (*optional*) – Time window for the query in minutes.

Returns List of departures for the given station.

Return type list

vgn.functions.**departure_schedule_for_line**(*stop_id: int, line_name: str, timespan: int = 10, timedelay: int = 5, limit_result: int = 0*) → List[vgn.data_classes.Departure]
List of Departures for a specific stop and line.

Parameters

- **line_name** – Name of the line. For example 'U2' for the underground line two.
- **stop_id** – The VGN stop identifier number.
- **limit_result** (*optional*) – Limit amount of returned results. Zero means no limit.
- **timedelay** (*optional*) – Time delay for the request in minutes.
- **timespan** (*optional*) – Time window for the query in minutes.

Returns List of departures for the given station and line.

Return type list

1.4 Rides

vgn.functions.**rides**(*transport_type: vgn.data_classes.TransportType, time_span: int = 60*) → List[vgn.data_classes.Ride]
All running and starting rides for a given transport type within a given time frame (default 60 minutes)

Parameters

- **transport_type** – Transportation type. For example Bus.
- **time_span** (*optional*) – Time window in minutes (default 60 minutes)

Returns List of rides for the given transport type within the time window.

Return type list

1.5 Routes

vgn.functions.**route**(*transport_type: vgn.data_classes.TransportType*, *ride_id: int*) → vgn.data_classes.Route

Route for a given transport type and ride number for the current operating day

Parameters

- **transport_type** – Transportation type. For example Bus.
- **ride_id** – Ride number for the given transportation type

Returns The route for the given ride_number

Return type *Route*

vgn.functions.**route_for_day**(*transport_type: vgn.data_classes.TransportType*, *ride_id: int*, *day: datetime.date*) → vgn.data_classes.Route

Route for a given transport type, ride number and operating day.

Parameters

- **transport_type** – Transportation type. For example Bus.
- **ride_id** – Ride number for the given transportation type.
- **day** – Operating day date for the request.

Returns The route for the given ride_number on the requested day.

Return type *Route*

CHAPTER 2

Data Classes

class vgn.data_classes.**Coordinates**(*latitude: float*, *longitude: float*)

Coordinates in WGS 84 Format in degrees.

class vgn.data_classes.**Departure**(*line_name: str*, *station_id: str*, *direction: str*, *direction_text: str*, *planned_departure_time: datetime.datetime*, *actual_departure_time: datetime.datetime*, *transport_type: vgn.data_classes.TransportType*, *coordinates: vgn.data_classes.Coordinates*, *ride_id: int*, *ride_type_id: int*, *vehicle_number: str*, *forecast: bool*)

Departure data object class.

class vgn.data_classes.**Ride**(*ride_id: int*, *line_name: str*, *direction: str*, *operating_day: datetime.date*, *start_time: datetime.datetime*, *end_time: datetime.datetime*, *start_station_id: str*, *end_station_id: str*, *vehicle_number: str*)

Ride data object class.

class vgn.data_classes.**Route**(*line_name: str, direction: str, direction_text: str, ride_id: int, operating_day: datetime.date, is_cancelled: bool, additional_ride: bool, vehicle_number: str, transport_type: vgn.data_classes.TransportType, route: List[vgn.data_classes.RoutePoint]*)

Route for a specific ride.

class vgn.data_classes.**RoutePoint**(*station_name: str*, *station_id: int*, *stop_point: str*, *planned_arrival_time: datetime.datetime*, *actual_arrival_time: datetime.datetime*, *planned_departure_time: datetime.datetime*, *actual_departure_time: datetime.datetime*, *direction_text: str*, *coordinates: vgn.data_classes.Coordinates*, *transit: bool*, *no_boarding: bool*, *no_get_off: bool*, *additional_stop: bool*)

Single stop of a route.

class vgn.data_classes.**Station**(*name: str, station_id: int, coordinates: vgn.data_classes.Coordinates, transport_types: List[vgn.data_classes.TransportType]*)

Station data object class.

class vgn.data_classes.**TransportType**

Type of transportation (e.g.: bus, tram, subway).

CHAPTER 3

Exceptions

Exceptions for VGN requests

exception vgn.exceptions.**VgnGetError**
Custom exception to be thrown if get request to VGN API does not succeed

Python Module Index

v

Index