pygtfs Documentation

Release 0.1.5.dev6+ge0be101

Yaron de Leeuw

Nov 12, 2018

Contents

1 Get it **1**

2 Basic usage **3**

3 gtfs2db **5**

4 Detailed refernce **7**

5 Contents: **9**

5.1 The schedule module 9
5.2 The loader module 11
5.3 GTFS entities 11
5.4 The gtfs2db script 16
5.5 Internally used modules 17

6 Indices and tables **19**

Python Module Index **21**

CHAPTER 1

Get it

The source is available on github: https://github.com/jarondl/pygtfs

CHAPTER 2

Basic usage

To include pygtfs functionality in your application, use *import pygtfs*.

The first thing you need to to is to create a new schedule object:

```
sched = pygtfs.Schedule(":memory:")
```

This will create an in memory sqlite database. Instead you can supply a filename to be used for sqlite (such as ('gtfs.sqlite'), or a sqlalchemy database connection.

Then you can load gtfs feeds into the databas, by using append:

```
pygtfs.append_feed(sched, "sample-gtfs-feed.zip")
```

Where the gtfs feed can be either a *.zip* file, or a folder full of *.txt* files. You can add as many feeds as you want into a single database, without fear of conflicts (but you can two stop names for one place, one from each feed for example). Another option to load feeds is to use the 'gtfs2db' script as explained later.

The Schedule object represents a collection of objects that correspond to the contents of a GTFS feed. You can get the list of agencies, stops, routes, etc. with fairly straightforwardly named attributes, see *pygtfs.schedule* for more details.

```
>>> sched.agencies
[<Agency BART: Bay Area Rapid Transit>, <Agency AirBART: AirBART>]
>>> sched.routes
[<Route AirBART: >, <Route 01: >, <Route 03: >, <Route 05: >, <Route 07: >, <Route␣
↪11: >]
```

For GTFS entities that are identified by a dataset-unique identifier, there is also a function to get them by id:

```
>>> sched.agencies_by_id('AirBART')
[<Agency AirBART: AirBART>]
>>> sched.stops_by_id('SFIA')
[<Stop SFIA: San Francisco Int'l Airport>]
```

The GTFS entity objects have attributes that correspond in name to the field definitions in the [GTFS reference](https://developers.google.com/transit/gtfs/reference).

```
>>> sched.stops_by_id('SFIA')[0].stop_name
u"San Francisco Int'l Airport"
>>> sched.routes[1].route_long_name
u'Pittsburg/Bay Point - SFIA/Millbrae'
```

GTFS entities which cross-reference each other can also be obtained straightforwardly with attributes (again, see "Reference" below for full details):

```
>>> sched.trips_by_id('01SFO10').service  # the service associated with trip 01SFO10
<Service WKDY (MTWThFSSu)>
```

CHAPTER 3

gtfs2db

setup.py install will also install a command-line script *gtfs2db* that takes a GTFS zip file or directory as an argument and will load the data into a database usable with pygtfs. Run *gtfs2db –help* for more.

CHAPTER 4

Detailed refernce

The best place to start is *pygtfs.schedule*

CHAPTER 5

Contents:

5.1 The schedule module

class pygtfs.schedule.**Schedule**(*db_connection*)

Represents the full database.

The schedule is the most important object in pygtfs. It represents the entire dataset. Most of the properties come straight from the gtfs reference. Two of them were renamed: calendar is called *services*, and calendar_dates *service_exceptions*. One addition is the *feeds* table, which is here to support more than one feed in a database.

Each of the properties is a list created upon access by sqlalchemy. Then, each element of the list as attributes following the gtfs reference. In addition, if they are related to another table, this can also be accessed by attribute.

Parameters **db_conection** – Either a sqlalchemy database url or a filename to be used with sqlite.

agencies

A list of *pygtfs.gtfs_entities.Agency* objects

agencies_by_id(*id*)

A list of *pygtfs.gtfs_entities.Agency* objects with matching id

agencies_query

A sqlalchemy.orm.Query object to fetch *pygtfs.gtfs_entities.Agency* objects

drop_feed(*feed_id*)

Delete a feed from a database by feed id

fare_rules

A list of *pygtfs.gtfs_entities.FareRule* objects

fare_rules_query

A sqlalchemy.orm.Query object to fetch *pygtfs.gtfs_entities.FareRule* objects

fares

A list of *pygtfs.gtfs_entities.Fare* objects

fares_by_id(*id*)
A list of *pygtfs.gtfs_entities.Fare* objects with matching id

fares_query
A sqlalchemy.orm.Query object to fetch *pygtfs.gtfs_entities.Fare* objects

feed_infos
A list of *pygtfs.gtfs_entities.FeedInfo* objects

feed_infos_query
A sqlalchemy.orm.Query object to fetch *pygtfs.gtfs_entities.FeedInfo* objects

feeds
A list of *pygtfs.gtfs_entities.Feed* objects

feeds_by_id(*id*)
A list of *pygtfs.gtfs_entities.Feed* objects with matching id

feeds_query
A sqlalchemy.orm.Query object to fetch *pygtfs.gtfs_entities.Feed* objects

frequencies
A list of *pygtfs.gtfs_entities.Frequency* objects

frequencies_query
A sqlalchemy.orm.Query object to fetch *pygtfs.gtfs_entities.Frequency* objects

routes
A list of *pygtfs.gtfs_entities.Route* objects

routes_by_id(*id*)
A list of *pygtfs.gtfs_entities.Route* objects with matching id

routes_query
A sqlalchemy.orm.Query object to fetch *pygtfs.gtfs_entities.Route* objects

service_exceptions
A list of *pygtfs.gtfs_entities.ServiceException* objects

service_exceptions_by_id(*id*)
A list of *pygtfs.gtfs_entities.ServiceException* objects with matching id

service_exceptions_query
A sqlalchemy.orm.Query object to fetch *pygtfs.gtfs_entities.ServiceException* objects

services
A list of *pygtfs.gtfs_entities.Service* objects

services_by_id(*id*)
A list of *pygtfs.gtfs_entities.Service* objects with matching id

services_query
A sqlalchemy.orm.Query object to fetch *pygtfs.gtfs_entities.Service* objects

shapes
A list of *pygtfs.gtfs_entities.ShapePoint* objects

shapes_query
A sqlalchemy.orm.Query object to fetch *pygtfs.gtfs_entities.ShapePoint* objects

stop_times
A list of *pygtfs.gtfs_entities.StopTime* objects

stop_times_query
A `sqlalchemy.orm.Query` object to fetch *pygtfs.gtfs_entities.StopTime* objects

stops
A list of *pygtfs.gtfs_entities.Stop* objects

stops_by_id(*id*)
A list of *pygtfs.gtfs_entities.Stop* objects with matching id

stops_query
A `sqlalchemy.orm.Query` object to fetch *pygtfs.gtfs_entities.Stop* objects

transfers
A list of *pygtfs.gtfs_entities.Transfer* objects

transfers_query
A `sqlalchemy.orm.Query` object to fetch *pygtfs.gtfs_entities.Transfer* objects

translations
A list of *pygtfs.gtfs_entities.Translation* objects

translations_query
A `sqlalchemy.orm.Query` object to fetch *pygtfs.gtfs_entities.Translation* objects

trips
A list of *pygtfs.gtfs_entities.Trip* objects

trips_by_id(*id*)
A list of *pygtfs.gtfs_entities.Trip* objects with matching id

trips_query
A `sqlalchemy.orm.Query` object to fetch *pygtfs.gtfs_entities.Trip* objects

5.2 The loader module

pygtfs.loader.**append_feed**(*schedule*, *feed_filename*, *strip_fields=True*, *chunk_size=5000*, *agency_id_override=None*)

pygtfs.loader.**delete_feed**(*schedule*, *feed_filename*, *interactive=False*)

pygtfs.loader.**list_feeds**(*schedule*)

pygtfs.loader.**overwrite_feed**(*schedule*, *feed_filename*, **args*, ***kwargs*)

5.3 GTFS entities

GTFS entities.

These are the entities returned by the various *pygtfs.schedule* lists. Most of the attributes come directly from the gtfs reference. Also, when possible relations are taken into account, e.g. a *Route* class has a *trips* attribute, with a list of trips for the specific route.

class pygtfs.gtfs_entities.**Agency**(***kwargs*)

agency_email

agency_fare_url

agency_id

agency_lang

agency_name

agency_phone

agency_timezone

agency_url

feed_id

id

routes

class pygtfs.gtfs_entities.**Fare**(***kwargs*)

agency_id

currency_type

fare_id

feed_id

id

payment_method

price

transfer_duration

transfers

class pygtfs.gtfs_entities.**FareRule**(***kwargs*)

contains_id

destination_id

fare_id

feed_id

origin_id

route_id

class pygtfs.gtfs_entities.**Feed**(***kwargs*)

agencies

fare_rules

fares

feed_append_date

feed_id

feed_name

feedinfo

frequencies

id

routes

service_exceptions

services

shape_points

stop_times

stops

transfers

translations

trips

class pygtfs.gtfs_entities.**FeedInfo**(***kwargs*)

feed_end_date

feed_id

feed_lang

feed_publisher_name

feed_publisher_url

feed_start_date

feed_version

class pygtfs.gtfs_entities.**Frequency**(***kwargs*)

end_time

exact_times

feed_id

headway_secs

start_time

trip_id

class pygtfs.gtfs_entities.**Route**(***kwargs*)

agency_id

fare_rules

feed_id

id

route_color

route_desc

route_id

route_long_name

route_short_name

route_text_color

route_type

route_url

trips

valid_extended_route_types = [0, 1, 2, 3, 4, 5, 6, 7, 100, 101, 102, 103, 104, 105, 10

class pygtfs.gtfs_entities.**Service**(***kwargs*)

end_date

feed_id

friday

id

monday

saturday

service_id

start_date

sunday

thursday

trips

tuesday

wednesday

class pygtfs.gtfs_entities.**ServiceException**(***kwargs*)

date

exception_type

feed_id

id

service_id

class pygtfs.gtfs_entities.**ShapePoint**(***kwargs*)

feed_id

shape_dist_traveled

shape_id

shape_pt_lat

shape_pt_lon

shape_pt_sequence

trips

class pygtfs.gtfs_entities.**Stop**(***kwargs*)

feed_id

id

location_type

parent_station

platform_code

stop_code

stop_desc

stop_id

stop_lat

stop_lon

stop_name

stop_times

stop_timezone

stop_url

transfers_from

transfers_to

translations

wheelchair_boarding

zone_id

class pygtfs.gtfs_entities.**StopTime**(***kwargs*)

arrival_time

departure_time

drop_off_type

feed_id

pickup_type

shape_dist_traveled

stop_headsign

stop_id

stop_sequence

timepoint

trip_id

class pygtfs.gtfs_entities.**Transfer**(***kwargs*)

feed_id

from_stop_id

min_transfer_time

to_stop_id

transfer_type

class pygtfs.gtfs_entities.**Translation**(***kwargs*)

feed_id

lang

trans_id

translation

class pygtfs.gtfs_entities.**Trip**(***kwargs*)

bikes_allowed

block_id

direction_id

feed_id

frequencies

id

route_id

service_id

shape_id

stop_times

trip_headsign

trip_id

trip_short_name

wheelchair_accessible

pygtfs.gtfs_entities.**create_foreign_keys**(**key_names*)

Create foreign key constraints, always including feed_id, and relying on convention that key name is the same

5.4 The gtfs2db script

This is a script to manage the database. Here is its help message:

```
 gtfs2db - convert a gtfs feed to a pygtfs database

Usage:
  gtfs2db append <feed_file> <database> [--chunk-size <integer>]
  gtfs2db overwrite <feed_file> <database> [-i, --interactive] [--chunk-size <integer>
↪]
  gtfs2db delete <feed_file> <database> [-i, --interactive]
```

(continues on next page)

(continued from previous page)

```
  gtfs2db list <database>
  gtfs2db (-h | --help)
  gtfs2db --version

Options:
  -h --help          Show this help screen.
  --version          Show version.
  -i --interactive   Ask before deleting or overwriting existing feeds.
  --chunk-size <int> How often to flush database. If memory consumption is high,
                     lower this number. [default: 10000]
  <feed_file>        The gtfs file on which to operate. Can be either a folder
                     containing .txt files, or a .zip file.
  <database>         The database. Can be either a file, which is interpreted
                     as an sqlite database stored in this file, or a sqlalchemy
                     database connection.

Commands:
  append            appends the gtfs feed to the database
  overwrite         delete any existing feeds which had the same original
                    filename as the new file, and then append the new file.
  delete            delete from the database any feeds with the name supplied.
  list              list existing feeds in the database.

Description:
  This is a tool to manage a database containing several gtfs feeds. The
  database is in a pygtfs 0.1.0 format, and can be stored as any database
  supported by sqlalchemy (the default being sqlite).
  The database file can later be used to create a `pygtfs.Schedule` instance.
```

5.5 Internally used modules

class pygtfs.feed.**CSV**(*rows*, *feedtype='CSVTuple'*, *columns=None*)

A CSV file.

next()

class pygtfs.feed.**Feed**(*filename*, *strip_fields=True*)

A collection of CSV files with headers, either zipped into an archive or loose in a folder.

python2_reader(*filename*)

python3_reader(*filename*)

read_table(*filename*, *columns*)

pygtfs.feed.**derive_feed_name**(*filename*)

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

p

pygtfs.feed, 17
pygtfs.gtfs_entities, 11
pygtfs.loader, 11
pygtfs.schedule, 9

Index

A

agencies (pygtfs.gtfs_entities.Feed attribute), 12
agencies (pygtfs.schedule.Schedule attribute), 9
agencies_by_id() (pygtfs.schedule.Schedule method), 9
agencies_query (pygtfs.schedule.Schedule attribute), 9
Agency (class in pygtfs.gtfs_entities), 11
agency_email (pygtfs.gtfs_entities.Agency attribute), 11
agency_fare_url (pygtfs.gtfs_entities.Agency attribute), 11
agency_id (pygtfs.gtfs_entities.Agency attribute), 11
agency_id (pygtfs.gtfs_entities.Fare attribute), 12
agency_id (pygtfs.gtfs_entities.Route attribute), 13
agency_lang (pygtfs.gtfs_entities.Agency attribute), 11
agency_name (pygtfs.gtfs_entities.Agency attribute), 12
agency_phone (pygtfs.gtfs_entities.Agency attribute), 12
agency_timezone (pygtfs.gtfs_entities.Agency attribute), 12
agency_url (pygtfs.gtfs_entities.Agency attribute), 12
append_feed() (in module pygtfs.loader), 11
arrival_time (pygtfs.gtfs_entities.StopTime attribute), 15

B

bikes_allowed (pygtfs.gtfs_entities.Trip attribute), 16
block_id (pygtfs.gtfs_entities.Trip attribute), 16

C

contains_id (pygtfs.gtfs_entities.FareRule attribute), 12
create_foreign_keys() (in module pygtfs.gtfs_entities), 16
CSV (class in pygtfs.feed), 17
currency_type (pygtfs.gtfs_entities.Fare attribute), 12

D

date (pygtfs.gtfs_entities.ServiceException attribute), 14
delete_feed() (in module pygtfs.loader), 11
departure_time (pygtfs.gtfs_entities.StopTime attribute), 15
derive_feed_name() (in module pygtfs.feed), 17
destination_id (pygtfs.gtfs_entities.FareRule attribute), 12
direction_id (pygtfs.gtfs_entities.Trip attribute), 16
drop_feed() (pygtfs.schedule.Schedule method), 9
drop_off_type (pygtfs.gtfs_entities.StopTime attribute), 15

E

end_date (pygtfs.gtfs_entities.Service attribute), 14
end_time (pygtfs.gtfs_entities.Frequency attribute), 13
exact_times (pygtfs.gtfs_entities.Frequency attribute), 13
exception_type (pygtfs.gtfs_entities.ServiceException attribute), 14

F

Fare (class in pygtfs.gtfs_entities), 12
fare_id (pygtfs.gtfs_entities.Fare attribute), 12
fare_id (pygtfs.gtfs_entities.FareRule attribute), 12
fare_rules (pygtfs.gtfs_entities.Feed attribute), 12
fare_rules (pygtfs.gtfs_entities.Route attribute), 13
fare_rules (pygtfs.schedule.Schedule attribute), 9
fare_rules_query (pygtfs.schedule.Schedule attribute), 9
FareRule (class in pygtfs.gtfs_entities), 12
fares (pygtfs.gtfs_entities.Feed attribute), 12
fares (pygtfs.schedule.Schedule attribute), 9
fares_by_id() (pygtfs.schedule.Schedule method), 9
fares_query (pygtfs.schedule.Schedule attribute), 10
Feed (class in pygtfs.feed), 17
Feed (class in pygtfs.gtfs_entities), 12
feed_append_date (pygtfs.gtfs_entities.Feed attribute), 12
feed_end_date (pygtfs.gtfs_entities.FeedInfo attribute), 13
feed_id (pygtfs.gtfs_entities.Agency attribute), 12
feed_id (pygtfs.gtfs_entities.Fare attribute), 12
feed_id (pygtfs.gtfs_entities.FareRule attribute), 12
feed_id (pygtfs.gtfs_entities.Feed attribute), 12
feed_id (pygtfs.gtfs_entities.FeedInfo attribute), 13
feed_id (pygtfs.gtfs_entities.Frequency attribute), 13
feed_id (pygtfs.gtfs_entities.Route attribute), 13
feed_id (pygtfs.gtfs_entities.Service attribute), 14
feed_id (pygtfs.gtfs_entities.ServiceException attribute), 14

feed_id (pygtfs.gtfs_entities.ShapePoint attribute), 14
feed_id (pygtfs.gtfs_entities.Stop attribute), 15
feed_id (pygtfs.gtfs_entities.StopTime attribute), 15
feed_id (pygtfs.gtfs_entities.Transfer attribute), 15
feed_id (pygtfs.gtfs_entities.Translation attribute), 16
feed_id (pygtfs.gtfs_entities.Trip attribute), 16
feed_infos (pygtfs.schedule.Schedule attribute), 10
feed_infos_query (pygtfs.schedule.Schedule attribute), 10
feed_lang (pygtfs.gtfs_entities.FeedInfo attribute), 13
feed_name (pygtfs.gtfs_entities.Feed attribute), 12
feed_publisher_name (pygtfs.gtfs_entities.FeedInfo attribute), 13
feed_publisher_url (pygtfs.gtfs_entities.FeedInfo attribute), 13
feed_start_date (pygtfs.gtfs_entities.FeedInfo attribute), 13
feed_version (pygtfs.gtfs_entities.FeedInfo attribute), 13
FeedInfo (class in pygtfs.gtfs_entities), 13
feedinfo (pygtfs.gtfs_entities.Feed attribute), 12
feeds (pygtfs.schedule.Schedule attribute), 10
feeds_by_id() (pygtfs.schedule.Schedule method), 10
feeds_query (pygtfs.schedule.Schedule attribute), 10
frequencies (pygtfs.gtfs_entities.Feed attribute), 12
frequencies (pygtfs.gtfs_entities.Trip attribute), 16
frequencies (pygtfs.schedule.Schedule attribute), 10
frequencies_query (pygtfs.schedule.Schedule attribute), 10
Frequency (class in pygtfs.gtfs_entities), 13
friday (pygtfs.gtfs_entities.Service attribute), 14
from_stop_id (pygtfs.gtfs_entities.Transfer attribute), 15

H

headway_secs (pygtfs.gtfs_entities.Frequency attribute), 13

I

id (pygtfs.gtfs_entities.Agency attribute), 12
id (pygtfs.gtfs_entities.Fare attribute), 12
id (pygtfs.gtfs_entities.Feed attribute), 12
id (pygtfs.gtfs_entities.Route attribute), 13
id (pygtfs.gtfs_entities.Service attribute), 14
id (pygtfs.gtfs_entities.ServiceException attribute), 14
id (pygtfs.gtfs_entities.Stop attribute), 15
id (pygtfs.gtfs_entities.Trip attribute), 16

L

lang (pygtfs.gtfs_entities.Translation attribute), 16
list_feeds() (in module pygtfs.loader), 11
location_type (pygtfs.gtfs_entities.Stop attribute), 15

M

min_transfer_time (pygtfs.gtfs_entities.Transfer attribute), 16
monday (pygtfs.gtfs_entities.Service attribute), 14

N

next() (pygtfs.feed.CSV method), 17

O

origin_id (pygtfs.gtfs_entities.FareRule attribute), 12
overwrite_feed() (in module pygtfs.loader), 11

P

parent_station (pygtfs.gtfs_entities.Stop attribute), 15
payment_method (pygtfs.gtfs_entities.Fare attribute), 12
pickup_type (pygtfs.gtfs_entities.StopTime attribute), 15
platform_code (pygtfs.gtfs_entities.Stop attribute), 15
price (pygtfs.gtfs_entities.Fare attribute), 12
pygtfs.feed (module), 17
pygtfs.gtfs_entities (module), 11
pygtfs.loader (module), 11
pygtfs.schedule (module), 9
python2_reader() (pygtfs.feed.Feed method), 17
python3_reader() (pygtfs.feed.Feed method), 17

R

read_table() (pygtfs.feed.Feed method), 17
Route (class in pygtfs.gtfs_entities), 13
route_color (pygtfs.gtfs_entities.Route attribute), 13
route_desc (pygtfs.gtfs_entities.Route attribute), 13
route_id (pygtfs.gtfs_entities.FareRule attribute), 12
route_id (pygtfs.gtfs_entities.Route attribute), 13
route_id (pygtfs.gtfs_entities.Trip attribute), 16
route_long_name (pygtfs.gtfs_entities.Route attribute), 13
route_short_name (pygtfs.gtfs_entities.Route attribute), 13
route_text_color (pygtfs.gtfs_entities.Route attribute), 14
route_type (pygtfs.gtfs_entities.Route attribute), 14
route_url (pygtfs.gtfs_entities.Route attribute), 14
routes (pygtfs.gtfs_entities.Agency attribute), 12
routes (pygtfs.gtfs_entities.Feed attribute), 13
routes (pygtfs.schedule.Schedule attribute), 10
routes_by_id() (pygtfs.schedule.Schedule method), 10
routes_query (pygtfs.schedule.Schedule attribute), 10

S

saturday (pygtfs.gtfs_entities.Service attribute), 14
Schedule (class in pygtfs.schedule), 9
Service (class in pygtfs.gtfs_entities), 14
service_exceptions (pygtfs.gtfs_entities.Feed attribute), 13
service_exceptions (pygtfs.schedule.Schedule attribute), 10
service_exceptions_by_id() (pygtfs.schedule.Schedule method), 10
service_exceptions_query (pygtfs.schedule.Schedule attribute), 10

service_id (pygtfs.gtfs_entities.Service attribute), 14
service_id (pygtfs.gtfs_entities.ServiceException attribute), 14
service_id (pygtfs.gtfs_entities.Trip attribute), 16
ServiceException (class in pygtfs.gtfs_entities), 14
services (pygtfs.gtfs_entities.Feed attribute), 13
services (pygtfs.schedule.Schedule attribute), 10
services_by_id() (pygtfs.schedule.Schedule method), 10
services_query (pygtfs.schedule.Schedule attribute), 10
shape_dist_traveled (pygtfs.gtfs_entities.ShapePoint attribute), 14
shape_dist_traveled (pygtfs.gtfs_entities.StopTime attribute), 15
shape_id (pygtfs.gtfs_entities.ShapePoint attribute), 14
shape_id (pygtfs.gtfs_entities.Trip attribute), 16
shape_points (pygtfs.gtfs_entities.Feed attribute), 13
shape_pt_lat (pygtfs.gtfs_entities.ShapePoint attribute), 14
shape_pt_lon (pygtfs.gtfs_entities.ShapePoint attribute), 14
shape_pt_sequence (pygtfs.gtfs_entities.ShapePoint attribute), 14
ShapePoint (class in pygtfs.gtfs_entities), 14
shapes (pygtfs.schedule.Schedule attribute), 10
shapes_query (pygtfs.schedule.Schedule attribute), 10
start_date (pygtfs.gtfs_entities.Service attribute), 14
start_time (pygtfs.gtfs_entities.Frequency attribute), 13
Stop (class in pygtfs.gtfs_entities), 14
stop_code (pygtfs.gtfs_entities.Stop attribute), 15
stop_desc (pygtfs.gtfs_entities.Stop attribute), 15
stop_headsign (pygtfs.gtfs_entities.StopTime attribute), 15
stop_id (pygtfs.gtfs_entities.Stop attribute), 15
stop_id (pygtfs.gtfs_entities.StopTime attribute), 15
stop_lat (pygtfs.gtfs_entities.Stop attribute), 15
stop_lon (pygtfs.gtfs_entities.Stop attribute), 15
stop_name (pygtfs.gtfs_entities.Stop attribute), 15
stop_sequence (pygtfs.gtfs_entities.StopTime attribute), 15
stop_times (pygtfs.gtfs_entities.Feed attribute), 13
stop_times (pygtfs.gtfs_entities.Stop attribute), 15
stop_times (pygtfs.gtfs_entities.Trip attribute), 16
stop_times (pygtfs.schedule.Schedule attribute), 10
stop_times_query (pygtfs.schedule.Schedule attribute), 10
stop_timezone (pygtfs.gtfs_entities.Stop attribute), 15
stop_url (pygtfs.gtfs_entities.Stop attribute), 15
stops (pygtfs.gtfs_entities.Feed attribute), 13
stops (pygtfs.schedule.Schedule attribute), 11
stops_by_id() (pygtfs.schedule.Schedule method), 11
stops_query (pygtfs.schedule.Schedule attribute), 11
StopTime (class in pygtfs.gtfs_entities), 15
sunday (pygtfs.gtfs_entities.Service attribute), 14

T

thursday (pygtfs.gtfs_entities.Service attribute), 14
timepoint (pygtfs.gtfs_entities.StopTime attribute), 15
to_stop_id (pygtfs.gtfs_entities.Transfer attribute), 16
trans_id (pygtfs.gtfs_entities.Translation attribute), 16
Transfer (class in pygtfs.gtfs_entities), 15
transfer_duration (pygtfs.gtfs_entities.Fare attribute), 12
transfer_type (pygtfs.gtfs_entities.Transfer attribute), 16
transfers (pygtfs.gtfs_entities.Fare attribute), 12
transfers (pygtfs.gtfs_entities.Feed attribute), 13
transfers (pygtfs.schedule.Schedule attribute), 11
transfers_from (pygtfs.gtfs_entities.Stop attribute), 15
transfers_query (pygtfs.schedule.Schedule attribute), 11
transfers_to (pygtfs.gtfs_entities.Stop attribute), 15
Translation (class in pygtfs.gtfs_entities), 16
translation (pygtfs.gtfs_entities.Translation attribute), 16
translations (pygtfs.gtfs_entities.Feed attribute), 13
translations (pygtfs.gtfs_entities.Stop attribute), 15
translations (pygtfs.schedule.Schedule attribute), 11
translations_query (pygtfs.schedule.Schedule attribute), 11
Trip (class in pygtfs.gtfs_entities), 16
trip_headsign (pygtfs.gtfs_entities.Trip attribute), 16
trip_id (pygtfs.gtfs_entities.Frequency attribute), 13
trip_id (pygtfs.gtfs_entities.StopTime attribute), 15
trip_id (pygtfs.gtfs_entities.Trip attribute), 16
trip_short_name (pygtfs.gtfs_entities.Trip attribute), 16
trips (pygtfs.gtfs_entities.Feed attribute), 13
trips (pygtfs.gtfs_entities.Route attribute), 14
trips (pygtfs.gtfs_entities.Service attribute), 14
trips (pygtfs.gtfs_entities.ShapePoint attribute), 14
trips (pygtfs.schedule.Schedule attribute), 11
trips_by_id() (pygtfs.schedule.Schedule method), 11
trips_query (pygtfs.schedule.Schedule attribute), 11
tuesday (pygtfs.gtfs_entities.Service attribute), 14

V

valid_extended_route_types (pygtfs.gtfs_entities.Route attribute), 14

W

wednesday (pygtfs.gtfs_entities.Service attribute), 14
wheelchair_accessible (pygtfs.gtfs_entities.Trip attribute), 16
wheelchair_boarding (pygtfs.gtfs_entities.Stop attribute), 15

Z

zone_id (pygtfs.gtfs_entities.Stop attribute), 15