

Package ‘MBTAr’

November 12, 2018

Type Package

Title Access Data from the Massachusetts Bay Transit Authority (MBTA)
Web API

Version 2.0.0

BugReports <https://github.com/justindbk/MBTAr/issues>

Depends R (>= 3.4.0)

LazyData true

Description Access to the MBTA's performance API for R. Creates an easy-to-use bundle of functions to work with all the built-in calls to the MBTA performance API, and some to work with the GTFS-compatible API V3 which replaced some functionality from the old performance API. Allows users to download realtime tracking data in dataframe format that is manipulable in standard R analytics functions. Pulls performance statistics in both realtime and for historical dates and trips (after July 2015).

License GPL-3

Imports jsonlite

RoxygenNote 6.1.1

NeedsCompilation no

Author Justin de Benedictis-Kessner [aut, cre]

Maintainer Justin de Benedictis-Kessner <justindbk@gmail.com>

Repository CRAN

Date/Publication 2018-11-12 10:00:23 UTC

R topics documented:

| | |
|--------------------------------|----|
| routes | 2 |
| Talertbyid | 3 |
| Talerts | 5 |
| Tdailymetrics | 8 |
| Theadways | 10 |
| Theadwaysperformance | 11 |

| | |
|-------------------------------|----|
| Tmetrics | 12 |
| Tpredictionsbyroute | 14 |
| Tpredictionsbystop | 15 |
| Tpredictionsbytrip | 17 |
| Troutes | 19 |
| Troutesbystop | 20 |
| Tschedulebyroute | 21 |
| Tschedulebystop | 22 |
| Tschedulebytrip | 23 |
| Tservertime | 24 |
| Tstopsbylocation | 25 |
| Tstopsbyroute | 26 |
| Tstopslookup | 27 |
| Ttravelperformance | 28 |
| Ttraveltimes | 29 |
| Tvehiclesbyroute | 30 |
| Tvehiclesbytrip | 31 |

| | |
|--------------|-----------|
| Index | 33 |
|--------------|-----------|

| | |
|--------|--|
| routes | <i>Routes and respective information for the MBTA system</i> |
|--------|--|

Description

Routes and respective information for the MBTA system

Usage

routes

Format

A data frame with columns:

route_type A value between 0 and 4 indicating the type of vehicle use on this route. 0 is light rail, 1 is heavy rail, 2 is commuter rail, 3 is bus, and 4 is ferry.

mode_name Official name for route: Rapid Transit, Commuter Rail, Limited Service, Local Bus, Key Bus Route (Frequent Service), Express Bus, or Ferry.

route_id GTFS-compatible id (text or number) for route.

route_name Longform name for route (text).

direction_0 First direction of travel.

direction_1 Second direction of travel.

Source

From the MBTA GTFS API V3 in November 2018. <https://www.mbta.com/developers/gtfs>

Examples

```
## Not run:
  routes

## End(Not run)
```

Talertbyid

Queries active and upcoming times for a particular alert.

Description

Returns all information about a given alert. Some fields may be empty for an alert.

Usage

```
Talertbyid(alert_id)
```

Arguments

alert_id Unique identifier for the alert. Example: "781"

Value

| | |
|-------------------|---|
| alert_id | The unique identifier for the alert |
| effect_name | The human-readable name for the effect. Example: "Shuttle bus" |
| effect | The GTFS-realtime-compatible code for the effect. Example: "DETOUR" |
| cause | The human-readable name for the cause. Sometimes empty. Example: "maintenance" |
| header_text | A brief summary of the situation (GTFS-realtime-compatible). Example: "Shuttle buses replacing Red Line service from Sat Jun 01, 2013 to Sun Jun 30, 2013 every Saturday and Sunday from 09:00 PM to end of service due to tie replacement" |
| short_header_text | A shortened version of header_text. Example: "Shuttle buses replacing Red Line service from Sat Apr 27 to Sun May 26 every Saturday and Sunday due to maintenance" |
| description_text | Additional details (GTFS-realtime-compatible). Example: "Affected stops: Alewife Station Davis Station Porter Square Station Harvard Square Station" |
| severity | Possible values: "Severe", "Moderate", "Minor" |
| created_dt | Date and time the alert was created, in epoch time. Example: "1361395938" |
| last_modified_dt | Date and time the alert was last modified, in epoch time. Example: "1361395938" |

| | |
|-------------------------|---|
| service_effect_text | Summarizes the service and the impact to that service. Example: "Minor Route 1 delay" |
| timeframe_text | Summarizes when an alert is in effect. Example: "starting Saturday" |
| alert_lifecycle | Identifies whether alert is a new or old, in effect or upcoming. Not intended to be human-readable. Possible values: "Upcoming", "New", "Ongoing", "Ongoing-Upcoming." See notes. |
| effect_start | Date and time of the start of the effect period, in epoch time. Example: "1367110800" |
| effect_end | Date and time of the end of the effect period, in epoch time. Can be empty if effect end is not known. Example: "1367130600" |
| affected_route_type | GTFS-compatible code for route type (i.e. mode). Appears for service alerts only (not access alerts). Example: "1" |
| affected_mode_name | Human-readable name for the mode. Example: "Subway" |
| affected_route_id | The unique GTFS-compatible identifier for the route. Example: "Red" |
| affected_route_name | The human-readable name for the route. Example: "Red Line" |
| affected_direction_id | The GTFS-compatible identifier for the direction. Example: "0" |
| affected_direction_name | Human-readable direction name. Example: "Westbound" |
| affected_trip_id | The GTFS-compatible unique identifier for the trip. Example: "CR-Newburyport-CR-Weekday-129" |
| affected_trip_name | Human-readable trip name. Example: "129 (5:00 pm from North Station)" |
| affected_stop_id | The GTFS-compatible unique identifier for the stop. Example: "70061" |
| affected_stop_name | The GTFS-compatible name for the stop (not unique). Example: "Alewife Station Red Line" |
| affected_route_hide | Whether this route should be hidden from users. See notes. Possible values: "true". Only included if "true." |
| affected_elev_id | Unique identifier for the elevator/escalator. Example: "926" |
| affected_elev_name | Human-readable name for the elevator/escalator. Example: "SOUTH STATION - Lobby to Street" |
| affected_elev_type | Type of the elevator/escalator. Possible values: "Elevator", "Escalator", "Lift" |

| | |
|------------------------------|---|
| affected_elev_stop_id | The GTFS-compatible unique identifier for the stop. Example: "70080" |
| affected_elev_stop_name | The GTFS-compatible name for the stop (not unique). Example: "South Station - Inbound" |
| affected_elev_stop_parent_id | The GTFS-compatible unique identifier for the larger station associated with the stop, if one exists. Can be empty if parent station does not exist. Example: "place_sstat" |

See Also

[Talerts](#)

| | |
|---------|--|
| Talerts | <i>Query all active and upcoming alerts.</i> |
|---------|--|

Description

Find current list of all MBTA alerts, including all routes and stops they affect, along with timeframes that the alert is in effect. Returns all possible information for a given alert, often with empty fields.

Usage

Talerts()

Value

| | |
|-------------------|---|
| alert_id | The unique identifier for the alert |
| effect_name | The human-readable name for the effect. Example: "Shuttle bus" |
| effect | The GTFS-realtime-compatible code for the effect. Example: "DETOUR" |
| cause | The human-readable name for the cause. Seomtimes empty. Example: "maintenance" |
| header_text | A brief summary of the situation (GTFS-realtime-compatible). Example: "Shuttle buses replacing Red Line service from Sat Jun 01, 2013 to Sun Jun 30, 2013 every Saturday and Sunday from 09:00 PM to end of service due to tie replacement" |
| short_header_text | A shortened version of header_text. Example: "Shuttle buses replacing Red Line service from Sat Apr 27 to Sun May 26 every Saturday and Sunday due to maintenance" |
| description_text | Additional details (GTFS-realtime-compatible). Example: "Affected stops: Alewife Station Davis Station Porter Square Station Harvard Square Station" |
| severity | Possible values: "Severe", "Moderate", "Minor" |

| | |
|-------------------------|---|
| created_dt | Date and time the alert was created, in epoch time. Example: "1361395938" |
| last_modified_dt | Date and time the alert was last modified, in epoch time. Example: "1361395938" |
| service_effect_text | Summarizes the service and the impact to that service. Example: "Minor Route 1 delay" |
| timeframe_text | Summarizes when an alert is in effect. Example: "starting Saturday" |
| alert_lifecycle | Identifies whether alert is a new or old, in effect or upcoming. Not intended to be human-readable. Possible values: "Upcoming", "New", "Ongoing", "Ongoing-Upcoming." See notes. |
| effect_start | Date and time of the start of the effect period, in epoch time. Example: "1367110800" |
| effect_end | Date and time of the end of the effect period, in epoch time. Can be empty if effect end is not known. Example: "1367130600" |
| affected_route_type | GTFS-compatible code for route type (i.e. mode). Appears for service alerts only (not access alerts). Example: "1" |
| affected_mode_name | Human-readable name for the mode. Example: "Subway" |
| affected_route_id | The unique GTFS-compatible identifier for the route. Example: "Red" |
| affected_route_name | The human-readable name for the route. Example: "Red Line" |
| affected_direction_id | The GTFS-compatible identifier for the direction. Example: "0" |
| affected_direction_name | Human-readable direction name. Example: "Westbound" |
| affected_trip_id | The GTFS-compatible unique identifier for the trip. Example: "CR-Newburyport-CR-Weekday-129" |
| affected_trip_name | Human-readable trip name. Example: "129 (5:00 pm from North Station)" |
| affected_stop_id | The GTFS-compatible unique identifier for the stop. Example: "70061" |
| affected_stop_name | The GTFS-compatible name for the stop (not unique). Example: "Alewife Station Red Line" |
| affected_route_hide | Whether this route should be hidden from users. See notes. Possible values: "true". Only included if "true." |
| affected_elev_id | Unique identifier for the elevator/escalator. Example: "926" |
| affected_elev_name | Human-readable name for the elevator/escalator. Example: "SOUTH STATION - Lobby to Street" |

| | |
|------------------------------|---|
| affected_elev_type | Type of the elevator/escalator. Possible values: "Elevator", "Escalator", "Lift" |
| affected_elev_stop_id | The GTFS-compatible unique identifier for the stop. Example: "70080" |
| affected_elev_stop_name | The GTFS-compatible name for the stop (not unique). Example: "South Station - Inbound" |
| affected_elev_stop_parent_id | The GTFS-compatible unique identifier for the larger station associated with the stop, if one exists. Can be empty if parent station does not exist. Example: "place_sstat" |

Note

Severity:

- "Severity" was created with the intent that it could drive presentation of alerts in a variety of ways - ordering, coloring, icons - and not with the intent that the words "severity," "mild," "moderate," or "severe" would necessarily be shown directly to customers.

Effect Periods:

- More than one 'effect_period' object can be present.
- 'effect_end' can be empty if the end time for an alert is not known.

Alert Lifecycle:

- "New" and "Ongoing" refer to alerts that are in effect now.
- "Upcoming" and "Ongoing-Upcoming" refer to alerts that will be in effect in the future.
- "Ongoing" and "Ongoing-Upcoming" refer to alerts that are "old news," like a station that is closed and has been for weeks.
- An example of an "Ongoing-Upcoming" alert would be a shuttle that has been happening every weekend for a month so far (if you retrieve the data on a weekday.)

Affected Services:

- The affected services for an alert can include either services or elevators/escalators but NOT both. If the 'services' object is empty (i.e. 'service' objects are not present) then the 'elevators' object will not be empty (i.e. 'elevator' objects will be present) and vice versa.
- More than one 'service' object can be present.
- Different service objects can have different combinations of attributes. They may have just a mode and route (affects an entire route), or mode and stop (affects all service at the stop.) Or be much more specific - including mode, route, direction, trip, and stop, indicating that it applies to one scheduled stop on one trip.
- Currently, the system does not allow creation of an alert that applies to multiple elevators/escalators. Therefore, only one 'elevator' object can be present. This may change in the future.
- For alerts that apply to elevators/escalators, 'parent_station' and 'parent_station_name' properties on the 'stop' object can be empty if parent station does not exist.

See Also

[Talertbyid](#)

Tdailymetrics

Query daily performance statistics.

Description

Returns day-level summary performance statistics for a specified date or range of dates.

Usage

```
Tdailymetrics(route_id=NULL,
from_date=(Sys.Date()-8),
to_date=Sys.Date()-1,
api_key)
```

Arguments

| | |
|-----------|--|
| route_id | GTFS-compatible route_id value for which metrics should be returned. If not specified, will return metrics for all routes. |
| from_date | The start of the time period that the metrics should be returned for. Must be in YYYY-MM-DD format. Defaults to 8 days ago. |
| to_date | The end of the time period that the metrics should be returned for. Must be in YYYY-MM-DD format. Defaults to yesterday. |
| api_key | API key for MBTA API. To obtain one, visit the MBTA Developer Portal (https://mbta.com/developers/mbta-performance/) |

Value

| | |
|----------------|---|
| service_date | Date of data for this performance metric |
| route_id | Route of data for this performance metric |
| threshold_id | Reference number for this type of performance metric |
| threshold_type | Type of performance metric (wait time-based or travel time-based) |
| threshold_name | Text name for this type of performance metric |
| metric_result | Corresponding result for this performance metric (percentage of trips that met or performed better than this threshold) |

See Also

[Tmetrics](#)

Examples

```

## Authenticate:

mykey <- NULL
mykey <- assign("mykey", value = test_key, envir = .GlobalEnv)
## use your own key from http://realtime.mbta.com/Portal/ instead of the test key

Tdailymetrics(route_id="Red", # metrics for the Red Line
from_date=(Sys.Date()-8), # 8 days ago
to_date=Sys.Date()-1, # to yesterday (a full week of performance)
api_key = mykey) # user-defined API key

## Returns:
#
# service_date route_id   threshold_id threshold_type   threshold_name metric_result
# 1   2015-11-05   Red threshold_id_01   wait_time           Headway           0.8623
# 2   2015-11-05   Red threshold_id_02   wait_time           Big Gap           0.9398
# 3   2015-11-05   Red threshold_id_03   wait_time           2X Headway       0.9769
# 4   2015-11-05   Red threshold_id_04   travel_time delayed < 3 min.  0.9653
# 5   2015-11-05   Red threshold_id_05   travel_time delayed < 6 min.  0.996
# 6   2015-11-06   Red threshold_id_01   wait_time           Headway           0.864
# 7   2015-11-06   Red threshold_id_02   wait_time           Big Gap           0.9443
# 8   2015-11-06   Red threshold_id_03   wait_time           2X Headway       0.9814
# 9   2015-11-06   Red threshold_id_04   travel_time delayed < 3 min.  0.9549
# 10  2015-11-06   Red threshold_id_05   travel_time delayed < 6 min.  0.9928
# 11  2015-11-07   Red threshold_id_01   wait_time           Headway           0.7365
# 12  2015-11-07   Red threshold_id_02   wait_time           Big Gap           0.89
# 13  2015-11-07   Red threshold_id_03   wait_time           2X Headway       0.952
# 14  2015-11-07   Red threshold_id_04   travel_time delayed < 3 min.  0.9853
# 15  2015-11-07   Red threshold_id_05   travel_time delayed < 6 min.  0.9979
# 16  2015-11-08   Red threshold_id_01   wait_time           Headway           0.7808
# 17  2015-11-08   Red threshold_id_02   wait_time           Big Gap           0.8972
# 18  2015-11-08   Red threshold_id_03   wait_time           2X Headway       0.9582
# 19  2015-11-08   Red threshold_id_04   travel_time delayed < 3 min.  0.9997
# 20  2015-11-08   Red threshold_id_05   travel_time delayed < 6 min.  0.9999
# 21  2015-11-09   Red threshold_id_01   wait_time           Headway           0.8515
# 22  2015-11-09   Red threshold_id_02   wait_time           Big Gap           0.9344
# 23  2015-11-09   Red threshold_id_03   wait_time           2X Headway       0.9768
# 24  2015-11-09   Red threshold_id_04   travel_time delayed < 3 min.  0.9403
# 25  2015-11-09   Red threshold_id_05   travel_time delayed < 6 min.  0.977
# 26  2015-11-10   Red threshold_id_01   wait_time           Headway           0.8698
# 27  2015-11-10   Red threshold_id_02   wait_time           Big Gap           0.9479
# 28  2015-11-10   Red threshold_id_03   wait_time           2X Headway       0.9805
# 29  2015-11-10   Red threshold_id_04   travel_time delayed < 3 min.  0.9725
# 30  2015-11-10   Red threshold_id_05   travel_time delayed < 6 min.  0.9976
# 31  2015-11-11   Red threshold_id_01   wait_time           Headway           0.8854
# 32  2015-11-11   Red threshold_id_02   wait_time           Big Gap           0.9605
# 33  2015-11-11   Red threshold_id_03   wait_time           2X Headway       0.9874
# 34  2015-11-11   Red threshold_id_04   travel_time delayed < 3 min.  0.9856
# 35  2015-11-11   Red threshold_id_05   travel_time delayed < 6 min.  0.9995
# 36  2015-11-12   Red threshold_id_01   wait_time           Headway           0.8573
# 37  2015-11-12   Red threshold_id_02   wait_time           Big Gap           0.9399

```

| | | | | | |
|------|------------|---------------------|-------------|------------------|--------|
| # 38 | 2015-11-12 | Red threshold_id_03 | wait_time | 2X Headway | 0.9772 |
| # 39 | 2015-11-12 | Red threshold_id_04 | travel_time | delayed < 3 min. | 0.9315 |
| # 40 | 2015-11-12 | Red threshold_id_05 | travel_time | delayed < 6 min. | 0.9806 |

Theadways *To get headways and train arrive times for a given station/route/direction in a certain time window.. Returns a list of headways at an origin station during a particular time period on a particular route.*

Description

To get headways and train arrive times for a given station/route/direction in a certain time window.. Returns a list of headways at an origin station during a particular time period on a particular route.

Usage

```
Theadways(from_stop_id, to_stop_id = NULL, route_id = NULL,
          direction_id = NULL, from_datetime = (Sys.time() - 1800),
          to_datetime = Sys.time(), api_key)
```

Arguments

| | |
|---------------|---|
| from_stop_id | GTFS-compatible stop_id for the origin stop for which headways should be returned. |
| to_stop_id | GTFS-compatible stop_id for the destination stop for which headways should be returned. If empty will use departures from origin stop with any destination stop on that route (i.e. not distinguishing between branches). |
| route_id | GTFS-compatible route_id value for which headways should be returned. If this is not included, headways for all routes between the from and to stop will be provided. |
| direction_id | Direction of travel for headways to be returned. If empty, will rely on |
| from_datetime | The start of the time period that the headways (arrival time at the origin stop) should fall within; converts to epoch time. |
| to_datetime | The end of the time period that the headways (arrival time at the origin stop) should fall within; converts to epoch time. |
| api_key | API key for MBTA Performance API. To obtain one, visit the MBTA Developer Portal (https://mbta.com/developers/mbta-performance/) |

Value

Arrival times for vehicles at the particular origin stop during the time window requested.

| | |
|-----------|---|
| route_id | GTFS-compatible route identifier for which headways are returned. |
| direction | Direction id (0/1) for which headways are returned. |

`current_dep_dt` Next departure time from origin stop during the time window specified.
`previous_dep_dt` Previous departure time at destination stop during the time window specified.
`headway_time_sec` Actual headway for next arriving vehicle during time window specified, in seconds.
`benchmark_headway_time_sec` Benchmark headway for each trip, in seconds.
`threshold_01_flag` Binary variable for whether a given headway was above the first threshold for delayed headways.
`threshold_02_flag` Binary variable for whether a given headway was above the second threshold for delayed headways.
`threshold_03_flag` Binary variable for whether a given headway was above the third threshold for delayed headways.

Examples

```

Theadways(from_stop_id = 70007, # Jackson Square Northbound
to_stop_id = 70034, # Malden Southbound
route_id = NULL,
direction_id = 1, # northbound
from_datetime = as.POSIXct("2015-08-03 00:00"),
to_datetime = as.POSIXct("2015-08-03 23:59"),
api_key = "wX9NwuHnZU2To07GmGR9uw")
#'
  
```

Theadwaysperformance *Function to return performance data based on headways surrounding a given passenger's arrival time at a station on a given route.*

Description

Function to return performance data based on headways surrounding a given passenger's arrival time at a station on a given route.

Usage

```

Theadwaysperformance(enter_time, exit_time = NULL, enter_route_name,
  exit_route_name, enter_stop_name, exit_stop_name, data, api_key)
  
```

Arguments

| | |
|------------------|--|
| enter_time | Entrance time of passenger at station |
| exit_time | End of window in which to look for train arrivals. |
| enter_route_name | Route name for which headways should be returned. |
| exit_route_name | Route name of exit stop for which headways should be returned. |
| enter_stop_name | Stop name (alpha, non-numeric) designating which station at which train headways information should be returned. |
| exit_stop_name | Stop name (alpha, non-numeric) designating towards which station train headways information should be returned. |
| data | Dataframe with named variables given by other arguments. |
| api_key | API key for MBTA Performance API. To obtain one, visit the MBTA Developer Portal (https://mbta.com/developers/mbta-performance/) |

Value

| | |
|---------------------|--|
| headway_mean | Average headway during the time period given between 'enter_time' and 'exit_time'. |
| headway_bench | Average of benchmark headways during time period given between 'enter_time' and 'exit_time'. |
| headway_perf | Average delay (actual headway - benchmark headway) during the time period given between 'enter_time' and 'exit_time'. |
| next_train | Arrival time of next train at given station that is after the entrance time given with 'enter_time'. |
| prev_train | Arrival time of previous train to be at station on given route immediately prior to the entrance time given with 'enter_time'. |
| headway_guess | Difference (in seconds) between arrival time of next train and entrance time given with 'enter_time'. |
| headway_bench_guess | Benchmark headway time for the train arriving next after passenger's arrival. Divide by two to find benchmark wait time. |

Tmetrics

Query real-time performance statistics.

Description

Returns real-time performance statistics for an origin-destination pair or systemwide for a specified time range.

Usage

```
Tmetrics(from_stop_id = NULL, to_stop_id = NULL, route_id = NULL, direction = NULL,
from_datetime = (Sys.time() - 3600), to_datetime = Sys.time(), api_key)
```

Arguments

| | |
|---------------|--|
| from_stop_id | GTFS-compatible stop_id for the origin stop for which metrics should be returned. If not specified, will return metrics for all routes. |
| to_stop_id | GTFS-compatible stop_id for the destination stop for which metrics should be returned. |
| route_id | GTFS-compatible route_id value for which metrics should be returned. If not specified, will return metrics for all routes. |
| direction | GTFS-compatible direction value (1 or 0) for which metrics in a particular direction should be returned. If this is not included, metrics will be aggregated for all directions. |
| from_datetime | The start of the time period that the metrics should be aggregated for; converts to epoch time. Defaults to 30 minutes ago. |
| to_datetime | The end of the time period that the metrics should be aggregated for; converts to epoch time. Defaults to now (system time). |
| api_key | API key for MBTA API. To obtain one, visit the MBTA Developer Portal (https://mbta.com/developers/mbta-performance) |

Value

| | |
|----------------|---|
| threshold_id | Reference number for this type of performance metric |
| threshold_type | Type of performance metric (wait time-based or travel time-based) |
| threshold_name | Text name for this type of performance metric |
| metric_result | Corresponding result for this performance metric (percentage of trips that met or performed better than this threshold) |

See Also

[Tdailymetrics](#)

Examples

```
## Authenticate:

mykey <- NULL
mykey <- assign("mykey", value = test_key, envir = .GlobalEnv)
## use your own key from http://realtime.mbta.com/Portal/ instead of the test key

Tmetrics(from_stop_id = NULL, # entire line
to_stop_id = NULL, # entire line
route_id = "Red", # Red Line
direction = 0, # southbound only
from_datetime = (Sys.time() - 3600), # 30 minutes ago
```

```

    to_datetime = Sys.time(), # now
    api_key = mykey) # user-defined API key

## Returns:
#   threshold_id threshold_type  threshold_name metric_result
# 1 threshold_id_01    wait_time      Headway          0.8395
# 2 threshold_id_02    wait_time      Big Gap           0.9395
# 3 threshold_id_03    wait_time      2X Headway        0.9809
# 4 threshold_id_04    travel_time    delayed < 3 min.  0.9792
# 5 threshold_id_05    travel_time    delayed < 6 min.  0.9963

```

Tpredictionsbyroute *Query predictions by route*

Description

Returns predicted upcoming arrivals and departures in the next hour for a particular route.

Usage

```

Tpredictionsbyroute(route_id,
  include_access_alerts = FALSE, include_service_alerts = TRUE, api_key)

```

Arguments

`route_id` GTFS-compatible `route_id` value for which predictions should be returned. Example: "Orange"

`include_access_alerts` Whether or not alerts pertaining to accessibility (elevators, escalators) should be returned. Data type: logical. Defaults to FALSE.

`include_service_alerts` Whether or not service alerts should be returned. Data type: logical. Defaults to TRUE.

`api_key` API key for MBTA API. To obtain one, visit the MBTA Developer Portal (<http://realtime.mbtta.com/Portal/>)

Value

`route_id` The unique GTFS-compatible identifier for the route for which predictions are returned. Example: "CR-Franklin"

`route_name` The human-readable name for the route for which predictions are returned. Example: "Franklin Line"

`route_type` The GTFS-compatible identifier for the type of service (mode). Example: "2"

`mode_name` The human-readable name for the type of service (mode). Example: "Commuter Rail"

| | |
|-------------------|--|
| direction_id | The GTFS-compatible identifier for the direction. Example: "0" |
| direction_name | The human-readable name for the direction. Example: "Outbound" |
| trip_id | The unique GTFS-compatible identifier for the trip. Example: "CR-Providence-CR-Weekday-815" |
| trip_name | The human-readable name for the trip. Example: "815 (4:35 pm from South Station)" |
| trip_headsign | The text that identifies the trip's destination to passengers. Example: "North Station" |
| vehicle_id | The GTFS-compatible unique identifier for the vehicle. Example: "1531" |
| vehicle_lat | The GTFS-compatible latitude of the vehicle. Example: "42.08997" |
| vehicle_lon | The GTFS-compatible longitude of the vehicle. Example: "-71.4388" |
| vehicle_bearing | GTFS-compatible bearing of the vehicle. This can be the compass bearing, or the direction towards the next stop or intermediate location. May be empty. Example: "259" |
| vehicle_speed | Identifies the vehicle's momentary speed, in meters per second. Example: "21" |
| vehicle_timestamp | Identifies the moment when the content of this feed has been created, in epoch time. Example: "1400855704" |
| stop_sequence | Identifies where the stop comes in the sequence of stops for this trip. Example: "2" |
| stop_id | The GTFS-compatible unique identifier for the stop. Example: "Back Bay" |
| stop_name | The GTFS-compatible name for the stop. Example: "Back Bay" |
| sch_arr_dt | Scheduled arrival time at the stop for the trip, in epoch time. Example: "1361986080" |
| sch_dep_dt | Scheduled departure time at the stop for the trip, in epoch time. Example: "1361986080" |
| pre_dt | Predicted time at the stop - departure time for origin stop and arrival time for all other stops - for the trip, in epoch time. Example: "1400855700" |
| pre_away | Predicted amount of time until the vehicle arrives at the stop, in seconds. Example: "339" |

Tpredictionsbystop *Query predictions by stop*

Description

Returns arrivals and departures in the next hour for a direction and route for a given stop.

Usage

```
Tpredictionsbystop(stop_id,
include_access_alerts = FALSE, include_service_alerts = TRUE, api_key)
```

Arguments

| | |
|------------------------|--|
| stop_id | GTFS-compatible stop_id value for which predictions should be returned. Example: "Back Bay" |
| include_access_alerts | Whether or not alerts pertaining to accessibility (elevators, escalators) should be returned. Data type: logical. Defaults to FALSE. |
| include_service_alerts | Whether or not service alerts should be returned. Data type: logical. Defaults to TRUE. |
| api_key | API key for MBTA API. To obtain one, visit the MBTA Developer Portal (http://realtime.mbta.com/Portal/) |

Value

| | |
|-------------------|--|
| stop_id | The GTFS-compatible unique identifier for the stop for which the predictions are returned. Example: "Back Bay" |
| stop_name | The GTFS-compatible name for the stop for which the predictions are returned. Example: "Back Bay" |
| route_type | The GTFS-compatible identifier for the type of service (mode). Example: "2" |
| mode_name | The human-readable name for the type of service (mode). Example: "Commuter Rail" |
| route_id | The unique GTFS-compatible identifier for the route. Example: "Red" |
| route_name | The human-readable name for the route. Example: "Red Line" |
| direction_id | The GTFS-compatible identifier for the direction. Example: "0" |
| direction_name | The human-readable name for the direction. Example: "Outbound" |
| trip_id | The unique GTFS-compatible identifier for the trip. Example: "CR-Providence-CR-Weekday-815" |
| trip_name | The human-readable name for the trip. Example: "815 (4:35 pm from South Station)" |
| trip_headsign | The text that identifies the trip's destination to passengers. Example: "North Station" |
| vehicle_id | The GTFS-compatible unique identifier for the vehicle. Example: "1531" |
| vehicle_lat | The GTFS-compatible latitude of the vehicle. Example: "42.08997" |
| vehicle_lon | The GTFS-compatible longitude of the vehicle. Example: "-71.4388" |
| vehicle_bearing | GTFS-compatible bearing of the vehicle. This can be the compass bearing, or the direction towards the next stop or intermediate location. May be empty. Example: "259" |
| vehicle_speed | Identifies the vehicle's momentary speed, in meters per second. Example: "21" |
| vehicle_timestamp | Identifies the moment when the content of this feed has been created, in epoch time. Example: "1400855704" |

| | |
|---------------|---|
| stop_sequence | Identifies where the stop comes in the sequence of stops for this trip. Example: "2" |
| stop_id | The GTFS-compatible unique identifier for the stop. Example: "Back Bay" |
| stop_name | The GTFS-compatible name for the stop. Example: "Back Bay" |
| sch_arr_dt | Scheduled arrival time at the stop for the trip, in epoch time. Example: "1361986080" |
| sch_dep_dt | Scheduled departure time at the stop for the trip, in epoch time. Example: "1361986080" |
| pre_dt | Predicted time at the stop - departure time for origin stop and arrival time for all other stops - for the trip, in epoch time. Example: "1400855700" |
| pre_away | Predicted amount of time until the vehicle arrives at the stop, in seconds. Example: "339" |

See Also

[Tpredictionsbyroute](#)

Tpredictionsbytrip *Query predictions by trip*

Description

Returns the predicted arrival and departure times for a given trip.

Usage

```
Tpredictionsbytrip(trip_id, api_key)
```

Arguments

| | |
|---------|--|
| trip_id | GTFS-compatible trip_id value for which vehicle positions should be returned. Data type: String. Example: "CR-Providence-CR-Weekday-807" |
| api_key | API key for MBTA API. To obtain one, visit the MBTA Developer Portal (http://realtime.mbtta.com/Portal/) |

Value

| | |
|------------|--|
| route_id | The unique GTFS-compatible identifier for the route for which vehicle positions are returned. Example: "CR-Providence" |
| route_name | The human-readable name for the route for which vehicle positions are returned. Example: "Providence/Stoughton Line" |
| route_type | The GTFS-compatible identifier for the type of service (mode). Example: "2" |
| mode_name | The human-readable name for the type of service (mode). Example: "Commuter Rail" |

| | |
|--------------------------------|--|
| <code>trip_id</code> | The unique GTFS-compatible identifier for the trip for which vehicle positions are returned. Example: "CR-Providence-CR-Weekday-815" |
| <code>trip_name</code> | The human-readable for the trip for which schedule is returned. Example: "815 (4:35 pm from South Station)" |
| <code>trip_headsign</code> | The text that identifies the trip's destination to passengers. Example: "North Station" |
| <code>direction_id</code> | The GTFS-compatible identifier for the direction. Example: "0" |
| <code>direction_name</code> | The human-readable name for the direction. Example: "Outbound" |
| <code>vehicle_id</code> | The GTFS-compatible unique identifier for the vehicle. Example: "1531" |
| <code>vehicle_lat</code> | The GTFS-compatible latitude of the vehicle. Example: "42.08997" |
| <code>vehicle_lon</code> | The GTFS-compatible longitude of the vehicle. Example: "-71.4388" |
| <code>vehicle_bearing</code> | GTFS-compatible bearing of the vehicle. This can be the compass bearing, or the direction towards the next stop or intermediate location. May be empty. Example: "259" |
| <code>vehicle_speed</code> | Identifies the vehicle's momentary speed, in meters per second. Example: "21" |
| <code>vehicle_timestamp</code> | Identifies the moment when the content of this feed has been created, in epoch time. Example: "1400855704" |
| <code>stop_sequence</code> | Identifies where the stop comes in the sequence of stops for this trip. Example: "2" |
| <code>stop_id</code> | The GTFS-compatible unique identifier for the stop. Example: "Back Bay" |
| <code>stop_name</code> | The GTFS-compatible name for the stop. Example: "Back Bay" |
| <code>sch_arr_dt</code> | Scheduled arrival time at the stop for the trip, in epoch time. Example: "1361986080" |
| <code>sch_dep_dt</code> | Scheduled departure time at the stop for the trip, in epoch time. Example: "1361986080" |
| <code>pre_dt</code> | Predicted time at the stop - departure time for origin stop and arrival time for all other stops - for the trip, in epoch time. Example: "1400855700" |
| <code>pre_away</code> | Predicted amount of time until the vehicle arrives at the stop, in seconds. Example: "339" |

See Also

[Tpredictionsbyroute](#) [Tpredictionsbystop](#)

Troutes *Query all T routes*

Description

Returns information about all routes for which information can be requested.

Usage

```
Troutes(api_key)
```

Arguments

`api_key` API key for MBTA API. To obtain one, visit the MBTA Developer Portal (<http://realtime.mbta.com/Portal/>)

Value

| | |
|-------------------------|--|
| <code>route_type</code> | The GTFS-compatible identifier for the type of service (mode). Example: "2" |
| <code>mode_name</code> | The human-readable name for the type of service (mode). Example: "Commuter Rail" |
| <code>route_id</code> | The unique GTFS-compatible identifier for the route. Example: "CR-Providence" |
| <code>route_name</code> | The human-readable name for the route. Example: "Providence/Stoughton Line" |

See Also

[Troutesbystop](#)

Examples

```
## Authenticate:

mykey <- NULL
mykey <- assign("mykey", value = test_key, envir = .GlobalEnv)
## use your own key from http://realtime.mbta.com/Portal/ instead of the test key

Troutes(api_key = mykey)

## returns:
# route_type mode_name route_id route_name
# 1 0 Subway Green-B Green Line B
# 2 0 Subway Green-C Green Line C
# 3 0 Subway Green-D Green Line D
# 4 0 Subway Green-E Green Line E
# 5 0 Subway Mattapan Mattapan Trolley
# 6 1 Subway Blue Blue Line
# 7 1 Subway Orange Orange Line
# 8 1 Subway Red Red Line
```

| | | | | |
|-------|---|---------------|--------------|-------------------|
| # 9 | 2 | Commuter Rail | CR-Fairmount | Fairmount Line |
| # 10 | 2 | Commuter Rail | CR-Fitchburg | Fitchburg Line |
| # ... | | | | |
| # 207 | 3 | Bus | 9702 | 9702 |
| # 208 | 3 | Bus | 9703 | 9703 |
| # 209 | 4 | Boat | Boat-F4 | Charlestown Ferry |
| # 210 | 4 | Boat | Boat-F1 | Hingham Ferry |
| # 211 | 4 | Boat | Boat-F3 | Hull Ferry |

Troutesbystop

Query all routes at a stop

Description

Returns routes that serve a particular stop.

Usage

```
Troutesbystop(stop_id = NULL, api_key)
```

Arguments

| | |
|---------|--|
| stop_id | GTFS-compatible stop_id value for which routes should be returned. Example: "70065" |
| api_key | API key for MBTA API. To obtain one, visit the MBTA Developer Portal (http://realtime.mbta.com/Portal/) |

Value

| | |
|------------|---|
| stop_id | The GTFS-compatible unique identifier for the stop for which routes are returned. Example: "70065" |
| stop_name | The GTFS-compatible name for the stop for which routes are returned. Example: "Porter Sq - Inbound" |
| route_type | The GTFS-compatible identifier for the type of service (mode). Example: "2" |
| mode_name | The human-readable name for the type of service (mode). Example: "Commuter Rail" |
| route_id | The unique GTFS-compatible identifier for the route. Example: "CR-Providence" |
| route_name | The human-readable name for the route. Example: "Providence/Stoughton Line" |

See Also

[Troutes](#)

Tschedulebyroute *Query the schedule by route*

Description

Returns the scheduled arrivals and departures in a direction for a particular route.

Usage

```
Tschedulebyroute(route_id, direction = NULL,
datetime = Sys.time(), max_time = 60, max_trips = 5, api_key)
```

Arguments

| | |
|-----------|---|
| route_id | GTFS-compatible route_id value for which schedule should be returned. Example: "CR-Providence" |
| direction | GTFS-compatible direction_id value on route for which schedule should be returned. If not included then schedule for all directions of the route will be returned. Example: "0" |
| datetime | Epoch time after which schedule should be returned. If included then must be within the next seven (7) days. If not included then schedule starting from the current datetime will be returned. Example: "1361989200" |
| max_time | Defines maximum range of time (in minutes) within which trips will be returned. Integer between 1 and 1440 (24 hours). If not included defaults to 60. Example: "120" |
| max_trips | Defines number of trips to return. Integer between 1 and 100. If not included defaults to 5. Example: "100" |
| api_key | API key for MBTA API. To obtain one, visit the MBTA Developer Portal (http://realtime.mbta.com/Portal/) |

Value

| | |
|----------------|--|
| route_id | The unique GTFS-compatible identifier for the route for which the schedule is returned. Example: "CR-Providence" |
| route_name | The human-readable name for the route for which the schedule is returned. Example: "Providence/Stoughton Line" |
| direction_id | The GTFS-compatible identifier for the direction. Example: "0" |
| direction_name | The human-readable name for the direction. Example: "Outbound" |
| trip_id | The unique GTFS-compatible identifier for the trip. Example: "CR-Providence-CR-Weekday-815" |
| trip_name | The human-readable name for the trip. Example: "815 (4:35 pm from South Station)" |
| stop_sequence | Identifies where the stop comes in the sequence of stops for this trip. Example: "2" |

| | |
|------------|---|
| stop_id | The GTFS-compatible unique identifier for the stop. Example: "Back Bay" |
| stop_name | The GTFS-compatible name for the stop. Example: "Back Bay" |
| sch_arr_dt | Scheduled arrival time at the stop for the trip, in epoch time. Example: "1361986080" |
| sch_dep_dt | Scheduled departure time at the stop for the trip, in epoch time. Example: "1361986080" |

See Also

[Tpredictionsbyroute](#) [Tschedulebystop](#) [Tschedulebytrip](#)

Tschedulebystop *Query schedule by stop*

Description

Returns scheduled arrival and departure times for a direction and route for a particular stop.

Usage

```
Tschedulebystop(stop_id, route_id,
direction = NULL, datetime = Sys.time(), max_time = 60, max_trips = 5,
api_key)
```

Arguments

| | |
|-----------|--|
| stop_id | GTFS-compatible stop_id value for which schedule should be returned. Example: "Back Bay" |
| route_id | GTFS-compatible route_id value on the stop for which schedule should be returned. If not included then schedule for all routes serving the stop will be returned. Example: "CR-Providence" |
| direction | GTFS-compatible direction_id value on route of the stop for which schedule should be returned. Bit (0 or 1). If included then route must also be included. If not included then schedule for all directions of the route serving the stop will be returned. Example: "0" |
| datetime | Epoch time after which schedule should be returned. If included then must be within the next seven (7) days. If not included then schedule starting from the current datetime will be returned. Example: "1361989200" |
| max_time | Defines maximum range of time (in minutes) within which trips will be returned. Integer between 1 and 1440 (24 hours). If not included defaults to 60. Example: "120" |
| max_trips | Defines number of trips to return. Integer between 1 and 100. If not included defaults to 5. Example: "100" |
| api_key | API key for MBTA API. To obtain one, visit the MBTA Developer Portal (http://realtime.mbta.com/Portal/) |

Value

| | |
|----------------|--|
| stop_id | The GTFS-compatible unique identifier for the stop for which the schedule is returned. Example: "Back Bay" |
| stop_name | The GTFS-compatible name for the stop for which the schedule is returned. Example: "Back Bay" |
| route_type | The GTFS-compatible identifier for the type of service (mode). Example: "2" |
| mode_name | The human-readable name for the type of service (mode). Example: "Commuter Rail" |
| route_id | The unique GTFS-compatible identifier for the route. Example: "CR-Providence" |
| route_name | The human-readable name for the route. Example: "Providence/Stoughton Line" |
| direction_id | The GTFS-compatible identifier for the direction. Example: "0" |
| direction_name | The human-readable name for the direction. Example: "Outbound" |
| trip_id | The unique GTFS-compatible identifier for the trip. Example: "CR-Providence-CR-Weekday-815" |
| trip_name | The human-readable name for the trip. Example: "815 (4:35 pm from South Station)" |
| sch_arr_dt | Scheduled arrival time at the stop for the trip, in epoch time. Example: "1361986080" |
| sch_dep_dt | Scheduled departure time at the stop for the trip, in epoch time. Example: "1361986080" |

Note

If the GTFS-compatible stop_id value in the stop parameter in the request is for a parent station then all routes that serve that parent station are returned.

See Also

[Tpredictionsbystop](#) [Tschedulebyroute](#) [Tschedulebytrip](#)

Tschedulebytrip

Query schedule by trip

Description

Returns scheduled arrival and departure times for a particular trip.

Usage

```
Tschedulebytrip(trip_id, datetime = Sys.time(), api_key)
```

Arguments

| | |
|----------|---|
| trip_id | GTFS-compatible trip_id value for which schedule should be returned. Example: "CR-Providence-CR-Weekday-807" |
| datetime | Epoch time after which schedule should be returned. If included then must be within the next seven (7) days. If not included then schedule starting from the current datetime will be returned, using Sys.time converted to epoch time. Example: "1361989200" |
| api_key | API key for MBTA API. To obtain one, visit the MBTA Developer Portal (http://realtime.mbtta.com/Portal/) |

Value

| | |
|----------------|--|
| route_id | The unique GTFS-compatible identifier for the route for which schedule is returned. Example: "CR-Providence" |
| route_name | The human-readable name for the route for which schedule is returned. Example: "Providence/Stoughton Line" |
| trip_id | The unique GTFS-compatible identifier for the trip for which schedule is returned. Example: "CR-Providence-CR-Weekday-815" |
| trip_name | The human-readable for the trip for which schedule is returned. Example: "815 (4:35 pm from South Station)" |
| direction_id | The GTFS-compatible identifier for the direction. Example: "0" |
| direction_name | The human-readable name for the direction. Example: "Outbound" |
| stop_sequence | Identifies where the stop comes in the sequence of stops for this trip. Example: "2" |
| stop_id | The GTFS-compatible unique identifier for the stop. Example: "Back Bay" |
| stop_name | The GTFS-compatible name for the stop. Example: "Back Bay" |
| sch_arr_dt | Scheduled arrival time at the stop for the trip, in epoch time. Example: "1361986080" |
| sch_dep_dt | Scheduled departure time at the stop for the trip, in epoch time. Example: "1361986080" |

See Also

[Tpredictionsbytrip](#) [Tschedulebystop](#) [Tschedulebyroute](#)

Tservertime

Queries the current MBTA API server time.

Description

Returns the server time. Useful if needing to sync user computer time with the epoch time of MBTA events.

Usage

```
Tservertime(api_key)
```

Arguments

api_key API key for MBTA API. To obtain one, visit the MBTA Developer Portal (<http://realtime.mbta.com/Portal/>)

Value

server_dt Server time, in epoch time. Example: "1361996667"

Tstopsbylocation *Query stops by geographic location*

Description

Returns information about the nearest stops to a particular location.

Usage

```
Tstopsbylocation(lat, lon, radius = 0.01)
```

Arguments

lat The latitude for location near which stops should be returned. Example: "42.352913"

lon The longitude for location near which stops should be returned. Example: "-71.064648"

radius Radius in degrees around the coordinates in which to search. Defaults to 0.01 degrees, or approximately half a mile.

Value

stop_id The GTFS-compatible unique identifier for the stop. Example: "70063"

address Street address or intersection at which stop is located. (note: can be empty for non-parent stations).

description Full description of stop or entrance/exit.

latitude The GTFS-compatible latitude of the station. Example: "42.3967399597168"

location_type Takes value of 0, 1, or 2, corresponding to stops, parent stations, and entrances/exits, respectively.

longitude The GTFS-compatible longitude of the station. Example: "-71.1218185424805"

name The GTFS-compatible name for the stop (not unique). Example: "State Street"

platform_code Number, 1 through 13, indicating platform within station serving commuter rail.

platform_name Direction towards which vehicles from that platform go.

| | |
|---------------------|--|
| wheelchair_boarding | Code for whether or not platform allows for wheelchair access. tkaes values of 0, 1, or 2, corresponding to no information, accessible, or inaccessible, respectively. |
| parent_station | The GTFS-compatible unique identifier for the station associated with the stop. (note: can be empty if stop does not have an associated station). Example: "place-davis" |

Tstopsbyroute *Query stops by route*

Description

Returns the stops for a particular route, specified by id or by name.

Usage

Tstopsbyroute(route_id = NULL, route_name = NULL)

Arguments

| | |
|------------|---|
| route_id | GTFS-compatible route_id value for which stops should be returned. Example: "Red" |
| route_name | The full human- readable name for the route for which stops are returned. Example: "Red Line" |

Value

| | |
|---------------------|--|
| stop_id | The GTFS-compatible unique identifier for the stop. Example: "70063" |
| address | Street address or intersection at which stop is located. (note: can be empty for non-parent stations). |
| description | Full description of stop or entrance/exit. |
| latitude | The GTFS-compatible latitude of the station. Example: "42.3967399597168" |
| location_type | Takes value of 0, 1, or 2, corresponding to stops, parent stations, and entrances/exits, respectively. |
| longitude | The GTFS- compatible longitude of the station. Example: "-71.1218185424805" |
| name | The GTFS-compatible name for the stop (not unique). Example: "State Street" |
| platform_code | Number, 1 through 13, indicating platform within station serving commuter rail. |
| platform_name | Direction towards which vehicles from that platform go. |
| wheelchair_boarding | Code for whether or not platform allows for wheelchair access. tkaes values of 0, 1, or 2, corresponding to no information, accessible, or inaccessible, respectively. |
| parent_station | The GTFS-compatible unique identifier for the station associated with the stop. (note: can be empty if stop does not have an associated station). Example: "place-davis" |

| | |
|--------------|--|
| Tstopslookup | <i>Search for information on a stop with a given name. Useful when stop_id is not known.</i> |
|--------------|--|

Description

Search for information on a stop with a given name. Useful when stop_id is not known.

Usage

```
Tstopslookup(stop_name)
```

Arguments

| | |
|-----------|-----------------------------|
| stop_name | Name of stop to search for. |
|-----------|-----------------------------|

Value

Dataframe with row for each potential matching stop and the following columns:

| | |
|---------------------|--|
| route_id | GTFS-compatible route_id for the stop match. |
| direction_id | Direction ID (0/1) |
| direction_name | Human-readable direction identifier |
| stop_order | Order in given direction of stop |
| stop_id | GTFS-compatible ID of stop |
| stop_name | Name of stop |
| parent_station | Parent station GTFS-compatible ID |
| parent_station_name | Parent station name |
| stop_lat | Stop latitude |
| stop_lon | Stop longitude |

Examples

```
## Not run:
Tstopslookup(stop_name = "Mattapan") # this would take a long time (~70 seconds) so is left out here

## End(Not run)
```

Ttravelperformance *Query performance statistics for many trips contained in a dataframe.*

Description

Returns travel time performance statistics for many sets of trips with specified origin-destination pairs and specific time periods.

Usage

```
Ttravelperformance(enter_time, exit_time = NULL, enter_route_name, exit_route_name,
enter_stop_name, exit_stop_name, data, api_key)
```

Arguments

| | |
|------------------|--|
| enter_time | Name of column in data that specifies entrance times for the trips for which performance is to be returned. |
| exit_time | Name of column in data that specifies exit times for the trips for which performance is to be returned. Defaults to 30 minutes from the enter_time value. |
| enter_route_name | Name of column in data that specifies GTFS-compatible route ID for entrance stations. |
| exit_route_name | Name of column in data that specifies GTFS-compatible route ID for exit stations. Travel time API request only available for single unlinked rides as of Release v2. |
| enter_stop_name | Name of column in data that specifies human-readable text name for the stop where trips began (origin name). |
| exit_stop_name | Name of column in data that specifies human-readable text name for the stop where trips ended (destination name). |
| data | Dataframe from which holds the previously named arguments. |
| api_key | API key for MBTA API. To obtain one, visit the MBTA Developer Portal (https://mbta.com/developers/mbta-performance/) |

Value

Dataframe containing the following:

| | |
|------------------|--|
| traveltime_mean | Mean travel time between origin and destination stops during the time period specified. |
| traveltime_bench | Benchmark travel times between origin and destination stops during the time period specified. Benchmarks come from historical data over the past 30 days for travel times. |
| traveltime_perf | Average travel time as a proportion of the benchmark time, or <code>traveltime_mean/traveltime_bench</code> . |

See Also[Ttraveltimes](#)

Ttraveltimes*Query travel times and travel time performance*

Description

Returns a list of travel times between an origin destination pair during a particular time period.

Usage

```
Ttraveltimes(from_stop_id, to_stop_id, route_id = NULL,
             from_datetime = (Sys.time() - 3600),
             to_datetime = Sys.time(),
             api_key)
```

Arguments

| | |
|---------------|--|
| from_stop_id | GTFS-compatible stop_id for the origin stop for which travel times should be returned. |
| to_stop_id | GTFS-compatible stop_id for the destination stop for which travel times should be returned. |
| route_id | GTFS-compatible route_id value for which travel times should be returned. If this is not included, travel times for all routes between the from and to stop will be provided. |
| from_datetime | The start of the time period that the travel time (arrival time at the destination stop) should fall within; converts to epoch time. |
| to_datetime | The end of the time period that the travel time (arrival time at the destination stop) should fall within; converts to epoch time. |
| api_key | API key for MBTA API. To obtain one, visit the MBTA Developer Portal (https://mbta.com/developers/mbta-performance) |

Value

| | |
|---------------------------|--|
| route_id | Route ID for which travel time are returned. |
| direction | Direction id for which travel times are returned. |
| dep_dt | Actual departure times from origin stop during the time window specified. |
| arr_dt | Actual arrival times at destination stop during the time window specified. |
| travel_time_sec | Actual travel times for all trips during time window specified, in seconds. |
| benchmark_travel_time_sec | Benchmark travel times for each trip, in seconds. Benchmarks are based on median travel times during the last 30 days. |

Tvehiclesbyroute *Query vehicles by a route*

Description

Returns vehicle positions for upcoming trips (including trips already underway) in a direction for a particular route.

Usage

```
Tvehiclesbyroute(route_id, api_key)
```

Arguments

| | |
|----------|--|
| route_id | GTFS-compatible route_id value for which vehicle positions should be returned. Example: "Red" |
| api_key | API key for MBTA API. To obtain one, visit the MBTA Developer Portal (http://realtime.mbtta.com/Portal/) |

Value

| | |
|-----------------|--|
| route_id | The unique GTFS-compatible identifier for the route for which vehicle positions are returned. Example: "CR-Franklin" |
| route_name | The human-readable name for the route for which vehicle positions are returned. Example: "Franklin Line" |
| route_type | The GTFS-compatible identifier for the type of service (mode). Example: "2" |
| mode_name | The human-readable name for the type of service (mode). Example: "Commuter Rail" |
| direction_id | The GTFS-compatible identifier for the direction. Example: "0" |
| direction_name | The human-readable name for the direction. Example: "Outbound" |
| trip_id | The unique GTFS-compatible identifier for the trip. Example: "CR-Providence-CR-Weekday-815" |
| trip_name | The human-readable name for the trip. Example: "815 (4:35 pm from South Station)" |
| trip_headsign | The text that identifies the trip's destination to passengers. Example: "North Station" |
| vehicle_id | The GTFS-compatible unique identifier for the vehicle. Example: "1531" |
| vehicle_lat | The GTFS-compatible latitude of the vehicle. Example: "42.08997" |
| vehicle_lon | The GTFS-compatible longitude of the vehicle. Example: "-71.4388" |
| vehicle_bearing | GTFS-compatable bearing of the vehicle. This can be the compass bearing, or the direction towards the next stop or intermediate location. May be empty. Example: "259" |

| | |
|-------------------|--|
| vehicle_speed | Identifies the vehicle's momentary speed, in meters per second. Example: "21" |
| vehicle_timestamp | Identifies the moment when the content of this feed has been created, in epoch time. Example: "1400855704" |

See Also

[Tvehiclesbytrip](#)

| | |
|-----------------|---|
| Tvehiclesbytrip | <i>Query vehicles by a particular trip.</i> |
|-----------------|---|

Description

Returns the predicted vehicle positions for a given trip.

Usage

```
Tvehiclesbytrip(trip_id, api_key)
```

Arguments

| | |
|---------|--|
| trip_id | GTFS-compatible trip_id value for which vehicle positions should be returned. Data type: String. Example: "CR-Providence-CR-Weekday-807" |
| api_key | API key for MBTA API. To obtain one, visit the MBTA Developer Portal (http://realtime.mbtta.com/Portal/) |

Value

| | |
|----------------|--|
| route_id | The unique GTFS-compatible identifier for the route for which vehicle positions are returned. Example: "CR-Providence" |
| route_name | The human-readable name for the route for which vehicle positions are returned. Example: "Providence/Stoughton Line" |
| route_type | The GTFS-compatible identifier for the type of service (mode). Example: "2" |
| mode_name | The human-readable name for the type of service (mode). Example: "Commuter Rail" |
| trip_id | The unique GTFS-compatible identifier for the trip for which vehicle positions are returned. Example: "CR-Providence-CR-Weekday-815" |
| trip_name | The human-readable for the trip for which schedule is returned. Example: "815 (4:35 pm from South Station)" |
| trip_headsign | The text that identifies the trip's destination to passengers. Example: "North Station" |
| direction_id | The GTFS-compatible identifier for the direction. Example: "0" |
| direction_name | The human-readable name for the direction. Example: "Outbound" |

| | |
|--------------------------------|--|
| <code>vehicle_id</code> | The GTFS-compatible unique identifier for the vehicle. Example: "1531" |
| <code>vehicle_lat</code> | The GTFS-compatible latitude of the vehicle. Example: "42.08997" |
| <code>vehicle_lon</code> | The GTFS-compatible longitude of the vehicle. Example: "-71.4388" |
| <code>vehicle_bearing</code> | GTFS-compatible bearing of the vehicle. This can be the compass bearing, or the direction towards the next stop or intermediate location. May be empty. Example: "259" |
| <code>vehicle_speed</code> | Identifies the vehicle's momentary speed, in meters per second. Example: "21" |
| <code>vehicle_timestamp</code> | Identifies the moment when the content of this feed has been created, in epoch time. Example: "1400855704" |

See Also

[Tvehiclesbyroute](#)

Index

*Topic **datasets**

routes, [2](#)

routes, [2](#)

Talertbyid, [3](#), [8](#)

Talerts, [5](#), [5](#)

Tdailymetrics, [8](#), [13](#)

Theadways, [10](#)

Theadwaysperformance, [11](#)

Tmetrics, [8](#), [12](#)

Tpredictionsbyroute, [14](#), [17](#), [18](#), [22](#)

Tpredictionsbystop, [15](#), [18](#), [23](#)

Tpredictionsbytrip, [17](#), [24](#)

Troutes, [19](#), [20](#)

Troutesbystop, [19](#), [20](#)

Tschedulebyroute, [21](#), [23](#), [24](#)

Tschedulebystop, [22](#), [22](#), [24](#)

Tschedulebytrip, [22](#), [23](#), [23](#)

Tservertime, [24](#)

Tstopsbylocation, [25](#)

Tstopsbyroute, [26](#)

Tstopslookup, [27](#)

Ttravelperformance, [28](#)

Ttraveltimes, [29](#), [29](#)

Tvehiclesbyroute, [30](#), [32](#)

Tvehiclesbytrip, [31](#), [31](#)