

Package ‘metro’

October 13, 2022

Title Washington Metropolitan Area Transit Authority API

Version 0.9.1

Description The Washington Metropolitan Area Transit Authority is a government agency operating light rail and passenger buses in the Washington D.C. area. With a free developer account, access their 'Metro Transparent Data Sets API' <<https://developer.wmata.com/>> to return data frames of transit data for easy analysis.

License MIT + file LICENSE

URL <https://github.com/kiernann/metro>, <https://kiernann.com/metro/>

BugReports <https://github.com/kiernann/metro/issues>

Depends R (>= 2.10)

Imports geodist (>= 0.0.6), hms (>= 1.0.0), httr (>= 1.4.2), jsonlite (>= 1.7.1), tibble (>= 3.0.4)

Suggests covr (>= 3.5.1), spelling (>= 2.2), testthat (>= 3.0.0)

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.1.1

NeedsCompilation no

Author Kiernan Nicholls [aut, cre]

Maintainer Kiernan Nicholls <kiernann@protonmail.com>

Repository CRAN

Date/Publication 2021-03-01 10:10:02 UTC

R topics documented:

bus_departs	2
bus_incidents	3
bus_path	5

bus_position	6
bus_routes	7
bus_schedule	8
bus_stops	10
elevator_incidents	11
metro	12
metro_lines	13
metro_routes	14
metro_stations	15
metro_stops	16
next_bus	16
next_train	17
rail_destination	19
rail_entrance	20
rail_incidents	21
rail_lines	22
rail_path	23
rail_positions	25
rail_stations	26
rail_times	27
standard_routes	29
station_info	30
track_circuits	31
wmata_key	32
Index	34

bus_departs	<i>Bus Schedule at Stop</i>
-------------	-----------------------------

Description

Returns a set of buses scheduled at a stop for a given date.

Usage

```
bus_departs(StopID, Date = NULL, api_key = wmata_key())
```

Arguments

StopID	7-digit regional stop ID.
Date	(Optional) Date for which to retrieve route and stop information.
api_key	Subscription key which provides access to this API. Defaults <code>Sys.getenv("WMATA_KEY")</code> via <code>wmata_key()</code> .

Format

A tibble with 1 row per bus departure and 8 variables:

ScheduleTime Date and time (UTC) when the bus is scheduled to stop at this location.

DirectionNum Denotes a binary direction (0 or 1) of the bus. There is no specific mapping to direction, but a different value for the same route signifies that the buses are traveling in opposite directions. Use the `TripDirectionText` column to show the actual destination of the bus.

StartTime Scheduled start date and time (UTC) for this trip.

EndTime Scheduled end date and time (UTC) for this trip.

RouteID Bus route variant identifier (pattern). This variant can be used in several other bus methods which accept variants. Note that customers will never see anything other than the base route name, so variants 10A, 10Av1, 10Av2, etc. will be displayed as 10A on the bus.

TripDirectionText General direction of the trip (e.g.: NORTH, SOUTH, EAST, WEST).

TripHeadsign Destination of the bus.

TripID Trip identifier. This can be correlated with the data in our bus schedule information as well as bus positions.

Value

Data frame containing scheduled arrival information.

See Also

<https://developer.wmata.com/docs/services/54763629281d83086473f231/operations/5476362a281d830c946a3console>

Other Bus Route and Stop Methods: [bus_path\(\)](#), [bus_position\(\)](#), [bus_routes\(\)](#), [bus_schedule\(\)](#), [bus_stops\(\)](#)

Examples

```
## Not run:
bus_departs(1001195, "2021-01-01")
```

```
## End(Not run)
```

bus_incidents

Bus Incidents

Description

Returns a set of reported bus incidents/delays for a given Route. Omit the Route to return all reported items.

Usage

```
bus_incidents(Route = NULL, api_key = wmata_key())
```

Arguments

Route	Base bus route; variations are not recognized (i.e.: C2 instead of C2v1, C2v2, etc.).
api_key	Subscription key which provides access to this API. Defaults <code>Sys.getenv("WMATA_KEY")</code> via <code>wmata_key()</code> .

Format

A data frame with 1 row per incident and 5 variables:

IncidentID Unique identifier for an incident.

IncidentType Free-text description of the incident type. Usually Delay or Alert but is subject to change at any time.

RoutesAffected Character string of routes affected. Routes listed are usually identical to base route names (i.e.: not 10Av1 or 10Av2, but 10A), but may differ from what our bus methods return.

Description Free-text description of the delay or incident.

DateUpdated Date and time (UTC) of last update.

Details

Note that the Route parameter accepts only base route names and no variations, i.e.: use 10A instead of 10Av1 and 10Av2.

Value

Data frame of bus incidents and delays.

See Also

<https://developer.wmata.com/docs/services/54763641281d83086473f232/operations/54763641281d830c946a3>

Other Incident APIs: `elevator_incidents()`, `rail_incidents()`

Examples

```
## Not run:  
bus_incidents()  
  
## End(Not run)
```

bus_path	<i>Bus Path Details</i>
----------	-------------------------

Description

For a given date, returns the set of ordered latitude/longitude points along a route variant along with the list of stops served.

Usage

```
bus_path(RouteID, Date = NULL, api_key = wmata_key())
```

Arguments

RouteID	Bus route variant, e.g.: 70, 10A, 10Av1.
Date	Date for which to retrieve route and stop information. Uses today's date if NULL (default).
api_key	Subscription key which provides access to this API. Defaults <code>Sys.getenv("WMATA_KEY")</code> via <code>wmata_key()</code> .

Format

A list with 4 elements:

RouteID Bus route variant.

Name Descriptive name for the route.

ShapePoint The coordinate path of the line in both directions.

Stops The coordinates of stops on a given line.

Value

A list with (1) set of ordered latitude/longitude points along a route variant along with (2) the list of stops served.

See Also

<https://developer.wmata.com/docs/services/54763629281d83086473f231/operations/5476362a281d830c946a3>

Other Bus Route and Stop Methods: `bus_departs()`, `bus_position()`, `bus_routes()`, `bus_schedule()`, `bus_stops()`

Examples

```
## Not run:
bus_path("70")

## End(Not run)
```

bus_position	<i>Bus Position</i>
--------------	---------------------

Description

Returns bus positions for the given route, with an optional search radius. If no parameters are specified, all bus positions are returned.

Usage

```
bus_position(
  RouteId = NULL,
  Lat = NULL,
  Lon = NULL,
  Radius = 1000,
  api_key = wmata_key()
)
```

Arguments

RouteId	Base bus route, e.g.: 70, 10A.
Lat	Center point Latitude, required if Longitude and Radius are specified.
Lon	Center point Longitude, required if Latitude and Radius are specified.
Radius	Radius (meters) to include in the search area, required if Latitude and Longitude are specified.
api_key	Subscription key which provides access to this API. Defaults <code>Sys.getenv("WMATA_KEY")</code> via <code>wmata_key()</code> .

Format

A data frame with 1 row per bus and 13 variables:

VehicleID Unique identifier for the bus. This is usually visible on the bus itself.

Lat Last reported Latitude of the bus.

Lon Last reported Longitude of the bus.

Distance Distance (meters) of the bus from the provided search coordinates. Calculated using `geodist::geodist()` and the "cheap ruler" method.

Deviation Deviation, in minutes, from schedule. Positive values indicate that the bus is running late while negative ones are for buses running ahead of schedule.

DateTime Date and time (UTC) of last position update.

TripID Unique trip ID. This can be correlated with the data returned from the schedule-related methods.

RouteID Base route name as shown on the bus. Note that the base route name could also refer to any variant, so a RouteID of 10A could refer to 10A, 10Av1, 10Av2, etc.

DirectionText General direction of the trip, not the bus itself (e.g.: NORTH, SOUTH, EAST, WEST).

TripHeadsign Destination of the bus.

TripStartTime Scheduled start date and time (UTC) of the bus's current trip.

TripEndTime Scheduled end date and time (UTC) of the bus's current trip.

BlockNumber

Details

Note that the RouteID parameter accepts only base route names and no variations, i.e.: use 10A instead of 10Av1 or 10Av2.

Value

Data frame containing bus position information.

See Also

<https://developer.wmata.com/docs/services/54763629281d83086473f231/operations/5476362a281d830c946a3>

Other Bus Route and Stop Methods: [bus_departs\(\)](#), [bus_path\(\)](#), [bus_routes\(\)](#), [bus_schedule\(\)](#), [bus_stops\(\)](#)

Examples

```
## Not run:
bus_position("70", 38.8895, -77.0353, 2000)

## End(Not run)
```

bus_routes

Bus Routes

Description

Returns a list of all bus route variants (patterns). For example, the 10A and 10Av1 are the same route, but may stop at slightly different locations.

Usage

```
bus_routes(api_key = wmata_key())
```

Arguments

api_key Subscription key which provides access to this API. Defaults `Sys.getenv("WMATA_KEY")` via [wmata_key\(\)](#).

Format

A tibble with 3 variables and one row per route variant:

RouteID Unique identifier for a given route variant. Can be used in various other bus-related methods.

Name Descriptive name of the route variant.

LineDescription Denotes the route variant's grouping – lines are a combination of routes which lie in the same corridor and which have significant portions of their paths along the same roadways.

Value

Data frame containing route variant information

See Also

<https://developer.wmata.com/docs/services/54763629281d83086473f231/operations/5476362a281d830c946a3>

Other Bus Route and Stop Methods: [bus_departs\(\)](#), [bus_path\(\)](#), [bus_position\(\)](#), [bus_schedule\(\)](#), [bus_stops\(\)](#)

Examples

```
## Not run:
bus_routes()

## End(Not run)
```

bus_schedule

Bus Schedule

Description

Returns schedules for a given route variant for a given date.

Usage

```
bus_schedule(
  RouteID,
  IncludingVariations = TRUE,
  Date = NULL,
  api_key = wmata_key()
)
```


Arguments

RouteID	Bus route variant, e.g.: 70, 10A, 10Av1, etc.
IncludingVariations	Whether or not to include variations if a base route is specified in RouteID. For example, if B30 is specified and IncludingVariations is set to TRUE (default), data for all variations of B30 such as B30v1, B30v2, etc. will be returned.
Date	(Optional) Date for which to retrieve route and stop information.
api_key	Subscription key which provides access to this API. Defaults Sys.getenv("WMATA_KEY") via wmata_key() .

Format

A data frame with 1 row per trip and 10 variables:

RouteID Bus route variant. This can be used in several other bus methods which accept variants.

TripDirectionText General direction of the trip (NORTH, SOUTH, EAST, WEST, LOOP, etc.).

TripHeadsign Descriptive text of where the bus is headed. This is similar, but not necessarily identical, to what is displayed on the bus.

StartTime Scheduled start date and time (UTC) for this trip.

EndTime Scheduled end date and time (UTC) for this trip.

TripID Unique trip ID. This can be correlated with the data returned from the schedule-related methods.

StopID 7-digit regional ID which can be used in various bus-related methods. If unavailable, the StopID will be 0 or NA

StopName Stop name. May be slightly different from what is spoken or displayed in the bus.

StopSeq Order of the stop in the sequence of StopTimes.

Time Scheduled departure date and time (UTC) from this stop.

Value

Data frame containing trip information

See Also

<https://developer.wmata.com/docs/services/54763629281d83086473f231/operations/5476362a281d830c946a3>

Other Bus Route and Stop Methods: [bus_departs\(\)](#), [bus_path\(\)](#), [bus_position\(\)](#), [bus_routes\(\)](#), [bus_stops\(\)](#)

Examples

```
## Not run:
bus_schedule("70")

## End(Not run)
```

bus_stops

*Bus Stop Search***Description**

Returns a list of nearby bus stops based on latitude, longitude, and radius. Omit all parameters to retrieve a list of all stops.

Usage

```
bus_stops(Lat = NULL, Lon = NULL, Radius = NULL, api_key = wmata_key())
```

Arguments

Lat	Center point Latitude, required if Longitude and Radius are specified.
Lon	Center point Longitude, required if Latitude and Radius are specified.
Radius	Radius (meters) to include in the search area, required if Latitude and Longitude are specified.
api_key	Subscription key which provides access to this API. Defaults <code>Sys.getenv("WMATA_KEY")</code> via <code>wmata_key()</code> .

Format

A tibble with 1 row per stop and 6 variables:

StopID String array of route variants which provide service at this stop. Note that these are not date-specific; any route variant which stops at this stop on any day will be listed.

Name Stop name. May be slightly different from what is spoken or displayed in the bus.

Lat Latitude.

Lon Longitude.

Distance Distance (meters) of the stop from the provided search coordinates. Calculated using `geodist::geodist()` and the "cheap ruler" method.

Routes Character string of route variants which provide service at this stop. Note that these are not date-specific; any route variant which stops at this stop on any day will be listed.

Value

Data frame containing stop information

See Also

<https://developer.wmata.com/docs/services/54763629281d83086473f231/operations/5476362a281d830c946a3>

Other Bus Route and Stop Methods: `bus_departs()`, `bus_path()`, `bus_position()`, `bus_routes()`, `bus_schedule()`

Examples

```
## Not run:
bus_stops(38.8895, -77.0353, 500)

## End(Not run)
```

elevator_incidents *Elevator/Escalator Outages*

Description

Elevator/Escalator Outages

Usage

```
elevator_incidents(StationCode = NULL, api_key = wmata_key())
```

Arguments

StationCode Station code. Use [rail_stations\(\)](#) to return a list of all station codes. Use NULL (default) to list all incidents.

api_key Subscription key which provides access to this API. Defaults `Sys.getenv("WMATA_KEY")` via [wmata_key\(\)](#).

Format

A data frame with 1 row per incident and 9 variables:

UnitName Unique identifier for unit, by type (a single elevator and escalator may have the same UnitName, but no two elevators or two escalators will have the same UnitName).

UnitType Type of unit. Will be ELEVATOR or ESCALATOR.

StationCode Unit's station code. Use this value in other rail-related APIs to retrieve data about a station.

StationName Full station name, may include entrance information (e.g.: Metro Center, G and 11th St Entrance).

LocationDescription Free-text description of the unit location within a station (e.g.: Escalator between mezzanine and platform).

SymptomDescription Description for why the unit is out of service or otherwise in reduced operation.

DateOutOfServ Date and time (UTC) unit was reported out of service.

DateUpdated Date and time (UTC) outage details was last updated.

EstimatedReturnToService Estimated date and time (UTC) by when unit is expected to return to normal service. May be NULL.

Details

Note that for stations with multiple platforms and therefore StationCodes (e.g.: Metro Center, L'Enfant Plaza, etc.), a distinct call is required for each StationCode.

Value

Data frame of *reported* elevator and escalator outages.

See Also

<https://developer.wmata.com/docs/services/54763641281d83086473f232/operations/54763641281d830c946a3>

Other Incident APIs: `bus_incidents()`, `rail_incidents()`

Examples

```
## Not run:
elevator_incidents()

## End(Not run)
```

metro

metro: A package for the Washington Metropolitan Area Transit Authority API

Description

The Metro Transparent Data Sets API provides developer access to JSON data for the order and location of rail stations by line, train arrival predictions for each station, service alerts, and elevator/escalator status. This package has been developed to make calls to the API using dedicated R functions with copied documentation and identical parameters/arguments. All functions return one or more data frames for easy analysis.

Endpoints

There are six endpoint categories returning JSON data.

1. Bus Route and Stop Methods, starting with `bus_*`:
 - Bus Position: `bus_position()`
 - Path details: `bus_path()`
 - Routes: `bus_routes()`
 - Schedule: `bus_schedule()`
 - Schedule at stop: `bus_departs()`
 - Stop search: `bus_stops()`
2. Incidents, ending with `*_incidents`:
 - Bus Incidents: `bus_incidents()`
 - Elevator/Escalator Outages: `elevator_incidents()`

- Rail Incidents: `rail_incidents()`
3. Misc Methods
 - Validate API Key: `wmata_validate()`
 4. Rail Station Information, starting with `rail_*`:
 - Lines: `rail_lines()`
 - Parking Information: Not yet covered.
 - Path Between Stations: `rail_path()`
 - Station Entrances: `rail_entrance()`
 - Station Information: `station_info()`
 - Station List: `rail_stations()`
 - Station Timings: `rail_times()`
 - Station to Station Information: `rail_destination()`
 5. Real-Time Predictions, starting with `next_*`:
 - Next Buses: `next_bus()`
 - Next Trains: `next_train()`
 6. Train Positions:
 - Live Train Positions: `rail_positions()`
 - Standard Routes: `standard_routes()`
 - Track Circuits: `track_circuits()`

 metro_lines

All WMATA Rail Lines

Description

All WMATA Rail Lines

Usage

```
metro_lines
```

Format

A tibble with 6 rows and 6 variables:

LineCode Two-letter abbreviation for the line (e.g.: RD, BL, YL, OR, GR, or SV).

DisplayName Full name of line color.

StartStationCode Start station code. For example, will be F11 (Branch Avenue) for the Green Line, A15 (Shady Grove) for the Red Line, etc. Use this value in other rail-related APIs to retrieve data about a station.

EndStationCode End station code. For example, will be E10 (Greenbelt) for the Green Line, B11 (Glenmont) for the Red Line, etc. Use this value in other rail-related APIs to retrieve data about a station.

InternalDestination Intermediate terminal station code(s). During normal service, some trains on some lines might end their trip prior to the StartStationCode or EndStationCode. A good example is on the Red Line where some trains stop at A11 (Grosvenor) or B08 (Silver Spring). NA if not defined.

Details

As of 2021-02-22 22:00:00 ET.

metro_routes	<i>All WMATA Bus Routes</i>
--------------	-----------------------------

Description

All WMATA Bus Routes

Usage

metro_routes

Format

A tibble 322 rows and 3 variables:

RouteID Unique identifier for a given route variant. Can be used in various other bus-related methods.

Name Descriptive name of the route variant.

LineDescription Denotes the route variant's grouping – lines are a combination of routes which lie in the same corridor and which have significant portions of their paths along the same roadways.

Details

As of 2021-02-22 22:00:00 ET.

metro_stations	<i>All WMATA Rail Stations</i>
----------------	--------------------------------

Description

All WMATA Rail Stations

Usage

metro_stations

Format

A tibble with 95 rows and 10 variables:

StationCode Station code for this station. Use this value in other rail-related APIs to retrieve data about a station.

StationName Full name for this station, as shown on the WMATA website.

StationTogether For stations with multiple platforms (e.g.: Gallery Place, Fort Totten, L'Enfant Plaza, and Metro Center), the additional StationCode will be listed here.

LineCodes Character vector of two-letter abbreviations (e.g.: RD, BL, YL, OR, GR, or SV) served by this station. If the station has an additional platform, the lines served by the other platform are listed in the LineCodes values for the record associated with the StationCode found in StationTogether.

Lat Latitude.

Lon Longitude.

Street Street address (for GPS use).

City City.

State State (abbreviated).

Zip Zip code.

Details

As of 2021-02-22 22:00:00 ET.

metro_stops	<i>All WMATA Bus Stops</i>
-------------	----------------------------

Description

All WMATA Bus Stops

Usage

```
metro_stops
```

Format

A tibble with 9,044 rows and 5 variables:

StopID String array of route variants which provide service at this stop. Note that these are not date-specific; any route variant which stops at this stop on any day will be listed.

Name Stop name. May be slightly different from what is spoken or displayed in the bus.

Lat Latitude.

Lon Longitude.

Routes Character string of route variants which provide service at this stop. Note that these are not date-specific; any route variant which stops at this stop on any day will be listed.

Details

As of 2021-02-22 22:00:00 ET.

next_bus	<i>Next Buses</i>
----------	-------------------

Description

Returns next bus arrival times at a stop.

Usage

```
next_bus(StopID, api_key = wmata_key())
```

Arguments

StopID 7-digit regional stop ID.

api_key Subscription key which provides access to this API. Defaults `Sys.getenv("WMATA_KEY")` via `wmata_key()`.

Format

A tibble 1 row per arrival with 8 variables:

StopID 7-digit regional ID which can be used in various bus-related methods. If unavailable, the StopID will be 0 or NA

StopName Stop name. May be slightly different from what is spoken or displayed in the bus.

RouteID Base route name as shown on the bus. This can be used in other bus-related methods. Note that all variants will be shown as their base route names (i.e.: 10Av1 and 10Av2 will be shown as 10A).

DirectionText Customer-friendly description of direction and destination for a bus.

DirectionNum Denotes a binary direction (0 or 1) of the bus. There is no specific mapping to direction, but a different value for the same route signifies that the buses are traveling in opposite directions. Use the DirectionText element to show the actual destination of the bus.

Minutes Minutes until bus arrival at this stop. Numeric value.

VehicleID Bus identifier. This can be correlated with results returned from bus positions.

TripID Trip identifier. This can be correlated with the data in our bus schedule information as well as bus positions.

Value

Data frame of bus arrivals.

See Also

<https://developer.wmata.com/docs/services/5476365e031f590f38092508/operations/5476365e031f5909e4fe3>

Other Real-Time Predictions: [next_train\(\)](#)

Examples

```
## Not run:
next_bus(StopID = "1001195")

## End(Not run)
```

next_train

Next Trains

Description

Returns next train arrival information for one or more stations. Will return an empty set of results when no predictions are available. Use "All" for the StationCodes parameter to return predictions for all stations.

Usage

```
next_train(StationCodes = NULL, api_key = wmata_key())
```

Arguments

StationCodes	Character vector of station codes. For all predictions, use NULL (default) or "All".
api_key	Subscription key which provides access to this API. Defaults <code>Sys.getenv("WMATA_KEY")</code> via <code>wmata_key()</code> .

Format

A tibble 1 row per arrival with 9 variables:

Car Number of cars on a train, usually 6 or 8, but might also NA.

Destination Abbreviated version of the final destination for a train. This is similar to what is displayed on the signs at stations.

DestinationCode Destination station code. Can be NA. Use this value in other rail-related APIs to retrieve data about a station.

DestinationName When DestinationCode is populated, this is the full name of the destination station, as shown on the WMATA website.

Group Denotes the track this train is on, but does not necessarily equate to Track 1 or Track 2. With the exception of terminal stations, predictions at the same station with different Group values refer to trains on different tracks.

Line Two-letter abbreviation for the line (e.g.: RD, BL, YL, OR, GR, or SV). May also be NA for trains with no passengers.

LocationCode Station code for where the train is arriving. Useful when passing in All as the StationCodes parameter. Use this value in other rail-related APIs to retrieve data about a station.

LocationName Full name of the station where the train is arriving. Useful when passing in "All" as the StationCodes parameter.

Min Minutes until arrival. Can be a numeric value, 0 (arriving), -1 (boarding), or NA.

Details

For terminal stations (e.g.: Greenbelt, Shady Grove, etc.), predictions may be displayed twice.

Some stations have two platforms (e.g.: Gallery Place, Fort Totten, L'Enfant Plaza, and Metro Center). To retrieve complete predictions for these stations, be sure to pass in both StationCodes.

For trains with no passengers, the DestinationName will be "No Passenger".

Value

Data frame of train arrivals.

See Also

<https://developer.wmata.com/docs/services/547636a6f9182302184cda78/operations/547636a6f918230da8553>

Other Real-Time Predictions: [next_bus\(\)](#)

Examples

```
## Not run:
next_train(StationCodes = c("A02", "B02"))

## End(Not run)
```

rail_destination	<i>Rail Station to Station Information</i>
------------------	--

Description

Returns a distance, fare information, and estimated travel time between any two stations, including those on different lines. Omit both parameters to retrieve data for all stations.

Usage

```
rail_destination(
  FromStationCode = NULL,
  ToStationCode = NULL,
  api_key = wmata_key()
)
```

Arguments

FromStationCode	Station code for the origin station. Use the rail_stations() function to return a list of all station codes.
ToStationCode	Station code for the destination station. Use the rail_stations() function to return a list of all station codes.
api_key	Subscription key which provides access to this API. Defaults <code>Sys.getenv("WMATA_KEY")</code> via wmata_key() .

Format

A tibble 1 row per with variables:

SourceStation Origin station code. Use this value in other rail-related APIs to retrieve data about a station.

DestinationStation Destination station code. Use this value in other rail-related APIs to retrieve data about a station.

CompositeMiles Average of distance traveled between two stations and straight-line distance (as used for WMATA fare calculations). For more details, please refer to WMATA's [Tariff on Fares](#).

RailTime Estimated travel time (schedule time) in minutes between the source and destination station. This is not correlated to minutes (Min) in Real-Time Rail Predictions.

PeakTime Fare during peak times (weekdays from opening to 9:30 AM and 3-7 PM (EST), and weekends from midnight to closing).

OffPeakTime Fare during off-peak times (times other than the ones described below).

SeniorDisabled Reduced fare for **senior citizens or people with disabilities**.

Value

A data frame containing station to station information

See Also

<https://developer.wmata.com/docs/services/5476364f031f590f38092507/operations/5476364f031f5909e4fe3>

Other Rail Station Information: [rail_entrance\(\)](#), [rail_lines\(\)](#), [rail_path\(\)](#), [rail_stations\(\)](#), [rail_times\(\)](#), [station_info\(\)](#)

Examples

```
## Not run:
rail_destination("A01", "A08")

## End(Not run)
```

rail_entrance	<i>Rail Station Entrances</i>
---------------	-------------------------------

Description

Returns a list of nearby station entrances based on latitude, longitude, and radius (meters). Omit search parameters to return all station entrances.

Usage

```
rail_entrance(Lat = NULL, Lon = NULL, Radius = NULL, api_key = wmata_key())
```

Arguments

Lat	(Optional) Center point Latitude, required if Longitude and Radius are specified.
Lon	(Optional) Center point Longitude, required if Latitude and Radius are specified.
Radius	(Optional) Radius (meters) to include in the search area, required if Latitude and Longitude are specified.
api_key	Subscription key which provides access to this API. Defaults <code>Sys.getenv("WMATA_KEY")</code> via wmata_key() .

Format

A tibble 1 row per entrance with 7 variables:

Name Name of the entrance (usually the station name and nearest intersection).

StationCode The station code associated with this entrance. Use this value in other rail-related APIs to retrieve data about a station.

StationTogether For stations containing multiple platforms (e.g.: Gallery Place, Fort Totten, L'Enfant Plaza, and Metro Center), the other station code (previously `StationCode2`).

Description Additional information for the entrance, if available. Currently available data usually shows the same value as the Name element.

Lat Latitude.

Lon Longitude.

Distance Distance (meters) of the entrance from the provided search coordinates. Calculated using `geodist::geodist()` and the "cheap ruler" method.

Value

A data frame of station entrances.

See Also

<https://developer.wmata.com/docs/services/5476364f031f590f38092507/operations/5476364f031f5909e4fe3>

Other Rail Station Information: [rail_destination\(\)](#), [rail_lines\(\)](#), [rail_path\(\)](#), [rail_stations\(\)](#), [rail_times\(\)](#), [station_info\(\)](#)

Examples

```
## Not run:
rail_entrance(38.8895, -77.0353)

## End(Not run)
```

rail_incidents

Rail Incidents

Description

Reported rail incidents (significant disruptions and delays to normal service). The data is identical to WMATA's Metrorail Service Status feed.

Usage

```
rail_incidents(api_key = wmata_key())
```

Arguments

`api_key` Subscription key which provides access to this API. Defaults `Sys.getenv("WMATA_KEY")` via `wmata_key()`.

Format

A data frame with 1 row incident and 5 variables:

IncidentID Unique identifier for an incident.

Description Free-text description of the incident.

IncidentType Free-text description of the incident type. Usually Delay or Alert but is subject to change at any time.

LinesAffected Character vector of line codes (e.g.: RD; or BL; OR; or BL; OR; RD;).

DateUpdated Date and time (UTC) of last update.

Value

Data frame of all rail incidents.

See Also

<https://developer.wmata.com/docs/services/54763641281d83086473f232/operations/54763641281d830c946a3>

Other Incident APIs: `bus_incidents()`, `elevator_incidents()`

Examples

```
## Not run:
rail_incidents()

## End(Not run)
```

`rail_lines`

Rail Lines

Description

Returns information about all rail lines.

Usage

```
rail_lines(api_key = wmata_key())
```

Arguments

`api_key` Subscription key which provides access to this API. Defaults `Sys.getenv("WMATA_KEY")` via `wmata_key()`.

Format

A tibble 1 row per line with 6 variables:

LineCode Two-letter abbreviation for the line (e.g.: RD, BL, YL, OR, GR, or SV).

DisplayName Full name of line color.

StartStationCode Start station code. For example, will be F11 (Branch Avenue) for the Green Line, A15 (Shady Grove) for the Red Line, etc. Use this value in other rail-related APIs to retrieve data about a station.

EndStationCode End station code. For example, will be E10 (Greenbelt) for the Green Line, B11 (Glenmont) for the Red Line, etc. Use this value in other rail-related APIs to retrieve data about a station.

InternalDestination Intermediate terminal station code(s). During normal service, some trains on some lines might end their trip prior to the StartStationCode or EndStationCode. A good example is on the Red Line where some trains stop at A11 (Grosvenor) or B08 (Silver Spring). NA if not defined.

Value

A data frame of rail lines.

See Also

<https://developer.wmata.com/docs/services/5476364f031f590f38092507/operations/5476364f031f5909e4fe3>

Other Rail Station Information: [rail_destination\(\)](#), [rail_entrance\(\)](#), [rail_path\(\)](#), [rail_stations\(\)](#), [rail_times\(\)](#), [station_info\(\)](#)

Examples

```
## Not run:
rail_lines()

## End(Not run)
```

rail_path	<i>Path Between Stations</i>
-----------	------------------------------

Description

Returns a set of ordered stations and distances between two stations on the *same line*.

Usage

```
rail_path(FromStationCode, ToStationCode, api_key = wmata_key())
```

Arguments

FromStationCode	Station code for the origin station. Use the Station List method to return a list of all station codes.
ToStationCode	Station code for the destination station. Use the Station List method to return a list of all station codes.
api_key	Subscription key which provides access to this API. Defaults Sys.getenv("WMATA_KEY") via <code>wmata_key()</code> .

Format

A tibble 1 row per arrival with 6 variables:

LineCode Two-letter abbreviation for the line (e.g.: RD, BL, YL, OR, GR, or SV) this station's platform is on.

StationCode Station code for this station. Use this value in other rail-related APIs to retrieve data about a station.

StationName Full name for this station, as shown on the WMATA website.

SeqNum Ordered sequence number.

DistanceToPrev Distance in meters to the previous station in the list, ordered by SeqNum. Converted from feet, rounded to the nearest meter.

Details

Note that this method is not suitable on its own as a pathfinding solution between stations.

Value

A data frame of stations on rail path.

See Also

<https://developer.wmata.com/docs/services/5476364f031f590f38092507/operations/5476364f031f5909e4fe3>

Other Rail Station Information: `rail_destination()`, `rail_entrance()`, `rail_lines()`, `rail_stations()`, `rail_times()`, `station_info()`

Examples

```
## Not run:
rail_path("A01", "A08")

## End(Not run)
```

rail_positions	<i>Live Train Positions</i>
----------------	-----------------------------

Description

Returns uniquely identifiable trains in service and what track circuits they currently occupy. Will return an empty set of results when no positions are available.

Usage

```
rail_positions(api_key = api_key)
```

Arguments

`api_key` Subscription key which provides access to this API. Defaults `Sys.getenv("WMATA_KEY")` via `wmata_key()`.

Format

A tibble with 1 row per train and 9 variables:

TrainId Uniquely identifiable internal train identifier.

TrainNumber *Non-unique* train identifier, often used by WMATA's Rail Scheduling and Operations Teams, as well as over open radio communication.

CarCount Number of cars. Can sometimes be 0 when there is no data available.

DirectionNum The direction of movement regardless of which track the train is on. Valid values are 1 or 2. Generally speaking, trains with direction 1 are northbound/eastbound, while trains with direction 2 are southbound/westbound.

CircuitId The circuit identifier the train is currently on. This identifier can be referenced from the Standard Routes method.

DestinationStationCode Destination station code. Can be NA. Use this value in other rail-related APIs to retrieve data about a station. Note that this value may *sometimes* differ from the destination station code returned by our Next Trains methods.

LineCode Two-letter abbreviation for the line (e.g.: RD, BL, YL, OR, GR, or SV). May also be NA in certain cases.

SecondsAtLocation Approximate "dwell time". This is not an exact value, but can be used to determine how long a train has been reported at the same track circuit.

ServiceType Service Type of a train, can be any of the following Service Types:

- **NoPassengers**: This is a non-revenue train with no passengers on board. Note that this designation of `NoPassengers` does not necessarily correlate with PIDS "No Passengers". As of 08/22/2016, this functionality has been reinstated to include all non-revenue vehicles, with minor exceptions.
- **Normal**: This is a normal revenue service train.
- **Special**: This is a special revenue service train with an unspecified line and destination. This is more prevalent during scheduled track work.
- **Unknown**: This often denotes cases with unknown data or work vehicles.

Details

Please refer to [this page](#) for additional details.

Value

A data frame of train positions.

See Also

<https://developer.wmata.com/docs/services/5763fa6ff91823096cac1057/operations/5763fb35f91823096cac1>

Other Train Positions: [standard_routes\(\)](#), [track_circuits\(\)](#)

Examples

```
## Not run:
rail_positions()

## End(Not run)
```

rail_stations

Rail Station List

Description

Returns a list of station location and address information based on a given LineCode. Use NULL (default) to return all stations. The response is an data frame identical to that returned in the Station Information method.

Usage

```
rail_stations(LineCode = NULL, api_key = wmata_key())
```

Arguments

LineCode	Two-letter line code abbreviation, or NULL (default): <ul style="list-style-type: none"> • RD - Red • YL - Yellow • GR - Green • BL - Blue • OR - Orange • SV - Silver
api_key	Subscription key which provides access to this API. Defaults <code>Sys.getenv("WMATA_KEY")</code> via wmata_key() .

Format

A tibble 1 row per station with 10 variables:

StationCode Station code for this station. Use this value in other rail-related APIs to retrieve data about a station.

StationName Full name for this station, as shown on the WMATA website.

StationTogether For stations with multiple platforms (e.g.: Gallery Place, Fort Totten, L'Enfant Plaza, and Metro Center), the additional `StationCode` will be listed here.

LineCodes Character vector of two-letter abbreviations (e.g.: RD, BL, YL, OR, GR, or SV) served by this station. If the station has an additional platform, the lines served by the other platform are listed in the `LineCodes` values for the record associated with the `StationCode` found in `StationTogether`.

Lat Latitude.

Lon Longitude.

Street Street address (for GPS use).

City City.

State State (abbreviated).

Zip Zip code.

Value

A data frame of stations on a rail line.

See Also

<https://developer.wmata.com/docs/services/5476364f031f590f38092507/operations/5476364f031f5909e4fe3>

Other Rail Station Information: [rail_destination\(\)](#), [rail_entrance\(\)](#), [rail_lines\(\)](#), [rail_path\(\)](#), [rail_times\(\)](#), [station_info\(\)](#)

Examples

```
## Not run:
rail_stations("RD")

## End(Not run)
```

 rail_times

Rail Station Timings

Description

Returns opening and scheduled first/last train times based on a given `StationCode`. Omit the `StationCode` to return timing information for all stations.

Usage

```
rail_times(StationCode = NULL, api_key = wmata_key())
```

Arguments

StationCode Station code. Use the [rail_stations\(\)](#) function to return a list of all station codes. Use NULL (default) to return times for all stations.

api_key Subscription key which provides access to this API. Defaults `Sys.getenv("WMATA_KEY")` via [wmata_key\(\)](#).

Format

A tibble 1 row per train with 8 variables:

StationCode Station code for this station. Use this value in other rail-related APIs to retrieve data about a station.

StationName Full name of the station.

DestinationStation Station code for the train's destination. Use this value in other rail-related APIs to retrieve data about a station.

Weekday Day of the week abbreviation. From list element names.

OpeningTime Station opening time. Converted to hms class with [hms::parse_hm\(\)](#), representing seconds since midnight of that Weekday.

FirstTime First train leaves the station at this time (ET). Converted to hms class with [hms::parse_hm\(\)](#), representing seconds since midnight of that Weekday.

LastTime Last train leaves the station at this time (ET). Converted to hms class with [hms::parse_hm\(\)](#), representing seconds since midnight of that Weekday. For times that were in the AM of the *next* Weekday, time is greater than 24 hours.

Details

Note that for stations with multiple platforms (e.g.: Metro Center, L'Enfant Plaza, etc.), a distinct call is required for each `StationCode` to retrieve the full set of train times at such stations.

Value

A *tidy* data frame of station schedules. Combined from a nested list of weekday times.

See Also

<https://developer.wmata.com/docs/services/5476364f031f590f38092507/operations/5476364f031f5909e4fe3>

Other Rail Station Information: [rail_destination\(\)](#), [rail_entrance\(\)](#), [rail_lines\(\)](#), [rail_path\(\)](#), [rail_stations\(\)](#), [station_info\(\)](#)

Examples

```
## Not run:
rail_times("A01")

## End(Not run)
```

standard_routes	<i>Standard Routes</i>
-----------------	------------------------

Description

Returns an ordered list of mostly revenue (and some lead) track circuits, arranged by line and track number. This data does not change frequently and should be cached for a reasonable amount of time.

Usage

```
standard_routes()
```

Format

A nested tibble with 1 row per track and 3 variables:

LineCode Abbreviation for the revenue line. Note that this also includes **YLRP** (Yellow Line Rush Plus).

TrackNum Track number (1 or 2).

TrackCircuits Array containing ordered track circuit information:

- **CircuitId**: An internal system-wide uniquely identifiable circuit number.
- **SeqNum**: Order in which the circuit appears for the given line and track. Sequences go from West to East and South to North.
- **StationCode**: If the circuit is at a station, this value will represent the station code. Otherwise, it will be NA. Use this value in other rail-related APIs to retrieve data about a station.

Details

Please refer to [this page](#) for additional details.

Value

A nested data frame of track circuits.

See Also

<https://developer.wmata.com/docs/services/5763fa6ff91823096cac1057/operations/57641afc031f59363c586>

Other Train Positions: [rail_positions\(\)](#), [track_circuits\(\)](#)

Examples

```
## Not run:
standard_routes()

## End(Not run)
```

station_info	<i>Rail Station Information</i>
--------------	---------------------------------

Description

Returns station location and address information based on a given `StationCode`. Similar to `rail_stations()`, but returns data on a given station(s) rather than all stations on a line.

Usage

```
station_info(StationCode, api_key = wmata_key())
```

Arguments

`StationCode` A character vector of one or more station codes. A separate call is made to the API for each code, do not provide more than 10.

`api_key` Subscription key which provides access to this API. Defaults `Sys.getenv("WMATA_KEY")` via `wmata_key()`.

Format

A tibble 1 row per station with 10 variables:

StationCode Station code for this station. Use this value in other rail-related APIs to retrieve data about a station.

StationName Full name for this station, as shown on the WMATA website.

StationTogether For stations with multiple platforms (e.g.: Gallery Place, Fort Totten, L'Enfant Plaza, and Metro Center), the additional `StationCode` will be listed here.

LineCodes Character vector of two-letter abbreviations (e.g.: RD, BL, YL, OR, GR, or SV) served by this station. If the station has an additional platform, the lines served by the other platform are listed in the `LineCodes` values for the record associated with the `StationCode` found in `StationTogether`.

Lat Latitude.

Lon Longitude.

Street Street address (for GPS use).

City City.

State State (abbreviated).

Zip Zip code.

Value

A data frame of stations.

See Also

<https://developer.wmata.com/docs/services/5476364f031f590f38092507/operations/5476364f031f5909e4fe3>

Other Rail Station Information: [rail_destination\(\)](#), [rail_entrance\(\)](#), [rail_lines\(\)](#), [rail_path\(\)](#), [rail_stations\(\)](#), [rail_times\(\)](#)

Examples

```
## Not run:
station_info(StationCode = c("A07", "B07"))

## End(Not run)
```

track_circuits	<i>Track Circuits</i>
----------------	-----------------------

Description

Returns a list of all track circuits including those on pocket tracks and crossovers. Each track circuit may include references to its right and left neighbors.

Usage

```
track_circuits()
```

Format

A nested tibble with 1 row per train and 9 variables:

Track Track number. 1 and 2 denote "main" lines, while 0 and 3 are connectors (between different types of tracks) and pocket tracks, respectively.

CircuitId An internal system-wide uniquely identifiable circuit number.

Neighbors Data frame containing track circuit neighbor information. Note that some track circuits have no neighbors in one direction. All track circuits have at least one neighbor.

- **NeighborType**: Left or Right neighbor group. Generally speaking, left neighbors are to the west and south, while right neighbors are to the east/north.
- **CircuitIds**: Data frame containing neighboring circuit IDs as list column.

Details

Please refer to [this page](#) for additional details.

Value

A data frame of nested track circuit information.

See Also

<https://developer.wmata.com/docs/services/5763fa6ff91823096cac1057/operations/57644238031f59363c586>

Other Train Positions: [rail_positions\(\)](#), [standard_routes\(\)](#)

Examples

```
## Not run:
track_circuits()

## End(Not run)
```

wmata_key

Find a WMATA API key

Description

All calls to the WMATA API must be accompanied by a personal API key. A free key can be obtained by subscribing to the default tier:

1. Sign up for a free WMATA developer account: <https://developer.wmata.com/>
2. Log in and subscribe to the free default tier.
3. Copy the primary or secondary key from your profile.
4. Pass your API key to a function through one of three ways:
 - (a) Use [Sys.setenv\(\)](#) to temporarily set define WMATA_KEY.
 - (b) Pass your key as a string to the `api_key` argument of any function.
 - (c) Store the the WMATA key as a line like `WMATA_KEY=xxxxyyzzz` in your `.Renviron` file, to be called by [wmata_key\(\)](#) via [Sys.getenv\(\)](#).

The WMATA also provides a demonstration key. This key should **never** be used in production applications, it is rate limited and subject to change at *any* time. The key can be found on the WMATA products page when not signed into a developer account.

The [wmata_validate\(\)](#) function can be used to verify a key is valid.

Usage

```
wmata_key(api_key = Sys.getenv("WMATA_KEY"))
```

```
wmata_validate(api_key = wmata_key())
```

Arguments

`api_key` Subscription key which provides access to this API. Defaults `Sys.getenv("WMATA_KEY")` via [wmata_key\(\)](#).

Details

Default tier sufficient for most casual developers. Rate limited to 10 calls/second and 50,000 calls per day. This product contains 8 APIs and all APIs needed for the functions in this package:

- Bus Route and Stop Methods
- GTFS (unused)
- Incidents
- Misc Methods
- Rail Station Information
- Real-Time Bus Predictions
- Real-Time Rail Predictions
- Train Positions

Value

For `wmata_key()`, a 32 character alphanumeric API key. For `wmata_validate()`, either TRUE for a valid key or an error if invalid.

See Also

<https://developer.wmata.com/docs/services/5923434c08d33c0f201a600a/operations/5923437c031f5914d0204>

Index

* **Bus Route and Stop Methods**

- bus_departs, 2
- bus_path, 5
- bus_position, 6
- bus_routes, 7
- bus_schedule, 8
- bus_stops, 10

* **Incident APIs**

- bus_incidents, 3
- elevator_incidents, 11
- rail_incidents, 21

* **Misc Methods**

- wmata_key, 32

* **Rail Station Information**

- rail_destination, 19
- rail_entrance, 20
- rail_lines, 22
- rail_path, 23
- rail_stations, 26
- rail_times, 27
- station_info, 30

* **Real-Time Predictions**

- next_bus, 16
- next_train, 17

* **Train Positions**

- rail_positions, 25
- standard_routes, 29
- track_circuits, 31

* **datasets**

- metro_lines, 13
- metro_routes, 14
- metro_stations, 15
- metro_stops, 16

- bus_departs, 2, 5, 7–10
- bus_departs(), 12
- bus_incidents, 3, 12, 22
- bus_incidents(), 12
- bus_path, 3, 5, 7–10
- bus_path(), 12

- bus_position, 3, 5, 6, 8–10
- bus_position(), 12
- bus_routes, 3, 5, 7, 7, 9, 10
- bus_routes(), 12
- bus_schedule, 3, 5, 7, 8, 8, 10
- bus_schedule(), 12
- bus_stops, 3, 5, 7–9, 10
- bus_stops(), 12

- elevator_incidents, 4, 11, 22
- elevator_incidents(), 12

- geodist::geodist(), 6, 10, 21

- hms::parse_hm(), 28

- metro, 12
- metro_lines, 13
- metro_routes, 14
- metro_stations, 15
- metro_stops, 16

- next_bus, 16, 18
- next_bus(), 13
- next_train, 17, 17
- next_train(), 13

- rail_destination, 19, 21, 23, 24, 27, 28, 31
- rail_destination(), 13
- rail_entrance, 20, 20, 23, 24, 27, 28, 31
- rail_entrance(), 13
- rail_incidents, 4, 12, 21
- rail_incidents(), 13
- rail_lines, 20, 21, 22, 24, 27, 28, 31
- rail_lines(), 13
- rail_path, 20, 21, 23, 23, 27, 28, 31
- rail_path(), 13
- rail_positions, 25, 29, 32
- rail_positions(), 13
- rail_stations, 20, 21, 23, 24, 26, 28, 31
- rail_stations(), 11, 13, 19, 28, 30

rail_times, [20](#), [21](#), [23](#), [24](#), [27](#), [27](#), [31](#)
rail_times(), [13](#)

standard_routes, [26](#), [29](#), [32](#)
standard_routes(), [13](#)
station_info, [20](#), [21](#), [23](#), [24](#), [27](#), [28](#), [30](#)
station_info(), [13](#)
Sys.getenv(), [32](#)
Sys.setenv(), [32](#)

track_circuits, [26](#), [29](#), [31](#)
track_circuits(), [13](#)

wmata_key, [32](#)
wmata_key(), [2](#), [4–7](#), [9–11](#), [16](#), [18–20](#), [22](#),
[24–26](#), [28](#), [30](#), [32](#), [33](#)
wmata_validate(wmata_key), [32](#)
wmata_validate(), [13](#), [32](#), [33](#)