MBTA

MBTA-REALTIME API DOCUMENTATION (V 2.0.1)

SEPTEMBER 8, 2014

Table of Contents

1. MBTA OPEN DATA OVERVIEW

The MBTA publishes the following data feeds about its service:

- **GTFS Schedule.** The full schedule of all MBTA service in the industry's leading format.
- **MBTA-realtime API.** Full-featured easy-to-use RESTful API with alert, vehicle location, and arrival prediction data (as well as access to schedule data).
- **GTFS-realtime.** Alert, vehicle location, and arrival-prediction data in a new standard format. Best for retrieving data for the whole system at once in a relatively small package, but must be extrapolated using GTFS data to be meaningful.
- **NextBus API.** MBTA bus location and prediction data is available through NextBus's industry-leading API. (Bus only.)
- **RSS alerts.** An easy way to add alert information to anything with an RSS reader.

Concepts, keys and IDs are consistent across data feeds wherever possible. This list does not include several older standalone real-time data feeds, which are deprecated.

This document covers the MBTA-realtime API. GTFS, GTFS-realtime, and the RSS alert feed documentation are available at http://realtime.mbta.com . NextBus API documentation is available from NextBus.

1.1 Use of MBTA data

Access to the MBTA GTFS-realtime feeds is governed by the language in the MassDOT Developers License Agreement (http://www.eot.state.ma.us/developers/) in addition to the following conditions:

- The MBTA reserves the right to suspend the data feed, modify the feed, or modify elements of the feed at any time at the MBTA's sole and absolute discretion.
- The MBTA does not guarantee any technical support of any kind to users.
- No user may execute polling commands more often than every 10 seconds. A user that polls more often than that or otherwise overtaxes the MBTA's system may be suspended or terminated from the data feed.

1.2 Getting help

More documentation is available at http://realtime.mbta.com .

The MBTA is happy to answer developer questions at developer@mbta.com . Developers are also encouraged to join the MBTA Developers discussion forum at https://groups.google.com/forum/?fromgroups#!forum/massdotdevelopers .

2. MBTA-REALTIME API INTRODUCTION

MBTA-realtime provides RESTful web services to provide data about MBTA services. Data are provided in XML, JSON, and JSON-P formats. You will need to register for a free API key to use the API, but you can use the open development key (provided in the query examples throughout this document and at http://realtime.mbta.com/Portal/Home/Download) to start playing with the API right now.

2.1 Format Documentation

A basic introduction to RESTful web services can be found at these sites:

- http://www.infoq.com/articles/rest-introduction
- http://en.wikipedia.org/wiki/REST

Basic summaries of the XML and JSON formats can be found at these sites:

- http://www.w3schools.com/xml/xml_whatis.asp
- http://www.w3schools.com/json/default.asp

2.2 Relationship with Other MBTA Data Feeds

The MBTA-realtime API is the best way to get arrival predictions for a specific line, route, station or stop, or to get alerts in any context.

Although MBTA-realtime does provide schedule data we do not recommend using it to retrieve the entire service schedule. GTFS is the best format for that purpose.

Similarly, developers who wish to continually get all the latest vehicle locations systemwide or all the latest arrival predictions systemwide are advised to use GTFS-realtime. For developers wishing to continually get all the latest alert information, both MBTA-realtime and GTFS-realtime offer their own advantages – GTFS-realtime is an emerging industry standard, but MBTA-realtime is easier to use and provides richer data.

2.3 MBTA-realtime Versions

This document describes the MBTA-realtime API v2. MBTA-realtime API v2 includes more fields and more features than v1, and no functionality in v1 has been removed.

The biggest new feature of MBTA-realtime API v2 is the addition of vehicle position and arrival prediction data. Other changes include JSONP support and new alert call parameters, like include_access_alerts to let you determine whether elevator/escalator/lift outage data should be returned, and new alert fields, like service_effect_text that provides a short summary (“Orange Line shuttle”) and alert_lifecycle that lets you easily distinguish between alerts affecting service right now and alerts about upcoming planned service changes.

Developers using MBTA-realtime API v1 should transition to v2. MBTA-realtime API v1 will be discontinued at the end of 2014.

Since no calls or fields have been removed, for many developers replacing “v1” with “v2” in the URL used to call the API will be enough to maintain existing functionality.

3. ACCESSING THE API

3.1 Getting Started: Getting Your Own Account and API Key

To access the web services, you are required to register for accounts and API keys using the MBTA-realtime Developer Portal (http://realtime.mbta.com/Portal/). The use of the Developer Portal is described in the following subsections.

3.1.1 REGISTER FOR AN ACCOUNT

To register for an account, visit the Developer Portal (http://realtime.mbta.com/Portal/) and click the "Register" link on the upper right-hand corner. Enter a username, password, email address, and phone number, and then click the "Register" button.

The Developer Portal will send back an email acknowledging the request for registration, along with a confirmation token, and a confirmation URL. Click the URL or visit http://realtime.mbta.com/Portal/Account/Confirmation and enter the token to complete the registration process. The account will be confirmed in the system.

3.1.2 LOG IN

To login to a registered developer account, visit the Developer Portal (http://realtime.mbta.com/Portal/) and click the "Log in" link on the upper right-hand corner. Enter the username and password, and then click the "Log in" button. The "Manage API Keys" page will open.

3.1.3 REGISTER FOR AN API KEY

To register for an API key, visit the "Manage API Keys" page, enter the name and description of the application which will use the API key, and then click the "Register" button.

The Developer Portal will send an email once the API Key has been granted. Note: this may take up to a day.

Note: An open development API key has been provided to all developers for use in getting started. It can be found at http://realtime.mbta.com/Portal/Home/Download . It may change or be discontinued at any time. Do not go into production using the open development key!

3.1.4 MONITOR API KEYS

To monitor API keys, visit the "Manage API Keys" page. The page will show all API keys that have been granted or are pending for the developer account. To see more information for an API key, click the API key. The "API Key Details" page will open to show the account and application linked to the API key, as well as usage details.

3.2 Usage Limits

The MBTA makes this API available because we want people to use it, and we want you to be able to access it as much as you need to deliver the best benefit to end users. We also want to behave predictably in the event of a usage spike, however, and safeguard the system so that one misbehaving application does not jeopardize the data for all users. So MBTA-realtime does have usage limits, but it is easy for you to increase your limit if you need to.

Initial usage limits are set as up to 10,000 requests per day per API key. This limit can be increased upon request. To request an increase please email developer@mbta.com and include the account username, application name, and API Key.

You will receive emails when approaching usage limits (at 80%, 90%, and 100% of your limit.)

3.3 Formats and Common Parameters

3.3.1 QUERY FORMAT

To access data using the web services, a query in the form of a URL text string needs to be used, containing the API key, web service name, and any required and/or optional input parameters.

The overall query format is:

```
http://realtime.mbta.com/developer/api/v2/<query>?api_key=<your api key>
&<parameter>=<required/optional parameters>
```

3.3.2 REQUESTING JSON, JSONP, AND XML

The "format" parameter lets you specify data in JSON, XML, or JSONP. "Format" is an optional parameter. If no format is specified, data is returned based on the request header.

Get data in JSON format

To get data in JSON format, include the "format" parameter and set to "json", or leave it out.

```
http://realtime.mbta.com/developer/api/v2/<query>?api_key=<your api key>
&format=json&<parameter>=<required/optional parameters>
```

or

```
http://realtime.mbta.com/developer/api/v2/<query>?api_key=<your api key>
&<parameter>=<required/optional parameters>
```

Get data in XML format

To get data in XML include the format parameter and set to "xml".

```
http://realtime.mbta.com/developer/api/v2/<query>
?api_key=<developer's api key>&format=xml&<parameter>=<required/optional parameters>
```

Get data in JSONP format

To get data in JSONP format include the "format" parameter and set to "jsonp" AND include a "jsonpcallback" parameter and set to name of callback function. The format for the web services request queries to get data in JSONP format is:

```
http://realtime.mbta.com/developer/api/v2/<query>?api_key=<your api key>
&format=jsonp&jsonpcallback=<name of callback function>
&<parameter>=<required/optional parameters>
```

3.3.3 COMMON PARAMETERS

The following sections will identify special parameters for each query. The following table identifies the parameters common to all queries, which support the features above.

Name	Description
api_key	Unique API key assigned to each developer
format (optional)	Format for response. Possible values: "json","xml", and "jsonp"
jsonpcallback (optional)	Function call requested.

4. AVAILABLE QUERIES

The table below lists the 20 queries available through the MBTA-realtime API. Generally information can be requested about routes, stops, the schedule, predicted arrivals/departures, vehicle locations, and alerts; and this information is available for a specified route, stop, or trip.

Query	Returns
Routes	
routes	list of all routes for which data can be requested
routesbystop	a list of routes that serve a particular stop
Stops	
stopsbyroute	a list of stops for a particular route
stopsbylocation	a list of the stops nearest a particular location
Schedule	
schedulebystop	scheduled arrivals and departures at a particular stop
schedulebyroute	scheduled arrivals and departures for a particular route
schedulebytrip	scheduled arrivals and departures for a particular trip
Predictions and Vehicle Locations	
predictionsbystop	arrival/departure predictions, plus vehicle locations and alert headers, for a stop
predictionsbyroute	arrival/departure predictions, plus vehicle locations and alert headers, for a route
predictionsbytrip	arrival/departure predictions, plus vehicle location, for a trip
vehiclesbyroute	vehicle locations for a route
vehiclesbytrip	vehicle location for a trip
Alerts	
alerts	List of all alerts, with all details
alertsbyroute	List of all alerts applicable to a route, with all details
alertsbystop	List of all alerts applicable to a stop, with all details
alertbyid	One alert, with all details
alertheaders	List of all alerts, header information only
alertheadersbyroute	List of all alerts applicable to a route, header information only
alertheadersbystop	List of all alerts applicable to a stop, header information only
Other	
servertime	the current server time

These queries are documented in full over the following pages. Examples and terminology are in JSON; for XML assume “object” means “element” and “property” means “attribute” unless otherwise stated.

4.1 Route Queries: routes and routesbystop

These queries return a list of routes, either all routes or routes that serve a specified stop.

4.1.1 ROUTES

This query will return a complete list of routes for which data can be requested through the web services.

Special Parameters

None.

Response Fields

Name	Description
route_list	Root object of the response
Mode	Child object of “route_list” Contains information for a mode
route_type	Property of “mode.” String representation of an Integer (0-7) The GTFS-compatible identifier for the type of service (mode) Example: “1”
mode_name	Property of “mode.” String. The human-readable name for the type of service (mode) Example: “Subway”
Route	Child object of “mode.” Contains information for a route
route_id	Property of “route.” String. The unique GTFS-compatible identifier for the route Example: “931_”
route_name	Property of “route.” String. The human-readable name for the route Example: “Red Line”
route_hide (optional)	Property of “route.” String representation of a Boolean. Whether this route should be hidden from users in some contexts Possible values: “true”. Only included if route_hide is “true”

Notes

Routes are returned in MBTA’s recommended display order: alphabetical by route_id except for bus routes where lettered routes (Silver Line and CT routes) are displayed before numbered routes, and numbered routes are displayed in ascending numerical order (i.e. 1, 2, 3 instead of 1, 10, 100).

An example of a route for which route_hide=”true” is route 62/76 (route_id 627) which is for trips that operate on a hybrid of route 62 and route 76. Any alert that affects route 62/76 will also affect either route 62 or route 76 or both, so listing route 62/76 as one of the affected routes is redundant.

Example

http://realtime.mbta.com/developer/api/v2/routes?api_key=wX9NwuHnZU2ToO7GmGR9uw&format=json

```
{
	mode: [{
		route_type: "0",
		mode_name: "Subway",
		route: [{
			"route_id": "810_",
			"route_name": "Green Line"
		},
		{
			"route_id": "812_",
			"route_name": "Green Line"
		},
		.
		.
		.
		]
	},
	{
		route_type: "1",
		mode_name: "Subway",
		route: [{
			route_id: "903_",
			route_name: "Orange Line"
		},
		{
			route_id: "913_",
			route_name: "Orange Line"
		},
		.
		.
		.
		]
	},
	{
		route_type: "3",
		mode_name: "Bus",
		route: [{
			route_id: "701",
			route_name: "CT1"
		},
		{
			route_id: "747",
			route_name: "CT2"
		},
		{
			route_id: "708",
			route_name: "CT3"
		},
		{
			route_id: "746",
			route_name: "Silver Line Waterfront",
			route_hide: "true"
		},

	.
	.
	.
	]
}
```

4.1.2 ROUTESBYSTOP

This query will return a list of routes that serve a particular stop.

Special Parameters

Name	Description
stop	GTFS-compatible stop_id value for which routes should be returned. Data type: String Example: “70065”

Response Fields

The “stop_id” and “stop_name” properties of route_list are the only difference between the response fields returned by the “routes” and “routesbystop” query.

Name	Description
route_list	Root object of the response document
stop_id	Property of “route_list.” String. The GTFS-compatible unique identifier for the stop for which routes are returned Example: “70065”
stop_name	Property of “route_list.” String. The GTFS-compatible name for the stop for which routes are returned Example: “Porter Sq - Inbound”
mode	Child object of route_list. Contains information for a mode that serves this stop
route_type	Property of “mode.” String representation of an Integer (0-7) The GTFS-compatible identifier for the type of service (mode) Example: “1”
mode_name	Property of “mode.” String. The human-readable name for the type of service (mode) Example: “Subway”
Route	Child object of “mode.” Contains information for a route
route_id	Property of “route.” String. The unique GTFS-compatible identifier for the route Example: “931_”
route_name	Property of “route.” String. The human-readable name for the route Example: “Red Line”
route_hide (optional)	Property of “route.” String representation of a Boolean. Whether this route should be hidden from users Possible values: “true”. Only included if route_hide is “true”

Notes

- 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.

- Routes are returned in MBTA's preferred display order: alphabetical by route_id except for bus routes where lettered routes (Silver Line and CT routes) are displayed before numbered routes, and numbered routes are displayed in ascending numerical order (i.e. 1, 2, 3 instead of 1, 10, 100).

Example

http://realtime.mbta.com/developer/api/v2/routesbystop?api_key=wX9NwuHnZU2ToO7GmGR9uw&stop=70065&format=json


```
{
	stop_id: "70065",
	stop_name: "Porter Sq - Inbound",
	mode: [{
		route_type: "1",
		mode_name: "Subway",
		route: [{
			route_id: "931_",
			route_name: "Red Line"
		},
		{
			route_id: "933_",
			route_name: "Red Line"
		}]
	}]
}
```


4.2 Stop Queries: stopsbyroute and stopsbylocation

stopsbyroute returns a list of all stops served by a route; stopsbylocation returns a list of all stops near a specified latitude/longitude.

4.2.1 STOPSBYROUTE

This query will return a list of stops for a particular route.

Special Parameters

Name	Description
route	GTFS-compatible route_id value for which stops should be returned Data type: String Example: "931_"

Response Fields

The responses from stopsbyroute and stopsbylocation are organized a little differently. The stop objects themselves only have one difference – the stop_order property in stopsbyroute is replaced by the distance property in stopsbylocation.

Name	Description
stop_list	Root object of the response
route_id	Property of stop_list. String. The unique GTFS-compatible identifier for the route for which stops are returned Example: "931_"

route_name	Property of stop_list. String. The human-readable name for the route for which stops are returned Example: “Red Line”
direction	Child object of stop_list. Contains information for a direction of the route
direction_id	Property of “direction.” String representation of a Bit (0 or 1) The GTFS-compatible identifier for the direction Example: “0”
direction_name	Property of “direction.” String. The human-readable name for the direction Example: “Southbound”
stop	Child object of “direction.” Contains all information for a stop on the direction of the route
stop_order	Property of “stop.” object. String representation of an Integer (starting with 1) Identifies where the stop comes in the order of stops for this route and direction (note: not guaranteed to be unique) Example: “1”
stop_id	Property of “stop.” String. The GTFS-compatible unique identifier for the stop Example: “70063”
stop_name	Property of “stop.” String. The GTFS-compatible name for the stop (not unique) Example: “Davis Sq - Inbound”
parent_station	Property of “stop.” String. 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”
parent_station_name	Property of “stop.” String. The human-readable name for the larger station associated with the stop. (note: can be empty if stop does not have an associated station) Example: “Davis Station”
stop_lat	Property of “stop.” String representation of a Float. The GTFS-compatible latitude of the station. Example: “42.3967399597168”
stop_lon	Property of “stop.” String representation of a Float. The GTFS-compatible longitude of the station. Example: “-71.1218185424805”

Notes

- The “stop_order” property is not guaranteed to be unique for a route and direction.
- The “parent_station” and “parent_station_name” properties can be empty if stop does not have an associated parent station.
- The route_id and route_name properties of the stop_list are currently not displayed. They will be displayed as properties of the stop_list in a future update.

Example

http://realtime.mbta.com/developer/api/v2/stopsbyroute?api_key=wX9NwuHnZU2ToO7GmGR9uw&route=931_&format=json


```
{
	direction: [{
		direction_id: "0",
		direction_name: "Southbound",
		stop: [{
			stop_order: "1",
			stop_id: "70061",
			stop_name: "Alewife Station Red Line",
			parent_station: "place-alfcl",
			parent_station_name: "Alewife Station",
			stop_lat: "42.3954277038574",
			stop_lon: "-71.1424865722656"
		},
		{
			stop_order: "2",
			stop_id: "70063",
			stop_name: "Davis Sq - Inbound",
			parent_station: "place-davis",
			parent_station_name: "Davis Station",
			stop_lat: "42.3967399597168",
			stop_lon: "-71.1218185424805"
		},
		.
		.
		.
		]
	},
	{
		direction_id: "1",
		direction_name: "Northbound",
		stop: [{
			stop_order: "1",
			stop_id: "70094",
			stop_name: "Ashmont Station Red Line Inbound",
			parent_station: "place-asmnl",
			parent_station_name: "Ashmont Station",
			stop_lat: "42.2846527099609",
			stop_lon: "-71.0644912719727"
		},
		{
			stop_order: "2",
			stop_id: "70092",
			stop_name: "Shawmut Station - Inbound",
			parent_station: "place-smmnl",
			parent_station_name: "Shawmut Station",
			stop_lat: "42.2931251525879",
			stop_lon: "-71.0657348632813"
		},
		.
		.
		.
		]
	}]
}
```

4.2.2 STOPSBYLOCATION

This query will return a list of the nearest stops from a particular location. Up to 15 are returned, within a 1-mile radius.

Special Parameters

Name	Description
lat	The latitude for location near which stops should be returned Data type: Float Example: “42.352913”
lon	The longitude for location near which stops should be returned Data type: Float Example: “-71.064648”

Response Fields

The responses from stopsbyroute and stopsbylocation are organized a little differently. The stop objects themselves only have one difference – the stop_order property in stopsbyroute is replaced by the distance property in stopsbylocation.

Name	Description
stop_list	Root object of the response
lat	Property of stop_list. String representation of a Float. The latitude for location near which stops are returned Example: “42.352913”
lon	Property of stop_list. String representation of a Float. The longitude for location near which stops are returned Example: “-71.064648”
stop	Child object of stop_list. Contains all information for a stop near the location
stop_id	Property of “stop.” String. The GTFS-compatible unique identifier for the stop Example: “70063”
stop_name	Property of “stop.” String. The GTFS-compatible name for the stop (not unique) Example: “Davis Sq - Inbound”
parent_station	Property of “stop.” String. 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”
parent_station_name	Property of “stop.” String. The human-readable name for the larger station associated with the stop. (note: can be empty if stop does not have an associated station) Example: “Davis Station”
stop_lat	Property of “stop.” String representation of a Float. The GTFS-compatible latitude of the station. Example: “42.3967399597168”

stop_lon	Property of “stop.” String representation of a Float. The GTFS-compatible longitude of the station. Example: “-71.1218185424805”
distance	Property of “stop.” Float. The distance of the stop from the requested location in miles Example: “0.00800655130296946”

Notes

- The “parent_station” and “parent_station_name” properties can be empty if stop does not have an associated parent station.
- Up to 15 stops in a one-mile radius of the location are returned ordered in ascending order of distance from the location.
- The lat and lon properties of the stop_list are currently not displayed. They will be displayed as properties of the stop_list in a future update.

Example

http://realtime.mbta.com/developer/api/v2/stopsbylocation?api_key=wX9NwuHnZU2ToO7GmGR9uw&lat=42.352913&lon=-71.064648&format=json


```
{
	stop: [{
		stop_id: "place-boyls",
		stop_name: "Boylston Station",
		parent_station: "",
		parent_station_name: "",
		stop_lat: "42.3530197143555",
		stop_lon: "-71.0645904541016",
		distance: "0.00800655130296946"
	},
	{
		stop_id: "70159",
		stop_name: "Boylston Station - Outbound",
		parent_station: "place-boyls",
		parent_station_name: "Boylston Station",
		stop_lat: "42.3530197143555",
		stop_lon: "-71.0645904541016",
		distance: "0.00800655130296946"
	},
	.
	.
	.
	]
}
```


4.3 Schedule Queries: schedulebystop, schedulebyroute, schedulebytrip

The schedule queries retrieve scheduled arrival and departure times at a stop, along a route, or for a specified trip. The information returned is organized as follows:

schedulebystop: “modes” contain “routes” contain “directions” contain “trips”

schedulebyroute: “route” contains “directions” contain “trips” contain “stops”

schedulebytrip: “trip” contains “stops”

4.3.1 SCHEDULEBYSTOP

This query will return scheduled arrivals and departures for a direction and route for a particular stop.

Special Parameters

Name	Description
stop	GTFS-compatible stop_id value for which schedule should be returned. String. Example: “Back Bay”
route (optional)	GTFS-compatible route_id value on the stop for which schedule should be returned. String. If not included then schedule for all routes serving the stop will be returned Example: “CR-Providence”
direction (optional)	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 (optional)	Epoch time after which schedule should be returned. Integer. 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 (optional)	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 (optional)	Defines number of trips to return. Integer between 1 and 100. If not included defaults to 5. Example: “100”

Response Fields

Name	Description
schedule	Root object of the response
stop_id	Property of “schedule.” String. The GTFS-compatible unique identifier for the stop for which the schedule is returned Example: “Back Bay”
stop_name	Property of “schedule.” String. The GTFS-compatible name for the stop for which the schedule is returned Example: “Back Bay”
mode	Child object of “schedule.” Contains information for a mode that serves this stop
route_type	Property of “mode.” String representation of an Integer (0-7) The GTFS-compatible identifier for the type of service (mode) Example: “2”

mode_name	Property of “mode.” String. The human-readable name for the type of service (mode) Example: “Commuter Rail”
route	Child object of “mode.” Contains information for a route that serves this stop
route_id	Property of “route.” String. The unique GTFS-compatible identifier for the route Example: “CR-Providence”
route_name	Property of “route.” String. The human-readable name for the route Example: “Providence/Stoughton Line”
direction	Child object of “route.” Contains information for a direction of the route
direction_id	Property of “direction.” String representation of a Bit (0 or 1). The GTFS-compatible identifier for the direction Example: “0”
direction_name	Property of “direction.” String. The human-readable name for the direction Example: “Outbound”
trip	Child object of “direction.” Contains information for a trip on a direction of the route
trip_id	Property of “trip.” String. The unique GTFS-compatible identifier for the trip Example: “CR-Providence-CR-Weekday-815”
trip_name	Property of “trip.” String. The human-readable name for the trip Example: “815 (4:35 pm from South Station)”
sch_arr_dt	Property of “trip.” String representation of an integer. Scheduled arrival time at the stop for the trip, in epoch time Example: “1361989260”
sch_dep_dt	Property of the “trip” object. String representation of an integer. Scheduled departure time at the stop for the trip, in epoch time Example: “1361989260”

Example

http://realtime.mbta.com/developer/api/v2/schedulebystop?api_key=wX9NwuHnZU2ToO7GmGR9uw&stop=Back%20Bay&route=CR-Providence&direction=0&format=json


```
{
	stop_id: "Back Bay",
	stop_name: "Back Bay",
	mode: [{
		route_type: "2",
		mode_name: "Commuter Rail",
		route: [{
			route_id: "CR-Franklin",
			route_name: "Franklin Line",
```

```
                        direction: [{
                               direction_id: "0",
                               direction_name: "Outbound",
                               trip: [{
                                      trip_id: "CR-Franklin-CR-Weekday-Franklin-Dec13-717",
                                      trip_name: "717 (4:20 pm from South Station)",
                                      sch_arr_dt: "1403555100",
                                      sch_dep_dt: "1403555100"
                               },
                               {
                                      trip_id: "CR-Franklin-CR-Weekday-Franklin-Dec13-715",
                                      trip_name: "715 (3:55 pm from South Station)",
                                      sch_arr_dt: "1403553600",
                                      sch_dep_dt: "1403553600"
                               }]
                        }]
                 },
                 {
                        route_id: "CR-Needham",
                        route_name: "Needham Line",
                        direction: [{
                               direction_id: "0",
                               direction_name: "Outbound",
                               trip: [{
                                      trip_id: "CR-Needham-CR-Weekday-Needham-Dec13-619",
                                      trip_name: "619 (4:00 pm from South Station)",
                                      sch_arr_dt: "1403553900",
                                      sch_dep_dt: "1403553900"
                               }]
                        },
                        .
                        .
                        .
                        {
                               direction_id: "1",
                               direction_name: "Inbound",
                               trip: [{
                                      trip_id: "CR-Needham-CR-Weekday-Needham-Dec13-618",
                                      trip_name: "618 (3:50 pm from Needham Heights)",
                                      sch_arr_dt: "1403554920",
                                      sch_dep_dt: "1403554920"
                               },
                               {
                                      trip_id: "CR-Needham-CR-Weekday-Needham-Dec13-616",
                                      trip_name: "616 (3:05 pm from Needham Heights)",
                                      sch_arr_dt: "1403552700",
                                      sch_dep_dt: "1403552700"
                               }]
                        }]
                 }]
          }]
}
```


4.3.2 SCHEDULEBYROUTE

This query will return the scheduled arrivals and departures in a direction for a particular route.

Special Parameters

Name	Description

route	GTFS-compatible route_id value for which schedule should be returned Data type: String Example: “CR-Providence”
direction (optional)	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 Data type: Bit (0 or 1) Example: “0”
datetime (optional)	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 Data type: Integer Example: “1361989200”
max_time (optional)	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 (optional)	Defines number of trips to return. Integer between 1 and 100. If not included defaults to 5. Example: “100”

Response Fields

Name	**Description**
schedule	Root object of the response
route_id	Property of “schedule.” String. The unique GTFS-compatible identifier for the route for which the schedule is returned Example: “CR-Providence”
route_name	Property of “schedule.” String. The human-readable name for the route for which the schedule is returned Example: “Providence/Stoughton Line”
direction	Child object of schedule. Contains information for a direction of the route
direction_id	Property of “direction.” String representation of a Bit (0 or 1) The GTFS-compatible identifier for the direction Example: “0”
direction_name	Property of “direction.” String. The human-readable name for the direction Example: “Outbound”
trip	Child object of “direction.” Contains information for a trip on a direction of the route
trip_id	Property of “trip.” String. The unique GTFS-compatible identifier for the trip Example: “CR-Providence-CR-Weekday-815”

trip_name	Property of “trip.” String. The human-readable for the trip Example: “815 (4:35 pm from South Station)”
stop	Child object of “trip.” Contains information for a stop on the trip
stop_sequence	Property of “stop.” String representation of an Integer (starting with 1). Identifies where the stop comes in the sequence of stops for this trip Example: “2”
stop_id	Property of “stop.” String. The GTFS-compatible unique identifier for the stop Example: “Back Bay”
stop_name	Property of “stop.” String. The GTFS-compatible name for the stop Example: “Back Bay”
sch_arr_dt	Property of “stop.” String representation of an Integer. Scheduled arrival time at the stop for the trip, in epoch time Example: “1361986080”
sch_dep_dt	Property of “stop.” String representation of an Integer. Scheduled departure time at the stop for the trip, in epoch time Example: “1361986080”

Example

http://realtime.mbta.com/developer/api/v2/schedulebyroute?api_key=wX9NwuHnZU2ToO7GmGR9uw&route=CR-Providence&direction=0&format=json


```
{
	route_id: "CR-Providence",
	route_name: "Providence/Stoughton Line",
	direction: [{
		direction_id: "0",
		direction_name: "Outbound",
		trip: [{
			trip_id: "CR-Providence-CR-Weekday-Providence-Dec13-811",
			trip_name: "811 (3:30 pm from South Station)",
			stop: [{
				stop_sequence: "1",
				stop_id: "South Station",
				stop_name: "South Station",
				sch_arr_dt: "1403551800",
				sch_dep_dt: "1403551800"
			},
			{
				stop_sequence: "2",
				stop_id: "Back Bay",
				stop_name: "Back Bay",
				sch_arr_dt: "1403552100",
				sch_dep_dt: "1403552100"
			},
			{
				stop_sequence: "3",
				stop_id: "Ruggles",
				stop_name: "Ruggles",
```

```
                        sch_arr_dt: "1403552280",
                        sch_dep_dt: "1403552280"
                },
                .
                .
                .
                {
                        stop_sequence: "11",
                        stop_id: "Attleboro",
                        stop_name: "Attleboro",
                        sch_arr_dt: "1403555100",
                        sch_dep_dt: "1403555100"
                },
                {
                        stop_sequence: "12",
                        stop_id: "South Attleboro",
                        stop_name: "South Attleboro",
                        sch_arr_dt: "1403555700",
                        sch_dep_dt: "1403555700"
                },
                {
                        stop_sequence: "13",
                        stop_id: "Providence",
                        stop_name: "Providence",
                        sch_arr_dt: "1403556300",
                        sch_dep_dt: "1403556300"
                }]
        },
        {
                trip_id: "CR-Providence-CR-Weekday-Providence-Dec13-917",
                trip_name: "917 (4:00 pm from South Station)",
                stop: [{
                        stop_sequence: "1",
                        stop_id: "South Station",
                        stop_name: "South Station",
                        sch_arr_dt: "1403553600",
                        sch_dep_dt: "1403553600"
                },
                .
                .
                .
                {
                        stop_sequence: "8",
                        stop_id: "Stoughton",
                        stop_name: "Stoughton",
                        sch_arr_dt: "1403555880",
                        sch_dep_dt: "1403555880"
                }]
        },
        {
                trip_id: "CR-Providence-CR-Weekday-Providence-Dec13-813",
                trip_name: "813 (4:08 pm from South Station)",
                stop: [{
                        stop_sequence: "1",
                        stop_id: "South Station",
                        stop_name: "South Station",
                        sch_arr_dt: "1403554080",
                        sch_dep_dt: "1403554080"
                },
                .
                .
                .
                {
```

```
                            stop_sequence: "15",
                            stop_id: "Wickford Junction",
                            stop_name: "Wickford Junction",
                            sch_arr_dt: "1403560440",
                            sch_dep_dt: "1403560440"
                        }]
                }]
        }]
}
```


4.3.3 SCHEDULEBYTRIP

This query will return the scheduled arrivals and departures for a particular trip.

Special Parameters

Name	Description
trip	GTFS-compatible trip_id value for which schedule should be returned Data type: String Example: “CR-Providence-CR-Weekday-807”
datetime (optional)	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 Data type: Integer Example: “1361989200”

Response Fields

Name	Description
schedule	Root object of the response
route_id	Property of “schedule.” String. The unique GTFS-compatible identifier for the route for which schedule is returned Example: “CR-Providence”
route_name	Property of “schedule.” String. The human-readable name for the route for which schedule is returned Example: “Providence/Stoughton Line”
trip_id	Property of “schedule.” String. The unique GTFS-compatible identifier for the trip for which schedule is returned Example: “CR-Providence-CR-Weekday-815”
trip_name	Property of “schedule.” String. The human-readable for the trip for which schedule is returned Example: “815 (4:35 pm from South Station)”
direction_id	Property of “schedule.” String representation of a Bit (0 or 1). The GTFS-compatible identifier for the direction Example: “0”
direction_name	Property of “schedule.” String. The human-readable name for the direction Example: “Outbound”
stop	Child object of “Schedule.” Contains information for a stop on the trip

stop_sequence	Property of “stop.” String representation of an Integer (starting with 1) Identifies where the stop comes in the sequence of stops for this trip Example: “2”
stop_id	Property of “stop.” String. The GTFS-compatible unique identifier for the stop Example: “Back Bay”
stop_name	Property of “stop.” String. The GTFS-compatible name for the stop Example: “Back Bay”
sch_arr_dt	Property of “stop.” String representation of an integer. Scheduled arrival time at the stop for the trip, in epoch time Example: “1361986080”
sch_dep_dt	Property of “stop.” String representation of an integer. Scheduled departure time at the stop for the trip, in epoch time Example: “1361986080”

Example

http://realtime.mbta.com/developer/api/v2/schedulebytrip?api_key=wX9NwuHnZU2ToO7GmGR9uw&trip=CR-Providence-CR-Weekday-Providence-Dec13-813&format=json


```
{
	route_id: "CR-Providence",
	route_name: "Providence/Stoughton Line",
	trip_id: "CR-Providence-CR-Weekday-Providence-Dec13-813",
	trip_name: "813 (4:08 pm from South Station)",
	direction_id: "0",
	direction_name: "Outbound",
	stop: [{
		stop_sequence: "1",
		stop_id: "South Station",
		stop_name: "South Station",
		sch_arr_dt: "1403554080",
		sch_dep_dt: "1403554080"
	},
	{
		stop_sequence: "2",
		stop_id: "Back Bay",
		stop_name: "Back Bay",
		sch_arr_dt: "1403554380",
		sch_dep_dt: "1403554380"
	},
	{
		stop_sequence: "3",
		stop_id: "Ruggles",
		stop_name: "Ruggles",
		sch_arr_dt: "1403554560",
		sch_dep_dt: "1403554560"
	},
	.
	.
	.
	{
		stop_sequence: "15",
		stop_id: "Wickford Junction",
		stop_name: "Wickford Junction",
```

```
            sch_arr_dt: "1403560440",
            sch_dep_dt: "1403560440"
        }]
}
```

4.4 Prediction and Vehicle Locations queries: predictions and vehicles by route, trip, stop

These queries return information organized in a manner much like the equivalent “schedule” queries:

predictionsbystop: ‘modes’ contain ‘routes’ contain ‘directions’ contain ‘trips’ contain ‘vehicles’

predictionsbyroute: ‘route’ contains ‘directions’ contain ‘trips’ contain ‘vehicles’ and ‘stops’

vehiclesbyroute: ‘route’ contains ‘directions’ contain ‘trips’ contain ‘vehicles’

predictionsbytrip: ‘trip’ contains ‘vehicles’ and ‘stops’

vehiclesbytrip: ‘trip’ contains ‘vehicles’

The “vehicles” queries don’t return anything not returned by the “predictions” queries, but if vehicle locations are all you need then the “vehicles” queries are a lower-overhead alternative.

These queries only return data about trips for which real-time information is available. Data may be unavailable for a trip because the particular service does not yet have real-time data, or because of a technical problem, or because the trip is not running; in these cases the trip is omitted. (Because of this the predictions calls are not substitutes for the schedule calls, even though schedule data is included.)

The “predictions” queries include alert headers for any alerts relevant to the returned predictions. Alerts can be some of the most important information for users, because they describe something out of the ordinary, a change the user would not know to expect. Each alert has a summary and an ID can be used to retrieve a much richer set of information using the “alert” query.

4.4.1 PREDICTIONSBYSTOP

This query will return predicted arrivals and departures in the next hour for a direction and route for a particular stop.

Special Parameters

Name	Description
stop	GTFS-compatible stop_id value for which predictions should be returned Data type: String Example: “Back Bay”
include_access_alerts (optional)	Whether or not alerts pertaining to accessibility (elevators, escalators) should be returned Data type: String representation of Boolean Possible values: “true” or “false”; default value: “false” If not included, then alerts pertaining to accessibility are not returned
include_service_alerts (optional)	Whether or not service alerts should be returned Data type: String representation of Boolean Possible values: “true” or “false”; default value: “true” If not included, then service alerts will be returned

Response Fields

Name	Description
prediction	Root object of the response
stop_id	Property of “prediction.” String. The GTFS-compatible unique identifier for the stop for which the predictions are returned Example: “Back Bay”
stop_name	Property of “prediction.” String. The GTFS-compatible name for the stop for which the predictions are returned Example: “Back Bay”
mode	Child object of “prediction.” Contains information for a mode that serves this stop
route_type	Property of “mode.” String representation of an Integer (0-7) The GTFS-compatible identifier for the type of service (mode) Example: “2”
mode_name	Property of “mode.” String. The human-readable name for the type of service (mode) Example: “Commuter Rail”
route	Child object of “mode.” Contains information for a route that serves this stop
route_id	Property of “route.” String. The unique GTFS-compatible identifier for the route Example: “CR-Providence”
route_name	Property of “route.” String. The human-readable name for the route Example: “Providence/Stoughton Line”
direction	Child object of “route.” Contains information for a direction of the route
direction_id	Property of “direction.” String representation of a Bit (0 or 1) The GTFS-compatible identifier for the direction Example: “0”
direction_name	Property of “direction.” String. The human-readable name for the direction Example: “Outbound”
trip	Child object of “direction.” Contains information for a trip on a direction of the route
trip_id	Property of “trip.” String. The unique GTFS-compatible identifier for the trip Example: “CR-Providence-CR-Weekday-815”
trip_name	Property of “trip.” String. The human-readable for the trip Example: “815 (4:35 pm from South Station)”

trip_headsign	Property of “trip.” String. The text that identifies the trip's destination to passengers Example: “North Station”
sch_arr_dt	Property of “trip.” String representation of an integer. Scheduled arrival time at the stop for the trip, in epoch time Example: “1361989260”
sch_dep_dt	Property of “trip.” String representation of an Integer. Scheduled departure time at the stop for the trip, in epoch time Example: “1361989260”
pre_dt	Property of “trip.” String representation of an Integer. 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	Property of “trip.” String representation of an Integer. Predicted amount of time until the vehicle arrives at the stop, in seconds Example: “339”
vehicle	Child object of “trip.” Contains information for a vehicle on the trip
vehicle_id	Property of “vehicle.” String. The GTFS-compatible unique identifier for the vehicle Example: “1531”
vehicle_lat	Property of “vehicle.” String representation of a Float. The GTFS-compatible latitude of the vehicle. Example: “42.08997”
vehicle_lon	Property of “vehicle.” String representation of a Float. The GTFS-compatible longitude of the vehicle. Example: “-71.4388”
vehicle_bearing (optional)	Property of “vehicle.” String representation of a Float. GTFS-compatable bearing of the vehicle. This can be the compass bearing, or the direction towards the next stop or intermediate location Example: “259”
vehicle_speed (optional)	Property of “vehicle.” String representation of a Float. Identifies the vehicle’s momentary speed, in meters per second Example: “21”
vehicle_timestamp	Property of “vehicle.” String representation of an Integer Identifies the moment when the content of this feed has been created, in epoch time Example: “1400855704”
alert_headers	Child object of “predictions” Contains a list of all alerts applicable to predictions returned by query
alert (optional)	Child object of “alert_headers” Contains information about one alert applicable to predictions returned by query
alert_id	Property of “alert.” Integer. The unique identifier for the alert Example: “781”
header_text	Property of “alert.” String.

	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”
effect_name	Property of “alert.” String. The human-readable name for the effect Example: “Shuttle bus”

Example

http://realtime.mbta.com/developer/api/v2/predictionsbystop?api_key=wX9NwuHnZU2ToO7GmGR9uw&stop=Providence&format=json

Example Request

JSON Response:


```
{
	stop_id: "Providence",
	stop_name: "Providence",
	mode: [{
		route_type: "2",
		mode_name: "Commuter Rail",
		route: [{
			route_id: "CR-Providence",
			route_name: "Providence/Stoughton Line",
			direction: [{
				direction_id: "1",
				direction_name: "Inbound",
				trip: [{
					trip_id: "CR-Providence-CR-Weekday-Providence-Dec13-
818",
					trip_name: "818 (1:10 pm from Wickford Junction)",
					trip_headsign: "South Station (Train 818)",
					sch_arr_dt: "1403545260",
					sch_dep_dt: "1403545260",
					vehicle: {
						vehicle_id: "1514",
						vehicle_lat: "41.83049",
						vehicle_lon: "-71.41402",
						vehicle_speed: "2",
						vehicle_timestamp: "1403546265"
					}
				}]
			}]
		}]
	}],
	alert_headers: [{
		alert_id: "28238",
		header_text: "Providence Line running with delays due to weather”,
		effect_name: "Delay”
	}]
}
```

4.4.2 PREDICTIONSBYROUTE

This query will return predictions for upcoming trips (including trips already underway) in a direction for a particular route.

Special Parameters

Name	Description
route	GTFS-compatible route_id value for which predictions should be returned Data type: String Example: “931_”
include_access_alerts (optional)	Whether or not alerts pertaining to accessibility (elevators, escalators) should be returned Data type: String representation of Boolean Possible values: “true” or “false”; default value: “false” If not included, then alerts pertaining to accessibility are not returned
include_service_alerts (optional)	Whether or not service alerts should be returned Data type: String representation of Boolean Possible values: “true” or “false”; default value: “true” If not included, then service alerts will be returned

Response Fields

Name	Description
predictions	Root object of the response
route_id	Property of “predictions.” String. The unique GTFS-compatible identifier for the route for which predictions are returned Example: “CR-Franklin”
route_name	Property of “predictions.” String. The human-readable name for the route for which predictions are returned Example: “Franklin Line”
route_type	Property of “predictions.” String representation of an Integer (0-7) The GTFS-compatible identifier for the type of service (mode) Example: “2”
mode_name	Property of “predictions.” String. The human-readable name for the type of service (mode) Example: “Commuter Rail”
direction	Child object of “predictions.” Contains information for a direction of the route
direction_id	Property of “direction.” String representation of a Bit (0 or 1). The GTFS-compatible identifier for the direction Example: “0”
direction_name	Property of “direction.” String. The human-readable name for the direction Example: “Outbound”
trip	Child object of “direction.” Contains information for a trip on a direction of the route

Name	Description
trip_id	Property of “trip.” String. The unique GTFS-compatible identifier for the trip Example: “CR-Providence-CR-Weekday-815”
trip_name	Property of “trip.” String. The human-readable for the trip Example: “815 (4:35 pm from South Station)”
trip_headsign	Property of “trip.” String. The text that identifies the trip's destination to passengers Example: “North Station”
vehicle	Child object of “trip.” Contains information for a vehicle on the trip
vehicle_id	Property of “vehicle.” String. The GTFS-compatible unique identifier for the vehicle Example: “1531”
vehicle_lat	Property of “vehicle.” String representation of a Float. The GTFS-compatible latitude of the vehicle. Example: “42.08997”
vehicle_lon	Property of “vehicle.” String representation of a Float. The GTFS-compatible longitude of the vehicle. Example: “-71.4388”
vehicle_bearing (optional)	Property of “vehicle.” String representation of a Float. GTFS-compatable bearing of the vehicle. This can be the compass bearing, or the direction towards the next stop or intermediate location Example: “259”
vehicle_speed (optional)	Property of “vehicle.” String representation of a Float. Identifies the vehicle’s momentary speed, in meters per second Data type: Float Example: “21”
vehicle_timestamp	Property of “vehicle.” String representation of an Integer Identifies the moment at which the vehicle’s real-time progress was measured, in epoch time Example: “1400855704”
stop	Child object of “trip.” Contains information for a stop on the trip
stop_sequence	Property of “stop.” String representation of an Integer (starting with 1) Identifies where the stop comes in the sequence of stops for this trip Example: “2”
stop_id	Property of “stop.” String. The GTFS-compatible unique identifier for the stop Example: “Back Bay”
stop_name	Property of “stop.” String. The GTFS-compatible name for the stop Example: “Back Bay”

Name	Description
sch_arr_dt	Property of “stop.” String representation of an Integer. Scheduled arrival time at the stop for the trip, in epoch time Example: “1361986080”
sch_dep_dt	Property of “stop.” String representation of an Integer. Scheduled departure time at the stop for the trip, in epoch time Example: “1361986080”
pre_dt	Property of “stop.” String representation of an Integer. 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	Property of “stop.” String representation of an Integer. Predicted amount of time until the vehicle arrives at the stop, in seconds Example: “339”
alert_headers	Child object of “predictions” Contains a list of all alerts applicable to predictions returned by query
alert (optional)	Child object of “alert_headers” Contains information about one alert applicable to predictions returned by query
alert_id	Property of “alert.” Integer. The unique identifier for the alert Example: “781”
header_text	Property of “alert.” String. 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”
effect_name	Property of “alert.” String. The human-readable name for the effect Example: “Shuttle bus”

Example

http://realtime.mbta.com/developer/api/v2/predictionsbyroute?api_key=wX9NwuHnZU2ToO7GmGR9uw&route=CR-Providence&format=json


```
{
	route_id: "CR-Providence",
	route_name: "Providence/Stoughton Line",
	route_type: "2",
	mode_name: "Commuter Rail",
	direction: [{
		direction_id: "0",
		direction_name: "Outbound",
		trip: [{
			trip_id: "CR-Providence-CR-Weekday-Providence-Dec13-913",
			trip_name: "913 (1:20 pm from South Station)",
			trip_headsign: "Stoughton (Train 913)",
			vehicle: {
				vehicle_id: "1524",
				vehicle_lat: "42.22306",
				vehicle_lon: "-71.14098",
				vehicle_bearing: "199",
				vehicle_speed: "45",
```

```
                        vehicle_timestamp: "1403545530"
                },
                stop: [{
                        stop_sequence: "5",
                        stop_id: "Route 128",
                        stop_name: "Route 128",
                        sch_arr_dt: "1403545620",
                        sch_dep_dt: "1403545620"
                },
                {
                        stop_sequence: "6",
                        stop_id: "Canton Junction",
                        stop_name: "Canton Junction",
                        sch_arr_dt: "1403545980",
                        sch_dep_dt: "1403545980",
                        pre_dt: "1403545980",
                        pre_away: "266"
                },
                {
                        stop_sequence: "7",
                        stop_id: "Canton Center",
                        stop_name: "Canton Center",
                        sch_arr_dt: "1403546160",
                        sch_dep_dt: "1403546160",
                        pre_dt: "1403546160",
                        pre_away: "446"
                }]
        }]
    },
    {
        direction_id: "1",
        direction_name: "Inbound",
        trip: [{
                trip_id: "CR-Providence-CR-Weekday-Providence-Dec13-818",
                trip_name: "818 (1:10 pm from Wickford Junction)",
                trip_headsign: "South Station (Train 818)",
                vehicle: {
                        vehicle_id: "1514",
                        vehicle_lat: "41.77901",
                        vehicle_lon: "-71.42352",
                        vehicle_bearing: "10",
                        vehicle_speed: "39",
                        vehicle_timestamp: "1403545543"
                },
                stop: [{
                        stop_sequence: "3",
                        stop_id: "Providence",
                        stop_name: "Providence",
                        sch_arr_dt: "1403545260",
                        sch_dep_dt: "1403545260",
                        pre_dt: "1403545860",
                        pre_away: "146"
                },
                {
                        stop_sequence: "4",
                        stop_id: "South Attleboro",
                        stop_name: "South Attleboro",
                        sch_arr_dt: "1403545920",
                        sch_dep_dt: "1403545920",
                        pre_dt: "1403546520",
                        pre_away: "806"
                },
                {
```

```
                        stop_sequence: "5",
                        stop_id: "Attleboro",
                        stop_name: "Attleboro",
                        sch_arr_dt: "1403546520",
                        sch_dep_dt: "1403546520",
                        pre_dt: "1403547120",
                        pre_away: "1406"
                    }]
            }]
      }],
      stats: "173"
}
```


4.4.3 VEHICLESBYROUTE

This query will return vehicle positions for upcoming trips (including trips already underway) in a direction for a particular route.

Special Parameters

Name	Description
route	GTFS-compatible route_id value for which vehicle positions should be returned Data type: String Example: “931_”

Response Fields

Name	Description
vehicles	Root object of the response
route_id	Property of “vehicles.” String. The unique GTFS-compatible identifier for the route for which vehicle positions are returned Example: “CR-Franklin”
route_name	Property of “vehicles.” String. The human-readable name for the route for which vehicle positions are returned Example: “Franklin Line”
route_type	Property of “vehicles.” String representation of an Integer (0-7). The GTFS-compatible identifier for the type of service (mode) Example: “2”
mode_name	Property of “vehicles.” String. The human-readable name for the type of service (mode) Example: “Commuter Rail”
direction	Child object of “vehicles.” Contains information for a direction of the route
direction_id	Property of “direction.” String representation of a Bit (0 or 1). The GTFS-compatible identifier for the direction Example: “0”

Name	Description
direction_name	Property of “direction.” String. The human-readable name for the direction Example: “Outbound”
trip	Child object of “direction.” Contains information for a trip on a direction of the route
trip_id	Property of “trip.” String. The unique GTFS-compatible identifier for the trip Example: “CR-Providence-CR-Weekday-815”
trip_name	Property of “trip.” String. The human-readable for the trip Example: “815 (4:35 pm from South Station)”
trip_headsign	Property of “trip.” String. The text that identifies the trip's destination to passengers Example: “North Station”
vehicle	Child object of “trip.” Contains information for a vehicle on the trip
vehicle_id	Property of “vehicle.” String. The GTFS-compatible unique identifier for the vehicle Example: “1531”
vehicle_lat	Property of “vehicle.” String representation of a Float. The GTFS-compatible latitude of the vehicle. Example: “42.08997”
vehicle_lon	Property of “vehicle.” String representation of a Float. The GTFS-compatible longitude of the vehicle. Example: “-71.4388”
vehicle_bearing (optional)	Property of “vehicle.” String representation of a Float GTFS-compatable bearing of the vehicle. This can be the compass bearing, or the direction towards the next stop or intermediate location Example: “259”
vehicle_speed (optional)	Property of “vehicle.” String representation of a Float. Identifies the vehicle’s momentary speed, in meters per second Example: “21”
vehicle_timestamp	Property of “vehicle.” String representation of an Integer. Identifies the moment when the content of this feed has been created, in epoch time Example: “1400855704”

Example

JSON Request:

http://realtime.mbta.com/developer/api/v2/vehiclesbyroute?api_key=wX9NwuHnZU2ToO7GmGR9uw&route=CR-Providence&format=json

JSON Response:

```
{
        route_id: "CR-Providence",
```

```
      route_name: "Providence/Stoughton Line",
      route_type: "2",
      mode_name: "Commuter Rail",
      direction: [{
            direction_id: "0",
            direction_name: "Outbound",
            trip: [{
                  trip_id: "CR-Providence-CR-Weekday-Providence-Dec13-913",
                  trip_name: "913 (1:20 pm from South Station)",
                  trip_headsign: "Stoughton (Train 913)",
                  vehicle: {
                        vehicle_id: "1524",
                        vehicle_lat: "42.133",
                        vehicle_lon: "-71.12097",
                        vehicle_bearing: "156",
                        vehicle_speed: "37",
                        vehicle_timestamp: "1403546582"
                  }
            }]
      },
      {
            direction_id: "1",
            direction_name: "Inbound",
            trip: [{
                  trip_id: "CR-Providence-CR-Weekday-Providence-Dec13-818",
                  trip_name: "818 (1:10 pm from Wickford Junction)",
                  trip_headsign: "South Station (Train 818)",
                  vehicle: {
                        vehicle_id: "1514",
                        vehicle_lat: "41.85787",
                        vehicle_lon: "-71.4068",
                        vehicle_bearing: "20",
                        vehicle_timestamp: "1403546580"
                  }
            }]
      }]
}
```


4.4.4 PREDICTIONSBYTRIP

This query will return the predicted arrivals and departures for a particular trip.

Special Parameters

Name	Description
trip	GTFS-compatible trip_id value for which predictions should be returned Data type: String Example: “CR-Providence-CR-Weekday-807”

Response Fields

Name	Description
prediction	Root object of the response.
route_id	Property of “prediction.” String. The unique GTFS-compatible identifier for the route for which predictions are returned Example: “CR-Providence”

route_name	Property of “prediction.” String. The human-readable name for the route for which predictions are returned Example: “Providence/Stoughton Line”
route_type	Property of “prediction.” String representation of an Integer (0-7). The GTFS-compatible identifier for the type of service (mode) Example: “2”
mode_name	Property of “prediction.” String. The human-readable name for the type of service (mode) Example: “Commuter Rail”
trip_id	Property of “prediction.” String. The unique GTFS-compatible identifier for the trip for which predictions are returned Example: “CR-Providence-CR-Weekday-815”
trip_name	Property of “prediction.” String. The human-readable for the trip for which schedule is returned Example: “815 (4:35 pm from South Station)”
trip_headsign	Property of “prediction.” String. The text that identifies the trip's destination to passengers Example: “North Station”
direction_id	Property of “prediction.” String representation of a Bit (0 or 1). The GTFS-compatible identifier for the direction Example: “0”
direction_name	Property of “prediction.” String. The human-readable name for the direction Example: “Outbound”
vehicle	Child object of “prediction.” Contains information for a vehicle on the trip
vehicle_id	Property of “vehicle.” String. The GTFS-compatible unique identifier for the vehicle Example: “1531”
vehicle_lat	Property of “vehicle.” String representation of a Float. The GTFS-compatible latitude of the vehicle. Example: “42.08997”
vehicle_lon	Property of “vehicle.” String representation of a Float. The GTFS-compatible longitude of the vehicle. Example: “-71.4388”
vehicle_bearing (optional)	Property of “vehicle.” String representation of a Float. GTFS-compatable bearing of the vehicle. This can be the compass bearing, or the direction towards the next stop or intermediate location Example: “259”
vehicle_speed (optional)	Property of “vehicle.” String representation of a Float. Identifies the vehicle’s momentary speed, in meters per second Example: “21”

vehicle_timestamp	Property of “vehicle.” String representation of an Integer. Identifies the moment when the content of this feed has been created, in epoch time Example: “1400855704”
stop	Child object of “trip.” Contains information for a stop on the trip
stop_sequence	Property of “stop.” String representation of an Integer (starting with 1) Identifies where the stop comes in the sequence of stops for this trip Example: “2”
stop_id	Property of “stop.” String. The GTFS-compatible unique identifier for the stop Example: “Back Bay”
stop_name	Property of “stop.” String. The GTFS-compatible name for the stop Example: “Back Bay”
sch_arr_dt	Property of “stop.” String representation of an Integer. Scheduled arrival time at the stop for the trip, in epoch time Example: “1361986080”
sch_dep_dt	Property of “stop.” String representation of an Integer. Scheduled departure time at the stop for the trip, in epoch time Example: “1361986080”
pre_dt	Property of “stop.” String representation of an Integer. 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	Property of “stop.” String representation of an Integer. Predicted amount of time until the vehicle arrives at the stop, in seconds Example: “339”

Example

http://realtime.mbta.com/developer/api/v2/predictionsbytrip?api_key=wX9NwuHnZU2ToO7GmGR9uw&trip=CR-Providence-CR-Weekday-Providence-Dec13-913&format=json


```
{
	route_id: "CR-Providence",
	route_name: "Providence/Stoughton Line",
	route_type: "2",
	mode_name: "Commuter Rail",
	trip_id: "CR-Providence-CR-Weekday-Providence-Dec13-913",
	trip_name: "913 (1:20 pm from South Station)",
	trip_headsign: "Stoughton (Train 913)",
	direction_id: "0",
	direction_name: "Outbound",
	vehicle: {
		vehicle_id: "1524",
		vehicle_lat: "42.14926",
		vehicle_lon: "-71.13107",
		vehicle_bearing: "121",
		vehicle_speed: "36",
		vehicle_timestamp: "1403546463"
```

```
	},
	stop: [{
		stop_sequence: "7",
		stop_id: "Canton Center",
		stop_name: "Canton Center",
		sch_arr_dt: "1403546160",
		sch_dep_dt: "1403546160"
	},
	{
		stop_sequence: "8",
		stop_id: "Stoughton",
		stop_name: "Stoughton",
		sch_arr_dt: "1403546640",
		sch_dep_dt: "1403546640",
		pre_dt: "1403546640",
		pre_away: "53"
	}]
}
```


4.4.5 VEHICLESBYTRIP

This query will return the predicted vehicle positions for a particular trip.

Special Parameters

Name	Description
trip	GTFS-compatible trip_id value for which vehicle positions should be returned Data type: String Example: “CR-Providence-CR-Weekday-807”

Response Fields

Name	Description
vehicles	Root object of the response.
route_id	Property of “vehicles.” String. The unique GTFS-compatible identifier for the route for which vehicle positions are returned Example: “CR-Providence”
route_name	Property of “vehicles.” String. The human-readable name for the route for which vehicle positions are returned Example: “Providence/Stoughton Line”
route_type	Property of “vehicles.” String representation of an Integer (0-7). The GTFS-compatible identifier for the type of service (mode) Example: “2”
mode_name	Property of “vehicles.” String. The human-readable name for the type of service (mode) Example: “Commuter Rail”

trip_id	Property of “vehicles.” String. The unique GTFS-compatible identifier for the trip for which vehicle positions are returned Example: “CR-Providence-CR-Weekday-815”
trip_name	Property of “vehicles.” String. The human-readable for the trip for which schedule is returned Example: “815 (4:35 pm from South Station)”
trip_headsign	Property of “vehicles.” String. The text that identifies the trip's destination to passengers Example: “North Station”
direction_id	Property of “vehicles.” String representation of a Bit (0 or 1). The GTFS-compatible identifier for the direction Example: “0”
direction_name	Property of “vehicles.” String. The human-readable name for the direction Example: “Outbound”
vehicle	Child object “vehicles.” Contains information for a vehicle on the trip
vehicle_id	Property of “vehicle.” String. The GTFS-compatible unique identifier for the vehicle Example: “1531”
vehicle_lat	Property of “vehicle.” String representation of a Float. The GTFS-compatible latitude of the vehicle. Example: “42.08997”
vehicle_lon	Property of “vehicle.” String representation of a Float. The GTFS-compatible longitude of the vehicle. Example: “-71.4388”
vehicle_bearing (optional)	Property of “vehicle.” String representation of a Float. GTFS-compatable bearing of the vehicle. This can be the compass bearing, or the direction towards the next stop or intermediate location Example: “259”
vehicle_speed (optional)	Property of “vehicle.” String representation of a Float. Identifies the vehicle’s momentary speed, in meters per second Data type: Float Example: “21”
vehicle_timestamp	Property of “vehicle.” String representation of an Integer. Identifies the moment when the content of this feed has been created, in epoch time Example: “1400855704”

Example

http://realtime.mbta.com/developer/api/v2/vehiclesbytrip?api_key=wX9NwuHnZU2ToO7GmGR9uw&trip=CR-Providence-CR-Weekday-Providence-Dec13-818&format=json


```
{
       route_id: "CR-Providence",
```

```
        route_name: "Providence/Stoughton Line",
        route_type: "2",
        mode_name: "Commuter Rail",
        trip_id: "CR-Providence-CR-Weekday-Providence-Dec13-818",
        trip_name: "818 (1:10 pm from Wickford Junction)",
        trip_headsign: "South Station (Train 818)",
        direction_id: "1",
        direction_name: "Inbound",
        vehicle: {
                vehicle_id: "1514",
                vehicle_lat: "41.85787",
                vehicle_lon: "-71.4068",
                vehicle_bearing: "20",
                vehicle_timestamp: "1403546776"
        }
}
```

4.5 Alert queries: alerts and alertheaders by route, stop and more

The alert queries provide information about service disruptions and changes – current and upcoming, planned and unplanned. The client can request full information or just headers, impacts to service (like a detour) or access (like an elevator outage) or both, and all alerts or just alerts applicable to a stop or a route. Fields in the data returned can further be used to screen out only issues that are coming up instead of in effect right now, or alerts representing conditions that have been true for a while, or minor issues.

4.5.1 ALERTS

This query will return details for all alerts.

Special Parameters

Name	Description
include_access_alerts (optional)	Whether or not alerts pertaining to accessibility (elevators, escalators) should be returned Data type: String representation of Boolean Possible values: “true” or “false”; default value: “false” If not included, then alerts pertaining to accessibility are not returned
include_service_alerts (optional)	Whether or not service alerts should be returned Data type: String representation of Boolean Possible values: “true” or “false”; default value: “true” If not included, then service alerts will be returned

Response Fields

Name	Description
alerts	Root object of the feed
alert	Child object of “alerts.” Contains information about a single alert
alert_id	Property* of “alert.” Integer. The unique identifier for the alert Example: “2585”

Name	Description
effect_name	Property* of “alert.” Integer. The human-readable name for the effect Example: “Shuttle bus”
effect	Property* of “alert.” String. The GTFS-realtime-compatible code for the effect Example: “DETOUR”
cause_name (optional)	Property* of “alert.” String. The human-readable name for the cause Example: “maintenance”
cause	Property* of “alert.” String. The GTFS-realtime-compatible code for the cause Example: “MAINTENANCE”
header_text	Property* of “alert.” String (max 230 chars). A brief summary of the situation (GTFS-realtime-compatible) Example: “Shuttle buses replacing Red Line service from Sat Apr 27, 2013 to Sun May 26, 2013 every Saturday and Sunday from 09:00 PM to end of service due to maintenance”
short_header_text	Property* of “alert.” String (max 140 chars). 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”
url (optional)	Property* of “alert.” String (max 255 chars). A URL for extra detail (optional, GTFS-realtime-compatible) Example: “http://mbta.com/about_the_mbta/t_projects/”
Image_url (not yet supported)	Not yet supported. Property* of “alert.” String (max 255 chars). Reserved for future use as an image that helps explain the alert.
description_text (optional)	Property* of “alert.” String (max 3000 chars.) Additional details (GTFS-realtime-compatible) Example: “Affected stops: Alewife Station Davis Station Porter Square Station Harvard Square Station”
severity	Property* of “alert.” String. Possible values: “Severe”, “Moderate”, “Minor”
created_dt	Property* of “alert.” String representation of an integer. Date and time the alert was created, in epoch time Example: “1361395938”
last_modified_dt	Property* of “alert.” String representation of an integer. Date and time the alert was last modified, in epoch time Example: “1361395938”
service_effect_text	Property* of “alert.” String. Summarizes the service and the impact to that service Example: “Minor Route 1 delay”

Name	Description
timeframe_text (optional)	Property* of “alert.” String. Summarizes when an alert is in effect Example: “starting Saturday”
alert_lifecycle	Property* of “alert.” String. Identifies whether alert is a new or old, in effect or upcoming. Not intended to be human-readible. See notes. Possible values: “Upcoming”, “New”, “Ongoing”, “Ongoing-Upcoming”
banner_text (optional)	Property* of “alert.” String. Contains text to be included on a website banner (for major issue) when option is selected. Example: “All MBTA service suspended starting at 9PM due to snowstorm”
announcement_text (not yet supported)	Not yet supported. Property* of “alert.” String. Reserved for future use as the exact announcement being made in stations (as text or as speech synthesis markup language or equivalent)
effect_periods	Child object of “alert.” Contains information about all time periods for which the alert will be in effect
effect_period	Child object of “effect_periods.” Contains information about a single time period
effect_start	Property of “effect-period.” String representation of an Integer. Date and time of the start of the effect period, in epoch time Example: “1367110800”
effect_end	Property of “effect-period.” String representation of an Integer. 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_services	Child object of “alert.” Contains information about the services or elevators affected by this alert
services	Child object of “affected_services” Contains information about the services affected by this alert (either “services” or “elevators” will contain objects – see following table)
elevators	Child object of “affected_services.” Contains information about elevators/escalators affected by this alert (see table to follow)

* In XML this is an element (not an attribute as is the case with most JSON properties.)

Child Objects of “services” – appears for service alerts

Name	Description
Service (appears for service alerts, not access)	Child object of “services.” Contains information about a service affected by this alert

Name	Description
route_type (optional)	Property of “service.” Integer (0-7). GTFS-compatible code for route type (i.e. mode) Example: “1”
mode_name (optional)	Property of “service.” String. Human-readable name for the mode Example: “Subway”
route_id (optional)	Property of “service.” String. The unique GTFS-compatible identifier for the route Example: “931_”
route_name (optional)	Property of “service.” String. The human-readable name for the route Example: “Red Line”
route_hide (optional)	Property of “service.” String representation of a Boolean. Whether this route should be hidden from users. See notes. Possible values: “true”. Only included if “true.”
direction_id (optional)	Property of “service.” String representation of a Bit (0 or 1) The GTFS-compatible identifier for the direction Example: “0”
direction_name (optional)	Property of “service.” String. Human-readable direction name Example: “Westbound”
trip_id (optional)	Property of “service.” String. The GTFS-compatible unique identifier for the trip Example: “CR-Newburyport-CR-Weekday-129”
trip_name (optional)	Property of “service.” String. Human-readable trip name Example: “129 (5:00 pm from North Station)”
stop_id (optional)	Property of “service.” String. The GTFS-compatible unique identifier for the stop Example: “70061”
stop_name (optional)	Property of “service.” String. The GTFS-compatible name for the stop (not unique) Example: “Alewife Station Red Line”

Child Objects of “elevators” – appears for access alerts

Name	Description
elevator (appears for access alerts, not service)	Child object of “elevators.” Contains information about an elevator/escalator affected by this alert
elev_id	Property of “elevator.” String. Unique identifier for the elevator/escalator Example: “926”

elev_name	Property of “elevator.” String. Human-readable name for the elevator/escalator Example: “SOUTH STATION – Lobby to Street”
elev_type	Property of “elevator.” String. Type of the elevator/escalator Possible values: “Elevator”, “Escalator”, “Lift”
stop	Child object of “elevator.” Contains information about a stop related to this elevator
stop_id	Property of “stop.” String. The GTFS-compatible unique identifier for the stop Example: “70080”
stop_name	Property of “stop.” String. The GTFS-compatible name for the stop (not unique) Example: “South Station – Inbound”
parent_station	Property of “stop.” String. 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”
parent_station_name (optional)	Property of “stop.” String. The human-readable name for the larger station associated with the stop, if one exists. Can be empty if parent station does not exist. Example: “South Station”

Notes

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 – includeing 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.

Example

http://realtime.mbta.com/developer/api/v2/alerts?api_key=wX9NwuHnZU2ToO7GmGR9uw&include_access_alerts=true&include_service_alerts=true&format=json


```
{
	alerts: [{
		alert_id: 33257,
		effect_name: "Delay",
		effect: "OTHER_EFFECT",
		cause_name: "disabled train",
		cause: "TECHNICAL_PROBLEM",
		header_text: "Red Line experiencing minor southbound delays due to
disabled train",
		short_header_text: "Red Line experiencing minor southbound delays due to
disabled train",
		description_text: "Affected stops: Harvard Station - Inbound Central Sq -
Inbound",
		severity: "Minor",
		created_dt: "1403548918",
		last_modified_dt: "1403548918",
		service_effect_text: "Minor Red Line delay",
		timeframe_text: "",
		alert_lifecycle: "New",
		banner_text: "Red Line experiencing minor southbound delays due to
disabled train",
		effect_periods: [{
			effect_start: "1403548917",
			effect_end: "1403568658"
		}],
		affected_services: {
			services: [{
				route_type: "1",
				mode_name: "Subway",
				route_id: "931_",
				route_name: "Red Line",
				direction_id: "0",
				direction_name: "Southbound",
				stop_id: "70067",
				stop_name: "Harvard Station - Inbound"
			},
			{
				route_type: "1",
				mode_name: "Subway",
				route_id: "931_",
				route_name: "Red Line",
				direction_id: "0",
```

```
                        direction_name: "Southbound",
                        stop_id: "70069",
                        stop_name: "Central Sq - Inbound"
                    },
                    {
                        route_type: "1",
                        mode_name: "Subway",
                        route_id: "933_",
                        route_name: "Red Line",
                        direction_id: "0",
                        direction_name: "Southbound",
                        stop_id: "70067",
                        stop_name: "Harvard Station - Inbound"
                    },
                    {
                        route_type: "1",
                        mode_name: "Subway",
                        route_id: "933_",
                        route_name: "Red Line",
                        direction_id: "0",
                        direction_name: "Southbound",
                        stop_id: "70069",
                        stop_name: "Central Sq - Inbound"
                    }],
                    elevators: []
            }
        },
        {
            alert_id: 33258,
            effect_name: "Accessibility",
            effect: "OTHER_EFFECT",
            cause_name: "maintenance",
            cause: "MAINTENANCE",
            header_text: "Elevator 983 PORTER SQUARE - Red Line Platforms to Lobby
unavailable due to maintenance",
            short_header_text: "Elevator 983 PORTER SQUARE - Red Line Platforms to
Lobby unavailable due to maintenance",
            description_text: "",
            severity: "Minor",
            created_dt: "1403548943",
            last_modified_dt: "1403548943",
            service_effect_text: "Elevator unavailable",
            timeframe_text: "",
            alert_lifecycle: "New",
            effect_periods: [{
                effect_start: "1403559660",
                effect_end: "1403568658"
            }],
            affected_services: {
                services: [],
                elevators: [{
                    elev_id: "983",
                    elev_name: "PORTER SQUARE - Red Line Platforms to Lobby",
                    elev_type: "Elevator",
                    stops: [{
                        stop_id: "70065",
                        stop_name: "Porter Sq - Inbound",
                        parent_station: "place-portr",
                        parent_station_name: "Porter Square Station"
                    },
                    {
                        stop_id: "70066",
                        stop_name: "Porter Sq - Outbound",
```

```
                                    parent_station: "place-portr",
                                    parent_station_name: "Porter Square Station"
                                }]
                            }]
                        }
                }]
}
```

4.5.2 ALERTSBYROUTE

This query will return alerts affecting a particular route.

Special Parameters

Name	Description
route	GTFS-compatible route_id value for which alerts should be returned Data type: String Example: “931_”
include_access_alerts (optional)	Whether or not alerts pertaining to accessibility (elevators, escalators) should be returned Data type: String representation of Boolean Possible values: “true” or “false”; default value: “false” If not included, then alerts pertaining to accessibility are not returned
include_service_alerts (optional)	Whether or not service alerts should be returned Data type: String representation of Boolean Possible values: “true” or “false”; default value: “true” If not included, then service alerts will be returned

Response Fields

Name	Description	Can be absent	Can be empty
alerts	Root object of the feed	No	No
route_id	Property of “alerts.” String. The unique GTFS-compatible identifier for the route for which alerts are returned Example: “931_”	No	No
route_name	Property of “alerts.” String. The human-readable name for the route for which alerts are returned Example: “Red Line”	No	No
alert	Child object “alerts.” Contains information about a single alert	No	No
alert_id	Property of “alert.” String. The unique identifier for the alert Example: “781”	No	No

All other fields are similar to that for Alerts (see Section 4.4).

Notes

Similar to Alerts (see Section 4.4)

- The route_id and route_name properties of alerts are currently not displayed. They will be displayed as properties of alerts in a future update.

Example

http://realtime.mbta.com/developer/api/v2/alertsbyroute?api_key=wX9NwuHnZU2ToO7GmGR9uw&route=931_&include_access_alerts=true&include_service_alerts=true&format=json


```
{
	alerts: [{
		alert_id: 33257,
		effect_name: "Delay",
		effect: "OTHER_EFFECT",
		cause_name: "disabled train",
		cause: "TECHNICAL_PROBLEM",
		header_text: "Red Line experiencing minor southbound delays due to
disabled train",
		short_header_text: "Red Line experiencing minor southbound delays due to
disabled train",
		description_text: "Affected stops: Harvard Station - Inbound Central Sq -
Inbound",
		severity: "Minor",
		created_dt: "1403548918",
		last_modified_dt: "1403548918",
		service_effect_text: "Minor Red Line delay",
		timeframe_text: "",
		alert_lifecycle: "New",
		effect_periods: [{
			effect_start: "1403548917",
			effect_end: "1403568658"
		}],
		affected_services: {
			services: [{
				route_type: "1",
				mode_name: "Subway",
				route_id: "931_",
				route_name: "Red Line",
				direction_id: "0",
				direction_name: "Southbound",
				stop_id: "70067",
				stop_name: "Harvard Station - Inbound"
			},
			{
				route_type: "1",
				mode_name: "Subway",
				route_id: "931_",
				route_name: "Red Line",
				direction_id: "0",
				direction_name: "Southbound",
				stop_id: "70069",
				stop_name: "Central Sq - Inbound"
			},
			{
				route_type: "1",
				mode_name: "Subway",
				route_id: "933_",
				route_name: "Red Line",
				direction_id: "0",
				direction_name: "Southbound",
				stop_id: "70067",
				stop_name: "Harvard Station - Inbound"
```

```
                        },
                        {
                               route_type: "1",
                               mode_name: "Subway",
                               route_id: "933_",
                               route_name: "Red Line",
                               direction_id: "0",
                               direction_name: "Southbound",
                               stop_id: "70069",
                               stop_name: "Central Sq - Inbound"
                        }],
                        elevators: []
                }
        },
        {
                alert_id: 33258,
                effect_name: "Accessibility",
                effect: "OTHER_EFFECT",
                cause_name: "maintenance",
                cause: "MAINTENANCE",
                header_text: "Elevator 983 PORTER SQUARE - Red Line Platforms to Lobby
unavailable due to maintenance",
                short_header_text: "Elevator 983 PORTER SQUARE - Red Line Platforms to
Lobby unavailable due to maintenance",
                description_text: "",
                severity: "Minor",
                created_dt: "1403548943",
                last_modified_dt: "1403548943",
                service_effect_text: "Elevator unavailable",
                timeframe_text: "",
                alert_lifecycle: "New",
                effect_periods: [{
                        effect_start: "1403559660",
                        effect_end: "1403568658"
                }],
                affected_services: {
                        services: [],
                        elevators: [{
                               elev_id: "983",
                               elev_name: "PORTER SQUARE - Red Line Platforms to Lobby",
                               elev_type: "Elevator",
                               stops: [{
                                      stop_id: "70065",
                                      stop_name: "Porter Sq - Inbound",
                                      parent_station: "place-portr",
                                      parent_station_name: "Porter Square Station"
                               },
                               {
                                      stop_id: "70066",
                                      stop_name: "Porter Sq - Outbound",
                                      parent_station: "place-portr",
                                      parent_station_name: "Porter Square Station"
                               }]
                        }]
                }
        }],
        route_id: "931_",
        route_name: "Red Line"
}
```


4.5.3 ALERTSBYSTOP

This query will return alerts affecting a particular stop.

Special Parameters

Name	Description
stop	GTFS-compatible stop_id value for which alerts should be returned Data type: String Example: “place-portr”
include_access_alerts (optional)	Whether or not alerts pertaining to accessibility (elevators, escalators) should be returned Data type: String representation of Boolean Possible values: “true” or “false”; default value: “false” If not included, then alerts pertaining to accessibility are not returned
include_service_alerts (optional)	Whether or not service alerts should be returned Data type: String representation of Boolean Possible values: “true” or “false”; default value: “true” If not included, then service alerts will be returned

Response Fields

Name	Description
alerts	Root object of the feed
stop_id	Property of the root object The GTFS-compatible unique identifier for the stop for which alerts are returned Data type: String Example: “place-portr”
stop_name	Property of the root object The GTFS-compatible name for the stop for which alerts are returned Data type: String Example: “Porter Square Station”
alert	Child object of the root object Contains information about a single alert
alert_id	Property of the “alert” object The unique identifier for the alert Data type: Integer Example: “781”

All other fields are similar to that for Alerts (see Section 4.4).

Notes

Similar to Alerts (see Section 4.4)

- The stop_id and stop_name properties of alerts are currently not displayed. They will be displayed as properties of alerts in a future update.

Example

http://realtime.mbta.com/developer/api/v2/alertsbystop?api_key=wX9NwuHnZU2ToO7GmGR9uw&stop=Porter%20Square&include_access_alerts=true&include_service_alerts=true&format=json

```
{
       alerts: [{
              alert_id: 33258,
              effect_name: "Accessibility",
              effect: "OTHER_EFFECT",
              cause_name: "maintenance",
              cause: "MAINTENANCE",
              header_text: "Elevator 983 PORTER SQUARE - Red Line Platforms to Lobby
unavailable due to maintenance",
              short_header_text: "Elevator 983 PORTER SQUARE - Red Line Platforms to
Lobby unavailable due to maintenance",
              description_text: "",
              severity: "Minor",
              created_dt: "1403548943",
              last_modified_dt: "1403548943",
              service_effect_text: "Elevator unavailable",
              timeframe_text: "",
              alert_lifecycle: "New",
              effect_periods: [{
                     effect_start: "1403559660",
                     effect_end: "1403568658"
              }],
              affected_services: {
                     services: [],
                     elevators: [{
                            elev_id: "983",
                            elev_name: "PORTER SQUARE - Red Line Platforms to Lobby",
                            elev_type: "Elevator",
                            stops: [{
                                   stop_id: "70065",
                                   stop_name: "Porter Sq - Inbound",
                                   parent_station: "place-portr",
                                   parent_station_name: "Porter Square Station"
                            },
                            {
                                   stop_id: "70066",
                                   stop_name: "Porter Sq - Outbound",
                                   parent_station: "place-portr",
                                   parent_station_name: "Porter Square Station"
                            }]
                     }]
              }
       }],
       stop_id: "Porter Square",
       stop_name: "Porter Square"
}
```


4.5.4 ALERTBYID

This query will return details for a particular alert.

Special Parameters

Name	Description
id	Unique identifier for the alert Date Type: Integer Example: “781”

include_access_alerts (optional)	Whether or not alerts pertaining to accessibility (elevators, escalators) should be returned Data type: String representation of Boolean Possible values: “true” or “false”; default value: “false” If not included, then alerts pertaining to accessibility are not returned
include_service_alerts (optional)	Whether or not service alerts should be returned Data type: String representation of Boolean Possible values: “true” or “false”; default value: “true” If not included, then service alerts will be returned

Response Fields

Name	Description	Can be absent	Can be empty
alert	Root object of the feed	No	No
alert_id	Property of “alert.” Integer. The unique identifier for the alert Example: “2585”	No	No

All other fields are similar to that for Alerts (see Section 4.4).

Notes

Similar to Alerts (see Section 4.4).

It is good practice to always include the parameter &include_service_alerts=true; you will need it when requesting an access alert, and it will have no effect when you are requesting a service alert.

Example

http://realtime.mbta.com/developer/api/v2/alertbyid?api_key=wX9NwuHnZU2ToO7GmGR9uw&id=33274&format=jsonhttp://54.81.189.97/developer/api/v2/alertbyid?api_key=wX9NwuHnZU2ToO7GmGR9uw&include_access=true&include_service=true&id=33274&format=json

JSON Response:


```
{
       alert_id: 33257,
       effect_name: "Delay",
       effect: "OTHER_EFFECT",
       cause_name: "disabled train",
       cause: "TECHNICAL_PROBLEM",
       header_text: "Red Line experiencing minor southbound delays due to disabled
train",
       short_header_text: "Red Line experiencing minor southbound delays due to
disabled train",
       description_text: "Affected stops: Harvard Station - Inbound Central Sq -
Inbound",
       severity: "Minor",
       created_dt: "1403548918",
       last_modified_dt: "1403548918",
       service_effect_text: "Minor Red Line delay",
       timeframe_text: "",
       alert_lifecycle: "New",
       effect_periods: [{
              effect_start: "1403548917",
              effect_end: "1403568658"
       }],
```

```
      affected_services: {
            services: [{
                  route_type: "1",
                  mode_name: "Subway",
                  route_id: "931_",
                  route_name: "Red Line",
                  direction_id: "0",
                  direction_name: "Southbound",
                  stop_id: "70067",
                  stop_name: "Harvard Station - Inbound"
            },
            {
                  route_type: "1",
                  mode_name: "Subway",
                  route_id: "931_",
                  route_name: "Red Line",
                  direction_id: "0",
                  direction_name: "Southbound",
                  stop_id: "70069",
                  stop_name: "Central Sq - Inbound"
            },
            {
                  route_type: "1",
                  mode_name: "Subway",
                  route_id: "933_",
                  route_name: "Red Line",
                  direction_id: "0",
                  direction_name: "Southbound",
                  stop_id: "70067",
                  stop_name: "Harvard Station - Inbound"
            },
            {
                  route_type: "1",
                  mode_name: "Subway",
                  route_id: "933_",
                  route_name: "Red Line",
                  direction_id: "0",
                  direction_name: "Southbound",
                  stop_id: "70069",
                  stop_name: "Central Sq - Inbound"
            }],
            elevators: []
      }
}
```


4.5.5 ALERTSHEADERS

This query will return headers for all alerts.

Special Parameters

Name	Description
include_access_alerts (optional)	Whether or not alerts pertaining to accessibility (elevators, escalators) should be returned Data type: String representation of Boolean Possible values: “true” or “false”; default value: “false” If not included, then alerts pertaining to accessibility are not returned

include_service_alerts (optional)	Whether or not service alerts should be returned Data type: String representation of Boolean Possible values: “true” or “false”; default value: “true” If not included, then service alerts will be returned

Response Fields

Name	Description
alert_headers	Root object of the feed
alert	Child object of “alert_headers.” Contains information about a single alert
alert_id	Property of “alert.” Integer. The unique identifier for the alert Example: “781”
header_text	Property of “alert.” String. 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”

Example

http://realtime.mbta.com/developer/api/v2/alertheaders?api_key=wX9NwuHnZU2ToO7GmGR9uw&include_access_alerts=true&include_service_alerts=true&format=json


```
{
       alert_headers: [{
              alert_id: 33257,
              header_text: "Red Line experiencing minor southbound delays due to
disabled train"
       },
       {
              alert_id: 33258,
              header_text: "Elevator 983 PORTER SQUARE - Red Line Platforms to Lobby
unavailable due to maintenance"
       }]
}
```


4.5.6 ALERTHEADERSBYROUTE

This query will return headers for alerts affecting a particular route.

Special Parameters

Name	Description
route	GTFS-compatible route_id value for which alert headers should be returned Data type: String Example: “931_”

include_access_alerts (optional)	Whether or not alerts pertaining to accessibility (elevators, escalators) should be returned Data type: String representation of Boolean Possible values: “true” or “false” Default value: “false” If not included, then alerts pertaining to accessibility are not returned
include_service_alerts (optional)	Whether or not service alerts should be returned Data type: String representation of Boolean Possible values: “true” or “false” Default value: “true” If not included, then service alerts will be returned

Response Fields

Name	Description
alert_headers	Root object of the feed
route_id	Property of alert_headers. The unique GTFS-compatible identifier for the route for which alert headers are returned Data type: String Example: “931_”
route_name	Property of alert_headers. String. The human-readable name for the route for which alert headers are returned Example: “Red Line”
alert	Child object of the root object Contains information about a single alert
alert_id	Property of the “alert” object. Integer. The unique identifier for the alert Example: “781”
header_text	Property of the “alert” object. String. 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”

Notes

Similar to Alerts (see Section 4.4)

- The route_id and route_name properties of alerts are currently not displayed. They will be displayed as properties of alerts in a future update.

Example

http://realtime.mbta.com/developer/api/v2/alertheadersbyroute?api_key=wX9NwuHnZU2ToO7GmGR9uw&route=931_&include_access_alerts=true&include_service_alerts=true&format=json


```
{
        alert_headers: [{
```

```
            alert_id: 33257,
            header_text: "Red Line experiencing minor southbound delays due to
disabled train"
      },
      {
            alert_id: 33258,
            header_text: "Elevator 983 PORTER SQUARE - Red Line Platforms to Lobby
unavailable due to maintenance"
      }],
      route_id: "931_",
      route_name: "Red Line"
}
```


4.5.7 ALERTHEADERSBYSTOP

This query will return headers for alerts affecting a particular stop.

Special Parameters

Name	Description
stop	GTFS-compatible stop_id value for which alert headers should be returned Data type: String Example: “place-portr”
include_access_alerts (optional)	Whether or not alerts pertaining to accessibility (elevators, escalators) should be returned Data type: String representation of Boolean Possible values: “true” or “false”; default value: “false” If not included, then alerts pertaining to accessibility are not returned
include_service_alerts (optional)	Whether or not service alerts should be returned Data type: String representation of Boolean Possible values: “true” or “false”; default value: “true” If not included, then service alerts will be returned

Response Fields

Name	Description
alert_headers	Root object of the feed
stop_id	Property of “alert_headers.” String. The GTFS-compatible unique identifier for the stop for which alert headers are returned Example: “place-portr”
stop_name	Property of “alert_headers.” String. The GTFS-compatible name for the stop for which alert headers are returned Example: “Porter Square Station”
alert	Child object of “alert_headers.” Contains information about a single alert
alert_id	Property “alert.” Integer. The unique identifier for the alert Example: “781”

Name	Description
header_text	Property of “alert.” String. 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”

Notes

Similar to Alerts (see Section 4.4)

- The stop_id and stop_name properties of alert_headers are currently not displayed. They will be displayed as properties of alert_headers in a future update.

Example

http://realtime.mbta.com/developer/api/v2/alertheadersbystop?api_key=wX9NwuHnZU2ToO7GmGR9uw&stop=Porter%20Square&include_access_alerts=true&include_service_alerts=true&format=json


```
{
       alert_headers: [{
              alert_id: 33258,
              header_text: "Elevator 983 PORTER SQUARE - Red Line Platforms to Lobby unavailable due to maintenance"
       }],
       stop_id: "Porter Square",
       stop_name: "Porter Square"
}
```


4.6 Other Queries

4.6.1 SERVERTIME

This query will return the current server time.

Special Parameters

None.

Response Fields

Name	Description
server_time	Root object of the response
server_dt	Property of server_time. String representation of an Integer. Server time, in epoch time Example: “1361996667”

Example

http://realtime.mbta.com/developer/api/v2/servertime?api_key=wX9NwuHnZU2ToO7GmGR9uw&format=json

JSON Response:

```
{
	"server_dt": "1361996838"
}
```

4.7 Errors

The following error messages may be returned:

4.7.1 INVALID QUERY

This error occurs when the query string is incorrectly formatted.

Example

```
http://realtime.mbta.com/developer/api/v2/routesstop?api_key=wX9NwuHnZU2ToO7GmGR9uw&st
op=place-portr
```

```
<error xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
	<message>The requested resource is not found</message>
</error>
```

4.7.2 INVALID API KEY

This error occurs when an invalid API key is used.

Example

```
http://realtime.mbta.com/developer/api/v2/routesbystop?api_key=1234567890&stop=place-
portr
```

```
<error xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
	<message>Invalid API Key</message>
</error>
```

4.7.3 MISSING REQUIRED QUERY PARAMETER

This error occurs when a required parameter is not provided.

Example

```
http://realtime.mbta.com/developer/api/v2/routesbystop?api_key=wX9NwuHnZU2ToO7GmGR9uw
```

```
<error xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
	<message>Missing required query parameter: stop</message>
</error>
```

4.7.4 INVALID QUERY PARAMETER

This error occurs when an invalid query parameter is provided.

Example

```
http://realtime.mbta.com/developer/api/v2/routesbystop?api_key=wX9NwuHnZU2ToO7GmGR9uw&
stop=place-portr&id=1
```

```
<error xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
       <message>Invalid query parameter: id</message>
</error>
```

4.7.5 DATA NOT AVAILABLE

This error occurs when data is not available, or when an incorrect parameter is used.

Example

```
http://realtime.mbta.com/developer/api/v2/routesbystop?api_key=wX9NwuHnZU2ToO7GmGR9uw&
stop=place-port
```

```
<error xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
       <message>Routes data is not available for stop place-port</message>
</error>
```

4.7.6 DATA USAGE LIMIT EXCEEDED

This error occurs when the usage limit for a particular API key has been exceeded.

```
<error xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
       <message>Invalid API Key</message>
</error>
```

Note: The error message is anticipated to be changed to “Data usage limit exceeded” in June 2013.

4.7.7 INSUFFICIENT PRIORITY

This error occurs when the API key does not have sufficient priority (this is possible only when available bandwidth is insufficient to handle all user requests and requests are being prioritized).

Note: not currently implemented.

5. ABOUT THIS DOCUMENT

5.1 Version History

Version #	Date	Change Author	Description of Change
2.0	2014/08/04	Dave Barker	Updated for API v2, and reorganized into separate docs for API, GTFS, GTFS-realtime, and RSS.
2.0.1	2014/09/08	Dave Barker	Added new parameters to schedule calls.