

# chicago transit authority

## Bus Tracker API documentation

### Introduction

The Bus Tracker API allows for querying data from the CTA Bus Tracker service, with an XML document as output. The system can provide estimated arrival times for buses as they approach bus stops, as well as route, service and vehicle location data. Information about buses and their estimated arrivals is updated about once every minute.

Note that in using this API, you must agree to our [License Agreement and Terms of Use](#).

### Table of Contents

<b>Introduction</b> .....	<b>1</b>
<b>Overview</b> .....	<b>2</b>
What is the Bus Tracker API? .....	2
What data is available through the API?.....	2
Will my application break if changes are made to the API?.....	2
How does the Developer API work? .....	2
Is there a limit to the number of requests I can make to the Developer API?.....	3
<b>Web Service</b> .....	<b>3</b>
<b>Reference</b> .....	<b>4</b>
Calling Version 2.....	5
JSON support using the “format=json” parameter in the requesting URL .....	5
getpredictions includes a delay and countdown field for all results.....	5
Time .....	6
Vehicles.....	8
Routes.....	12
Route Directions .....	15
Stops.....	17
Patterns.....	20
Predictions.....	24
Error codes.....	32
<b>Developer License Agreement and Terms of Use</b> .....	<b>34</b>
<b>Acknowledgements</b> .....	<b>40</b>

## Overview

### What this document covers

This document explains how to request information and what information is provided through the two alerts-related APIs.

### What is the Bus Tracker API?

The Bus Tracker API allows developers to request and retrieve data directly from BusTime (the system which produces estimated arrival times and which provides location and route information in real-time).

Developers can make HTTP requests for data and receive XML responses from the Bus Tracker web server to build applications that help transit riders.

### What data is available through the API?

Data available through the API includes:

- Vehicle locations
- Route data (route lists, stop lists geo-positional route definitions, etc.)
- Prediction Data
- Service Bulletins

### Will my application break if changes are made to the API?

No. The API is fully backward compatible, allowing time for developers to upgrade their applications to make use of new API features.

### How does the Developer API work?

The developer API uses the same data from the BusTime system, which powers CTA Bus Tracker. Information about the location, direction and status of CTA buses is fed from each bus and delivered to the BusTime system, which then can show where buses are or estimate arrival times to stops ahead of a bus.

Data is updated about once per minute, and arrival estimations are based on how long it normally takes for a bus to get from one place to the next. Because traffic conditions and other unexpected delays occur, we can't predict precisely when a bus will arrive—only estimate based on normal travel times during the time of day where an estimate is occurring.

In order to use the API, the user must sign in to their CTA Bus Tracker account and then request an API key. Only one key will be available per account. Once the request has been approved, the user will be sent an e-mail will be sent to the user containing the API key.

The key allows the user to make calls to the API as it is included in every data request.

## Is there a limit to the number of requests I can make to the Developer API?

Yes. By default, one API key can make a maximum of 10,000 requests. If you believe that you will require more than 10,000 daily requests, you must request that the cap on your key be raised to handle the additional traffic.

## Why we're providing this API

The hope is that, in places where travel information is either desired or deemed useful, that information about the state of CTA services can help people make empowered decisions about their trip. By providing an API into this information already published on our Web site, we hope to see applications of this information included in mobile device applications, public displays, Web sites and more.

## Legal notice

By using this API, you agree to our Developer Terms of Use, which is in the appendix of this document. It's important that you, the developer, understand that this service is provided on an as-is basis and without any guarantees as to availability or accuracy. You must read and agree to the full Developer Terms of Use to use this API.

## Web Service

The Bus Tracker API is a web service that uses HTTP/1.1 as its application protocol. Each type of call/request to the API is an HTTP GET call to a unique URL. Parameters are encoded in the HTTP GET request by following the URL with a "?" and "argument=value" pairs separated by "&".

The response is a well-formed XML or JSON document with a Content-Type of "text/xml".

For example, to request the current system time through the developer API, a program or script will make a HTTP/1.1 GET request to the following URL with parameters:

**<http://www.ctabustracker.com/bustime/api/v2/gettime?key=89dj2he89d8j3j3ksjhdue93j>**

The **www.ctabustracker.com** is the host on which the Developer API is servicing HTTP requests. The service is running on port 80.

The version of the API that is being accessed is built into the URL. In the above example, "v1" represents version 1.0 of the API.

The "key" parameter represents the API key assigned to the developer making the request. All requests to the API must be accompanied by a valid API key.

## Reference

This section describes all requests that can be made to the Bus Tracker API. The description of each request includes a complete set of possible arguments along with the response XML and schema.

## Definitions

- **Delayed Vehicle** – The state entered by a vehicle when it has been determined to be stationary for more than a pre-defined time period.
- **Direction** – Common direction of travel of a route.
- **Off-route Vehicle** – State entered by a transit vehicle when it has strayed from its scheduled pattern.
- **Pattern** – A unique sequence of geo-positional points (waypoints and stops) that combine to form the path that a transit vehicle will repetitively travel. A route often has more than one possible pattern.
- **Route** – One or more set of patterns that together form a single service.
- **Service Bulletin** – Text-based announcements affecting a set of one or more services (route, stops, etc.).
- **Stop** – Location where a transit vehicle can pick-up or drop-off passengers. Predictions are only generated at stops.
- **Waypoint** – A geo-positional point in a pattern used to define the travel path of a transit vehicle.

## Common Parameters

All request URLs have these parameters in common:

Name	Supported Versions	Required?	Example	Description
<b>version</b>	All	Yes	/v2/	The version of the API being used. Legal values are v1 and v2.
<b>locale</b>	All	No	locale=en	The language that the response should be in. See "List Supported Languages" for more details on how to use this field.
<b>format</b>	v2	No	format=json	The format of the response. Legal values are "xml" and "json". XML is the default if no format is requested.

## Version 2 Release Notes

Version 2 of the Developer API contains a number of changes:

- The URL of the request changes.
- JSON responses are supported.
- **Getpredictions** results always include delay and countdown fields

### Calling Version 2

Version 2 of the API is used by including "v2" in the request URL as follows:

<http://ctabustracker.com/bustime/api/v2/gettime?key=89dj2he89d8j3j3ksjhdue93j>

### JSON support using the "format=json" parameter in the requesting URL

XML is still the default response. If the format=xml, or no format is requested, then the response should match an older, version 1, XML response, with the exceptions of the new fields in **getpredictions**.

### getpredictions includes a delay and countdown field for all results

The **dly** field in the **getpredictions** response is either "true" or "false". A "true" value means that the bus has been delayed and its arrival time is not known. This used to be included only when the value was "true", but in version 2 it is always included.

The **prcdtn** field in the **getpredictions** response is a countdown, in minutes, until the bus arrives. When it reaches '0', then the arrival is imminent.

## Time

**Base URL:** <http://www.ctabustracker.com/bustime/api/v2/gettime>

### Parameters:

Name	Value	Description
Key	string (required)	25-digit Bus Tracker API access key.

### Response

A well-formed XML or JSON document containing the current system time will be returned as a response to **gettime**.

### Response Fields:

Name	Description
<b>bustime-response</b>	Root element of the response document.
<b>Error</b>	Child element of the root element. Contains a message if the processing of the request resulted in an error.
<b>Tm</b>	Child element of the root element containing the current system date and (local) time. Date and time is represented in the following format: YYYYMMDD HH:MM:SS. Month is represented as two digits where January is equal to "01" and December is equal to "12". Time is represented using a 24-hour clock.

### Remarks

Use the **gettime** request to retrieve the current system date and time. Since BusTime is a time-dependent system, it is important to synchronize your application with BusTime's system date and time.

The time specified is the local time.

### Schema

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="bustime-response" type=" bustime-response" />
  <xs:complexType name="bustime-response">
    <xs:sequence>
      <xs:element name="error" type="error" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="tm" type="xs:string" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="error">
    <xs:sequence>
      <xs:element name="msg" type="xs:string" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

## **Example**

Request:

<http://www.ctabustracker.com/bustime/api/v2/gettime?key=89dj2he89d8j3j3ksjhdue93j>

Response:

```
<?xml version="1.0"?>
<bustime-response>
  <tm>20090611 14:42:32</tm>
</bustime-response>
```

**Request:**

<http://ctabustracker.com/bustime/api/v2/gettime?key=89dj2he89d8j3j3ksjhdue93j&format=json>

**Response:**

```
{
  "bustime-response": {
    "tm": "20130104 14:11:54"
  }
}
```

## Vehicles

**Base URL:** <http://www.ctabustracker.com/bustime/api/v2/getvehicles>

### Parameters:

Name	Value	Description
<b>Key</b>	string (required)	25-digit Bus Tracker API access key.
<b>Vid</b>	comma-delimited list of vehicle IDs (not available with <b>rt</b> parameter)	Set of one or more vehicle IDs whose location should be returned. For example: 509,392,201,4367 will return information for four vehicles (if available). A maximum of 10 identifiers can be specified.
<b>Rt</b>	comma-delimited list of route designators (not available with <b>vid</b> parameter)	Set of one or more route designators for which matching vehicles should be returned. For example: X3,4,20 will return information for all vehicles currently running those three routes (if available). A maximum of 10 identifiers can be specified.
<b>tmres</b>	string (optional)	<p>Resolution of time stamps. Set to "s" to get time resolution to the second. Set to "m" to get time resolution to the minute. If omitted, defaults to "m".</p> <p>Date and time is represented in the following format:</p> <p>If specified as "s"</p> <p><i>YYYYMMDD HH:MM:SS</i></p> <p>If specified as "m"</p> <p><i>YYYYMMDD HH:MM</i></p> <p>Month is represented as two digits where January is equal to "01" and December is equal to "12". Time is represented using a 24-hour clock.</p>

### Response

A well-formed XML or JSON document will be returned as a response to **getvehicles**. The response will include the most-recent status for each vehicle.

### Response Fields:

Name	Description
<b>bustime-response</b>	Root element of the response document.
<b>error</b>	Child element of the root element. Message if the processing of the request resulted in an error.
<b>vehicle</b>	Child element of the root element. Encapsulates all information available for a single vehicle in the response.
<b>vid</b>	Child element of the <b>vehicle</b> element. Alphanumeric string representing the vehicle ID (ie. bus number)
<b>tmstamp</b>	Child element of the <b>vehicle</b> element. Date and local time of the last positional update of the vehicle. Date and time is represented in the following format: YYYYMMDD HH:MM. Month is represented as two digits where January is equal to "01" and December is equal to "12". Time is represented using a 24-hour clock.
<b>lat</b>	Child element of the <b>vehicle</b> element. Latitude position of the vehicle in decimal degrees (WGS 84).
<b>lon</b>	Child element of the <b>vehicle</b> element. Longitude position of the vehicle in decimal degrees (WGS 84).
<b>hdg</b>	Child element of the <b>vehicle</b> element. Heading of vehicle as a 360° value, where 0° is North, 90° is East, 180° is South and 270° is West.
<b>pid</b>	Child element of the <b>vehicle</b> element. Pattern ID of trip currently being executed.
<b>pdist</b>	Child element of the <b>vehicle</b> element. Linear distance in feet that the vehicle has traveled into the pattern currently being executed.
<b>rt</b>	Child element of the <b>vehicle</b> element. Route that is currently being executed by the vehicle (ex. "20").
<b>des</b>	Child element of the <b>vehicle</b> element. Destination of the trip being executed by the vehicle (ex. "Austin").
<b>dly</b>	Child element of the <b>vehicle</b> element. The value is "true" if the vehicle is delayed. The <b>dly</b> element is only present if the vehicle is delayed.
<b>tablockid</b>	<i>Child element of the <b>vehicle</b> element. TA's version of the scheduled block identifier for the work currently being performed by the vehicle.</i>
<b>tatripid</b>	<i>Child element of the <b>vehicle</b> element. TA's version of the scheduled trip identifier for the vehicle's current trip.</i>
<b>zone</b>	<i>Child element of the <b>prd</b> element. The zone name if the vehicle has entered a defined zone, otherwise blank.</i>

## Remarks

Use the **getvehicles** request to retrieve vehicle information (i.e., locations) of all or a subset of vehicles currently being tracked by BusTime.

Use the **vid** parameter to retrieve information for one or more vehicles currently being tracked.

Use the **rt** parameter to retrieve information for vehicles currently running one or more of the specified routes.

**Note:** The **vid** and **rt** parameters cannot be combined in one request. If both parameters are specified on a request to **getvehicles**, only the first parameter specified on the request will be processed.

## Schema

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="bustime-response" type="bustime-response"/>
  <xs:complexType name="bustime-response">
    <xs:sequence>
<xs:element name="error" type="error" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="vehicle" type="vehicle" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="error">
<xs:sequence>
<xs:element name="vid" type="xs:string" minOccurs="0" maxOccurs="1"/>
<xs:element name="rt" type="xs:string" minOccurs="0" maxOccurs="1"/>
<xs:element name="msg" type="xs:string" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="vehicle">
    <xs:sequence>
<xs:element name="vid" type="xs:string" minOccurs="1" maxOccurs="1"/>
<xs:element name="tmpstmp" type="xs:string" minOccurs="1" maxOccurs="1"/>
<xs:element name="lat" type="xs:double" minOccurs="1" maxOccurs="1"/>
<xs:element name="lon" type="xs:double" minOccurs="1" maxOccurs="1"/>
      <xs:element name="hdg" type="xs:int" minOccurs="1" maxOccurs="1"/>
      <xs:element name="pid" type="xs:int" minOccurs="1" maxOccurs="1"/>
<xs:element name="pdist" type="xs:int" minOccurs="1" maxOccurs="1"/>
<xs:element name="rt" type="xs:string" minOccurs="1" maxOccurs="1"/>
<xs:element name="des" type="xs:string" minOccurs="1" maxOccurs="1"/>
<xs:element name="dly" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
<xs:element name="tablockid" type="xs:string" minOccurs="1" maxOccurs="1"/>
<xs:element name="tatripid" type="xs:string" minOccurs="1" maxOccurs="1"/>
<xs:element name="zone" type="xs:string" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

## Example

Request:

<http://www.ctabustracker.com/bustime/api/v2/getvehicles?key=89dj2he89d8j3j3ksjhdue93j&vid=509.392>

Response:

```
<?xml version="1.0"?>
<bustime-response>
  <vehicle>
    <vid>509</vid>
    <tmstmp>20090611 10:28</tmstmp>
    <lat>41.92124938964844</lat>
    <lon>-87.64849853515625</lon>
    <hdg>358</hdg>
    <pid>3630</pid>
    <pdist>5678</pdist>
    <rt>8</rt>
    <des>Waveland/Broadway</des>
    <tablockid>2 -701</tablockid>
    <tatripid>108</tatripid>
    <zone>Bay 1</zone>
  </vehicle>
```

```

<vehicle>
  <vid>509</vid>
  <tmstamp>20090611 10:28</tmstamp>
  <lat>41.92124938964844</lat>
  <lon>-87.64849853515625</lon>
  <hdg>358</hdg>
  <pid>3630</pid>
  <pdist>5678</pdist>
  <rt>8</rt>
  <des>Waveland/Broadway</des>
  <tablockid>2 -701</tablockid>
  <tatripid>108</tatripid>
  <zone>Bay 1</zone>
</vehicle>
</bustime-response>

```

**Example:**

The JSON document below is a response to the following request:

**Request:**

<http://ctabustracker.com/bustime/api/v2/getvehicles?key=89dj2he89d8j3j3ksjhdue93j&vid=6438,1295&tmres=s&format=json>

**Response:**

```

{"bustime-response": {"vehicle": [
  {
    "des": "Michigan/Chicago",
    "lon": "-87.61421203613281",
    "tablockid": "3 -712",
    "tatripid": "105797",
    "hdg": "177",
    "rt": "3",
    "pid": 5342,
    "tmstamp": "20120620 13:02:38",
    "vid": "6438",
    "lat": "41.72489577073317",
    "pdist": 2950,
    "zone": ""
  },
  {
    "des": "Michigan/Chicago",
    "lon": "-87.61560402664483",
    "tablockid": "3 -714",
    "tatripid": "105801",
    "hdg": "179",
    "rt": "3",
    "pid": 5342,
    "tmstamp": "20120620 13:02:52",
    "vid": "1295",
    "lat": "41.779821508071",
    "zone": "Bay 1"
  }
]}

```

## Routes

**Base URL:** <http://www.ctabustracker.com/bustime/api/v2/getroutes>

### Parameters:

Name	Value	Description
Key	string (required)	25-digit Bus Tracker API access key.

### Response

A well-formed XML or JSON document will be returned as a response to **getroutes**.

### Response Fields:

Name	Description
<b>bustime-response</b>	Root element of the response document.
<b>error</b>	Child element of the root element. Message if the processing of the request resulted in an error.
<b>route</b>	Child element of the root element. Encapsulates a route serviced by the system.
<b>rt</b>	Child element of the <b>route</b> element. Alphanumeric designator of a route (ex. "20" or "X20").
<b>rtnm</b>	Child element of the <b>route</b> element. Common name of the route (ex. "Madison" for the 20 route).

### Remarks

Use the **getroutes** request to retrieve the set of routes serviced by the system.

### Schema

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="bustime-response" type="bustime-response"/>
  <xs:complexType name="bustime-response">
    <xs:sequence>
      <xs:element name="error" type="error" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="route" type="route" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="error">
```

```

    <xs:sequence>
      <xs:element name="msg" type="xs:string" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
<xs:complexType name="route">
  <xs:sequence>
    <xs:element name="rt" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="rtnm" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>

```

## Example

Request:

<http://www.ctabustracker.com/bustime/api/v2/getroutes?key=89dj2he89d8j3j3ksjhdue93j>

Response:

```

<?xml version="1.0"?>
<bustime-response>
  <route>
    <rt>1</rt>
    <rtnm>Indiana/Hyde Park</rtnm>
  </route>
  <route>
    <rt>2</rt>
    <rtnm>Hyde Park Express</rtnm>
  </route>
  <route>
    <rt>3</rt>
    <rtnm>King Drive</rtnm>
  </route>
  <route>
    <rt>X3</rt>
    <rtnm>King Drive Express</rtnm>
  </route>
  .
  .
  .
</bustime-response>

```

### Request

<http://ctabustracker.com/bustime/api/v2/getroutes?key=89dj2he89d8j3j3ksjhdue93j&format=json>

### Response

```

{"bustime-response": {
  "routes": [
    {
      "rt": "1",
      "rtnm": "Bronzeville/Union Station"
      "rtclr": "#ffffff"
    },
    {
      "rt": "2",
      "rtnm": "Hyde Park Express"
      "rtclr": "#dc78af"
    },
    .
    .
    .
  ]
}

```



## Route Directions

**Base URL:** <http://www.ctabustracker.com/bustime/api/v2/getdirections>

### Parameters:

Name	Value	Description
<b>key</b>	string (required)	25-digit Bus Tracker API access key.
<b>rt</b>	Single route designator (required)	Alphanumeric designator of a route (ex. "20" or "X20") for which a list of available directions is to be returned.

### Response

A well-formed XML or JSON document will be returned as a response to **getdirections**.

### Response Fields:

Name	Description
<b>bustime-response</b>	Root element of the response document.
<b>error</b>	Child element of the root element. Message if the processing of the request resulted in an error.
<b>dir</b>	Child element of the root element. Direction that is valid for the specified route designator. For example, "East Bound".

### Remarks

Use the **getdirections** request to retrieve the set of directions serviced by the specified route.

### Schema

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="bustime-response" type="bustime-response"/>
  <xs:complexType name="bustime-response">
    <xs:sequence>
      <xs:element name="error" type="error" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="dir" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="error">
    <xs:sequence>
      <xs:element name="rt" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="msg" type="xs:string" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```
</xs:sequence>
</xs:complexType>
</xs:schema>
```

## **Example**

Request:

<http://www.ctabustracker.com/bustime/api/v2/getdirections?key=89dj2he89d8j3j3ksjhdue93j&rt=20>

Response:

```
<?xml version="1.0"?>
<bustime-response>
  <dir>Eastbound</dir>
  <dir>Westbound</dir>
</bustime-response>
```

## **Request**

<http://ctabustracker.com/bustime/api/v2/getdirections?key=89dj2he89d8j3j3ksjhdue93j&rt=20&format=json>

## **Response**

```
{"bustime-response": {
  "directions": [
    {
      "dir": "Eastbound"
    },
    {
      "dir": "Westbound"
    }
  ]
}}
```

## Stops

**Base URL:** <http://www.ctabustracker.com/bustime/api/v2/getstops>

### Parameters:

Name	Value	Description
<b>key</b>	string (required)	25-digit Bus Tracker API access key.
<b>rt</b>	Single route designator (required)	Alphanumeric designator of the route (ex. "20" or "X20") for which a list of available stops is to be returned.
<b>dir</b>	Single route direction (required)	Direction of the route (ex. "East Bound") for which a list of available stops is to be returned.

### Response

A well-formed XML or JSON document will be returned as a response to **getstops**.

### Response Fields:

Name	Description
<b>bustime-response</b>	Root element of the response document.
<b>error</b>	Child element of the root element. Message if the processing of the request resulted in an error.
<b>stop</b>	Child element of the root element. Encapsulates all descriptive information about a particular stop.
<b>stpid</b>	Child element of the <b>stop</b> element. Unique identifier representing this stop.
<b>stpnm</b>	Child element of the <b>stop</b> element. Display name of this stop (ex. "Madison and Clark")
<b>lat</b>	Child element of the <b>stop</b> element. Latitude position of the stop in decimal degrees (WGS 84).
<b>lon</b>	Child element of the <b>stop</b> element. Longitude position of the stop in decimal degrees (WGS 84).

## Remarks

Use the **getstops** request to retrieve the set of stops for the specified route and direction.

Stop lists are only available for a valid route/direction pair. In other words, a list of all stops that service a particular route (regardless of direction) cannot be requested.

## Schema:

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="bustime-response" type="bustime-response"/>
  <xs:complexType name="bustime-response">
    <xs:sequence>
      <xs:element name="error" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="stop" type="stop" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="error">
    <xs:sequence>
      <xs:element name="rt" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="dir" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="msg" type="xs:string" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="stop">
    <xs:sequence>
      <xs:element name="stpid" type="xs:int" minOccurs="1" maxOccurs="1"/>
      <xs:element name="stpnm" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="lat" type="xs:double" minOccurs="1" maxOccurs="1"/>
      <xs:element name="lon" type="xs:double" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

## Example

Request:

<http://www.ctabustracker.com/bustime/api/v2/getstops?key=89dj2he89d8j3j3ksjhdue93j&rt=20&dir=East%20Bound>

Response:

```
<?xml version="1.0"?>
<bustime-response>
  <stop>
    <stpid>4727</stpid>
    <stpnm>1633 W Madison</stpnm>
    <lat>41.881265</lat>
    <lon>-87.66849</lon>
  </stop>
  <stop>
    <stpid>9604</stpid>
    <stpnm>Austin & Pleasant/Fulton</stpnm>
    <lat>41.885206667</lat>
    <lon>-87.77487333333333</lon>
  </stop>
  <stop>
    <stpid>9605</stpid>
    <stpnm>Austin & Randolph/West End</stpnm>
    <lon>41.88386333333333</lon>
    <lat>-87.77485666666667</lat>
  </stop>
  <stop>
    <stpid>9603</stpid>
    <stpnm>Austin & South Blvd/Corcoran</stpnm>
    <lat>41.886908333</lat>
```

```
<lon>-87.77493667</lon>
</stop>
</bustime-response>
```

**Request**

<http://ctabustracker.com/bustime/api/v2/getstops?key=89dj2he89d8j3j3ksjhdue93j&rt=20&dir=East%20Bound&format=json>

**Response**

```
{ "bustime-response": {
  "stops": [
    {
      "stpid": "1577",
      "stpnm": "1509 S Michigan",
      "lat": 41.861706666665,
      "lon": -87.623969999999
    },
    {
      "stpid": "1564",
      "stpnm": "3000 S Michigan",
      "lat": 41.840606666667,
      "lon": -87.623206666667
    },
    ...
  ]
}
```

## Patterns

**Base URL:** <http://www.ctabustracker.com/bustime/api/v2/getpatterns>

### Parameters:

Name	Value	Description
<b>key</b>	string (required)	25-digit Bus Tracker API access key.
<b>pid</b>	comma-delimited list of pattern IDs (not available with <b>rt</b> parameter)	Set of one or more pattern IDs whose points should be returned. For example: 56,436,1221 will return points from three (3) patterns. A maximum of 10 identifiers can be specified.
<b>rt</b>	Single route designator (not available with <b>pid</b> parameter)	Route designator for which all active patterns should be returned.

### Response

A well-formed XML or JSON document will be returned as a response to **getpatterns**.

### Response Fields:

Name	Description
<b>bustime-response</b>	Root element of the response document.
<b>error</b>	Child element of the root element. Message if the processing of the request resulted in an error.
<b>ptr</b>	Child element of the root element. Encapsulates a set of points which define a pattern.
<b>pid</b>	Child element of the <b>ptr</b> element. ID of pattern.
<b>ln</b>	Child element of the <b>ptr</b> element. Length of the pattern in feet.
<b>rtdir</b>	Child element of the <b>ptr</b> element. Direction that is valid for the specified route designator. For example, "East Bound".
<b>pt</b>	Child element of the <b>ptr</b> element. Child element of the root element. Encapsulates one a set of geo-positional points (including stops) that when connected define a pattern.

<b>seq</b>	Child element of the <b>pt</b> element. Position of this point in the overall sequence of points.
<b>typ</b>	Child element of the <b>pt</b> element. 'S' if the point represents a Stop, 'W' if the point represents a waypoint along the route.
<b>stpid</b>	Child element of the <b>pt</b> element. If the point represents a stop, the unique identifier of the stop.
<b>stpnm</b>	Child element of the <b>pt</b> element. If the point represents a stop, the display name of the stop.
<b>pdist</b>	Child element of the <b>pt</b> element. If the point represents a stop, the linear distance of this point (feet) into the requested pattern.
<b>lat</b>	Child element of the <b>pt</b> element. Latitude position of the point in decimal degrees (WGS 84).
<b>lon</b>	Child element of the <b>pt</b> element. Longitude position of the point in decimal degrees (WGS 84).

## Remarks

Use the **getpatterns** request to retrieve the set of geo-positional points and stops that when connected can be used to construct the geo-positional layout of a pattern (i.e., route variation).

Use **pid** to specify one or more identifiers of patterns whose points are to be returned. A maximum of 10 patterns can be specified.

Use **rt** to specify a route identifier where all active patterns are returned. The set of active patterns returned includes: one or more patterns for the specified route (all patterns that are currently being executed by at least one vehicle on the specified route).

Note: The **pid** and **rt** parameters cannot be combined in one request. If both parameters are specified on a request to **getpatterns**, only the first parameter specified on the request will be processed.

## Schema

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="bustime-response" type="bustime-response"/>
  <xs:complexType name="bustime-response">
    <xs:sequence>
      <xs:element name="error" type="error" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="ptr" type="ptr" minOccurs="0" maxOccurs="10"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="error">
    <xs:sequence>
      <xs:element name="pid" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="rt" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="msg" type="xs:string" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```

<xs:complexType name="ptr">
  <xs:element name="pid" type="xs:int" minOccurs="1" maxOccurs="1"/>
  <xs:element name="ln" type="xs:int" minOccurs="1" maxOccurs="1"/>
  <xs:element name="rtmdir" type="xs:string" minOccurs="1" maxOccurs="1"/>
  <xs:element name="pt" type="pt" minOccurs="1" maxOccurs="unbounded"/>
</xs:complexType>
<xs:complexType name="pt">
  <xs:sequence>
    <xs:element name="seq" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="typ" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="stpid" type="xs:int" minOccurs="0" maxOccurs="1"/>
    <xs:element name="stpnm" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="pdist" type="xs:float" minOccurs="0" maxOccurs="1"/>
    <xs:element name="lat" type="xs:double" minOccurs="1" maxOccurs="1"/>
    <xs:element name="lon" type="xs:double" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>

```

## Example

Request:

<http://www.ctabustracker.com/bustime/api/v2/getpatterns?key=89dj2he89d8j3j3ksjhdue93j&rt=20&pid=954>

Response:

```

<?xml version="1.0"?>
<bustime-response>
<ptr>
<pid>954</pid>
  <ln>35569</ln>
  <rtmdir>East Bound</rtmdir>
  <pt>
    <seq>1</seq>
    <typ>S</typ>
    <stpid>409</stpid>
    <stpnm>Madison & Pulaski</stpnm>
    <lat>41.880641167057</lat>
    <lon>-87.725835442543</lon>
    <pdist>0.0</pdist>
  </pt>
  <pt>
    <seq>2</seq>
    <typ>W</typ>
    <lat>41.880693089146</lat>
    <lon>-87.725765705109</lon>
  </pt>
  <pt>
    <seq>3</seq>
    <typ>W</typ>
    <lat>41.880693089146</lat>
    <lon>-87.725674510002</lon>
    <pdist>97.0</pdist>
  </pt>
  .
  .
  .
  </ptr>
</bustime-response>

```

## Request

<http://ctabustracker.com/bustime/api/v2/getpatterns?key=89dj2he89d8j3j3ksjhdue93j&rt=20&pid=954&format=json>

## Response

```
{
  "bustime-response": {
    "ptr": [
      {
        "pid": 2063,
        "ln": 45245.0,
        "rtDir": "EAST",
        "pt": [
          {
            "seq": 1,
            "lat": 41.880348006551,
            "lon": -87.774017572665,
            "typ": "S",
            "stpid": "386",
            "stpm": "Madison \u0026 Austin Terminal",
            "pdist": 0.0
          },
          {
            "seq": 2,
            "lat": 41.880034074881,
            "lon": -87.774034738541,
            "typ": "W",
            "pdist": 0.0
          },
          {
            "seq": 3,
            "lat": 41.880070021289,
            "lon": -87.771502733231,
            "typ": "S",
            "stpid": "388",
            "stpm": "Madison \u0026 Mayfield",
            "pdist": 704.0
          }
        ]
      }
    ]
  }
}
```

## Predictions

**Base URL:** <http://www.ctabustracker.com/bustime/api/v2/getpredictions>

### Parameters:

Name	Value	Description
<b>key</b>	string (required)	25-digit Bus Tracker API access key.
<b>stpid</b>	comma-delimited list of stop IDs (not available with <b>vid</b> parameter)	Set of one or more stop IDs whose predictions are to be returned. For example: 5029,1392,2019,4367 will return predictions for the four stops. A maximum of 10 identifiers can be specified.
<b>rt</b>	comma-delimited list of route designators (optional, available with <b>stpid</b> parameter)	Set of one or more route designators for which matching predictions are to be returned.
<b>vid</b>	comma-delimited list of vehicle IDs (not available with <b>stpid</b> parameter)	Set of one or more vehicle IDs whose predictions should be returned. For example: 509,392,201,4367 will return predictions for four vehicles. A maximum of 10 identifiers can be specified.
<b>top</b>	number (optional)	Maximum number of predictions to be returned.

### Response

A well-formed XML or JSON document will be returned as a response to **getpredictions**.

### Response Fields:

Name	Description
<b>bustime-response</b>	Root element of the response document.
<b>error</b>	Child element of the root element. Message if the processing of the request resulted in an error.
<b>prd</b>	Child element of the root element. Encapsulates a predicted arrival or departure time for the specified set of stops or vehicles.
<b>tmstamp</b>	Child element of the <b>prd</b> element. Date and time (local) the prediction was

	generated. Date and time is represented in the following format: YYYYMMDD HH:MM. Month is represented as two digits where January is equal to "01" and December is equal to "12". Time is represented using a 24-hour clock.
<b>typ</b>	Child element of the <b>prd</b> element. Type of prediction. 'A' for an arrival prediction (prediction of when the vehicle will arrive at this stop). 'D' for a departure prediction (prediction of when the vehicle will depart this stop, if applicable). Predictions made for first stops of a route or layovers are examples of departure predictions.
<b>stpid</b>	Child element of the <b>prd</b> element. Unique identifier representing the stop for which this prediction was generated.
<b>stpnm</b>	Child element of the <b>prd</b> element. Display name of the stop for which this prediction was generated.
<b>vid</b>	Child element of the <b>prd</b> element. Unique ID of the vehicle for which this prediction was generated.
<b>dstp</b>	Child element of the <b>prd</b> element. Linear distance (feet) left to be traveled by the vehicle before it reaches the stop associated with this prediction.
<b>rt</b>	Child element of the <b>prd</b> element. Alphanumeric designator of the route (ex. "20" or "X20") for which this prediction was generated.
<b>rtdir</b>	Child element of the <b>prd</b> element. Direction of travel of the route associated with this prediction (ex. "East Bound").
<b>des</b>	Child element of the <b>prd</b> element. Final destination of the vehicle associated with this prediction.
<b>prdtm</b>	Child element of the <b>prd</b> element. Predicted date and time (local) of a vehicle's arrival or departure to the stop associated with this prediction. Date and time is represented in the following format: YYYYMMDD HH:MM. Month is represented as two digits where January is equal to "01" and December is equal to "12". Time is represented using a 24-hour clock.
<b>dly</b>	Child element of the <b>prd</b> element. "true" if the vehicle is delayed. The <b>dly</b> element is only present if the vehicle that generated this prediction is delayed.
<b>tablockid</b>	Child element of the <b>prd</b> element. TA's version of the scheduled block identifier for the work currently being performed by the vehicle.
<b>tatripid</b>	Child element of the <b>prd</b> element. TA's version of the scheduled trip identifier for the vehicle's current trip.
<b>prdctdn</b>	Child element of the <b>prd</b> element. This is the time left, in minutes, until the bus arrives at this stop.
<b>zone</b>	Child element of the <b>prd</b> element. The zone name if the vehicle has entered a defined zones, otherwise blank.

## Remarks

Use the **getpredictions** request to retrieve predictions for one or more stops or one or more vehicles. Predictions are always returned in ascending order according to **prdtm**.

Use the **vid** parameter to retrieve predictions for one or more vehicles currently being tracked. A maximum of 10 vehicles can be specified.

Use the **stpid** parameter to retrieve predictions for one or more stops. A maximum of 10 stops can be specified.

Note: The **vid** and **stpid** parameters cannot be combined in one request. If both parameters are specified on a request to **getpredictions**, only the first parameter specified on the request will be processed.

Calls to **getpredictions** without specifying the **vid** or **stpid** parameter is not allowed.

Use the **top** parameter to specify the maximum number of predictions to return. If **top** is not specified, then all predictions matching the specified parameters will be returned.

## Schema

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="bustime-response" type="bustime-response"/>
  <xs:complexType name="bustime-response">
    <xs:sequence>
      <xs:element name="error" type="error" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="prd" type="prediction" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="error">
    <xs:sequence>
      <xs:element name="stpid" type="xs:int" minOccurs="0" maxOccurs="1"/>
      <xs:element name="vid" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="msg" type="xs:string" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="prediction">
    <xs:sequence>
      <xs:element name="tmstmp" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="typ" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="stpid" type="xs:int" minOccurs="1" maxOccurs="1"/>
      <xs:element name="stpnm" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="vid" type="xs:int" minOccurs="1" maxOccurs="1"/>
      <xs:element name="dstp" type="xs:int" minOccurs="1" maxOccurs="1"/>
      <xs:element name="rt" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="rtdir" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="des" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="prdtm" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="dly" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
      <xs:element name="tablockid" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="tatripid" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="zone" type="xs:string" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

## Example

Request:

<http://www.ctabustracker.com/bustime/api/v2/getpredictions?key=89dj2he89d8j3j3ksjhdue93j&rt=20&stpid=456>

## Response:

```
<?xml version="1.0"?>
<bustime-response>
<tm></tm>
  <prd>
    <tmstamp>20090611 14:34</tmstamp>
    <typ>A</typ>
    <stpid>456</stpid>
    <stpnm>Madison & Jefferson</stpnm>
    <vid>2013</vid>
    <dstp>891</dstp>
    <rt>20</rt>
    <rtdir>West Bound</rtdir>
    <rtdst>Austin</rtdst>
    <prdtm>20090611 14:40</prdtm>
    <tablockid>3 -701</tablockid>
    <tatripid>106</tatripid>
    <zone></zone>
  </prd>
  <prd>
    <tmstamp>20090611 14:34</tmstamp>
    <typ>A</typ>
    <stpid>456</stpid>
    <stpnm>Madison & Jefferson</stpnm>
    <vid>6435</vid>
    <dstp>1587</dstp>
    <rt>20</rt>
    <rtdir>West Bound</rtdir>
    <rtdst>Austin</rtdst>
    <prdtm>20090611 14:48</prdtm>
    <tablockid>3 -706</tablockid>
    <tatripid>108</tatripid>
    <zone>Bay 1</zone>
  </prd>
</bustime-response>
```

## Request

<http://ctabustracker.com/bustime/api/v2/getpredictions?key=89dj2he89d8j3j3ksjhdue93j&rt=20&stpid=456&format=json>

## Response

```
{ "bustime-response": {
  "prd": [
    {
      "rid": "7028",
      "triped": "57217175",
      "schdtm": "20130104 15:08",
      "tmstamp": "20130104 15:00",
      "typ": "A",
      "stpnm": "87th Street \u0026 Wentworth",
      "stpid": "9405",
      "vid": "",
      "dstp": 0,
      "rt": "87",
      "rtdir": "Eastbound",
      "des": "91st/Commercial",
      "prdtm": "20130104 15:08",
      "tablockid": "87 -706",
      "tatripid": "1007569",
      "dly": false,
      "prdctdn": "8",
      "zone": ""
    },
  ],
}
```

}  
}  
]

## Service Bulletins

**Note:** In addition to Service Bulletins posted in the Bus Tracker system, CTA also publishes a Customer Alerts API that includes upcoming alerts, numerical severity ranking and both train and bus information. Visit <http://www.transitchicago.com/developers> for more information.

We consider the Service Bulletins API in the Bus Tracker system to be deprecated and strongly encourage you to use the Customer Alerts API.

**Base URL:** <http://www.ctabustracker.com/bustime/api/v2/getservicebulletins>

### Parameters:

Name	Value	Description
<b>Key</b>	string (required)	25-digit Bus Tracker API access key.
<b>Rt</b>	comma-delimited list of route designators (required if <b>stpid</b> not specified)	Alphanumeric designator of the route(s) (ex. "20" or "X20") for which a list of service bulletins is to be returned. If combined with <b>rtdir</b> , only one route can be specified.
<b>Rtdir</b>	Single route direction (optional)	Direction of travel of the route specified in the <b>rt</b> parameter. The <b>rt</b> parameter is required when using the <b>rtdir</b> parameter.
<b>Stpid</b>	Comma-delimited list of stop IDs (required if <b>rt</b> not specified)	Set of one or more stop IDs for which service bulletins are to be returned. For example: 5029,1392,2019,4367 will return predictions for the four stops (if available). If combined with <b>rt</b> and <b>rtdir</b> , only one stop can be specified.

### Response

A well-formed XML or JSON document will be returned as a response to **getservicebulletins**.

### Response Fields:

Name	Description
<b>bustime-response</b>	Root element of the response document.
<b>Error</b>	Child element of the root element. Message if the processing of the request resulted in an error.
<b>Sb</b>	Child element of the root element. Encapsulates all data about a service bulletin.

<b>Nm</b>	Child element of the <b>sb</b> element. Unique name/identifier of the service bulletin.
<b>Sbj</b>	Child element of the <b>sb</b> element. Service bulletin subject. A short title for this service bulletin.
<b>Dtl</b>	Child element of the <b>sb</b> element. Service bulletin detail. Full text of the service bulletin.
<b>Brf</b>	Child element of the <b>sb</b> element. Service bulletin brief. A short text alternative to the service bulletin detail.
<b>Prty</b>	Child element of the <b>sb</b> element. Service bulletin priority. The possible values are "High," "Medium," and "Low".
<b>Srvc</b>	Child element of the <b>sb</b> element. Each <b>srvc</b> element represents one or a combination of route, direction and stop for which this service bulletin is valid. If the <b>srvc</b> element is not present, the service bulletin affects all routes and stops.
<b>Rt</b>	Child element of <b>srvc</b> . Alphanumeric designator of the route (ex. "20" or "X20") for which this service bulletin is in effect.
<b>Rtdir</b>	Child element of <b>srvc</b> . Direction of travel of the route for which this service bulletin is in effect.
<b>Stpid</b>	Child element of <b>srvc</b> . ID of the stop for which this service bulletin is in effect.
<b>Stpnm</b>	Child element of <b>srvc</b> . Name of the stop for which this service bulletin is in effect.

## Remarks

Use the **getservicebulletins** for a list of service bulletins that are in effect for a route(s) (**rt**), route & direction (**rt & rtdir**), route & direction & stop (**rt & rtdir & stpid**), or stop(s) (**stpid**). At a minimum, the **rt** or **stpid** parameter must be specified.

A service bulletin (**sb**) definition without a **srvc** element indicates a "system-wide" service bulletin. System-wide service bulletins are valid for all routes/stops in the system.

## Schema

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="bustime-response" type="bustime-response"/>
  <xs:complexType name="bustime-response">
    <xs:sequence>
      <xs:element name="error" type="error" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="sb" type="servicebulletin" minOccurs="1" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```

<xs:sequence>
  <xs:element name="rt" type="xs:string" minOccurs="0" maxOccurs="1"/>
  <xs:element name="rtmdir" type="xs:string" minOccurs="0" maxOccurs="1"/>
  <xs:element name="stpid" type="xs:int" minOccurs="0" maxOccurs="1"/>
  <xs:element name="msg" type="xs:string" minOccurs="1" maxOccurs="1"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="servicebulletin">
  <xs:sequence>
    <xs:element name="nm" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="sbj" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="dtl" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="brf" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="prty" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="srvc" type="affectedservice" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="affectedservice">
  <xs:sequence>
    <xs:element name="rt" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="rtmdir" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="stpid" type="xs:int" minOccurs="0" maxOccurs="1"/>
    <xs:element name="stpnm" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>

```

## Example

The XML document below is a response to the following request:

Request:

<http://www.ctabustracker.com/bustime/api/v2/getservicebulletins?key=89dj2he89d8j3j3ksjhdue93j&stpid=456>

Response:

```

<?xml version="1.0"?>
<bustime-response>
  <sb>
    <sbj>Stop Relocation</sbj>
    <dtl>The westbound stop located at Madison/Lavergne has been moved to the northeast corner at
Madison/Lavergne.</dtl>
    <brf> The westbound stop located at Madison/Lavergne has been moved to the northeast corner at
Madison/Lavergne.</brf>
    <prty>low</prty>
    <srvc>
      <rt>20</rt>
    </srvc>
  </sb>
  <sb>
    <sbj>Stop Relocations/Eliminations</sbj>
    <dtl>Bus stops are being changed to provide faster travel time.</dtl>
    <brf>Bus stops are being changed to provide faster travel time.</brf>
    <prty>low</prty>
    <srvc>
      <stpid>456</stpid>
    </srvc>
  </sb>
</bustime-response>

```

## Error codes

Error Message	Related API Calls	Description
Internal server error - Unable to complete request at this time	All	The most general error message, given when we cannot find a more specific error message to send.
Invalid locale parameter	All	(Only in BusTime version 3.0+) The requested locale string is not in a proper format. The proper format is "la" or "la_CO" where la is a legal ISO 639 code and CO is a legal ISO 3166 code. Examples are "en" and "es_ES".
No API access permitted	All	The Developer API has been disabled by the Transit Authority.
No API access key supplied	All	The 'key=<DevKey>' parameter is missing from the API request.
Invalid API access key supplied	All	The given Developer key is not assigned to any users.
No version requested	All	The request URL is missing the version.
Unsupported version requested	All	The request URL contains an unsupported version. Currently only 'v1' is supported.
Unsupported function	N/A	The request contains a function name that is not supported by the API.
Transaction limit for current day has been exceeded.	All	The user, identified by the Developer Key, has already reached the maximum number of API calls allowed for the day.
No data found for parameter(s)	All except <b>gettime</b>	No results were found that matched the given parameters.
No parameter provided	<b>getpattern</b>	Required 'rt' or 'pid' parameters are missing.
No parameter provided	<b>getpredictions</b>	The required 'stpid' or 'vid' parameters are missing.
No parameter provided	<b>getservicebulletins</b>	The required 'rt' or 'stpid' parameters are missing.
dir parameter missing	<b>getstops</b>	The required 'dir' parameter is missing.
rt parameter missing	<b>getdirections,</b> <b>getstops,</b> <b>getservicebulletins</b>	The required 'rt' parameter is missing.
Either rt or vid parameter must be specified	<b>getvehicles</b>	The request is required to contain either a 'rt' or 'vid' parameter.
Invalid parameter provided	<b>getpatterns,</b> <b>getpredictions</b>	The stop ID or pattern ID does not match any known ID.
Maximum number of pid identifiers exceeded	<b>getpattern</b>	The 'pid' parameter contains more than 10 pattern IDs.

Invalid top parameter provided	<b>getpredictions</b>	The 'top' parameter is not an integer or contains extra characters. For instance "top=10" is legal but "top=10." is not.
Maximum number of <x> identifiers exceeded	<b>getpredictions, getvehicles</b>	The 'stpid' or 'vid' parameter contains too many IDs. <x> shows the maximum allowed in a single request.
No arrival times	<b>getpredictions</b>	The given stop has no scheduled arrival times.
No service scheduled	<b>getpredictions</b>	The given stop has no service scheduled.

# Developer License Agreement and Terms of Use

This Developer License Agreement and Terms of Use (“Agreement”) governs the use by developers and third parties (“you”) of electronic data owned and maintained by Chicago Transit Authority (“CTA,” “we” or “us”), including, but not limited to, CTA Bus Tracker data, CTA Train Tracker data, transitchicago.com Alerts data, General Transit Feed Specification (“GTFS”) and Scheduled Service data (collectively, “CTA Data”). CTA can change this Agreement at any time by posting a revised Agreement as CTA deems necessary. Your continued use of any CTA Data constitutes acceptance of those changes.

## I. Limited License

1. Subject to your compliance with this Agreement, CTA grants you a limited, non-exclusive, non-assignable, non-transferable and revocable license to use, reproduce, distribute, display, process and create derivative works of CTA Data as provided by CTA through the application programming interfaces (“APIs”), data feeds or otherwise for the sole purpose of assisting mass transportation (i.e., bus or rail) riders or in furtherance of promoting public transportation. Except as expressly provided in this Agreement, CTA Data may not be used, in whole or in part, for any other purpose, without the express permission of CTA.
2. By using the CTA Data, you agree to be bound by this Agreement. **IF YOU DO NOT AGREE TO BE BOUND BY THIS AGREEMENT, YOU MAY NOT USE CTA DATA.**

## II. Ownership of CTA Data

1. All right, title and interest in and to patents, copyrights, trademarks, and other intellectual property and ownership rights embodied in the original CTA Data, including, but not limited to, any new or useful art, discovery, improvement, technical development or invention, whether or not patentable, and all related know-how, designs, mask works, formulae, processes, trade secrets, ideas, artwork, software, hardware, graphics, layout, and modules, are CTA’s sole and exclusive property and are protected from unauthorized use, copying and distribution by U.S. and international laws, regulations and international treaties.
2. You acknowledge that CTA owns all rights in the CTA Data and you do not acquire any rights in the CTA Data by virtue of your use of the CTA Data under this Agreement or otherwise.
3. All uses of the CTA Data by you inure to the sole benefit of CTA to the extent they relate to intellectual property and other proprietary rights owned by CTA. Your use of the CTA Data does not in any way grant you the right to sell the CTA Data by itself as a stand-alone product.
4. You agree not to challenge, directly or indirectly, CTA’s rights in the CTA Data or assert any right, title or interest in or to the CTA Data or assist any third party in doing so.
5. Subject to the license to use the CTA Data and CTA Brand Assets (defined below), you retain ownership of your application, technology, website, feature, product, service, or derivative work, etc. that incorporates the CTA Data (collectively and each, an “application”) (including any code therein) and you are solely responsible for claims made in connection with or related to your application, and you agree to defend, indemnify and hold CTA harmless for any claims with respect thereto.

### III. Limitations on Use of CTA Data and Guidelines for Use of the CTA Brand Assets

Your use of CTA Data is subject to the following limitations and restrictions:

1. You and your application must not express or imply any affiliation or relationship with or sponsorship or endorsement by CTA.
  2. You may cache CTA Data you receive from CTA in order to improve your application’s user experience, but you agree to take reasonable efforts to keep the CTA Data up to date.
  3. You will not sell, auction or barter any CTA Data separate from your application. If you are acquired by or merge with a third party, you can continue to use CTA Data within your application subject to the terms of this Agreement, but you may not transfer the CTA Data outside your application.
  4. Subject to you abiding by this agreement and the [CTA Trademark Guidelines for Developers](#) (incorporated herein by reference), CTA grants you a limited, non-exclusive, non-assignable, non-transferable and revocable license and right to use the names, logos, trademarks, title treatments and art (the “CTA Brand Assets”) as provided by CTA that relate to the CTA Data, including the CTA, CTA BUS TRACKER, and CTA TRAIN TRACKER word marks and the CTA Bus Tracker Logo and CTA Train Tracker Logo, for your use solely in connection with offering your application.
  5. You may not use the CTA Brand Assets or any trademarks or terms confusingly similar to CTA’s trademarks in the name of your application or in any way that may suggest that CTA is affiliated with, endorses, or sponsors you or your application or that is likely to tarnish or dilute the CTA Brand Assets. You must comply with the [CTA Trademark Guidelines for Developers](#) which contain important restrictions, as well as examples of uses that are and are not acceptable.
  6. When using the CTA Data, you have the option, but not the obligation, to credit CTA. If you elect to include a credit line, it should read as follows: “Data provided by Chicago Transit Authority,” “Data provided by CTA” or “Powered by CTA data” or similar, descriptive language.
- Solely for purposes of a credit line accompanying a use of CTA Bus Tracker data, you have the option, but not the obligation, to use the logo icon shown below as more fully described in the appendices of this document, provided that such use otherwise comports with this Agreement. If you elect to include a credit line, it should read: “CTA Bus Tracker data provided by Chicago Transit Authority. The CTA Bus Tracker logo is a trademark of Chicago Transit Authority” or similar, descriptive language.



See [Branding Guide](#) for larger version

- Solely for purposes of a credit line accompanying a use of CTA Train Tracker data, you have the option, but not the obligation, to use the CTA Train Tracker logo shown below as more fully described in the appendices of this document, provided that such use otherwise comports with this Agreement. If you elect to include a credit line, it should read: “CTA Train Tracker data provided by Chicago Transit Authority. The CTA Train Tracker logo is a trademark of Chicago Transit Authority” or similar descriptive language.



See [Branding Guide](#) for larger version



## IV. Quality Control

1. You acknowledge that the CTA Brand Assets have acquired a most valuable goodwill and that it is of great importance to CTA that you, in the offering of any products or services in connection with the CTA Brand Assets, maintain the high standards and reputation CTA has established such that the goodwill associated with the CTA Brand Assets is protected at all times. You agree to maintain the same high standards of quality for any applications bearing the CTA Brand Assets.
2. You acknowledge that CTA owns all rights in the CTA Brand Assets and you do not acquire any rights in the CTA Brand Assets by virtue of its use pursuant to this Agreement.
3. All uses of the CTA Brand Assets by you inure to the sole benefit of CTA.
4. You agree not to challenge, directly or indirectly, CTA's rights in the CTA Brand Assets or assert any right, title or interest in or to the CTA Brand Assets or assist any third party in doing so.
5. You agree to cooperate with CTA and provide CTA with specimens of use of the CTA Brand Assets promptly upon CTA's request.
6. You are solely responsible for your application and its code and content and all uses you make of the CTA Data and CTA Brand Assets. This includes ensuring your use and your application's use of the CTA Data and CTA Brand Assets meets the requirements of this Agreement.

## V. Termination

1. Notwithstanding any other provision in this Agreement, if you are in breach of your obligations under this Agreement, including, without limitation, the [CTA Trademark Guidelines for Developers](#), CTA has the right to terminate this Agreement immediately.
2. CTA can take enforcement action against you and any or all of your applications if CTA determines, in our sole judgment, that you or your application violates this Agreement. Enforcement actions can include disabling your application, restricting you and your application's access to the CTA Data, terminating our agreements with you, or any other action as we in our sole discretion deem appropriate.
3. Upon any termination or expiration of this Agreement, you will promptly delete all CTA Data from your application, computer systems or other storage devices and discontinue all use of the CTA Brand Assets. You agree to certify in writing that it has performed its obligations under this Section V if CTA requests a certification.

## VI. No Guarantee or Warranty of Availability of CTA Data

1. CTA reserves the right at any time and from time to time to discontinue posting the CTA Data (or any part thereof), temporarily or permanently, with or without notice to you.
2. You agree that CTA will not be liable to you for any modification, suspension or discontinuance of the availability of the CTA Data.
3. CTA reserves the right to alter, correct and modify CTA Data at any time without prior notice to you.



## VII. Disclaimer of Warranties

1. YOU AGREE THAT CTA DATA IS PROVIDED ON AN "AS IS" AND "AS AVAILABLE" BASIS. TO THE FULLEST EXTENT PERMITTED BY LAW, CTA, ITS EMPLOYEES, OFFICERS, DIRECTORS AND AGENTS (COLLECTIVELY, THE "CTA PARTIES") MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, AS TO THE CTA DATA, ITS ACCURACY, COMPLETENESS, USEFULNESS, TIMELINESS, RELIABILITY OR FUNCTIONALITY, OR THAT ANY PRODUCTS OR SERVICES MADE AVAILABLE BY THE CTA OR THIRD PARTIES INCORPORATING THE CTA DATA, AND/OR THE SECURITY ASSOCIATED WITH THE TRANSMISSION OF CTA DATA. THE CTA PARTIES HEREBY DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, ACCURACY, AND NON-INFRINGEMENT. YOU AGREE THAT USE OF CTA DATA IS AT YOUR SOLE RISK. SOME JURISDICTIONS DO NOT ALLOW THE DISCLAIMER OF IMPLIED OR OTHER WARRANTIES, SO THE ABOVE DISCLAIMER MAY NOT APPLY TO YOU.
2. THE CTA PARTIES ARE NOT RESPONSIBLE FOR, AND DO NOT ENDORSE, THIRD PARTY CONTENT, APPLICATIONS AND/OR MATERIALS INCORPORATING CTA DATA THAT MAY BE MADE AVAILABLE FROM TIME TO TIME ON CHICAGOTRANSITAUTHORITY.COM OR ELSEWHERE (COLLECTIVELY, THE "CTA WEB SITE") (EVEN IF THE CTA WEB SITE PROVIDES A LINK TO OR OTHERWISE ADVERTISES SUCH THIRD PARTY CONTENT, APPLICATIONS AND/OR MATERIALS) AND SPECIFICALLY DISCLAIMS ANY RESPONSIBILITY OR LIABILITY TO ANY PERSON OR ENTITY, INCLUDING, WITHOUT LIMITATION, ANY PERSON OR ENTITY THAT MAY USE OR RELY ON SUCH THIRD PARTY CONTENT, APPLICATIONS AND/OR MATERIALS, EVEN IF THEY INCORPORATE CTA DATA, FOR ANY LOSS, DAMAGE (WHETHER ACTUAL, CONSEQUENTIAL OR OTHERWISE), INJURY, CLAIM, LIABILITY OR OTHER CLAIM OF ANY KIND BASED UPON OR RESULTING FROM SUCH THIRD PARTY CONTENT, APPLICATIONS AND/OR MATERIALS.

## VIII. Limitation of Liability

YOU AGREE THAT THE CTA PARTIES WILL NOT BE LIABLE FOR LOSSES OR DAMAGES OF ANY KIND ARISING FROM THE USE OF CTA DATA, INCLUDING, BUT NOT LIMITED TO, DIRECT, INDIRECT, INCIDENTAL, PUNITIVE AND CONSEQUENTIAL LOSSES OR DAMAGES, REGARDLESS OF WHETHER SUCH LOSSES OR DAMAGES ARISE BASED UPON CONTRACT, NEGLIGENCE, STRICT LIABILITY IN TORT, WARRANTY OR OTHER LEGAL THEORY AND REGARDLESS OF WHETHER SUCH LOSS OR DAMAGES WERE FORESEEABLE, EVEN IF THE CTA PARTIES WERE ADVISED OF OR SHOULD HAVE KNOWN OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT WILL THE CTA PARTIES' TOTAL LIABILITY TO YOU FOR LOSS OR DAMAGES EXCEED TEN U.S. DOLLARS. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

## IX. Indemnification

1. You agree to indemnify, defend and hold the CTA Parties harmless from and against any and all fines, suits, proceedings, claims, causes of action, demands, or liabilities (including reasonable attorney's fees) of any kind or of any nature arising out of or in connection with: (i) your use of CTA Data; (ii) a breach or alleged breach of your representations, warranties, agreements or obligations in this Agreement; (iii) any actual or alleged infringement, misappropriation, or violation of any third party's intellectual property or other rights, including, without limitation, patents, copyrights, trademarks, service marks, or trade secrets by you or your application; and (iv) the development, marketing, distribution, licensing, sale and/or use of any applications, technology, products or services sold, licensed or otherwise distributed by you.



2. You agree not to settle any claim, action or lawsuit subject to this Section IX without the prior written consent of CTA.
3. You agree that CTA will have the right, but not the obligation, to participate in any claim, action or lawsuit for which you have an indemnification obligation under this Agreement.

## **X. Binding Effect; No Assignability without Express Consent**

1. The terms of this Agreement are binding upon both you and CTA and inure to the benefit of the successors and assigns of the parties as if they were parties to this Agreement.
2. None of the rights, duties or obligations under this Agreement may be assigned by you without the express written consent of an authorized CTA representative.

## **XI. Severability**

To the extent a court of competent jurisdiction determines that any part or provision of this Agreement is unenforceable as a matter of law, such part or provision of this Agreement will be deemed severable and the remainder of this Agreement will survive and remain enforceable.

## **XII. Applicable Law and Forum; Compliance with Laws**

1. The laws of the State of Illinois govern all rights and obligations under this Agreement, without giving effect to any principles of conflicts of laws.
2. Any use of the CTA Data or CTA Brand Assets will be deemed made in the State of Illinois, USA, regardless of your location. You agree that any dispute with CTA arising out of any use of the CTA Data, the CTA Brand Assets or this Agreement must be brought by you exclusively in the state or federal courts situated in Cook County, State of Illinois. You agree that such venue is appropriate.
3. You agree to at all times observe and comply with all applicable laws, ordinances, rules, regulations and executive orders of Federal, state and local government entities, now existing or hereinafter in effect, which may in any manner affect the performance of this Agreement. Provision(s) required by law, ordinances, rules, regulations or executive orders to be inserted herein will be deemed inserted herein whether or not they appear in this Agreement or, upon application by either party, this Agreement will forthwith be amended to literally make such insertion; however, in no event will failure to insert such provision(s) prevent the enforcement of this Agreement.

## **XIII. Entire Agreement**

This Agreement constitutes the complete and exclusive agreement between CTA and you with respect to the subject matter hereof and supersedes all prior oral or written understandings, communications, or agreements not specifically incorporated herein. CTA reserves the right to modify or revoke this Agreement at any time.

## **XIV. No Waiver**

Waiver by CTA of strict performances of any provision of this Agreement will not be a waiver of or prejudice CTA's right to require strict performance of the same provision in the future or of any other provision of this Agreement.

## **XV. No Agency, Joint Venture or Affiliation**

No provision of this Agreement, nor any act by you or CTA, will be deemed or construed by either of the parties or by third parties, to create any relationship of third party beneficiary, or of principal or agent, or of limited or general partnership, or of joint venture, or of any association or relationship involving you and CTA.

## Acknowledgements

The [CTA Bus Tracker](#) service is powered by [BusTime](#), a product of [Clever Devices, Inc.](#) Both the API itself and the majority of this document were produced and developed by people at Clever Devices.

Special thanks go to Harper Reed and Dan O'Neil for their support and encouragement, and to the independent development community, for showing such great interest in developing applications with CTA data, leading to the creation of this official API.

Thank you.