Transport for London

# TfL Live Bus & River Bus Arrivals API

## Interface Documentation

**Transport for London**
50 Victoria Street
Westminster
London
SW1H 0TL

www.tfl.gov.uk/developers
developers@tfl.gov.uk

Please pass any queries to developers@tfl.gov.uk with *Bus & River Bus Arrivals API Feedback* in the subject line

## Table of Contents

**Document history**

| Document Version | Software Version | Date | Name | Revision Type |
|---|---|---|---|---|
| 1.0 | 1.0 | 05/04/2012 | TS | Version for issue |
| 1.1 | 1.0 | 03/05/2012 | TS | Updates made following feedback from Beta |
| 1.2 | 1.0 | 30/05/2012 | TS | Updates made following internal feedback |
| 1.3 | 1.0 | 12/10/2012 | AR | Guidelines added to the fields 'DestinationName' and 'RegistrationNumber' in chapter 6. |
| 1.4 | 1.0 | 28/03/2013 | TS | Added River Bus information. Updated stop types to be displayed. Clarify use of StartTime for flexible messages. |

# 1 Introduction

## 1.1 The Live Bus & River Bus Arrivals API

TfL's Countdown system provides real-time bus and river bus arrival and service disruption information for passengers across London. Using data from iBus (tfl.gov.uk/iBus), the system provides passengers with an accurate and complete information service for every one of London's 19,000 bus stops as well as for the TfL river bus piers. This data is currently provided over the web, via SMS and also on on-street signs.

A key part of TfL's strategy is to provide data openly. This API is designed to enable application developers to subscribe to live bus information and use this data to develop innovative services appropriate to their market and clients.

The data that is accessible via the API is taken from the same source systems as TfL's Countdown service. This ensures that the information supplied is consistent with other delivery channels, meaning that the end user is presented with consistent information.

## 1.2 Accessing the API

In order to access the API it is necessary to register at the TfL developer's area – http://www.tfl.gov.uk/developers

## 1.3 Attribution

Please do not include any TfL branding in your application or give the impression it is an official TfL application. Please add the attribution "Data provided by Transport for London".

## 1.4 Support

SLAs or guaranteed support are not offered with this API.

## 1.5 API Overview

The TfL Live Bus & River Bus Arrivals API is controlled via a number of different HTTP requests and parameters.

The API is based on JSON, however the responses deviate away from the JSON standard. This is primarily to optimise the performance of the API. It also allows the streaming API to be consistent with the instant API. This document describes and provides examples of the output format in order for developers to utilise the data.

A request is structured as follows:

```
http://server/virtualDirectory/type/version?HTTP parameters
```

# 2 Data served by the API

The data that is made available by this interface can broadly be put into two categories; real time data and reference data. By providing both categories of data over a single interface, developers are able to request real time data alongside the contextualising reference data in a single request. Data provided is stop centric; therefore the stop information is central to the request.

The scale of the bus network means that there is a large amount of data that can be accessed using this API. Some indicative figures to give an idea of the magnitude of the data are as follows:

- Over **19,000** bus stops
- Over **700** bus routes
- Over **8000** buses
- Approximately **130,000** bus arrival predictions at any point in time

In addition to the bus network data, the TfL River Bus network is also included in the API. This data is modelled in the same way as the bus data and therefore is provided in the same API.

Definitions of all data fields are provided in section 6.

## 2.1 Real time data

Live Bus & River Bus Arrivals provides the predicted time until a bus or river bus is expected to arrive at a stop. The system provides live bus/river bus arrival times, including destinations, for the next 30 minutes. This data is refreshed at source (tfl.gov.uk/countdown) every 30 seconds. It is therefore unnecessary to request this data at any interval more frequent than this.

Flexible messages provide service information and are assigned to specific bus stops. They are updated much less frequently than bus arrival information although may be added or removed at any time. It should be noted that flexible messages have both a start time and an expiry time. For operational reasons, messages are often added to the system prior to them being valid. It is therefore important that they are only displayed during the specified time window.

Stop closure information is the least variable of the real time data. Like service information messages, these closures may happen at any time.

## 2.2 Reference data

Reference data is primarily provided by the interface in order to give context to the real time data supplied. Additionally, developers have the option of requesting only reference data items and not real-time information.

Reference data is also sourced from TfL's Countdown system, this data is updated on a fortnightly basis in order to ensure that the data remains consistent with the rapidly changing London bus network. Reference data is versioned using a Baseversion, which is available via the API. Additional reference data, including bus timetables is available on the TfL developer area (TfL.gov.uk/developers).

Section 6 - Glossary of terms indicates whether a data item is reference data or real time data.

# 3 Service Behaviour

## 3.1 Instant and Streaming requests

The service provides two types of request to users; instant (request/response) and streaming.

Instant requests are made by the client and the server will respond with a single message. After this the client will not receive any further updates unless they make further requests.

Streaming requests are made by the client, in response the server will continually serve data to satisfy the request until the connection is terminated. Streaming data requires additional authentication owing to the potentially very large amounts of data to be transferred.

The data source for both instant and streaming requests is consistent, ensuring that the data provided to the public remains the same.

## 3.2 Caching

Data is cached in the system for a period of **30 seconds**. Hence there is no benefit to the developer in querying any of the data services any more frequently than once every 30 seconds.

## 3.3 URA Versioning

An increase in the minor version indicates new functions on the server side, but queries against older versions must return the same responses, i.e. a Client using URA V1.1 must get the same answer from a URA V1.2 or URA V1.3 server. An increase in the major version indicates a non-compatible change in the protocol.

## 3.4 HTTP status codes

In normal use, the service will provide HTTP status codes. The use of these codes is as follows:

| HTTP Code | Reason |
|---|---|
| 200 OK | Service is working correctly and the response contains data. |
| 400 Bad Syntax | If the URL is malformed, such as it is not in exact conformance with this document, then the HTTP code of 400 is returned.<br><br>The client should not retry and needs to change the request |
| 401 Unauthorized | The Client did not provide, or provided incorrect, authentication details (username and password) and therefore the request was not served.<br><br>Authentication is required in order to stream data. |
| 408 Request Timeout | The server timed out waiting for the request. |
| 416 Requested Range Not Satisfiable | The Client used a filter criteria used was not within range. |

| | |
|---|---|
| | The filter criteria used may not exist in the static data, or may be formally out of range. |
| 500 Internal Server Error | A generic error message, given when no more specific messages are suitable. |
| 502 Bad Gateway | There is an error in the upstream data supply or the supply is not available. |

**Table 1 - HTTP Status Codes**

# 4 Technical guide to using the service

This guide relates to version 1.0 of the API.

The service is accessed by requesting a specific URL. This URL instructs the server as to what data should be returned.

The server responds with a UTF-8 JSON message. Depending on the request, the response can be composed of 5 different array types; Stop, Prediction, Flexible Message, Baseversion or URA Version arrays. Full details of the response format and these arrays are detailed in section 4.2 of this document.

## 4.1 Request

There are two different URLs for the service, one is for instant (request/response) usage and the other is for data streaming. The API for these services is the same; with the exception of deletion messages which are only necessary when streaming.

In order to obtain the URLs for the API it is necessary to register at the TfL developer's area – http://www.tfl.gov.uk/developers

As part of a request it is necessary to specify the parameters for the service. In specifying this string, the client instructs the server as to what data items should be returned and also provides the filters to restrict the data that is returned. Parameters should be added as comma separated values.

The full list of parameters that can be set are listed below.

### 4.1.1 Request parameters

| Parameter | Type | Valid Values | Default | Multiple Values Allowed | Description |
|---|---|---|---|---|---|
| Stream | Boolean | True/False | False | No | Enables streaming. In order to stream data authentication is required. |
| StopAlso | Boolean | True/False | False | No | If set to true, the service will return Stop arrays (Type 0) for stops even if they do not have predictions or flexible messages currently associated with them.<br><br>This will only work for |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | stop based requests. It will not work when selecting based on a route (LineID, LineName) |
| ReturnList | Comma-separated Strings | StopPointName,StopID,StopCode1,StopCode2, StopPointState, StopPointType, StopPointIndicator, Towards, Bearing, Latitude,Longitude,VisitNumber, TripID, VehicleID, RegistrationNumber,LineID,LineName,DirectionID,DestinationText,DestinationName,EstimatedTime, MessageUUID, MessageText,MessageType,MessagePriority, StartTime,ExpireTime, BaseVersion | StopPointName, LineName, EstimatedTime | No | List of fields to return to client. |
| Circle | Comma-separated Strings | GPS Coordinates plus radius, Format is Circle=Latitude, Longitude,Radius (in m),e.g.<br><br>Circle=12.3121412,14.1231241,100 | Unset | No | Filter response for stops within a given radius (in meters) from a specified latitude and longitude. (WGS84 coordinate system) |
| StopPointName | Comma-separated Strings | Every String | Unset | Yes | Filters response for only stops with that name |
| StopID | Comma-separated Strings | Every String | Unset | Yes | Filters response for only stops with that StopID |
| StopCode1 | Comma-separated | Every String | Unset | Yes | Filters response for only stops with that Stop |

| | | | | | |
|---|---|---|---|---|---|
| | Strings | | | | Code 1 |
| StopCode2 | Comma-separated Strings | Every String | Unset | Yes | Filters response for only stops with that Stop Code 2 |
| StopPointType | Comma-separated Strings | Every String | Unset | Yes | Filters response for only stops with that StopPointType |
| Towards | Comma-separated Strings | Every String | Unset | Yes | Filters response for only stops with the specific Towards value |
| Bearing | Comma-separated Numbers | Integer from 0…359 | Unset | Yes | Filters response for only stops with the specific Bearing value |
| StopPointState | Comma-separated Numbers | Positive Integer | Unset | Yes | Filters response for only stops with the specific StopPointState. |
| VisitNumber | Comma-separated Numbers | Positive Integer | Unset | Yes | Filters response for only predictions with that sequence counter value |
| LineID | Comma-separated Numbers | Every String | Unset | Yes | Filters response for predictions only with that Line ID |
| LineName | Comma-separated Numbers | Every String | Unset | Yes | Filters response for predictions only with that Line Text |
| DirectionID | Number | 1 or 2 | Unset | No | Filters response for predictions only with that Direction |
| DestinationText | Comma-separated Strings | Every String | Unset | Yes | Filters response for only predictions with that Destination Text |
| DestinationName | Comma-separated Strings | Every String | Unset | Yes | Filters response for only predictions with that Destination Short Text |
| VehicleID | Comma-separated Strings | Every String | Unset | Yes | Filters response for only predictions for a specific Vehicle |
| TripID | Comma-separated Numbers | Positive Integers | Unset | Yes | Filters response for only predictions for a specific Journey |

| | | | | | |
|---|---|---|---|---|---|
| Registration Number | Comma-separated Strings | Every String | Unset | Yes | Filters response for only predictions for a vehicle with the specific Registration Number |
| StopPointIndicator | Comma-separated Strings | Every String | Unset | Yes | Filters response for stops with the specific Stop Indicator |
| MessageType | Comma-separated Numbers | Positive Integer | Unset | Yes | Filters response for only Message with that type. |
| MessagePriority | Comma-separated Numbers | Positive Integer 1..10 | Unset | Yes | Filters response for Message with that Priority |

**Table 2 - Request Parameters**

Note that to transform a usual String into an Escaped String, the following modifications have to be made:

- Escape'&' by '\a'
- Escape',' by '\c'
- Escape result of string as JSON String by RFC 4627
- Percent-encoded result as URL String by RFC 3986

### 4.1.2 Request guidelines

- Clients should reduce the ReturnList to the absolute minimum in order to reduce load on both the client and server
- The client should not request MessageType or MessagePriority unless MessageText is part of the ReturnList
- The client should request the ExpireTime when requesting either the ExpectedTime OR MessageText fields
- The client should request the StartTime when requesting the MessageText field
- Leaving out filter criteria means that the client will get the full set of data. E.g. leaving out the message field will return all message types
- Filter criteria are not case sensitive
- The “If-Modified-Since” HTTP feature is not supported

## 4.2 Response

The server responds with a UTF-8 JSON message. Depending on the request, the response can be composed of 5 different array types; Stop, Prediction, Flexible

Message, Baseversion or URA Version arrays. The sequence of these arrays in the response is undefined, except that the URA Version array always appears first.

An 'empty' response will contain only a URA Version array.

The sequence of fields returned within each response array will always follow the sequence detailed below, regardless of the order specified in the request. If a field is not specified in the ReturnList of the request then it is skipped.

In responses to Stream requests, white space only data may be sent by the server as a heartbeat. The client MUST ignore any whitespace only transmission.

### 4.2.1 Stop array

The stop array contains reference data about bus stops and piers. It does not contain any real time data. This array is particularly useful if real time data is not required. It is also used in order to return data for a stop when a stop does not have any predictions or flexible messages associated with it (the StopAlso request parameter should be set to 'true' to return these arrays).This will only work for stop based requests. It will not work when selecting based on a route (LineID, LineName)

The primary key of the Stop array is StopID.

| Sequence Nr | Field | JSON-Type | Valid Values | Stop Primary Key |
|---|---|---|---|---|
| 0 | ResponseType | Number | For the Stop array this is always 0 | No |
| 1 | StopPointName | String | As per Static Data | No |
| 2 | StopID | String | As per Static Data | Yes |
| 3 | StopCode1 | String | As per Static Data | No |
| 4 | StopCode2 | String | As per Static Data | No |
| 5 | StopPointType | String | As per Static Data | No |
| 6 | Towards | String | As per Static Data | No |
| 7 | Bearing | Number | As per Static Data, Integer 0…359 degrees | No |
| 8 | StopPointIndicator | String | As per Static Data | No |
| 9 | StopPointState | Number | Integer >= 0 | No |
| 10 | Latitude | Number | Every Number | No |
| 11 | Longitude | Number | Every Number | No |

**Table 3 - Stop array**

### 4.2.2 Prediction array

The prediction array contains the predicted arrival times for particular buses / river buses at stops. The array also contains the reference data for the routes, stops and vehicles that appear in the predictions.

The prediction array has a compound primary key comprised of StopID, VisitNumber, DestinationText and VehicleID.

| Sequence Nr | Field | JSON-Type | Valid Values | Prediction Primary Key |
|---|---|---|---|---|
| 0 | ResponseType | Number | For the Prediction array this is always 1 | No |
| 1 | StopPointName | String | As per Static Data | No |
| 2 | StopID | String | As per Static Data | Yes |
| 3 | StopCode1 | String | As per Static Data | No |
| 4 | StopCode2 | String | As per Static Data | No |
| 5 | StopPointType | String | As per Static Data | No |
| 6 | Towards | String | As per Static Data | No |
| 7 | Bearing | Number | As per Static Data, Integer 0…359 degrees | No |
| 8 | StopPointIndicator | String | As per Static Data | No |
| 9 | StopPointState | Number | Integer >= 0 | No |
| 10 | Latitude | Number | Every Number | No |
| 11 | Longitude | Number | Every Number | No |
| 12 | VisitNumber | Number | Every positive Integer | Yes |
| 13 | LineID | String | As per Static Data | No |
| 14 | LineName | String | As per Static Data | No |
| 15 | DirectionID | Number | 1 or 2 | No |
| 16 | DestinationText | String | As per Static Data | Yes |

| | | | | |
|---|---|---|---|---|
| 17 | DestinationName | String | As per Static Data | No |
| 18 | VehicleID | String | Every String | Yes |
| 19 | TripID | Number | As per Static Data | No |
| 20 | RegistrationNumber | String | As per Static Data | No |
| 21 | EstimatedTime | Number | Every positive Integer (UTC as per Unix Epoch in milliseconds). To convert to UTC Epoch divide by 1000. 64 bit. | No |
| 22 | ExpireTime | Number | Every positive Integer (UTC as per Unix Epoch in milliseconds). To convert to UTC Epoch divide by 1000. 64 bit.<br><br>This field should always be requested if the client is requesting EstimateTime.<br><br>In stream mode, this field will be set to 0 when the prediction needs to be deleted. | No |

**Table 4 - Prediction array**

### 4.2.3 Flexible Message array

The flexible message array returns flexible messages that are associated with bus stops or piers. These messages inform passengers about incidents and service disruptions relevant to their journey. As with the prediction array, the flexible message array also contains reference data.

The flexible message array has a compound primary key consisting of StopID and MessageUUID.

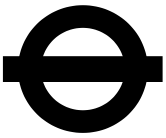
| Sequence Nr | Field | JSON-Type | Valid Values | FLM Primary Key |
|---|---|---|---|---|
| 0 | ResponseType | Number | 2 | No |
| 1 | StopPointName | String | As per Static Data | No |
| 2 | StopID | String | As per Static Data | Yes |
| 3 | StopCode1 | String | As per Static Data | No |
| 4 | StopCode2 | String | As per Static Data | No |

| | | | | |
|---|---|---|---|---|
| 5 | StopPointType | String | As per Static Data | No |
| 6 | Towards | String | As per Static Data | No |
| 7 | Bearing | Number | As per Static Data, Integer 0…359 degrees | No |
| 8 | StopPointIndicator | String | As per Static Data | No |
| 9 | StopPointState | Number | Integer >= 0 | No |
| 10 | Latitude | Number | Every Number | No |
| 11 | Longitude | Number | Every Number | No |
| 12 | MessageUUID | String | Every String | Yes |
| 13 | MessageType | Number | Integer >= 0 | No |
| 14 | MessagePriority | Number | Integer 1...10 (where 1 is the highest priority). | No |
| 15 | MessageText | String | Any String | No |
| 16 | StartTime | Number | Every positive Integer (UTC as per Unix Epoch in milliseconds). To convert to UTC Epoch divide by 1000. 64 bit. | |
| 17 | ExpireTime | Number | Every positive Integer (UTC as per Unix Epoch in milliseconds). To convert to UTC Epoch divide by 1000. 64 bit.<br><br>In stream mode, this field will be set to 0 when the flexible message needs to be deleted. | No |

**Table 5 - Flexible Message array**

### 4.2.4 Baseversion array

The baseversion is used by LBSL to version static data. Where appropriate, it should be used when trying to join the Live Bus & River Bus Arrivals data with other LBSL data sets. Section 6 ‘Glossary of terms‘ details which data items are Baseversion controlled. These items can only change with a change of baseversion, therefore clients may wish to use the Baseversion as an indicator as to whether they need to update this data item.



If the Baseversion request field is set, the server answers will always contain a separate array containing just the baseversion. In stream mode this array will be send out when the baseversion changes.

The client must not make assumptions about the format of the name

| Sequence Nr | Field | JSON-Type | Valid Values |
|---|---|---|---|
| 0 | ResponseType | Number | 3 |
| 1 | Version | String | As per Static Data |

**Table 6 - Baseversion array**

### 4.2.5 URA Version array

The URA version array provides the version of the URA that is being used. The array will always be the first array in the response.

The timestamp provides the time that the response was processed. It is synchronised with other timestamps provided by the Countdown system. For streaming requests, this timestamp is only provided on connection to the stream.

| Sequence Nr | Field | JSON-Type | Valid Values |
|---|---|---|---|
| 0 | ResponseType | Number | 4 |
| 1 | Version | String | Integer.Integer (e.g. “1.1”) |
| 2 | TimeStamp | Number | Every positive Integer (UTC as per Unix Epoch in milliseconds). To convert to UTC Epoch divide by 1000. 64 bit. |

**Table 7 - URA version array**

# 5 Examples of Usage

The examples provided below are for guidance only. They provide guides as to the types of the request that can be made using this API.

Note that in order to improve readability, whitespace and line breaks have been used in the examples below. It should not be assumed that the response provided by the server will conform exactly to this format.

## 5.1 Instant request examples

### 5.1.1 Request predictions and flexible messages for a single stop

This request is for prediction and flexible message information for a single stop. This may either be a bus stop or a pier. This sort of request is likely to be used by mobile applications in order to display real-time data for a user selected stop.

**Request**

/interfaces/ura/instant?StopCode1=52053&DirectionID=1&VisitNumber=1&ReturnList=StopCode1,StopPointName,LineName,DestinationText,EstimatedTime,MessageUUID,MessageText,MessagePriority,MessageType,ExpireTime

**Example Content Response**

[4,"1.0",1334925465143] - URA Version array.

[1,"Green Park Station","52053","22","Piccadilly Cir",1334925458000,1334927227146] - first Prediction array

[1,"Green Park Station","52053","14","Warren Street",1334925830000,1334927247004]

[1,"Green Park Station","52053","22","Piccadilly Cir",1334925731000,1334926994196]

...

[1,"Green Park Station","52053","14","Warren Street",1334926824000,1334926832021]

[1,"Green Park Station","52053","22","Piccadilly Cir",1334926836000,1334926844473]

[1,"Green Park Station","52053","14","Warren Street",1334927168000,1334927176525]

[[2,"Green Park Station","21961","8a56a2ac359ff7df0136074830af4b26_99",0,3,"Test message for Green Park",1333545681000] –first Flexible Message array

### 5.1.2 Request predictions for multiple stops

This request is for predictions for two routes at multiple bus stops. This sort of request is likely to be used to allow a passenger that has a choice of stops that serve their destination to determine which to depart from.

**Request**

/interfaces/ura/instant?StopCode1=58726,51586&LineName=c10,507&ReturnList=StopPointName,Li

| |
|---|
| neName,DestinationText,EstimatedTime,ExpireTime |
| **Example Content Response**<br>[4,"1.0",1334928941477] - URA Version array.<br>[1,"Marsham Street","507","Victoria",1334929424000,1334930553134] - first Prediction array<br>[1,"Marsham Street","507","Victoria",1334929784000,1334930548807]<br>[1,"Marsham Street","507","Victoria",1334928961000,1334930715448]<br>[1,"Marsham Street","507","Victoria",1334930151000,1334930544495]<br>[1,"Marsham Street","507","Victoria",1334930534000,1334930593771]<br>[1,"Page Street","C10","Victoria",1334929908000,1334930534667]<br>[1,"Page Street","C10","Victoria",1334929194000,1334930548463] |

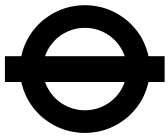
### 5.1.3 Request stops within a specified radius

This request returns static data for all 'Open' bus stops within a radius of a specified point. This is likely to be of use when returning bus stops that are close to a mobile user's location.

| |
|---|
| **Request**<br>/interfaces/ura/instant?Circle=51.49598,-0.14191,250&StopPointState=0&ReturnList=StopCode1,StopPointName,Bearing,StopPointIndicator,StopPointType,Latitude,Longitude |
| **Example Content Response**<br>[4,"1.0",1334930109388] - URA Version array<br>[0,"Bressenden Place / Victoria Station","91545","STBC",165,"CN",51.497219,-0.141818] - first Stop array<br>[0,"Victoria Station, Bus Station Stand",null,null,null,null,51.496169,-0.143633]<br>[0,"Victoria Bus Station, Stand D",null,null,null,null,51.496103,-0.14401]<br>…<br>[0,"Victoria Station","56026","STBC",92,"F",51.496239,-0.143514]<br>[0,"Victoria Station","57096","STBC",338,"H",51.495648,-0.143106] |

### 5.1.4 Request all predictions for a specific vehicle

This request returns prediction information for a specific vehicle. This is likely to be used in an application designed to allow a passenger to view the predicted arrival times (for the next 30 minutes) of their specific bus. This would give the user an indication of how long until they reach their destination.



| **Request** |
| --- |
| /interfaces/ura/instant?RegistrationNumber=LX59DDF&ReturnList=StopCode1,EstimatedTime,ExpireTime,Baseversion,RegistrationNumber |
| **Example Content Response** |
| [4,"1.0",1334932175872] - URA Version array<br>[3,"20120417"]– Baseversion array<br>[1,"52954","LX59DDF",1334933329000,1334933330949] – first Prediction array<br>[1,"52374","LX59DDF",1334933383000,1334933385304]<br>[1,"74720","LX59DDF",1334932474000,1334933795889]<br>[1,"49429","LX59DDF",1334932293000,1334933884537]<br>[1,"50476","LX59DDF",1334933045000,1334933053570] |

## 5.2 Stream examples

### 5.2.1 Stream prediction data for a single stop

This request is to return a stream of continuous data for two stops.

| **Request** |
| --- |
| /interfaces/ura/stream?Stopid=99,13551&ReturnList=Stoppointname,VehicleID,RegistrationNumber,LineName,DestinationName,EstimatedTime,ExpireTime |
| **Example Content Response** |
| [4,"1.0",1332280681000] - URA Version array.<br>[1,"Green Park Station","Z19","Finsbury Park Interchange",942,"LJ51DBV",1332280682000,1332280682000] –first Prediction array<br>[1,"Green Park Station","19","Finsbury Park Interchange",938,"X538GGO",1332280650000,1332280650000] –second Prediction array<br>…<br>[1,"Green Park Station","Z19","Finsbury Park Interchange",942,"LJ51DBV",1332280682000,1332280682000]<br>[1,"Green Park Station","19","Finsbury Park Interchange",938,"X538GGO",1332280650000,1332280650000]<br>[1,"Green Park Station","9","Aldwych",14811,"W141EON ",1332280976000,1332280976000]<br>[1,"Green Park Station","19","Finsbury Park Interchange",1578,"LF52URT,1332281123000,1332281123000] |

### 5.2.2 Stream all data for all stops

This request is to return a stream of continuous data for all stops.

**Request**

/interfaces/ura/stream?ReturnList=StopPointName,StopID,StopCode1,StopCode2,StopPointState,StopPointType,StopPointIndicator,Towards,Bearing,Latitude,Longitude,VisitNumber,TripID,VehicleID,RegistrationNumber,LineID,LineName,DirectionID,DestinationText,DestinationName,EstimatedTime,MessageUUID,MessageText,MessageType,MessagePriority,ExpireTime,BaseVersion

**Example Content Response**

[4,"1.0",1332197936000] - URA Version array.

[3,"20111225"] - Baseversion array

[1,"Centre Common Road / War Memorial","17522","25633","490005205N","STBC","ELTHAM OR GROVE PARK",323,"N",0,51.413472,0.072582,1,"Z161","Z161",1,"North Greenwich","North Greenwich Station",18225,254131,"YN06JXY ",1332197937000,1332197937000] – first Prediction array

[1,"Cedar Way","10107","16095","03700083","STBR","SLOUGH",303,null,0,51.497457,-0.556007,1,"81","81",2,"Slough","Slough Bus Station",14502,856858,"YN55NHA ",1332198209000,1332198209000] –second Prediction array

…

[1,"New Cross Gate Station","26343","22474","490000156O","STBC","PECKHAM OR BRICKLAYERS ARMS",246,"O",0,51.475048,-0.039426,1,"Z53","Z53",1,"Whitehall","Horse Guards Parade",21188,300000,"TEST_21188 ",1332197619000,1332197619000]

# 6 Glossary of terms

The API used is not specific to TfL. As a result, many of the names used are generic so that they can be populated with different data according to the transport authority using the service. For this reason it is necessary to provide a reference between the field names in this API and those used elsewhere by TfL. A description of each field is provided along with an indication whether the data item is reference or real time.

Reference data will not change within a particular baseversion.

| Field | TfL static data | Description | Data type |
|---|---|---|---|
| Bearing | Heading | Direction the vehicle is travelling in when it arrives at the bus-stop. This is expressed from 0° to 359° | Reference |
| DestinationName | Long_Destination_Name | The full length destination name of the trip the vehicle is on. The destination name is based on the route and end point of the trip.<br><br>Note: This is currently displaying the wrong information and should not be used, instead the 'DestinationText' should be used. This will be corrected in version 2.0 of the API. | Real time |
| DestinationText | Short_Destination_Name | The abbreviated destination name of the trip the vehicle is on. The destination text is based on the route and end point of the trip. | Real time |
| DirectionID | Direction | This identifies the direction of the trip that the vehicle is on. It indicates whether the vehicle is on an outbound or inbound trip. | Reference |
| EstimatedTime | N/A | This is the predicted time of arrival for the vehicle at a specific stop.<br><br>It is an absolute time in UTC as per Unix Epoch (in milliseconds). To convert to UTC Epoch divide by 1000. | Real time |
| ExpireTime | N/A | This is the time at which the corresponding prediction or flexible message is no longer valid and should stop being displayed.<br><br>It is an absolute time in UTC as per Unix Epoch (in milliseconds). To | Real time |

| | | | |
|---|---|---|---|
| | | convert to UTC Epoch divide by 1000.<br><br>In stream mode, this time may be set to 0 to indicate that a message should be deleted. | |
| Latitude | Location_Latitude | The latitude of the stop. This is expressed using the WGS84 coordinate system. | Reference |
| LineID | Contract_Line_No | The identifier of a route. This is an internal identifier and is not equal to the route number displayed on the front of the bus / river bus. It should not be displayed to the public. | Reference |
| LineName | Service_Line_No | This is the route number that is displayed on the front of the bus / river bus and on any publicity advertising the route. | Reference |
| Longitude | Location_Longitude | The longitude of the stop. This is expressed using the WGS84 coordinate system. | Reference |
| MessagePriority | N/A | Messages are assigned a priority in order for them to be ranked. Since it is possible for a stop to be assigned multiple messages it is important to ensure priority is given.<br><br>Priorities are between 1 and 10 (where 1 is the highest priority). By default the message priority is set to 3. | Real time |
| MessageText | N/A | The text of the message. This should be displayed to the public. | Real time |
| MessageType | N/A | Messages are assigned a type. This is predominantly in order to define how they should be displayed on on-street signs, however can be used to alter display on other devices.<br><br>0: “Normal”,<br>1: “Special”,<br>2: “Full Matrix” | Real time |
| MessageUUID | N/A | This is the unique identifier of the | Real time |

| | | | |
|---|---|---|---|
| | | flexible message. | |
| RegistrationNumber | N/A | The registration number of the vehicle that the prediction belongs to.<br><br>Any registration numbers containing the following values should be excluded from any selection or publication.<br><br>Prefixed with 'X_': These vehicles have been decommissioned from the London bus fleet.<br><br>'NEW' within the first five characters: These are placeholders for new vehicles soon to enter the London bus fleet.<br><br>These should not appear in the prediction array as they are not active vehicles, if they do then please contact us via developers@tfl.gov.uk. | Reference |
| ResponseType | N/A | This indicates the response array type. There are 5 different response arrays:<br><br>0: "Stop array"<br><br>1: "Prediction array"<br><br>2: "Flexible Message array"<br><br>3: "Baseversion array"<br><br>4: "URA Version array" | Reference |
| StartTime | N/A | This is the time at which the flexible message becomes valid. It should not be displayed to the public before this time.<br><br>It is an absolute time in UTC as per Unix Epoch (in milliseconds). To convert to UTC Epoch divide by 1000. | Real time |
| StopID | Stop_Code_LBSL | This is the alphanumeric identifier of a bus stop or pier used by LBSL. | Reference |

| | | | |
|---|---|---|---|
| | | It SHOULD NOT be displayed to the public. | |
| StopCode1 | SMS_Code | This is the public code for the bus stop or pier. It is displayed on the bus stop flag or pier, the Countdown website and should be used for all public facing applications.<br><br>This field relates to the NaPTAN NaptanCode without the<br><br>area code. The area code for London is “1”. As an example, the bus stop code for Southwark Station heading northbound is: 77293. The NaPTAN NaptanCode for this stop is therefore 177293 | Reference |
| StopCode2 | NaPTAN_Code | This is the unique national identifier of the bus stop or pier – i.e. the NaPTAN AtcoCode as defined in the DfT NaPTAN data model. | Reference |
| StopPointIndicator | Point_Letter | The letter(s) that are displayed on top of the bus stop flag (e.g. SA). These are used to help passengers easily identify a bus stop or pier from others in the locality.<br><br>It should be noted that not all bus stops or piers are assigned a point letter and that stop point indicators are not unique. | Reference |
| StopPointName | Stop_Name | The name of the bus stop / pier. | Reference |
| StopPointState | N/A | The different stop states and their definitions are provided below:<br><br>0: “Open”: Bus stop or pier is being served as usual<br><br>1: “Temporarily Closed” : Vehicles are not serving the stop but may be serving a nearby bus stop or pier, predictions may be available<br><br>2: “Closed” : Vehicles are not serving the stop. Stop should display the closed message and | Real time |

| | | | |
|---|---|---|---|
| | | **predictions should not be shown**.<br><br>3: “Suspended” : Vehicles are not serving the stop. Stop should display the closed message and **predictions should not be shown**. (On street signs may not show any messages in this scenario) | |
| StopPointType | Stop_Type | Indicates the type of stop as categorized by TfL. The full list of these stop point types is available in Appendix A. | Reference |
| Towards | Towards | Identifies the primary location(s) that are visited by routes serving a stop. The ‘towards’ text relates to the stop and, for buses, is displayed on the bus stop flag on the street. | Reference |
| TripID | Journey_Idx | The identifier of the specific trip that the prediction is for. | Reference |
| VehicleID | N/A | The unique identifier of the vehicle. This is an internal identifier and should not be displayed to the public. | Reference |
| Version | | Either used as baseversion of the data (Baseversion array) or URA version (Version array) | Reference |
| VisitNumber | N/A | Indicates whether the prediction is for the first time the vehicle visits a stop on that trip. On some routes, where the vehicle ‘loops’ the same stop may be visited more than once. | Reference |

**Table 8 - Glossary of terms**

# Appendix A. Stop Point Types

TfL bus stops are assigned different types depending on their usage. The table below provides a reference for the stop point types. It should be noted that the API includes stop types that **should not** be displayed to the public. These are intentionally left in the data as they help to indicate the roads that are served by the bus route.

The table indicates which stop types should NOT be displayed to the public. It is expected that users of the data adhere to this in order to avoid confusion to passengers.

In order to determine which bus stops are currently served by one or more routes the static data that TfL make available should be consulted.

| Stop Point Type | Description | Public display? |
|---|---|---|
| STBR | Stop - Bus Request | Yes |
| STBC | Stop - Bus Compulsory | Yes |
| STZZ | Stop - Other | Yes |
| STBN | Stop - No Flag (Hail + Ride Time table) | Yes (note no physical stop at this location) |
| STBS | Stop - Live Bus Stand | Yes |
| STBE | Stop - Dead Bus Stand | No |
| STCC | Stop - Coach Compulsory | No |
| STTS | Stop - Taxi Stop | No |
| STSS | Stop - Bus Station Stop | Yes |
| STTP | Stop - Timing Point for CAESAR | No |
| STDM | Stop - Dummy (No Physical Stop) | No |
| STVA | Stop - Virtual - iBus Announce | Yes (note no physical stop at this location) |
| STCR | Stop - Coach Request | No |
| STDL | Stop - Bus Request Coach Req | No |
| STDJ | Stop - Bus Compulsory Coach Req | No |
| SHCP | Stop - Hail & Ride Comm Point Start | No |
| SHCE | Stop - Hail & Ride Comm Point End | No |
| SLRS | Stop London River Services | Yes. Should be used to provide River Bus information. |

**Table 9 - Stop Point Types**