Timetabled Vehicle Journey Service Reference Manual API

Document identity:	RM/API/TVJS
Revision:	PA4
Date:	2017-03-28

Timetabled Vehicle Journey Service Reference Manual API

This document is part of the definition and design of the Hogia PubTrans system. The design of the Hogia PubTrans system is the property of Hogia Public Transport Systems.

Confidentiality

This document is confidential and may only be distributed to parties outside of Hogia Public Transport Systems AB after signing a non-disclosure agreement.

Copyright

Revision History

Revision	Date	Update	Updated by
PA1	2017-03-13	New document, divided in two API:s	Emma Fernlof
PA2	2017-03-21	Updated name of service. Updated origin/primary destination details. Updated response example	P-A Partanen
PA3	2017-03-21	Changed optional on attributes to reflect DOI and implementation.	Johan Havås
PA4	2017-03-28	Added information about detection from stop point keys in use case. Updated URLs so they conform to implementation. Added explanation of what current journey means.	Johan Havås

Table of content

1 Introduction

The systems on-board often requires data about planned operation and about the current working and status of the vehicle collected by other systems and sources. The Timetabled Vehicle Journey Service serves this purpose.

1.1 Scope

This document describes the API for providing operational status and data to vehicles.

1.2 Usage

In section 2 the most common use cases are described. Start exploring the API's by finding your case. Here you will find what service to call and what data to use.

1.3 Client Requirements

Clients to the API must be tolerant to:

- That optional properties can be null or left out.
- That arrays can contain any number of elements including zero.
- That returned data can be extended with new properties without notice.

Client must handle returned status codes according to http standards.

The "current journey" that is returned in the response is a journey that ends at latest a configurable time before the request starts at earliest a configurable time after the request. If two journeys fulfill this criteria the latter is returned in the response. The same rule applies for which block is chosen when requesting all journeys in a block.

2 Use Cases

This section describes a few common use cases and how to utilise data from the APIs.

2.1 Detecting Arrival and Departures at Stops

The purpose with the API is to provide an on-board computer with data enough to correctly detect arrival and departures at stops.

2.1.1 Retrieving data

For this purpose, the Timetabled Vehicle Journey Service is used to retrieve the data for the current journey.

2.1.2 Using data

For each point in the Calls sequence, there are the following data available:

- *Sequence Number* that defines the expected order of passage.
- *Location* in latitude/longitude of the point.
- Optionally a *Detection* object with the following (all optional) data:
    - *Entering Distance*, which is the distance before the location regarded as within the point.
    - *Existing Distance*, which is the distance after the location regarded as within the point.
    - *Passing Direction* and *Passing Direction Max Deviation*, which limits the possible directions to pass this point.

It's up to the implementer to define an algorithm that correctly detects the arrivals and departures in correct sequence using the data available in the API.

All values needed to support the detection object is not available in PubTrans by default. This API supports fetching this information from stop point keys in the following format, and the persons that enters data into PubTrans should make sure this is satisfied.

DeviceName = ”STOP_DETECTION”
ParameterData = ”EnteringDistance=10;ExitingDistance=20;PassingDirectionMaxDeviation=45”

TypeCode = “ADD_INFO”

2.2 Displaying Destinations

The purpose of the API is to provide head signs and internal signs with data in order to display line number and destinations.

2.2.1 Retrieving data

For this purpose, the Timetabled Vehicle Journey Service is used to retrieve the data for the current journey. The on-board system must provide a journey identification (line/journey or a global journey id) and must be able to detect arrivals and departures at stops.

2.2.2 Using data

Destination data is provided per each stop, which means that the destination can change before each stop. This may differ from other solutions where there is a fixed destination for the whole journey.

For presenting destination, only data in the *Destination Display* object on each stop shall be used. The "line number" to present is the *Line Designation*, which may be alphanumeric. *Primary destination* is for displaying the destination text. Eventually a sub destination is available in *Secondary Destination*. Note that the secondary destination may have different meanings.

Some destination display devices cannot use the texts directly but uses prepared display imagens. This usually requires specific control codes to be sent to the device in order to select what line and destination should be displayed. If these codes are imported into PubTrans, they are provided in the *Keys* collection in the Destination Display object. What key to use and the provided control codes must be agreed in advance between the persons that configures the destination displays and the persons that enters data into PubTrans.'

2.3 Displaying Next and Following Stops

The purpose of the API is to provide a sequence of stop names to be displayed as the vehicle progresses on the service journey.

2.3.1 Retrieving data

For this purpose, the Timetabled Vehicle Journey Service is used to retrieve the data for the current journey. The on-board system must provide a journey identification (line/journey or a global journey id) and must be able to detect arrivals and departures at stops.

2.3.2 Using data

For next and following stops presentation the *Stop Point* object shall be used.

There are several different versions of stop names:

- The *Name* version is maximum 50 characters.
- The *Short Name* version is maximum 16 characters.

Which one to use depends on the capability of the displays, for example if scrolling is possible.

3 Available API calls

This section describes the available API calls and an overview over the response data. For details regarding response data, see 4 Data Model Reference. For examples of use cases, see

NOTE: The request URL must conform to IETF RFC 1738. Note that this also applies to all parameters.

3.1 Timetabled Vehicle Journey Service

Use to get a specific vehicle journey or vehicle journeys for a block or line.

3.1.1 Request

3.1.1.1 Variant with Line and Journey Numbers

`GET api/timetabledvehiclejourney/v1/{account}/lines/{line}/{journey}?key=F8D37CA5-2061-45EB-BD9C-B0D77B5613CA&includeCalls=true&estimatedStartTime=2017-03-28T05:34:00`

Parameter name	Meaning	Comment/example
`account`	Identifies the user/customer.	Provided by Hogia.
`line`	The technical line number	
`journey`	The journey number within the line.	
`key`	Necessary to access the service. Specific for each customer.	Configured in the service by Hogia.
`includeCalls`	Specifics if stop call information should be included or not.	Default is true.
`estimatedStart-Time`	Estimated start time of the journey.	Default is the time the request is received.

3.1.1.2 Variant with Global Journey Numbers

`GET api/timetabledvehiclejourney/v1/{account}/journeys/{journey}?key=F8D37CA5-2061-45EB-BD9C-B0D77B5613CA&includeCalls=true&estimatedStartTime=2017-03-28T05:34:00`

Parameter name	Meaning	Comment/example
`account`	Identifies the user/customer.	Provided by Hogia.
`journey`	The global journey number.	I.e. a PubTrans Vehicle Journey GID.
`key`	Necessary to access the service. Specific for each customer.	Configured in the service by Hogia.
`includeCalls`	Specifics if stop call information should be included or not.	Default is true.

Parameter name	Meaning	Comment/example
`estimatedStart-Time`	Estimated start time of the journey.	Default is the time the request is received.

NOTE: This variant can also be used to get dead runs.

3.1.1.3 Variant with Block

`GET api/timetabledvehiclejourney/v1/{account}/blocks/{block}?key=F8D37CA5-2061-45EB-BD9C-B0D77B5613CA&includeCalls=true&estimatedStartTime=2017-03-28T05:34:00`

Parameter name	Meaning	Comment/example
`account`	Identifies the user/customer.	Provided by Hogia.
`block`	The block number	
`key`	Necessary to access the service. Specific for each customer.	Configured in the service by Hogia.
`includeCalls`	Specifics if stop call information should be included or not.	Default is true.
`estimatedStart-Time`	Estimated start time of the block.	Default is the time the request is received.

NOTE: This variant returns all journeys, both vehicle journeys and deadruns within a given block number.

NOTE: The account used must have block access rights.

3.1.2 Response

The response contains a sequence of vehicle journeys. Normally one item is returned. If no matching vehicle journey is found, a http 404 is returned. For detailed information about return codes, see 3.1.2.2

3.1.2.1 Response Data Model

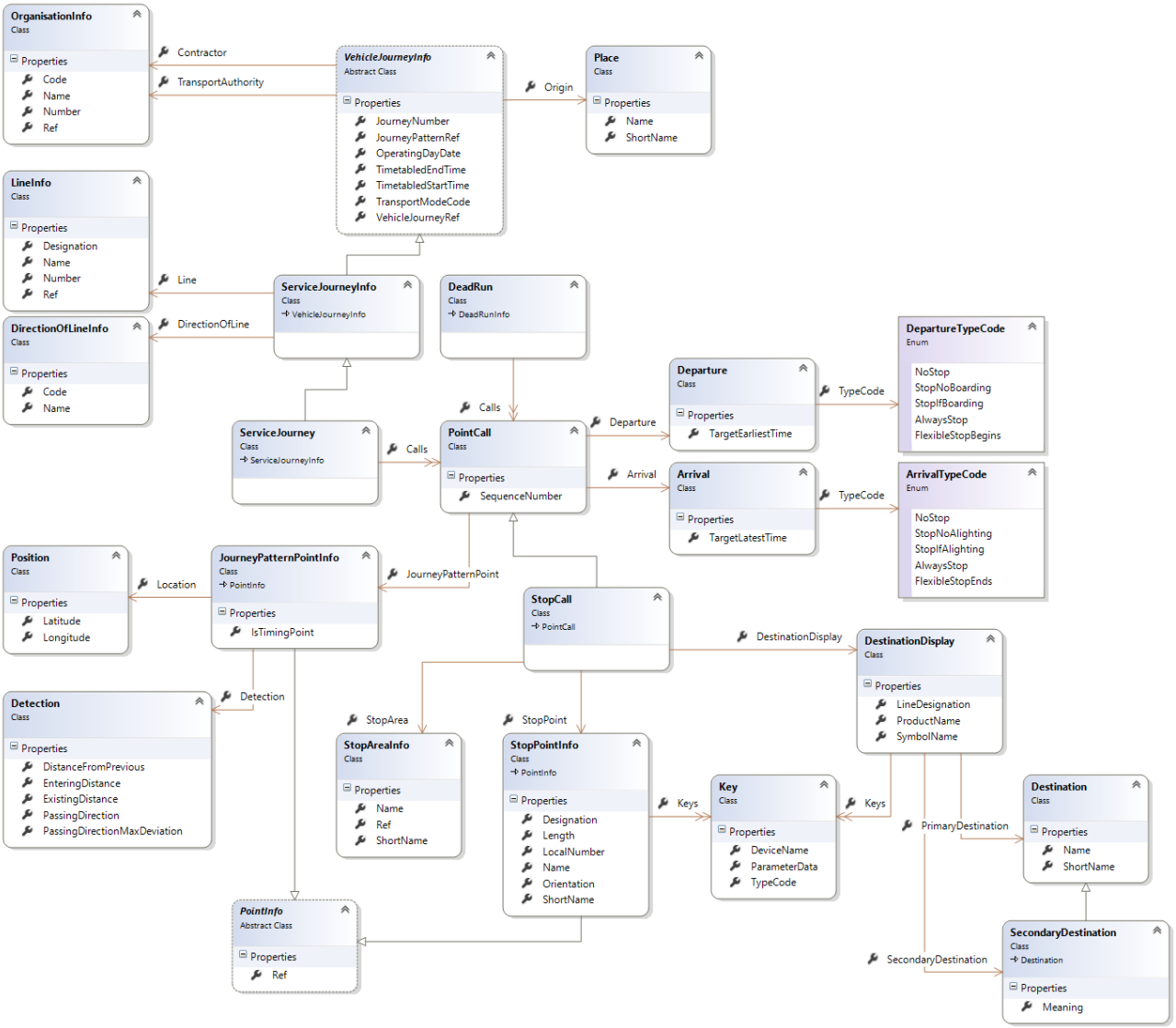

Fig. 1. Vehicle Journey Data

3.1.2.2 Response Example


```
[
 {
  "OperatingDayDate": "2017-03-28T00:00:00+02:00",
  "VehicleJourneyRef": "9015001095045168",
  "JourneyNumber": "45168",
  "JourneyPatternRef": "4010000445164382",
```

"Line": {
"Ref": "9011001095000000",
"Designation": "22",
"Number": "950",
"Name": "Ersätter Tvärbanan"
},
"TimedJourneyPatternRef": "4010000589435819",
"TransportModeCode": "BUS",
"DirectionOfLine": {
"Code": "1",
"Name": "Solna"
},
"TransportAuthority": {
"Ref": "9010001000000000",
"Code": "SL",
"Name": "Storstockholms Lokaltrafik",
"Number": "1"
},
"Contractor": {
"Ref": "9013001001500000",
"Code": "ARR",
"Name": "Arriva",
"Number": "15"
},
"TimetabledStartTime": "2017-03-28T08:48:00+02:00",
"TimetabledEndTime": "2017-03-28T09:02:00+02:00",
"Origin": {
"Name": "",
"ShortName": ""
},
"Calls": [
{

"SequenceNumber": 1,
"JourneyPatternPoint": {
"Ref": "9025001000010664",
"IsTimingPoint": true,
"Location": {
"Latitude": 59.3071932996622,
"Longitude": 18.1078200699196
},
"Detection": {
"EnteringDistance": 20.0,
"ExitingDistance": 10.0,
"PassingDirection": 345.0,
"PassingDirectionMaxDeviation": 15.0,
"DistanceFromPrevious": 791.0
}
},	},
"StopArea": {
"Ref": "9021001010665000",
"Name": "Sickla udde",
"ShortName": "Sickla udde"
},
"StopPoint": {
"Ref": "9022001010665005",
"Name": "Sickla udde",
"ShortName": "Sickla udde",
"Designation": "",
"LocalNumber": 5,
"Length": null,
"Orientation": null,
"Keys": [
{
"ParameterData": "Text=Sickla udde",

"TypeCode": "STOPNAME",
"DeviceName": "HAFAS"
},
{
"ParameterData": "Key=110665;LevelNight=3;LevelDay=5",
"TypeCode": "STOPNAME",
"DeviceName": "INIT_INTERIOR_PA"
},
{
"ParameterData": "Text=Sickla udde",
"TypeCode": "STOPNAME",
"DeviceName": "INIT_INTERIOR_SIGN"
}
]
},
"Departure": {
"TargetEarliestTime": "2017-03-28T08:50:00+02:00",
"DepartureType": "STOP_IF_BOARDING"
},
"DestinationDisplay": {
"ProductName": "",
"SymbolName": "",
"LineDesignation": "22B",
"PrimaryDestination": {
"Name": "Gullmarsplan",
"ShortName": ""
},
"SecondaryDestination": {
"Name": "",
"ShortName": "",
"Meaning": "UNKNOWN"
},

```
    "Keys": [
     {
      "ParameterData": "Destination=1141",
      "TypeCode": "O_DESTIN",
      "DeviceName": "SIGN"
     }
    ]
   }
  },
  {
   "SequenceNumber": 2,
   "JourneyPatternPoint": {
    "Ref": "9025001000010679",
    "IsTimingPoint": false,
    "Location": {
     "Latitude": 59.3026033213799,
     "Longitude": 18.1039034144013
    },
    "Detection": {
     "EnteringDistance": 20.0,
     "ExistingDistance": 10.0,
     "PassingDirection": 345.0,
     "PassingDirectionMaxDeviation": 15.0,
     "DistanceFromPrevious": 791.0
    },
    "StopArea": {
     "Ref": "9021001010676000",
     "Name": "Sickla kaj",
     "ShortName": "Sickla kaj"
    },
    "StopPoint": {
     "Ref": "9022001010676002",
```

"Name": "Sickla kaj",
"ShortName": "Sickla kaj",
"Designation": "",
"LocalNumber": 2,
"Length": null,
"Orientation": null,
"Keys": [
{
"ParameterData": "Text=Sickla kaj",
"TypeCode": "STOPNAME",
"DeviceName": "HAFAS"
},
{
"ParameterData": "Key=110676;LevelNight=3;LevelDay=5",
"TypeCode": "STOPNAME",
"DeviceName": "INIT_INTERIOR_PA"
},
{
"ParameterData": "Text=Sickla kaj",
"TypeCode": "ADD_INFO",
"DeviceName": "INIT_INTERIOR_SIGN"
},
{
"ParameterData": "Text=Sickla kaj",
"TypeCode": "STOPNAME",
"DeviceName": "INIT_INTERIOR_SIGN"
},
{
"ParameterData": "Text=(S)",
"TypeCode": "ADD_INFO",
"DeviceName": "TOM"
}

```
    ]
   },
   "Arrival": {
    "TargetLatestTime": "2017-03-28T08:52:18+02:00",
    "ArrivalType": "STOP_IF_ALIGHTING"
   },
   "Departure": {
    "TargetEarliestTime": "2017-03-28T08:51:53+02:00",
    "DepartureType": "STOP_IF_BOARDING"
   },
   "DestinationDisplay": {
    "ProductName": "",
    "SymbolName": "",
    "LineDesignation": "22B",
    "PrimaryDestination": {
     "Name": "Gullmarsplan",
     "ShortName": ""
    },
    "SecondaryDestination": {
     "Name": "",
     "ShortName": "",
     "Meaning": "UNKNOWN"
    },
    "Keys": [
     {
      "ParameterData": "Destination=1141",
      "TypeCode": "O_DESTIN",
      "DeviceName": "SIGN"
     }
    ]
   }
  },
```

```
{
 "SequenceNumber": 3,
 "JourneyPatternPoint": {
  "Ref": "9025001000082001",
  "IsTimingPoint": true,
  "Location": {
   "Latitude": 59.2987648412832,
   "Longitude": 18.081504988612
  },
  "DistanceFromPrevious": 791.0,

  "Detection": {
   "EnteringDistance": 20.0,
   "ExistingDistance": 10.0,
   "PassingDirection": 345.0,
   "PassingDirectionMaxDeviation": 15.0
   }
 },
 "StopArea": {
  "Ref": "9021001011725000",
  "Name": "Gullmarsplan",
  "ShortName": "Gullmarsplan"
 },
 "StopPoint": {
  "Ref": "9022001011725009",
  "Name": "Gullmarsplan",
  "ShortName": "Gullmarsplan",
  "Designation": "Z",
  "LocalNumber": 9,
  "Length": null,
  "Orientation": null,
  "Keys": [
```

```
    {
      "ParameterData": "Text=Gullmarsplan",
      "TypeCode": "STOPNAME",
      "DeviceName": "HAFAS"
    },
    {
      "ParameterData": "Key=111725;LevelNight=3;LevelDay=5",
      "TypeCode": "STOPNAME",
      "DeviceName": "INIT_INTERIOR_PA"
    },
    {
      "ParameterData": "Text=Gullmarsplan",
      "TypeCode": "STOPNAME",
      "DeviceName": "INIT_INTERIOR_SIGN"
    },
    {
      "ParameterData": "Text=Gullmarsplan",
      "TypeCode": "ADD_INFO",
      "DeviceName": "INIT_INTERIOR_SIGN"
    },
    {
      "ParameterData": "Text=(T,S)",
      "TypeCode": "ADD_INFO",
      "DeviceName": "TOM"
    }
  ]
},
"Arrival": {
  "TargetLatestTime": "2017-03-28T09:02:00+02:00",
  "ArrivalType": "STOP_IF_ALIGHTING"
},
"DestinationDisplay": {
```

```
      "ProductName": "",
      "SymbolName": "",
      "LineDesignation": "22B",
      "PrimaryDestination": {
        "Name": "Gullmarsplan",
        "ShortName": ""
      },
      "SecondaryDestination": {
        "Name": "",
        "ShortName": "",
        "Meaning": "UNKNOWN"
      },
      "Keys": [
        {
          "ParameterData": "Destination=1141",
          "TypeCode": "O_DESTIN",
          "DeviceName": "SIGN"
        }
      ]
    }
  }
 ]
}
```


3.2]Status Codes returned

The following http status codes will be returned on specific situations. In addition, any other official status code may be returned.

HTTP CODE	Comment
200 OK	Journey or block found and data returned.
400 Bad request	Indicates errors in request. The request will not be processed.
401 Unauthorized	Key not provided.

HTTP CODE	Comment
403 Forbidden	Key and account mismatch, or account not valid.
404 Not found.	No data for the requested journey or block is found.
500 Internal Server Error	An unexpected error condition was encountered in the server.
503 Service Unavailable	Service cannot respond at the moment.

4 Data Model Reference

This section contains details about each object returned from the API methods.

4.1 Vehicle Journey

4.1.1 Vehicle Journey Info

Field name	Meaning	Comment/example
$type	Object type, optional.	See 4.1.2 and 0.
OperatingDayDate	The scheduled date of the journey.	
VehicleJourneyRef	A unique identifier for the journey.	Usually PubTrans GID.
JourneyNumber	The journey number.	
JourneyPatternRef	A unique identifier for the journey pattern.	Usually PubTrans ID.
TransportModeCode	The transport mode.	"BUS", "TRAM" etc.
TransportAuthority	Information about the transport authority that provides the journey.	See 4.4.1.
Contractor	Information about the operator that is contracted to operate the journey.	Optional. See 4.4.2.
TimetabledStartTime	The date and time when the journey is planned to start.	
TimetabledEndTime	The date and time when the journey is planned to end.	
Origin	The place where the journey comes from.	If origin name is missing in PubTrans then name and shortname of the first stopcall is returned.

4.1.2 Service Journey Info

Field name	Meaning	Comment/example
$type	Object type.	Optional. Constant value: `ServiceJourneyInfo` or `ServiceJourney`
All the fields from Vehicle Journey, see 4.1.1		
Line	Information about the journey's line.	See 4.2.14.2 .
DirectionOfLine	Information about the journey's direction on the line.	See 4.2.2.

Field name	Meaning	Comment/example
Calls	Array with points that shall be called at during the journey.	Only present for $type = `ServiceJourney` and then at least two items.

4.1.3 Dead Run Info

Field name	Meaning	Comment/example
$type	Object type.	Optional. Constant value: `DeadRunInfo` or `DeadRun`
All the fields from Vehicle Journey, see 4.1.1		
Calls	Array with points that shall be called at during the journey.	Only present for $type = `DeadRun` and then at least two items.

4.2 Lines and Destinations

4.2.1 Line

Field name	Meaning	Comment/example
$type	Optional. Object type.	Optional. Constant value: `LineInfo`
Ref	A unique identifier for the journey.	Usually PubTrans GID.
Designation	The public line number displayed to passengers.	Note: this value can be alphanumeric!
Number	Technical line number.	
Name	Name of line or stretch description.	Optional.

4.2.2 Direction of Line

Field name	Meaning	Comment/example
$type	Object type.	Optional. Constant value: `DirectionOfLineInfo`
Code	A value that is unique per direction.	Usually “1” or “2”.
Name	Name of direction.	Optional.

4.2.3 Destination Display

Field name	Meaning	Comment/example
$type	Object type, optional.	Constant value: `DestinationDisplay`

Field name	Meaning	Comment/example
ProductName	Name of public transport product.	Optional.
SymbolName	Controls any symbols in display.	Optional.
LineDesignation	Displayed line number.	
PrimaryDestination	Names of primary destination.	
SecondaryDestination	Names of secondary destination.	Optional.
MeaningOfSecondaryDestination	E = End station/terminus of route. M = Message V = Via T = Transfer at this place to continue to Primary Destination C = Continues to this place after transferring at Primary Destination U = Unknown	Optional.
DisplayKeys	Device specific information.	Optional. See 4.5.1.

4.3 Calls and Times

4.3.1 Point Call

Field name	Meaning	Comment/example
$type	Object type.	Optional. Constant value: `PointCall`
SequenceNumber	The order of the call on the vehicle journey	Starts from 1. There are no holes in the sequence.
JourneyPatternPoint	The point to call.	See 4.3.2.
Arrival	Arrival time and type.	See 4.3.6.
Departure	Departure time and type.	See 4.3.7.

4.3.2 Journey Pattern Point Info

Field name	Meaning	Comment/example
$type	Object type.	Optional. Constant value: `JourneyPatternPointInfo`
Ref	Id of journey pattern point.	
IsTimingPoint	Boolean that indicates if this is a point where the driver should respect the departure time.	

Field name	Meaning	Comment/example
Location	The position of the point.	Optional. Always in Latitude/Longitude.
Detection	Additional data for point detection.	Optional. See 4.3.3.
DistanceFromPrevious	Meters from previous point.	Optional. First is always zero.

4.3.3 Detection

Field name	Meaning	Comment/example
$type	Object type.	Optional. Constant value: `Detection`
EnteringDistance	Meters before location regarded as within the point.	Optional.
ExitingDistance	Meters after location regarded as within the point.	Optional.
PassingDirection	Defines angles for restricting passing the point in certain direction.	Optional.
PassingDirectionMaxDeviation		

4.3.4 Stop Point Info

Field name	Meaning	Comment/example
$type	Object type.	Optional. Constant value: `StopPointInfo`
Ref	Id of the stop point.	
Designation	Track, gate, stop etc. as shown to the public.	Optional. This is the for local orientation within a stop area, bus terminal or station.
Length	The stops capacity in meters.	Optional.
Orientation	The heading of vehicles at the stop.	Optional. Degrees.
LocalNumber	The display order of stops within a stop area.	
Name	Full name.	Optional. Max 50 characters. When missing stop area name is used.
Short Name	Shortened name.	Optional. Max 16 characters. When missing stop area short name is used.

4.3.5 Stop Area Info

Field name	Meaning	Comment/example
$type	Object type.	Optional. Constant value: `StopAreaInfo`
Ref	Id of the stop area.	
Name	Full name.	Max 50 characters.
Short Name	Shortened name.	Optional. Max 16 characters.

4.3.6 Arrival

Field name	Meaning	Comment/example
$type	Object type.	Optional. Constant value: `Arrival`
TargetLatestTime	The latest arrival time expected.	Arrival after this time is LATE.
DepartureTypeCode	Usage of the departure.	*NoStop* and *StopNoBoarding* means that the departure should NOT be presented public.

4.3.7 Departure

Field name	Meaning	Comment/example
$type	Object type.	Optional. Constant value: `Arrival`.
TargetEarliestTime	The earliest permitted departure time.	Departure before this time is EARLY.
DepartureTypeCode	Usage of the departure.	*NoStop* and *StopNoAlighting* means that the arrival should NOT be presented public.

4.4 Organisational

4.4.1 Transport Authority

Field name	Meaning	Comment/example
$type	Optional. Object type.	Optional. Constant value: `OrganisationalInfo`.
Ref	A unique identifier.	Usually PubTrans GID.
Code	Short abbreviation.	
Name	Full name	

4.4.2 Contractor

Field name	Meaning	Comment/example
$type	Optional. Object type.	Optional. Constant value: `OrganisationalInfo`.

Field name	Meaning	Comment/example
Ref	A unique identifier.	Usually PubTrans GID.
Code	Short abbreviation.	
Number	The contractor number.	
Name	Full name	

4.5 Other

4.5.1 DisplayKey

Field name	Meaning	Comment/example
$type	Object type, optional.	Constant value: **Key**.
DeviceName	Name of devices for which this key applies.	
TypeCode	Name of data type.	
ParameterData	The data.	This data is customer specific.

5 References

Document	Description
HTTP CODES	https://en.wikipedia.org/wiki/List_of_HTTP_status_codes
RFC 1738	https://www.ietf.org/rfc/rfc1738.txt