Hogia
Public Transport Systems

PubTrans Web Services 2
Reference Manual

Document identity:	RM-PWS/2
Revision:	AB
Date:	2022-08-22

PubTrans Web Services 2 Reference Manual

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.

Copyright

Revision History

Revision	Version	Date	Update	Updated by
A	2.0	2008-04-22	Preliminary version for external usage	Stefan Fjällemark
B	2.0	2008-11-14	Release 2.0.0	Jimmy Vilén
C	2.0	2009-04-21	Release 2.0.1	Jimmy Vilén
D	2.1	2009-08-20	Release 2.1.0	Jimmy Vilén
E	2.2	2009-09-21	Release 2.2.0	Jimmy Vilén
F	2.2	2009-12-04	Release 2.2.0 Patch 02	Jimmy Vilén
G	2.2	2010-03-26	Release 2.2.4	Jimmy Vilén
H	2.3	2010-03-29	Release 2.3.0	Jimmy Vilén
J	2.3	2010-04-21	Release 2.3.2	Jimmy Vilén
K	2.3	2010-04-29	Release 2.3.3	Jimmy Vilén
L	2.5	2010-11-19	Release 2.5.0	Jimmy Vilén
M	2.6	2010-12-13	Release 2.6.0	Jimmy Vilén
N	2.6	2011-02-10	Updated description for GetDirectJourneysBetweenStops and GetCurrentDeviations.	Jimmy Vilén
O	2.7	2011-08-25	Added Get Contract With Lines	Martin Jonsson
P	2.7	2012-01-12	Updated prerequisite with view info	Martin Jonsson
Q	2.7	2012-03-15	Updated pwsConfig and added deadlock chapter Release 2.7.3	Martin Jonsson
R	2.8	2013-04-17	Added methods GetDetailedStopsInfo, GetSiteInfo and modified result tables Zone and LineInContract. Release 2.8.0	Andreas Olsson
S	2.9	2014-02-21	Added table Line Destination at Stop to method GetStopsWithLines in StopDataService.	Kristoffer Rasmussen

T	2.10	2014-08-06	Added ExposedInPrintMediaYesNo and DynamicMediaInformPassengersConditionCode to DatedServiceJourney response object. Updated database installation instructions (section 3.6.5)	Kristoffer Rasmussen
U	2.10	2015-09-10	Corrections on response objects: Section 18.3 DatedVehicleJourney: Origin is optional, i.e. can be NULL if information is missing in DOI. Section 18.3-18.5 DatedVehicleJourney /DatedDeparture/DatedArrival: clarification on vehicle journey state and advance order condition. General corrections in document – some sections were duplicated in previous revision. Document identity changed to “RM”.	Kristoffer Rasmussen
V	2.10	2016-06-03	Added information on Group Of Lines deducted from other PubTrans data (section (16.2 and 16.3).	Kristoffer Rasmussen
W	2.10	2017-06-09	Changes to description of parameters for GetCurrentDeviations. Changed DatedServiceJourney.ProductType to ProductCode to reflect current implementation and name standards. Overall document review. Corrections and clarifications.	Tobias Wärre
X	2.10	2018-05-21	Changes in installation	Anders Rydell
Y	2.11	2018-12-18	Added Method GetLineVariants and return values. Added *Definitions* chapter with *Global Identifiers*.	Tobias Wärre
Z	2.12	2021-04-19	Updated installation instructions. Updated Stop Point Response. Updated Line At Stop Response	Daniel Schlicker
AA	2.12.1	2022-03-03	Corrections: - TransportAuthority response object - IsControlledByOrganisationalUnitId is nullable - LineDesignation, 1-8 characters in DatedDeparture - LineDesignation, 1-8 characters and missing ExistsFromDate	Benjamin Block

			and ExistsUptoDate in OrganisationalUnit Additions: - LastModifiedUtcDateTime to JourneyPatternPointInZone - Description for ExistsFromDate and ExistsUptoDate in LineInZone	
AB	2.12.3	2022-08-22	Removed DisplayOrderNumber. Removed IsODirectionOfLineGid Added descriptions. Removed PubTrans 4 info	Benjamin Block/Ted Roserud

Release Notes

Release notes are found in document RN-PWS/2.

Table of Content

1 Introduction 7
1.1 Background..7
1.2 Terminology 7

2 Introduction 8
2.1 Overview..8
2.2 Services..8
2.3 Licensing 9
2.4 Compatibility..9
2.5 Recommended Usage 10
2.6 Interoperability...11

3 Operating Guidelines 13
3.1 Operating Requirements....................................13
3.2 Hosting Environment 13
3.3 Security...13
3.4 Monitoring 14
3.5 Logging 14
3.6 Installation 15

4 User Guidelines.. 24
4.1 Accessing the Web Services................................24
4.2 Parameters in Web Methods 24
4.3 Result Sets 26
4.4 Error Handling 27

Part 1: Service Contracts 29

5 Organisational Data Service 29
5.1 Get Contractor Info..29
5.2 Get Contractors with Operated Lines 30
5.3 Get Service Function for Line.............................31
5.4 Get Contracts With Lines....................................32

6 Topology Data Service 34
6.1 Get Zones 34

7 Stop Data Service 35
7.1 Get Stops Common for Line 35
7.2 Get Stops Info 35
7.3 Get Stops with Lines..36
7.4 Get Detailed Stop Info 37
7.5 Get Site Info 39

8 Line Data Service 40
8.1 Get Lines 40
8.2 Get Lines with Stops..41
8.3 Get Stops On Line 42
8.4 Get Stops On Direction Of Line 43
8.5 Get Stops On Service Journey.............................43
8.6 Get Lines by Direction Of Line Gid 44
8.7 Get Lines by Gid 44
8.8 Get Lines by Line Designation 45
8.9 Get Line Variants 46

9 Timetable Data Service.................................... 48
9.1 Get Timetable at Stop 48
9.2 Get Timetable for Service Journey.................... 48

10 General Message Service 50
10.1 Get Current Deviations.................................... 50

11 Stop Monitoring Service 52
11.1 Get Departures at Stop.................................... 52
11.2 Get Direct Journeys between Stops 53
11.3 Get Calls at Stop 54

12 Vehicle Monitoring Service 56
12.1 Get Calls for Service Journey 56
12.2 Get Call 57

Part 2: Response Reference 59

13 Organisational data 59
13.1 Organisational Unit... 59
13.2 Transport Authority.. 60
13.3 Contractor... 60
13.4 Contract 61
13.5 Line In Contract 61
13.6 Service Function For Organisational Unit..... 62
13.7 Communication Address For Organisational Unit.. 63

14 Topology 64
14.1 Zone.. 64
14.2 Line in Zone 64

15 Stops 66
15.1 Stop Area 66
15.2 Stop Point 66
15.3 Line at Stop.. 67
15.4 Line Destination at Stop 68
15.5 Site 69
15.6 Site Neighbouring Stop Area 70
15.7 Journey Pattern Point In Zone 70

16 Lines 71
16.1 Line... 71
16.2 Group Of Lines 71
16.3 Line In Group Of Lines................................... 73
16.4 Direction Of Line 74
16.5 Operated Lines... 74
16.6 Stop on Line 75
16.7 Line Variants.. 76

17 General messages 78
17.1 Deviation Message 78
17.2 Deviation Message Variant 78

18 Arrivals and Departures 80
18.1 Departure.. 80
18.2 Direct Journeys between Stops 84
18.3 Dated Service Journey..................................... 86
18.4 Dated Departure 88
18.5 Dated Arrival 91

19 Deviations 93
19.1 Deviation Message Version 93
19.2 Deviation Message Variant.............................93
19.3 Service Journey Deviation...............................94
19.4 Arrival Deviation 94
19.5 Departure Deviation..94
19.6 Publication Scope...95
19.7 Target Audience...95
20 Definitions... 96
20.1 Global Identifiers 96
21 References 105

1 Introduction

This document describes the Hogia PubTrans® *Web Services* (PWS), a simple way to exchange data with the PubTrans system.

1.1 Background

Over the last couple of years, Web services have expanded to become more popular with application developers — and for good reason. Web services technology represents an important way for businesses to communicate with each other and with clients as well. Unlike traditional client/server models, such as a Web server or Web page system, Web services do not provide the user with a GUI. Instead, Web services share business logic, data and processes through a programmatic interface across a network. The applications interface with each other, not with the users. Developers can then add the Web service to a GUI (such as a Web page or an executable program) to offer specific functionality to users.

The global acceptance of Web services, which includes standard protocols for application-to-application communication, has changed software development. For example, the functions that Web services now provide include security, distributed transaction coordination, and reliable communication. The benefits of the changes in Web services should be reflected in the tools and technologies that developers use.

Hogia Public Transport Systems has identified web services as a key technology in the future system integration scenarios. With this in mind Hogia Public Transport Systems offers access to data through web services.

1.2 Terminology

The terminology used in this document is based on common definitions for the PubTrans® system. This section highlights some of the most important aspects.

Term	Definition
Fault	A mechanism (SOAP faults) for handling errors/exceptions when calling web services.
GID	A 16-digit number that identifies a business entity in PubTrans. For more information, see [IS-PT/I/DOI/4].
Message	A message is a self-contained unit of data that may consist of several parts, including a body and headers.
Notification	Making a call-back to a client.
Service	A service is a construct that exposes one or more endpoints, with each endpoint exposing one or more service operations.

2 Introduction

The PubTrans system is designed to be an open system with several public interfaces. These interfaces expose an advanced model for managing data about public transportation. PubTrans® Web Services (PWS) is a simplified access to PubTrans® Public Interfaces. PWS is implemented as a layer upon the public interfaces. The purpose of PWS is to simplify integration of third party applications by providing a number of methods for the most common tasks and to allow for custom extensions.

2.1 Overview

PubTrans® Web Services offers a simple way to retrieve data through a number of web services. The web services are designed to meet the requirements of the most common data access scenarios, which mean that direct access to PubTrans® Public Interfaces in many cases can be replaced by using the web services instead.

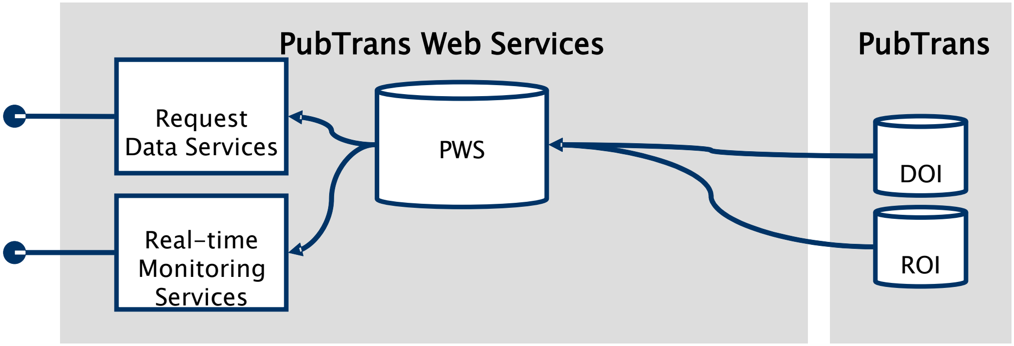


Fig. 1. PWS system structure

PWS supports the following type of services:

- **Request Data Services**, which provides data about planned operation, i.e. data from DOI.
- **Real-time Monitoring Services**, which provides real-time data about current situation, i.e. data from ROI.

2.2 Services

A *logical service* is a grouping of web methods with similar functionality. A logical service is the licence unit in PWS. Within the different types of services there may be one or several logical services.

2.2.1 Request Data Services

The request data services gets data from Data Output Interface (DOI):

- **Organisational Data Service**, which provides data business participants, their organisational structure, and where special responsibilities/functions are placed in the organisation.
- **Topology Data Service**, which provides data about zones.
- **Stop Data Service**, which provides data about stops and stations.

- **Line Data Service**, which provides data about lines, group of lines, and line operators.
- **Timetable Data Service**, which provides data about planned operation (schedules).

2.2.2 Real-time Monitoring Services

The request data services gets data from Real-time Output Interface (ROI):

- **General Message Service**, which provides information about general information messages affecting stops, vehicles or services, including deviations.
- **Stop Monitoring Service**, which provides data about forecasted and actual arrival and departures, and service messages.
- **Vehicle Monitoring Service**, which provides data about forecasted and actual arrival and departures, and service messages.

2.3 Licensing

In order to use PubTrans® Web Services, you must obtain a user licence from Hogia Public Transport Systems. Licences for developing and test purposes can be obtained for a smaller administrative fee.

A license for production usage requires an agreement of the extent of usage, expressed as maximum number of requests or notifications per hour per service and client system. The licence fee is based on these factors and PWS will limit the access to the web services accordingly. If the usage exceeds the licensed levels, PWS will respond with a license exception fault to the client.

IMPORTANT NOTE:

License parameters shall be configured to the agreed settings and limits. If these settings and limits are altered to values that exceed the licensed usage, Hogia Public Transport Systems reserve the right to charge one year license and support fees for the extended usage.

2.4 Compatibility

PubTrans® Web Services is available for PubTrans 5.

2.4.1 Unsupported services and methods

The table below shows which methods that are supported on which platform from a specific release.

Service	Method	Target	PWS/2 PT5
General Message Service	Get Current Deviations	ROI	2.1.0
Line Data Service	Get Lines	DOI	2.0.0
	Get Lines by Direction of Line Gid	DOI	n/a
	Get Lines by Gid	DOI	n/a
	Get Lines by Line Designation	DOI	n/a
	Get Lines With Stops	DOI	2.0.0

Service	**Method**	**Target**	**PWS/2 PT5**
	Get Stops On Line	DOI	2.0.0
	Get Stops On Direction Of Line	DOI	2.0.0
	Get Stops On Service Journey	DOI	2.0.0
	Get Line Variants	DOI	2.11.0
Organisational Data Service	Get Contractor Info	DOI	2.0.0
	Get Service Function For Line	DOI	2.1.0
Stop Data Service	Get Stops Common for Lines	DOI	2.1.0
	Get Stops Info	DOI	2.0.0
	Get Stops with Lines	DOI	2.0.0
	Get Detailed Stops Info	DOI	2.8.0
	Set Site Info	DOI	2.8.0
Stop Monitoring Service	Get Calls At Stop	ROI	2.0.0
	Get Departures at Stop	ROI	n/a
	Get Direct Journeys between Stops	DOI	2.0.0
	Get Direct Journeys between Stops	ROI	n/a
Timetable Data Service	Get Timetable At Stop	DOI	2.0.0
	Get Timetable for Service Journey	DOI	2.0.0
Topology Data Service	Get Zones	DOI	2.0.0
Vehicle Monitoring Service	Get Calls for Service Journey	ROI	2.0.0
	Get Call	DOI/ROI	2.0.0
Internal procedures	Create Stop With Line Temp Table	DOI	2.0.0
	Create Stop With Line Validity Temp Table	DOI	n/a

2.5 Recommended Usage

The web services do not replace the PubTrans® public interfaces. There are several reasons for still using the public interfaces:

- When requesting large amount of data, for example providing complete updates of journey planners or other kind of systems that require timetable updates at regular intervals.
- When performance is a critical issue. Web services impose an overhead that can become critical when the number of requests increase, for instance in large real-time systems.

- When complex custom transformation of data is required to adapt data for specific application needs, which require development of customised data adapters.

2.5.1 Types of Web Methods

Each web method falls into one of the following categories:

- Batch usage: the web method is intended for daily batch updates, which means that you can cache the result, valid at least the current day. This type of method returns more data, which means that the response time can be significant (due to data transfer time), and therefore not suitable for interactive use.
- Interactive usage: the web method is intended for frequent and fast data retrieval. This type of method returns a small amount of data, and often snapshot of real time data. Data from an interactive only method cannot be cached.

The type of usage is specified for each web method later in this document.

2.6 Interoperability

PWS implementation is based on Windows Communication Foundation (WCF) in .NET Framework version 3.5. This means that PWS supports the common standards for web service interoperability that is supported by WCF.

2.6.1 Windows Communication Foundation

Windows Communication Foundation (WCF) is a set of .NET technologies for building and running connected systems. It is a new breed of communications infrastructure built around the Web services architecture. Advanced Web services support in WCF provides secure, reliable, and transacted messaging along with interoperability. The service-oriented programming model of WCF is built on the Microsoft .NET Framework and simplifies development of connected systems.

WCF unifies a broad array of distributed systems capabilities in a composable and extensible architecture, spanning transports, security systems, messaging patterns, encodings, network topologies, and hosting models.

WCF provides support for web services infrastructure protocols through the concept of *channels* and web services application protocols through the concepts of *contracts*. Interoperability for application protocols is accomplished through XML Schema description language 1.0 (XSD) and Web Services Description Language (WSDL) 1.1.

Infrastructure protocols interoperability is provided by the WS-* specifications family. WCF channels provide support for a number of WS-* infrastructure protocols. WCF channels are configured via binding elements. See http://msdn2.microsoft.com/en-us/library/ms734776.aspx for the full list of the WS-* infrastructure protocols implemented by various WCF binding elements.

2.6.2 Minimum Interoperability Requirements

PWS 2.0 is based on the WCF version that is a part of the .NET Framework release 3.5. As a minimum, a client must at least conform to **WS-I Basic Profile 1.1**, which corresponds to the **BasicHttpBinding** in WCF.

Web methods in the request data service returns a data set containing one or several tables. The returned data set are a serialised .NET **System.Data.DataSet**. Clients must be able to handle this data type.

2.6.3 Interoperability with .NET

Clients to PWS 2.0 and later and that are implemented using .NET Framework 3.5 can take full advantage of Windows Communication Foundation (WCF) for security, transport and encoding.

Web methods in the request data service returns a data set of type **System.Data.DataSet** with the *response tables* as **System.Data.DataTable** objects in the data sets **Tables** collection. When there is references between the response tables, it is possible to programmatically create a master – detail linking using a **System.Data.DataRelation**. This will simplify the navigation through the returned data. Optionally, PWS can be configured to add the data relations in the result set. For additional information on how to use **DataSet**, **DataTable** and **DataRelation**, see Microsoft's documentation.

2.6.4 Interoperability with Java

For PWS 2.0 client development in Java, we refer to project Tango. Project Tango develops and evolves the Web Services Interoperability Technologies (WSIT) that enables interoperability between the Java platform and Windows Communication Foundation (WCF). See https://wsit.dev.java.net/ for more information.

3 Operating Guidelines

This chapter is targeted to system administrators that installs and operates PWS.

3.1 Operating Requirements

3.1.1 Software Requirements

PWS 2 runs under operating system versions Windows XP, Windows Vista, and Windows 2003 Server or later. .NET Framework 4.8 must be installed on the machine running PWS 2.

PWS 2 requires DOI and ROI replicas running on Microsoft SQL Server 2005 or later.

If IIS shall be used for hosting, IIS 7 or later is recommended (see section 3.2 below for details about hosting environments).

3.1.2 Hardware Requirements

Hardware requirements for running PWS are depending on how PWS will be used. In general, the PWS service itself will not be the bottleneck. The most important factors are the performance of the database server hosting the DOI and ROI databases, and the throughput between service and client.

3.2 Hosting Environment

PWS can be hosted by Internet Information Services (IIS), Windows Process Activation Service (WAS), a Windows service, or by a managed application (this option is often referred to as self hosting) on Windows Vista, Windows XP or Windows Server 2003 or later.

It is important to note that running a service or any extension from an un-trusted host compromises security.

The current version of PWS is delivered as a service host application, which means that it runs as a Windows service. If other hosting environment is desired, please contact Hogia Public Transport Systems. For additional information of hosting environments, see http://msdn2.microsoft.com/en-us/library/ms729846.aspx.

3.3 Security

3.3.1 Windows Account

As default, PWS is installed to run under the **Local System** account. The Network Service account is a special built-in account that has reduced privileges similar to an authenticated user account and it does not have a password that an administrator needs to manage.

You can run PWS under any another account.

3.3.2 Database Access

PWS must be granted INSERT, UPDATE and DELETE permissions to the DOI and ROI databases.

If you have configured SQL Server to require Integrated Windows Security, the account that PWS runs under must granted access to the DOI and ROI databases.

3.3.3 Service Access

PWS security can be configured for options supported by Windows Communication Foundation (WCF). If your web service requires special security, please contact Hogia Public Transport Systems.

3.4 Monitoring

PWS can be monitored by tools that can read Windows Performance Counters (WPC). The following WPC's are available:

- Number of calls, total number of calls.
- Calls duration, average amount of time for each call.
- Calls failed total number of calls that fails.
- Calls outstanding, number of active calls.
- Security calls not authorized, number of calls that didn't authorize.

These counters can also be monitored on service level, endpoint level and method level. There are more counters that can be monitored included in WCF. For additional information of WPC's with WCF, see http://msdn.microsoft.com/en-us/library/ms735098.aspx.

PWS is using work tables for increasing the performance on some complex queries. To monitor that these tables are up to date and not empty there is a service that uses REST-technology.

It can be called by a HTTP GET on **http://*server*:*port*/Pws/PwsDiagnosticsService/Status**. The result will look something like this:

```
<GetPwsDiagnosticsResponse xmlns="http://schemas.datacontract.org/2004/07/Hogia.PubTrans.WebServices.ResponseData" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <StopWithLine i:nil="true" />
    <StopWithLineValidity2>
    <MaxDate>2009-06-13T00:00:00</MaxDate>
    <MinDate>2008-09-12T00:00:00</MinDate>
    <Rows>12091</Rows>
  </StopWithLineValidity2>
</GetPwsDiagnosticsResponse>
```

3.5 Logging

PWS can log events in the following levels:

- **Fatal**: Errors of a type that stops PWS to work correctly.
- **Error**: Operational errors returned as SOAP faults to the client.
- **Warning**: Other events that can indicate problems, but does not cause faults sent to the client.
- **Info**: For each call of a web method: is name, time of call and client name.

- **Debug**: For each call of a web method: all parameters with name and value, time of processing the request, time spend in the database processing the request, the result set and/or return status.

During normal operation, logging shall be made on the **Warning** level, which means that **Info** and **Debug** messages shall not be logged. For analysis of usage of PWS, the **Info** level can be used under normal operation. The **Debug** level may only be used for analysing technical problems.

There is also internal logging and tracing in WCF that can be turned on in the configuration file for PWS. For additional information of message logging and tracing, see http://msdn.microsoft.com/en-us/library/ms731055.aspx.

3.6 Installation

The current release of PWS is delivered as a self hosted Windows Service. However, PWS can be run in other hosting environments, such as Internet Information Server (IIS) or Windows Activation Service (WAS). Please, contact Hogia Public Transport Systems if other hosting models are required.

3.6.1 Prerequisites

In order to install PubTrans Web Services, the following prerequisites must be fulfilled:

- To install PWS, you must have access to and the right to create objects in a DOI and ROI database. PWS uses DOI 4 and ROI 3 databases. Note that RDA must be installed and configured to save the ROI XML-stream data into a ROI 3-compliant database.
- DOI and ROI replicas must have their views installed.
- You must decide on which Windows account you shall use. The default installation runs the service under the **Local System** account. After installation, you can change this to any other account.
- You must decide the settings for security level, type of transport, type of encoding and port number to use for the web services. Note that the same service can be started with different settings for security, transport and encoding on different ports. The default is that no services are enabled.
- You must have Microsoft .NET Framework 4.8 installed on the server on which PWS shall run.

3.6.2 Installation

This section describes how to install PWS as a self hosted Windows service. The installation is made in the following steps:

1. Make sure all prerequisites are clear.
2. Run the Windows installer package **Pws Service Host Application 2.x.x. Setup.msi**.
3. Check the End User License Agreement if you accept the agreement, if so check the box, otherwise end the installation.
4. Click next and then change the location if needed, then click next.
5. Start the installation by clicking Install.
6. Click close and the installation is completed.
7. To install the database objects see 3.6.5 Install database objects.

8. To configure the application see 3.6.7.

3.6.3 Uninstallation

This section describes how to uninstall PWS. This is done when removing the application completely or just removing it before installing a newer version. The uninstallation is made in the following steps:

1. Start the uninstallation from `Add and Remove programs`.
2. Click Ok and the uninstallation is finishing.

3.6.4 Upgrading

When upgrading web services first uninstall the old version according to chapter 3.6.3 Uninstallation.

Copied configuration files will not be removed.

Install the new version according to chapter 3.6.2.

Always remember to check release notes for changes in configuration files after the old version you upgraded from.

You must also run the sql scripts against the database or databases, see 3.6.5 Install database objects.

3.6.5 Install database objects

To install the database object it is important that you have the right permission to create objects in the DOI and ROI database if there is one present.

The scripts to be installed is located in the folder where the web service is installed under a sub folder, `Sql`. Depending on what version of PubTrans, choose the scripts in the sub folder representing the correct version.

Note on database access: The installation script creates a default user “PWSUser” and a database role “PWSReader.. The user “PWSUser” is granted the role of “db_datareader” as well as “PWSReader”. If a different database user should be used to access the DOI/ROI databases, make sure that this user is granted both roles of “db_datareader” and “PWSReader.

1. Run the script for the DOI database.
    a. Run the script `PT5_PWS_DOI.sql` against the DOI database.
2. Now run the script for the ROI database.
    a. Run the script `PT5_PWS_ROI.sql` against the ROI database.
    b. Before running the sql script `PT5_PWS_Create_Synonyms.sql` you must change the name in the script file. Change the name in bold to the name of the DOI database, `SET @DatabaseName =` **`'ptDOI4'`** to the correct name of DOI database.
    c. Run the script against the ROI database.
3. Now run the script for the PWSD database.
    a. Create database PWSD (if not already existing) and execute script `PT5_PWS_PWSD.sql` on the PWSD database.

   b. Before running the sql script `PT5_PWS_Create_Synonyms.sql` on the PWSD database, you must change the name in the script file. Change the name in bold to the name of the DOI database, `SET @DatabaseName =` **`'ptDOI4'`** `to` the correct name of DOI database.

4. Change the database name in the script `PWS_Create_sqljob.sql if needed.` Run the script `PWS_Create_sqljob.sql` for the sql jobs. This is only required to be applied once on the first install. If it is executed when the job already exists the script will not change the old job.

3.6.5.1 Preventing Deadlocks

ROI database is frequently accessed and updated and sometimes deadlocks can occur, to minimize the risk of deadlock it's a recommendation that the ROI database is configured to use READ_COMMITTED_SNAPSHOT

3.6.6 Alteration of installation

After installation has been made, you cannot change the target DOI and ROI databases without reinstallation of PWS or running the SQL installation script for the new target database before changing the connection string in the configuration file.

3.6.7 Configuration of services

There are several configuration files with PubTrans Web Services. Several of them needs changes when doing a new installation. Example files will be created when used the first time with default values if they don't exists.

The application installs example files that at a new installation should be copied before changing values.

Configuration files that needs to be copied and changed when doing a new installation:

- PWSConfig_example.xml to PWSConfig.xml
- LicenseConfig_example.xml to LicenseConfig.xml
- PwsServiceHostApplication.exe_Example.config to PwsServiceHostApplication.exe.config

3.6.7.1 PWSConfig.xml

This file contains general information about the service. It contains the connections strings for database access, version of PubTrans that is installed etc. There is also a parameter for database timeout that might be tweaked if the server is slow at periods or it is running on some slow hardware. There might be a lot of reason why the database calls takes time but there is a possibility to change the timeout with this parameter.

The WebServerAddress element represent the port that the built in web server is hosting a monitoring page that will show some info about the service and which web services that is configured and is running.

Real time window is configured by the two parameters, RoiRequestOffsetForwardHours and RoiRequestOffsetBackwardHours. These values are used for all methods that are handling real time data. It is important that this window doesn't exceed the window of real time data in ROI database in PubTrans.

NoOfDaysToIncludeDaylightAdjustment should be set to the max number of days a journey is in progress. A journey in DOI can be in progress for max 4 days. Default value for NoOfDaysToIncludeDaylightAdjustment is one day.

3.6.7.2 PWS Legacy mode

To provide backward compability if a method is changed and the old behavior is preferred these method can use legacy mode. The old methods is accessible by specifying that the service should use the old method in PWSConfig.xml

Example:

```
  <PWS_Legacy>
            <Legacy_GetDirectJourneysbetweenStops Use_Legacy="true" />
  </PWS_Legacy>
```

<u>Currently available methods</u>

GetDirectJourneysBetweenStops, which was changed in version 2.6.0. The old method is now accessible by specifying that the service should use the old method:

```
<Legacy_GetDirectJourneysbetweenStops Use_Legacy="true" />
```

3.6.7.3 LicenseConfig.xml

This file defines which clients are able to call the services and which service. It also defines how many calls the client can make per hour on each service.

Different clients can have different configuration values.

```
<License applicationName="PWS Service Host Application"
         holderName="Hogia"
         validFromDate="2008-01-01"
         invalidFromDate="2009-12-31"
         licenseHandling="true" >

  <Client userName="pws"
          password="pws"
          defaultMaxCallsPerHour="100"
          defaultTimeOutSeconds="10"
          defaultNotificationIntervalSeconds="60" >
    <GrantedServices>
      <Service name="Hogia.PubTrans.WebServices.GeneralMessageService" />
      <Service name="Hogia.PubTrans.WebServices.StopMonitoringService" maxCallsPerHour="10" />
      <Service name="Hogia.PubTrans.WebServices.StopDataService" />
      <Service name="Hogia.PubTrans.WebServices.LineDataService" />
      <Service name="Hogia.PubTrans.WebServices.TopologyDataService" />
      <Service name="Hogia.PubTrans.WebServices.OrganisationalDataService" />
      <Service name="Hogia.PubTrans.WebServices.VehicleMonitoringService" />
      <Service name="Hogia.PubTrans.WebServices.TimetableDataService" />
    </GrantedServices>
  </Client>

</License>
```

On the service element there is a possibility to add values like maxCallsPerHour that overrides the defaultMaxCallsPerHour for a specific service. All default values on the client can be overridden on service level except username and password.

There are some global values that also need to be changed upon installation. The validFromDate och invalidFromDate are values that reflect the license agreement for PWS.

The licenseHandling is a way to turn of username/password verification when calling a web service. When running PWS in a production environment, this value should always be **true**.

3.6.7.4 PwsServiceHostApplication.exe.config

This configuration file is a standard app.config file where all services that is accessible is defined. The transport and security is also configured in this file.

Each service has a service section, se below.

```
<services>
  <service behaviorConfiguration="PwsServiceBehavior" name="Hogia.PubTrans.WebServices.Contracts.DataSet.GeneralMessageService">
    <endpoint address="" binding="basicHttpBinding" bindingConfiguration="PwsServiceBinding"
      name="GeneralMessageService" contract="Hogia.PubTrans.WebServices.Contracts.DataSet.IGeneralMessageService" />
    <host>
      <baseAddresses>
        <add baseAddress="https://172.16.4.145:9980/Pws/GeneralMessageService" />
      </baseAddresses>
    </host>
  </service>
</services>
```

The service has an attribute that tells which behaviour configuration it shall use, all services has the same configuration as default. As default all services has one endpoint configured with basicHttpBinding. The endpoint also has an attribute to a configuration for the binding. All endpoints has the same configuration as default.

The service has a host section which defines the baseAddress for the service. This address is the external address that the clients will use to call the web service. The reason for using the external address is if the client is gathering metadata by calling the service with *?wsdl* added to the url.

In the binding configuration there is attributes for receiveTimeout that might be tweaked to handle large result sets that take some time on the server to create. MaxBufferSize and maxReceivedMessageSize also needs to be changed so it can handle larger results. As default these attributes are rather high and can be tweaked for the required usage.

The security mode is TransportCredentialOnly as default which means that it will use http and username/password authentication for calling the service. For more info on how you configure the service to use https please read the chapter *3.6.8 Security with SSL.*

```
<bindings>
  <basicHttpBinding>
    <binding name="PwsServiceBinding" closeTimeout="00:10:00" openTimeout="00:10:00"
      receiveTimeout="00:10:00" sendTimeout="00:10:00" bypassProxyOnLocal="false"
      hostNameComparisonMode="StrongWildcard" maxBufferSize="65536000"
      maxBufferPoolSize="524288" maxReceivedMessageSize="65536000"
      messageEncoding="Text" textEncoding="utf-8" transferMode="Buffered"
      useDefaultWebProxy="false" >
      <readerQuotas maxStringContentLength="8192" maxArrayLength="20971520" />
      <security mode="TransportCredentialOnly">
        <transport clientCredentialType="Basic" />
      </security>
    </binding>
  </basicHttpBinding>
</bindings>
```

In the behaviour configuration there are settings for getting the metadata from the service. This is necessary if you want to auto generate a proxy client with Visual Studio to use when calling the methods on the services.

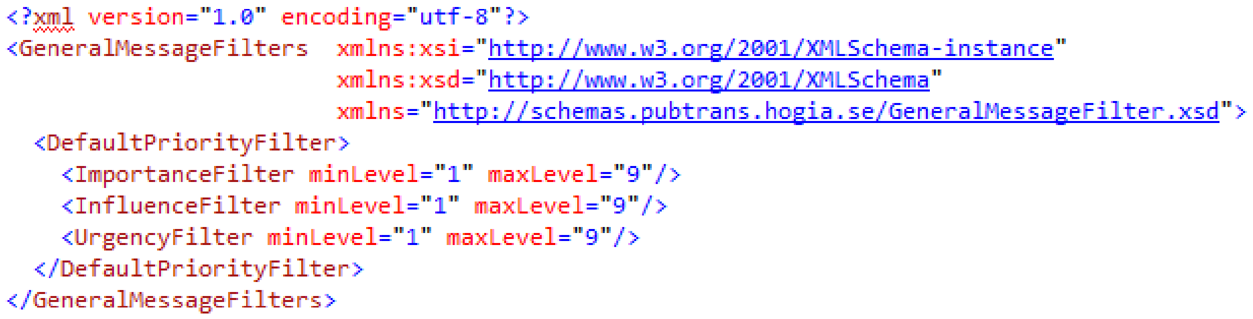
```
<behaviors>
  <serviceBehaviors>
    <behavior name="PwsServiceBehavior">
      <dataContractSerializer maxItemsInObjectGraph="999999" />
      <serviceDebug includeExceptionDetailInFaults="true" />
      <serviceMetadata httpGetEnabled="true" />
      <serviceCredentials>
        <userNameAuthentication
          userNamePasswordValidationMode="Custom"
          customUserNamePasswordValidatorType="Hogia.PubTrans.WebServices.Common.CustomUsernamePasswordValidator,
                                               PwsCommonComponents" />
      </serviceCredentials>
    </behavior>
  </serviceBehaviors>
  <endpointBehaviors>
    <behavior name="PwsEndpointBehavior">
      <dataContractSerializer maxItemsInObjectGraph="999999" />
    </behavior>
  </endpointBehaviors>
</behaviors>
```

We also use a custom username validator that helps validate the username and password that the clients use when calling the services. When running in a production environment the attribute includeExceptionDetailInFaults should be false and the implementation of the service should return faults that give the client only the information that they need to know.

Endpoints for backwards compatibility is commented out. Endpoints with backwards compatibility to version 2.12.1 exists for the following services: LineDataService, TimetableDataService and OrganisationalDataService.

3.6.7.5 GeneralMessageFilters.xml

This configuration file describes for PubTrans Web Services what to interpret as a general deviation message.



```
<?xml version="1.0" encoding="utf-8"?>
<GeneralMessageFilters  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                        xmlns="http://schemas.pubtrans.hogia.se/GeneralMessageFilter.xsd">
  <DefaultPriorityFilter>
    <ImportanceFilter minLevel="1" maxLevel="9"/>
    <InfluenceFilter minLevel="1" maxLevel="9"/>
    <UrgencyFilter minLevel="1" maxLevel="9"/>
  </DefaultPriorityFilter>
</GeneralMessageFilters>
```

Here you can change the minimum level and maximum level för importance, influence and urgency on deviation messages. These levels must match the deviation message for it to be interpret as a general message.

The example above has min level at 1 and max level at 9 which means that all messages will be returned as general message.

3.6.7.6 PWSD Database configuration, changing line variant date interval

The line variant data is generated by a sql job which is executed once every 24 hours. By default, the job generates data in the interval 'todays date' minus 1 day => 'todays date' plus 30 days. These values can be changed by executing the sql script `PWS_SetLineVariantInterval.sql` with appropriate parameters to stored procedure `pwsSetLineVariantDateInterval` in the PWSD database. The parameters to set is two numbers; the first is for setting number of days before today and the second is for setting no of days in the future from today. Example syntax for setting the default values (yesterday until 30 days ahead):

```
[dbo].[pwsSetLineVariantDateInterval] -1, 30
```

3.6.8 Security with SSL

If the client shall access the web service through a secure connection with **https** you have to make some additional steps above the normal installation. These instructions are made for using our windows service to host PubTrans Web Services. If using the IIS as a host then these instructions may vary some.

First you need to have a certificate which is signed by a root certificate that is known by both the server and the client that is calling the service. If the client don't have the root certificate then they must import that to their clients in the certificate console at trusted root certification authorities.

If you create a new server certificate then it is imported that the CN has the value of the address that is used to call the service. For example if the baseAddress to the service is *https://172.16.4.145:9980/Pws/GeneralMessageService* then the CN must be *172.16.4.145*. In the subject field on the certificate it will say CN = 172.16.4.145. If the service is called by a domain name for example https://www.hogia.com:9980/Pws/GeneralMessageService, then the CN shall be www.hogia.com.

3.6.8.1 Server side

On the server side there is some settings that needs to be added to the existing configuration file for the PubTrans Web Services. In the *PwsServiceHostTestApplication.exe.config* file change the security mode for all services to Transport. This tells the WCF framework that we want to use certificate on the server to authenticate when using https.

First the baseAddress for the service must be changed to https instead of http. This has to be changed on all services that should use SSL for transport encryption.

```
<host>
  <baseAddresses>
    <add baseAddress="https://172.16.4.145:9980/Pws/GeneralMessageService" />
  </baseAddresses>
</host>
```

In the binding configuration you must use Transport as security mode. See the example below.

```
<security mode="Transport">
  <transport clientCredentialType="Basic" />
</security>
```

There are also some changes that have to be made under service behaviour. The httpsGetEnabled must be set to true if the client shall have the ability to download metadata about the service and create a proxy client. There is also a section with serviceCredentials that tells the service where to find the certificate and what certificate it should use.

```
<serviceMetadata httpsGetEnabled="true" httpGetEnabled="false" />
<serviceCredentials>
  <serviceCertificate findValue="d5f629b2c88d98808b8a05c2138759bb956a2110"
      storeLocation="LocalMachine"
      storeName="My"
      x509FindType="FindByThumbprint" />
  <userNameAuthentication userNamePasswordValidationMode="Custom"
                          customUserNamePasswordValidatorType="Hogia.PubTra
</serviceCredentials>
```

In the example above where looking for the certificate by thumbprint, there are other ways to find it and the x509FindType attribute tells what type it shall use for finding the certificate. The findValue is the value that it looks for. If the certificate changes than the thumbprint also changes so this value must then be updated with the new value for the service to find the correct certificate. The attribute storeLocation and storeName describe where in the certificate store the certificate is located.

Next step is to configure a port with SSL, this is necessary when using a self-hosted windows service. For doing this we use a tool called httpcfg.exe. For more info how to use httpcfg.exe and to download it if doesn't exists on the system hosting PWS2, see http://msdn.microsoft.com/en-us/library/ms733791.aspx.

When configuring the port for SSL you need to know which port your using for the service, you can find it in the configuration for the service. In the examples where using port 9980 for connection to the service through https.

In the command windows on the server type:

```
httpcfg.exe query ssl
```

This will list all ports that is configured for SSL. If there is a configuration for the port that you want to use you can remove the existing with:

```
httpcfg.exe delete ssl -i 0.0.0.0:9980
```

Then you need to add a new configuration for the port. The configuration needs to be bound to a specific thumbprint from the certificate you are using for the service. Get the correct thumbprint and remove all spaces and enter it with:

```
httpcfg.exe set ssl -i 0.0.0.0:9980 -h d5f629b2c88d98808b8a05c2138759bb956a2110
```

When updating or changing a certificate the configuration for the port must be removed and added with the new thumbprint.

If a port is configured for use with https then you can't use that port for http, if want to use http on that port instead of https then you need to delete it as mentioned above.

Now the configuration is done on the server.

3.6.8.2 Client side

The client also needs to be configured using https for transport. But on the client it is more straight forward and much less things that need to be changed.

First the address for the endpoint must change so it will use https instead of http.

```
<endpoint address="https://172.16.4.145:9980/Pws/GeneralMessageService"
```

Then the binding configuration for the endpoint must be changed so it uses the Transport mode, see example below.

```
<security mode="Transport">
  <transport clientCredentialType="Basic" proxyCredentialType="None"
    realm="" />
  <message clientCredentialType="UserName" algorithmSuite="Default" />
</security>
```

If the server is using a root certificate that the client doesn’t know about then it needs to be imported on the client in trusted root certification authorities.

That should be the only things that have to be done on the client.

3.6.9 Test the web services

Make sure that the service is started.

Then open a web browser window and enter the url: **http://*servername:portnumber***, which is found in the **PwsConfig.xml** element **WebServerAddress.** An information web page will be shown if the installation is correct.

4 User Guidelines

This chapter contains general guidelines for client developers.

4.1 Accessing the Web Services

4.1.1 Getting the Service Contract

Once PWS is installed and the services are started, the service contract can be obtained by any tool that supports *WS-I Profile Specification 1.1*. From this data, a proxy client can be generated and used for accessing PWS. A service contract is available on **http://*server*:*port*/Pws/*ServiceName*?*wsdl***. The default port number PWS is installed with is 9980.

4.1.2 Using the Services

A service is available on **http://*server*:*port*/Pws/*ServiceName***. The default port number PWS is installed with is 9980.

The contracts of the services are described in *Part 1: Service Contracts*. In this part, each web method is described with its parameters and which result that is returned.

Request data services returns data set. The detailed descriptions of the returned data sets are in *Part 2: Response Reference*. Note that the same types of tables are returned by several web methods.

The rest of this chapter contains the general guidelines for using PWS.

4.1.3 Real time

When accessing a service method that returns real time data then a real time window is used by the service. If the requested time of data is outside the real time window then an exception will be returned to the client. The real time window is configurable in ***PWSConfig.xml***.

If some methods has other behaviours besides these then that is noted at the method in, *Part 1: Service Contracts*.

4.2 Parameters in Web Methods

4.2.1 Data Types

Parameter values follow the general PubTrans® data types. Some of these data types are documented in this user manual, and a full list can be found in the documentation of PubTrans® Public Interfaces, DOI and ROI. In the documentation, the name of the .NET data types is used. The table below maps these data types to their corresponding XML data types.

Data type	XML data type	Comment
Boolean	boolean	True or False.
Date	date	Without time part. Note that some environments do not distinguish between Date and DateTime values. In that case, be sure that the time part is 00:00:00. If

Data type	XML data type	Comment
		the data contains a time other then 00:00:00 then a ClientFault will be generated.
DateTime	dateTime	Full date and time value.
Gid	decimal	Corresponds to a 16 digit integer value. See [IS-PT/I/DOI/4]
Integer	decimal	Corresponds to a 32-bit signed integer.
Numeric(16)	decimal	Corresponds to a 16 digit integer value.
String	string	UNICODE string value.
TimeSpan	duration	
Code	string	Corresponds to a 8 character string.

4.2.2 Optional Parameters

In PWS 2.0 and later, it is no longer required to provide dummy values for optional parameters. If a value must be provided, supply a **null** value (**Nothing** in Visual Basic).

For backwards compatibility, PWS 2.0 still support .NET **MinValue** and **MaxValue** for numerical values (including **Date**, **DateTime** and **TimeSpan**). Please note, that this behaviour is not guaranteed to be supported in releases after PWS 2.0.

4.2.3 Parameters with Wildcards

For some string parameters, wildcards can be used to filter the result set. Add wildcard character **%** in end of parameter value to match values starting with the parameter value. Add % in both start and end to match values containing the parameter value. Note that wildcard in the beginning of a value may slow down performance and response time.

4.2.4 Multiple Value Parameters

A multiple valued parameter is a list of comma separated values. In the documentation, a multiple value parameter is noted with **[n-m]** after the data type name, where n is the minimum required number of values and m is the maximum number of values. If the wrong number of values is provided, PWS returns a fault.

4.2.5 Period of Dates

PWS has a limitation for which dates data can be retrieved. This means that parameters that defines validity in a period of days (*fromDate*, *uptoDate*) must be within the possible dates; otherwise a fault will be returned.

The possible period of dates is from a configurable number of days before current date up to but not including the *Published Upto Date* setting for the transport authority. If data for several transport authorities are requested, the *uptoDate* parameter must be less or equal than the lowest *Published Upto Date* setting for any transport authority in PubTrans.

When selecting data, PWS always regards the *fromDate* as inclusive and the *uptoDate* as exclusive. For example, to select data valid in December 2007, *fromDate* should be 2007-12-01 and *uptoDate* should be 2008-

01-01. Note that setting *uptoDate* to 2007-12-31 will not include the last day in December 2007 in the search for valid objects.

4.3 Result Sets

This section contains general guidelines for how to interpret data in the result sets. Result sets for request data methods contain a data set with one or several tables. See section 2.6 above for details regarding interoperability.

4.3.1 Object Versions

All objects in PubTrans are versioned in some way or another. In general, all objects have the attributes *Exists From Date* and an *Exists Upto Date*.

- The attribute *Exists From Date* specifies the first date of validity for that specific object version.
- The attribute *Exists Upto Date* specifies the first date of invalidity (and not the last date of validity). If the value of this attribute is null, it denotes that the object version at the moment has infinite validity. Later updates of the object will set this value to a date.

If the attributes *Exists From Date* and an *Exists Upto Date* are present in a result set, that result set may contain more than one version of the same object. For instance the same stop can be present in two versions with different names. Versions do not overlap in time, so maximum one object version can be valid at a specific date.

Object versioning is described in more detail in [IS-PT/I/DOI/4].

4.3.2 Sorting Order for Line Numbers

Presenting line numbers in public information shall use the attribute Line Designation, not the Line Number. The Line Number is always numeric and should be regarded as an internal technical identifier for the line.

Because Line Designation may be alphanumeric, the sort order is not given, but can vary between installation sites of PWS. PWS provides a separate attribute Display Order that shall be used for sorting lines for display purposes, e.g. in a drop down box.

PWS can be configured to use different algorithms. The configured sorting algorithm applies per site. In some methods, it is possible to override the configured default sorting algorithm.

Currently, the following algorithms for calculating the Display Order attribute are supported:

0. Uses the transport authority number and the full four digit line number. Values can range from 0 to 9999999.
1. Uses the line number attribute, i.e. the technical identifier for the line. Values can range from 0 to 9999.
2. Uses the last three digits in the line number attribute. Values can range from 0 to 999.
3. Uses the transport authority number and last three digits in the line number attribute. Values can range from 0 to 999999.

For additional sorting algorithms, please contact Hogia Public Transport Systems.

4.4 Error Handling

From PWS 2.0, errors are returned as SOAP-faults. This mechanism enables structured error handling in the clients, because the client proxy usually will have support for exceptions.

4.4.1 Faults

PWS returns a SOAP fault when an exception occurs that prevent PWS to perform the requested operation. The fault will contain the following fault details:

- **MethodName**: the name of the method that caused the fault.
- **Parameters**: a list of provided parameters - name and value.
- **Reason**: the reason of the fault.
- **Message**: additional information about the fault and how to correct the fault.

4.4.2 Type of Faults

PWS return different reasons for faults depending on what that caused the fault:

- **Client Fault**: returned when the client has made a logical error when calling a web method. This error has to be corrected in the implementation of the client.
- **Not Supported Fault**: returned when the client has called a method which is not supported on the specific PubTrans® environment. See RN-PWS/2 for which methods that are supported on different PubTrans® environments e.g. PubTrans® 4 or 5.
- **Configuration Fault**: returned when PWS detects error in the configuration. This error has to be corrected by the system administrator of the site where PWS is running.
- **Implementation Fault**: returned when PWS detects an error in the internal logic of PWS. This error has to be corrected by Hogia Public Transport Systems.
- **Runtime Fault**: returned when PWS cannot carry out the requested operation due to that some resource temporarily is not available. This error has to be analysed and corrected by the system administrator of the site where PWS is running.
- **License Fault**: returned when the licensed maximum of usage is exceeded. This indicates that a client is using PWS more than expected. If this is a desired behaviour, the system administrator need to obtain an extended license from Hogia Public Transport Systems; otherwise, the behaviour of the client should be optimised.

Part 1: Service Contracts

This part describes the web service contracts, i.e. methods with their request parameters and response data. Request parameters are of two types; required and optional. Response data is a dataset containing one or several tables that in this part only is listed with their names. The tables are fully documented in section, Part 2: Response Reference.

5 Organisational Data Service

The Organisational Data Service provides data business participants, their organisational structure and where special responsibilities/functions are placed in the organisation.

5.1 Get Contractor Info

Gets name, address and phone numbers for operators that are contracted by transport authorities.

5.1.1 Usage

This method can be used in interactive mode if table Operated Lines is not included; otherwise only in batch mode.
This method is **obsolete** and shall not be used when developing new applications, use Get Contractor with Operated Lines.

5.1.2 Request parameters

Name	Data type	Opt?	Default	Description
atDate	Date	No		Controls which version of information that is returned.
forTransportAuthorityGid	Gid	Yes	Any	Option to filter the result to contractors for the specified transport authority.
withTypeOfServiceFunctionCode	String(1-8)	Yes	Any	Option to filter the result in table # 3 to contain only functions of the specified type. For valid service function codes, see DOI documentation for data type *typServicFunctionCode*.
includeOperatedLinesTable	Boolean	Yes	False	If true, the response includes the *Operated Lines* table.
useLineDisplayOrderVariant	Byte	Yes	1	Defaults to 1.

5.1.3 Response data

Index	Table Name	Page	Opt?	Note
0	Organisational Unit	59	No	
1	Transport Authority	60	No	
2	Contractor	60	No	Operators of lines.
3	Service Function For Organisational Unit	63	No	
4	Operated Lines	74	Yes	

5.2 Get Contractors with Operated Lines

Gets name, address and phone numbers for operators that are contracted by transport authorities. There is also an option to get the lines operated by the contractors.

5.2.1 Usage

This method can be used in interactive mode if table Operated Lines is not included; otherwise only in batch mode.

5.2.2 Request parameters

Name	Data type	Opt?	Default	Description
fromDate	Date	No		Controls which version of information that is returned.

uptoDate	Date	Yes	⇒	Defaults to fromDate + 1 day.
forTransportAuthorityGid	Gid	Yes	Any	Option to filter the result to contractors for the specified transport authority.
withTypeOfServiceFunctionCode	String(1-8)	Yes	Any	Option to filter the result in table # 3 to contain only functions of the specified type. For valid service function codes, see DOI documentation for data type *typServicFunctionCode*.
includeOperatedLinesTable	Boolean	Yes	False	If true, the response includes the *Operated Lines* table.
useLineDisplayOrderVariant	Byte	Yes	1	Defaults to 1.

5.2.3 Response data

Index	Table Name	Page	Opt?	Note
0	Organisational Unit	59	No	
1	Transport Authority	60	No	
2	Contractor	60	No	Operators of lines.
3	Service Function For Organisational Unit	63	No	
4	Operated Lines	74	Yes	

5.3 Get Service Function for Line

Get information about a service function for a line, for example find the address, phone number and location of a Lost and found office.

5.3.1 Usage

This method can be used in interactive and batch mode.

5.3.2 Request parameters

Name	Data type	Opt?	Default	Description
atDate	Date	No		Controls which version of information that is returned.
forLineGid	Gid	No		The line for which service information should be retrieved.

Name	Data type	Opt?	Default	Description
withServiceFunctionCode	String(1-8)	Yes	Any	Option to filter the result in table # 2 to contain only functions of the specified type. For valid service function codes, see DOI documentation for data type *typServicFunctionCode*.

5.3.3 Response data

Index	Table Name	Page	Opt?	Note
0	Organisational Unit	59	No	
1	Service Function For Organisational Unit	63	No	
2	Communication Address For Organisational Unit	63	No	

5.4 Get Contracts With Lines

Get contracts and related information. There is also an option to get the lines under the contracts.

5.4.1 Usage

This method can be used in interactive mode if table Line In Contract is not included; otherwise only in batch mode.

5.4.2 Request parameters

Name	Data type	Opt?	Default	Description
fromDate	Date	No		Controls which version of information that is returned.
uptoDate	Date	Yes	⇒	Defaults to fromDate + 1 day.
OnlyWithinTransportAuthorityGid	Gid	Yes	Any	Option to filter the result to contractors for the specified transport authority.
forLineGids	Gid [1-4]	Yes	All	Filters the result set to only contain contracts promoted to the specified lines.
includeLinesInContractTable	Boolean	Yes	False	If true, the response includes the *Lines in Contract* table.

5.4.3 Response data

Index	Table Name	Page	Opt?	Note
0	Organisational Unit	59	No	
1	Transport Authority	60	No	
2	Contract	61	No	
3	Line In Contract	61	Yes	

6 Topology Data Service

The Topology Data Service provides data about points, links and zones.

6.1 Get Zones

This method returns zones of a given type. Optionally, it also returns the lines that operate stops in the zones.

6.1.1 Usage

This method can be used in interactive when table Line in Zone is not included; otherwise only in batch mode. Note that, in batch mode, this method might take some time to complete.

6.1.2 Request parameters

Name	Data type	Opt?	Default	Description
fromDate	Date	No		Controls which version of information that is returned.
uptoDate	Date	Yes	⇒	Defaults to fromDate + 1 day.
forTransportAuthorityGid	Numeric(16)	Yes	All	Filters the result set to only contain zones (and lines) within the specified transport authority.
forTypeOfZone	String(1-8)	Yes	All	The type code of the zone. See [IS-PT/I/DOI/4] for valid values of *typZoneTypeCode*.
includeTableLineInZone	Boolean	Yes	False	If true, the result set will include table Line in Zone; otherwise not.

6.1.3 Response data

If there are no common stops for the specified lines, the response will be empty.

Index	Table Name	Page	Opt?	Note
0	Zone	64	No	
1	Line in Zone	64	Yes	

6.1.4 Examples

To find all valid zones for municipalities during June 2008:

```
GetZones( 2008-06-01, 2008-07-01, null, "MUNICIPZ", false)
```

7 Stop Data Service

The Stop Data Service provides data about places, sites, stops and stations.

7.1 Get Stops Common for Line

Get common stops for 1 to 4 lines by line Gid.

7.1.1 Usage

This method can be used in interactive and batch mode. Note that this method might take some time to complete.

7.1.2 Request parameters

Name	Data type	Opt?	Default	Description
atDate	Date	No		Controls which version of information that is returned.
forLineGids	Gid [1-4]	No		Comma separated list of 1 – 4 line Gids.

7.1.3 Response data

If there are no common stops for the specified lines, the response will be empty.

Index	Table Name	Page	Opt?	Note
0	Stop Area	66	No	

7.2 Get Stops Info

Get info about valid stops.

7.2.1 Usage

This method can be used in interactive and batch mode. Retrieving all stops might take some time to complete. Searching with partial name will limit the result set and shorten the response time.

7.2.2 Request parameters

Name	Data type	Opt?	Default	Description
atDate	Date	No		Controls which version of information that is returned.
forTransportAuthorityGid	Gid	Yes	Any	If provided, filters the result set to only contain stops info for the transport authority.

withPartialName	String(1-50)	Yes	Any	If provided, limits the result to stop area names, stop points names and stop point road names containing the parameter value. Wildcards can be used, see 4.2.3 for usage.

7.2.3 Response data

Index	Table Name	Page	Opt?	Note
0	Stop Area	66	No	
1	Stop Point	66	No	

7.3 Get Stops with Lines

Get information about all stops valid for a given period, and optionally which lines that are valid at the stops in that period.

7.3.1 Usage

This method can be used in interactive mode if table Line at Stop is not included in the result set; otherwise in batch mode.

Retrieving all stops might take some time to complete. Searching with partial name will limit the result set and shorten the response time.

7.3.2 Request parameters

Name	Data type	Opt?	Default	Description
fromDate	Date	No		Controls which version of information that is returned.
uptoDate	Date	Yes	⇒	Defaults to fromDate + 1 day.
forTransportAuthorityGid	Numeric(16)	Yes	All	Filters the result set to only contain stop areas and lines within the specified transport authority.
withPartialName	String(1-50)	Yes	Any	If provided, limits the result to stop area names, stop points names and stop point road names containing the parameter value. Wildcards can be used, see 4.2.3 for usage.
includeTableLines-AtStop	Boolean	Yes	False	If true, the result set will include table Line at Stop; otherwise not.
includeTableDestinationAtStop	Boolean	Yes	False	If true, the result set will include table Line Destination At Stop; otherwise not.

7.3.3 Response data

Index	Table Name	Page	Opt?	Note
0	Stop Area	66	No	
1	Stop Point	66	No	
2	Line at Stop	67	Yes	
3	Line Destination at Stop	68	Yes	

7.4 Get Detailed Stop Info

Get info about valid stops with an option to include Zone and Site information.

7.4.1 Usage

This method can be used in interactive and batch mode. Retrieving all stops might take some time to complete. Searching with Gid will limit the result set and shorten the response time.
The method is primarily intended for batch use.

7.4.2 Request parameters

Name	Data type	Opt?	Default	Description
atDate	Date	No		Controls which version of information that is returned.

Name	Data type	Opt?	Default	Description
forTransportAuthorityGid	Gid	Yes	Any	If provided, filters the result sets to only contain info for the specified transport authority.
atStopGid	Gid	Yes	Any	If provided, filters the result sets to contain only rows related to the specified stop area, stop point or journey pattern point at the specified date.
atZoneGid	Gid	Yes	Any	If provided, limits the result sets to contain only rows related to the specified zone
atSiteGid	Gid	Yes	Any	If provided, limits the result sets to contain only rows related to the specified site
atSiteNumber	Numeric (1-8)	Yes	Any	If provided, limits the result sets to contain only rows related to the specified site. Should be used in combination with forTransportAuthorityGid because SiteNumber is only unique within a transport authority.
atStopAreaWithPartialName	String(1-50)	Yes	Any	If provided, limits the result to stops having stop area name containing the parameter value. Wildcards can be used, see Parameters with Wildcards for usage.
includeZoneTables	Boolean	Yes	False	If provided as True then both table Zone and JourneyPatternPointInZone are populated in the response
includeSiteTables	Boolean	Yes	False	If provided as True then both table Site and SiteNeighbouringStopArea are populated in the response

7.4.3 Response data

Index	Table Name	Page	Opt?	Note
0	Stop Area	66	No	
1	Stop Point	66	No	
2	Zone	64	Yes	Returned if includeZoneTables is set to true.
3	Journey Pattern Point In Zone	70	Yes	Returned if includeZoneTables is set to true.
4	Site	68	Yes	Returned if includeSiteTables is set to true
5	Site Neighbouring Stop Area	70	Yes	Returned if includeSiteTables is set to true

7.5 Get Site Info

Get info about valid sites.

7.5.1 Usage

This method can be used in interactive and batch mode. Retrieving all sites might take some time to complete. Searching with partial name will limit the result set and shorten the response time.
The method is primarily intended for batch use.

7.5.2 Request parameters

Name	Data type	Opt?	Default	Description
atDate	Date	No		Controls which version of information that is returned.
atSiteWithPartial Name	String(1-50)	Yes	Any	If provided, limits the result to sites containing the parameter value. Wildcards can be used, see Parameters with Wildcards for usage.
forTransportAuth orityGid	Gid	Yes	Any	If provided, filters the result set to only contain site info for the transport authority.
atSiteGid	Gid	Yes	Any	If provided, filters the result set to only contain site info for the specified site.
atStopGid	Gid	Yes	Any	If provided, filters the result sets to contain only rows related to the specified stop area, stop point or journey pattern point.
includeStopArea-Tables	Boolean	Yes	False	If provided as True then both table StopArea and SiteNeighbouringStopArea are populated in the response

7.5.3 Response data

Index	Table Name	Page	Opt?	Note
0	Site	68	No	
1	Stop Area	66	Yes	Returned if IncludeStopAreaTables is set to true
2	Site Neighbouring Stop Area	70	Yes	Returned if IncludeStopAreaTables is set to true

8 Line Data Service

The line information web service provides general data about lines, directions, group of lines and operators of lines.

8.1 Get Lines

Get data about all lines in a PubTrans database or selected lines for a specified transport authority, group of line, products, line designations and/or stop area/stop point.

8.1.1 Usage

This method can be used in interactive and batch mode. Retrieving all lines might take some time to complete. In general, response time will be shorter the narrower the search is.

8.1.2 Request parameters

Name	Data type	Opt?	Default	Description
atDate	Date	No		Controls which version of information that is returned.
forTransportAuthorityGid	Gid	Yes	Any	If provided, filters the result to lines managed by the specified transport operator; otherwise all lines in PubTrans are returned.
withPurposeOfLineGroupingCode	String(1-8)	Yes	Any	If provided, sets the qualifying parameter for InGroupOfLines.
inGroupOfLinesCode	String(1-8)	Yes	Any	If provided, filters the result set to only contain lines that is in the specific group.
inGroupOfLinesWithPartialName	String(1-50)	Yes	Any	If provided, limits the result set to lines in groups, which names contains the parameter value. . Wildcards can be used, see 4.2.3 for usage.
atStopGid	Gid	Yes	Any	If provided, filters the result set to contain only lines that operate the specified stop area, stop point or journey pattern point at the specified date.
atStopWith-PartialName	String(1-50)	Yes	Any	If provided, limits the result to lines at stops where the parameter value matches stop area name, stop point name or stop point road name. Wildcards can be used, see 4.2.3 for usage.
withProductCodes	String(1-8) [1-4]	Yes	Any	Comma separated list of product codes. If provided, filters the result set to only contain lines of the specified products.

Name	Data type	Opt?	Default	Description
withLineDesignations	String(1-4) [1-4]	Yes	Any	If provided, filters the result set to only contain the line(s) with the specified designations. Wildcards can be used, see 4.2.3 for usage.
includeNonPublic-Lines	Boolean	Yes	No	If provided, filters the result to also include lines not normally showed to the public.
useLineDisplayOrderVariant	Byte	Yes	Setting	Defaults to setting in PWS configuration.

8.1.3 Response data

Index	Table Name	Page	Opt?	Note
0	Line	71	No	
1	Group Of Lines	71	No	
2	Line In Group Of Lines	73	No	
3	Direction Of Line	74	No	

8.2 Get Lines with Stops

Get all valid lines within a period, optionally get all stops for each line within the same period.

8.2.1 Usage

This method can only be used in batch mode.

8.2.2 Request parameters

Name	Data type	Opt?	Default	Description
fromDate	Date	No		Controls which version of information that is returned.
uptoDate	Date	Yes	⇒	Defaults to fromDate + 1 day.
forTransportAuthorityGid	Numric(16)	Yes	All	Filters the result set to only contain stop areas and lines within the specified transport authority.
includeTableStopOnLine	Boolean	Yes	False	

8.2.3 Response data

Index	Table Name	Page	Opt?	Note
0	Line	71	No	
1	Group Of Lines	71	No	
2	Line In Group Of Lines	73	No	
3	Direction Of Line	74	No	
4	Stop on Line	75	Yes	

8.3 Get Stops On Line

Get all stop points ordered by direction one for specified line.

8.3.1 Usage

This method can only be used in interactive mode.

8.3.2 Request parameters

Name	Data type	Opt?	Default	Description
forLineGid	Gid	No		
fromDate	Date	No		Controls which version of information that is returned.
uptoDate	Date	Yes	⇒	Defaults to fromDate + 1 day.

8.3.3 Response data

Index	Table Name	Page	Opt?	Note
0	Stop on Line	75	Yes	

8.4 Get Stops On Direction Of Line

Get all stop points for specified direction of line.

8.4.1 Usage

This method can only be used in interactive mode.

8.4.2 Request parameters

Name	Data type	Opt?	Default	Description
forDirectionOfLineGid	Gid	No		
fromDate	Date	No		Controls which version of information that is returned.
uptoDate	Date	Yes	⇒	Defaults to fromDate + 1 day.

8.4.3 Response data

Index	Table Name	Page	Opt?	Note
0	Stop on Line	75	No	

8.5 Get Stops On Service Journey

Get all stop points for specified service journey at a specified operating date.

8.5.1 Usage

This method can only be used in interactive mode.

8.5.2 Request parameters

Name	Data type	Opt?	Default	Description
forServiceJourneyGid	Gid	No		
forOperatingDayDate	Date	No		Controls which version of information that is returned.

8.5.3 Response data

Index	Table Name	Page	Opt?	Note
0	Stop on Line	75	Yes	

8.6 Get Lines by Direction Of Line Gid

Get data about directions of 1 to 4 lines, using the Gid of directions of lines.

8.6.1 Usage

This method is **obsolete** and shall not be used when developing new applications.

8.6.2 Request parameters

Name	Data type	Opt?	Default	Description
atDate	Date	No		Controls which version of information that is returned.
forDirectionOfLineGids	Gid [1-4]	No		Comma separated list of 1 – 4 direction of line gid’s.

8.6.3 Response data

Index	Table Name	Page	Opt?	Note
0	Line	71	No	
1	Group Of Lines	71	No	
2	Line In Group Of Lines	73	No	
3	Direction Of Line	74	No	

8.7 Get Lines by Gid

Get data about 1 to 4 lines using the Gid of lines.

8.7.1 Usage

This method is **obsolete** and shall not be used when developing new applications.

8.7.2 Request parameters

Name	Data type	Opt?	Default	Description
atDate	Date	No		Controls which version of information that is returned.
forLineGids	Gid [1-4]	No		Comma separated list of 1 – 4 line gid's

8.7.3 Response data

Index	Table Name	Page	Opt?	Note
0	Line	71	No	
1	Group Of Lines	71	No	
2	Line In Group Of Lines	73	No	
3	Direction Of Line	74	No	

8.8 Get Lines by Line Designation

Get data about 1 to 4 lines using line designations.

8.8.1 Usage

This method is **obsolete** and shall not be used when developing new applications.

8.8.2 Request parameters

Name	Data type	Opt?	Default	Description
atDate	Date	No		Controls which version of information that is returned.
withLineDesignations	String(1-4) [1-4]	No		If provided, filters the result set to only contain the line(s) with the specified designations. Wildcards can be used, see 4.2.3 for usage.

8.8.3 Response data

Index	Table Name	Page	Opt?	Note
0	Line	71	No	
1	Group Of Lines	71	No	
2	Line In Group Of Lines	73	No	
3	Direction Of Line	74	No	

8.9 Get Line Variants

Get all the valid line variants for the lines in the specified time interval and within the selected filter parameters. It is only guaranteed that correct data is available within the date interval that is set by configuration script, se Changing line variant date interval.

8.9.1 Usage

This method can be used in interactive and batch mode. Retrieving all lines with line variants might take some time to complete.

8.9.2 Request parameters

Name	Data type	Opt?	Default	Description
fromDate	Date	No		Period for which to retrieve variants.
uptoDate	Date	Yes	⇒	Defaults to fromDate + 1 day.
forTransportAuthorityGid	Numeric(16)	Yes	All	Filters the result set to only contain line variants for lines within the specified transport authority.
forLineGid	Numeric(16)	Yes	All	Filters the result set to only contain line variants for the specified line.

8.9.3 Response data

Index	Table Name	Page	Opt?	Note
0	Line Variant	76	No	

9 Timetable Data Service

9.1 Get Timetable at Stop

9.1.1 Usage

This method supports handling of daylight saving time changes according to [UM-PT/I/ROI/3].

This method can only be used in batch mode.

9.1.2 Request parameters

Name	Data type	Opt?	Default	Description
atStopGid	Gid	No		Find arrivals and/or departures at this stop. StopAreaGid, StopPointGid or JourneyPatternPointGid are accepted.
forOperatingDayDate	Date	No		Find arrivals and/or departures for this operating day date.
includeArrivalsTable	Boolean	No		If false, the table will be returned but will be empty.
includeDepartures-Table	Boolean	No		If false, the table will be returned but will be empty.

9.1.3 Response data

Index	Table Name	Page	Opt?	Note
0	Dated Service Journey	86	No	Journeys with departures and/or arrivals on the requested stop on the requested operating day date. If both parameters *includeDepartures-Table* and *includeArrivalsTable* are false, no journeys are included in the response.
1	Dated Departure	88	No	Will be empty if parameter *include-DeparturesTable* is false.
2	Dated Arrival	91	No	Will be empty if parameter *include-ArrivalsTable* is false.

9.2 Get Timetable for Service Journey

Gets a timetable for a specified service journey including all calls (arrivals and/or departures).

9.2.1 Usage

This method can only be used in batch mode.

9.2.2 Request parameters

Name	Data type	Opt?	Default	Description
forSeviceJourneyGid	Gid	No		Find arrivals and/or departures for this journey.
atOperatingDayDate	Date	No		Find arrivals and/or departures for this operating day date.
includeArrivalsTable	Boolean	No		If false, the table will be returned but will be empty.
includeDepartures-Table	Boolean	No		If false, the table will be returned but will be empty.

9.2.3 Response data

Index	Table Name	Page	Opt?	Note
0	Dated Service Journey	86	No	The requested journey, if this journey has departures and/or arrivals on the requested operating day date. If both parameters *includeDepartures-Table* and *includeArrivalsTable* are false, no journey is included in the response.
1	Dated Departure	88	No	Will be empty if parameter *include-DeparturesTable* is false.
2	Dated Arrival	91	No	Will be empty if parameter *include-ArrivalsTable* is false.

10 General Message Service

The general message service provides information about general information messages affecting stops, vehicles or services, including deviations.

10.1 Get Current Deviations

Gets the valid general messages approved for public presentation at the time of the request.

The configuration for what is a general message is done in the configuration file GeneralMessageFilters.xml. This gives the possibility to filter deviation messages based on importance, influence and urgency level.

When a StopGid is used for atStopGid the service will get all the other type of Gids (JourneyPatternPointGid, StopPointGid and StopAreaGid) for that point and use as search criteria for deviations.

If there is no other gids for that point only the supplied gid will be used. For example, a JourneyPatternPointGid is supplied and there is no StopPoint for that point at that day, only the JourneyPatternPointGid will be used as search criteria.

If neither aStopGid nor forLineGids all current deviations are returned.

10.1.1 Usage

This method can only be used in interactive mode.

10.1.2 Request parameters

Name	Data type	Opt?	Default	Description
atStopGid	Gid	Yes		A Stop Point Gid, Stop Area Gid or Journey Pattern Point Gid can be used. Cannot be combined with forLineGids.
forLineGids	Gid [1-3]	Yes		Comma separated list of 1 – 3 line gid's. Cannot be combined with atStopGid.
forTimeWindowStartDateTime	DateTime	No		The date part may only be today or tomorrow.
forTimeWindowDuration	Byte	Yes	59 min	The result is limited by both these parameters, whichever is hit first. These filters are applied by sorting departures after is *Target DateTime*.
withDepartureMaxCount	Integer	Yes	-	

10.1.3 Response data

Index	Table Name	Page	Opt?	Note
0	Deviation Message	78	No	
1	Deviation Message Variant	78	No	

Index	Table Name	Page	Opt?	Note
2	Publication Scope	95	No	
3	Target Audience	95	No	

11 Stop Monitoring Service

The stop monitoring web service provides stop centric data about departures and arrivals at a certain stop.

11.1 Get Departures at Stop

Get departures at specified Stop Area or Stop Point. It returns a data set with a list of departures at the stop and contains data about the planned, estimated and observed departure times and several additional information elements. It is possible to filter the selection of calls.

This method is **obsolete** shall not be used when developing new applications. When using this method in PubTrans 5 it uses PubTrans 4 format on DirectionOfLineGids.

For PubTrans® 5 use GetCallsAtStop instead.

11.1.1 Usage

This method can only be used in interactive mode.

11.1.2 Request parameters

Name	Data type	Opt?	Default	Description
atStopGid	Gid	No	-	Stop Point Gid or Stop Area Gid can be used.
forDirectionOfLineGids	Gid [1-6}	Yes	Any	
forTimeWindowStartDateTime	DateTime	No		Must be within the timeframe of valid data in ROI, which is dependent on the PubTrans system configuration.
forTimeWindowDuration	Byte	Yes	59 min	The result is limited by both these parameters, whichever is hit first. These filters are applied by sorting departures after is *Target DateTime*. Note that PWS limits the time window to current time plus 12 hours. NOTE: *withDepartureMaxCount* is not supported in a PubTrans 4 environment.
withDepartureMaxCount	Integer	Yes	9999	

11.1.3 Response data

Index	Table Name	Page	Opt?	Note
0	Departure	80	No	

11.2 Get Direct Journeys between Stops

Get timetable for direct journeys between two stops including real time information if is available.

Service Journey Deviations is returned when there is a deviation that is concerned for the whole journey. Departure Deviations is returned if there is a deviation that concerns the departure at fromStopAreaGid where the Arrival Deviations is returned if there is a deviation that concerns arrivals at toStopAreaGid.

No arrival or departure deviation is returned for arrival and departures between fromStopAreaGid and toStopAreaGid.

11.2.1 Usage

This method can only be used in interactive mode.

11.2.2 Request parameters

Name	Data type	Opt?	Default	Description
fromStopAreaGid	Gid	No		The stop to find departures at, i.e. the origin stop.
toStopAreaGid	Gid	No		Direction: a stop that must be called after the origin stop.

Name	Data type	Opt?	Default	Description
forTimeWindowStartDateTime	DateTime	No		The earliest departure time from the origin stop. This must be the local time.
forTimeWindowDuration	Byte	Yes	59 min	The result is limited by both these parameters, whichever is hit first. These filters are applied to the planned departure times from the *fromStopArea*.
withDepartureMaxCount	Integer	Yes	99999	
forLineGids	Gid [1-4]	Yes	All	Filters the result set to only contain journeys on the specified lines. Cannot be combined with forProducts.
forProducts	Code [1-4]	Yes	All	Filters the result set to only contain journeys on lines that is categorised as one of the specified products. Cannot be combined with forLineGids.
purposeOfLineGroupingCode	Code	Yes	PRODUCT	Filters the result set to only contain journeys on lines that belongs to the specified code. Used only in combination with forProducts.

11.2.3 Response data

Index	Table Name	Page	Opt?	Note
0	Direct Journeys between Stops	84	No	
1	Deviation Message Version	93	Yes	
2	Deviation Message Variant	93	Yes	
3	Service Journey Deviation	94	Yes	
4	Departure Deviation	94	Yes	
5	Arrival Deviation	94	Yes	
6	Target Audience	95	Yes	

11.3 Get Calls at Stop

Returns calls (arrivals and departures) at a specific stop area or stop point. This method supports handling of daylight saving time changes per [UM-PT/I/ROI/3].

11.3.1 Usage

This method can be used in interactive mode. If the selected time window is outside (before or after) the production timetable window in PubTrans®, the method returns only data about planned operation.

11.3.2 Request parameters

Name	Data type	Opt?	Default	Description
atStopGid	Gid	No		The stop to find departures at. (StopArea, StopPoint or JourneyPatternPoint)
forTimeWindowStartDateTime	DateTime	No		Selected arrivals and departures must have a target date/time greater or equal to this value.
forTimeWindowDurationMinutes	Byte	No		Selected arrivals and departures must have a target date/time lesser than the time windows start time plus this value.
includeArrivalsTable	Boolean	No		If false, the table will be returned but will be empty.
includeDepartures-Table	Boolean	No		If false, the table will be returned but will be empty.
includeDeviationTables	Boolean	No		If true, six extra tables are returned with deviation information. Only available if the request is within the real-time window in PWS.

11.3.3 Response data

Index	Table Name	Page	Opt?	Note
0	Dated Service Journey	86	No	
1	Dated Departure	88	No	Will be empty if parameter *include-DeparturesTable* is false.
2	Dated Arrival	91	No	Will be empty if parameter *include-ArrivalsTable* is false.
3	Deviation Message Version	93	Yes	
4	Deviation Message Variant	93	Yes	
5	Service Journey Deviation	94	Yes	
6	Departure Deviation	94	Yes	
7	Arrival Deviation	94	Yes	
8	Target Audience	95	Yes	

12 Vehicle Monitoring Service

The stop monitoring web service provides vehicle centric data about departures and arrivals at a certain stop.

12.1 Get Calls for Service Journey

Return calls (arrivals and departures) for a specific service journey. This method supports handling of daylight saving time changes according to [UM-PT/I/ROI/3].

12.1.1 Usage

This method can be used in interactive mode. If the selected time window is outside (before or after) the production timetable window in PubTrans®, the method returns only data about planned operation.

12.1.2 Request parameters

Name	Data type	Opt?	Default	Description
forServiceJourneyIdOrGid	Gid	No		The global id for the service journey
atOperatingDate	Date	No		The operating date for the service journey.
atStopGid	Gid	Yes		If provided, limits the result set to contain only calls at the stop.
includeArrivalsTable	Boolean	No		If false, the table will be returned but will be empty.
includeDeparturesTable	Boolean	No		If false, the table will be returned but will be empty.
includeDeviationTables	Boolean	No		If true, six extra tables are returned with deviation information. Only available if the request is within the real-time window in PWS.

12.1.3 Response data

Index	Table Name	Page	Opt?	Note
0	Dated Service Journey	86	No	
1	Dated Departure	88	No	Will be empty if parameter *include-DeparturesTable* is false.
2	Dated Arrival	91	No	Will be empty if parameter *include-ArrivalsTable* is false.
3	Deviation Message Version	93	Yes	
4	Deviation Message Variant	93	Yes	
5	Service Journey Deviation	94	Yes	

Index	Table Name	Page	Opt?	Note
6	Departure Deviation	94	Yes	
7	Arrival Deviation	94	Yes	
8	Target Audience	95	Yes	

12.2 Get Call

Returns information about a specific call (arrival and departure).

This method supports handling of daylight saving time changes according to [UM-PT/I/ROI/3].

Not supported in a PubTrans® 4 environment.

12.2.1 Usage

This method can be used in interactive mode. If the selected time window is outside (before or after) the production timetable window in PubTrans®, the method returns only data about planned operation.

12.2.2 Request parameters

Name	Data type	Opt?	Default	Description
forDatedServiceJourneyIdOrGid	Numeric(16)	No		
atOperatingDayDate	Date	No		The operating day of the service journey.
atSequenceNumber	Integer	No		The calls sequence number on the service journey.

12.2.3 Response data

Index	Table Name	Page	Opt?	Note
0	Dated Service Journey	86	No	
1	Dated Departure	88	No	
2	Dated Arrival	91	No	
3	Deviation Message Version	93	Yes	
4	Deviation Message Variant	93	Yes	
5	Service Journey Deviation	94	Yes	
6	Departure Deviation	94	Yes	
7	Arrival Deviation	94	Yes	
8	Target Audience	95	Yes	

Part 2: Response Reference

This part contains detailed descriptions of the data returned in the response of a web service method call. Which data that are returned by a specific method are documented in

13 Organisational data

13.1 Organisational Unit

The organisational units are sorted in ascending order after the field *Name*. If another sort order is required, the consuming application has to redo the sorting.

This table contains all organisational units referred by the subsequent tables in this response. Therefore, the client application should apply additional filtering to this table, for instance by only showing records that is referred to by a subsequent table in the dataset.

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Unique id of the object. Can be used when caching and refreshing info about this object.
Code	String(1-8)	No	Alphanumeric code for the organisation unit.
Name	String(1-50)	No	Name of organisation.
MailAddressName	String(1-50)	Yes	Postal delivery address.
VisitingAddressName	String(1-50)	Yes	Visiting street address.
Postcode	String(1-10)	Yes	
PostofficeName	String(1-50)	Yes	
CountryName	String(1-50)	Yes	
PhoneNumber	String(1-50)	Yes	Phone number to main organisational unit.
CoordinateSystemName	String(1-50)	Yes	The name of the reference system used for the location coordinates below.
CentroidNorthingCoordinate	String(1-30)	Yes	The location of the organisation unit to be shown on a map.
CentroidEastingCoordinate	String(1-30)	Yes	
ExistsFromDate	Date	No	
ExistsUptoDate	Date	Yes	Null means infinite existence.

13.2 Transport Authority

The table contains the transport authorities with reference to which organisational unit it is.

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Unique id of the object. Can be used when caching and refreshing info about this object.
Gid	Numeric(16)		
OrganisationId	Numeric(16)	No	Refers to field *Id* in table *Organisational Unit*.
TimetableReleasedForPublicUseUptoDate	Date	No	Timetables available in PubTrans for this transport authority are not for public use from this date.

13.3 Contractor

The table contains the contractors with reference to which organisational unit it is.

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Unique id of the object. Can be used when caching and refreshing info about this object.
Gid	Numeric(16)		
IsOrganisationId	Numeric(16)	No	Refers to field *Id* in table *Organisational Unit*.
IsPromotedToTransportAuthorityId	Numeric(16)	No	Refers to field *Id* in table *Transport Authority*.

13.4 Contract

The table contains Contract with reference to which organisational unit and Transport Authority it belongs to

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Unique id of the object. Can be used when caching and refreshing info about this object.
ReferenceName	String(1-50)	No	
Name	String(1-50)	Yes	
Note			
IsPromotedToTransportAuthorityId	Numeric(16)	No	Refers to field *Id* in table *Transport Authority*.
IsPromotedToContractorOrganisationId	Numeric(16)	No	Refers to field *Id* in table *Organisational Unit*.
ExistFromDate	Date	No	
ExistUpToDate	Date	Yes	Null means infinite existence.

13.5 Line In Contract

The table contains Line In Contract with reference to which Contract and Transport Authority it belongs to

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Unique id of the line, not necessary unique in this result. Can be used when caching and refreshing info about this line.
Gid	Numeric(16)	No	Gid for the line
Name	String(1-50)	Yes	Name of the line

Field name	Data type	Null?	Description
Number	Numeric(4)	No	Technical number of the line. Note that this value is not unique in PubTrans.
Designation	String(1-8)	No	Public identifier of the line to be used in all presentation.
TransportModeCode	String(1-8)	No	Default transport mode code for the line.
LevelOfOperationCode	String(1-8)	No	For valid values, see [IS-PT/I/DOI/4], data type *typLevelOfOperationCode*.
IsDefinedByTransportAuthorityId	Numeric(16)	No	Refers to field *Id* in table *Transport Authority*.
IsPromotedInContractId	Numeric(16)	No	Refers to field *Id* in table *Contract*.
ControlsLineByOrganisationalUnitId	Numeric(16)	Yes	Refers to field *Id* in table OrganisationalUnit
ExistFromDate	Date	No	
ExistUpToDate	Date	Yes	Null means infinite existence.

13.6 Service Function For Organisational Unit

The table contains the specific service functions with reference to which organisational unit it is.

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Unique id of the object. Can be used when caching and refreshing info about this object.
TypeOfServiceFunctionCode	String(1-8)	No	For valid values, see [IS-PT/I/DOI/4], data type *typServiceFunctionCode*.
TypeOfServiceFunctionName	String(1-50)	No	Name of service function.
Note	String(1-255)	Yes	Information to be presented about this service function, for instance opening hours of a lost and found office.
IsUsedByOrganisationalUnitId	Numeric(16)	No	Refers to field *Id* in table *Organisational Unit*. This is the organisational unit for which the service function is carried out.
IsDelegatedByOrganisationalUnitId	Numeric(16)	No	Refers to field *Id* in table *Organisational Unit*. This is the organisational unit that has the administrative responsibility for the delegation.

Field name	Data type	Null?	Description
IsDelegatedToOrganisationalUnitId	Numeric(16)	No	Refers to field *Id* in table *Organisational Unit*. This is the organisational unit that carries out the service function.
PhoneNumber	String(1-50)	Yes	Phone number to service function.

13.7 Communication Address For Organisational Unit

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Unique id of the object. Can be used when caching and refreshing info about this departure.
ProtocolCode	String(1-8)	No	Type of communication address: For valid values, see [IS-PT/I/DOI/4], data type *typProtocolCode*. This code needs to be mapped to a language resource that translates it into a descriptive name in the preferred presentation language(s).
Name	String(1-50)	No	The actual communication address, e.g. a phone number, email address etc.
IsUsedByOrganisationalUnitId	Numeric(16)	No	Refers to field *Id* in table *Organisational Unit*.

14 Topology

14.1 Zone

The zones areas are sorted in ascending order after the field *Gid*. If another sort order is required, the consuming application has to redo the sorting.

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Unique id of the object. Can be used when caching and refreshing info about this object..
Gid	Numeric(16)	No	Zone Gid.
Code	String(1-8)	Yes	Short identifier of the zone.
Number	Integer	No	The number of the zone. Unique within the transport authority at a specific point in time.
Name	String(1-50)	Yes	The name of the zone.
ShortName	String(1-16)	Yes	The short name of the zone.
TypeCode	String(1-8)	No	The type code of the zone. See [IS-PT/I/DOI/4] for explanation of values of *typZoneTypeCode*.
TransportAuthorityCode	String(1-8)	No	The transport authority that manages the zone.
IsDefinedByTransportAuthorityId	Numeric(16)	No	Reference to the TransportAuthorityId object
IsDefinedByTransportAuthorityGId	Numeric(16)	No	Reference to the TransportAuthorityGid
CoordinateSystemName	String(1-50)	Yes	The name of the coordinate system used for the coordinates for this stop area.
CentroidNorthingCoordinate	String(1-30)	Yes	
CentroidEastingCoordinate	String(1-30)	Yes	
ExistsFromDate	Date	No	
ExistsUptoDate	Date	Yes	Null means infinite existence.

14.2 Line in Zone

Field name	Data type	Null?	Description
Gid	Numeric(16)	No	Gid for the line.
Name	String(1-50)	Yes	The name of the line.

Field name	Data type	Null?	Description
Number	Numeric(4)	No	Technical number of the line. Note that this value is not unique in PubTrans.
Designation	String(1-8)	No	Public identifier of the line to be used in all presentation.
DisplayOrder	Integer	No	Value that specify the display order when sorting the lines for public presentation.
TransportAuthorityCode	String(1-8)	No	The transport authority for the line.
OperatorCode	String(1-8)	No	The operator of the line.
ExistsFromDate	Date	No	The first traffic date for this line in the zone(s) within the interval specified by the parameters fromDate and uptoDate.
ExistsUptoDate	Date	Yes	The last traffic date for this line in the zone(s) + 1 day within the interval specified by the parameters fromDate and uptoDate.
OperatesZoneId	Numeric(16)	No	Reference to column *Id* in table *Zone*.

15 Stops

15.1 Stop Area

The stop areas are sorted in ascending order after the field *Gid*. If another sort order is required, the consuming application has to redo the sorting.

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Unique id of the object. Can be used when caching and refreshing info about this departure.
Gid	Numeric(16)	No	Stop Area Gid.
Code	String(1-8)	Yes	Short identifier for some stop areas.
Number	Integer	No	The number of the stop area unique within the transport authority.
Name	String(1-50)	No	Full name of stop area.
ShortName	String(1-16)	Yes	Short version of name of stop area. Can be used when presentation space is limited.
TypeCode	String(1-8)	No	AIRPORT, BUSTERM, FERRYBER, METROSTN, RAILWSTN, TRAMSTN, SHIPBER, TAXITERM, UNKNOWN
ExistsFromDateTime	Date	No	
ExistsUptoDateTime	Date	Yes	Last day of validity. May be NULL if no last day is set.
DefaultInterchangeDuration	TimeSpan	Yes	
InterchangePriority	Numeric(1-2)	Yes	
CoordinateSystemName	String(1-50)	Yes	The name of the coordinate system used for the coordinates for this stop area.
CentroidNorthingCoordinate	String(1-30)	Yes	
CentroidEastingCoordinate	String(1-30)	Yes	

15.2 Stop Point

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Unique id of the object. Can be used when caching and refreshing info about this departure.
Gid	Numeric(16)	No	Stop Area Gid.

Field name	Data type	Null?	Description
Name	String(1-50)	No	Full name of stop point.
ShortName	String(1-16)	Yes	Short version of name of stop point. Can be used when presentation space is limited.
Designation	String(1-4)	No	The public identification of a stop point within a stop area.
LocalNumber	Integer	No	Local number within a stop area.
RoadElementName	String(1-50)	Yes	Name of road where the stop point is located.
IsJourneyPatternPointId	Numeric(16)	No	Reference to the journey pattern point object.
IsJourneyPatternPointGid	Numeric(16)	No	Reference to the journey pattern point object.
IsIncludedInStopAreaId	Numeric(16)	No	Reference to column *Id* in table *StopArea*.
IsIncludedInStopAreaGid	Numeric(16)	No	Reference to the stop area.
TypeCode	String(1-8)	No	BUSSTOP, REFUGE, PLATFORM, TRACK, GATE, PIER, ENTRANCE, EXIT, UNKNOWN
ExistsFromDateTime	Date	No	
ExistsUptoDateTime	Date	Yes	Last day of validity. May be NULL if no last day is set.
ForBoardingYesNo	Booelan	No	True if boarding is permitted at the stop point.
OrientationDegrees	Numeric(5,2)	Yes	0.00 to 360.00 degrees.
LengthMeters	Numeric(12,2)	Yes	Available space for vehicles at the stop point.
InsideStationYesNo	Boolean	Yes	
IndoorsYesNo	Boolean	Yes	
FictitiousYesNo	Boolean	No	
MainDirectionName	String(1-50)	Yes	
CoordinateSystemName	String(1-50)	Yes	The name of the coordinate system used for the coordinates for this stop point.
LocationNorthingCoordinate	String(1-30)	Yes	
LocationEastingCoordinate	String(1-30)	Yes	

15.3 Line at Stop

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Id for the line.

Field name	Data type	Null?	Description
Gid	Numeric(16)	No	Gid for the line.
Name	String(1-50)	Yes	The name of the line.
Number	Numeric(4)	No	Technical number of the line. Not that this value is not unique in PubTrans.
Designation	String(1-8)	No	Public identifier of the line to be used in all presentation.
DisplayOrder	Integer	No	Value that specify the display order when sorting the lines for public presentation.
DefaultTransportModeCode	String(1-8)	No	The code for default transport mode of the line. Transport mode for individual service journeys may differ from this value. Valid codes are BUS, FERRY, METRO, SHIP, TAXI, TRAIN, and TRAM.
TransportAuthorityCode	String(1-8)	No	The code for the transport authority that manages the line.
TransportAuthorityName	String(1-50)	No	The name for the transport authority that manages the line.
OperatorId	Numeric(16)	No	The Id for the operator of the line.
OperatorGid	Numeric(16)	No	The Gid for the operator of the line.
OperatorCode	String(1-8)	No	The code for the operator of the line.
OperatorName	String(1-50)	No	The name for the operator of the line.
ExistsFromDate	Date	No	
ExistsUptoDate	Date	Yes	Null means infinite existence.
OperatesStopAreaId	Numeric(16)	No	Reference to column *Id* in table *StopArea*.
OperatesStopPointId	Numeric(16)	No	Reference to column *Id* in table *StopPoint.*

15.4 Line Destination at Stop

Field name	Data type	Null?	Description
DestinationDisplayId	Numeric(16)	No	Unique Id for each destination display.
LineGid	Numeric(16)	No	Gid for the line.
IsOnDirectionOfLineGid	Numeric(16)	No	Gid for the direction.
IsJourneyPatternPointGid	Numeric(16)	No	Point in pattern for which the destination display concerns.

Field name	Data type	Null?	Description
LineDesignation	String(1-8)	No	Public identifier of the line to be used in all presentation.
PrimaryDestinationName	String(1-50)	No	
PrimaryDestinationShortName	String(1-16)	No	
SecondaryDestinationName	String(1-50)	No	
SecondaryDestinationShortName	String(1-16)	No	
StopAreaId	Numeric(16)	No	Reference to column *Id* in table *StopArea*.
StopPointId	Numeric(16)	No	Reference to column *Id* in table *StopPoint*.
LineId	Numeric(16)	No	Reference to column *Id* in table *Line*.

15.5 Site

The sites are sorted in ascending order after the field *Gid*. If another sort order is required, the consuming application has to redo the sorting.

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Unique id of the object. Can be used when caching and refreshing info about this departure.
Gid	Numeric(16)	No	Unique at a specific point in time.
Number	Integer	No	The number of the site is unique within *IsDefinedByTransportAuthorityId* at a specific point in time.
Name	String(1-50)	No	Full name of Site.
ShortName	String(1-16)	Yes	Short version of name of the site. Can be used when presentation space is limited.
TypeCode	String(1-8)	No	SITE, STOPAREA, NEIGHBOU, ADDRESS, STREET, OTHER
IsDefinedByTransportAuthorityId	Numeric(16)	No	Reference to the TransportAuthorityId object
IsDefinedByTransportAuthorityGid	Numeric(16)	Yes	Reference to the TransportAuthorityGid object
Abbreviation	String (1-8)	Yes	
InternalComment	String(1-255)	Yes	
PublicNote	String(1-255)	Yes	

Field name	Data type	Null?	Description
CoordinateSystemName	String(1-50)	Yes	The name of the coordinate system used for the coordinates for this stop area.
CentroidNorthingCoordinate	String(1-30)	Yes	
CentroidEastingCoordinate	String(1-30)	Yes	
ExistsFromDateTime	Date	No	
ExistsUptoDateTime	Date	Yes	Last day of validity. May be NULL if no last day is set.

15.6 Site Neighbouring Stop Area

This table describes the relation between sites and stop areas. This table can therefore be used to resolve which sites a stop area is related to.

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Unique id of the object.
IsSiteId	Numeric(16)	No	Reference to the Site object.
IsSiteGid	Numeric(16)	No	Reference to the Site object.
IsNeighbouringStopAreaGid	Numeric(16)	Yes	Reference to the StopArea object.

15.7 Journey Pattern Point In Zone

This table describes the relation between zones and journey pattern points. This table can therefore be used to resolve which zones a stop point is located in.

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Unique id of the object.
IsJourneyPatternPointGid	Numeric(16)	No	Reference to the JourneyPatternPointGid object.
IsInZoneGid	Numeric(16)	No	Reference to the ZoneGid object.
LastModifiedUtcDateTime	Date	No	
ExistsFromDate	Date	No	
ExistsUptoDate	Date	Yes	Last day of validity. May be NULL if no last day is set.

16 Lines

16.1 Line

The departures are sorted in ascending order after the field *Gid*. If another sort order is required, the consuming application has to redo the sorting. Because a line designation (= public identifier) is not necessarily unique, the result may contain more that one line with the same designation. Use the field *Gid* or *Is For Line Gid* to distinguish the lines.

Field name	**Data type**	**Null?**	**Description**
Id	Numeric(16)	No	Unique id of the object. Can be used when caching and refreshing info about this departure.
Gid	Numeric(16)	No	Gid for the line.
Name	String(1-50)	Yes	The name of the line.
Number	Numeric(4)	No	Technical number of the line. Not that this value is not unique in PubTrans. Do not use this value in public information.
Designation	String(1-8)	No	Public identifier, i.e the public line “number”, of the line to be used in all presentation.
DefaultTransportModeCode	String(1-8)	No	The code for default transport mode of the line. Transport mode for individual service journeys may differ from this value. Valid codes are BUS, FERRY, METRO, SHIP, TAXI, TRAIN, and TRAM.
TransportAuthorityCode	String(1-8)	No	The code for the transport authority that manages the line.
TransportAuthorityName	String(1-50)	No	The name of the transport authority that manages the line.
DisplayOrder	Integer	No	Value that specify the display order when sorting the lines for public presentation.
ExistsFromDate	Date	No	The first date the object version is valid.
ExistsUptoDate	Date	Yes	The date from when the object version becomes invalid. NULL means infinite validity.

16.2 Group Of Lines

The returned table includes both explicit Group Of Lines from PubTrans (defined in the GroupOfLines table in DOI) as well as goups of lines deducted from other data:

- Zones, which are a type of grouping where the line members are deducted from which journey patterns has points in the zone. The purpose of line grouping are the same value as the code for type of zone.
- 'TRANMODE' which are a type of grouping deducted from the line default transport mode. There will be one group per transport mode and transport authority.

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Unique id of the object. Can be used when caching and refreshing info about this departure. Note that Group Of Lines created from deducted data (zones and transport mode) are given fictive IDs since they are not present as Group Of Line objects in PubTrans.
TransportAuthorityCode	String(1-8)	Yes	The organisation code for the managing transport authority.
Code	String(1-8)	Yes	The code for the Group Of Line.
Name	String(1-50)	No	Name of the grouping, for instance name of geographic area, sales district or city. Can be combined by TransportAuthorityCode and Name depending on configuration.
PurposeOfGroupingCode	String(1-8)	No	For valid values, see [IS-PT/I/DOI/4], data type *typLineGroupingCode*. Note that additional values can be added as customisations that need to be considered by consumers of this field.
ExistsFromDate	Date	No	The first date the object version is valid.
ExistsUptoDate	Date	Yes	The date from when the object version becomes invalid. NULL means infinite validity.

16.3 Line In Group Of Lines

Note that a line can be in several groups, also in several groups with the same purpose code, for example a line that operates across several geographical districts.

Field name	Data type	Null?	Description
Id	Numeric(16)	No	For PubTrans-objects: Unique id of the object. Can be used when caching and refreshing info. For lines included in "fictive" Group Of Lines not present in PubTrans Id is set to 0. View section 16.2 – Group Of Line for more details.
IsLineId	Numeric(16)	No	Refers to field *Id* in table *Line*.

Field name	Data type	Null?	Description
IsLineGid	Numeric(16)	No	
IsInGroupOfLinesId	Numeric(16)	No	Refers to field *Id* in table *Group of Lines*.
ExistsFromDate	Date	No	The first date the object version is valid.
ExistsUptoDate	Date	Yes	The date from when the object version becomes invalid. NULL means infinite validity.

16.4 Direction Of Line

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Unique id of the object. Can be used when caching and refreshing info about this departure.
Gid	Numeric(16)	No	Gid for the direction of line.
DirectionCode	String(1)	No	“1” or “2”.
Name	String(1-50)	Yes	The main heading for the direction of line.
IsOnLineId	Numeric(16)	No	Refers to the field *Id* in the table *Line*.
ExistsFromDate	Date	No	The first date the object version is valid
ExistsUptoDate	Date	Yes	The data from when the object version becomes invalid. NULL means infinite validity.

16.5 Operated Lines

The table contains the lines with reference to the contractor(s) that operate the line. Note that a line can be operated by more than one contractor.

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Unique id of the line, not necessary unique in this result. Can be used when caching and refreshing info about this line.
Gid	Numeric(16)	No	Gid for the line.
Name	String(1-50)	Yes	The name of the line.
Number	Numeric(4)	No	Technical number of the line. Not that this value is not unique in PubTrans.
Designation	String(1-8)	No	Public identifier of the line to be used in all presentation.

Field name	Data type	Null?	Description
DisplayOrder	Integer	No	Value that specify the display order when sorting the lines for public presentation.
LevelOfOperationCode	String(1-8)	No	For valid values, see [IS-PT/I/DOI/4], data type *typLevelOfOperationCode*.
IsDefinedByTransportAuthorityId	Numeric(16)	No	Refers to field *Id* in table *Transport Authority*.
IsOperatedByContractorId	Numeric(16)	No	Refers to field *Id* in table *Contractor*.
IsControlledByOrganisationalUnitId	Numeric(16)	Yes	Refers to field *Id* in table *Organisational Unit*. Null if the line is not controlled by an organisational unit.
ExistsFromDate	Date	No	The first date the object version is valid.
ExistsUptoDate	Date	Yes	The date from when the object version becomes invalid. NULL means infinite validity.
ContractorOperatesLineFromDate	Date	No	The first date the object is operating the line.
ContractorOperatesLineUptoDate	Date	Yes	The date when the object stops operating the line. NULL means infinite validity.

16.6 Stop on Line

Field name	Data type	Null?	Description
StopAreaId	Numeric(16)	No	
StopAreaGid	Numeric(16)	No	The global number of the stop area.
StopAreaName	String(1-50)	Yes	Short identifier for some stop areas.
StopAreaShortName	String(1-16)	Yes	Short version of name of stop area. Can be used when presentation space is limited.
StopAreaAbbreviation	String(1-8)	Yes	
StopPointId	Numeric(16)	No	
StopPointGid	Numeric(16)	No	The globally unique number of the stop area
StopPointName	String(1-50)	No	Full name of stop point.
StopPointShortName	String(1-16)	Yes	Short version of name of stop point. Can be used when presentation space is limited.
StopPointDesignation	String(1-4)	Yes	

Field name	Data type	Null?	Description
StopPointLocalNumber	Numeric(3)	No	
JourneyPatternPointGid	Numeric(16)	No	The globally unique journey pattern point .number of the stop point.
TransportAuthorityCode	String(1-8)	No	The code for the transport authority that owns the stop point.
ExistsFromDate	Date	No	The first date the object version is valid.
ExistsUptoDate	Date	Yes	The date from when the object version becomes invalid. NULL means infinite validity.
IsOnLineId	Numeric(16)	No	Refers to the field *Id* in the table *Line*.
IsOnLineGid	Gid	No	The globally unique number for the line.
IsOnDirectionOfLineGid	Gid	No	The globally unique number for the direction of line.

16.7 Line Variants

Field name	Data type	Null?	Description
LineGid	Gid	No	The globally unique number for the line.
LineVariantCodeId	Numeric(4)	No	Line variant number. Serial number starting from 1 unique within each line. This number does not change.
LineVariantShortName	String(1-50)	No	Line variant description containing stop area abbreviation on start- and end stop area.*, **
LineVariantName	String(1-1000)	No	Line variant description including full stop area names.*
StartStopAreaGid	Gid	No	The globally unique number for the StopArea.
EndStopAreaGid	Gid	No	The globally unique number for the StopArea.

*) Intermediate stop areas and special signs indicating boarding- and alighting restrictions are added when needed to make the description unique:

'*' Always stop
'+' Stop when necessary (alighting/boarding)
'_' Stop for alighting
'-' Stop for boarding
'.' No boarding/alighting

The stop area abbreviation and name used in LineVariantShortName and LineVariantName is the abbreviation and name that is valid in PubTrans when the sql job run(default once a day) that prepare data for LineVariants. If a StopArea has ceased to be valid, then the last valid version of that StopArea is used (this happens when the timespan covers time before and after a StopArea ceases to be valid).

**) If stop area abbreviation is missing in PubTrans the first letters of the stop area name are used. If stop area name is missing the stop area number is used.

17 General messages

17.1 Deviation Message

This table contains the deviation message header.

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Unique id of the object. Can be used when caching and refreshing info about this object.
DeviationReasonStandardCategoryCode	String(1-8)	No	For explanation, see [UM-PT/I/ROI/3], data type *typDeviationReasonStandardCategoryCode*.
ValidFromDateTime	DateTime	No	First day of validity for message
ValidUpToDateTime	DateTime	Yes	First day when message is invalid
PublishFromDateTime	DateTime	No	
PublishUptoDateTime	DateTime	Yes	
PublicNote	String(1-255)	Yes	
IsOnLineGid	Numeric(16)	Yes	OBSOLETE: Always null. Use table Publication Scope to find what lines a deviation message concerns. This column will be dropped in next main release of PWS.
ImportanceLevel	Numeric(1)	Yes	For explanation, see [UM-PT/I/ROI/3], chapter “Presenting Deviations” about “Priority”.
InfluenceLevel	Numeric(1)	Yes	
UrgencyLevel	Numeric(1)	Yes	

17.2 Deviation Message Variant

This table contains deviation message variants, i.e. different versions for different presentation purposes and languages. Only deviation message variants, that are intended to the public, are approved for publication and is valid at the time of the request is returned. See [UM-PT/I/ROI/3] for detailed information about Deviation Messages.

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Unique id of the object. Can be used when caching and refreshing info about this object.
IsPartOfDeviationMessageId	Numeric(16)	No	The field refers to field *Id* in the *Deviation Message* table.

Field name	Data type	Null?	Description
Content	Text(1-3600)	No	Actual text or hyperlink to content.
ContentTypeShortName	String(1-16)	No	Text/plan, text/html or urn/mp3. Other types can be added.
UsageTypeShortName	String(1-16)	No	An additional hint of for what media type and media limitation the content is adapted for.
LanguageCode	String(1-8)	Yes	A language code according ISO 639.1, or NULL if language invariant.
IsPublic	Boolean	No	If true, this message variant is intended and approved to be presented to the public.

18 Arrivals and Departures

18.1 Departure

The departures are sorted in ascending order after the field *Target DateTime*. If another sort order is required, the consuming application has to redo the sorting.

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Unique id of the departure. Can be used when caching and refreshing info about this departure.
TransportMode	String(1-8)	Yes	Bus, tram, metro, train, ferry, ship, taxi or unspecified (null).
ProductName	String(1-50)	Yes	Name of product under which the line is marketed.
ProductCode	String(1-8)	Yes	Code of product under which the line is marketed.
SymbolName	String(1-50)	Yes	Name of symbol to identify the product or line.
LineDesignation	String(1-8)	No	Identifier of the line known to the public, i.e. line number and similar. Note that this value is not unique and cannot be used as an identifier within your application. Use the lines GID instead for this purpose.
LineGid	Gid	No	The globally unique identifier of the line for the departure.
DirectionOfLineGid	Gid	No	The globally unique identifier of the direction of line for the departure.
OperatingDayDate	Date	No	The operating day date for the service journey that performs this departure.
ServiceJourneyGid	Gid	No	The GID for the service journey that performs this departure. The GID is unique within the operating day.
TransportAuthorityId	Numeric(16)	Yes	
TransportAuthorityGid	Gid	Yes	
TransportAuthorityCode	String(1-8)	No	Refers to field *Code* in table *Organisational Unit*.
TransportAuthorityName	String(1-50)	No	Name of transport authority that provides the service for the departure.

Field name	Data type	Null?	Description
ContractorId	Numeric(16)	Yes	
ContractorGid	Gid	Yes	
ContractorCode	String(1-8)	No	Refers to field *Code* in table *Organisational Unit*.
ContractorName	String(1-50)	No	Name of the contractor that operates the service for the departure.
PrimaryDestinationName	String(1-50)	No	Text describing the heading of the vehicle as presented to the passengers at the specific stop. The short text version can be used to optimise the usage of limited space to present destination.
PrimaryDestinationShortName	String(1-16)	Yes	
SecondaryDestinationName	String(1-50)	No	Text describing supplementary heading of the vehicle as presented to the passengers at the specific stop. A supplementary heading can be a via-place, a precision of main heading or a type of service. The short text version can be used to optimise the usage of limited space to present destination.
SecondaryDestinationShortName	String(1-16)	Yes	
ExpectedToBeMonitored	Boolean	No	If true, real-time is expected for this departure; otherwise only timetabled data is available.
PresentationType	Integer	No	See 18.1.1 below for explanation of this field.
TimetabledEarliestDateTime	DateTime	No	The long term departure time.
TargetDateTime	DateTime	No	The short term departure time. Defaults to *Timetabled Earliest DateTime*. May change in real-time. This is the time that shall be used for presentation to the public/passengers.
EstimatedDateTime	DateTime	Yes	If available, is the most likely departure time. Do not use this time for presentation to the public/passengers.
ObservedDateTime	DateTime	Yes	If available, is the actual time departed.
IsAtStopPointGid	Gid	No	The stop point GID for where the call is targeted to take place. Usage, see section 18.1.1 Stop Point Designation.

Field name	Data type	Null?	Description
IsPlannedAtStopPointGid	Gid	No	The stop point GID for where the call is planned to take place. Usage, see section 18.1.1 Stop Point Designation.
IsAtStopPointDesignation	String(1-4)	Yes	The public identification of the stop point within the stop area where the call is targeted to take place. Usage, see section 18.1.1 Stop Point Designation.
IsPlannedAtStopPointDesignation	String(1-4)	Yes	The public identification of the stop point within the stop area where the call is planned to take place. Usage, see section 18.1.1 Stop Point Designation.

18.1.1 Stop Point Designation

Different values of *IsAtStopPointGid* and *IsPlannedAtStopPointGid* indicates that the call has been moved to a stop point other that the planned one. If this is the case, it is recommended to present the departure a redirection note from the planned to the target stop point, using the designations.

18.1.2 Arrival Presentation Type

Indicates how a arrival should be presented. This value represents an aggregate of the richer state model PubTrans provides through ROI. The reason is to make it easier for a web service consumer to decide how a arrival should be presented. For information about the full state model and full functionality in PubTrans, see [UM-PT/I/DOI/3].

Presentation Type	Presentation	Remark
0 = Normal	Show using the *Target DateTime*.	Time could either be absolute (hh:mm) or relative (minutes to arrival). Fractions of minutes shall be rounded down to agreed resolution, normally whole minute.
1 = Cancelled	Show as **cancelled** using the *Timetabled DateTime*.	If it is not possible to show a cancelled arrival as cancelled, one alternative is to not showing it at all.
2 = Replaced	Show as **cancelled** using the *Timetabled DateTime* and remark that a replacement is available.	A reference to the replacement vehicle journey is currently not available.
3 = Unreliable	Show using the *Target DateTime* and remark that the time is unreliable.	
4 = Due	Show using the *Target DateTime* and/or remark that the arrival is due.	That a vehicle is due can be presented in many ways: replacing minutes to arrival with ‘due’, blinking arrival time, sound etc.

Presentation Type	Presentation	Remark
5 = Arrived	Show using the *Target DateTime* and/or remark that the vehicle has arrived.	If current date/time < target date/time, there is time left until arrival; otherwise the vehicle may arrive as soon as possible.
7 = Waiting	Show using the *Target DateTime* and/or remark that the vehicle is waiting for a connecting arrival.	This occurs if the vehicle has to wait for a connection over its target departure time, i.e. when current date/time > planned date/time.
8 = Arrived	Show as departed or not at all.	When showing as arrived, one way to present is to use *Timetabled DateTime* as reference time and the *Observed DateTime* as actual arrival time.
9 = Hidden	Do not show.	

18.1.3 Departure Presentation Type

Indicates how a departure should be presented. This value represents an aggregate of the richer state model PubTrans provides through ROI. The reason is to make it easier for a web service consumer to decide how a departure should be presented. For information about the full state model and full functionality in PubTrans, see [UM-PT/I/DOI/3].

Presentation Type	Presentation	Remark
0 = Normal	Show using the *Target DateTime*.	Time could either be absolute (hh:mm) or relative (minutes to departure). Fractions of minutes shall be rounded down to agreed resolution, normally whole minute.
1 = Cancelled	Show as **cancelled** using the *Timetabled DateTime*.	If it is not possible to show a cancelled departure as cancelled, one alternative is to not showing it at all.
2 = Replaced	Show as **cancelled** using the *Timetabled DateTime* and remark that a replacement is available.	A reference to the replacement vehicle journey is currently not available.
3 = Unreliable	Show using the *Target DateTime* and remark that the time is unreliable.	
5 = At Stop	Show using the *Target DateTime* and/or remark that the vehicle is at stop.	If current date/time < target date/time, there is time left until departure; otherwise the vehicle may depart as soon as possible.
6 = Boarding	Show using the *Target DateTime* and/or remark that the vehicle is boarding.	

Presentation Type	Presentation	Remark
7 = Waiting	Show using the *Target DateTime* and/or remark that the vehicle is waiting for a connecting arrival.	This occurs if the vehicle has to wait for a connection over its target departure time, i.e. when current date/time > planned date/time.
8 = Departed	Show as departed or not at all.	When showing as departed, one way to present is to use *Timetabled DateTime* as reference time and the *Observed DateTime* as actual departure time.
9 = Hidden	Do not show.	

18.2 Direct Journeys between Stops

This table contains planned and actual journeys between stops.

Field name	Data type	Null?	Description
DatedVehicleJourneyId	Numeric(16)	No	Unique id of the object. Can be used when caching and refreshing info about this object.
ServiceJourneyGid	Numeric(16)	No	The Gid of the Service Journey.
OperatingDayDate	Date	No	The operating date of the service journey.
ContractorGid	Gid	Yes	The operator of the journey.
LineDesignation	String(1-8)	No	The public line number.
JourneyNumber	String(1-8)	No	The public service journey number.
DirectionOfLineName	String(1-50)	Yes	The name of direction this direction of the line.
DirectionOfLineDescription	String(1-255)	Yes	The description for this direction of the line.
OriginName	String(1-50)	Yes	The name of the origin for this journey.
OriginShortName	String(1-16)	Yes	The short name the origin for this journey.
OriginPlaceGid	Gid	Yes	The place Gid of the origin.
PrimaryDestinationName	String(1-50)	Yes	Destination texts that is shown to the public on the vehicle at the origin stop.
PrimaryShortDestinationName	String(1-16)	Yes	
PrimaryDestinationPlaceGid	Gid	Yes	The primary destination place Gid.
SecondaryDestinationName	String(1-50)	Yes	Secondary destination text that is shown to the public on the vehicle.
SecondaryDestinationShortName	String(1-16)	Yes	

Field name	Data type	Null?	Description
SecondaryDestinationPlaceGid	Gid	Yes	The place Gid for the secondary destination.
ExpectedToBeMonitored	Boolean	No	If true, real-time is expected for this departure; otherwise only timetabled data is available. See notes below.
DepartureId	Numeric(16)	Yes	The Id in the table CallOnTimedJourneyPattern in PubTrans.
DepartureStopPointGid	Gid	No	Stop Point Gid deducted from the parameter @fromStopAreaGid.
DepartureType	Integer	No	See 18.2.2 for how to use this value.
DepartureSequenceNumber	Integer	No	The sequence number of the departure deducted from the parameter @fromStopAreaGid.
PlannedDepartureDateTime	DateTime	No	The timetabled time for the departure. Deducted from the parameter @fromStopAreaGid.
ObservedDepartureDateTime	DateTime	Yes	NULL if ROI data is not available.
ArrivalId	Numeric(16)	Yes	The Id in the table CallOnTimedJourneyPattern in PubTrans. Deducted from the parameter @toStopAreaGid.
ArrivesToStopPointGid	Gid	No	StopPointGid on this journey deducted from the parameter @toStopAreaGid.
ArrivalType	Integer	No	See 18.2.1 for how to use this value.
ArrivalSequenceNumber	Integer	No	The sequence number of the arrival. Deducted from the parameter @toStopAreaGid.
PlannedArrivalDateTime	DateTime	No	The date and time for the arrival to the stop. Deducted from the parameter @toStopAreaGid.
ObservedArrivalDateTime	DateTime	Yes	NULL if ROI data is not available.

18.2.1 Arrival Types

The table below describes how the Arrival Type shall be used for presentation.

Type	Presentation
0 = No stop	Do not show the arrival time.
1 = Stops, but alighting is not permitted	Do not show the arrival time.
2 = Stop if requested	Show arrival time.
3 = Always stops	Show arrival time.
5 = Flexible alighting from (but not including) previous point and up to and including this point.	Show arrival time.

18.2.2 Departure Types

The table below describes how the Arrival Type shall be used for presentation.

Type	Presentation
0 = No stop	Do not show the departure time.
1 = Stops, but boarding is not permitted	Do not show the departure time.
2 = Stop if requested	Show departure time.
3 = Always stops	Show departure time.
5 = Flexible boarding from this point up to but not including next point.	Show departure time.

18.3 Dated Service Journey

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Unique id of the *Dated Service Journey*. Can be used when caching and refreshing info about this object.
IsDatedVehicleJourneyId	Numeric(16)	No	Unique id of the planned instance.
OperatingDayDate	Date	No	The date to which the service journey is associated in planning.
Gid	ServiceJourneyGid	No	
IsWorkedOnDirectionOfLineGid	DirectionGid	No	This is always the DOI4 type of Gid, regardless of target platform.

Field name	Data type	Null?	Description
LineDesignation	String(1-8)	No	The public identification of the line, usually a line number.
TransportModeCode	String(1-8)	Yes	BUS, TRAM, MENTRO, TRAIN, FERRY, SHIP, TAXI. A NULL value means 'unknown'
TransportAuthorityCode	String(1-8)	No	The code for the transport authority that manages the line.
TransportAuthorityName	String(1-50)	No	The name of the transport authority that manages the line.
ContractorCode	String(1-8)	No	The code of the operator that operates the service journey.
ContractorName	String(1-50)	No	The name of the operator that operates the service journey.
ExpectedToBeMonitoredYesNo	Boolean	No	If true, real-time is expected for this departure; otherwise only timetabled data is available.
IsAssignedToVehicleGid	VehicleGid or VirtualVehicleGid	Yes	The vehicle that is operating this service journey. If null, no vehicle has been assigned yet or information is not available.
State	Integer	No	

Field name	Data type	Null?	Description
PredictionState	Integer	No	State, Prediction State and state for arrivals and departures controls how information shall be presented. This is described in detail chapter "Interpreting States" in [UM-PT/I/ROI/3]. When real time data is not available - State for journeys is by default set to 2 – "Expected". For journeys with advance order condition, State is by default set to 0 – "Not expected". PredictionState is set to 0.
OriginName	String(1-50)	Yes	The name to be used for presentation of from where the service journey origins, i.e. "comes from".
OriginShortName	String(1-16)	Yes	
ProductCode	String(1-8)	Yes	Code of product under which the line is marketed.
ExposedInPrintMediaYesNo	Boolean	Yes	Describes (true/false) if the journey should be announced to the public
DynamicMediaInformPassengersCondition-Code	String(1-8)	Yes	Describes under which conditions passengers should be informed: ALWAYS ONLYIF_O (ordered) ONLYIF_S (signed on) NEVER

18.4 Dated Departure

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Id of the *Dated Departure*. Unique in combination with *IsOnDatedServiceJourneyId.*
IsOnDatedServiceJourneyId	Numeric(16)	No	Refers to field *Id* in table *Dated Service Journey*.
JourneyPatternSequenceNumber	Integer	No	The sequence number of the Departure, counted from start of the service journey.
IsTimetabledAtJourneyPatternPointGid	Gid	No	The journey pattern point where the call is planned to take place.
IsTargetedAtJourneyPatternPointGid	Gid	No	The journey pattern point where the call is targeted to take place. Defaults to *IsTimetabledAtJourneyPatternPointGid.*
WasObservedAt-JourneyPatternPointGid	Gid	Yes	The journey pattern point where the call actually took place.
TimetabledEarliestDateTime	DateTime	No	The long term planned departure date/time.
TargetDateTime	DateTime	No	The short term planned departure date/time. Defaults to *TimetabledEarliestDateTime*. May change in real-time. This is the time that shall be used for presentation to the public/passengers.
EstimatedDateTime	DateTime	Yes	The most likely arrival date/time. If available, it can be used for presentation to the public/passengers in preference of *Target Date Time*. Only available in response from real time methods, otherwise NULL.
ObservedDateTime	DateTime	Yes	The actual date/time of arrival. A none null value indicates that the vehicle has arrived at the stop. Only available in response from real time methods, otherwise NULL.

Field name	Data type	Null?	Description
State	Integer	No	States for service journeys, arrivals and departures controls how information shall be presented. This is described in detail chapter “Interpreting States” in [UM-PT/I/ROI/3]. When real time data is not available - state is by default set to 2 – “Expected”.
Type	Integer	No	See section 18.2.2 above.
ProductName	String(1-50)	Yes	
LineDesignation	String(1-8)	No	The public identification of the line, usually a line number.
PrimaryDestinationName	String(1-50)	No	Text describing the heading of the vehicle as presented to the passengers at the specific stop. The short text version can be used to optimise the usage of limited space to present destination.
PrimaryDestinationShortName	String(1-16)	Yes	
SecondaryDestinationName	String(1-50)	Yes	Text describing supplementary heading of the vehicle as presented to the passengers at the specific stop. A supplementary heading can be a via-place, a precision of main heading or a type of service. The short text version can be used to optimise the usage of limited space to present destination.
SecondaryDestinationShortName	String(1-16)	Yes	
SecondaryDestinationType	String(1)	No	C Continues: passengers must transfer at primary destination to reach secondary destination. E End station: usually a precision of the primary destination or terminus of the service journey. M Message. T Transfer: passengers must transfer at the secondary destination to continue to primary destination. U Undefined. V Via: Service journey passes secondary destination on its way to primary destination.
SymbolName	String(1-50)	Yes	

Field name	Data type	Null?	Description
PresentationType	Integer	No	See 18.1.1 above for explanation of this field.

18.5 Dated Arrival

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Unique id of the Arrival. Can be used when caching and refreshing info about this departure.
IsOnDatedServiceJourneyId	Numeric(16)	No	Refers to field *Id* in table *Dated Service Journey*.
JourneyPatternSequenceNumber	Integer	No	The sequence number of the Arrival, counted from start of the service journey.
IsTimetabledAtJourneyPatternPointGid	Gid	No	The journey pattern point where the call is planned to take place.
IsTargetedAtJourneyPatternPointGid	Gid	No	The journey pattern point where the call is targeted to take place. Defaults to *IsTimetabledAtJourneyPatternPointGid.*
WasObservedAt-JourneyPatternPointGid	Gid	Yes	The journey pattern point where the call actually took place.
TimetabledLatestDateTime	DateTime	No	The long term planned arrival date/time.
TargetDateTime	DateTime	No	The short term planned arrival date/time. Defaults to *Timetabled Latest Date Time* May change in real-time. This is the time that shall be used for presentation to the public/passengers.
EstimatedDateTime	DateTime	Yes	The most likely arrival date/time. If available, it can be used for presentation to the public/passengers in preference of *Target Date Time*. Only available in response to real time methods, otherwise NULL.
ObservedDateTime	DateTime	Yes	The actual date/time of arrival. A non null value indicates that the vehicle has arrived at the stop. Only available in response to real time methods, otherwise NULL.
State	Integer	No	States for service journeys, arrivals and departures controls how information

Field name	Data type	Null?	Description
			shall be presented. This is described in detail chapter “Interpreting States” in [UM-PT/I/ROI/3]. When real time data is not available - state is by default set to 2 – “Expected”.
Type	Integer	No	See section 18.2.1 above.
PresentationType	Integer	No	See section 18.1.1 abovefor explanation of this field.

19 Deviations

19.1 Deviation Message Version

This table contains a specific Deviation Message Version.

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Unique id. Can be used when caching and refreshing info about this object.
PublicNote	String(1-255)	No	Information in default language.
InternalNote	String(1-255)	Yes	Information NOT for publication.
PriorityImportanceLevel	Integer	No	The three priority dimensions each have a value between 1 and 9, where a higher number corresponds to a higher priority. Please consult [UM-PT/I/ROI/3] for how to interpret these values.
PriorityInfluenceLevel	Integer	No	
PriorityUrgencyLevel	Integer	No	
TargetAudienceCustomer	Bit	Yes	
TargetAudiencePassenger	Bit	Yes	
TargetAudienceStaff	Bit	Yes	

19.2 Deviation Message Variant

This table contains variants of a Deviation Message Version. There is a variant for each type of content and language that the user have provided to PubTrans®

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Unique id. Can be used when caching and refreshing info about this object.
IsPartOfDeviation-MessageVersionId	Numeric(16)	No	Refers to field *Id* in table *Deviation Message Version*.
Content	String(1-3600)	No	Actual text or hyperlink to content.
ContentTypeLongCode	String(1-16)	No	The type of content in the field *Content*.
UsageTypeLongCode	String(1-16)	No	
LanguageCode	String(1-8)	Yes	Language code of content according ISO 639.1.

19.3 Service Journey Deviation

This table holds the relation between a Dated Service Journey and a Deviation Message Variant.

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Unique id. Can be used when caching and refreshing info about this object.
IsOnDatedVehicleJourneyId	Numeric(16)	No	Refers to field *IsDatedVehicleJourneyId* in table *Dated Service Journey*.
HasDeviationMessageVersionId	Numeric(16)	No	Refers to field *Id* in table *Deviation Message Version*.
ConsequenceLongCode	String(1-16)	No	Please consult [UM-PT/I/ROI/3] for how to interpret this value.

19.4 Arrival Deviation

This table holds the relation between a Dated Arrival and a Deviation Message Variant.

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Unique id. Can be used when caching and refreshing info about this object.
IsOnArrivalId	Numeric(16)	No	Refers to field *Id* in table *Dated Arrival*.
HasDeviationMessageVersionId	Numeric(16)	No	Refers to field *Id* in table *Deviation Message Version*.
ConsequenceLongCode	String(1-16)	No	Please consult [UM-PT/I/ROI/3] for how to interpret this value.
AffectsPreviousDepartures	Boolean	No	

19.5 Departure Deviation

This table holds the relation between a Dated Departure and a Deviation Message Variant.

Field name	Data type	Null?	Description
Id	Numeric(16)	No	Unique id. Can be used when caching and refreshing info about this object.
IsOnDepartureId	Numeric(16)	No	Refers to field *Id* in table *Dated Departure*.
HasDeviationMessageVersionId	Numeric(16)	No	Refers to field *Id* in table *Deviation Message Version*.
ConsequenceLongCode	String(1-16)	No	Please consult [UM-PT/I/ROI/3] for how to interpret this value.
AffectsLaterArrivalsYesNo	Boolean	No	

19.6 Publication Scope

Field name	Data type	Null?	Description
IsForDeviation-MessageVersionId	Numeric(16)	No	The field refers to field *Id* in the *Deviation Message* table.
ConcernsLineGid	Numeric(16)	No	

19.7 Target Audience

Field name	Data type	Null?	Description
IsForDeviation-MessageVersionId	Numeric(16)	No	The field refers to field *Id* in the *Deviation Message* table.
TypeCode	String(1-8)	No	

20 Definitions

20.1 Global Identifiers

This text is copied and adapted from the PubTrans Interface document [IS-PT/I/DOI/4].

PubTrans uses a numbering convention called *Global Identifiers* or *GID* for short. A GID is a numeric identifier that has the same value for the same object across all installations of PubTrans databases. GID is used extensively in the interfaces as keys for referring to PubTrans objects. It is assumed that a client application can map its internal data model to the proper GID when exchanging data with PubTrans.

20.1.1 GID Construction

A GID is a 16-digit number. Currently, there are two main types of GID:

Key-based: A 16-digit number constructed by concatenating numeric attributes associated with the object. Key based GIDs starts with the number '9'.

Abstract: A 16-digit number that does not contain any special meaning. An abstract GID does not start with the number '9'.

20.1.1.1 Key based GID

Key based GIDs consist of a four-digit class identifier, a three-digit Transport Authority number and a 9-digit value divided into one or several fields.

Each field contains a numeric attribute data from the object. The fields have different representation for each class; either constructed by concatenating numeric attributes of the object in question or as an abstract number that does not contain any special meaning. This is explained for each type of GID below.

A key based GID is defined in the context of a Transport Authority, i.e. it is unique within a Transport Authority. If an object is referenced by more than one Transport Authority, it will have a GID for each Transport Authority. Thus, it is possible that several different GIDs refer the same object in PubTrans.

A GID is uniquely referring zero or one object version at a specific point in time, but the same GID may refer different object versions at different times.

Each field has two reserved values; zero that means 'unknown' and an all-9-value which is reserved for testing purposes. A GID with a zero-field does not refer any object. The zero-value field is used in some cases to let PubTrans assign the field automatically. A GID with an all-9-value shall only be used for test purposes.

20.1.1.2 Abstract GID

An abstract GID is a non-descriptive 16-digit number. An abstract GID is always less than 9000000000000000. Abstract GIDs are assigned automatically by PubTrans. Because abstract GIDs are not based on key data provided from external systems, there is no way to see if two abstract GIDs from different PubTrans database instances represents the same object or not. In fact, abstract GIDs in different PubTrans databases never overlap, because an abstract GID is globally unique within a PubTrans group.

20.1.2 Object Types with GID

The following object types (classes) have global identifiers in PubTrans.

Class Name	Class Id
Block	9041
Bridging Device	9095
Contractor	9013
Dead Run	9016
Deviation Case	9076
Deviation Message (obsolete)	*9071*
Direction (Direction of Line)	9014
Direction of Line (obsolete)	*9012*
Duty	9061
Employee	9051
Journey Pattern Point	9025

Class Name	Class Id
Line	9011
Place	9091
Service Journey	9015
Station Entrance Point	9023
Stop Area	9021
Stop Point	9022
Transport Authority	9010
Vehicle	9031
Virtual Vehicle	9038
Zone	9081

20.1.3 GID Format

Bold numbers indicate fixed values, and *italic* numbers indicate variable fields.

20.1.3.1 Transport Authority

9	**0**	**1**	**0**	*1*	*2*	*3*	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Class Id				Transport Authority Number			Not Used								

A *transport authority numbe*r must be between 1 and 998.

Transport authority numbers should be coordinated between co-operating transport authorities, preferably on a national level.

20.1.3.2 Line

9	**0**	**1**	**1**	*1*	*2*	*3*	*1*	*2*	*3*	*4*	**0**	**0**	**0**	**0**	**0**
Class Id				Transport Authority Number			Line Number				Not used				

A *line number* must be between 1 and 9998. The line number must be unique within a transport authority.

A line number is not necessarily the number presented to the public. In PubTrans, the public identification of the line is called *line designation*, which is a separate attribute from the line number.

If alphanumeric characters are used in the public identification of the line, it is necessary to define a numeric counterpart for use in GID, which can be considered as an internal alternative identification of the line.

If a line is operated by more than one transport authority (split responsibility), the line will have one GID per transport authority, with different transport authority number for, but preferably with same line number.

20.1.3.3 Direction of Line (old format)

This GID is obsolete in PubTrans 5. It will be supported in DOI 3 for backwards compatibility. It is replaced by the GID described in section 20.1.3.5 below.

9	**0**	**1**	**2**	*1*	*2*	*3*	*1*	*2*	*3*	*4*	*1*	**0**	**0**	**0**	**0**
Class Id				Transport Authority Number			Line Number				Direction	Not used			

A *direction-of-line number* must be 0 or 1.

Direction of line is used to separate journey patterns and routes into two main groups running in opposite directions.

20.1.3.4 Contractor

9	**0**	**1**	**3**	*1*	*2*	*3*	*1*	*2*	*3*	*4*	**0**	**0**	**0**	**0**	**0**
Class Id				Transport Authority Number			Contractor Number				Not Used				

A *contractor number* must be between 1 and 9998. A contractor number must be unique within a transport authority.

The contractor number denotes a vehicle operator organisation that is engaged to operate public transportation within a region governed by a transport authority. If the same vehicle operators are engaged by several transport authorities, they will have a GID within each transport authority.

It is assumed that it is the transport authority that coordinates and assigns the contractor number for each engaged vehicle operator.

20.1.3.5 Direction (Direction of Line)

This GID replaces the direction of line GID in section 20.1.3.3 above and will be used in PubTrans 5 and in DOI 4. The terms *direction* and *direction of line* are used interchangeably.

9	**0**	**1**	**4**	*1*	*2*	*3*	*1*	*2*	*3*	*4*	*1*	**0**	**0**	**0**	**0**
Class Id				Transport Authority Number			Line Number				Direction	Not used			

A *direction number* must be 1 (corresponding to 1 in the obsolete direction of line GID format) or 2 (corresponding to 0 in the obsolete direction of line GID format).

Direction (of line) is used to separate journey patterns and routes into two main groups running in opposite directions.

20.1.3.6 Service Journey

9	**0**	**1**	**5**	*1*	*2*	*3*	*1*	*2*	*3*	*4*	*1*	*2*	*3*	*4*	*5*
Class Id				Transport Authority Number			Line Number				Journey Number				

A *service journey number* must be between 1 and 99998. A service journey number must be unique within a line and for a specific operating day.

The service journey GID uniquely defines a service journey within PubTrans. If it is important to retain the service journey GID over time, it is strongly recommended to avoid renumbering of the service journeys in the source system each time data is provided to PubTrans.

20.1.3.7 Dead Run

9	**0**	**1**	**6**	*1*	*2*	*3*	*1*	*2*	*3*	*4*	*1*	*2*	*3*	*4*	*5*
Class Id				Transport Authority Number			Contractor Number				Dead Run Number				

A *dead run number* must be between 1 and 99998. In the context of one transport authority, a dead run number must be unique within the contractor for a specific operating day.

Dead run numbers can be assigned sequentially over time to avoid duplicate numbers.

20.1.3.8 Stop Area

9	**0**	**2**	**1**	*1*	*2*	*3*	*1*	*2*	*3*	*4*	*5*	*6*	**0**	**0**	**0**
Class Id				Transport Authority Number			Stop Area Number						Not used		

A *stop area number* must be between 1 and 999998. A stop area number must be unique within a transport authority.

In PubTrans it is possible to define a stop area that can be utilised by several transport authorities. If several transport authorities share the operation at a stop area, they can assign a GID each.

20.1.3.9 Stop Point

9	**0**	**2**	**2**	*1*	*2*	*3*	*1*	*2*	*3*	*4*	*5*	*6*	*1*	*2*	*3*
Class Id				Transport Authority Number			Stop Area Number						Local Stop Point Number		

A *local stop point number* must be between 1 and 998. A local stop point number must be unique within a stop area.

This is the local number within the stop area assigned to each stop point, e.g. track or gate number. There is also a GID for globally numbering of stop points within a transport authority, see Journey Pattern Point in section 20.1.3.11 below.

20.1.3.10 Station Entrance Point

9	**0**	**2**	**3**	*1*	*2*	*3*	*1*	*2*	*3*	*4*	*5*	*6*	*1*	*2*	*3*
Class Id				Transport Authority Number			Stop Area Number						Local Station Entrance Number		

A *local station entrance point number* must be between 1 and 998. A local station entrance point number must be unique within a stop area.

20.1.3.11 Journey Pattern Point

9	**0**	**2**	**5**	*1*	*2*	*3*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
Class Id				Transport Authority Number			Journey Pattern Point Number								

A *journey pattern point number* must be between 1 and 999 999 998. A journey pattern point number must be unique within a transport authority.

A journey pattern point is any type of point that can occur in a journey pattern, i.e. a point that can be used as a reference point in a scheduling system. Journey pattern points are: stop points, via-points and parking points. The journey pattern point number usually corresponds to a stop number.

20.1.3.12 Vehicle

9	**0**	**3**	**1**	*1*	*2*	*3*	*1*	*2*	*3*	*4*	*1*	*2*	*3*	*4*	*5*
Class Id				Transport Authority Number			Contractor Number				Vehicle Number				

A *vehicle number* must be between 1 and 99 998. In the context of one transport authority, a vehicle number must be unique within a contractor.

If a vehicle is identified in context of more than one transport authority, a vehicle will have a GID for each transport authority.

If a vehicle becomes operated by another vehicle operator (usually as a change of ownership) and still is in the context of a transport authority using PubTrans, the vehicle will get a new GID, because the contractor number will change. Probably the vehicle number as well, because each vehicle operator assigns internal vehicle numbers. Thus, a vehicle GID is not an identifier for the physical vehicle. However, PubTrans can handle several vehicle-GIDs referencing to one physical vehicle if some other identifier for the physical vehicle is provided at import of vehicle data.

20.1.3.13 Virtual Vehicle

9	**0**	**3**	**8**	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*
Class Id				Virtual Vehicle Number											

A *virtual vehicle number* must be between 1 and 999 999 999 998. A virtual vehicle number corresponds to one vehicle within one PubTrans system at a certain point in time.

Virtual vehicle numbers are used as temporary vehicle-identifier tags when information about the actual vehicle numbers is not available. Virtual vehicle numbers should only be used when it is not possible to use actual vehicle numbers. If virtual vehicle numbers are used, then it is preferable that a virtual vehicle number is attached for as long as possible to a certain vehicle.

20.1.3.14 Block

9	**0**	**4**	**1**	*1*	*2*	*3*	*1*	*2*	*3*	*4*	*1*	*2*	*3*	*4*	*5*
Class Id				Transport Authority Number			Contractor Number				Block Number				

A *block number* must be between 1 and 99998. In the context of one transport authority, a block number must be unique within a contractor.

20.1.3.15 Employee

9	**0**	**5**	**1**	*1*	*2*	*3*	*1*	*2*	*3*	*4*	*1*	*2*	*3*	*4*	*5*
Class Id				Transport Authority Number			Contractor Number				Employee Number				

An *employee number* must be between 1 and 99998. In the context of one transport authority, an employee number must be unique within a contractor.

The employee GID is mostly used to identify drivers within different vehicle operators.

20.1.3.16 Duty

9	**0**	**6**	**1**	*1*	*2*	*3*	*1*	*2*	*3*	*4*	*1*	*2*	*3*	*4*	*5*
Class Id				Transport Authority Number			Contractor Number				Duty Number				

A *duty number* must be between 1 and 99998. In the context of one transport authority, a duty number must be unique within a contractor.

20.1.3.17 Deviation Message

This GID is obsolete in PubTrans 5.

9	**0**	**7**	**1**	*1*	*2*	*3*	*1*	*2*	*3*	*1*	*2*	*3*	*4*	*5*	*6*
Class Id				Transport Authority Number			Message System Number			Message Number					

A deviation message GID consists of two parts: a *message system number* that must be between 1 and 998, and a *deviation message number* that must be between 1 and 999998.

The message system number identifies the source system of the message and the message number is a unique identifier of the deviation message within that system. Both numbers are assigned sequentially by PubTrans

20.1.3.18 Deviation Case

9	**0**	**7**	**6**	*1*	*2*	*3*	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Class Id				Transport Authority Number			Not used	Deviation Case Number							

A deviation case number must be between 1 and 99999998.

The deviation case number is assigned sequentially by PubTrans

20.1.3.19 Zone

9	0	8	1	1	2	3	1	2	1	2	3	4	5	6	7
Class Id				Transport Authority Number			Type of Zone Number		Zone Number						

A zone GID consists of two parts: a *type-of zone-number* that must be between 1 and 98, and a *zone number* that must be between 1 and 9 999 998.

The following zone type numbers is reserved:

Zone type	Usage
10	Administrative zone of transport authority.
11	Local administrative zone (corresponding to Swedish ‘kommun’).
12	Regional administrative zone (corresponding to Swedish ‘län’).
13-19	Custom defined administrative zone types.
20	Tariff zones.
21-29	Custom defined additional tariff zone types. Can be used if different types of tariff zones are used in parallel.
30-39	Custom analysis zone types.
40-49	Custom technical systems zone type, e.g. radio coverage zones.
50	Parking area
51-89	Reserved for future use
90-98	Reserved for system supplier specific use.

20.1.3.20 Place

9	0	9	1	*1*	*2*	*3*	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Class Id				Transport Authority Number			Not used	Place Number							

A *place number* must be between 1 and 99 999 998 and must be unique within a transport authority.

20.1.3.21 Bridging Device

9	**0**	**9**	**5**	*1*	*2*	*3*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
Class Id				Transport Authority Number			Bridging Device Number								

A *bridging device number* must be between 1 and 999 999 998, both values inclusive. The values 0 and 999 999 999 are reserved for special purposes. A bridging device number must be unique within a transport authority.

21 References

Document	Description
[IS-PT/I/DOI/4]	PubTrans Data Output Interface 4 – Interface Specification
[UM-PT/I/DOI/3]	PubTrans Data Output Interface 3 – User manual
[UM-PT/I/ROI/3]	PubTrans Real-time Output Interface 3 – User manual