Delegated Regulation (EU) N° 885/2013 of 15 May 2013 with regard to the provision of information services for safe and secure parking places for trucks and commercial vehicles

European Access Point for Truck Parking Data

User Manual: Web Service description to push static DATEX II data related to safe and secure Truck Parking Areas to the European Access point

V1.1

March 2016

EUROPEAN COMMISSION

Directorate-General for Mobility and Transport
Directorate C —Innovative & sustainable mobility
Unit C3 — Intelligent Transport Systems

Contact: MOVE-ITS@ec.europa.eu

European Commission
B-1040 Brussels

Table of Contents

1 Introduction

1.1 Background

The ITS Directive (2010/40/EU) provides a legal framework in order to support the coordinated and interoperable deployment and use of ITS across Europe. The Directive empowers the Commission to adopt delegated acts. Six priority actions have been defined in the ITS Directive and one of them – Action "e" – concerns the provision of information services for safe and secure parking places for trucks and commercial vehicles.

Specifications have been adopted by the Commission (delegated Regulation (EU) N°.885/2013 on 15 May 2013. More Information on the ITS Directive and delegated Regulation 885/2013 (including links to the actual legal documents) can be found at http://ec.europa.eu/transport/themes/its/road/action_plan/.

1.2 European Access Point for Truck Parking Data

Article 5 of delegated regulation 885/2013 requires the public or private parking operators and service providers to share and exchange data related to the parking areas through a national or international access point. The adopted delegated act does not oblige the European Commission to develop an access point. However, the Commission has decided to develop a **European Access Point for Truck Parking Data**, since this need has been repeatedly expressed by Member States during the preparation phase with Member States experts and after the adoption of delegated regulation 885/2013.

The European Access Point for Truck Parking aims to collect truck parking data according to regulation 885/2013 in order to make the collected data available as DATEX II export. At this moment the Access Point will only **support static data** as defined in Article 4(1) and 4(2) of the delegated regulation.

DATEX II
http://www.datex2.eu/
http://www.datex2.eu/sites/www.datex2.eu/files/DATEXIIv2.0-DevGuide.pdf

DATEX II – Part 6
http://www.datex2.eu/content/datex-ii-parking-submitted-standardisation
http://www.datex2.eu/sites/www.datex2.eu/files/TC_278_prCEN_TS_16157-6_E.pdf

European Open Data Portal: http://open-data.europa.eu/en/data/dataset/etpa
TENTEC Portal: http://ec.europa.eu/transport/infrastructure/tentec/tentec-portal/map/maps.html?layer=38

The sole responsibility for data quality and the necessary updating/maintenance obligations remain with the Member States that are submitting the data.

1.3 Scope of this document

A web-based data collection tool of the European Access Point enables Member States to directly enter data according to delegated regulation 885/2013 through a web application. Access to the data entry/collection tool will be granted by the EC to registered Member State representatives only. The entered data will be stored in the database of the European Commission. The tool can further be used to maintain and update the data. The web application tool provides functionalities in order to enable Member States to edit and maintain/update their truck parking data prior to submitting them for publication in DATEX II format on the open data portal of the EC ("export").

This web tool also proposes to Member States to submit data directly in Datex II format compliant with the Truck Parking profile as submitted to standardisation either by:

- Uploading a Datex II file from the web application
- Calling a SOAP **web service** to push Datex II.

The purpose of this document is to provide all the information that is needed for integrating this web service in Responsible Authorities local systems. This document also provides examples of client requests using SOAP-UI (open-source web service testing application) as generic client as well as Java code.
C#.Net source code won't be provided in this document but a client can be easily derived from the example shared on the Datex II forum at: http://www.datex2.eu/content/datex-ii-interfaces-building-soap-and-rest-web-services-java-and-cnet

In case of questions please contact:
MOVE-ITS@ec.europa.eu

2 Service Overview

The web service provides the ability to push static data on Truck Parking Areas to the European Access point by implementing the Datex II truck parking profile.

Please note that using the web service is similar than using the interactive upload function from the truck parking web application. Only the latest valid data submitted by a responsible authority will be published on the EU Open Data Portal. This web service cannot be used for Incremental /partial pushes.

On demand, and to deal with specific cases of federal countries, a single user account (e.g. at federal government level) can be used to push data on behalf of other responsible authorities of the same country (e.g. at regional level).

The web service will then associate files to the target responsible authorities based on the following rules:

- ParkingTable identifiers must start with the country code of the user account supplying data. (e.g. DE for Germany)
- ParkingTable identifiers must be equal to the identifier of a known responsible authority (e.g. DE-BE-ITP / DE-HH-ITP / DE-HB-IT with BE, HH and HB being respectively acronyms for German regions *Berlin, Freie und Hansestadt Hamburg, Freie Hansestadt Bremen* …)

This service is derived from the one published on the Datex II web portal at
http://www.datex2.eu/content/datex-ii-exchange-psm-3

and discussed on the forum :

http://www.datex2.eu/content/datex-ii-interfaces-building-soap-and-rest-web-services-java-and-cnet

Difference with Datex II Exchange PSM (in relation to links provided above):

- Only the push service is implemented. Data is made not accessible from a pull web service but only from the EU Open Data Portal.
- Payload must validate the Datex II Truck Parking profile as submitted to standardisation.
- A user/password authentication is required in order to clearly identified the submitter

3 Before you start

Before you start working with our web service, you need the Web Service Description and an applicative user account.

3.1.1 WSDL

SOAP web services are described through Web Services Description Language files (WSDL).

WSDL can be retrieved directly from the targeted environment:

- Acceptance

  https://webgate.acceptance.ec.europa.eu/tentec/policy/etpa/supplierPushService?wsdl

- Production

  https://webgate.ec.europa.eu/tentec/policy/etpa/supplierPushService?wsdl

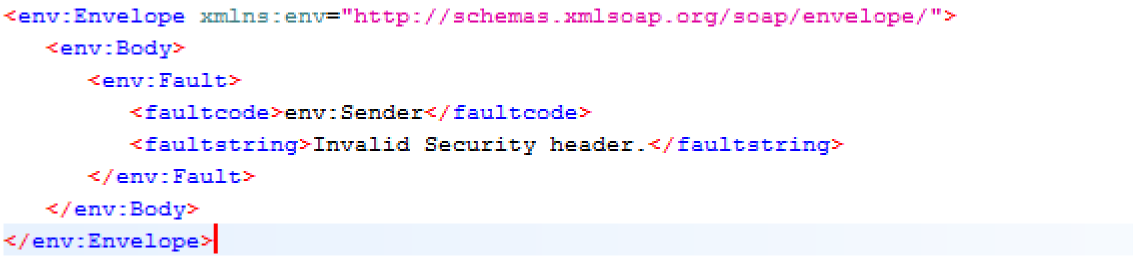
- Or directly with the "supplierPushService.zip" archive provided with this document (includes xjb binding customisation files):

3.1.2 Login/password

A basic authentication to access our web service is required. A login and password can be requested to your contact point at the European Commission: MOVE-ITS@ec.europa.eu

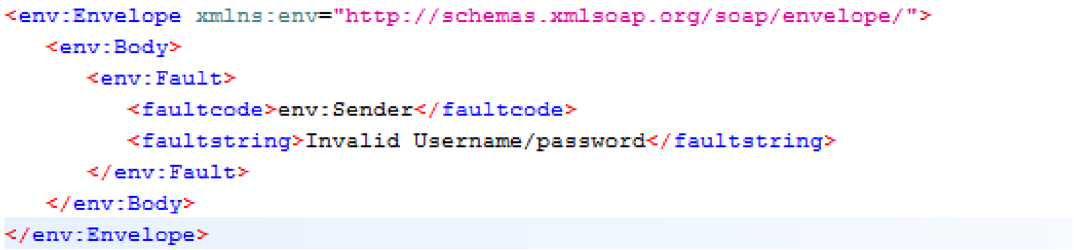
The web services are authenticated by "usernametoken" without HTTP basic authentication (pure wsse). Confidentiality is ensured by using HTTPS.

If you do not set the authentication information in the SOAP header, or if an incorrect login/password is used, you will get the following responses:



```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
   <env:Body>
      <env:Fault>
         <faultcode>env:Sender</faultcode>
         <faultstring>Invalid Security header.</faultstring>
      </env:Fault>
   </env:Body>
</env:Envelope>
```

Or



```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
   <env:Body>
      <env:Fault>
         <faultcode>env:Sender</faultcode>
         <faultstring>Invalid Username/password</faultstring>
      </env:Fault>
   </env:Body>
</env:Envelope>
```

The username password must be provided in compliance with the Web Services Security UsernameToken Profile 1.1 http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd

4 Calling the web service using SOAP-UI

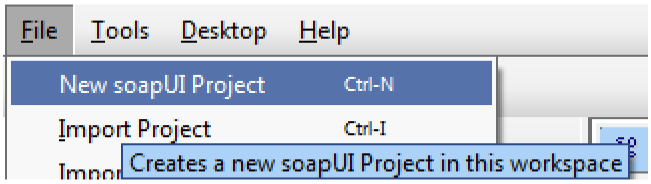
4.1 Download Soap-UI

The open source version of Soap-UI can be downloaded at

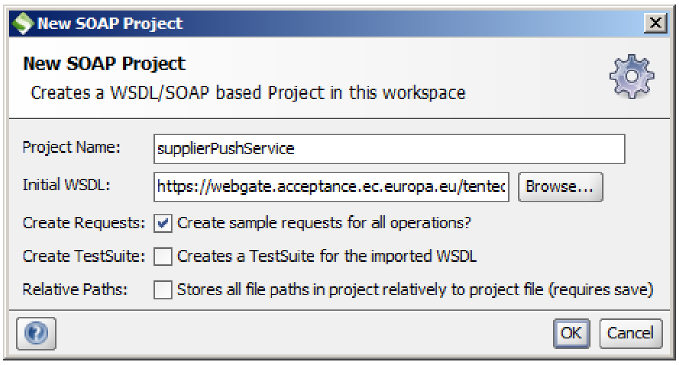
http://www.soapui.org/downloads/latest-release.html

4.2 Create a supplierPushService project

- Create a new SoapUI Project


File
Tools
Desktop
Help
New soapUI Project Ctrl-N
Import Project Ctrl-I
Impor
Creates a new soapUI Project in this workspace

- Set information about the WSDL.



New SOAP Project

Creates a WSDL/SOAP based Project in this workspace

Project Name: supplierPushService

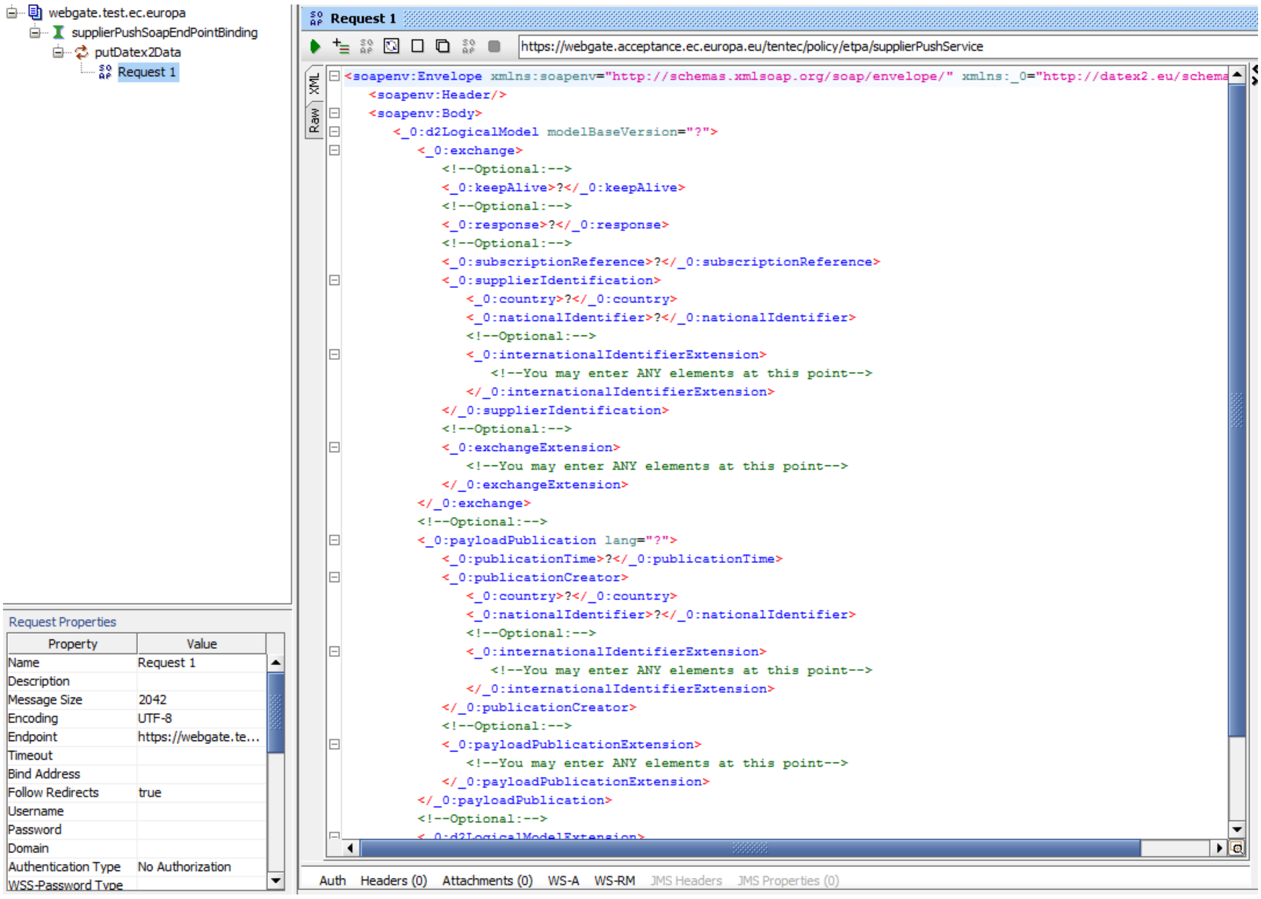
Initial WSDL: https://webgate.acceptance.ec.europa.eu/tentec

Create Requests: ☑ Create sample requests for all operations?

Create TestSuite: ☐ Creates a TestSuite for the imported WSDL

Relative Paths: ☐ Stores all file paths in project relatively to project file (requires save)

- A new project is created, that includes all the operations you can call on the specified web service



https://webgate.acceptance.ec.europa.eu/tentec/policy/etpa/supplierPushService

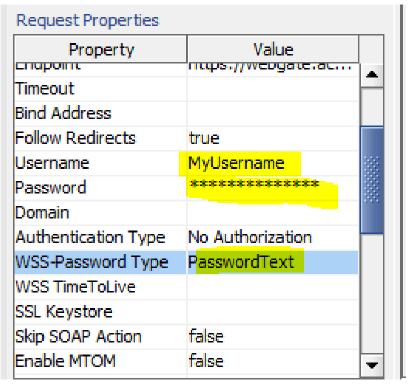
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:_0="http://datex2.eu/schema
   <soapenv:Header/>
   <soapenv:Body>
      <_0:d2LogicalModel modelBaseVersion="?">
         <_0:exchange>
            <!--Optional:-->
            <_0:keepAlive>?</_0:keepAlive>
            <!--Optional:-->
            <_0:response>?</_0:response>
            <!--Optional:-->
            <_0:subscriptionReference>?</_0:subscriptionReference>
            <_0:supplierIdentification>
               <_0:country>?</_0:country>
               <_0:nationalIdentifier>?</_0:nationalIdentifier>
               <!--Optional:-->
               <_0:internationalIdentifierExtension>
                  <!--You may enter ANY elements at this point-->
               </_0:internationalIdentifierExtension>
            </_0:supplierIdentification>
            <!--Optional:-->
            <_0:exchangeExtension>
               <!--You may enter ANY elements at this point-->
            </_0:exchangeExtension>
         </_0:exchange>
         <!--Optional:-->
         <_0:payloadPublication lang="?">
            <_0:publicationTime>?</_0:publicationTime>
            <_0:publicationCreator>
               <_0:country>?</_0:country>
               <_0:nationalIdentifier>?</_0:nationalIdentifier>
               <!--Optional:-->
               <_0:internationalIdentifierExtension>
                  <!--You may enter ANY elements at this point-->
               </_0:internationalIdentifierExtension>
            </_0:publicationCreator>
            <!--Optional:-->
            <_0:payloadPublicationExtension>
               <!--You may enter ANY elements at this point-->
            </_0:payloadPublicationExtension>
         </_0:payloadPublication>
         <!--Optional:-->
         <_0:d2LogicalModelExtension>
```

Request Properties

Property	Value
Name	Request 1
Description	
Message Size	2042
Encoding	UTF-8
Endpoint	https://webgate.te...
Timeout	
Bind Address	
Follow Redirects	true
Username	
Password	
Domain	
Authentication Type	No Authorization
WSS-Password Type	

4.3 Call the web service

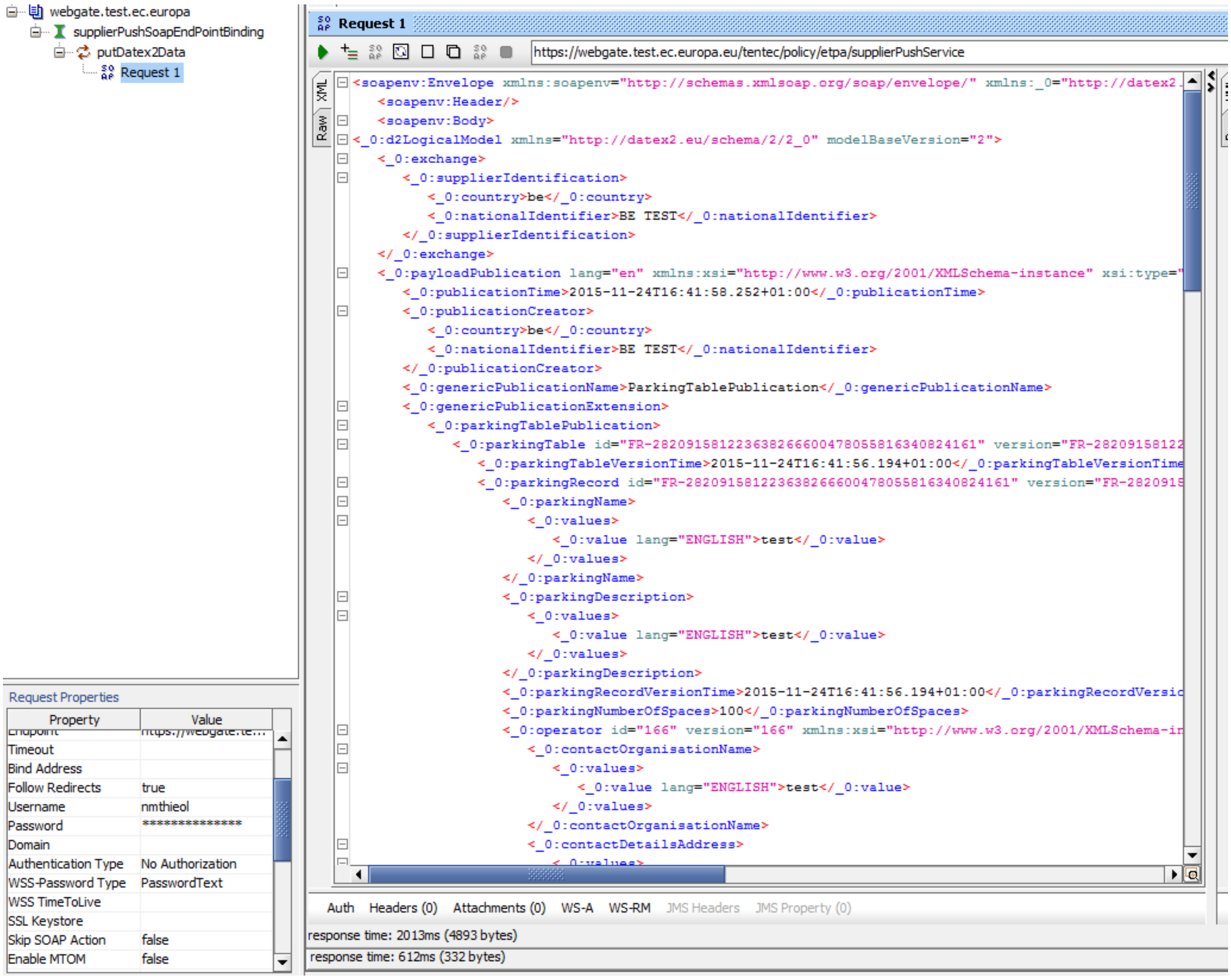
- Set the login, password and password type (passwordText)



Request Properties

Property	Value
Endpoint	https://webgate.ac...
Timeout	
Bind Address	
Follow Redirects	true
Username	MyUsername
Password	***************
Domain	
Authentication Type	No Authorization
WSS-Password Type	PasswordText
WSS TimeToLive	
SSL Keystore	
Skip SOAP Action	false
Enable MTOM	false

- Fill in the payload with a data structure compliant with the Truck Parking Profile.



Request Properties

Property	Value
Endpoint	https://webgate.te...
Timeout	
Bind Address	
Follow Redirects	true
Username	nmthieol
Password	**************
Domain	
Authentication Type	No Authorization
WSS-Password Type	PasswordText
WSS TimeToLive	
SSL Keystore	
Skip SOAP Action	false
Enable MTOM	false

Request 1

https://webgate.test.ec.europa.eu/tentec/policy/etpa/supplierPushService

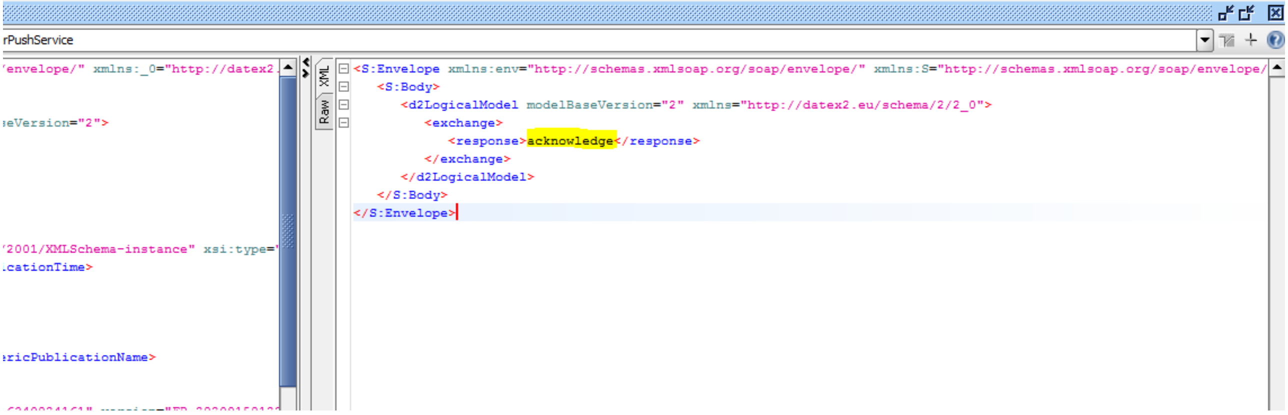
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:_0="http://datex2
   <soapenv:Header/>
   <soapenv:Body>
<_0:d2LogicalModel xmlns="http://datex2.eu/schema/2/2_0" modelBaseVersion="2">
   <_0:exchange>
      <_0:supplierIdentification>
         <_0:country>be</_0:country>
         <_0:nationalIdentifier>BE TEST</_0:nationalIdentifier>
      </_0:supplierIdentification>
   </_0:exchange>
   <_0:payloadPublication lang="en" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="
      <_0:publicationTime>2015-11-24T16:41:58.252+01:00</_0:publicationTime>
      <_0:publicationCreator>
         <_0:country>be</_0:country>
         <_0:nationalIdentifier>BE TEST</_0:nationalIdentifier>
      </_0:publicationCreator>
      <_0:genericPublicationName>ParkingTablePublication</_0:genericPublicationName>
      <_0:genericPublicationExtension>
         <_0:parkingTablePublication>
            <_0:parkingTable id="FR-282091581223638266600478055816340824161" version="FR-28209158122
               <_0:parkingTableVersionTime>2015-11-24T16:41:56.194+01:00</_0:parkingTableVersionTime
               <_0:parkingRecord id="FR-282091581223638266600478055816340824161" version="FR-2820915
                  <_0:parkingName>
                     <_0:values>
                        <_0:value lang="ENGLISH">test</_0:value>
                     </_0:values>
                  </_0:parkingName>
                  <_0:parkingDescription>
                     <_0:values>
                        <_0:value lang="ENGLISH">test</_0:value>
                     </_0:values>
                  </_0:parkingDescription>
                  <_0:parkingRecordVersionTime>2015-11-24T16:41:56.194+01:00</_0:parkingRecordVersic
                  <_0:parkingNumberOfSpaces>100</_0:parkingNumberOfSpaces>
                  <_0:operator id="166" version="166" xmlns:xsi="http://www.w3.org/2001/XMLSchema-ir
                     <_0:contactOrganisationName>
                        <_0:values>
                           <_0:value lang="ENGLISH">test</_0:value>
                        </_0:values>
                     </_0:contactOrganisationName>
                     <_0:contactDetailsAddress>
```

response time: 2013ms (4893 bytes)

response time: 612ms (332 bytes)

- Send the request to the server by clicking on ▶

- If the operation terminates successfully, the server responses by an acknowledge:



rPushService

```
<S:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:S="http://schemas.xmlsoap.org/soap/envelope/
   <S:Body>
      <d2LogicalModel modelBaseVersion="2" xmlns="http://datex2.eu/schema/2/2_0">
         <exchange>
            <response>acknowledge</response>
         </exchange>
      </d2LogicalModel>
   </S:Body>
</S:Envelope>
```

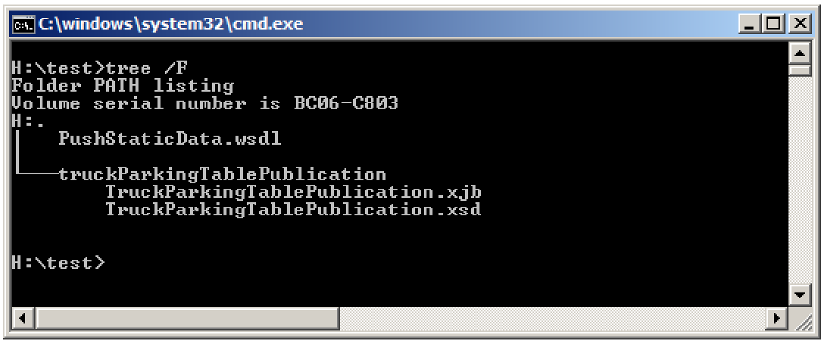
5 Calling the web service from a Java client

5.1 Generate the portable artifacts

Use the tool wsimport, included in the java development kit to generate the portable artefacts.

- Unzip the archive provided in § 3.1.1

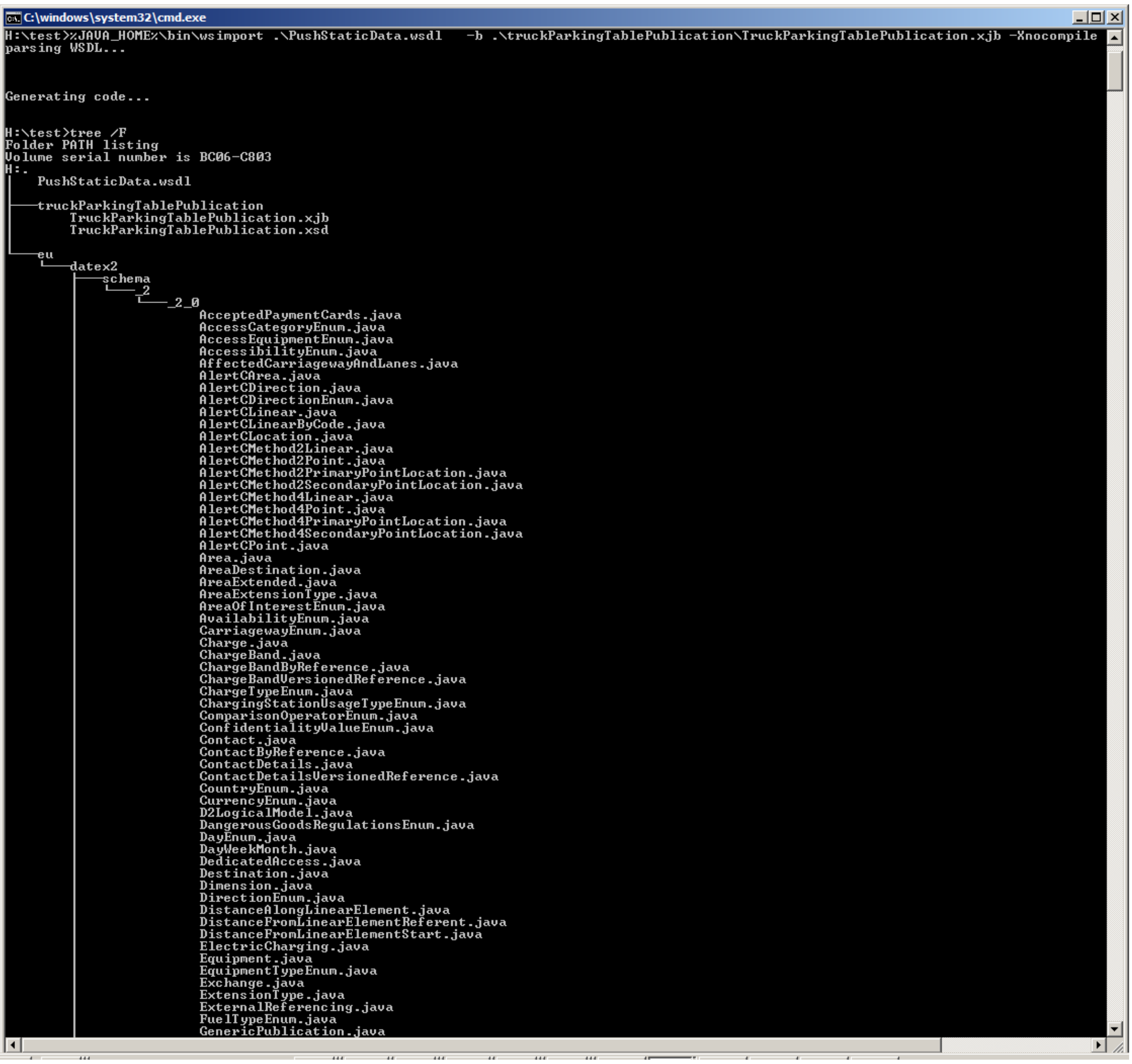
- Your working directory should be structured this way :



```
H:\test>tree /F
Folder PATH listing
Volume serial number is BC06-C803
H:.
│   PushStaticData.wsdl
│
└───truckParkingTablePublication
        TruckParkingTablePublication.xjb
        TruckParkingTablePublication.xsd


H:\test>
```

- Run the following command line

```
%JAVA_HOME%\bin\wsimport .\PushStaticData.wsdl -b
.\truckParkingTablePublication\TruckParkingTablePublication.xjb -Xnocompile -
XadditionalHeaders
```

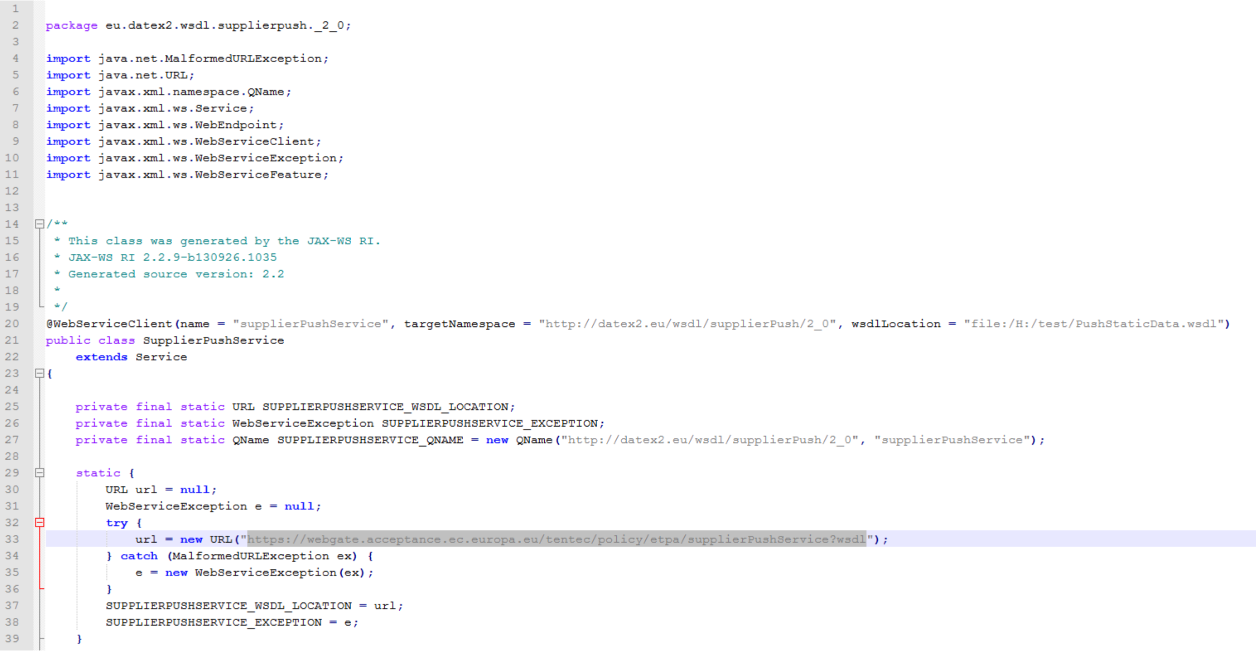


```
H:\test>%JAVA_HOME%\bin\wsimport .\PushStaticData.wsdl   -b .\truckParkingTablePublication\TruckParkingTablePublication.xjb -Xnocompile
parsing WSDL...



Generating code...


H:\test>tree /F
Folder PATH listing
Volume serial number is BC06-C803
H:.
│   PushStaticData.wsdl
│
├───truckParkingTablePublication
│       TruckParkingTablePublication.xjb
│       TruckParkingTablePublication.xsd
│
└───eu
    └───datex2
        ├───schema
        │   └───_2
        │       └───_2_0
        │               AcceptedPaymentCards.java
        │               AccessCategoryEnum.java
        │               AccessEquipmentEnum.java
        │               AccessibilityEnum.java
        │               AffectedCarriagewayAndLanes.java
        │               AlertCArea.java
        │               AlertCDirection.java
        │               AlertCDirectionEnum.java
        │               AlertCLinear.java
        │               AlertCLinearByCode.java
        │               AlertCLocation.java
        │               AlertCMethod2Linear.java
        │               AlertCMethod2Point.java
        │               AlertCMethod2PrimaryPointLocation.java
        │               AlertCMethod2SecondaryPointLocation.java
        │               AlertCMethod4Linear.java
        │               AlertCMethod4Point.java
        │               AlertCMethod4PrimaryPointLocation.java
        │               AlertCMethod4SecondaryPointLocation.java
        │               AlertCPoint.java
        │               Area.java
        │               AreaDestination.java
        │               AreaExtended.java
        │               AreaExtensionType.java
        │               AreaOfInterestEnum.java
        │               AvailabilityEnum.java
        │               CarriagewayEnum.java
        │               Charge.java
        │               ChargeBand.java
        │               ChargeBandByReference.java
        │               ChargeBandVersionedReference.java
        │               ChargeTypeEnum.java
        │               ChargingStationUsageTypeEnum.java
        │               ComparisonOperatorEnum.java
        │               ConfidentialityValueEnum.java
        │               Contact.java
        │               ContactByReference.java
        │               ContactDetails.java
        │               ContactDetailsVersionedReference.java
        │               CountryEnum.java
        │               CurrencyEnum.java
        │               D2LogicalModel.java
        │               DangerousGoodsRegulationsEnum.java
        │               DayEnum.java
        │               DayWeekMonth.java
        │               DedicatedAccess.java
        │               Destination.java
        │               Dimension.java
        │               DirectionEnum.java
        │               DistanceAlongLinearElement.java
        │               DistanceFromLinearElementReferent.java
        │               DistanceFromLinearElementStart.java
        │               ElectricCharging.java
        │               Equipment.java
        │               EquipmentTypeEnum.java
        │               Exchange.java
        │               ExtensionType.java
        │               ExternalReferencing.java
        │               FuelTypeEnum.java
        │               GenericPublication.java
```

5.2 Implement the service endpoint

- Access the default client stub generated by wsimport in package *eu.datex2.wsdl.supplierpush._2_0*

- Change the default wsdl location to either the acceptance or production environment (can also be define at construction time)



```

package eu.datex2.wsdl.supplierpush._2_0;

import java.net.MalformedURLException;
import java.net.URL;
import javax.xml.namespace.QName;
import javax.xml.ws.Service;
import javax.xml.ws.WebEndpoint;
import javax.xml.ws.WebServiceClient;
import javax.xml.ws.WebServiceException;
import javax.xml.ws.WebServiceFeature;


/**
 * This class was generated by the JAX-WS RI.
 * JAX-WS RI 2.2.9-b130926.1035
 * Generated source version: 2.2
 *
 */
@WebServiceClient(name = "supplierPushService", targetNamespace = "http://datex2.eu/wsdl/supplierPush/2_0", wsdlLocation = "file:/H:/test/PushStaticData.wsdl")
public class SupplierPushService
    extends Service
{

    private final static URL SUPPLIERPUSHSERVICE_WSDL_LOCATION;
    private final static WebServiceException SUPPLIERPUSHSERVICE_EXCEPTION;
    private final static QName SUPPLIERPUSHSERVICE_QNAME = new QName("http://datex2.eu/wsdl/supplierPush/2_0", "supplierPushService");

    static {
        URL url = null;
        WebServiceException e = null;
        try {
            url = new URL("https://webgate.acceptance.ec.europa.eu/tentec/policy/etpa/supplierPushService?wsdl");
        } catch (MalformedURLException ex) {
            e = new WebServiceException(ex);
        }
        SUPPLIERPUSHSERVICE_WSDL_LOCATION = url;
        SUPPLIERPUSHSERVICE_EXCEPTION = e;
    }
```

5.3 Implement a service client with JAX-WS RI

For this first basic draft implementation, a simple JRE (6 or above) without any additional library is required (Assumption: the SSL certificate for webgate.ec.europa.eu is trusted).

- Service Consumer

```
package eu.test;

import java.util.ArrayList;
import java.util.List;

import javax.xml.ws.BindingProvider;
import javax.xml.ws.Holder;
import javax.xml.ws.handler.Handler;

import eu.datex2.schema._2._2_0.CountryEnum;
import eu.datex2.schema._2._2_0.D2LogicalModel;
import eu.datex2.schema._2._2_0.Exchange;
import eu.datex2.schema._2._2_0.GenericPublication;
import eu.datex2.schema._2._2_0.InternationalIdentifier;
import eu.datex2.schema._2._2_0.ObjectFactory;
import eu.datex2.wsdl.supplierpush._2_0.SupplierPushInterface;
import eu.datex2.wsdl.supplierpush._2_0.SupplierPushService;

public class ServiceConsumer {

    /**
     * @see <a href="https://xwss.java.net/Securing_JAVASE6_WebServices.html">Securing A JAVA-SE 6 WebService <a>
     * @param args
     */
    public static void main(String[] args) {
              //Get web service instance
              SupplierPushService pushService = new SupplierPushService();
              SupplierPushInterface supplierService = pushService.getSupplierPushSoapEndPoint();

              //Authentication
              BindingProvider bp = (BindingProvider)supplierService;
              List<Handler> chain = new ArrayList<Handler>();
              chain.add(new SecurityHandler());
              bp.getBinding().setHandlerChain(chain);

              //Build d2Model object
              ObjectFactory factory = new ObjectFactory();
              D2LogicalModel d2Model =  factory.createD2LogicalModel();

              d2Model.setModelBaseVersion("2");
              //Datex Exchange information
              Exchange exchangeInfo = factory.createExchange();
              InternationalIdentifier supplierInfo = factory.createInternationalIdentifier();
              supplierInfo.setCountry(CountryEnum.BE);
              supplierInfo.setNationalIdentifier("test");
              exchangeInfo.setSupplierIdentification(supplierInfo);
              d2Model.setExchange(exchangeInfo);

              d2Model.setPayloadPublication(GenericParkingTablePublication());
              Holder<D2LogicalModel> holder= new Holder<D2LogicalModel>(d2Model);
              supplierService.putDatex2Data(holder);
    }


    /**
     * Generate the truck parking table publication Object Structure
     */
    private static GenericPublication GenericParkingTablePublication()
    {
              ObjectFactory factory = new ObjectFactory();
              GenericPublication publication = factory.createGenericPublication();
              //TODO Build the payload here
              //.....
              //.....
              //.....
              return publication;
    }
}
```

- Security Handler

```
package eu.test;

import java.util.HashSet;
import java.util.Set;

import javax.xml.namespace.QName;
import javax.xml.soap.SOAPElement;
import javax.xml.soap.SOAPEnvelope;
import javax.xml.soap.SOAPFactory;
import javax.xml.soap.SOAPHeader;
import javax.xml.ws.handler.Handler;
import javax.xml.ws.handler.MessageContext;
import javax.xml.ws.handler.soap.SOAPHandler;
import javax.xml.ws.handler.soap.SOAPMessageContext;

public class SecurityHandler implements SOAPHandler<SOAPMessageContext> {

    @Override
    public void close(MessageContext context) {
              // TODO Auto-generated method stub

    }

    @Override
    public boolean handleFault(SOAPMessageContext context) {
              return true;
    }

    @Override
    public boolean handleMessage(SOAPMessageContext context) {
              Boolean outboundProperty =

              (Boolean) context.get(MessageContext.MESSAGE_OUTBOUND_PROPERTY);

              if (outboundProperty.booleanValue()) {

              try {
                        SOAPEnvelope envelope = context.getMessage().getSOAPPart().getEnvelope();
                        SOAPFactory factory = SOAPFactory.newInstance();
                        String prefix = "wsse";
                        String uri = "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd";
                        SOAPElement securityElem = factory.createElement("Security", prefix, uri);
                        SOAPElement tokenElem =factory.createElement("UsernameToken", prefix, uri);
                        tokenElem.addAttribute(QName.valueOf("wsu:Id"), "UsernameToken-2");
                        tokenElem.addAttribute(QName.valueOf("xmlns:wsu"),"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd");
                        SOAPElement userElem = factory.createElement("Username", prefix, uri);
                        userElem.addTextNode("MyUserName");
                        SOAPElement pwdElem =factory.createElement("Password", prefix, uri);
                        pwdElem.addTextNode("MyPassword");
                        pwdElem.addAttribute(QName.valueOf("Type"),"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText");
                        tokenElem.addChildElement(userElem);
                        tokenElem.addChildElement(pwdElem);
                        securityElem.addChildElement(tokenElem);
                        System.out.println(envelope.getHeader());
                        SOAPHeader header = envelope.addHeader();
                        header.addChildElement(securityElem);
                        } catch (Exception e) {
                                  e.printStackTrace(); //Exception not managed in this draft example
                        }
              }
              return true;
    }

    @Override
    public Set<QName> getHeaders() {
              //System.out.println("Init header allowed to prevent mustunderstand not understood exception");
        QName wsse = new QName(
                "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd",
                "Security",
                "wsse");

        final HashSet<QName> headers = new HashSet<QName>();
        headers.add(wsse);

        return headers;
    }

}
```

Please note that usage of bindingProvider (to set the user/password) is not supported due to HTTP Basic authorization being added automatically to HTTP requests

References to wsse policies have been removed from the published WSDL to prevent undesired initiatives from JAX-WS RI.

5.4 Implement a service client with Apache CXF (alternative JAX-WS implementation)

- Prerequisites

Apache CXF libraries must be added in the project dependencies (Maven group id org.apache.cxf / artefact cxf-bundle)

Find more information on how to install cxf on you application server at http://cxf.apache.org/docs/application-server-specific-configuration-guide.html

- Service Consumer

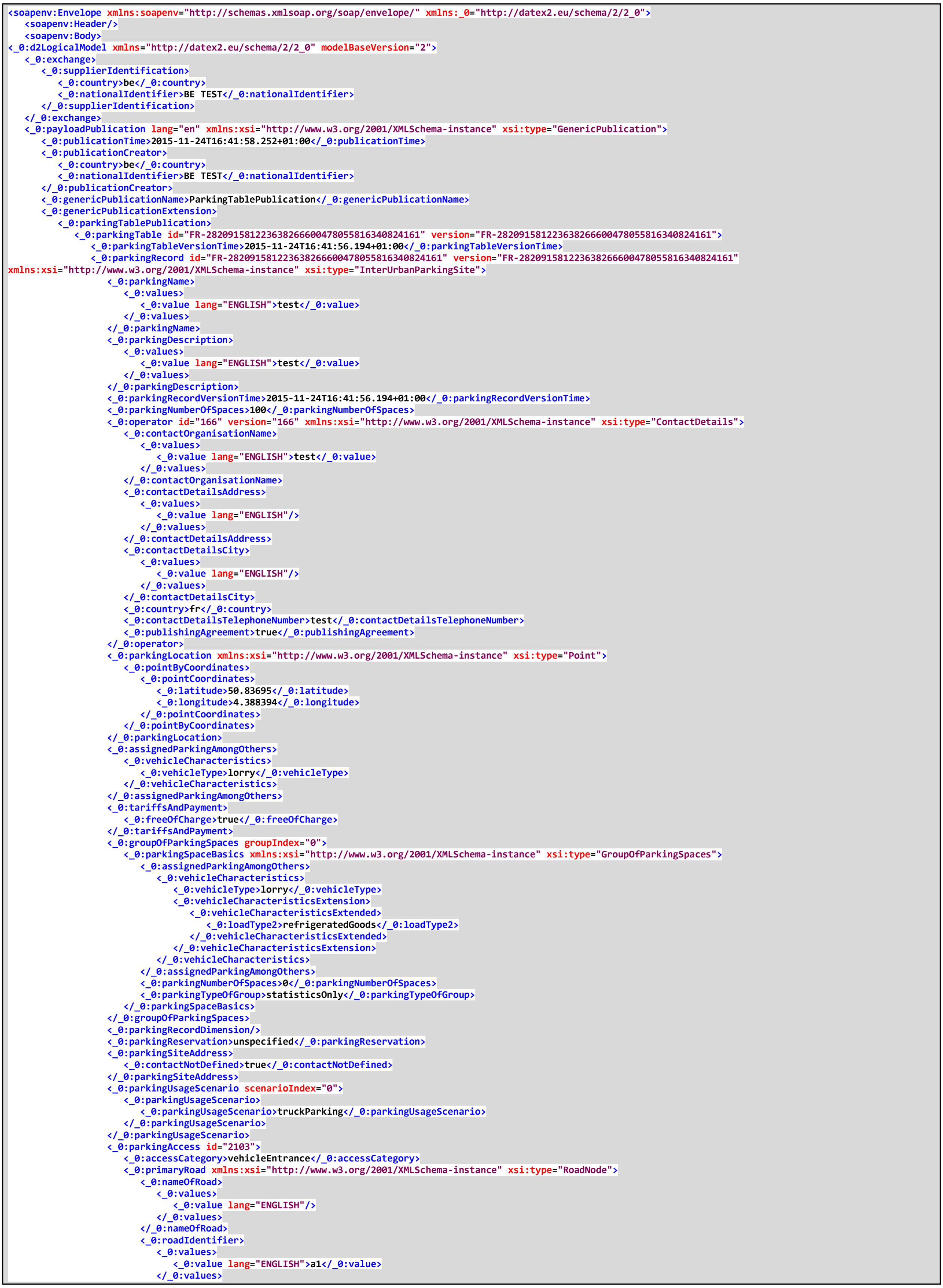
```
package eu.test.client;

import java.util.HashMap;
import java.util.Map;

import javax.xml.ws.Holder;

import org.apache.cxf.endpoint.Client;
import org.apache.cxf.endpoint.Endpoint;
import org.apache.cxf.ws.security.wss4j.WSS4JOutInterceptor;
import org.apache.ws.security.WSConstants;
import org.apache.ws.security.handler.WSHandlerConstants;

import eu.datex2.schema._2._2_0.CountryEnum;
import eu.datex2.schema._2._2_0.D2LogicalModel;
import eu.datex2.schema._2._2_0.Exchange;
import eu.datex2.schema._2._2_0.GenericPublication;
import eu.datex2.schema._2._2_0.InternationalIdentifier;
import eu.datex2.schema._2._2_0.ObjectFactory;
import eu.datex2.wsdl.supplierpush._2_0.SupplierPushInterface;
import eu.datex2.wsdl.supplierpush._2_0.SupplierPushService;

public class ServiceConsumer {

    @SuppressWarnings("unchecked")
    public static void main(String[] args) {
              SupplierPushService pushService = new SupplierPushService();
              SupplierPushInterface supplierService = pushService.getSupplierPushSoapEndPoint();
              // Build d2Model object
              ObjectFactory factory = new ObjectFactory();
              D2LogicalModel d2Model = factory.createD2LogicalModel();

              // Datex Exchange information
              Exchange exchangeInfo = factory.createExchange();
              InternationalIdentifier supplierInfo = factory.createInternationalIdentifier();
              supplierInfo.setCountry(CountryEnum.BE);
              supplierInfo.setNationalIdentifier("test");
              exchangeInfo.setSupplierIdentification(supplierInfo);
              d2Model.setExchange(exchangeInfo);
              d2Model.setModelBaseVersion("2");

              d2Model.setPayloadPublication(GenericParkingTablePublication());
              Holder<D2LogicalModel> holder = new Holder<D2LogicalModel>(d2Model);


              Client client = org.apache.cxf.frontend.ClientProxy.getClient(supplierService);
              Endpoint cxfEndpoint = client.getEndpoint();

              // can be replaced by a SpringFramework cxf.xml configuration file
              Map outProps = new HashMap<String, Object>();
              outProps.put(WSHandlerConstants.ACTION, WSHandlerConstants.USERNAME_TOKEN);
              outProps.put(WSHandlerConstants.USER, "nmthieol");
              outProps.put(WSHandlerConstants.PASSWORD_TYPE, WSConstants.PW_TEXT);
              outProps.put(WSHandlerConstants.PW_CALLBACK_CLASS, ClientPasswordCallback.class.getName());
              WSS4JOutInterceptor wssOut = new WSS4JOutInterceptor(outProps);
              cxfEndpoint.getOutInterceptors().add(wssOut);

              // call eu web service
              try {
                        supplierService.putDatex2Data(holder);
              } catch (Exception e) {
                        System.out.println(e);
              }

    }

    /**
     * Generate the truck parking table publication Object Structure
     */
    private static GenericPublication GenericParkingTablePublication() {
              ObjectFactory factory = new ObjectFactory();
              GenericPublication publication = factory.createGenericPublication();
              // TODO Build the payload here
              // .....
              // .....
              // .....
              return publication;
    }
}
```

- ClientPasswordCallback

```
package eu.test.client;

import java.io.IOException;
import javax.security.auth.callback.Callback;
import javax.security.auth.callback.CallbackHandler;
import javax.security.auth.callback.UnsupportedCallbackException;

import org.apache.ws.security.WSPasswordCallback;

public class ClientPasswordCallback implements CallbackHandler {

    public void handle(Callback[] callbacks) throws IOException,
            UnsupportedCallbackException {
        WSPasswordCallback pc = (org.apache.ws.security.WSPasswordCallback) callbacks[0];

        if ("myUsername".equals(pc.getIdentifier())) {
            pc.setPassword("myPassword");
        } // else {...} - can add more users, access DB, etc.
    }
}
```

6 Annexes

- Example of soap request used for tests in paragraph 4 :



```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:_0="http://datex2.eu/schema/2/2_0">
   <soapenv:Header/>
   <soapenv:Body>
<_0:d2LogicalModel xmlns="http://datex2.eu/schema/2/2_0" modelBaseVersion="2">
   <_0:exchange>
      <_0:supplierIdentification>
         <_0:country>be</_0:country>
         <_0:nationalIdentifier>BE TEST</_0:nationalIdentifier>
      </_0:supplierIdentification>
   </_0:exchange>
   <_0:payloadPublication lang="en" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="GenericPublication">
      <_0:publicationTime>2015-11-24T16:41:58.252+01:00</_0:publicationTime>
      <_0:publicationCreator>
         <_0:country>be</_0:country>
         <_0:nationalIdentifier>BE TEST</_0:nationalIdentifier>
      </_0:publicationCreator>
      <_0:genericPublicationName>ParkingTablePublication</_0:genericPublicationName>
      <_0:genericPublicationExtension>
         <_0:parkingTablePublication>
            <_0:parkingTable id="FR-282091581223638266600478055816340824161" version="FR-282091581223638266600478055816340824161">
               <_0:parkingTableVersionTime>2015-11-24T16:41:56.194+01:00</_0:parkingTableVersionTime>
               <_0:parkingRecord id="FR-282091581223638266600478055816340824161" version="FR-282091581223638266600478055816340824161"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="InterUrbanParkingSite">
                  <_0:parkingName>
                     <_0:values>
                        <_0:value lang="ENGLISH">test</_0:value>
                     </_0:values>
                  </_0:parkingName>
                  <_0:parkingDescription>
                     <_0:values>
                        <_0:value lang="ENGLISH">test</_0:value>
                     </_0:values>
                  </_0:parkingDescription>
                  <_0:parkingRecordVersionTime>2015-11-24T16:41:56.194+01:00</_0:parkingRecordVersionTime>
                  <_0:parkingNumberOfSpaces>100</_0:parkingNumberOfSpaces>
                  <_0:operator id="166" version="166" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="ContactDetails">
                     <_0:contactOrganisationName>
                        <_0:values>
                           <_0:value lang="ENGLISH">test</_0:value>
                        </_0:values>
                     </_0:contactOrganisationName>
                     <_0:contactDetailsAddress>
                        <_0:values>
                           <_0:value lang="ENGLISH"/>
                        </_0:values>
                     </_0:contactDetailsAddress>
                     <_0:contactDetailsCity>
                        <_0:values>
                           <_0:value lang="ENGLISH"/>
                        </_0:values>
                     </_0:contactDetailsCity>
                     <_0:country>fr</_0:country>
                     <_0:contactDetailsTelephoneNumber>test</_0:contactDetailsTelephoneNumber>
                     <_0:publishingAgreement>true</_0:publishingAgreement>
                  </_0:operator>
                  <_0:parkingLocation xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Point">
                     <_0:pointByCoordinates>
                        <_0:pointCoordinates>
                           <_0:latitude>50.83695</_0:latitude>
                           <_0:longitude>4.388394</_0:longitude>
                        </_0:pointCoordinates>
                     </_0:pointByCoordinates>
                  </_0:parkingLocation>
                  <_0:assignedParkingAmongOthers>
                     <_0:vehicleCharacteristics>
                        <_0:vehicleType>lorry</_0:vehicleType>
                     </_0:vehicleCharacteristics>
                  </_0:assignedParkingAmongOthers>
                  <_0:tariffsAndPayment>
                     <_0:freeOfCharge>true</_0:freeOfCharge>
                  </_0:tariffsAndPayment>
                  <_0:groupOfParkingSpaces groupIndex="0">
                     <_0:parkingSpaceBasics xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="GroupOfParkingSpaces">
                        <_0:assignedParkingAmongOthers>
                           <_0:vehicleCharacteristics>
                              <_0:vehicleType>lorry</_0:vehicleType>
                              <_0:vehicleCharacteristicsExtension>
                                 <_0:vehicleCharacteristicsExtended>
                                    <_0:loadType2>refrigeratedGoods</_0:loadType2>
                                 </_0:vehicleCharacteristicsExtended>
                              </_0:vehicleCharacteristicsExtension>
                           </_0:vehicleCharacteristics>
                        </_0:assignedParkingAmongOthers>
                        <_0:parkingNumberOfSpaces>0</_0:parkingNumberOfSpaces>
                        <_0:parkingTypeOfGroup>statisticsOnly</_0:parkingTypeOfGroup>
                     </_0:parkingSpaceBasics>
                  </_0:groupOfParkingSpaces>
                  <_0:parkingRecordDimension/>
                  <_0:parkingReservation>unspecified</_0:parkingReservation>
                  <_0:parkingSiteAddress>
                     <_0:contactNotDefined>true</_0:contactNotDefined>
                  </_0:parkingSiteAddress>
                  <_0:parkingUsageScenario scenarioIndex="0">
                     <_0:parkingUsageScenario>
                        <_0:parkingUsageScenario>truckParking</_0:parkingUsageScenario>
                     </_0:parkingUsageScenario>
                  </_0:parkingUsageScenario>
                  <_0:parkingAccess id="2103">
                     <_0:accessCategory>vehicleEntrance</_0:accessCategory>
                     <_0:primaryRoad xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="RoadNode">
                        <_0:nameOfRoad>
                           <_0:values>
                              <_0:value lang="ENGLISH"/>
                           </_0:values>
                        </_0:nameOfRoad>
                        <_0:roadIdentifier>
                           <_0:values>
                              <_0:value lang="ENGLISH">a1</_0:value>
                           </_0:values>
                        </_0:roadIdentifier>
                        <_0:roadDestination>
                           <_0:values>
                              <_0:value lang="ENGLISH">South</_0:value>
                           </_0:values>
                        </_0:roadDestination>
                        <_0:junctionName>
                           <_0:values>
                              <_0:value lang="ENGLISH"/>
                           </_0:values>
                        </_0:junctionName>
                     </_0:primaryRoad>
                     <_0:location xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Point">
                        <_0:pointByCoordinates>
                           <_0:pointCoordinates>
                              <_0:latitude>50.83695</_0:latitude>
                              <_0:longitude>4.388394</_0:longitude>
                           </_0:pointCoordinates>
                        </_0:pointByCoordinates>
                     </_0:location>
                  </_0:parkingAccess>
                  <_0:parkingStandardsAndSecurity>
                     <_0:labelSecurityLevel>none</_0:labelSecurityLevel>
                     <_0:labelServiceLevel>none</_0:labelServiceLevel>
                     <_0:labelSecurityLevelSelfAssessment>unknown</_0:labelSecurityLevelSelfAssessment>
                     <_0:labelServiceLevelSelfAssessment>unknown</_0:labelServiceLevelSelfAssessment>
                     <_0:parkingSecurity>none</_0:parkingSecurity>
                     <_0:parkingAdditionalSecurity>
                        <_0:values>
                           <_0:value lang="ENGLISH"/>
                        </_0:values>
                     </_0:parkingAdditionalSecurity>
                     <_0:parkingSupervision>none</_0:parkingSupervision>
                     <_0:parkingSecurityNationalClassification>
                        <_0:values>
                           <_0:value lang="ENGLISH"/>
                        </_0:values>
                     </_0:parkingSecurityNationalClassification>
                  </_0:parkingStandardsAndSecurity>
                  <_0:interUrbanParkingSiteLocation>motorway</_0:interUrbanParkingSiteLocation>
               </_0:parkingRecord>
            </_0:parkingTable>
         </_0:parkingTablePublication>
      </_0:genericPublicationExtension>
   </_0:payloadPublication>
</_0:d2LogicalModel>
   </soapenv:Body>
</soapenv:Envelope>
```