Implementation Guide for DataHub Webservice Interface

1. Table of content

2. Obtaining help

For any questions pertaining to the use of this interface or the ATS Engros system, you can contact Energinet.dk on either e-mail or phone at:

- datahub@energinet.dk
- (+45) 70 22 28 10

3. SOAP interface

The SOAP interface to the DataHub and consequently for the ATS Engros system is documented in the WSDL file, which can be found here: http://energinet.dk/da/el/datahub/sider/datahub.aspx

The service contract will not be made available for download through the endpoint URL, so a service proxy must be created from the file. The endpoint shown in the WSDL file must be overridden when connecting to ATS.

3.1 Service Endpoint

For ATS Engros, the URL for the service is:
https://datahub-ats.energinet.dk/DatahubATSEngros/DatahubATS.svc
Please note that this URL is configured to require Client Authenticated TLS handshake, as per RFC5246, in particular section 7.4.4 in particular.

The URL for the DataHub is not yet publicly available.

Energinet.dk suggests two independent means of testing connectivity to the service endpoint:

- By using a web browser of the host operating system
- By using a custom console application that can interact with the webservice by invoking the peekMessage operation.

For further information on this, please refer to the document "How to use certificates in ATS", which can be found in the FAQ section on the ATS Engros website.

3.2 General Error codes

3.2.1 Transport level (HTTP)

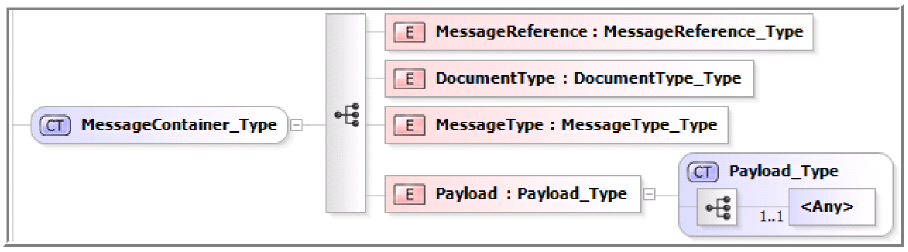
Error code	Type	Meaning
401	Security	Access Denied – in case of issues obtaining the users identity.
403	Security	Problem establishing SSL channel with client certificate
404	System	Requested resource not found (e.g. incorrect SOAP address)
413	System	Content length too large
500	System	In case of any unidentified errors.

3.2.2 Application level (SOAP)

Error code	Type	Meaning
MP-MED-0000	System	General Failure
MP-MED-0001	Syntax	Schema validation of service operation (SOAP request) failed
MP-MED-0002	Security	System configuration error
MP-MED-0003	Security	User not authorised (e.g. no rights for the operation, user blocked or inactive)
MP-MED-0004	Security	Unknown SOAP request
MP-MED-0005	System	Back-end timeout

3.3 Parameters to SOAP operations

A complex datatype, MessageContainer_Type, has been introduced for the sendMessageRequest, peekMessageResponse & getMessageResponse operations.


CT MessageContainer_Type
E MessageReference : MessageReference_Type
E DocumentType : DocumentType_Type
E MessageType : MessageType_Type
E Payload : Payload_Type
CT Payload_Type
1..1
<Any>

Element	Type	Notes
MessageReference	xs:string[0..35]	The MessageReference is used to identify the data transfer of a Business Message from the sending system. The MessageReference must be unique over time.
DocumentType	xs:string[0..200]	The DocumentType refers to the type of the business message in the message part of the data exchange. See **Fejl! Henvisningskilde ikke fundet.** for a full list of available values.
MessageType	xs:string={XML}	The MessageType indicates the enclosed message format, this can be "XML"
Payload	xs:any processContents=skip	Contains the actual Business Message in ebIX message format.

3.3.1 Handling message payload

The entire message must always be encoded using UTF-8.

- For XML this follows from the encoding attribute in the XML declaration
```
<?xml version="1.0" encoding="utf-8"?>
```

3.4 Namespaces of XML documents
The namespace of the payload can either be defined in the payload, or in the MessageContainer.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:SendMessageRequest xmlns:urn="urn:www:datahub:dk:b2b:v01">
      <urn:MessageContainer>
        <urn:MessageReference>MsgRef001</urn:MessageReference>
        <urn:DocumentType>RequestMPCharacteristics</urn:DocumentType>
        <urn:MessageType>XML</urn:MessageType>
        <urn:Payload>
          <DK_RequestMPCharacteristics
              xmlns="un:unece:260:data:EEM-DK_RequestMPCharacteristics:v01">
            <HeaderEnergyDocument>
              <Identification>MES032</Identification>
              <DocumentType listAgencyIdentifier="260">E10</DocumentType>
              <Creation>2002-11-07T12:00:00Z</Creation>
              <!-- ...snip... -->
            </HeaderEnergyDocument>
```

Example showing namespace of payload being defined as the default namespace inside the payload.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:SendMessageRequest xmlns:urn="urn:www:datahub:dk:b2b:v01"
      xmlns:mm="un:unece:260:data:EEM-DK_RequestMPCharacteristics:v01">
      <urn:MessageContainer>
        <urn:MessageReference>MsgRef001</urn:MessageReference>
        <urn:DocumentType>RequestMPCharacteristics</urn:DocumentType>
        <urn:MessageType>XML</urn:MessageType>
        <urn:Payload>
          <mm:DK_RequestMPCharacteristics>
            <mm:HeaderEnergyDocument>
              <mm:Identification>MES032</mm:Identification>
              <!-- ...snip... -->
            </mm:HeaderEnergyDocument>
```

Example showing namespace of payload being defined on the root element.

3.5 Example showing SOAP Fault
All errors returned to the client by DataHub will be on the form, shown below and are completely neutral to the syntax of the payload used by the actor.
The part of the faultstring to the right of the colon is an identifier, which can by used by 2nd level support to retrieve further details upon request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>soapenv:Client</faultcode>
      <faultstring>B2B-009:2127360337054</faultstring>
      <faultactor />
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

4. Description of operations
All operations are invoked by the client and are considered successful unless a SOAP Fault is returned.

4.1 sendMessage
The sendMessage operation is invoked in order to transmit a business document (the payload) to DataHub for processing. DataHub performs basic security and syntax checking synchronously and

returns the messageId from the payload as a confirmation that it has taken ownership of the document and will proceed to process it. If a semantical or business related error arises during processing, DataHub can send an RSM-009 (Acknowledgement or APERAK) to the actor with the source of the error, otherwise the actor can treat the message as being successfully processed.

4.1.1 Error codes

The following error codes can be returned as part of the synchronous validation by DataHub

Error Code	Type	Meaning
B2B-001	Security	The given DocumentType is not recognised
B2B-002	Security	The user of the SendMessage operation is not allowed to send this type of message (DocumentType) for its role
B2B-003	Syntax	The provided Ids are not unique and have been used before
B2B-004	Syntax	Content size of Payload too large for the given MessageType, se Forskrift F, bilagsrapport 4, section 2.9)
B2B-005	Syntax	Syntax validation failed for Business Message in Payload
B2B-006	Syntax	MessageType does not match the Business Message in Payload
B2B-007	System	Internal transformation failed
B2B-008	Security	Sender Identification in the Business Message is not authorised or user of the SendMessage operation has no relation with the organisation (i.e. Sender Identification)
B2B-009	System	The provided Ids are not unique in the Business Message (e.g. same TransactionId or TimeseriesId used in the same message), or duplicate Ids in requests when calling the SendMessage operation in parallel.
B2B-010	Syntax	Sender Role and/or Recipient Role not provided (see [RSM] dependency matrices)
B2B-011	Security	Invalid recipient
B2B-900	System	Internal server error

4.2 peekMessage

peekMessage is a nonmutating operation and can safely be called periodically in a loop by the client. It is advised to implement a simple scheduler, which calls peekMessage at regular intervals when no message is waiting and immediately after a successful dequeueOperation in order to empty the queue for outgoing messages.

4.2.1 Error codes

Error Code	Type	Meaning
B2B-900	System	Internal server error

4.3 dequeueMessage

4.3.1 Error codes

Error Code	Type	Meaning
B2B-201	System	Cannot dequeue the current message in the MessageQueue (i.e. the MessageId does not match the MessageId that has been peeked before)
B2B-900	System	Internal server error