DIGITALISERINGS
KATALOGET

SF1601 PROGRAMMER'S GUIDE

Examples and practicalities

KOMBIT
Kommunernes it-fællesskab

KOMBIT
Kommunernes it-fællesskab

INDHOLD

1. INTRODUCTION

This guide is targeted developers implementing SF1601 Send Post. It features examples of the most common use cases and practical information. Make sure to become acquainted with the Get-Started guides and MeMo ressources, that are listed in the References section.

This guide focuses on development tasks. The tasks each Municipality (authority) has to complete are described in "*SF1601 - Vejledning til tilslutning og test til afsendelse af post.pdf*" that you find in the documentation package for SF1601.

1.1. Prerequisites

- Your organisation has
  a. A pilot-municipality that you are testing with
  b. Access to Serviceplatformen administration
  c. An IT-system registered in Serviceplatformen as "Anvendersystem" with client certificate
  d. Access to Message Broker AMQP (Beskedfordeler) UI
  e. Created an AMQP queue (Dueslag) subscribing to NgDP events
  f. Requested NgDP Service Agreement with the municipality
- Your pilot-municipality has
  a. Registered Serviceplatformen as a sender-IT-system in NgDP
  b. Registered their NgDP API-key with Serviceplatformen
  c. Has approved the Service Agreement
- You, the developer, have
  a. Read and understood the Get-Started guides referenced below
  b. A copy of the private version of the certificate registered with your IT-system and the password to its private key

1.2. Compatible frameworks

The usage pattern is as follows:

(1) Request a SAML-token from the Security Token Service (REST)
(2) Exchange the SAML-token to an Access Token (REST)
(3) Invoke the service with the Access Token (REST)

All three step are simple basic HTTP-requests and thus compatible with all frameworks. There are code examples available on how to fetch a SAML-token using Java CXF and .NET 4.7.

1.3. References

[REF1] Get-Started guides

https://digitaliseringskataloget.dk/kom-godt-i-gang-vejledninger

Reading and understanding the guides *Certifikater*, *Webservice*, *Webservice OIOIDWS (REST)* and *Beskedfordeler* (Message Broker) is essential. These guides are in Danish.

[REF2] MeMo

https://www.digitaliser.dk/resource/5248921/

This page contains documentation, examples, list of allowed filetypes, MeMo XML schema, messages Markup etc.

[REF3] MeMo-lib for .NET and Java

https://bitbucket.org/nc-dp/

”*The main purpose of this library is to handle the XML serialization and deserialization of MeMo messages. It also provides support for handling MeMo messages contained in zip archives.*”.

[REF4] Technical Integration NgDP

https://www.digitaliser.dk/resource/5765802

It is recommended that you browse through this document, so you know what it contains. In case you need it at some point.

[REF5] SF1514 - Sikkerhed - Hent Token fra Security Token Service

SF1514 - Sikkerhed - Hent Token fra Security Token Service | Digitaliseringskatalog (digitaliseringskataloget.dk)

The documentation contains ”Snitfladebeskrivelse-STS-REST Service.pdf” describing how to fetch a SAML-token using REST.

1.4. Client implementation

When you have established a client that can fetch a SAML-token, exchange it to an Access Token, and then invoke a REST service with GET or POST, then you are good to go. This is described in the Get-Started guide *Webservice OIOIDWS (REST).*

Note that both XML and JSON is used in the two SF1601 web services. Regarding request-data, the most common errors are due to:

- Wrong sequence of XML-elements
- Missing or wrong namespaces
- Missing mandatory elements

So be aware to check your request against the MeMo or Fjernprint (physical print) XML schema before sending in the beginning. Also check out the MeMo validator in [REF2] (*Brug af Skema og Schematron i MeMo.pdf* and *MeMo_Schematron.zip*). The XML schema for Fjernprint you find in the documentation package.

Commonly data-classes are used to map and access request and response data. You can generate the request classes from the MeMo/Fjernprint XML schemas, which is illustrated in the appendix, and you have to create the response-classes manually. You can also generate the classes from the OpenAPI specs using e.g. Swagger, but this has not been fully tested yet. To build MeMo-requests you can also use [REF3] MeMo-Lib.

Be aware, that MeMo date/time elements such as <createdDateTime> must be in UTC format "yyyy-mm-ddThh:nn:ssZ". If you specify the time-part in other than time-Zulu the validation will fail.

Tip when testing: When you have fetched a SAML-token and exchanged it to an Access Token, you can use any HTTP user-agent such as Postman, Fiddler or SOAPUI to call the services. The services require 2-way-TLS a.k.a. client-certificate-authentication, so remember to add the private version of your client-certificate.

The SAML-tokens are valid 8 hours and the Access Tokens are valid 1 hour, and you must reuse them until they expire. Remember to add logic that checks, if a token is about to expire, and renew them when needed. Also remember, that tokens are authority-specific, so your token-handling logic must take this into account.

Configuration parameters for environment *ExtTest* are listed in Appendix - configuration parameters.

Service Entity IDs to use when reuesting a SAML-token:

- http://entityid.kombit.dk/service/kombipostafsend/1
- http://entityid.kombit.dk/service/postforespoerg/1

1.5. Employee-signature and NemLog-In roles

Access to Serviceplatformen Administration and Message Broker UI is enforced by NemLog-In and can only be granted to persons who have an employee-signature (Medarbejdersignatur) associated with your organisation. The NemLog-In administrator in your organisation can assign the needed roles. They are (test and production):

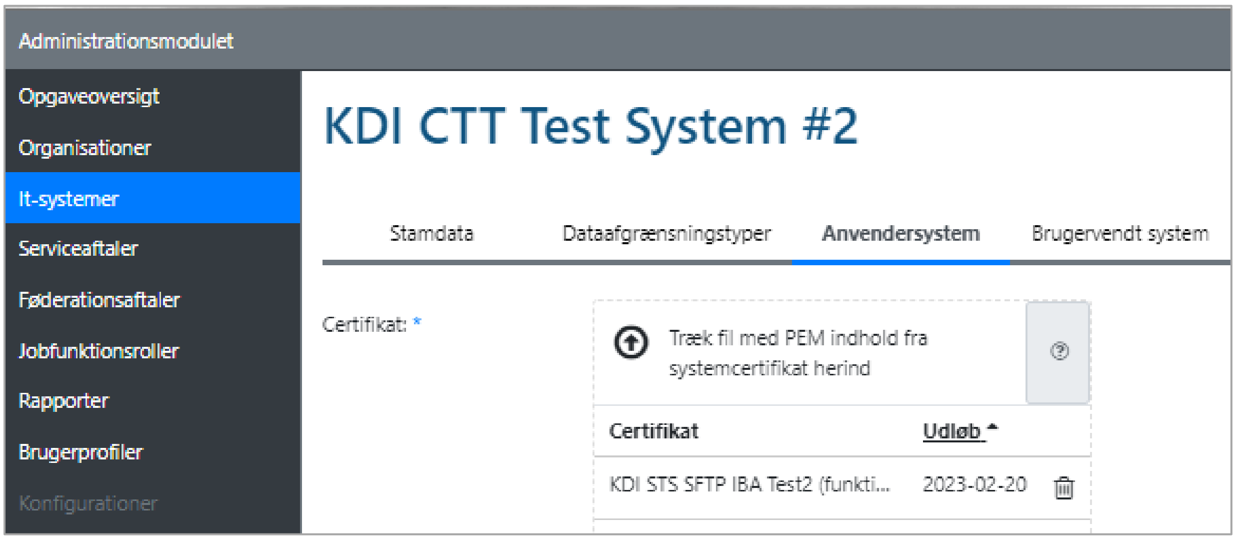
- KOMBIT STS Administrationsmodulet Leverandøradministrator

- KOMBIT STS Administrationsmodulet (test) Leverandøradministrator
- Beskedfordeler - Anvendersystemadministrator
- Beskedfordeler- Ekstern test - Anvendersystemadministrator

Usually one or two persons within an organisation are granted these roles, so you have to refer to them regarding the related tasks described in this document.

1.6. Serviceplatformen Configuration

These tasks are performed in Serviceplatformen Administration. Register your IT-system in Serviceplatformen Administration with a client (FOCES) certificate:



KDI CTT Test System #2

Certifikat: *

Træk fil med PEM indhold fra systemcertifikat herind

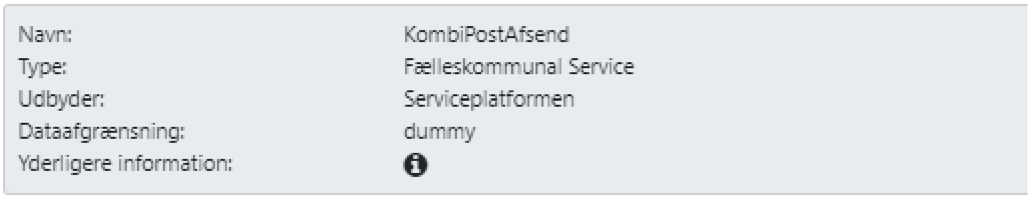
Certifikat	Udløb
KDI STS SFTP IBA Test2 (funkti...	2023-02-20

It is described in the get-started guide *Certifikater* how your NemID-administrator orders certificates for Test and Production respectively. Remember that here you register the public version of your certificate (*In your code you are using the private version of your certificate - the version that has a password-protected private key*).

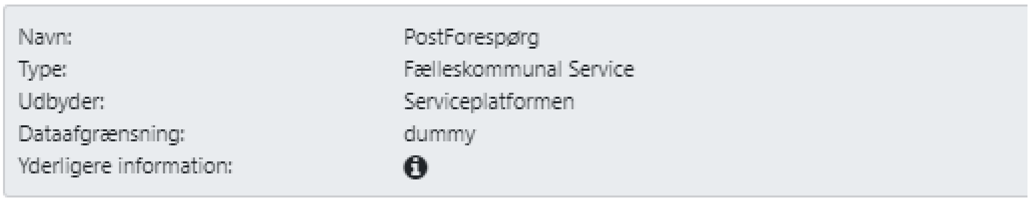
Be aware, that you must use test certificates in the test-environment *ExtTest*. Message Broker in the test environment will not work with certificates issued for production use.

We request a Service Agreement for each Municipality, that we need to send digital post on behalf of. Each Municipality must then approve their Service Agreement. In this example we are testing on behalf of authority KOMBIT. The services are named:

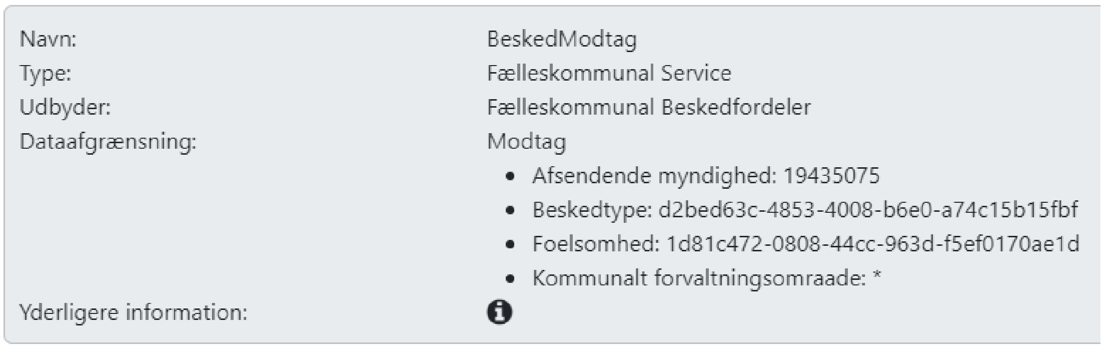
- *KombiPostAfsend* - Send digital and physical post
- *PostForespørg* - Query subscription status and organisations in NgDP
- *BeskedHent* - Message Broker fetch



Navn:	KombiPostAfsend
Type:	Fælleskommunal Service
Udbyder:	Serviceplatformen
Dataafgrænsning:	dummy
Yderligere information:	ℹ



Navn:	PostForespørg
Type:	Fælleskommunal Service
Udbyder:	Serviceplatformen
Dataafgrænsning:	dummy
Yderligere information:	ℹ



Navn:	BeskedModtag
Type:	Fælleskommunal Service
Udbyder:	Fælleskommunal Beskedfordeler
Dataafgrænsning:	Modtag • Afsendende myndighed: 19435075 • Beskedtype: d2bed63c-4853-4008-b6e0-a74c15b15fbf • Foelsomhed: 1d81c472-0808-44cc-963d-f5ef0170ae1d • Kommunalt forvaltningsomraade: *
Yderligere information:	ℹ

Values you need for the Message Broker Service Agreement:

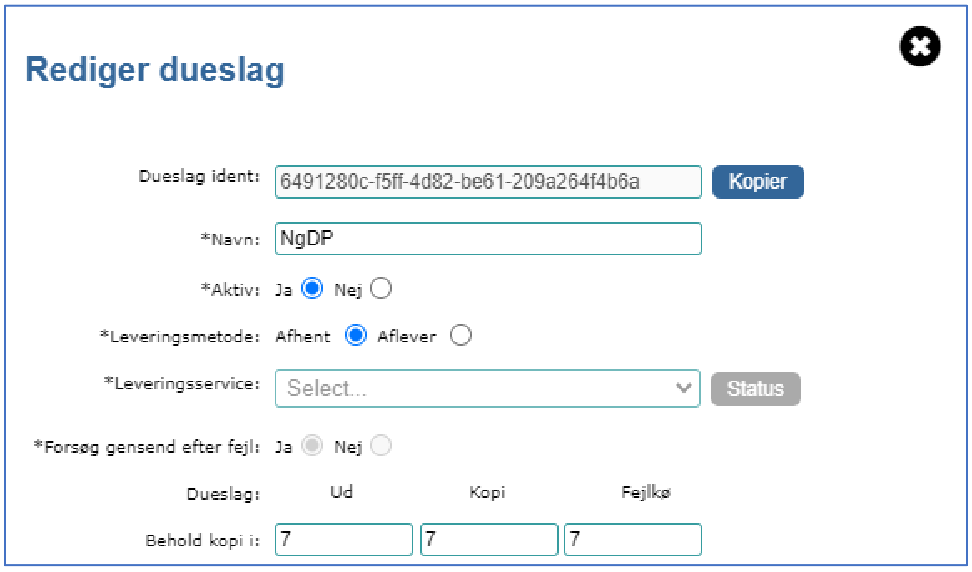
PKO_PostStatus - message type - `d2bed63c-4853-4008-b6e0-a74c15b15fbf`
Ikke fortrolige data - low confidentiality - `1d81c472-0808-44cc-963d-f5ef0170ae1d`

Once these Service Agreements are approved, we can request SAML-tokens that will grant us access to Message Broker and Post Web services on behalf of each authority.

1.7. Message Broker configuration

These tasks are performed in Message Broker UI. Serviceplatformen forwards requests to either NgDP or physical post supplier, and the validation takes place there. In case of NgDP, the MeMo messages are validated asynchronously. So you have to check the receipts in Message Broker in order to know, if final delivery was successful, or if there were validation errors you need to address. You can see examples of these receipts in Sending digital post.

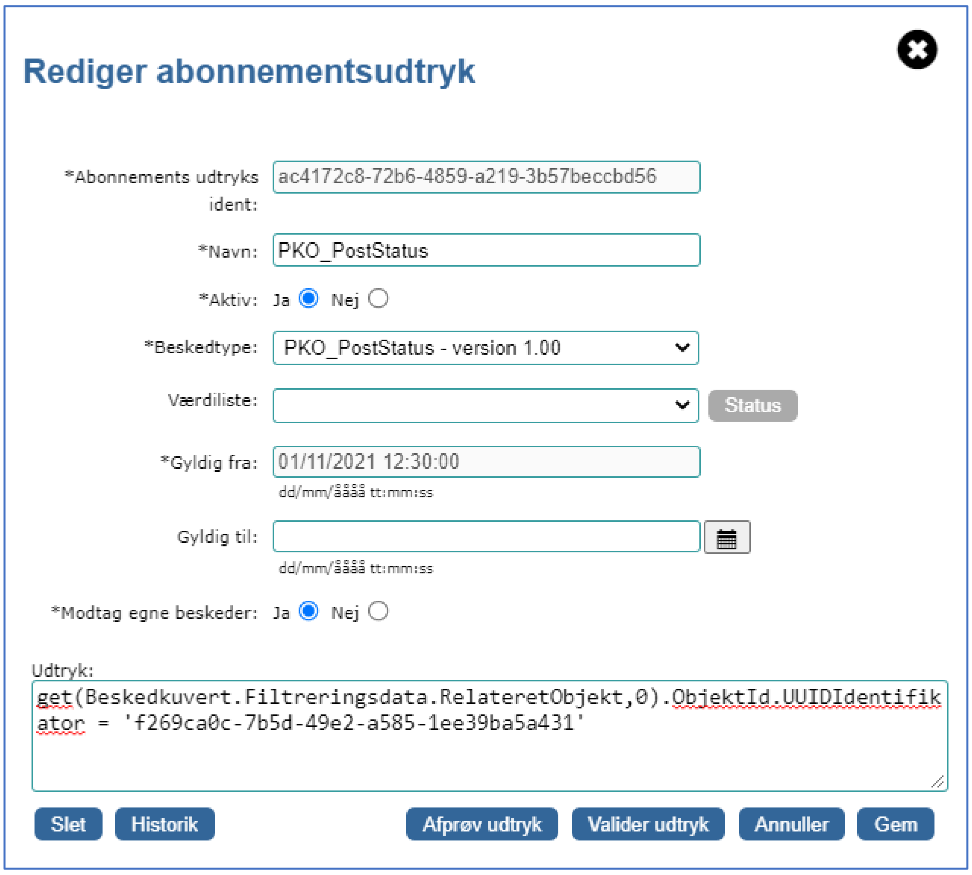
In this example we have created a “Dueslag” (queue):



Rediger dueslag

Dueslag ident:	6491280c-f5ff-4d82-be61-209a264f4b6a		
*Navn:	NgDP		
*Aktiv:	[✓ Ja] Nej		
*Leveringsmetode:	[✓ Afhent] Aflever		
*Leveringsservice:	Select...		
*Forsøg gensend efter fejl:	[✓ Ja] Nej		
Dueslag:	Ud	Kopi	Fejlkø
Behold kopi i:	7	7	7

With subscription to message type PKO_PostStatus



Rediger abonnementsudtryk

*Abonnements udtryks ident:	ac4172c8-72b6-4859-a219-3b57beccbd56
*Navn:	PKO_PostStatus
*Aktiv:	[✓ Ja] Nej
*Beskedtype:	PKO_PostStatus - version 1.00
Værdiliste:	
*Gyldig fra:	01/11/2021 12:30:00 dd/mm/åååå tt:mm:ss
Gyldig til:	 dd/mm/åååå tt:mm:ss
*Modtag egne beskeder:	[✓ Ja] Nej

Udtryk:

```
get(Beskedkuvert.Filtreringsdata.RelateretObjekt,0).ObjektId.UUIDIdentifik
ator = 'f269ca0c-7b5d-49e2-a585-1ee39ba5a431'
```

When Message Broker distributes messages of type PKO_PostStatus, it indicates the origin of the event (the IT-system that sent the Post the event is related to) in the element RelateretObjekt/ObjektId with the UUID of the IT-system.

```
<ns2:Haendelsesbesked>
...
   <ns2:RelateretObjekt>
      <ns2:ObjektId>
         <UUIDIdentifikator>f269ca0c-7b5d-49e2-a585-1ee39ba5a431</UUIDIdentifikator>
      </ns2:ObjektId>
      ...
   </ns2:RelateretObjekt>
...
</ns2:Haendelsesbesked>
```

In this case the UUID of the IT-system we have used in this guide. So by adding the following criteria in our subscription, we only receive copy of messages that are related to Post we have sent:

```
get(Beskedkuvert.Filtreringsdata.RelateretObjekt,0).ObjektId.UUIDIdentifikator = '<UUID of
your IT-system>'
```

2. SF1601 SEND POST EXAMPLES

Remember, that you must always specify these three mandatory HTTP request headers (see *Webservice OIOIDWS* guide):

- Authorization
- x-TransaktionsId
- x-TransaktionsTid

2.1. Subscription status - Digital Post

EP_SP1 Spoerg_tilmelding_digital_post_SP

Input is citizen identifier ”cprNumber” which has length 10. CPR is an akronym of the ”Central Person Register”.

Request

```
GET /service/PostForespoerg_1/digitalpost?cprNumber=1234567890 HTTP/1.1
Host: exttest.serviceplatformen.dk
Authorization: Holder-of-key ed0f32cd-d4e5-4216-b7ae-4f5e57a61650
x-TransaktionsId: c975cae5-8e7d-47f4-b264-c4d0b68ece4d
x-TransaktionsTid: 2021-12-03T09:03:33Z
```

Response

```
x-transaktionsid: 08d13a2f-35bf-4f70-aa78-16e77197a1a1
x-transaktionstid: 2021-12-07T10:29:12Z
Content-Type: application/json

{"result":false}
```

Note: if input is invalid format or there is no citizen with specified ID, the service will simply return ”false” - no error message.

To query subscription status of a company or an authority, use query parameter ”cvrNumber” instead. CVR is an acronym of the ”Central Virksomhed (Enterprise) Register”. The values have length 8.

2.2. Subscription status - NemSMS

EP_SP2 - Spoerg_tilmelding_nemsms_SP

Request

```
GET /service/PostForespoerg_1/nemsms?cprNumber=1234567890 HTTP/1.1
Host: exttest.serviceplatformen.dk
Authorization: Holder-of-key ed0f32cd-d4e5-4216-b7ae-4f5e57a61650
x-TransaktionsId: c975cae5-8e7d-47f4-b264-c4d0b68ece4d
x-TransaktionsTid: 2021-12-03T09:03:33Z
```

Response

```
x-transaktionsid: 08d13a2f-35bf-4f70-aa78-16e77197a1a1
x-transaktionstid: 2021-12-07T10:29:12Z
Content-Type: application/json

{"result":false}
```

Use query parameter “cvrNumber” instead to query subscription status of a company.

2.3. Organisations and contact points

EP_SP5 - Spoerg_organisation_digital_post_SP

How to lookup authorities and companies. A request with no conditions returns all organisations:

Request

```
GET /service/PostForespoerg_1/organisations/ HTTP/1.1
Host: exttest.serviceplatformen.dk
Authorization: Holder-of-key ed0f32cd-d4e5-4216-b7ae-4f5e57a61650
x-TransaktionsId: c975cae5-8e7d-47f4-b264-c4d0b68ece4d
x-TransaktionsTid: 2021-12-03T09:03:33Z
```

Response

```
{
  "currentPage" : 0,
  "totalPages" : 8,
  "elementsOnPage" : 100,
  "totalElements" : 761,
  "organisations" : [ {
    "id" : "14d6e9ad-fc0e-3918-0000-0000000001fe",
    "name" : "De Økonomiske Råd",
    "cvrNumber" : "90196359",
    "type" : "AUTHORITY",
    "authorityType" : "STATE",
    "authorityTerms" : true,
    "logoAvailable" : false,
    "targets" : [ "CITIZEN", "COMPANY", "AUTHORITY" ],
    "rightsIntroductionCompleted" : false
  }, { ...
```

100 results per page and 8 pages total in this examples. You can then do paging by specifying page number e.g. ”?page=2”.

Here we filter on name (there is implicit wildcard before and after the search-value):

Request

```
GET /service/PostForespoerg_1/organisations/?name=KOMBIT HTTP/1.1
Host: exttest.serviceplatformen.dk
Authorization: Holder-of-key ed0f32cd-d4e5-4216-b7ae-4f5e57a61650
x-TransaktionsId: c975cae5-8e7d-47f4-b264-c4d0b68ece4d
x-TransaktionsTid: 2021-12-03T09:03:33Z
```

Response

```
{
  "currentPage" : 0,
  "totalPages" : 1,
  "elementsOnPage" : 1,
  "totalElements" : 1,
  "organisations" : [ {
    "id" : "81dd3d9e-70ee-46e9-a305-1719186e7bc7",
    "name" : "KOMBIT",
    "cvrNumber" : "19435075",
    "type" : "AUTHORITY",
    "authorityType" : "OTHER",
    "authorityTerms" : true,
    "logoAvailable" : false,
    "targets" : [ "AUTHORITY", "CITIZEN", "COMPANY" ],
    "rightsIntroductionCompleted" : false
  } ]
}
```

Alternatively you can find the organisation of interest by instead specifying ”?cvrNumber=xxxxxxxx”. Now that we have the ”id” of the organisation, we can query contact points:

Request

```
GET /service/PostForespoerg_1/organisations/81dd3d9e-70ee-46e9-a305-1719186e7bc7/contact-
points/ HTTP/1.1
Host: exttest.serviceplatformen.dk
Authorization: Holder-of-key ed0f32cd-d4e5-4216-b7ae-4f5e57a61650
x-TransaktionsId: c975cae5-8e7d-47f4-b264-c4d0b68ece4d
x-TransaktionsTid: 2021-12-03T09:03:33Z
```

Response

```
{
   ...
  "contactPoints" : [ {
    "id" : "4302bc81-93d2-4490-b425-ddf9374606a4",
    "name" : "KOMBIT Pas test",
    ...
    "active" : true,
    "visible" : true,
    ...
    "organisationId" : "81dd3d9e-70ee-46e9-a305-1719186e7bc7",
    "contactPointCodes" : [ ],
    "contactGroups" : [ {
      "id" : "79a592a7-6249-49bd-a02f-16e98638d86b",
      "organisationalUnit" : false,
      "name" : "KOMBIT Kommune",
      "description" : "TEST",
      "organisation" : {
        "id" : "81dd3d9e-70ee-46e9-a305-1719186e7bc7",
        "name" : "KOMBIT",
        "cvrNumber" : "19435075",
        "type" : "AUTHORITY",
         ...
      },
      "targets" : [ "CITIZEN", "COMPANY", "AUTHORITY" ],
      "postkasseIds" : [ ]
    } ],
      ...
```

The contact points are used with SF1606 Receive Post and are specified by the authorities as routing rules. This is described in SF1606 Programmer’s Guide.

2.4. Send digital post

EP_SP3 Afsend_postforsendelse_SP_kombi

Request

```
POST /service/KombiPostAfsend_1/kombi HTTP/1.1
Host: exttest.serviceplatformen.dk
Authorization: Holder-of-key ed0f32cd-d4e5-4216-b7ae-4f5e57a61650
x-TransaktionsId: c975cae5-8e7d-47f4-b264-c4d0b68ece4d
x-TransaktionsTid: 2021-12-03T09:03:33Z
Content-type: application/xml
Accept: application/xml

<kombi_request>
   <KombiValgKode>Digital Post</KombiValgKode>
   <memo:Message xmlns:memo="https://DigitalPost.dk/MeMo-1" memoVersion="1.1"
memoSchVersion="1.1.0">
      <memo:MessageHeader>
         <memo:messageType>DIGITALPOST</memo:messageType>
         <memo:messageUUID>e4cf3010-d051-43f2-8aa2-c86b05f3af71</memo:messageUUID>
         <memo:label>Besked #234</memo:label>
         <memo:mandatory>false</memo:mandatory>
         <memo:legalNotification>false</memo:legalNotification>
         <memo:Sender>
            <memo:senderID>19435075</memo:senderID>
            <memo:idType>CVR</memo:idType>
            <memo:label>KOMBIT</memo:label>
         </memo:Sender>
         <memo:Recipient>
            <memo:recipientID>1705880000</memo:recipientID>
            <memo:idType>CPR</memo:idType>
         </memo:Recipient>
      </memo:MessageHeader>
      <memo:MessageBody>
         <memo:createdDateTime>2021-12-09T10:31:35Z</memo:createdDateTime>
         <memo:MainDocument>
            <memo:File>
               <memo:encodingFormat>text/plain</memo:encodingFormat>
               <memo:filename>Message.txt</memo:filename>
               <memo:language>da</memo:language>
   <memo:content>RW4gbnllIGJlc2tlZA0KTWVkIGxpbmplc2tpZnQNClZpcmtlciBkZXQ/</memo:content>
            </memo:File>
         </memo:MainDocument>
      </memo:MessageBody>
   </memo:Message>
</kombi_request>
```

- We set KombiValgKode = “Digital Post”
- You must specify a new messageUUID for each message. Sending new messages with same UUID as a previous message will be ignored.
- The “label” attribute is actually message *subject*
- You must specify the CVR-number of the authority you are sending on behalf of

- Recipient in this example is a test-citizen we created in the NgDP test environment

Response

```
<kombi_response xmlns:ns2="http://kombit.dk/xml/schemas/kontekst/2017/01/01/">
   <Result>true</Result>
   <TransmissionID>b76dd95f-8bc6-4d0e-89fe-9618ec2dee5c</TransmissionID>
   <HovedoplysningerSvarREST>
      <SvarReaktion>
         <Advis>
            <AdvisId>DigitalPost</AdvisId>
            <AdvisTekst>RECEIVED</AdvisTekst>
         </Advis>
      </SvarReaktion>
   </HovedoplysningerSvarREST>
</kombi_response>
```

The status “Received” means that the request has been acknowledged by Serviceplatformen and delivered to the NgDP service. The validation and processing of the message takes place afterwards in NgDP and the result is returned via Message Broker. The parameter *TransmissionID* links the receipt to the request.

If an error occurs the response looks as follows, here example when trying to send on behalf of an authority that has not registered their NgDP API key at Serviceplatformen:

Error response

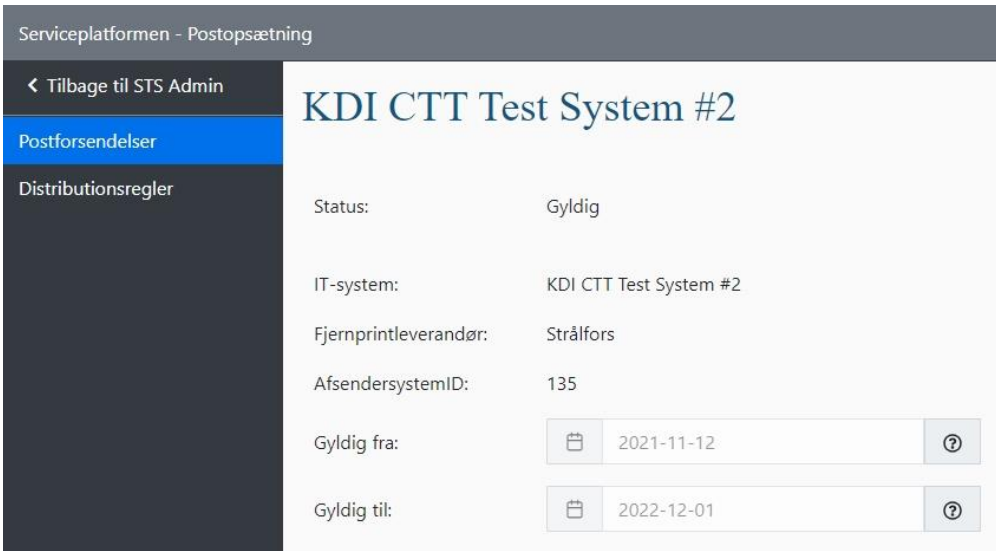
```
<error_response xmlns:ns2="http://kombit.dk/xml/schemas/kontekst/2017/01/01/">
   <HovedoplysningerSvarREST>
      <SvarReaktion>
         <Fejl>
            <FejlId>DP0005</FejlId>
            <FejlTekst>Server internal error during source system communication:
null</FejlTekst>
            <KildeId>19dc3a1d-c629-4853-b566-7fe49cf674ce</KildeId>
         </Fejl>
      </SvarReaktion>
   </HovedoplysningerSvarREST>
</error_response>
```

Regarding allowed MIME-types in MeMo, please see [REF2] “*Tilladte filtyper i Digital Post.pdf*”. Currently the list is:

MainDocument	
PDF	application/pdf
HTML	text/html
TXT	text/plain
AdditionalDocument	
PDF	application/pdf
HTML	text/html
TXT	text/plain
DOC	application/msword
DOCX	application/vnd.openxmlformats-officedocument.wordprocessingml.document
RTF	application/rtf
BMP	image/bmp
GIF	image/gif
JPG	image/jpg
PNG	image/png
TIF	image/tiff
XLS	application/vnd.ms-excel
XLSX	application/vnd.openxmlformats-officedocument.spreadsheetml.sheet
ODT	application/vnd.oasis.opendocument.text
ODS	application/vnd.oasis.opendocument.spreadsheet
TechnicalDocument	
XML	application/xml
XML	text/xml

2.5. Send physical post

For details on physical post data model, please see the documentation for SF1600. Before you can send from your IT-system on behalf of an authority, they need to create a “Postforsendelse” rule in Serviceplatformen administration. Here is an example of a rule created for our example IT-system on behalf of authority KOMBIT, that directs physical post to Strålfors:



KDI CTT Test System #2

Status:	Gyldig
IT-system:	KDI CTT Test System #2
Fjernprintleverandør:	Strålfors
AfsendersystemID:	135
Gyldig fra:	2021-11-12
Gyldig til:	2022-12-01

The following is a minimum example that you can use to test, that messages are delivered correctly:

Request

```
POST /service/KombiPostAfsend_1/kombi HTTP/1.1
Host: exttest.serviceplatformen.dk
Authorization: Holder-of-key ed0f32cd-d4e5-4216-b7ae-4f5e57a61650
x-TransaktionsId: c975cae5-8e7d-47f4-b264-c4d0b68ece4d
x-TransaktionsTid: 2021-12-03T09:03:33Z
Content-type: application/xml
Accept: application/xml

<kombi_request>
   <KombiValgKode>Fysisk Post</KombiValgKode>
   <ns1:ForsendelseISamling
      xmlns:ns1="urn:oio:fjernprint:1.0.0"
      xmlns:ns2="urn:oio:dkal:1.0.0"
      xmlns:ns3="urn:oio:adir:dagpenge:2009.07.01"
      xmlns:ns4="http://rep.oio.dk/itst.dk/xml/schemas/2006/01/17/"
      xmlns:ns5="http://rep.oio.dk/ebxml/xml/schemas/dkcc/2005/03/15/"
      xmlns:ns6="http://rep.oio.dk/ebxml/xml/schemas/dkcc/2003/02/13/">
      <ns1:ForsendelseI>
         <ns1:AfsendelseIdentifikator>dc3303Soa44e-9c1c-
46aaf36974de</ns1:AfsendelseIdentifikator>
         <ns1:ForsendelseTypeIdentifikator>265</ns1:ForsendelseTypeIdentifikator>
         <ns1:ForsendelseModtager>
            <ns2:AfsendelseModtager>
               <ns3:CPRnummerIdentifikator>0000000000</ns3:CPRnummerIdentifikator>
            </ns2:AfsendelseModtager>
            <ns1:ModtagerAdresse>
               <ns4:PersonName>Test Testesen</ns4:PersonName>
               <ns5:StreetName>Testvej</ns5:StreetName>
               <ns6:StreetBuildingIdentifier>3</ns6:StreetBuildingIdentifier>
               <ns5:PostCodeIdentifier>2300</ns5:PostCodeIdentifier>
               <ns6:CountryIdentificationCode scheme="iso3166-
alpha2">DK</ns6:CountryIdentificationCode>
            </ns1:ModtagerAdresse>
         </ns1:ForsendelseModtager>
         <ns2:FilformatNavn>pdf</ns2:FilformatNavn>
         <ns2:MeddelelseIndholdData>JVBERi0...VPRg==</ns2:MeddelelseIndholdData>
         <ns1:TransaktionsParametreI/>
         <ns1:DokumentParametre/>
      </ns1:ForsendelseI>
   </ns1:ForsendelseISamling>
</kombi_request>
```

- We set KombiValgKode = “Fysisk Post”
- *AfsendelseIdentifikator* is the equivalent of messageUUID in MeMo
- For value of *ForsendelseTypeIdentifikator* please consult the authority
- Use dummy value for *CPRnummerIdentifikator* (0000000000) when sending to citizen
- Even if *TransaktionsParametreI* and *DokumentParametre* are empty elements, they must be included or else schema validation will fail.

Response

```
<kombi_response xmlns:ns2="http://kombit.dk/xml/schemas/kontekst/2017/01/01/">
   <Result>true</Result>
   <HovedoplysningerSvarREST>
      <SvarReaktion>
         <Advis>
            <AdvisId>FjernprintStrålfors</AdvisId>
            <AdvisTekst>true</AdvisTekst>
         </Advis>
      </SvarReaktion>
   </HovedoplysningerSvarREST>
</kombi_response>
```

Note that values for *AdvisId* and *AdvisTekst* differ from digital post response. If the validation failes the response looks as follows:

Error response

```
<error_response xmlns:ns2="http://kombit.dk/xml/schemas/kontekst/2017/01/01/">
   <HovedoplysningerSvarREST>
      <SvarReaktion>
         <Fejl>
            <FejlId>DP00016</FejlId>
            <FejlTekst>Source System responded with error: : Skemafejl i afsendt besked.
(The 'urn:oio:adir:dagpenge:2009.07.01:CPRnummerIdentifikator' element is invalid - The
value '000000000' is invalid according to its datatype
'http://rep.oio.dk/cpr.dk/xml/schemas/core/2005/03/18/:PersonCivilRegistrationIdentifierType
' - The Pattern constraint failed. Line: 7 Column: 8)</FejlTekst>
            <KildeId>afd21f3d-11c7-4f51-b2a6-f31d6480a9fb</KildeId>
         </Fejl>
      </SvarReaktion>
   </HovedoplysningerSvarREST>
</error_response>
```

2.6. Message broker events

In the messages you receive from Message Broker, the receipt is stored in message element /Beskeddata/Base64 and you need to decode the value. For details, please see "*PKO_PostStatus.pdf*" in the documentation package.Here are two examples:

Example of positive receipt - delivered

```
<PKO_PostStatus xmlns="http://serviceplatformen.dk/xml/print/PKO_PostStatus/1/types">
   <TransmissionId>b76dd95f-8bc6-4d0e-89fe-9618ec2dee5c</TransmissionId>
   <MessageUUID>e4cf3010-d051-43f2-8aa2-c86b05f3af71</MessageUUID>
   <KanalKode>Digital Post</KanalKode>
   <TransaktionsDatoTid>2021-12-09T09:31:37.461Z</TransaktionsDatoTid>
   <TransaktionsStatusKode>Afleveret Digital Post</TransaktionsStatusKode>
   <CorrelationId>b76dd95f-8bc6-4d0e-89fe-9618ec2dee5c</CorrelationId>
   <FejlDetaljer/>
</PKO_PostStatus>
```

Example of negative receipt - validation error

```
<PKO_PostStatus xmlns="http://serviceplatformen.dk/xml/print/PKO_PostStatus/1/types">
   <TransmissionId>686b7828-a2d6-44f7-bfa1-8d24401bd6bb</TransmissionId>
   <MessageUUID>ad3acfe4-e82f-40bb-b892-7040436a52ad</MessageUUID>
   <KanalKode>Digital Post</KanalKode>
   <TransaktionsDatoTid>2021-11-26T08:58:13.005Z</TransaktionsDatoTid>
   <TransaktionsStatusKode>Fejlet</TransaktionsStatusKode>
   <CorrelationId>686b7828-a2d6-44f7-bfa1-8d24401bd6bb</CorrelationId>
   <FejlDetaljer>
      <FejlTekst>javax.xml.stream.XMLStreamException:
com.fasterxml.jackson.core.JsonParseException: tag name "filename" is not allowed. Possible
tag names are: &lt;encodingFormat&gt;
 at [row,col {unknown-source}]: [22,9]
 at [Source: (com.ctc.wstx.sr.ValidatingStreamReader); line: 22, column: 24]</FejlTekst>
   </FejlDetaljer>
</PKO_PostStatus>
```

Note the following metadata in the message envelope, that you might find useful when parsing received messages:

ObjektHandling

Same as *TransaktionsStatusKode* in Beskeddata but in UUID-representation. Here example with UUID for "*Afleveret Digital Post*" (see PKO_PostStatus.pdf).

```
...
  <ns2:Filtreringsdata>
    <ns2:ObjektRegistrering>
      <ns2:ObjektHandling>
        <UUIDIdentifikator>f7161a89-5068-4023-bc80-d7f4daad2a2e</UUIDIdentifikator>
...
```

BeskedAnsvarligAktoer

Indicates the source of the event. Here example with UUID for "Digital Post".

```
...
<ns2:Filtreringsdata>
   ...
   <ns2:BeskedAnsvarligAktoer>
      <UUIDIdentifikator>d92bb9c2-d826-4028-8fb5-d8b88b4377d4</UUIDIdentifikator>
...
```

2.7. Using automatic channel selection

An alternative pattern is to use automatic channel selection, where Serviceplatformen will query and determine whether a citizen or an authority wants to receive digital or physical post, and then send the Post accordingly. Here we simply specify KombiValgKode = "Automatisk Valg":

```
<kombi_request>
   <KombiValgKode>Automatisk Valg</KombiValgKode>
   <fjernprint:ForsendelseISamling>
      <fjernprint:ForsendelseI>
      ...
      </fjernprint:ForsendelseI>
   </fjernprint:ForsendelseISamling>
   <memo:Message>
      ...
   </memo:Message>
</kombi_request>
```

Namespaces removed in example for clarity. You then need to specify both *ForsendelseI* (physical post) and *Message* (digital post), since you don't know which one will be used. Be aware that the two data models differ in structure and naming, so you have to figure out how to map the same message into both models.

2.8. Replies and threaded messages

You allow a recipient to reply to a message by simply adding *reply*=true right after *label* in the MeMo message header. Replies are directed back via the contact point you specify under *Sender* and there must be a Distribution Rule that matches which points to your receiving system (see SF1606 Programmer's Guide). You add metadata in *ReplyData*. This is described in [REF2] *Vejledning til dataopmærkning i MeMo.pdf* chapter 3.7 *ReplyData*. On the same page you find *MeMo_Scenarier_og_Eksempler.zip* that contains a practical example (Scenarie 4) of a message thread.

The basic elements in threaded messaging

```
<Message>
   <MessageHeader>
      ...
      <label>Svar på Underretning</label>
      <reply>true</reply>
      ...
      <Sender>
         ...
         <ContactPoint>
            ...
         </ContactPoint>
      </Sender>
      <Recipient>
         ...
         <ContactPoint>
            ...
         </ContactPoint>
      </Recipient>
      <ReplyData>
         ...
      </ReplyData>
   </MessageHeader>
   <MessageBody>
      ...
   </MessageBody>
</Message>
```

Refer to the two mentioned ressources for more information.

3. APPENDIX - CONFIGURATION PARAMETERS

For environment ExtTest.

Security Token Service

Endpoint	https://adgangsstyring.eksterntest-stoettesystemerne.dk/runtime/services/kombittrust/14/certificatemixed
Issuer	https://adgangsstyring.eksterntest-stoettesystemerne.dk/
Certificate subject	test-ekstern-adgangsstyring (funktionscertifikat)
Certificate thumbprint	7002cf221d1d3979eca623599e43e0b6b4c8920c

Access Token Service

Endpoint	https://exttest.serviceplatformen.dk/service /AccessTokenService_1/token

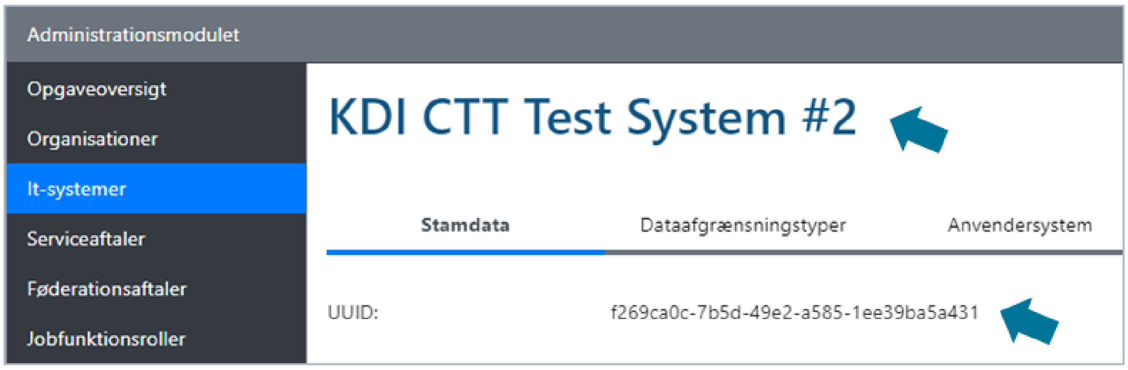
Message Broker (Beskedfordeler)

Endpoint	beskedfordeler.eksterntest-stoettesystemerne.dk
Port	5671
Entity ID	http://beskedfordeler.eksterntest-stoettesystemerne.dk/service/afhent/1
Virtual host name	BF
Your Queue	<insert here> ("Dueslag" UUID)

Your IT-system

Name	<insert here>
UUID	<insert here>
Client certificate thumbprint	<insert here>
AMQP Queue	<insert here>

Parameters of your IT-system are found in Serviceplatformen administration:

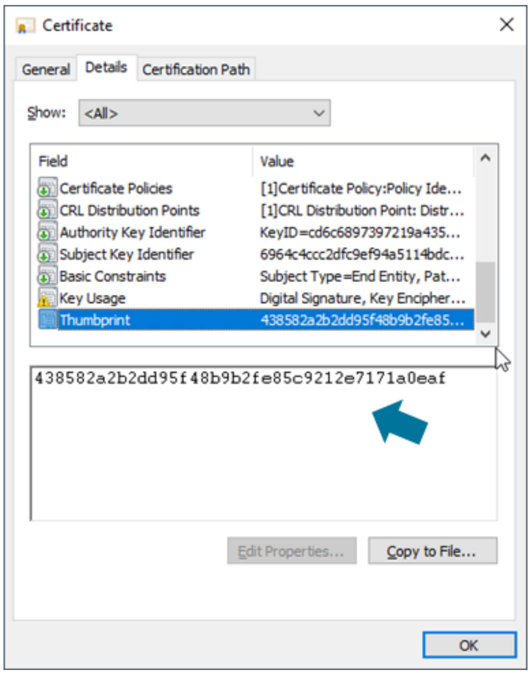


KDI CTT Test System #2

UUID: f269ca0c-7b5d-49e2-a585-1ee39ba5a431

You need the UUID of your IT-system when applying a filter on your queue-subscription in Message Broker UI.

Thumbprint of your client certificate you find under ”Details -> Thumbprint”:

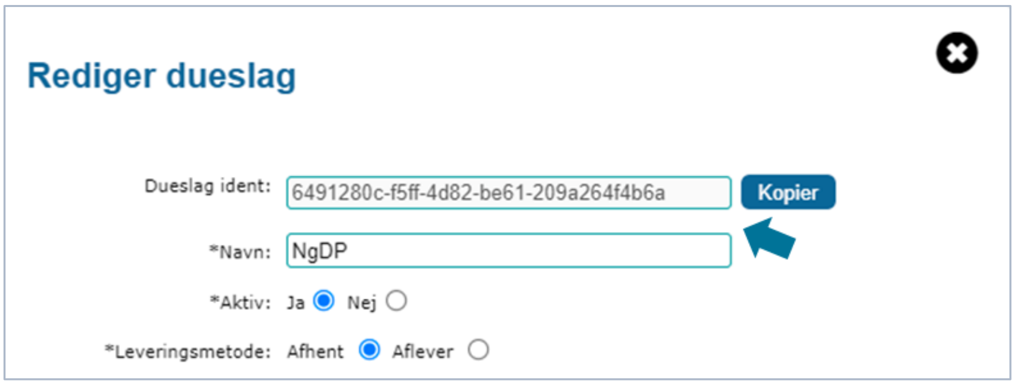


Show: <All>

Field	Value
Certificate Policies	[1]Certificate Policy:Policy Ide...
CRL Distribution Points	[1]CRL Distribution Point: Distr...
Authority Key Identifier	KeyID=cd6c6897397219a435...
Subject Key Identifier	6964c4ccc2dfc9ef94a5114bdc...
Basic Constraints	Subject Type=End Entity, Pat...
Key Usage	Digital Signature, Key Encipher...
Thumbprint	438582a2b2dd95f48b9b2fe85...

```
438582a2b2dd95f48b9b2fe85c9212e7171a0eaf
```

You find the Message Broker queue ID on the “Dueslag” you created:



Rediger dueslag

Dueslag ident:	6491280c-f5ff-4d82-be61-209a264f4b6a
*Navn:	NgDP
*Aktiv:	[✓ Ja] Nej
*Leveringsmetode:	[✓ Afhent] Aflever

4. APPENDIX - CREATING DATA CLASSES FROM XSD

You can create data classes from the XML schema’s using the xsd.exe tool from Visual Studio. Simply refer to all the schema files in an XML-dokument like this (example from Message Broker message envelope schema):

Example xsd.exe parameters file - Message envelope schema

```
<xsd xmlns="http://microsoft.com/dotnet/tools/xsd/">
   <generateClasses language="CS">
      <schema>1.1\OrganisationFaelles.xsd</schema>
      <schema>1.1\SagDokObjekt.xsd</schema>
      <schema>1.1\Part.xsd</schema>
      <schema>cached\xmldsig-core-schema.xsd</schema>
      <schema>.\Beskedkuvert.xsd</schema>
   </generateClasses>
</xsd>
```

Note that the last file is referred to with ".\" in front. This little hack will name the output file accordingly (*Beskedkuvert.cs*). If you don't do this, it will try to name the output file as a concatenation of all referenced schema-files and the filename will be too long. From a Visual Studio Developer prompt run the following command:

```
xsd.exe /p:xsd_exe_parameters.xsd /classes /e:Haendelsesbesked
```

In parameter "/e:" you specify the name of the root element. You can do likewise with the Fjernprint schema files that you find in SF1600 documentation package, or the MeMo schema files you find in [REF2].

VERSIONSHISTORIK

Version	Dato	Ændringer
1.0	08-02-2022	Updated version in new template
1.1	25-08-2022	Updated the section 1.2 Compatible frameworks. It is now recommended to use REST when fetching a SAML-token.