DIGITALISERINGS KATALOGET

SF1606 PROGRAMMER'S GUIDE

Examples and practicalities



KOMB:T

Kommunernes it-fællesskab

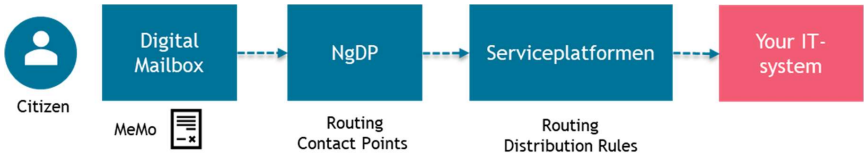
1. INTRODUCTION 3
1.1. Prerequisites 3
1.2. Compatible frameworks 4
1.3. References 4
1.4. Employee-signature and NemLog-In roles 4
2. SERVICEPLATFORMEN CONFIGURATION 5
2.1. Register your IT-system 5
2.2. Request a Service Agreement 5
2.3. Register the callback endpoint 6
3. MUNICIPALITY TASKS 6
3.1. Digital Post - We registered Serviceplatformen as receiver-system 7
3.2. Digital Post - We registered a contact point 8
3.3. Serviceplatformen - We created a default receiver-system 9
3.4. Serviceplatformen - We registered a Distribution rule 10
4. ESTABLISH THE WEBSERVICE 11
4.1. Establish hosting 11
4.2. Implement 2-way-TLS 11
4.3. Implement code to accept message 11
4.4. Implement Ok response 11
4.5. Implement Error response 12
5. TEST THE SETUP 12
5.1. Send message from a citizen mailbox 12
5.2. Send message from a SF1601 client 13
5.3. Check reveived messages 14
6. APPENDIX 15
6.1. Message log in Digital Post at Digitaliseringsstyrelsen 15
6.2. How to create test citizens 16
VERSION HISTORY 17

1. INTRODUCTION

This guide is targeted developers implementing SF1606 Modtag (Receive) Post. It features examples of the most common use cases and practical information. Make sure to become acquainted with the "Kom-godt-i-gang" (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 "*SF1606 - Vejledning til tilslutning og test af modtagelse af post.pdf*" that you find in the documentation package for SF1606.

Each municipality registers Serviceplatformen as a receiver-system in Digital Post at Digitaliseringsstyrelsen. It then specifies contact points having Serviceplatformen as receiver-system. When a citizen sends a MeMo to one of these contact points, the message is routed to Serviceplatformen. Once received at Serviceplatformen, distribution rules defined by each municipality determine to which IT-system the message is delivered.



Thus each municipality can specify by configuration in Digital Post at Digitaliseringsstyrelsen and Serviceplatformen respectively, where they would like particular messages delivered.

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. Requested *PostModtag* (Post Receive) Service Agreement with the municipality
- Your pilot-municipality has
  a. Registered Serviceplatformen as *Modtager* (receiver) IT-system in Digital Post at Digitaliseringsstyrelsen
  b. Created contact points in Digital Post at Digitaliseringsstyrelsen connected to Serviceplatformen
  c. Has approved the Service Agreement
  d. Created distribution rules pointing to your system
- You, the developer, have

a. Read and understood the Get-Started guides referenced below

1.2. Compatible frameworks

You have to establish a webserver and an endpoint to which MeMo messages are delivered. You can use any HTTP webserver framework that supports 2-way-TLS (client-certificate authentication). The received MeMo message is XML.

1.3. References

[REF1] Get-Started guides

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

Reading and understanding the guide *Certifikater* is essential. The guide is in Danish. Here is described by example how to implement 2-way-TLS.

[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 Digital Post

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] SF1606 - Vejledning til tilslutning og test af modtagelse af post.pdf

Detailed description of municipality tasks (in Danish).

1.4. Employee-signature and NemLog-In roles

Access to Serviceplatformen Administration 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

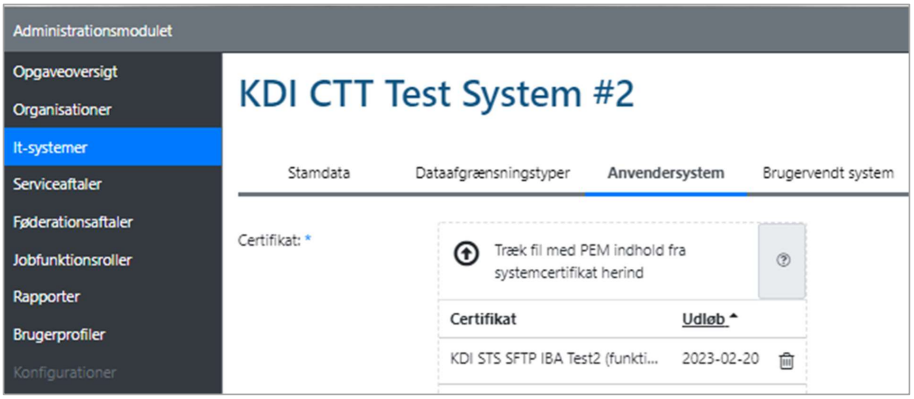
Usually one or two persons within an organisation are granted these roles, so you have to refer to them regarding the related tasks described in this document.

2. SERVICEPLATFORMEN CONFIGURATION

These tasks are performed in Serviceplatformen Administration.

2.1. Register your IT-system

Register your IT-system in Serviceplatformen Administration with a client (FOCES) certificate:



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. Be aware, that you must use test certificates in the test-environment *ExtTest*.

2.2. Request a Service Agreement

Request a Service Agreement for each Municipality, that you need to receive 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 service is:

Name in ADM: PostModtag

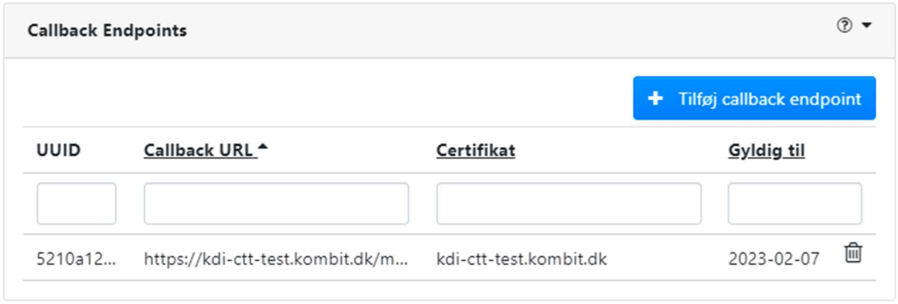
UUID: 9fa1a826-049f-43dc-b573-3294b71683af

Navn:	PostModtag
Type:	Fælleskommunal Service
Udbyder:	Serviceplatformen
Dataafgrænsning:	dummy
Yderligere information:	

Once Service Agreement is approved, the IT-system will automatically be available for selection in the distribution rules interface (you may have to specify the callback endpoint first as well).

2.3. Register the callback endpoint

Register the callback endpoint on your "Anvendersystem" along with the public version of the HTTPS-certificate you are using on the endpoint. Here is the endpoint we registered for this test:



Once you have completed these tasks, each municipality can now create distribution rules that will forward messages to your IT-system. For reference, all municipality tasks are described in the following chapter.

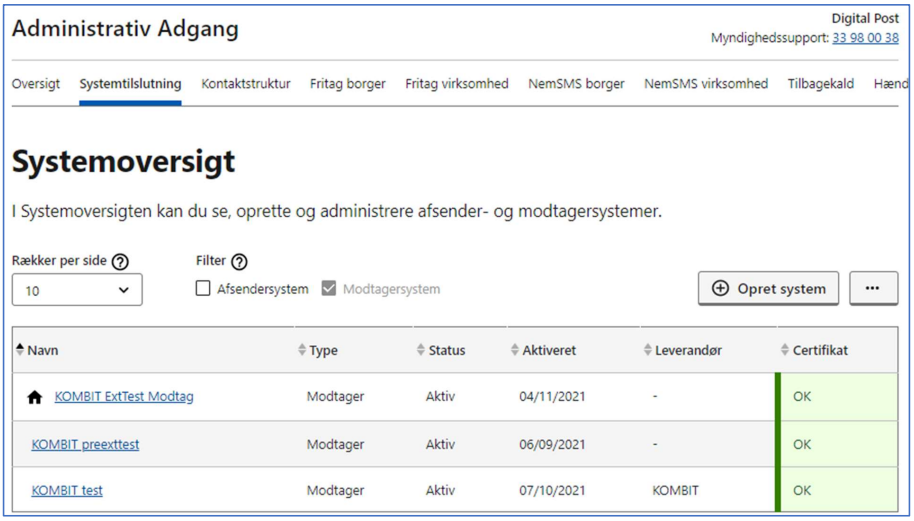
3. MUNICIPALITY TASKS

This is a brief description of the tasks we completed as an authority, in order to be able to receive digital post in our test-system. For testing purposes we had KOMBIT registered as an authority in Digital Post at Digitaliseringsstyrelsen test-environment. The municipality you are testing with must complete the same tasks. If you are assisting a municipality with this, you find more details in [REF5].

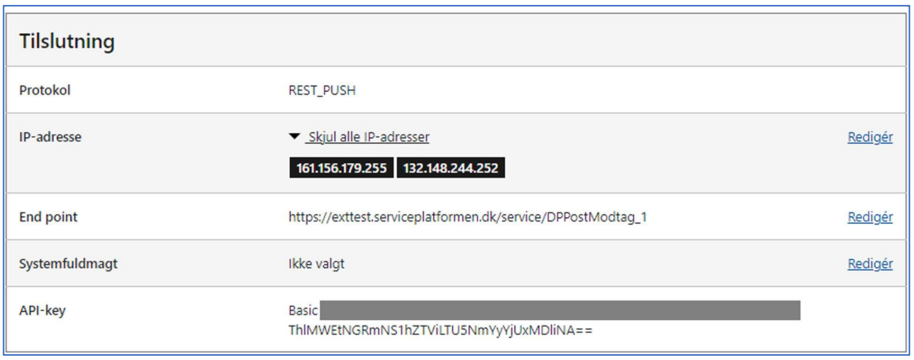
3.1. Digital Post - We registered Serviceplatformen as receiver-system

This configuration is done in https://admin.test.digitalpost.dk/. It is described in Technical Integration NgDP chapter 13 *Access to the administration portal on the test environment*, how the municipality gets access to the site.

Here is the overview of registered sender- and receiversystems:



Here we (KOMBIT) registered Serviceplatformen as receiver-system:

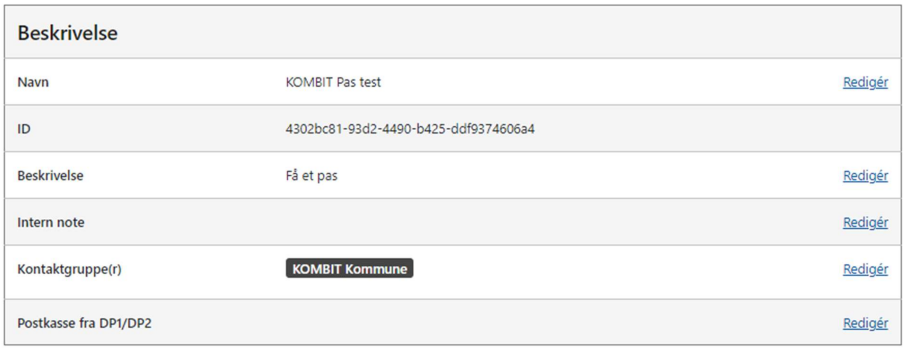


3.2. Digital Post - We registered a contact point

Here is the contact point “KOMBIT Pas test”, that we created for testing purposes:



Below we see the contact point UUID (ID), that we need to create the Distribution rule in Serviceplatformen:



We selected Målgruppe (target group) to be all three categories - Authority, Citizen, Company:



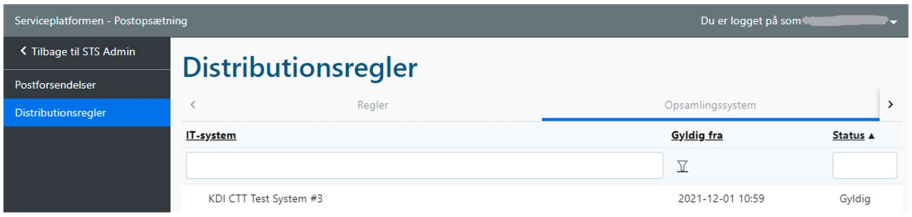
Here we see that this contact point is connected to the receiver-system, that we created for Serviceplatformen:



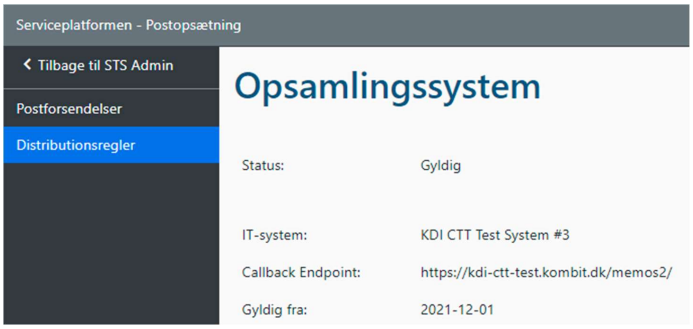
Now messages sent to this contact point will by routed by Digital Post at Digitaliseringsstyrelsen to Serviceplatformen on behalf of our authority KOMBIT.

3.3. Serviceplatformen - We created a default receiver-system

This is important to know - if Serviceplatformen receives a message, that does not match a distribution rule, it will be delivered to the *Opsamlingssystem* (default receiver-system). Provided that the municipality has specified a such. It is the right tab in *Distributionsregler*:



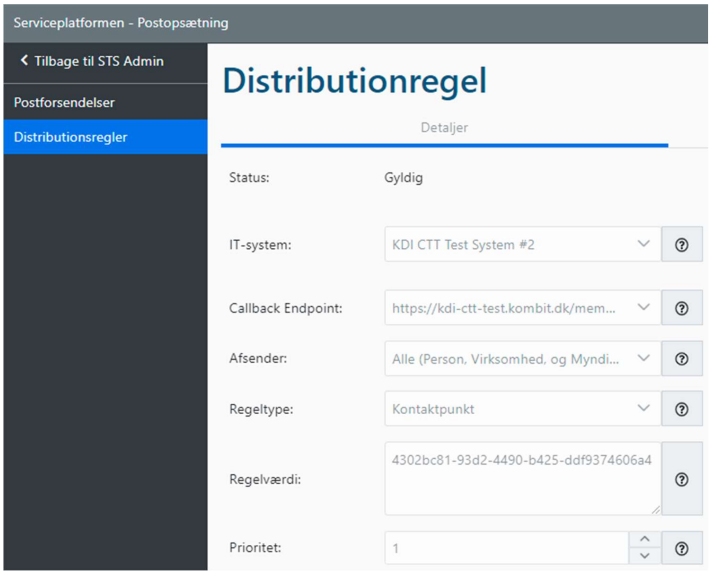
And here we have the default receiver-system we created for KOMBIT:



In our test setup it points to a different endpoint on our server, so that we can catch messages that do not match a distribution rule.

3.4. Serviceplatformen - We registered a Distribution rule

Here is the distribution rule we created. We specified Regeltype (rule type) = “Kontaktpunkt” (contact point) and Regelværdi (rule value) the UUID of the contact point we created.



Now all the prerequisites are in place, and we are ready to send test-messages to our IT-system.

4. ESTABLISH THE WEBSERVICE

4.1. Establish hosting

The endpoint to which you want to receive messages must be reachable from Serviceplatformen, so you have to establish a webservice that it can reach. This is of course obvious, but we have included the task here because it may require planning and budgetting in your company to get it done. For development and testing purposes, and for implementation in production. Since the webservice is exposed to the outside, don't forget to apply appropriate security measures.

4.2. Implement 2-way-TLS

Configure your webservice to accept and require client-certificate authentication, which is the same as 2-way-TLS, establishing a secure connection based on mutual trust. You find a practical example in [REF1] demonstrating how to do it with Nginx and PHP. Serviceplatformen authenticates itself with the callback-certificate, that you find in the certificate packages for exttest and production respectively. Make sure to validate the certificate e.g. by checking the thumbprint/fingerprint. You are required to use a valid HTTPS-certificate - it cannot be self-signed or expired, and it must be issued to the domain.

4.3. Implement code to accept message

In our test-setup we simply dump the request to a file, so we can inspect the received message. How you implement parsing, validation and persistance of the message is entirely up to you. Of course it would be a good idea to map the request body into a MeMo-Lib object [REF3]. Make sure to register the two x-Transaktion HTTP headers along with the message.

4.4. Implement Ok response

Kommenterede [F1]: Skal opdateres med, at det også er muligt at sende 202 retur. Kan eventuelt være 4.1.1

When a message is successfully parsed, validated and persisted, reply with HTTP 200 reponse code and a *memos_response / HovedoplysningerSvarREST / Advis* element. For details see the guide Transaktionsspor og fejlbehandling ved servicekald (in Danish). We echo the two X-Transaktion headers we received in the request:

Response HTTP-headers

```
...
X-Transaktionstid: 2022-02-09 00:09:21.695
X-Transaktionsid: 620306310e53f9f7291f11b42ba3dcd0
Content-Type: application/xml
...
```

And here is a sample response:

memos_response example - OK

```
<memos_response>
   <HovedoplysningerSvarREST xmlns="http://kombit.dk/xml/schemas/kontekst/2017/01/01/">
      <SvarReaktion>
         <Advis>
            <AdvisId>7aafe81b-a377-40ac-86c0-0779713503d3</AdvisId>
            <AdvisTekst>Message received and accepted</AdvisTekst>
            <KildeId>my-it-system.dk</KildeId>
         </Advis>
      </SvarReaktion>
   </HovedoplysningerSvarREST>
</memos_response>
```

This indicates to Serviceplatformen, that all is well. The parameter *KildeId* is Danish for *SourceId*.

4.5. Implement Error response

In case you cannot accept the message, you reply with HTTP 500 and a Fejl (Error) element. Again remember to echo the two X-Transaktion headers.

memos_response example - Error

```
<memos_response>
   <HovedoplysningerSvarREST xmlns="http://kombit.dk/xml/schemas/kontekst/2017/01/01/">
      <SvarReaktion>
         <Fejl>
            <FejlId>message.file.encodingFormat.required</FejlId>
            <FejlTekst>Unable to extract and parse main document</FejlTekst>
            <KildeId>my-it-system.dk</KildeId>
         </Fejl>
      </SvarReaktion>
   </HovedoplysningerSvarREST>
</memos_response>
```

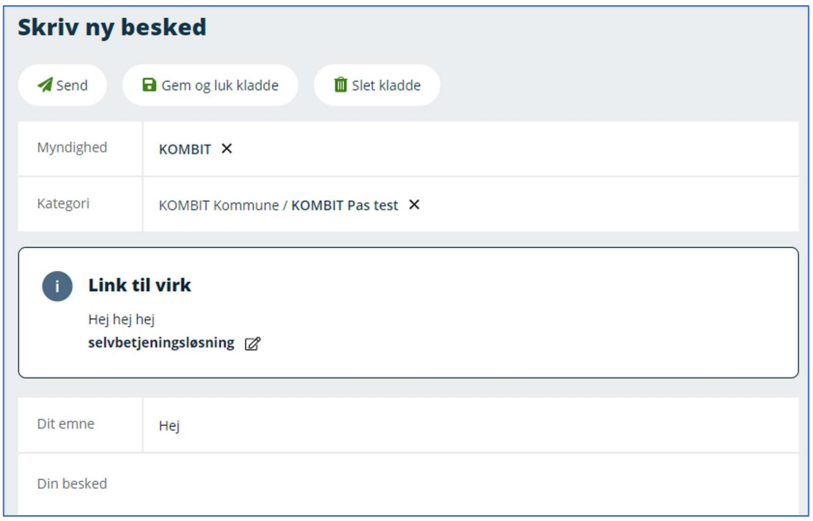
For values in <FejlId> see chapter *Error Codes* in guide Technical Integration NgDP.

5. TEST THE SETUP

You can test the setup by sending a message from a test-citizen mailbox, or by implementing and using your own SF1601 client. You simply specify the contact point you had created, that is routed to your IT-system by a distribution rule.

5.1. Send message from a citizen mailbox

From our test-citizens mailbox, we selected the contact point “KOMBIT Pas test” as Kategori (category).



When sent, this message is routed to our IT-system.

5.2. Send message from a SF1601 client

See SF1601 Programmer's Guide on how to implement a client that can send messages. In the messages we sent, that we want routed to our own system, we simply specify the contact point under Recipient:

MeMo example with recipient contact point

```
...
<memo:MessageHeader>
   ...
   <memo:Recipient>
      <memo:recipientID>19435075</memo:recipientID>
      <memo:idType>CVR</memo:idType>
      <memo:label>KOMBIT</memo:label>
      <memo:ContactPoint>
         <memo:contactPointID>4302bc81-93d2-4490-b425-ddf9374606a4</memo:contactPointID>
         <memo:label>KOMBIT Pas test</memo:label>
      </memo:ContactPoint>
   </memo:Recipient>
</memo:MessageHeader>
...
```

5.3. Check reveived messages

Here is an example of a message we received in our testing, using a SF1601 client and a message where our contact point is specified under recipient.

Request HTTP-headers

```
POST /memos/ HTTP/1.1
Content-Length: 1206
Connection: keep-alive
Host: kdi-ctt-test.kombit.dk
X-Transaktionstid: 2022-02-09 00:09:21.695
X-Transaktionsid: 620306310e53f9f7291f11b42ba3dcd0
Content-Type: application/xml
...
```

Request body

```
<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>2f4f4bd6-9965-44c1-b091-7be0cfaf4b3a</memo:messageUUID>
      <memo:label>Besked</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>19435075</memo:recipientID>
         <memo:idType>CVR</memo:idType>
         <memo:label>KOMBIT</memo:label>
         <memo:ContactPoint>
            <memo:contactPointID>4302bc81-93d2-4490-b425-ddf9374606a4</memo:contactPointID>
            <memo:label>KOMBIT Pas test</memo:label>
         </memo:ContactPoint>
      </memo:Recipient>
   </memo:MessageHeader>
   <memo:MessageBody>
      <memo:createdDateTime>2021-11-26T15:06:24Z</memo:createdDateTime>
      <memo:MainDocument>
         <memo:File>
            <memo:encodingFormat>text/plain</memo:encodingFormat>
            <memo:filename>Besked.txt</memo:filename>
            <memo:language>da</memo:language>
   <memo:content>RW4gbnllIGJlc2tlZA0KTWVkIGxpbmplc2tpZnQNClZpcmtlciBkZXQ/</memo:content>
         </memo:File>
      </memo:MainDocument>
   </memo:MessageBody>
</memo:Message>
```

Here are details of the certificate, that was authenticated, when Serviceplatformen called our endpoint:

```
SSL_CLIENT_FINGERPRINT: 6e136d12f13284253489c371064566a3a9b903a6
SSL_CLIENT_V_REMAIN: 538
SSL_CLIENT_V_END: May 18 19:42:41 2023 GMT
SSL_CLIENT_SERIAL: 5DC28650
SSL_PROTOCOL: TLSv1.2
SSL_CLIENT_I_DN: CN=TRUST2408 OCES CA IV,O=TRUST2408,C=DK
SSL_CLIENT_S_DN: CN=kombit-serviceplatformen-callback-test
(funktionscertifikat)+serialNumber=CVR:19435075-FID:98159228,O=KOMBIT A/S //
CVR:19435075,C=DK
SSL_CLIENT_VERIFY: SUCCESS
```

6. APPENDIX

6.1. Message log in Digital Post at Digitaliseringsstyrelsen

When troubleshooting you can ask the municipality in question to check the Hændelseslog (eventslog) in the administrative portal at Digitaliseringsstyrelsen:



They can specify various filters, which is not illustrated here. Here is an example list of events:



6.2. How to create test citizens

Creating a citizen in the test-environment for Digital Post at Digitaliseringsstyrelsen is useful for testing purposes. Getting access to the Digital Post test-environment is described in Vejledning til Administrativ Adgang (in Danish) and also in the document *Technical Integration NgDP*, that you find on the same page. Once you have access to the portal https://testportal.test.digitalpost.dk/login, you can create a citizen. This portal is in Danish.

Note that it is not permitted to use production data in Digital Post at Digitaliseringsstyrelsen. Therefore when creating a test-citizen make sure it's not a real social security number (CPR number). It is posible to use test CPR numbers that doesn't comply with modulus 11 check ie. 000000-11111.

This is the form for creating a test-citizen:



We can then access the citizens mailbox here:

https://post.demo.borger.dk/

VERSION HISTORY

Version	Dato	Ændringer
1.0	16-02-2022	First version
1.1	29-06-2026	The concept of NGDP is replaced by Digital Post at Digitaliseringsstyrelsen.