Technical Integration - Front page

Exported on 18/02/2026

Table of Contents

1 TI Versioning 7
1.1 Versioning 7
2 Introduction and overview 17
2.1 Background 17
2.2 Target group 17
2.3 List of terms 17
2.4 Minimum requirements for authorities 21
2.5 More information 21
2.6 Overview of DP 21
3 Querying and searching 35
3.1 Domain names 35
3.2 Outgoing IP 37
3.3 Querying and searching resources 37
4 Contact registry services - TI 44
4.1 Querying Contacts 44
4.2 Searching 44
4.3 Using the body of the GET request 45
4.4 Querying using POST with body for non-caching retrievals 47
4.5 How to query closed contacts 48
4.6 Contact registration lists exposed by Digital Post 57
5 System registry services - TI 60
5.1 Organisations 60
5.2 Systems 61
5.3 Contact structure 63
5.4 Querying in System registry APIs 65
5.5 Fetching hidden contact points and contact groups 70
5.6 Find contact-groups structured below other contact-groups 71
5.7 Updating items in system registry 72
5.8 Organisation contact-group/point subscription 77

5.9 Endpoint exposed in the system-subscription-store 77
5.10 Notifications 78
6 Mailbox services - TI 79
6.1 Mailboxes 79
6.2 Accesses 81
6.3 Messages 83
6.4 Documents 87
6.5 Files 89
6.6 Folders 93
6.7 System fetches 94
6.8 Querying for Messages 96
6.9 Common use case examples 97
6.10 Update newly created draft 100
6.11 Update a draft with different recipient (Citizen) 103
6.12 Update a draft with different masked recipient 105
6.13 Automatic data scrubbing of the filename 109
6.14 Uploading invalid html to a file 123
6.15 Trying to send a message without content in the main document 124
6.16 Replying to an unreplyable message (reply = false) 125
7 Event log services - TI 127
7.1 Querying the event-log 127
7.2 Event Log Index Events 131
8 Push notification integration - TI 140
8.1 Some general info about push notifications via DP 140
8.2 Registering as a push notification tenant (aka “I want to send push notifications”) 140
8.3 Creating FCM service account + private key 141
8.4 Interaction with “push notifications” 145
8.5 Subscription to push notification 149
8.6 Visual guide for push notification flows 151
9 Identity registry services - TI 152

9.1 Identities 152
9.2 Direct privileges 153
9.3 Grantees 155
9.4 Identity groups 155
9.5 Privilege types 156
9.6 Querying Identities, Direct privileges, Privilege Type 157
9.7 Identities 157
9.8 Direct Privileges 158
9.9 Identity-group privileges 159
9.10 Privilege Types 162
9.11 Direct privilege 163
9.12 Creating Direct Privilege 164
9.13 Privilege group 165
9.14 Querying the Privilege Group 165
9.15 Querying in Kibana 169
10 Distribution - TI 173
10.1 Distribution Services 173
10.2 MeMo 173
10.3 Distribution REST services 174
10.4 Inbound services 174
10.5 Outbound services 176
10.6 Sending single MeMo 178
10.7 Sending large amount of messages 180
10.8 Receipts 185
10.9 Outbound receipt REST push request 193
10.10 At-least once principle 194
10.11 Flow for resending messages 195
10.12 Flow for resending business receipts - REST Push protocol 197
10.13 MeMo validation 197
10.14 HTML whitelist for document validation 198
11 Access request registry 212

11.1 Access request registry - introduction....214
11.2 Purpose of the registry....214
11.3 Privilege requests....214
11.4 Delegation requests....215
11.5 Appointed delegation requests....215
11.6 Connection agreement requests....215
11.7 Terms approval requests....215
11.8 User administrator's statement of truth privilege requests....215
11.9 Lost user administrator privilege requests....215
11.10 Special privilege requests....215
11.11 Delegated support admin privilege request....216
11.12 Third party intermediary request....216
11.13 Concepts....216
11.14 Access request registry - common use case examples....217
11.15 User administrator delegates privilege to employee identified with an e-mail address....217
11.16 Usage of generic identity id in access requests....219
11.17 1. Add documentation element....221
11.18 2. Upload byte content to documentation....224
11.19 1. Citizen A submits request....226
11.20 2. Citizen B queries to see incoming requests....228
11.21 3. Citizen B approves request....229
12 Sender-/Receiver Systems....233
12.1 Rate-limiting....233
12.2 Patterns for integration to Digital Post....234
12.3 Sender system....234
12.4 Recipient system....235
13 Encoding formats, Environments and Error codes....240
13.1 Encoding format whitelist for files of documents....240
13.2 Access to environments....242
13.3 SMS character encoding....244
13.4 Error codes....244

14 Java/.Net Core, Security perspective, MeMo-lib and Test 262
14.1 Reference Systems for Java and .Net Core 262
14.2 Security Perspective 269
14.3 MeMo-lib 270
15 Access to Test environments 271
15.1 Access to the administration portals on the test environment 271
15.2 Access to Test Portal on the test environment 273
15.3 Access to Administrative Access on the test environment 275
16 Troubleshooting, SFTP server, SDLC, OpenID Connect, Connect 277
16.1 Troubleshooting 277
16.2 SFTP server 295
16.3 OIO OpenID Connect to Digital Post 300
16.4 Connect to Digital Post: Test and Prod 306

1 TI Versioning

1.1 Versioning

Version:	1.51
Status:	Final
Author:	Netcompany

Document history

Version	Date	Comments
1.33	28.01.2022	**Updated the following** • Event Log Index Events • Digital Post Receipt domain model • Sending MeMo message over REST PUSH • REST protocol examples

1.34	22.02.2022	**Included the following**
		• At least once-principle
		  • Specifies that Digital Post operates with a principle which ensures that all messages are delivered at least once
		• Requirements for messages and attachments
		  • Specifies the allowed sizes of attachments and messages
		• Fetching registration status list
		  • Specifies that the interface defaults to JSON. Use the accept header if XML is preferred.
		• Bulk receipt list (massekvitteringsliste)
		  • Specifies that only XML return values are supported for this endpoint
		Updated the following
		• List of terms
		  • Specifies that DP uses UUID version 4
		• Rate-limiting
		  • Minor update
		• Distribution REST services
		  • Specifies that the interface defaults to JSON. Use the accept header if XML is preferred.
		• DP receipt domain model
		  • Specifies that COMPLETED means that the message has been validated
		• Sending DP/DP2 messages via REST
		  • Specifies that the interface defaults to JSON. Use the accept header if XML is preferred.
		• Event Log Index Events
		  • New event with ID 81 added to show events in the event log in the Rights Portal regarding access request acitivities.
		• Fetching registration status for contact
		  • accept header added to the documentation
		• Querying identities, Direct privileges, Privilege Type
		  • Specifies that you can also revisit the OpenAPI specification for more background
		• Back-end validation and error codes in distribution
		  • Specifies that TransmissionID must be unique to be valid

1.35	17.06.2022	**Updated the following**
		• Querying and searching resources
		  • Specifies the use of quotations
		• Querying the event-log
		  • Description of Querying for more than 10k using “next”
		• Flow for resending messages
		  • Description of retrying to default recipient system and system deactivation

1.36	14.09.2022	**Added the following**
		• Certificates
		  • Information about how Digital Post currently supports OCES2 and OCES3
		• Push Notification Integration: Creating FCM service account + private key
		  • Guide to how view clients can offer push notifications
		Updated the following
		• 14: Access to the Test Portal on the test environment
		  • Content has been deleted as DIGST has created an external guide for the Test Portal of Digital Post
		Removed/rearranged the following sections
		• “15: Test Portal: Create a test-identity and login to Administrative Access” has been changed to “15: How to login to Administrative Access with test-identities” as the previous content can be seen in above-mentioned guide

1.37	21.11.2022	**Updated the following** • 3.3 Size requirements • Added word to clarify paragraph

1.38	30.05.2023	**Updated the following**
		• Domain names
		  • Updated External IP for the domain to access the REST API by sender and receiver systems using mutual SSL
		• Querying and searching resources
		• Querying Contacts
		  • Details about the query parameter `isBulkLookup`
		• Encoding format whitelist for files of documents
		  • Added additional documents
		• Exporting certificates for upload to Administrative Access
		  • Sending messages using the SMTP protocol will be faced out 16 August 2023
		• Event log services
		  • Link to a full comprehensive list of all events stored in the event log
		• Creating and working with drafts
		  • Usage of generic identity id in access requests
		• Front-end validation and error codes in the Viewclient
		• Back-end validation and error codes in distribution
		• Software Development Life Cycle (SDLC) for the API
		• HTML whitelist for document validation
		  • Updated elements and attributes

1.39	20.06.2023	**Updated the following**
		• Contact Status
		  • Added the column statusDate
		• Messages
		  • Added the columns Dele messages and Fetch unread status
		• Afsendelse (resource)
		  • Added the columns VedhaeftningSamlingKvantitet, MeddelelseKvitteringsTypeNavn, MeddelelseKvitteringPostkasseIdentifikator, MeddelelseFESDmetadata, MeddelelseTidsfristDato and MeddelelseTidsfristTeks
		• HTML whitelist for document validation
		  • Updated the Styles table
		• Overall strategy for the Software Development Life Cycle (SDLC) for the API
		Included the following
		• Comment: PublicRegistrationStatus
		• Contact Subscription
		  • The purpose of the contact subscription is to persist subscriptions for sendersystems, on either a set of specific citizens, organisations or all changes fitting a category.
		• Subscribing to changes in the system registry
		  • The purpose of the system subscription is to persist subscriptions for sendersystems, on either a set of specific organisations, or all changes fitting a category. When a change is made to a contact point is made or a new contactpoint is created, the subscription components finds all matching subscriptions, and notifiesthe sendersystem on all the matching subscriptions.
		• Error codes
		  • System Fetch services
		Removed/rearranged the following sections
		• Structure of the API

1.40	24.07.2023	• **Introduction**
		  • Added additional terms to the terms list
		  • Expanded the description of the usage of mutual SSL and API token
		  • Updated the list of trusted certificate authorities
		• **Access to Test environment**
		  • Updated onboarding flow to start with log-in to Testportalen
		• **Encoding formats**
		  • Updated E-mails from Digital Post
		• **Event Log**
		  • Updated Event Log Index Events with more elements and examples of events.
		• **OIO OpenID Connect to Digital Post**
		  • Added details regarding logout in Digital Post and NemLog-in
		  • Added details for MitID AppSwitch and eID-Gateway support
		  • Updated client enrollment section with more detailed descriptions and any relevant restrictions where applicable
		• **Querying and searching:**
		  • Added limitation of 100 characters for each search field and added examples

1.41	18.10.2023	• **Contact registry services**
		  • Full Access added as a role that can use contact registry
		  • PublicRegistrationstatus updated with status = CLOSED
		• **Distribution**
		  • Added section “Should I send single messages or bulks?”
		• **Encoding formats, Environments and Error codes**
		  • Added new error codes related to full access
		• **Event log services**
		  • Updated documentation on querying the event log.
		  • A default time interval on 3 weeks has been added
		  • Wildcard searches are no longer supported
		  • contentResposible added to example of element in event log
		• **Mailbox services**
		  • Added full access as a new role that can use mailbox services
		• **Troubleshooting, SFTP server, SDLC, OpenID Connect**
		  • Updated fingerprints for accepted issuer CAs
		  • Updated link to OCES certificates
		  • Updated section “OIO OpenID Connect to Digital Post”

1.42	14.11.2023	• **Access to Test environments** • Updated steps for creating test users with MitID Simulator.

1.43	22.01.2024	• **Access to Test environments**
		  • Updated steps for creating test users with MitID Simulator on devtest4
		• **Contact Registry Service**
		  • Added “How to query closed companies”
		  • Added “Contact registration lists exposed by Digital Post”
		• **Encoding formats, Environments and Error Codes**
		  • Added error codes for SFTP
		• **Sender-/Receiver systems**
		  • Updated Rate-limiting

1.44	23-05-2024	• **SMS character encoding**
		  • Added “SMS character encoding” under chapter 13
		• **Removed sections in Distribution - TI**
		  • Legacy services: Sending DP/DP2 messages via REST
		  • Legacy services: Sending DP/DP2 messages via SFTP
		  • Legacy services: Sending DP/DP2 via SMTP
		  • Legacy error codes
		  • Handling of reply destination in Digital Post contact structure when sending DP/DP2
		  • Distribution SMTP services
		  • SMTP MeMo example
		  • Fetching registration status for contact
		  • Fetching registration status list
		  • Fetching contact registration status part list
		  • Bulk receipt list (massekvitteringsliste)
		  • Contact-registry in Record format
		• **Mailbox services**
		  • “Full access” has been renamed to “Læse- og skriveadgang (Full power of attorney)”
		• **System registry service**
		  • Removed legacy SMTP and Receipt Format
		• **Sender-/Receiver Systems**
		  • Rate-limiting for view clients has been added

1.45	09-08-2024	• **Distribution**
		  • Added reference to MeMo version 1.2 and concurrent support of versions
		  • Updated Inbound services to include new endpoint for bulk transmissions
		  • Updated HTML whitelist for document validation
		  • Added Global scope to Lenient and added new attributes. Removed “lang” attribute from Span element, since it is Global now.
		• **Event log services**
		  • Added reference to concurrent support of MeMo versions
		• **Mailbox services**
		  • Added reference to MeMo version 1.2 and concurrent support of versions
		• **System registry services**
		  • Model for Organisation has been expanded with `privacyPolicyUrl`
		  • Model for System has been expanded with `memoTransitionDateTime`
		• **Minor updates**
		  • Business receipt error codes
		  • Contact Registry data model
		  • Flow for resending messages

1.46	25-10-2024	• **Error codes**
		  • Added new error codes for new memo-lib validation
		• **Bulk MeMo Sftp service**
		  • Updated to reflect new receipt removal
		• **Mapping between error codes and receipt status**
		  • Added table for error codes and corresponding receipt status
		• **Invalid filename characters**
		  • Added the list of invalid characters in filenames, introduced alongside MeMo v. 1.2.
		  • Added data scrubbing of the filename, when sent through a mailbox.
		• **Child privileges**
		  • Added how to create a child privilege using identity-groups REST endpoint
		• **Distribution-validator error codes**
		  • Removed some errors that cannot occur.
		• **MeMo**
		  • Modified list of invalid character to exclude space (which is now a valid character)

1.47	25-03-2025	• **Mailbox model**
		  • Adding originalRepresentative to forwardData on mailbox
		• **OpenAPI Swagger UI links added to services**
		  • Removed model diagrams. Use OpenAPI schemas.
		• **Roadmap removal**
		• **Extended vocabulary**

1.48	03-06-2025	• **Identity registry services**
		  • Added new endpoint for identity lookup without storing sensitive data (CPR)
		• **Contact registry services**
		  • Added new endpoint for contact lookup without storing sensitive data (CPR)
		• **Mailbox services**
		  • Added new endpoint for fetching unread messages without storing sensitive data (CPR)
		• **Querying Identities, Direct Privileges, Privilege Type**
		  • Added “Searching for identities using POST with body”
		• **Querying Contacts**
		  • Added “Querying using POST with body for non-caching retrievals”
		• **Reference Systems for Java and .Net Core**
		  • Removed dead link reference
		• **Forward message to trusted recipient and authority**
		  • Updated example to be up to date with current data model.
		• **Forward message to e-mail address**
		  • Updated example to be up to date with current data model.
		• **Minor updates**
		  • Removed links to internal pages.

1.49	08-09-2025	• **Removed sections in Reference Systems for Java and .Net Core**
		  • system-smtp / Smtp
		• **Access request registry - introduction**
		  • For special privilege requests, partial mailblock on curator privileges has been added
		• **Identity registry services**
		  • Extended query for fetching privilege types with query parameters
		• **Querying Identities, Direct Privileges, Privilege Type**
		  • Added response example of when fetching internal privilege types
		• **Back-end validation and error codes in distribution**
		  • Updated validation and error codes
		• **Front-end validation and error codes in the Viewclient**
		  • Updated error codes

1.50	10-11-2025	• **Access request registry - introduction**
		  • Added a new access request type “Lost user administrator privilege requests” and description of its principles
		• **Encoding format whitelist for files of documents**
		  • Extended document types
		• **Identity registry services**
		  • Updated description for query privilege type
		• **ReplyData mail threads**
		  • Updated examples for reply data
		• **At-least once principle**
		  • Added descriptions of events flows that will be resent under infrastructure instability
		• **MeMo**
		  • Updated MeMo library description
		• **Querying for Messages**
		  • Updated example
		• **Mailbox services**
		  • Added service for “Fetch mailbox overview for a single mailbox”
		• **Querying Identities, Direct Privileges, Identity-group privileges, and Privilege Type**
		  • Updated example under “Identity-group privileges”
		• **Subscribing to changes in the system registry**
		  • Deleted service for “Delete subscription”

1.51	16-02-2026	• **Mailbox services**
		  • Updated required roles
		• **Querying Identities, Direct Privileges, Identity-group privileges, and Privilege Type**
		  • Updated privilege type description
		• **Push notification integrations**
		  • Updated example under “Subscription to push notification”
		• **Querying the event-log**
		  • Updated description & examples for “Using wildcard”

References

Reference	Title	Author	Version
All User Manuals https://digst.dk/it-loesninger/digital-post/vejledninger-og-begivenheder/		Netcompany and DIGST	Latest

User Manual for Administration Portal https://digitaliser.dk/digital-post/vejledninger/administrativ-adgang	'Administrative Access User Guide'	Netcompany and DIGST	Latest
User Manual for Rights Management Portal https://digitaliser.dk/digital-post/vejledninger/rettighedsportalen	'Rights Management Portal User Guide'	Netcompany and DIGST	Latest

2 Introduction and overview

2.1 Background

This document contains the technical system documentation in order for authorities to integrate their systems to Digital Post. Such systems are referred to as sender- and recipient systems.

Setting up sender- and receiver systems and accessing log files etc. is handled in the administration portal 'Administrativ Adgang'. Follow the user guide (See 'Reference') for more information on how to navigate in the portal and how to set up your systems once they are ready.

Disclaimer: This is not a complete guide in building or adjusting sender and receiver systems. It does however provide all the technical information of the Digital Post solution needed to build or adjust your own sender and receiver systems for the new infrastructure.

"DP" refer to "Digital Post".

2.2 Target group

This guide is primarily intended for developers to understand the design, technical implementation and integrations. As well as for architects and business analysts to ensure that technical integrations are documented and that access has been established. Readers are assumed to have knowledge of technical terms (e.g. web certificates, IDP), common protocols (e.g. HTTP, SFTP) and industry standards (e.g. RESTful services, TLS, JSON).

In addition, this guide describe the high-level processes involved in adjusting sender- and receiver systems, which may be relevant for project planning.

Note: Only integrations from the Digital Post perspective are documented and not the detailed implementation.

For the process of integration, activating, and establishing as well as deactivating sender- and receiver systems in the administration portal 'Administrativ Adgang', see 'References'.

2.2.1 For authorities

This guide is intended for authorities with the need to adjust, adapt or build their receiver- and sender systems integrating to DP.

2.2.2 For businesses

This guide is intended for businesses with their own recipient - and sender systems. For businesses that do not have a dedicated receiver system it is still possible to read digital post at http://virk.dk .

2.3 List of terms

Term / English	Danish	Description
Authority	Myndighed	An authority is official institutions, all municipalities and regions and some independent institutions funded by public finance. An authority is identified by the CVR-number. Every authority can have multiple sender- and recipient systems.

Business	Virksomhed	A business is a private corporation able to receive and send Digital Post. A business can only send Digital Post to an authority. Most businesses use their own mailbox at http://virk.dk but some have their own recipient (and sender) systems.
Organisation	Organisation	A collective term used for both businesses and authorities.
Contact structure	Kontaktstruktur	The contact structure is the authority's structure of how the authority receives and distributes mails for the receiver systems from DP. The contact structure may also be used to provide a link to a self-service solution (selvbetjeningsløsning).
Contact point	Kontaktpunkt	The contact point is a unique entity in the authority's contact structure used to indicate where mails need to be distributed to. Externally to the end users (citizens and other businesses), these are presented as "subjects".
Contact registry	Kontaktregister	The contact registry supports the look up of information about registration of citizens and businesses to Digital Post and NemSMS. Registration status is used for distribution of messages and for authorities to look if a citizen is exempted or not. See section 4 Contact registry service for more info.
Classification	Klassifikation	Classification is a structured way of listing terms and subjects in relation to common public administration tasks, they can be linked to contact points to further earmark incoming mail towards the correct receiver within an authority.
Classification code	Klassifikationskode	Classification code is a unique identification number for a common public administration task.
Classification name	Klassifikationsnavn	Classification name is the name of a common public administration task.
Classification type	Klassifikationstype	Classification types indicate which public administration classification system is in use e.g. FORM or KLE.

CVR	CVR nummer	CVR-nr. is a business' identification number in Denmark. The CVR number is fetched from the Contact Registry.
CPR	CPR nummer	The CPR number is a unique and universal identifier of a physical person registeret in the central person registry “Det Centrale Personregister”
Location	Lokationskode	The location code is used to indicate what physical device or location the contact point is connected to e.g. SOR or GLN
DP	Digital Post	It is used interchangeably as a reference to the Digital Post solution
Report Link	Rapportlink	ReportLinks are links connected to a contact point with a description. When a contact point containing a ReportLink is used, the user will be presented with the link before proceeding to send mail via the contact point. E.g. this can be used to inform the user that a self-service solution should be used instead of sending digital post. This is only available for public authorities.
Search word	Søgeord	Search words are used as ‘tags’ to further describe with alternative words the subject or related subjects of a contact point, making the contact point easier to find through search.
Sender system	Afsendersystem	Sender system is a system which is used for sending mails via DP.
Recipient system	Modtagersystem	Recipient system is a system which is able to receive mails via Digital Post. Can also be called receiver system.
System	System	A common term for sender- and recipient systems connected to Digital Post.
System registry	Systemregister	The system registry is responsible for storing information about the authorities and businesses that are sending messages via the solution. It also contains the Contact Structure as well as all registered sender and recipient systems.

Authority registry	Myndigheds register	The authority registry is part of the system registry.
Target group	Målgruppe	The target group is exposed to a contact point.
Technical person	Teknisk person	In the administration portal every organization must provide contact information on a technical resource e.g. a system administrator.
UUID	UUID	UUID is short for "Universal Unique Identifier" and is a 128-bit number used to identify digital objects in Digital Post. Version 4 is used to generate UUIDs.
View client	Visningsklient	A view client displays the messages and make them accessible for end users without sender-/ recipient systems.
Public View client	Offentlig visningsklient	A view client provided by the Danish public sector. Borger.dk for citizens and virk.dk for businesses.
Virk	Virk Virk.dk	The public view client for business.
Borger.dk	Borger DK Borger.dk BDK	The public view client for citizens.
Notification	Advisering	A notification is information which is triggered when information is received, In the context of Digital Post often related to a message received in a users mailbox where a notification can be send thought mobile push-notification, email or SMS.
Legal notification	Forkyndelse	Legal notifications issued by a danish court.
Mandatory	Obligatorisk	Mandatory digital post will be delivered even if the recipient is exempted.
Bulk transaction	Masseforsendelse	A transaction of sending multiple single messages at once.
Exemption	Fritagelse	Legally exempt from receiving digital post.

2.4 Minimum requirements for authorities

The following are the minimum requirements in terms of sender- and receiver systems in order to integration with Digital Post.

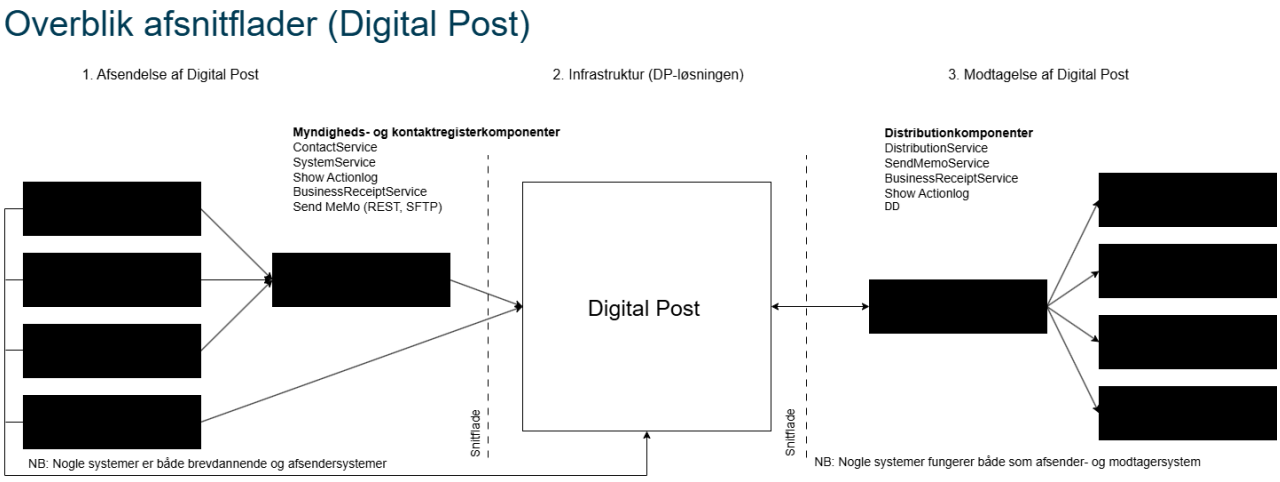
2.4.1 Recipient systems

- Businesses may have one receiver system.
- Authorities must have at least one receiver systems, and can have multiple. (Five is recommended).
    - At least one receiver system for messages sent to the authority.
        - At least one of the receiver systems has to be able to receive 99,5 MB sized messages
- All messages related to tasks as an authority has to be collected and authorities have to store the messages in its own systems.
- Has the same requirements in terms of file size as the Digital Post solution.

2.5 More information

For a comprehensive overview of guides in relation to Digital Post solution please go to: https://digst.dk/it-loesninger/digital-post/vejledninger/

2.6 Overview of DP



2.6.1 Mutual SSL authentication using API key

This section aims to give a concise overview of how sender- and recipient-systems are expected to interact with the REST API of Digital Post. This section does not go into details on how mutual SSL/TLS works (see What is mTLS? | Mutual TLS | Cloudflare) or how OCES certificates are obtained (see Certifikater i MitID Erhverv - MitID Erhverv).

Background

For sender- and recipient-systems to communicate securely with Digital Post they are expected to utilize mutual SSL where both the initiator and the responder are presenting a certificate. That way, both parties can create a secure channel of communication and verify the identity of each other.

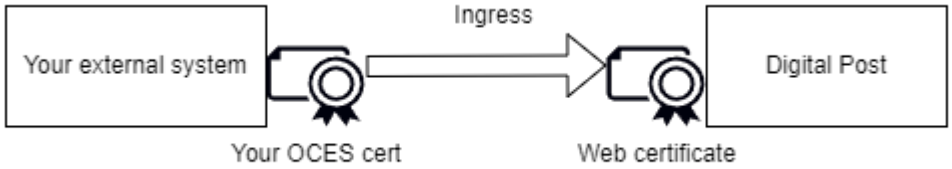
The reasoning behind why Digital Post requires sender- and recipient-systems to authenticate using mutual SSL is because it is an industry standard for two parties to establish a secure connection and exchange sensitive

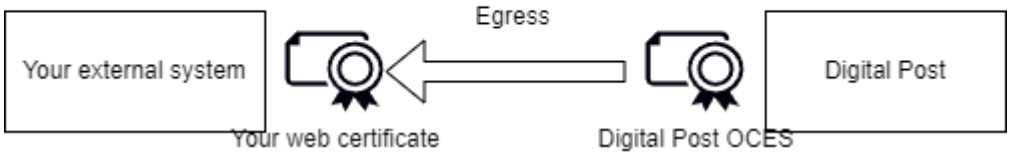
information where both parties can verify the identity of the other party. *Additionally*, Digital Post requires an API key used **together** with a mutual SSL connection to avoid binding the client certificates to specific systems.

Certificates used in the HTTPS traffic

When performing API calls not just any certificate can be used. Digital Post has strict requirements for the format and use of certificates. There is always one web certificate and one OCES certificate in motion as part of the mutual SSL. The pattern is the same, however, it depends on who is initiating the communication. This can be described as follows, where ingress is a sender- or recipient system connecting to Digital Post and egress where Digital Post connects to a external system

- **Ingress**: Sender systems present their **OCES certificate** and Digital Post presents a valid **web certificate**
- **Egress**: Digital Post presents its valid **OCES certificate**, while recipient system presents a valid **web certificate.**





Certificates used to authenticate

As shown in the above figure, the mutual SSL authentication uses both an OCES certificate and a web certificate. The OCES certificate is a certificate as defined by the OCES specification and are issued through MitID Erhverv. In the specification there are a couple of different certificate types. Digital Post supports the following types

- Organisationscertifikater (organisation certificates)
- Systemcertifikater (System certificates)

During ingress calls you, as the client, is expected to present a certificate which is issued to your organisation. This certificate is not issued in the context of Digital Post but is something which you need your MitID administration to issue for you. You can find more information about how to obtain an OCES certificate here.

During ingress Digital Post, as the server, exposes a web certificate. Web certificates are certificates issued by generally trusted certificate authorities, for further details see section *Web certificates policy*.

Similarly during egress, where Digital Post is the client and calling you as the server, Digital Post will present its OCES certificate and you are expected to expose your server with a valid web certificate. To increase the security during egress communication you can pin the certificate which Digital Post is using to ensure that you are only accepting calls from Digital Post. You can find the public certificates here: Digitaliser.dk - OCES certifikater til Digital Post løsningen.

As the above diagram shows, during ingress communication you as the client is expected to present a valid OCES certificate.

Before calling the API

Before you can begin calling the Digital Post REST API you must create a sender- or recipient-system in Administrative Access. Here you are presented with a range of options, that are required for the system to be operational. Refer to the dedicated guide for Administrative Access for further details.

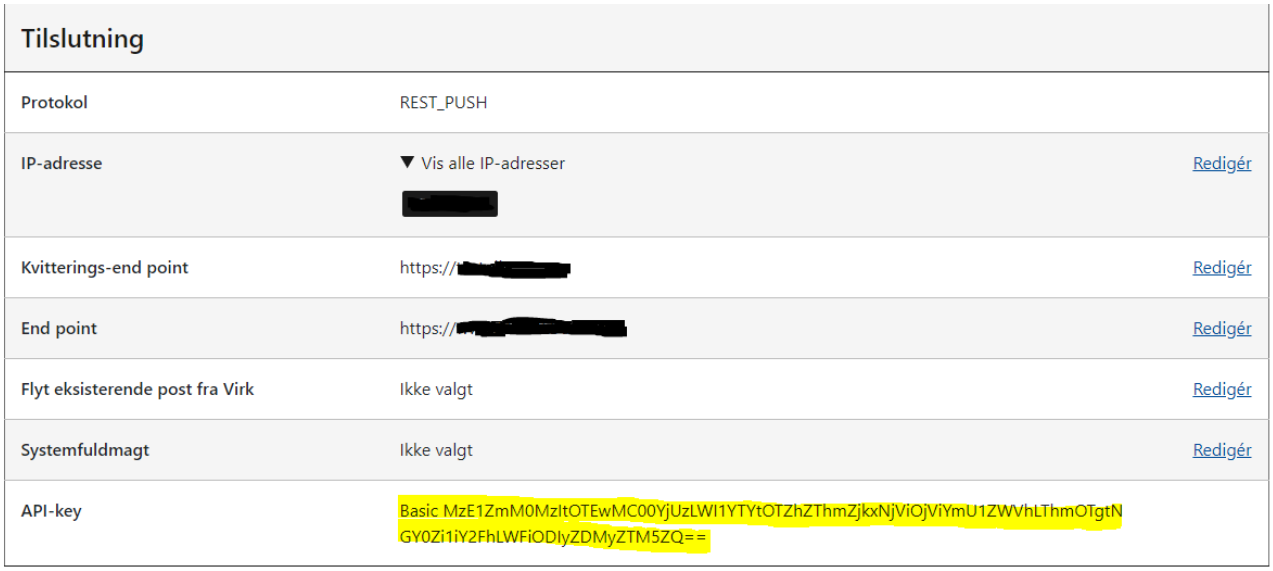
Once you have created an active system and chosen one of the available rest service protocols you are presented with an API key. You must save this value. Additionally, you can also choose to delegate the systems to a vendor, more on this later.

API key

The API key is an attribute on REST systems in the system-registry. It is used by Digital Post to identify which system that is calling the API, not as a way to establish a secure channel of communication (this is why mSSL is still required). The API key is a random string of text that is unique to a given system. When the system is created through Administrative Access an API key is always automatically assigned, note that the system still must be valid and active to be used.

The value that is presented in Administrative Access, and the one that the user should pay attention to is the encoded value. Where the system will perform encoding of the systems ID together with the value of the token. This is to ensure the least amount of work from the integrator since they will then be able to send the API key directly as presented in the authorization header and not having to worry about encoding.

From the Administrative Access, the API key can be retrieved from system details as the screenshot below



To avoid confusion only the encoded value is present in the UI, however taking a look at the raw response from the server, we can see the encoded key together with the raw value

```
{
...
 "apiToken" : {
    "id" : "ef1ec3b0-0136-47e1-8ef2-f396d2db6e58",
    "version" : 0,
    "value" : "5bbe5eea-8f98-4f4f-bcaa-ab822d32e39e",
    "authenticationToken" : "Basic
MzE1ZmM0MzItOTEwMC00YjUzLWI1YTYtOTZhZThmZjkxNjViOjViYmU1ZWVhLThmOTgtNGY0Zi1iY2FhLWFiO
DIyZDMyZTM5ZQ==",
    "lastUpdated" : "2021-05-10T13:45:46.155Z"
```

```
  }
  ...
}
```

In addition to the `value` property, which is the persisted value of the API key, the details also return a property named `authenticationToken` . The encoded value (shown in the UI and in `authenticationToken`) is encoded using base64 encoding following the The 'Basic' HTTP Authentication Scheme, and the “username” is the id of the system and the “password” is the `value` of the key.

Mutual SSL using API key

To call Digital Post it is not necessary to upload or otherwise register your OCES certificate to your REST System. However, it is still *mandatory* for the client to present the OCES certificate in the mutual SSL requests. In addition, the API key `authenticationToken` value must **always** be added in the `Authorization` header.

Why does Digital Post require both mSSL and API key to be present? These reasons are, but not limited to, the following

- Organisations with sender- and recipient-systems are not required to upload their OCES certificates in Digital Post, but there is still a need for unambiguously identifying the system in order to obtain authorization for the REST API
- No need for maintaining certificates if they expire or are compromised in Digital Post
- Delegation of systems can be done using vendors' certificate
- OCES certificates are not required to be unique across the entire Digital Post solution and can thus be reused for all systems an organisation has

The reason why we gain these benefits while still having both a secure and reliable solution where certificates are no longer required to be unique is that when initiating a call, Digital Post does the following checks when authenticating the sender-/recipient-system. If any of the following checks fail the call is rejected

- A valid Organisationscertifikat or Systemcertifikat is used to establish a secure mutual SSL connection
- An API key is presented in the authorization header
- The API key can be used to find the system
- The system is valid and active
- **The system belongs to the same organisation as is present in the callers OCES certificate**. This is done by matching the CVR numbers. If you need a vendor to perform an actions on your behalf, read the section *Delegated Sender- and Receiversystems*.
- The callers IP matches the IP or is within the IP range as configured in Administrative Access.

Given this setup, you cannot register your certificate preemptively in Digital Post since we do not have any direct binding to a specific certificate but rather uses the CVR number present in the OCES certificate when your system is authenticating via mutual SSL when calling Digital Post. This means that you do not upload your certificate in Administrative Access, rather you authenticate using the certificate.

⚠ You are required to use both mutual SSL and API token when calling Digital Posts REST API

In this setup, Digital Post does not know your certificate, as it matches using CVR you are free to use as many and as few certificates as you desire. And you are not required eforehand to register any specific certificate in Digital Post. This method is secure for both sender- and recipient-systems and Digital Post since the caller can verify the identity of Digital Post, and Digital Post can with confidence extract the CVR from the callers OCES certificate since we verify the certificate chain.

Examples

See the example of using the current implementation of mutual SSL using cURL on the TEST environment. In this example, the certificate `my_oces_certificate.cer` is not known to Digital Post because it was not uploaded. However, as mentioned above, uploading a certificate to the system after creation is no longer required since Digital Post can authenticate using the OCES certificate presented by the caller, and find the system based on the API key provided in the Authorization header

```
curl -v \
--http1.1 \
--key "my_oces_certificate.pkcs8" \
--key-type pem \
--cert-type pem \
--cert "my_oces_certificate.cer" \
-H "Authorization: Basic
ZTU0NWIzMzktODU0Mi00YTMwLWFlNzUtYzY3ZTRkMmE3Yjk5OjE1NjMyYThkLTQwN2MtNGMzYS1iN2IxLWFlY
mFhNTE0ZmNhYg==" \
"https://api.test.digitalpost.dk/apis/v1/contacts/"
```

Delegated Sender- and Recipient-systems

If you as an organisation or authority want to outsource or delegate a specific sender- or recipient-system you need to apply an additional settings during the setup of the system in Administrative Access. You need to specify the “Giv din systemleverandør fuldmagt (Valgfrit)” field with the CVR number of the *vendor* that you want to delegate to. Once this is done, Digital Post has noted this special relation, and the vendor can then use **their own OCES certificates** to act on behalf of your organization, and Digital Post then ensure that only they are allowed to use that system.

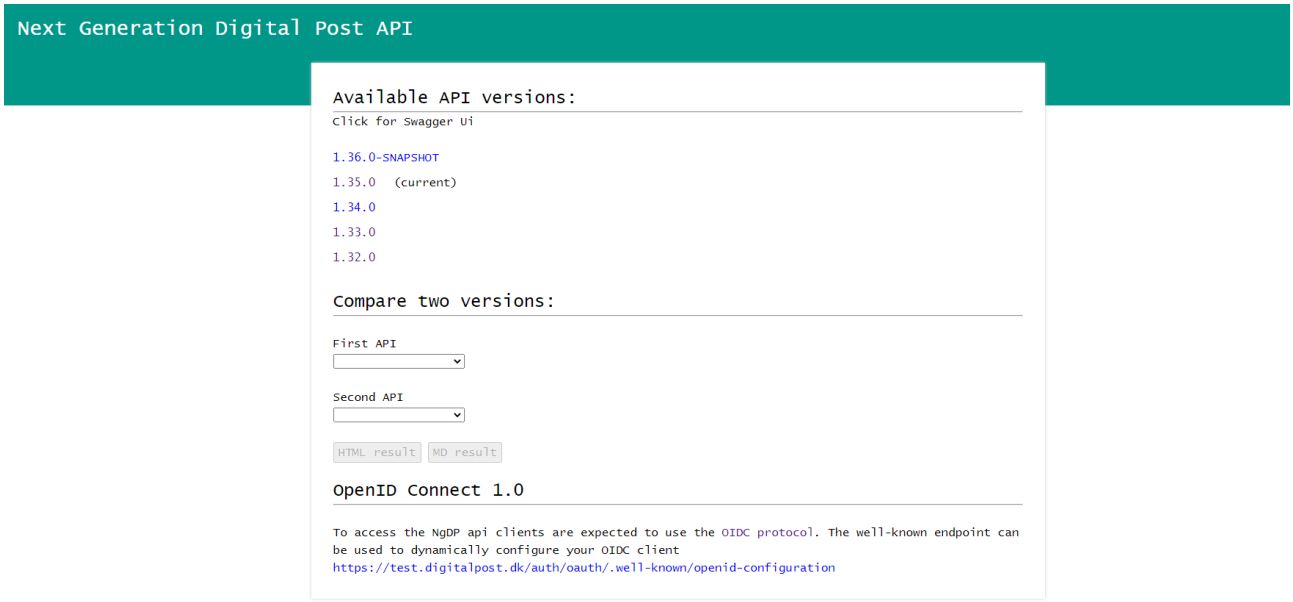
⚠ Delegated sender- and recipient-systems are required to use both mutual SSL and API key.

2.6.2 Open API description

Digital Post defines all of the externally exposed services in an OpenAPI specification. The intention is to make the API easily digestable, and to provide a programing language-agnostic description of the rest API. Eliminating any guess work when integrating with the rest API and to avoid having to rely directly on this documentation as the source of truth. Additionally this also provides versioning for the API so that you are able to see the differences between different versions of the API.

Digital Post also provides a tool that developers can use to compare different versions of the API, which should help when upgrading between the different versions. Generally Digital Post provides a new OpenAPI description after each release.

The OpenAPI definition can be found following the URL; https://test.digitalpost.dk/api/ or https://api.test.digitalpost.dk/api/ .



Note that both URLs point to the same destination and provides the same information.

When following above link you will be presented with a web-page show casing a handful of the lastest description as well as a marking for the currently used page. Additionally a snapshot version containing the upcoming changes are also present, however this version is subject to change. By clicking on one of the versions you will be directed to a "SwaggerUI" where you are able to navigate the all the services in a user friendly way. Note that this page, contains all externally served services, which include both services for sendersystems, receiversystems, administrative access and view clients for both citizens and companies.

> ⚠ The Swagger UI does currently not support mutual SSL

All consumers of the Digital Post API are encouraged to utilize the OpenAPI description to reduce errors and help when updating between version.

2.6.3 REST Implementation

This section gives a general introduction to how Digital Post implements the HTTP based API. The general convention is to follow the REST concepts. The general concept is that data points are treated as resources, where you can fetch lists, a single resource, or creating a resource.

Endpoints

A resource is always referenced using the plural name in the path with lower casing. Also if the name of a resource consist of multiple words (such as contact point) they are separated with a `-` (dash).

HTTP Method	Endpoint	Term	Description
GET	`/resources/{id}`	Fetch	Fetches a single resource by its ID
GET	`/resources/`	Query	Queries all resources in a list, using query parameters to filter the results
POST	`/resources/`	Create	Creates a new resource
PUT	`/resources/{id}`	Update	Updates a resource
PATCH	`/resources/{id}`	Patch update	Updates/modifies a resource, providing only a partial resource
DELETE	`/resources/{id}`	Delete	Deletes a resource

This section describes a root level resource, such as Mailbox or Contact. However the same pattern also applies to sub-resources, such as Contact points, which are placed under an Organisation `/organisations/{id}/contact-points/`

Precondition headers

To ensure that updates from different sources does not accidentally overwrite each other Digital Post implements optimistic locking thought a set of the precondition headers as specified by https://datatracker.ietf.org/doc/html/rfc7232 .

The optimistic locking is implemented by all resources having a version, that is then expected by the caller to provide as they modify a resource. Then if another client modified the resource in the meantime, the API will then reject the update since the caller is no longer working on the most recent and would potentially overwrite the changes. By applying optimistic locking, and it means that Digital Post can avoid having pessimistic locking, where the caller would have to explicitly lock a resource while editing to prevent others from editing. And required the call to both lock and unlock a resource while editing.

- ETag; is present as a response header when fetching a resource
- If-Match header must always be set by the calling when updating or deleting a resource. The value is expected to be the exact version of the modified resource. Multiple values or wildcard is not accepted by the api.

Fetch

```
GET /resources/78eb6ffa-cdc4-412e-bcb3-e1e0ef552ee9
```

Fetch a single resource by its ID. All resources always contain both a version and an ID at the root level of the resource. Additionally sub-resources (or objects) often also contain their own ID and version. The result conforms to the following structure in JSON:

```
{
  "id": "78eb6ffa-cdc4-412e-bcb3-e1e0ef552ee9",
  "version": 12
  ...
}
```

The main purpose of having a version on resources is to have optimistic locking, where the caller provides the version that it modified. Then if another client modified the resource in the meantime, the API will then reject the update since the caller is no longer working on the most recent and would potentially overwrite the changes. By applying optimistic locking, and it means that Digital Post can avoid having pessimistic locking, where the caller would have to explicitly lock a resource while editing to prevent others from editing. And required the call to both lock and unlock a resource while editing.

Return codes

The HTTP response status codes indicate whether a specific request has been completed:

HTTP Status code	HTTP Status	Description
200	OK	The resource is returned in the body
401	UNAUTHORIZED	User is not authorized
403	FORBIDDEN	User didn't have access to resource defined in path parameters
404	NOT FOUND	The resource from ID was not found

ETag

The entity version is also always returned in the HTTP response header in the ETag:

```
ETag: "12"
```

Query

`GET /resources/`

Returns a search result with a list of resources matching the query. The caller can then limit the results by providing one or more filters thought query parameters. When no parameter is provided, all resources are returned that the caller have access to. For instance, if a contact is calling the `/contacts/` with no parameters only the citizens own contact is returned where as when a sender system is doing the same query all contacts are returned.

The matching resource results are always wrapped in a search result which enables pagination. The result therefore conforms to the following structure in JSON, where `<resource>` is the name of the specific resource.

```
{
    "currentPage": 0,
    "totalPages": 1,
    "elementsOnPage": 6,
    "totalElements": 6,
    "<resources>": [
      {
        "id": "78eb6ffa-cdc4-412e-bcb3-e1e0ef552ee9",
        "version": 12
        ...
      },
      ...
      {
        "id": "c291d912-f932-46ca-b7f7-4415cabcf4a3",
        "version": 1
        ...
      }
      ...
    ]
}
```

Return codes

The HTTP response status codes indicate whether a specific request has been completed:

HTTP Status code	HTTP Status	Description
200	OK	The call was successful
400	BAD REQUEST	The query arguments do not match the expected formal definition
401	UNAUTHORIZED	User is not authorized
403	FORBIDDEN	User didn't have access to resource defined in path parameters

HTTP Status code	HTTP Status	Description
404	NOT FOUND	Some resource defined in path parameters did not exist

Specifying a query

`GET /resources/?type=CITIZEN&postCode=2100`

Will list all citizens living in postcode 2100. Type and postCode will be bound to the `QueryCommand` and resolved and formulated as a query for the implementing datastore (Elasticsearch or JPA) in the Persistent Service (with appropriate query infrastructure components) to resolve the result.

If nothing is found the service returns an empty result:

```
{
    "currentPage": 0,
    "totalPages": 1,
    "elementsOnPage": 0,
    "totalElements": 0,
    "<resources>": []
}
```

When querying the contacts it is possible to use the body of the GET request, see more under section 4.1 Querying Contacts.

Create

`POST /resources/`

When creating a new resource the client is expected to provide the entire resource, however id and version is can either be omitted or provided with null values, since the service will always create these on the fly. If the creation is successful the service will provide the created resource as part of the response body. Where computed field such as id, version, transaction id and creating timestamps will be populated.

```
{
  "id": null,
  "version": null,
  ...
}
```

The service will return the created entity upon successful creation of the entity:

```
{
  "id": "c291d912-f932-46ca-b7f7-4415cabcf4a3",
  "version": 0,
  ...
```

```
}
```

Return codes

The HTTP response status codes indicate whether a specific request has been completed:

HTTP Status code	HTTP Status	Description
201	CREATED	Upon successful creation of the resource
400	BAD REQUEST	The JSON does not conform to the expected structure or the content of the JSON does not validate according to validation rules
401	UNAUTHORIZED	User is not authorized
403	FORBIDDEN	User didn't have access to resource defined in path parameters

Update

`PUT /resources/c291d912-f932-46ca-b7f7-4415cabcf4a3`

When updating a resource the calling is expected to initially fetch the specific resource, when do the necessary edits and then return the entire updated resource. Since the caller request body is treated as the new state of the resource, clients should be careful when updating, since omitting parts of the resource will be treated as a removal. Since resources are optimistically locked, the caller must also provide the If-Match precondition header with the version which was modified.

Updates an existing resource identified by ID. Input is the HTTP entity following the following JSON structure:

```
{
  "id": "c291d912-f932-46ca-b7f7-4415cabcf4a3",
  "version": 0,
  ...
}
```

The service will return the updated entity upon successful update of the entity:

```
{
  "id": "c291d912-f932-46ca-b7f7-4415cabcf4a3",
  "version": 1,
  ...
}
```

Return codes

The HTTP response status codes indicate whether a specific request has been completed:

HTTP Status code	HTTP Status	Description
200	OK	Upon successful update of the resource
400	BAD REQUEST	The JSON does not conform to the expected structure or the content of the JSON does not validate according to validation rules
401	UNAUTHORIZED	User is not authorized
403	FORBIDDEN	User did not have access to resource defined in path parameters
409	CONFLICT	The resource version mismatched. The resource was updated in mid-air

Precondition headers

The exact resource version must be communicated to the service using the `If-Match` Request Header, for version verification.

```
If-Match: 0
```

Note that the api does accept wildcards or multiple values.

The new resource version then returned upon a successfully update in the response deader `ETag` :

```
ETag: "1"
```

Patch Update

`PATCH /resources/c291d912-f932-46ca-b7f7-4415cabcf4a3`

Some resource are also updateable using the PATCH http method. When patch updating, unlike normal update, the client is only expected to provide the part of the resource which they have modified. This can especially be useful when editing large resource that contains lots of information. Like the normal update the caller is also expected to provide the precondition If-Match header.

Updates an existing resource identified by ID. Input is specific fields of the HTTP entity following the below shown JSON structure:

```
{
  "id": "c291d912-f932-46ca-b7f7-4415cabcf4a3",
  "version": 0,
  ...
}
```

The service will modify the entity on the exposed fields only, leaving the rest unmodified. The service will return the modified entity upon successful update:

```
{
  "id": "c291d912-f932-46ca-b7f7-4415cabcf4a3",
  "version": 1,
  ...
}
```

Return codes

The HTTP response status codes indicate whether a specific request has been completed:

HTTP Status code	HTTP Status	Description
200	OK	Upon successful update of the resource
400	BAD REQUEST	The JSON does not conform to the expected structure or the content of the JSON does not validate according to validation rules
401	UNAUTHORIZED	User is not authorized
403	FORBIDDEN	User didn't have access to resource defined in path parameters
409	CONFLICT	The resource version mismatched. The resource was updated in mid-air

Precondition headers

The exact resource version must be communicated to the service using the `If-Match` Request Header, for version verification.

```
If-Match: 0
```

Note that the api does accept wildcards or multiple values.

The new resource version then returned upon a successfully update in the response deader `ETag`:

```
ETag: "1"
```

Delete

`DELETE /resources/c291d912-f932-46ca-b7f7-4415cabcf4a3`

Deletes an existing resource identified by ID. Similar to update, the caller is also expected to provide the precondition If-Match header.

Return codes

The HTTP response status codes indicate whether a specific request has been completed:

HTTP Status code	HTTP Status	Description
204	NO CONTENT	Upon successful delete
401	UNAUTHORIZED	User is not authorized
403	FORBIDDEN	User didn't have access to resource defined in path parameters
409	CONFLICT	The resource version mismatched. The resource was updated in mid-air

3 Querying and searching

3.1 Domain names

Additional information on how to access environments can be found in the section: “Access to environments”.

This section gives an overview of the different domains that are in Digital Post as well as a description.

Recommended firewall usage, is to dynamically detect IP of the DNS name, so changes to IP are picked up automatically.

External IP are only here for information purpose, and is not expected to be a source of truth.

3.1.1 QA environment

Hostname	External IP	Description
https://admin-qa.test.digitalpost.dk	80.198.95.55	The domain for the Administrative Access portal that is used to configure contact-structure, system-setup, event-log, exemption etc.
https://testportal-qa.test.digitalpost.dk	80.198.95.55	The portal to manage and create test data in Digital Post
https://app-qa.test.digitalpost.dk	80.198.95.55	The portal for managing push-notification and revoking app access
https://api-qa.test.digitalpost.dk/	80.198.95.13	The domain to access the REST API of Digital Post by sender and receiver systems using mutual SSL
https://gateway-qa.test.digitalpost.dk/	80.198.95.55	The domain to access the REST API by view clients (such as http://borger.dk)
http://sftp-qa.test.digitalpost.dk	188.64.157.65	The domain that exposes the SFTP server for sender and receiver systems

3.1.2 Test environment

Hostname	External IP	Description
https://admin.test.digitalpost.dk	80.198.95.45	The domain for the Administrative Access portal that is used to configure contact-structure, system-setup, event-log, exemption etc.
https://testportal.test.digitalpost.dk	80.198.95.45	The portal to manage and create test data in Digital Post
https://app.test.digitalpost.dk	80.198.95.45	The portal for managing push-notification and revoking app access
https://api.test.digitalpost.dk/	80.198.95.10	The domain to access the REST API of Digital Post by sender and receiver systems using mutual SSL
https://test.digitalpost.dk/	80.198.95.45	The domain to access the REST API by view clients (such as borger.dk)
http://sftp.test.digitalpost.dk	80.198.95.42	The domain that exposes the SFTP server for sender and receiver systems

3.1.3 Production

This table contains the domain and IPs for the production environment. Note that production is not yet open for business.

Hostname	External API	Description
https://admin.digitalpost.dk	80.198.95.24	The domain for the Administrative Access portal that is used to configure contact-structure, system-setup, event-log, exemption etc.
https://app.digitalpost.dk	80.198.95.24	The portal for managing push-notification and revoking app access
https://api.digitalpost.dk	80.198.95.23	The domain that exposes the REST API of Digital Post by sender and receiver systems using mutual SSL

Hostname	External API	Description
https://gateway.digitalpost.dk	80.198.95.24	The domain to access the REST API by view clients (such as http://borger.dk)

3.2 Outgoing IP

The IP address used for outgoing traffic from Digital Post til external parties is `80.198.95.62` for both QA, TEST and Production.

3.3 Querying and searching resources

The components, that expose an endpoint for querying, all offer the querying/searching functionality that is described here.

3.3.1 Eventually consistent

Resources received when querying (GET) the URL

```
/resource-name(in plural)/
```

are only eventually consistent with resources from unsafe methods (Safe (HTTP Methods) - MDN Web Docs). This means that after a `POST`, `PUT`, `PATCH`, or `DELETE` the state change might not be immediately available for this type of querying. `GET` using ID for fetching a single resource will always be consistent.

3.3.2 SearchResult

The result of searching will be a resource specific implementation of a SearchResult containing a list of resources and paging information.

Example JSON

```
{
  "next": "",
  "currentPage": 0,
  "totalPages": 1,
  "elementsOnPage": 1,
  "totalElements": 1,
  "resource-name(in plural)": []
}
```

3.3.3 Paging

It is possible to paginate the results using the following parameters:

page	Zero-based page index (0..n)
size	The size of the page to be returned. If not specified the default page size is 100. The upper limit is 10.000, as set by ElasticSearch
next	An encoded string representing the last element on the previous page. Necessary when needing to page through more than 10.000 resources. The number of resources returned on the next page can be adjusted by changing the value of size.

Example

Using 'page' and 'size'

```
/resource-name(in plural)/?page=1&size=10
```

Using 'next'

In order to utilize next when paginating results it is necessary to a regular query against the endpoint which will then provide a next value that can be used for paging through all resources mathing the request. That is, first we query normally

```
/resource-name(in plural)/?page=0&size=100
```

which returns

```
{
    "currentPage": 0,
    "next": "nextValue1"
    "totalPages": 100,
    "elementsOnPage": 100,
    "totalElements": 10000,
    "resource-name(in plural)": [
        {Resource1},
        {Resource2},
        ...
        {Resource100}
    ]
}
```

then to get the next page we query using the next parameter instead of specifying page and size as follows

```
/resource-name(in plural)/?next=nextValue1
```

which returns

```
{
    "currentPage": 0,
```

```
    "next": "nextValue2"
    "totalPages": 1000,
    "elementsOnPage": 100,
    "totalElements": 100000,
    "resource-name(in plural)": [
        {Resource101},
        {Resource102},
        ...
        {Resource200}
    ]
}
```

To increase the readability of the above examples, the parameter ‘next' has been simplified.
A real life example of the value for the parameter 'next’ could look like:

`WyAiMjAyMS0xMi0wOFQwOTo1ODo1Ni45NDkiLCAiRkRKVVg3TE5mbzFFa1ZjUFJsbmFCU3FrTUl6UHFDVXAiIF0=`

The ‘next’ value can also be combined with other search parameters

```
/resource-name(in plural)/?field1=value1&next=nextValue
```

Important notes when using ‘next’

When paging through a registry using the next parameter it is important to sort the results in ascending order such that the resources with the newest values, e.g. lastUpdated, are sorted at the very end of the results instead of at the beginning. If you sort values in a non-deterministic way and without using a tiebreaking field in case of duplicate values the returned search result may be inconsistent and resources may be skipped or missed.

Please note that when utilizing the next parameter, it is important to be aware that the value of the currentPage will always be 0, regardless of how many times the next parameter is used.

3.3.4 Sorting

Sorting can be done using the following parameter

sortFields	One or more fields to sort by. Each field can be appended the desired sort order by separating them using a colon. If a sort order is not provided it will default to ‘asc’ for that particular field. The sort takes place in the order given. If no sortFields are provided it defaults to a resource specific default sort field and order. The fields can be a nested field in the resource-structure using dot (.) between the elements. For example *field1.subField1*.

Example

```
/resource-name(in plural)/?sortFields=field1.subField1:asc,field2:desc
```

This example sorts on subField1 ascending first and where the subField1 of the resources is equal, it goes on to sort on field2 descending.

3.3.5 Filtering

Filtering can be used to only fetch certain fields like for instance ID and version.

fields	One or more name of the fields you wish to fetch. The fields can be a nested field in the resource structure using dot (.) between the elements. For example *field1.subField1*

Example

```
/resource-name(in plural)/?fields=id,version,field1.subField1
```

returns only those fields of each resource:

```
{
    "currentPage": 0,
    "totalPages": 2,
    "elementsOnPage": 100,
    "totalElements": 130,
    "resource-name(in plural)": [
        {
            "id": "5e3cd399-84df-4fc3-856f-a3bb1fb2a21f",
            "version": 3,
            "field1": {
                "subFields1": "value"
            }
        },
        {
            "id": "0a6ea322-7ca0-49f9-93c2-fddce3de2dae",
            "version": 11,
            "field1": {
                "subFields1": "value"
            }
        },
        ...
    ]
}
```

3.3.6 Searching

The individual resources may override specific search functionality for a field. In which case it will be documented under that service and have own parameter description in Open Api. Besides the specific search options these are available

any	One or more search terms to search across all fields in the resource

Field name from resource	Specify one or more fields from resource each with one or more search terms. A field can be a nested field in the resource-structure using dot (.) between the elements. For example *field1.subField1*. If more values are given to a single specific field, the matches will be where either match (OR). If more search fields are given, the matches will be where all match (AND) - see below override option.

Operator on search term	As mentioned above, given more than one search field will by default make sure all different fields MUST match. I.e. an AND is placed between them. This functionality can be overridden by adding the an operator prefixed to the search value. The following is supported:
	• |
	  • OR
	• &
	  • AND
	• !
	  • NOT
	Examples:
	`/?param=|alfa,|bravo,!charlie` - equivalent to `/?param=|alfa&param=|bravo&param=!charlie`
	This operator is currently only available on these generics search parameters, and thus not on the fixed filters the individual endpoint offers.

Quoted	A search term may be put in quotes, in which case the it will search for the entire term without doing any attempts to match only part of the term, fuzziness or the like. “ and ' are both allowed. The search is still done case insensitive.

⚠ **Note that there is a limitation of 100 characters on the length of each value used for the search fields**

Examples

Using a parameter that exceeds 100 characters :

```
/resource-name(in plural)/?
message=alpha%20beta%20kappa%20zeta%20eta%20phi%20epsilon%20delta...add%20more%20until%20you%20reach%20100&subject=MESSAGE
```

we get the following response

```
{
    "code": "digital.post.error",
    "message": "IllegalArgumentException: message length of (number of characters) exceeds limit of 100",
```

```
    "fieldErrors": []
}
```

Using multiple parameters that are below 100 characters :

```
/resource-name(in plural)/?message=value1,value&subject=MESSAGE
```

Using 'any' field:

```
/resource-name(in plural)/?any=Netcompany&any=Digitaliseringsstyrelsen
/resource-name(in plural)/?any=Netcompany,Digitaliseringsstyrelsen
```

searches for 'Netcompany' OR 'Digitaliseringsstyrelsen' across all fields in a resource.

With specific fields:

```
/resource-name(in plural)/?recipient.recipientId=14814833,43720082
```

returns resources where recipientId equals 14814833 OR recipientId equals 43720082.

```
/resource-name(in plural)/?
recipient.recipientId=14814833&label=Fra%20Digitaliseringsstyrelsen
```

returns resources where recipientId equals 14814833 AND label equals 'Fra Digitaliseringsstyrelsen'.

Operator:

```
/resource-name(in plural)/?recipient.recipientId=|14814833&label=|
Fra%20Digitaliseringsstyrelsen
```

returns resources where recipientId equals 14814833 OR label equals 'Fra Digitaliseringsstyrelsen'

Using wildcard:

```
/resource-name(in plural)/?field1.subField2=Flytterod*
```

return matches where subfield2 starts with 'Flytterod'.

Possible wildcards:

*	Match zero or more characters, including an empty one
?	Matches any single character

Wildcard search is only possible when specific field is provided - not using the 'any' search field.

Using quotations:

```
/resource-name(in plural)/?field1="Københavns Kommune - Borger Service"
```

will only return with a full match and thus not return other matches such as “Københavns Kommune - Affald“.

4 Contact registry services - TI

Unable to render include or excerpt-include. Could not retrieve page.

4.1 Querying Contacts

- Searching
    - Examples
        - Searching with CPR
        - With searchField lastUpdated
- Using the body of the GET request
- Querying using POST with body for non-caching retrievals
- How to query closed contacts

For description of common search functionality in Digital Post, please revisit the section **Querying and searching resources**.

Querying contacts is done using a GET request to the `/contacts/` endpoint.

The result is a `ContactSearchResult`, which looks like this in JSON format

```
{
  "currentPage": 0,
  "next": "string",
  "totalPages": 1,
  "elementsOnPage": 1,
  "totalElements": 1,
  "contacts": [...]
}
```

Fetching a specific contact can be done as a `GET` request to the `/contacts/{id}` endpoint. The result is the contact with the specified ID. The definition of a Contact can be found the OpenAPI specification, which also includes a short description of all fields and types used in the Contact registry. Please consult the OpenAPI description section for details.

4.2 Searching

Besides the general functionality described above, the Contact registry offers search using the following parameters:

Field	Description
lastUpdated	Get all contacts that have been updated between the provided timestamp and now. Note that the timestamp should be provided in UTC time following ISO8601. Note that unlike the other parameters, only **one** lastUpdated parameter can be searched for
createdDate	Get all contacts that have been created after the provided timestamp.

Field	Description
isBulkLookup	Indicator used to specify the search parameters in the request body instead of query parameters

4.2.1 Examples

Generally the format is

```
/contacts/?<parameter>=<value>
/contacts/?<parameter>=<value>,<value>,<value>
/contacts/?<parameter>=<value>&<parameter>=<value>&<parameter>=<value>
```

It's possible to mix-and-match different parameters. For lines 2 and 3 it's possible to add as many parameters/values as desired.

Searching with CPR

A common use case of the contact registry is searching for a Contact with a specific CPR number. This can be done using the following method

```
GET https://api.digitalpost.dk/apis/v1/contacts/?cprNumber=1234567890
```

Which will return a search result which contains all Contacts with that cprNumber. This will of course always be either 0 or 1 results (note that 1234567890 is a fictive cpr number). It is also possible to search after contacts using multiple CPR numbers

```
/contacts/?cprNumber=1234567890,0987654321
/contacts/?cprNumber=1234567890&cprNumber=0987654321
```

return matches where cprNumber equals 1234567890 or 0987654321

With searchField lastUpdated

```
/contacts/?lastUpdated=2020-04-27T11:01:43.652Z
```

return matches where lastUpdated is later than the 11:01:43 the 27th April 2020 (UTC timezone).

4.3 Using the body of the GET request

It is possible to query contacts while providing the parameters inside a request body. This functionality is provided to support filtering for a larger number of contacts in a single request. In order to provide a uniform API there is a limit to the number of combined `cvrNumber` and `cprNumber` which can be provided by the caller. The limit is currently 10.000 to match the number of search results provided, however this value can be changed to ensure performance in the API. It is currently only `cvrNumber` and `cprNumber` which are counted towards the request limit.

To query the contact registry using the request body the query parameter `isBulkLookup` must be used and set to `true` as shown

```
GET https://api.digitalpost.dk/apis/v1/contacts/?isBulkLookup=true
```

and the remaining query parameters are stored in the body

```
{
    "cvrNumber": ["43585118","43585045","30826191"]
}
```

Note that parameter value is an array of strings and not a comma-separated string as is the case when querying using query parameters.
Using the bulk query functionality supports the same filtering functionality as when using query parameters, however the structure is slightly different. For example, say you want to search for a list of CPR numbers, but only include those that are registered for Digital Post using the public registration status the request body should be

```
{
    "cprNumber": ["cpr1", "cpr2"],
    "mailboxSubscription": {
        "publicRegistrationStatus": ["REGISTERED"]
    }
}
```

That is, the structure of the request body follows the structure of the Contact model, but all values for fields from the Contact model should be given as an array of string values. It is also possible to page through the results by specifying page, size, and next in the request body

```
{
    "page": 3,
    "size": 10,
    "cprNumber": ["cpr1", "cpr2", ..., "cpr1000"],
    "mailboxSubscription": {
        "publicRegistrationStatus": ["REGISTERED"]
    }
}
```

When the `isBulkLookup` query parameter is equal to anything other than `true` it will be considered `false` and the request body ignored. Furthermore, when performing a bulk query you cannot specify any other query parameters and if done a HTTP 400 bad request will be returned

```
-- request
GET https://api.digitalpost.dk/apis/v1/contacts/?
isBulkLookup=true&cprNumber=1234567890

-- response
{
    "code": "digital.post.error",
```

```
    "message": "ValidationException: Validation failed",
    "fieldErrors": [
        {
            "resource": "target",
            "field": "bulkLookup",
            "code": "invalid.bulk.search",
            "rejectedValue": true
        }
    ]
}
```

Setting the bulk parameter to true and not providing the body will also be regarded as a bad request.

4.4 Querying using POST with body for non-caching retrievals

Besides from the GET with body query it is possible to make a query with POST of either a single or several CPR or CVR numbers or ids. This method is supported in addition to the GET method for compatibility reasons with frontend functionality (i.e Axios methods).

The POST with body is used in cases where sensitive information such as CPR numbers shall not occur in the URL and thus possible to cache for specific browsers. A POST query with a using a body for a value is made in the following manner:

```
POST https://api.digitalpost.dk/apis/v1/contacts/?isBulkLookup=true
```

and with a body containing either a single value or several values:

```
{
    "cprNumber": "1911112024"
}
```

```
{
    "cprNumber": ["1912112024","2008112024"]
}
```

If the `isBulkLookup` is not provided it will result in a `”Method Not allowed”`. Furthermore as with the GET body it will result in the following if the `isBulkLookup` is set to anything but `true`:

```
{
    "timestamp": "2025-04-15T06:33:45.716+00:00",
    "status": 405,
    "error": "Method Not Allowed",
    "path": "/contacts/"
}
```

Unlike the GET using it in combination with parameters such as `isBulkLookup=true&cprNumber=1234567890?` doesn’t result in aHTTP 400 bad request.

Any request without a body will however result in:

```
<problem xmlns="urn:ietf:rfc:7807">
    <type>about:blank</type>
    <title>Bad Request</title>
    <status>400</status>
    <detail>Failed to read request</detail>
    <instance>/contacts/</instance>
</problem>
```

4.5 How to query closed contacts

When a company ceases to exist or a citizen is deceased then their contact is closed. This information is acquired from external integrations such as datafordeler or virk.
Contacts that are closed for more then 5 years (citizen) and for more then 10 (company) are deleted from the solution.

When querying as Authority Sender system the user can see both CLOSED and ACTIVE contacts when querying the `/contacts/` endpoint.

In order to filter based on the status the user should use the following query endpoints.

For getting only the **CLOSED** contacts

```
GET https://api.digitalpost.dk/apis/v1/contacts/?
mailboxSubscription.publicRegistrationStatus=CLOSED
```

This should return the result like this:

```
{
  "currentPage": 0,
  "next": "string",
  "totalPages": 1,
  "elementsOnPage": 1,
  "totalElements": 1,
  "contacts": [
           {
            "id": "d913ca7b-2904-469c-944a-66fe5eaa4ef9",
            "version": 3,
            "type": "COMPANY",
            "transactionId": "FnlWgA6rw7IxTZwA1prHTE4bPcqncLbh",
            "cvrNumber": "31418992",
            "mailboxSubscription": {
                "id": "435f6236-2fc5-4bde-8b97-26559292218d",
                "version": 0,
                "publicRegistrationStatus": "CLOSED"
            },
            "lastUpdated": "2023-12-11T10:40:04.872Z",
            "createdDate": "2021-09-02T19:47:05.181Z"
        },
  ]
}
```

This will return the results with `publicRegistrationStatus` set to CLOSED

Active contacts can both have `REGISTERED` or `EXEMPT` as `publicRegistrationStatus`. It is also possible to query only for `REGISTERED` or `EXEMPT` status of the mailbox subscription.

For getting all **ACTIVE** contacts:

```
GET https://api.digitalpost.dk/apis/v1/contacts/?
mailboxSubscription.publicRegistrationStatus=REGISTERED,EXEMPT
```

this should return response like this which consists of a list of contacts that have their status as active.


```
{
  "currentPage": 0,
  "next": "string",
  "totalPages": 1,
  "elementsOnPage": 1,
  "totalElements": 1,
  "contacts": [
          {
            "id": "0be4d7ef-336d-486c-85a5-a49f47c856df",
            "version": 0,
            "type": "COMPANY",
            "transactionId": "FnlVLh3oQPYjF8dbZBkx6IaOSDDqI13q",
            "cvrNumber": "44486164",
            "mailboxSubscription": {
                "id": "81bf8158-01ed-4d6e-b9cc-23c0378b410c",
                "version": 0,
                "publicRegistrationStatus": "EXEMPT",
                "startTime": "2023-12-11T10:30:03.664Z"
            },
            "lastUpdated": "2023-12-11T10:30:03.664Z",
            "createdDate": "2023-12-11T10:30:03.664Z",
            "eligibleForVoluntaryRegistration": false
        },
  ]
}
```


4.5.1 Subscribing to NemSMS

One of the common use cases for both citizens, sender-systems and citizen service employees is subscribing a `Contact` to NemSMS. To perform this action we first have to find the relevant contact, depending on the user this can be done multiple ways. To simplify this example we assume the caller is a sender-system in an authority, which have access to the entire dataset of contacts. Remember to always consult the OpenAPI definition to see endpoints, parameters and resource types.

4.5.2 Querying the contact

Let us assume that we want to find the contact for a citizen, since we as the sender-system know that his cpr number is `1111111234` we can go ahead and search for that `cprNumber`;

```
GET https://api.digitalpost.dk/apis/v1/contacts/?cprNumber=1111111234
```

Which gives us the following response;


```
{
    "currentPage": 0,
    "totalPages": 1,
    "elementsOnPage": 1,
    "totalElements": 1,
    "contacts": [
        {
            "id": "c69912c4-11d5-4c62-97e6-79fcb1d7b99d",
            "version": 9,
            "type": "CITIZEN",
            "transactionId": "F640l9yALrjfHfASwcvYC0o5TUP7hjNZ",
            "cprNumber": "1111111234",
            "mailboxSubscription": {
                "id": "e19bbf71-587a-46a5-8392-dfaf7fd50a6a",
                "version": 9,
                "publicRegistrationStatus": "EXEMPT",
                "exemptionStart": "2021-07-15",
                "startTime": "2021-05-21T07:01:48.370Z"
            },
            "lastUpdated": "2021-07-15T15:06:50.004Z",
            "status": {
                "id": "7abe9012-435c-457c-84aa-170ec83d275f",
                "version": 0,
                "type": "ACTIVE",
                "changedDate": "2021-05-21"
            },
            "eligibleForVoluntaryRegistration": false
        }
    ]
}
```


4.5.3 Fetching the contact

Since Digital Post only promises eventual consistency, the search index and the persistence store might be out of sync when we query. We therefore must also fetch the Contact before editing;

```
GET https://api.digitalpost.dk/apis/v1/contacts/c69912c4-11d5-4c62-97e6-79fcb1d7b99d
```

Which then gives the following result;

```
{
    "id": "c69912c4-11d5-4c62-97e6-79fcb1d7b99d",
    "version": 9,
    "type": "CITIZEN",
    "transactionId": "F640l9yALrjfHfASwcvYC0o5TUP7hjNZ",
    "cprNumber": "1111111234",
    "mailboxSubscription": {
        "id": "e19bbf71-587a-46a5-8392-dfaf7fd50a6a",
        "version": 9,
        "publicRegistrationStatus": "EXEMPT",
        "exemptionStart": "2021-07-15",
        "startTime": "2021-05-21T07:01:48.370Z"
    },
    "lastUpdated": "2021-07-15T15:06:50.004Z",
    "status": {
        "id": "7abe9012-435c-457c-84aa-170ec83d275f",
        "version": 0,
        "type": "ACTIVE",
        "changedDate": "2021-05-21"
    },
    "eligibleForVoluntaryRegistration": false
}
```


4.5.4 Updating the contact

Now that we have fetched the Contact we can go a head and subscribe him to NemSMS. We do that by adding the `nemSmsSubscription` to the JSON structure, along with the phone number that should be subscribed. For the sake of the example we use `20202020`;


```
{
    "id": "c69912c4-11d5-4c62-97e6-79fcb1d7b99d",
    "version": 9,
    "type": "CITIZEN",
    "transactionId": "F640l9yALrjfHfASwcvYC0o5TUP7hjNZ",
    "cprNumber": "1111111234",
    "mailboxSubscription": {
        "id": "e19bbf71-587a-46a5-8392-dfaf7fd50a6a",
        "version": 9,
        "publicRegistrationStatus": "EXEMPT",
        "exemptionStart": "2021-07-15",
        "startTime": "2021-05-21T07:01:48.370Z"
    },
    "nemSmsSubscription": {
      "mobileNumber": "20202020"
    },
    "lastUpdated": "2021-07-15T15:06:50.004Z",
    "status": {
        "id": "7abe9012-435c-457c-84aa-170ec83d275f",
```

```
        "version": 0,
        "type": "ACTIVE",
        "changedDate": "2021-05-21"
    },
    "eligibleForVoluntaryRegistration": false
}
```

We can then update the Contact with above context using this request;

```
PUT https://api.digitalpost.dk/apis/v1/contacts/c69912c4-11d5-4c62-97e6-79fcb1d7b99d
If-Match: 9
Contect-Type: application/json
```

Note that we must always provide the precondition header `If-Match` to ensure that no-one made any changes to the Contact while we where editing. Since the API compares the current state to our request and figures out what changed.

If the request completes successfully we get the updated Contact in the response body;

```
{
    "id": "c69912c4-11d5-4c62-97e6-79fcb1d7b99d",
    "version": 10,
    "type": "CITIZEN",
    "transactionId": "F640l9yALrjfHfASwcvYC0o5TUP7hjNZ",
    "cprNumber": "1111111234",
    "mailboxSubscription": {
        "id": "e19bbf71-587a-46a5-8392-dfaf7fd50a6a",
        "version": 9,
        "publicRegistrationStatus": "EXEMPT",
        "exemptionStart": "2021-07-15",
        "startTime": "2021-05-21T07:01:48.370Z"
    },
    "nemSmsSubscription": {
      "id": "21bc5b76-d914-4da0-8895-ab3d725183af",
      "version": 0,
      "verificationTime": null,
      "confirmedDateTime": null,
      "mobileNumber": "20202020"
    },
    "lastUpdated": "2021-07-20T10:00:50.004Z",
    "status": {
        "id": "7abe9012-435c-457c-84aa-170ec83d275f",
        "version": 0,
        "type": "ACTIVE",
        "changedDate": "2021-05-21"
    },
    "eligibleForVoluntaryRegistration": false
}
```

Note how the version of the Contact have now increased by one, and the `nemSmsSubscription` have been assigned an id and a version. However, before the Contact can be contacted using the NemSMS he first must verify

that he owns the number. We can see that currently both the `confirmedDateTime` and the `verificationTime` are null, meaning that he have yet to perform the verification.

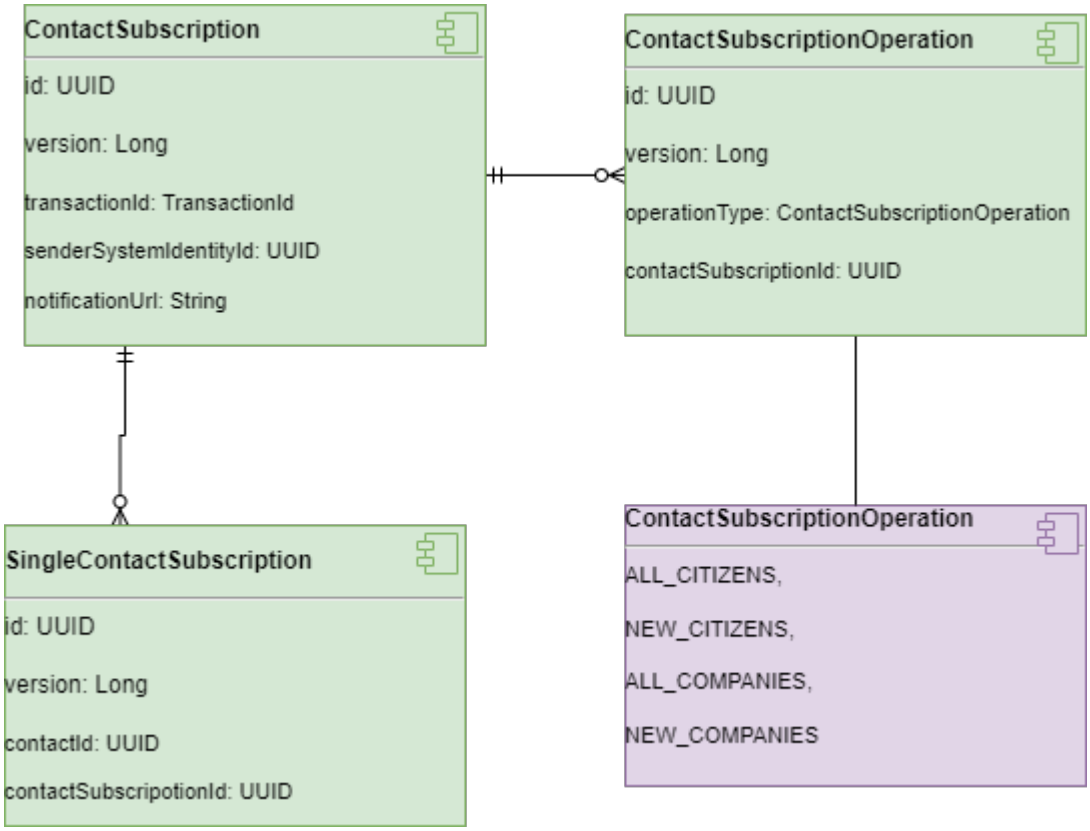
The contact registry will send an SMS to the number that was used, with either a pin code or a link. The pin code is used when it is the citizen or an employee in the company is doing the registration. Whereas the link will be used when assisted signup is done (so when a citizen service employee or a sender system is doing the signup on behalf of the citizen or company).

If the contact is already signed up for NemSMS but we want to change the number, we must simply replace the entire `nemSmsSubscription` similar to how the initial signup is done. And the verification process will again start over.

Additionally, if the Contact is already using the same number elsewhere in Digital Post, e.g. for notification when he receives new messages in the mailbox, he will not have to verify the number again. Instead, when we do the signup both the `verificationTime` and `confirmedDateTime` will be filled with the same data indicating that the number was already verified.

4.5.5 Contact Subscription

The purpose of the contact subscription is to persist subscriptions for sendersystems, on either a set of specific citizens, organisations or all changes fitting a category. When a change to the Contact is made the subscription components identifies all the matching criteria of a subscription, and notifies the sendersystem on all the matching subscriptions.



Contact Subscription

The `ContactSubscription` is the anchor of the subscription for any given sendersystem. It contains the notification endpoint and the reference to the sendersystem.

Column	Description	Required	Comment
id	The identifier of the subscription	Yes	
version	The current version of the subscription resource	Yes	Incremented on each update
transactionId	The identifier of the transaction of which the subscription was either created or last updated	Yes	Added or updated by the application on create or update
senderSystemId	The uuid of the sendersystem taken from the access token which for calling sendersystem	Yes	The access token replaces the mutual SSL connection during successful authentication in the API gateway
notificationUrl	The URL of which the sendersystem is notified on every relevant change	No	The URL must be https. The subscription is deemed inactive if the URL is not present

Contact Subscription Operation

The `ContactSubscriptionOperation` table contains all the operations which the sendersystem is subscribed to. The operations are bulk subscription to enable the sendersystem to get notified for all changes or any new entities. These can be used in tandem with explicit CPR and CVR subscriptions.

Column	Description	Required	Comment
id	The identifier of the subscription	Yes	
version	The current version of the subscription operation resource	Yes	Incremented on each update

Column	Description	Required	Comment

operationType	The type of bulk operations that are contained in the subscription	Yes	The values which the column can be;
			`ALL_CITIZENS`
			`NEW_CITIZENS`
			`ALL_COMPANIES`
			`NEW_COMPANIES`

contactSubscriptionId	The foreign key to Contact Subscription	Yes	

Single Contact Subscription

The `SingleContactSubscription` table contains all the contact ids which are explicitly defined by the sendersystem on its Contact Subscription.

Column	Description	Required	Comment
id	The identifier of the subscription	Yes	
version	The current version of the subscription resource	Yes	Incremented on each update
contactId	The contact id that the sendersystem has an explicit subscription for	Yes	
contactSubscriptionId	The foreign key to Contact Subscription	Yes	

4.5.6 Using the contact subscriptions

4.5.7 Contact Subscription

In order to maintain local copy of the contact-registry or have own registration lists updated with the newest contact data, the authorities can subscribe to changes to contacts registrations.

Authority sender system will as a part of the subscription specify an endpoint were they will get notified every time there is an update to a contact that are included in their subscription.

An authority can only have one subscription per sender system.

The subscription can either be made on individual contacts or by the usages of subscriptionOperators.

A subscription on individual contacts will contain contactIDs of all the specific contact the subscriber want to be notified about. The amount of contactIDs should be kept on an minimum, however there is no maximum limit. If the

authority sender system tries to provide a contactID that do not exist in Digital Post contact registry, the contact will be filtered out of the subscription (Nb. No error will be thrown).

SubscriptionOperators are some pre-defined filters for common use cases. There is a total of four operator:

- ALL_CITIZENS - The operator includes all updates to already existing citizens in the contact-registry.
- NEW_CITIZENS - The operator includes all citizens created in the contact-registry.
- ALL_COMPANIES - The operator includes all updates to already existing companies in the contact-registry.
- NEW_COMPANIES - The operator includes all citizens created in the contact-registry.

The operators can be used individually or several in the same subscription.

4.5.8 Endpoint exposed in the contact-subscription-store

Service	URL	Data returned	Usage	Required roles
Create subscription	POST /contacts/subscriptions/	created subscription	Creating a new subscription to get notified when organisations changed. Only one subscription per sender system.	AUTHORITY_SENDER_SYSTEM
List subscriptions	GET /contacts/subscriptions/	List of all subscriptions	Listing all subscriptions	AUTHORITY_SENDER_SYSTEM
Update subscription	PUT /contacts/subscriptions/{subscription-id}	Updated subscription	Updating a subscription	AUTHORITY_SENDER_SYSTEM
Delete subscription	DELETE /contacts/subscriptions/{subscription-id}		Deleting a subscription on the subscription ID with an “if-match” header matching the version	AUTHORITY_SENDER_SYSTEM

Example of a subscription creation with usage of individual contactIDs

```
{
   "singleContactSubscriptions":[
      "1e7ad5a8-f1a9-454a-a171-947b56737bd7",
      "fab79010-ab67-41e2-8936-d3b9c726cc84",
      "193e93d7-9d64-4826-85e4-9851e3b45664",
      "a9accf48-3d7a-4057-9c71-387df6d79000",
      "1100f22e-0efc-4b97-8948-6588112179f9",
      "1ce50ba0-f290-40fd-90b6-3c1e9717b3f7",
```

```
      "521f3356-d1c5-4812-bc69-59423fce52b7"
   ],
   "notificationUrl": "https://postman-echo.com/post"
}
```

Example of a subscription creation with usage of subscription operators:

```
{
  "subscriptionOperations": [
    "ALL_CITIZENS"
  ],
  "notificationUrl": "https://postman-echo.com/post"
}
```

4.5.9 Notifications

A push notification will be sent to the specified endpoint every time a contact is updated or created, and which matches the criteria from the subscription. The notification contain the id and the version of the contact.

Example of Notification:

```
{
  "id": "521f3356-d1c5-4812-bc69-59423fce52b7"
  "version": "1"
}
```

4.6 Contact registration lists exposed by Digital Post

Digital Post expose a file based lists of contacts registrations to authority sender system via SFTP.

The list of registered contacts contains information about whether a contact is registered for Digital Post and/or NemSms in the Digital Post solution. Only registered Contacts will be present on the list. So that a contact who is exempt from receiving Digital Post and does not have a mobile number registered and verified for NemSms, will not be a part of the lists.

A contact can be present up to a maximum of two time on a list, as each contact will have a specific row for registrations to Digital Post and a row for registrations to NemSms.

On the single SFTP server, each authority sender system has its own `/contacts/` folder, and each of these folders contains that system's respective list. Every night new lists will be created and uploaded to the folder. At the same time, the lists from the previous day will be removed, so only the newest lists will be found in the folder. A new list can be expected to be uploaded between 3-4 am every night.

4.6.1 CSV format

The naming convension of the CSV file is `contacts.csv`
The file is in a semicolon format as the following:

Row 1: Heading that describes the header fields
Row 2: Header fields data

Row 3: Heading that describes the registration data fields
Row 4-n: Registration data fields

Header fields

Field name	Description
DannetDatoTid	Date time for when the list is created
SystemIdentifikator	List is the same for every system, so this field is always 0.
KompletIndikator	Indicate whether the list is complete. Always 1.

Registration data fields

Field name	Description
Modtager	Is the CPR or CVR of a contact.
ModtagerType	Indicates whether the contact is a citizen or company. P = citizen, V = company
IndholdsType	Indicates whether the contact is registered for Digital Post or NemSms. S = NemSms, D = Digital Post.
Tilmeldt	Always 1.

Example of CSV format

```
DannetDatoTid;SystemIdentifikator;KompletIndikator
2022-07-12 00:11:26;0;1
Modtager;ModtagerType;Indholdstype;Tilmeldt
0504554402;P;D;1
0504554403;P;D;1
0504554403;P;S;1
99881101;V;D;1
77227711;V;D;1
0101010000;P;D;1
1212826357;P;D;1
12112114;V;D;1
1211112114;P;D;1
10101010;V;D;1
2424454554;P;D;1
2609881234;P;D;1
1010947896;P;D;1
2609881233;P;D;1
12345679;V;D;1
11223344;V;D;1
```

```
44332233;V;D;1
12341234;V;D;1
12341234;V;P;1
0707920707;P;D;1
0707920707;P;S;1
2609881236;P;D;1
2609881237;P;D;1
11554477;V;D;1
0101101234;P;D;1
```

5 System registry services - TI

The systems registry is responsible for four logical domains in Digital Post:

- Organisation: Used to keep track of which companies are authorities, their authority type, their logo as well as who is allowed to send legal and mandatory messages
- System: The system resource is a representation of connected sender and receiver systems, their technical details such as IP's service protocol and API token
- Contact structure: The contact structure is the structure used to define how the authority can be contacted and how the messages are directed to different departments inside the authority. It consists of the `ContactPoint` and `ContactGroup` resources, which together define the contact structure
- Contact structure subscription: A possibility for sender systems to subscript to changes to all or some contact structures

These logical domains are editable accessible through the following services which all are exposed by the system registry.

5.1 Organisations

Service	URL	Usage	Required roles	Open API
Query organisations	`GET /organisations/`	Fetching one or multiple organisations by CVR number, name, type and searchTerm	• Any role	Swagger UI
Fetch organisation	`GET /organisations/{organisation-id}`	Fetching a single organisation by organisationId	• Any role	Swagger UI
Update organisation	`PUT /organisations/{organisation-id}`	Updating an organisation	• Employee • Rights administrator • Organisation administrator • System manager • Contact administrator	Swagger UI

Service	URL	Usage	Required roles	Open API
Add logo content to organisation	`PUT /organisations/{id}/logo`	Adding logo to authorities	• Employee • Rights administrator • Organisation administrator • System manager • Contact administrator	Swagger UI
Get logo content	`GET /organisations/{id}/logo`	Getting logo byte content	• Any role	Swagger UI
Delete logo	`DELETE /organisations/{id}/logo`	Removed logo from organisation	• Employee • Rights administrator • Organisation administrator • System manager • Contact administrator	Swagger UI

5.2 Systems

Service	URL	Usage	Required roles	Open API
List Systems	`GET /organisations/{organisation-id}/systems/`	Fetching all systems from an organisation, that the user is allowed to view	• System administrator • Contact administrator • Organisation administrator • Delegated Support Administrator	Swagger UI

Service	URL	Usage	Required roles	Open API
Add a system	`POST /organisations/{organisation-id}/systems/`	Add a system to an organisation	• Organisation Administrator • System Manager	Swagger UI
Remove a system	`DELETE /organisations/{organisation-id}/systems/{id}`	Remove a system from an organisation	• Organisation Administrator • System Manager	Swagger UI
Change default Recipient	`POST /organisations/{organisation-id}/systems/{id}/makedefaultrecipient`	Change Default recipient system, this action will mark the target system as RECIPIENT_DEFAULT and remove the marking on the existing	• Organisation Administrator • System Manager	Swagger UI
Renew API Token	`POST /organisations/{organisation-id}/systems/{id}/renewapitoken`	Renew API token. Create new API token if not already existed or Update API token value if not existed. Existing OCES certificate will be removed	• Organisation Administrator • System Manager	Swagger UI
Fetch system	`GET /organisations/{organisation-id}/systems/{system-id}`	Fetching system by ID	• System administrator • Contact administrator • Organisation administrator • Delegated Support Administrator	Swagger UI

Service	URL	Usage	Required roles	Open API
Update system	`PUT /organisations/{organisation-id}/systems/{system-id}`	Updating a system by Id	• System administrator • Organisation administrator	Swagger UI
Upload SSH-key for system	`POST /organisations/{organisation-id}/systems/{system-id}/sshkey`	Creating/updating SSH-key for system	• System administrator • Organisation administrator	Swagger UI

5.3 Contact structure

The contact structure consist of the `ContactPoint` and `ContactGroup` resources. The structure can be read anonymously since users should be able to identity where to contact an authority before sending the message, which requires the an authorized user.

Service	URL	Usage	Required roles	OpenAPI
Query contact points	`GET /organisations/{organisation-id}/contact-points/`	Fetching some or all contact points	• Any role	Swagger UI
Create a contact point	`POST /organisations/{organisation-id}/contact-points/`	Add a contact point	• Contact Administrator • System Manager	Swagger UI

Service	URL	Usage	Required roles	OpenAPI
Fetch contact point	`GET /organisations/{organisation-id}/contact-points/{contact-point-id}`	Fetching contact point by ID	• Any role	Swagger UI
Update contact point	`PUT /organisations/{organisation-id}/contact-points/{contact-point-id}`	Updating a contact point by ID	• Contact Administrator • System Manager	Swagger UI
Delete contact point	`DELETE /organisations/{organisation-id}/contact-points/{id}`	Delete a contact point given the ID	• Contact Administrator • System Manager	Swagger UI
Query contact groups	`GET /organisations/{organisation-id}/contact-groups/`	Fetching some or all contact groups	• Any role	Swagger UI
Create a contact group	`POST /organisations/{organisation-id}/contact-groups/`	Add a contact group	• Contact Administrator • System Manager	Swagger UI

Service	URL	Usage	Required roles	OpenAPI
Fetch contact group	`GET /organisations/{organisation-id}/contact-groups/{contact-group-id}`	Fetching contact group by ID	• Any role	Swagger UI
Update contact group	`PUT /organisations/{organisation-id}/contact-groups/{contact-group-id}`	Updating a contact group by ID	• Contact Administrator • System Manager	Swagger UI
Delete contact group	`DELETE /organisations/{organisation-id}/contact-groups/{id}`	Delete a contact group given the ID	• Contact Administrator • System Manager	Swagger UI
Query contact groups across organisations	`GET /contact-groups/`	Finding a contact groups without knowing which organisation it belongs to	• Any role	Swagger UI
Query contact points across organisations	`GET /contact-points/`	Finding a contact points without knowing which organisation it belongs to	• Any role	Swagger UI

5.4 Querying in System registry APIs

For description of common search functionality, please revisit the section **Querying and searching** resources.

The system registry exposes endpoint to fetch organisation. An organisation in the system registry contains multiple sub-resources, these are sender and receiver systems (called systems), contact point and contact groups.

Each have their own set of points that can be queried. This is done to accommodate multiple users editing simultaneously in both the contact structure (contact point and groups) and systems of a single organisation.

The following five endpoints have been exposed externally from System Registry:

- /organisations/
  - Queries all organisations the user is allowed to see. A system manager can see all organisations, while an anonymous user can only see authorities. Likewise is it with regards to how much is returned. An organisation administrator can see all fields of own organisation but only certain fields of authorities.
- /contact-groups/
  - Queries all contact groups across all organisations. A contact group can have a parent contact group or no parent if the group is placed at the top of the hierarchy.
- /organisations/{organisation-id}/contact-groups/
  - Queries contact groups under specified organisation.
- /organisations/{organisation-id}/contact-points/
  - Returns list of contact-points under specified organisation. Each contact-point has a list of contact-groups (it can now belong to more than one group). If the contact-point is at the top at the hierarchy, the contact-group list is empty. A contact-point **cannot** both exist in a group **and** also at the root-level.
- /organisations/{organisation-id}/systems/
  - Returns a list of sender and receiver systems that the given organisation currently has created.

The result is an `OrganisationSearchResult`, which looks like this in JSON:

```
{
  "currentPage": 0,
  "totalPages": 1,
  "elementsOnPage": 1,
  "totalElements": 1,
  "organisations": []
}
```

5.4.1 Fetching organisations on ID

```
/organisations/70b29451-a265-427c-ac7b-cda01413a080
```

will fetch the organisation with id 70b29451-a265-427c-ac7b-cda01413a080 (Random UUID with no actual organisation behind it).

Searching for organisations

Searching can be done using the generic search mentioned above and/or using the following parameters:

id	Organisation ID(s) to search for. Used to fetch many organisations with known IDs at the same time
name	Name(s) of organisation(s) to search for
cvrNumber	One or more CVR numbers to search for

systemId	One or more systemIds to search for
type	A organisation type to search for. One of `AUTHORITY` `COMPANY`.

Examples

Generally the format is:

```
/organisations/?<parameter>=<value>
/organisations/?<parameter>=<value>,<value>,<value>
/organisations/?<parameter>=<value>&<parameter>=<value>&<parameter>=<value>
```

It's possible to mix-and-match different parameters. For lines 2 and 3 it's possible to add as many parameters/values as desired.
Example:

```
/organisations/?cvrNumber=1234567890
```

returns the organisation with CVR number 1234567890

```
/organisations/?cvrNumber=1234567890,2345678901
```

would get us the two organisations with the given CVR numbers.

5.4.2 Searching for ContactPoints across all organisations or within an Organisation

Querying for ContactPoints within an organisation is done using a GET request to either of the endpoints:

*/organisations/*id/contact-points/

/contact-points/

Where id is the id of the organisation.

The result is a page of contactPointSearchResult, which looks like this in JSON:

```
{
  "currentPage": 0,
  "totalPages": 1,
  "elementsOnPage": 1,
  "totalElements": 1,
  "contactPoints": []
}
```

It can be paged, just like the organisationSearchResult and the structure of a contactpoint is part of the organisation

`/organisations/{id}/contact-points/`

The endpoint supports each searching against all properties on the ContactPoint. Additionally the endpoint also has the following search options:

isRoot	boolean. If true, the only contact-point that is not related to any contact-groups is returned. If provided as false, only contact-point associated to any contact-group is returned
searchTerm	One or more search terms to search for
searchTermOrName	One or or more search words that can be found in either search terms OR name fields
postkasseId	One id to search for
postkasseEmneId	One id to search for

Examples

Generally the format is:

```
/organisations/id/contact-points/?<parameter>=<value>
/organisations/id/contact-points/?<parameter>=<value>,<value>,<value>
/organisations/id/contact-points/?
<parameter>=<value>&<parameter>=<value>&<parameter>=<value>
```

It's possible to mix-and-match different parameters. For lines 2 and 3 it's possible to add as many parameters/ values as desired.

```
/organisations/70b29451-a265-427c-ac7b-cda01413a080/contact-points/?
```

returns all the contact points for the organisation with id 70b29451-a265-427c-ac7b-cda01413a080

```
/organisations/70b29451-a265-427c-ac7b-cda01413a080/contact-points/?
searchTerm=Folkeskole
```

returns all the contact points with search term "Folkeskole" belonging to the organisation with ID 70b29451-a265-427c-ac7b-cda01413a080

```
/organisations/70b29451-a265-427c-ac7b-cda01413a080/contact-points/?
searchTerm=Folkeskole&contactGroupId=456,789
```

returns all the contact points that are in contact groups with either ID 456 or 789 that also include search term "Folkeskole" belonging to the organisation with ID 70b29451-a265-427c-ac7b-cda01413a080

```
/organisations/70b29451-a265-427c-ac7b-cda01413a080/contact-points/?
searchTermOrName=Røntgen*&active=true&visible=true
```

returns all the contact points where either search terms starts with “Røntgen” OR name starts with “Røntgen“ AND is active AND visible, belonging to the organisation with ID 70b29451-a265-427c-ac7b-cda01413a080

5.4.3 Searching for Systems within an organisation

Searching for systems is done using a GET request to the */organisations/id/systems/* endpoint.

The result is a SystemSearchResult, which looks like this in JSON:

```
{
  "currentPage": 0,
  "totalPages": 1,
  "elementsOnPage": 1,
  "totalElements": 1,
  "systems": []
}
```

Example

```
/organisations/70b29451-a265-427c-ac7b-cda01413a080/systems/
```

returns all systems belonging to organisation with ID 70b29451-a265-427c-ac7b-cda01413a080. You can search for any property on a System.

5.4.4 Searching for Contact Groups across all organisations or within an organisation

Querying contact-groups is done using a GET request to either of the endpoints

- */contact-groups/*
- */organisations/id/contact-groups/*

The result is a ContactGroupSearchResult, which looks like this in JSON:

```
{
  "currentPage": 0,
  "totalPages": 1,
  "elementsOnPage": 1,
  "totalElements": 1,
  "contactGroups": []
}
```

Examples

```
/organisations/70b29451-a265-427c-ac7b-cda01413a080/contact-groups/
```

returns all contact groups belonging to organisation with id 70b29451-a265-427c-ac7b-cda01413a080. There are not parameters to narrow down the results.

```
/organisations/70b29451-a265-427c-ac7b-cda01413a080/contact-groups/?name=pas
```

returns all contact groups that have “pas” in their name somewhere. This includes examples like: “Bestil nyt pas” and just “Pas”.
These parameters are available for use when searching for contact groups:

id	ID of the organisation to search for contact points in
name	name of the contact group

5.5 Fetching hidden contact points and contact groups

Contact points and contact groups both have the boolean field “visible” that determines if a point or group is hidden. The need for this is to support deep-links in the contact point structure. Given the necessary privileges a query with the parameter “visibility“ can be made. The visibility parameter has three different values:

- VISIBLE - Shows only points and groups with visible = true
- HIDDEN - Shows only points and groups with visible = false
- ALL - Shows all points and groups regardless of visibility

Examples

`/organisations/70b29451-a265-427c-ac7b-cda01413a080/contact-groups/?visibility=HIDDEN`

Returns all contact groups belonging to organization with id `70b29451-a265-427c-ac7b-cda01413a080`, that have visible = false

`/organisations/70b29451-a265-427c-ac7b-cda01413a080/contact-points/?visibility=VISIBLE`

Returns all contact points belonging to organization with id `70b29451-a265-427c-ac7b-cda01413a080`, that have visible = true

`/organisations/70b29451-a265-427c-ac7b-cda01413a080/contact-groups/?id=fd8ca657-157c-3ffe-0000-000000000e96&visibility=HIDDEN`

Returns a single contact group with id `fd8ca657-157c-3ffe-0000-000000000e96` belonging to organization with id `70b29451-a265-427c-ac7b-cda01413a080,` that have visible = false

`/organisations/70b29451-a265-427c-ac7b-cda01413a080/contact-points/?id=163710ae-5078-350c-0000-00000000306a&visibility=ALL`

Returns a single contact point with id `163710ae-5078-350c-0000-00000000306a` and organization id `70b29451-a265-427c-ac7b-cda01413a080` regardless of visibility.

5.6 Find contact-groups structured below other contact-groups

This section is an example approach on how view clients can locate all the contact-groups that are placed below a given contact-group in the contact-structure. This is not intended as a copy paste implementation, but should be used as an point of inspiration.

The endpoint:

```
/organisations/{organisation-uuid}/contact-groups/
```

Returns list of contact-groups under a specified organisation like the example below. Each group has a `parent` referring to either it's parent-group or is empty (if the group is placed at the top of the hierarchy). The example shown below targets the resource at `/organisations/14d6e9ad-fc0e-3918-0000-00000000017b/contact-groups/`


```
{
    "currentPage": 0,
    "totalPages": 1,
    "elementsOnPage": 39,
    "totalElements": 39,
    "contactGroups": [
        {
            "id": "fd8ca657-157c-3ffe-0000-000000001bd0",
            "organisationalUnit": false,
            "name": "EGTO-InCorp",
            "visible": true,
            "description": "",
            "organisation": {
                "id": "14d6e9ad-fc0e-3918-0000-00000000017b",
                "name": "Silkeborg Kommune",
                "cvrNumber": "29189641",
                "type": "AUTHORITY",
                "authorityType": "UNKNOWN",
                "authorityTerms": false,
                "logoAvailable": false,
                "targets": []
            },
            "parent": {
                "id": "fd8ca657-157c-3ffe-0000-0000000016cd",
                "organisationalUnit": false,
                "name": "Silkeborg Kommune",
                "visible": true,
                "description": "",
                "organisation": {
                    "id": "14d6e9ad-fc0e-3918-0000-00000000017b",
                    "name": "Silkeborg Kommune",
                    "cvrNumber": "29189641",
                    "type": "AUTHORITY",
                    "authorityType": "UNKNOWN",
                    "authorityTerms": false,
                    "logoAvailable": false,
```

```
                    "targets": []
                },
                "targets": []
            },
            "targets": []
        }
        .
        .
        .
}
```

A group does not have a link to the groups located below the group in the contact structure. To find all children of the group (and the children of the children etc.) the following recursive code snippet can be used. The type ContactGroup is a simple mapping from the JSON object to a typescript class.

```
private contactGroups: Array<ContactGroup> = this.getGroupsFromNgDp()

public findChildren(startGroupId: string): Array<string> {
  const arrayWithIds:Array<string> = [];
  this.addGroupToArray(arrayWithIds, startGroupId);
  return arrayWithIds;
}

private addGroupToArray(arrayWithIds: Array<string>, groupId: string): void {
  this.contactGroups.filter(contactGroup => contactGroup.parent.id ===
groupId).forEach(contactGroup => {
    this.addGroupToArray(arrayWithIds, contactGroup.id);
    arrayWithIds.push(contactGroup.id);
  })
}
```

5.7 Updating items in system registry

This page describes how to, and what to include when updating items in the System registry.

Update an organisation information

Updating an organisation is done using a PUT request to the following endpoints:
/organisations/{orgId}
/organisations/{orgId}/contact-points/{id}
/organisations/ {orgId}/contact-groups/{id}
/organisations/ {orgId}/systems/{id}

With an PUT request the organisationId needs to be set. If making a PUT request on a contact-points, contact-groups or systems their ID needs to be set too.

When updating a full organisation only a few fields are optional. The optional fields are shown in the table below. If a field is not shown in the table, it means that the field is mandatory.

Organisation Entity	**Optional Fields**
Organisation	• privacyPolicyUrl
Contact-Point Entity	**Optional Fields**
ContactPoint	• memoSizeThresholdMB (defaults to 95.5 MB) • allowedNumberOfAttachements (defaults to 10) • ContactPointCode • ContactGroup ((id of multiple or no groups) • ReportLink • RecommededAttributes • internalDescription
ReportLink	• description • label
recommendedAttributes	• name
ContactPointCode	• codeVersion (not optional for type: FORM and KLE)
Contact-Group Entity	**Optional Fields**
ContactGroup	• parentId (id of parent group, null if top)
System Entity	**Optional Fields**
System	• activeTo (only available if activeFrom is set) • activeFrom • supplier • businessContactEmail • standardSystemTemplate (are allowed to be empty) • contactPoints (id of none to multiple contact points)
standardSystemTemplateId	optional

System Entity	Optional Fields
allowedIps	optional
receiptFormat	optional
receiptEndpoint	Should be present for sender system and be empty for recipient systems
endpoint	REST_PULL recipient systems and sender systems

Examples

IDs in examples will have to be fitted to the current environment.

Example of a PUT request for organisations:

```
{
    "authorityId": "string",
    "name": "string",
    "cvrNumber": "string",
    "type": "AUTHORITY",
    "privacyPolicyUrl": "string",
    "legalNotificationAllowed": true,
    "mandatoryPostAllowed": true,
    "systemFetch": true,
    "legalContact": {
        "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
        "version": 0,
        "name": "string",
        "email": "string",
        "phoneNumber": "string"
    }
}
```

Example of a PUT request for contactGroups

```
{
    "name": "string",
    "description": "string",
    "parentId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "organisationalUnit": true,
    "contactPoints": [
        "3fa85f64-5717-4562-b3fc-2c963f66afa6"
    ]
```

```
}
```

Example of a PUT request for contactPoints

```
{
    "name": "string",
    "description": "string",
    "internalDescription": "string",
    "targets": [
        "UNKNOWN"
    ],
    "memoSizeThresholdMB": 5,
    "allowedNumberOfAttachments": 0,
    "active": true,
    "visible": true,
    "systemId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "reportLink": {
        "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
        "version": 0,
        "externalLink": "https://digidp.atlassian.net/wiki",
        "externalLinkText": "https://digidp.atlassian.net/wiki",
        "description": "string",
        "label": "string"
    },
    "contactGroups": [
        "3fa85f64-5717-4562-b3fc-2c963f66afa6"
    ],
    "contactPointCode": [
        {
            "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
            "version": 0,
            "name": "string",
            "type": "CUSTOM",
            "codeVersion": "string",
            "contactPointCodeTypeName": "string",
            "code": "string"
        }
    ],
    "recommendedAttributes": [
        {
            "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
            "version": 0,
            "name": "string"
        }
    ],
    "searchTerms": [
        {
            "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
            "version": 0,
            "term": "string"
        }
```

```
    ]
}
```

Example of a PUT request for systems

```
{
    "name": "string",
    "endpoint": "string",
    "receiptEndpoint": "string",
    "certificateSerialNumber": "string",
    "activeFrom": "2019-01-19T07:57:05.294Z",
    "activeTo": "2021-01-19T07:57:05.294Z",
    "technicalContact": {
        "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
        "version": 0,
        "name": "string",
        "email": "string",
        "phoneNumber": "string"
    },
    "serviceProtocol": "REST_PUSH",
    "standardSystemTemplateId": null,
    "systemTypes": [
        "RECIPIENT_DEFAULT"
    ],
    "allowedIps": [
        {
            "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
            "version": 0,
            "ip": "string"
        }
    ],
    "contactPoints": [
        "3fa85f64-5717-4562-b3fc-2c963f66afa6",
        "3fa85f64-5717-4562-b3fc-2c963f66afa6"
    ],
    "receiptFormat": "MEMO",
    "supplier": "string",
    "businessContactEmail": "test@test.com"
}
```

Please note that neither the receiptEndpoint nor the endpoint are allow to be a .local domain.

Note: If “standardSystemTemplateId” is specified, common system fields after an update have to remain consistent with related StandardSystemTemplate. Otherwise mismatch exception is thrown and response states BAD_REQUEST (400).

Only the first default system created can be inactive (activeFrom set in the future), the next created default systems cannot be inactive.

5.8 Organisation contact-group/point subscription

In order to have a local registration list updated with the newest contact-point data, the authorities can subscribe to changes to contact-points and contact-groups.

As part of the subscription, the authority sender system will specify an endpoint where they will get notified every time there is an update to a contact-point/group that is included in their subscription.

An authority can only have one subscription per sender system.

The subscription can either be made on individual organisations or by the usage of subscriptionoperations.

When an organisation has made a subscription, it will contain all the CVR numbers of all the organisations that the subscribing organisation is interested in receiving notifications about. The amount of CVR numbers should be kept to a minimum, however there is no maximum limit. If the authority sender system tries to provide a CVR that does not exist in Digital Post system registry, the contact will be filtered out of the subscription (Nb. No error will be thrown).

`OrganisationsSubscriptionOperations` are some pre-defined filters for common use cases. There is a total of two operations:

- `ALL_ORGANISATIONS` - The operation includes only already existing contact-groups/points.
- `NEW_ORGANISATIONS` - The operation includes only newly created contact-groups/points.

One subscription can contain one or multiple operations.

5.9 Endpoint exposed in the system-subscription-store

Service	URL	Usage	Required roles	OpenAPI
Create subscription	POST /organisations/subscriptions/	Creating a new subscription to get notified when organisations changed. Only one subscription per sender system.	AUTHORITY_SENDER_SYSTEM	Swagger UI
List subscriptions	GET /organisations/subscriptions/	Listing all subscriptions	AUTHORITY_SENDER_SYSTEM	Swagger UI
Update subscription	PUT /organisations/subscriptions/{subscription-id}	Updating a subscription	AUTHORITY_SENDER_SYSTEM	Swagger UI

Example of a subscription creation with usage of individual contactIDs

```
{
```

```
   "organisationSubscriptions":[
      "12345678",
      "12345679",
      "12345671",
      "12345672",
      "12345673"
   ],
   "notificationUrl": "https://postman-echo.com/post"
}
```

Example of a subscription creation with usage of subscription operations:

```
{
  "subscriptionOperations": [
    "ALL_ORGANISATIONS"
  ],
  "notificationUrl": "https://postman-echo.com/post"
}
```

5.10 Notifications

A push notification via a REST POST request will be sent to the specified endpoint every time a contact-point/group is updated or created, and which matches the criteria from the subscription. The notification contain the type of resource CONTACT_POINT/CONTACT_GROUP, uuid of resource, version of resource, uuid of organisation the resource belongs to.

Example of Notification:

```
{
  "resourceId": "521f3356-d1c5-4812-bc69-59423fce52b7",
  "resourceVersion": 1,
  "type": "CONTACT_POINT",  (can be CONTACT_POINT/CONTACT_GROUP)
  "organisationId": "521f3356-d1c5-4812-bc69-59423fce52bb"
}
```

6 Mailbox services - TI

Blue text is related to feature “Allowing forwarding from SENT folder” and will be is supported in a later release.

The following services are exposed from the mailbox.

6.1 Mailboxes

Service	URL	Usage	Required roles	OpenAPI
Query mailboxes	`GET /mailboxes/`	Fetching all mailboxes user has access to with optional paging.	• Borger • Skriveadgang • Avanceret adgang • Kurator • Likvidator • Bobestyrer • Læseadgang • Legal ejer af lukket virksomhed • Læse-/ skriveadgang Basisadgang • Systemadministrator • TilbagekaldsadministratorBorger	Swagger UI

Service	URL	Usage	Required roles	OpenAPI
Fetch mailbox	`GET /mailboxes/{mailbox-id}`	Fetching a single mailbox.	• Borger • Skriveadgang • Avanceret adgang • Kurator • Likvidator • Bobestyrer • Læseadgang • Legal ejer af lukket virksomhed • Læse-/ skriveadgang Basisadgang • Systemadministrator • TilbagekaldsadministratorBorger	Swagger UI
Update mailbox	`PUT /mailboxes/{mailbox-id}`	Creating and updating email- sms - and push-notification subscriptions. Setting introduction completed flag.	• Borger • Avanceret adgang • Kurator • Likvidator • Bobestyrer • Læseadgang • Legal ejer af lukket virksomhed • Læse-/ skriveadgang	Swagger UI
List trustees	`GET /mailboxes/trustees`	Fetching all trustees of the mailbox owner, with optional sorting.	• Borger • Basis • Medarbejder • Kurator / bobestyrer • Læse- og skriveadgang (Full power of attorney)	Swagger UI

Service	URL	Usage	Required roles	OpenAPI
Fetch sender information	`GET /mailboxes/{id}/sender-information/`	Fetch a list of senders for messages to a specific mailbox. Each element of the list contains information about most recently received message, number of messages and unread messages. This list excludes information about senders containing CPR numbers	• Borger • Avanceret adgang • Kurator • Likvidator • Bobestyrer • Basisadgang • Læseadgang • Legal ejer af lukket virksomhed • Læse-/ skriveadgang	Swagger ui
Fetch mailbox overview for a single mailbox	`GET /mailboxes/overview/?cprNumber={cprNumber}`	Fetch a mailbox overview that details if notifications are available and give user a quick look on the available items. Only usable by My Overview client.	• Mit overblik client	Swagger ui

6.2 Accesses

Service	URL	Usage	Required roles	Open API
Create access	`POST /mailboxes/{mailbox-id}/accesses/`	Creating Access. Creating email-sms - and push-notification subscriptions. Setting introduction completed flag.	• Borger • Admin • Kurator / bobestyrer • Partsrepræsentant (læseadgang) • Læse- og skriveadgang (Full power of attorney)	Swagger UI

Service	URL	Usage	Required roles	Open API
Fetch access	`GET /mailboxes/{mailbox-id}/accesses/{access-id}`	Fetching a single Access	• Borger • Admin • Kurator / bobestyrer • Partsrepræsentant (læseadgang) • Læse- og skriveadgang (Full power of attorney)	Swagger UI
Update access	`PUT /mailboxes/{mailbox-id}/accesses/{access-id}`	Creating and updating email-sms - and push-notification subscriptions. Setting introduction completed flag.	• Borger • Admin • Kurator / bobestyrer • Partsrepræsentant (læseadgang) • Læse- og skriveadgang (Full power of attorney) • System Mailbox	Swagger UI
Query accesses	`GET /mailboxes/{mailbox-id}/accesses/`	Fetching all accesses on mailbox user is allowed to see	• Borger • Admin • Kurator / bobestyrer • Partsrepræsentant (læseadgang) • Læse- og skriveadgang (Full power of attorney)	Swagger UI

Service	URL	Usage	Required roles	Open API
Verify Access	`PUT /mailboxes/{mailbox-id}/accesses/{access-id}/subscriptions/{subscription-id}/verification`	Updates the verification time on an Access' subscription	• Borger • Admin • Kurator / bobestyrer • Partsrepræsentant (læseadgang) • Læse- og skriveadgang (Full power of attorney)	Swagger UI

6.3 Messages

Service	URL	Usage	Required roles	Open API
Create draft message	`POST /mailboxes/{mailbox-id}/messages/`	Creating a new draft message. Will be located in DRAFTS folder.	• Borger • Skriver • Basis • Læse- og skriveadgang (Full power of attorney)	Swagger UI
Create reply	`POST /mailboxes/{mailbox-id}/messages/{message-id}/reply`	Creating a reply template from a message. Will be a new draft message located in DRAFTS folder.	• Borger • Skriver • Basis • Læse- og skriveadgang (Full power of attorney)	Swagger UI

Service	URL	Usage	Required roles	Open API
Update draft message	`PUT /mailboxes/{mailbox-id}/messages/{message-id}`	Updating a message.	• Borger • Skriver • Basis • Læse- og skriveadgang (Full power of attorney)	Swagger UI
Send draft message	`PUT /mailboxes/{mailbox-id}/messages/{message-id}/send`	Sending the message. Will move message to SENT folder. If the message is a reply, a userActivity with type REPLIED is added to userActivities in the original message.	• Borger • Skriver • Basis • Læse- og skriveadgang (Full power of attorney)	Swagger UI
Forward message to e-mail address or trusted recipient	`POST /mailboxes/{mailbox-id}/messages/{message-id}/forward`	Forwarding a received or sent message to an e-mail address. Forwarding a received or sent message to a trusted recipient (present in users saml token). A userActivity with type FORWARDED is added to userActivities in the original message.	• Borger • Basis • Medarbejder • Kurator / bobestyrer • Partsrepræsentant (læseadgang) • Læse- og skriveadgang (Full power of attorney)	Swagger UI

Service	URL	Usage	Required roles	Open API
Forward message to authority	`POST / mailboxes/ {mailbox-id}/ messages/ {message-id}/ forward`	Forwarding a received or sent message to an authority using either CVR or contact point. A userActivity with type FORWARDED is added to userActivities in the original message.	• Borger • Basis • Medarbejder • Kurator / bobestyrer • Læse- og skriveadgang (Full power of attorney)	Swagger UI
Query messages	`GET / mailboxes/ {mailbox-id}/ messages/`	Fetching multiple messages in a mailbox with optional paging, searching, filtering and sorting.	• Borger • Skriver • Basis • Medarbejder • Admin • Kurator / bobestyrer • Partsrepræsentant (læseadgang) • Læse- og skriveadgang (Full power of attorney)	Swagger UI
Fetch message	`GET / mailboxes/ {mailbox-id}/ messages/ {message-id}`	Fetching a single message.	• Borger • Skriver • Basis • Medarbejder • Admin • Kurator / bobestyrer • Partsrepræsentant (læseadgang) • Læse- og skriveadgang (Full power of attorney)	Swagger UI

Service	URL	Usage	Required roles	Open API
Update message	`PUT /mailboxes/{mailbox-id}/messages/{message-id}`	Updating a message's flags, folder and note	• Borger • Skriver • Basis • Medarbejder • Kurator / bobestyrer • Partsrepræsentant (læseadgang) • Læse- og skriveadgang (Full power of attorney)	Swagger UI
Patch message	`PATCH /mailboxes/{mailbox-id}/messages/{message-id}`	Updating one or more of certain predetermined fields of a message such as folderId, flags, recipient etc.	• Borger • Skriver • Basis • Medarbejder • Kurator / bobestyrer • Partsrepræsentant (læseadgang) • Læse- og skriveadgang (Full power of attorney)	Swagger UI
Delete message	`DELETE /mailboxes/{mailbox-id}/messages/{message-id}`	Deleting a message	• Borger • Medarbejder • Tilbagekaldsadministrator • Læse- og skriveadgang (Full power of attorney) (only for drafts)	Swagger UI
Fetch unread status	`GET /mailboxes/messages/unread/exists`	Returns true/false of whether the mailbox of a given CPR/CVR contains unread REGULAR messages from within the last 6 months	• Borgerservice • Erhvervsservice • Systemforvalter • Læse- og skriveadgang (Full power of attorney)	Swagger UI

Service	URL	Usage	Required roles	Open API
Fetch unread status without query params (recommended over GET equivalent)	`POST /mailboxes/messages/unread/exists/`	Returns true/false of whether the mailbox of a given CPR/CVR contains unread REGULAR messages from within the last 6 months, but specifying the type (CPR/CVR) and CPR/CVR value within the body rather than as parameters.	• Borgerservice • Erhvervsservice • Systemforvalter • Læse- og skriveadgang (Full power of attorney)	Swagger UI

6.4 Documents

Service	URL	Usage	Required roles	Open API
List documents	`GET /mailboxes/{mailbox-id}/messages/{message-id}/documents/`	Fetching all documents of a message	• Borger • Skriver • Basis • Medarbejder • Admin • Kurator / bobestyrer • Partsrepræsentant (læseadgang) • Læse- og skriveadgang (Full power of attorney)	Swagger UI

Service	URL	Usage	Required roles	Open API
Fetch document	`GET /mailboxes/{mailbox-id}/messages/{message-id}/documents/{document-id}`	Fetching a single document	• Borger • Skriver • Basis • Medarbejder • Admin • Kurator / bobestyrer • Partsrepræsentant (læseadgang) • Læse- og skriveadgang (Full power of attorney)	Swagger UI
Create document	`POST /mailboxes/{mailbox-id}/messages/{message-id}/documents/`	Creating a document	• Borger • Skriver • Basis • Læse- og skriveadgang (Full power of attorney)	Swagger UI
Update document	`PUT /mailboxes/{mailbox-id}/messages/{message-id}/documents/{document-id}`	Updating a document	• Borger • Skriver • Basis • Læse- og skriveadgang (Full power of attorney)	Swagger UI

Service	URL	Usage	Required roles	Open API
Delete document	`DELETE /mailboxes/{mailbox-id}/messages/{message-id}/documents/{document-id}`	Deleting a single document	• Borger • Skriver • Basis • Læse- og skriveadgang (Full power of attorney) (only when messageState is draft)	Swagger UI

6.5 Files

Service	URL	Usage	Required roles	Open API
List files	`GET /mailboxes/{mailbox-id}/messages/{message-id}/documents/{document-id}/files/`	Fetching all files of a document	• Borger • Skriver • Basis • Medarbejder • Admin • Kurator / bobestyrer • Partsrepræsentant (læseadgang) • Læse- og skriveadgang (Full power of attorney)	Swagger UI

Service	URL	Usage	Required roles	Open API
Fetch file	`GET /mailboxes/{mailbox-id}/messages/{message-id}/documents/{document-id}/files/{file-id}`	Fetching a single file	• Borger • Skriver • Basis • Medarbejder • Admin • Kurator / bobestyrer • Partsrepræsentant (læseadgang) • Læse- og skriveadgang (Full power of attorney)	Swagger UI
Fetch file content	`GET /mailboxes/{mailbox-id}/messages/{message-id}/documents/{document-id}/files/{file-id}/content`	Fetching file content bytes	• Borger • Skriver • Basis • Medarbejder • Admin • Kurator / bobestyrer • Partsrepræsentant (læseadgang) • Læse- og skriveadgang (Full power of attorney)	Swagger UI

Service	URL	Usage	Required roles	Open API
Create file	`POST /mailboxes/{mailbox-id}/messages/{message-id}/documents/{document-id}/files/`	Creating a file	• Borger • Skriver • Basis • Læse- og skriveadgang (Full power of attorney)	Swagger UI
Update file	`PUT /mailboxes/{mailbox-id}/messages/{message-id}/documents/{document-id}/files/{file-id}`	Updating a file	• Borger • Skriver • Basis • Læse- og skriveadgang (Full power of attorney)	Swagger UI

Service	URL	Usage	Required roles	Open API
Delete file	`DELETE /mailboxes/{mailbox-id}/messages/{message-id}/documents/{document-id}/files/{file-id}`	Deleting a single file from a document	• Borger • Skriver • Basis • Læse- og skriveadgang (Full power of attorney) (only for drafts)	Swagger UI
Update file content	`PUT /mailboxes/{mailbox-id}/messages/{message-id}/documents/{document-id}/files/{file-id}/content`	Update the contents of a file	• Borger • Skriver • Basis • Læse- og skriveadgang (Full power of attorney)	Swagger UI

6.6 Folders

Service	URL	Usage	Required roles	Open API
Create folder	`POST /mailboxes/{mailbox-id}/folders/`	Creating a folder.	• Borger • Medarbejder • Kurator / bobestyrer • Partsrepræsentant (læseadgang) • Læse- og skriveadgang (Full power of attorney)	Swagger UI
Fetch folder	`GET /mailboxes/{mailbox-id}/folders/{folder-id}`	Fetching a single folder.	• Borger • Skriver • Basis • Medarbejder • Admin • Kurator / bobestyrer • Partsrepræsentant (læseadgang) • Læse- og skriveadgang (Full power of attorney)	Swagger UI
Query folders	`GET /mailboxes/{mailbox-id}/folders/`	Fetching folders in a mailbox with optional paging.	• Borger • Skriver • Basis • Medarbejder • Admin • Kurator / bobestyrer • Partsrepræsentant (læseadgang) • Læse- og skriveadgang (Full power of attorney)	Swagger UI
Update folder	`PUT /mailboxes/{mailbox-id}/folders/{folder-id}`	Updating a folder.	• Borger • Medarbejder • Kurator / bobestyrer • Partsrepræsentant (læseadgang) • Læse- og skriveadgang (Full power of attorney)	Swagger UI

Service	URL	Usage	Required roles	Open API
Delete folder	`DELETE /mailboxes/{mailbox-id}/folders/{folder-id}`	Deleting a folder.	• Borger • Medarbejder • Læse- og skriveadgang (Full power of attorney)	Swagger UI
Fetch folder summary	`GET /mailboxes/{id}/folders/summary/`	Fetches a list of folders and a summary of total and unread amount of messages for each folder	• Borger • Skriver • Basis • Medarbejder • Admin • Kurator / bobestyrer • Partsrepræsentant (læseadgang) • Læse- og skriveadgang (Full power of attorney)	Swagger UI

6.7 System fetches

Service	URL	Usage	Required roles	Open API
Create SystemFetch	`POST /mailboxes/{mailbox-id}/system-fetches/`	Creates a system fetch job that starts emptying a mailbox into a recipient system (only received messages that are not in the deleted folder).	• Admin	Swagger UI

Service	URL	Usage	Required roles	Open API
Fetch SystemFetch	`GET /mailboxes/{mailbox-id}/system-fetches/{system-fetch-id}`	Fetches existing system fetch.	• Admin	Swagger UI
Update SystemFetch	`PUT /mailboxes/{mailbox-id}/system-fetches/{system-fetch-id}`	Updates a system fetch. Only option is to request it STOPPED using the status type. Stops the running job that fetches	• Admin	Swagger UI
Delete SystemFetch	`DELETE /mailboxes/{mailbox-id}/system-fetches/{system-fetch-id}`	Stops running job and deletes SystemFetch resource	• Admin	Swagger UI
List SystemFetch	`GET /mailboxes/{mailbox-id}/system-fetches/`	Returns all SystemFetch for the mailbox	• Admin	Swagger UI

6.8 Querying for Messages

For description of common search functionality, please revisit the section “Querying and searching resources”.

Querying messages is done using a GET request to the */mailboxes/{id}/messages/* endpoint.

The result is a messageSearchResult, which looks like this in JSON:

```
{
  "currentPage": 0,
  "totalPages": 1,
  "elementsOnPage": 1,
  "totalElements": 1,
  "messages": []
}
```

6.8.1 Searching

Besides, and in combination with, the general searching described above, the following parameters can be used:

folderId	One or more UUIDs of the folder(s) to filter by. Either separated by comma of by repeating the request parameter. Folders can be found using `GET /mailboxes/{id}/folders/`
dateFrom	Used to search for messages created on or after this date. Format: yyyy-MM-dd.
dateTo	Used to search for messages created on or before this date. Format: yyyy-MM-dd.

Examples

Folder:

```
/mailboxes/{id}/messages/?folderId=5d2e4e6a-40dc-477a-
ade9-5bd53afd1d3e,5e8c5915-3433-4adf-b80c-c8f94e198284
/mailboxes/{id}/messages/?folderId=5d2e4e6a-40dc-477a-
ade9-5bd53afd1d3e&folderId=5e8c5915-3433-4adf-b80c-c8f94e198284
```

returns messages located in in either of the two given folders.

```
/mailboxes/{id}/messages/?folderId=!5d2e4e6a-40dc-477a-ade9-5bd53afd1d3e
```

returns messages not in the given folder.

Date range:

```
/mailboxes/{id}/messages/?dateFrom=2020-04-01&dateTo=2020-04-15
```

searches for messages created between 2020-04-01 and 2020-04-15 - both included.

6.8.2 Masked CPR

When querying the messages all data connected to senders and recipients that include CPR will be masked by replacing the last 4 digits with **** in the response. This includes the fields: `sender.senderId`, `recipient.recipientId`, `sender.attentionData.contentResponsibleId`, `sender.representative.representativeId`, `forwardData.originalSender`, `forwardData.originalContentResponsible`, and `forwardData.originalRepresentative`.

6.9 Common use case examples

6.9.1 Update access

The If-Match header must be set to the version of the Access ressource. The latest can be fetched using:

`GET /mailboxes/8deb553b-0536-4671-9c9e-239f202d56e0/accesses/ce65def2-eb5b-4d4d-86da-4f70ffe9f6e1`


```
{
    "id": "ce65def2-eb5b-4d4d-86da-4f70ffe9f6e1",
    "version": 3,
    "transactionId": "EnIowtKlvFhKtjc5UJkmsoqiphRf2jyO",
    "createdDateTime": "2020-07-03T08:24:15.230Z",
    "lastUpdated": "2020-07-03T08:32:33.173Z",
    "accessType": "OWNER",
    "mailboxId": "8deb553b-0536-4671-9c9e-239f202d56e0",
    "introductionCompleted": false,
    "emailNotificationSubscriptions": []
}
```


- in this case, the version shown in the response is 3.

Adding smsNotificationSubscription and emailNotificationSubscriptions:

`PUT /mailboxes/8deb553b-0536-4671-9c9e-239f202d56e0/accesses/ce65def2-eb5b-4d4d-86da-4f70ffe9f6e1`

```
{
    "introductionCompleted": false,
    "smsNotificationSubscription": {
      "mobileNumber": "29892630"
    },
    "emailNotificationSubscriptions": [
      {
        "email": "test@nc.dk"
      }
    ]
```

```
}
```

Updating the email-address:

`PUT /mailboxes/8deb553b-0536-4671-9c9e-239f202d56e0/accesses/ce65def2-eb5b-4d4d-86da-4f70ffe9f6e1`

```
{
    "introductionCompleted": false,
    "smsNotificationSubscription": {
      "id": "8e2b1cf0-1b3c-4db6-950a-663f26209f3d",
      "version": 0,
      "unlistedNumber": false,
      "mobileNumber": "29892630"
    },
    "emailNotificationSubscriptions": [
      {
        "id": "87052e65-67da-4bbc-bbd7-ecd78cfa1928",
        "version": 0,
        "email": "test2@nc.dk"
      }
    ]
}
```

6.9.2 Create folder

Endpoint for creation of folders: `/mailboxes/{mailboxId}/folders/`

Example:

`POST /mailboxes/8deb553b-0536-4671-9c9e-239f202d56e0/folders/`

With a body that follows one of structures below.

Folder creation without a parent folder:

```
{
    "folderType": "USER_DEFINED",
    "name": "My archive"
}
```

Folder creation with a parent folder:

```
{
    "parentFolderId" : "0cd94f1a-4e6b-4065-8f1c-ac95371df03b",
    "folderType": "USER_DEFINED",
    "name": "Sub folder"
}
```

6.9.3 Create draft message

Endpoint for creation of a draft message: `/mailboxes/{mailboxId}/messages/`

Example:

`POST /mailboxes/8deb553b-0536-4671-9c9e-239f202d56e0/messages/`

Create a draft message with default values.

This *empty* request will construct a message, placed in the default folder DRAFTS, with one main html document ready for content:

```
{
}
```

Returns:


```
{
  "id": "a00aca0a-920c-4a81-945a-1ecddca52964",
  "version": 0,
  "mailboxId": "8deb553b-0536-4671-9c9e-239f202d56e0",
  "folderId": "767442c8-9b93-41f0-8ced-509c397ab934",
  "transactionId": "EpYjn8kP0N7wVASwtRTEWYCgrbjyM0US",
  "createdDateTime": "2020-08-17T17:41:26.634Z",
  "lastUpdated": "2020-08-17T17:41:26.634Z",
  "messageType": "REGULAR",
  "forward": false,
  "reply": false,
  "flag": false,
  "legallyNotified": false,
  "read": false,
  "welcomeMessage": false,
  "errorMessage": false,
  "sender": {
    "id": "98776603-058d-46aa-afaf-8b3b7381ca47",
    "version": 0,
    "senderId": "010378****",
    "senderIdType": "CPR"
  },
  "replyData": [],
  "documents": [
    {
      "id": "f63fc5b8-9651-4aa9-b131-8eeb05d85ee5",
      "version": 0,
      "documentType": "MAIN",
      "label": "Hoveddokument",
      "files": [
        {
          "id": "cbd20048-194c-42fa-aa3a-fa38191047cd",
          "version": 0,
          "encodingFormat": "text/html",
          "filename": "hoveddokument.html",
```

```
          "language": "da"
        }
      ],
      "actions": []
    }
  ]
}
```

If the senderIdType is a CPR, the value of senderId will be masked and the last 4 digits will be replaced with ****.

6.9.4 Update draft message

- Update newly created draft
- Update a draft with different recipient (Citizen)
- Update a draft with different masked recipient

A draft message can be created as described in the "Create draft message".

6.10 Update newly created draft

The only fields on the draft that can be updated are: `recipient`, `label`, `flag`, `read` and `folderId`. In order to update the fields, it is necessary to enrich the body of the current draft with new fields.

The following is an example of updating the recipient and label, as seen below.

```
"recipient": {
    "recipientId": "44556679",
    "recipientIdType": "CVR"
},
```

```
    "label": "Hejsa Lone Defaultssen",
```

In order to update the draft, the label and recipient should be added to the existing draft and sent in the body of the PUT request. Please note if the recipient contains `recipientIdType: CVR` then the recipient must be present in Digital Post, otherwise the update will fail during validation.

Both the label and recipient from above are added in the example of the PUT request body:

```
{
    "id": "2ea949fe-ebf3-4035-926c-7dea3dc0c6b5",
    "version": 0,
    "mailboxId": "e000a1d2-2933-44a6-a126-071ccdb4e090",
    "folderId": "a3fe6924-aac1-46e9-bcb4-1138b14a19a0",
    "transactionId": "EyLyCQBRxwxyFsfb4bzJrSXeDUiLCsvO",
    "createdDateTime": "2021-02-10T13:55:06.914Z",
    "lastUpdated": "2021-02-10T13:55:06.914Z",
    "messageType": "REGULAR",
    "forward": false,
    "reply": false,
    "flag": false,
```

```
    "legallyNotified": false,
    "read": true,
    "welcomeMessage": false,
    "errorMessage": false,
    //NEWLY ADDED LABEL DATA
    "label": "Hejsa Lone Defaultssen",
    //SENDER DATA CREATED DURING CREATION OF THE DRAFT
    "sender": {
        "id": "64bad6cd-9d26-4b6d-866d-4f689c0e0107",
        "version": 0,
        "senderId": "230792****",
        "senderIdType": "CPR",
        "label": "Lone 2076524521 Defaultssen"
    },
    //NEWLY ADDED RECIPIENT DATA
     "recipient": {
     "recipientId": "44556679",
     "recipientIdType": "CVR"
    },
    "replyData": [],
    "documents": [
        {
            "id": "4d253d55-ac00-4332-9b2f-fd4b269a164b",
            "version": 0,
            "documentType": "MAIN",
            "label": "Hoveddokument",
            "files": [
                {
                    "id": "b8509a8f-d454-4cb2-b9de-b6d86061278b",
                    "version": 0,
                    "encodingFormat": "text/html",
                    "filename": "hoveddokument.html",
                    "language": "da"
                }
            ],
            "actions": []
        }
    ]
}
```

This body should be used to update the draft on the endpoint provided below:

`PUT /mailboxes/{mailboxId}/messages/{messageId}`

Example:

`/mailboxes/e000a1d2-2933-44a6-a126-071ccdb4e090/messages/2ea949fe-ebf3-4035-926c-7dea3dc0c6b5`

The last part of the URL needs to match your message ID on your draft you are updating and the If-Match header must be set to the draft version.

When the put request is done, the message draft is updated with new recipient data, label and the response will look like below with the version number of the draft updated:

```
{
    "id": "2ea949fe-ebf3-4035-926c-7dea3dc0c6b5",
    "version": 1,
    "mailboxId": "e000a1d2-2933-44a6-a126-071ccdb4e090",
    "folderId": "a3fe6924-aac1-46e9-bcb4-1138b14a19a0",
    "transactionId": "EyM2UOwkrnoeJtc2DHMBvCMqx3GGEnG0",
    "createdDateTime": "2021-02-10T13:55:06.914Z",
    "lastUpdated": "2021-02-10T14:27:25.962Z",
    "messageType": "REGULAR",
    "label": "Hejsa Lone Defaultssen",
    "forward": false,
    "reply": false,
    "flag": false,
    "legallyNotified": false,
    "read": true,
    "welcomeMessage": false,
    "errorMessage": false,
    "sender": {
        "id": "64bad6cd-9d26-4b6d-866d-4f689c0e0107",
        "version": 0,
        "senderId": "230792****",
        "senderIdType": "CPR",
        "label": "Lone 2076524521 Defaultssen"
    },
    "recipient": {
        "id": "34abe116-424f-478c-ac21-e583c7415945",
        "version": 1,
        "recipientId": "44556679",
        "recipientIdType": "CVR"
    },
    "replyData": [],
    "documents": [
        {
            "id": "4d253d55-ac00-4332-9b2f-fd4b269a164b",
            "version": 0,
            "documentType": "MAIN",
            "label": "Hoveddokument",
            "files": [
                {
                    "id": "b8509a8f-d454-4cb2-b9de-b6d86061278b",
                    "version": 0,
                    "encodingFormat": "text/html",
                    "filename": "hoveddokument.html",
                    "language": "da"
                }
            ],
            "actions": []
        }
    ]
}
```

6.11 Update a draft with different recipient (Citizen)

In case the draft needs to be updated with a citizen as a recipient, the last 4 digits of the CPR will be masked with **** in the response to the PUT request.

Body of the PUT request with a new not masked CPR :


```
{
    "id": "2ea949fe-ebf3-4035-926c-7dea3dc0c6b5",
    "version": 1,
    "mailboxId": "e000a1d2-2933-44a6-a126-071ccdb4e090",
    "folderId": "a3fe6924-aac1-46e9-bcb4-1138b14a19a0",
    "transactionId": "EyM2UOwkrnoeJtc2DHMBvCMqx3GGEnG0",
    "createdDateTime": "2021-02-10T13:55:06.914Z",
    "lastUpdated": "2021-02-10T14:27:25.962Z",
    "messageType": "REGULAR",
    "label": "Hejsa Lone Defaultssen",
    "forward": false,
    "reply": false,
    "flag": false,
    "legallyNotified": false,
    "read": true,
    "welcomeMessage": false,
    "errorMessage": false,
    "sender": {
        "id": "64bad6cd-9d26-4b6d-866d-4f689c0e0107",
        "version": 0,
        "senderId": "230792****",
        "senderIdType": "CPR",
        "label": "Lone 2076524521 Defaultssen"
    },
    "recipient": {
        "id": "22618fbd-04ff-4e8c-bc71-cfb54ffb5dbe",
        "version": 1,
        "recipientId": "0604982023",
        "recipientIdType": "CPR"
        "label": "Flemming Jensen"
    },
    "replyData": [],
    "documents": [
        {
            "id": "4d253d55-ac00-4332-9b2f-fd4b269a164b",
            "version": 0,
            "documentType": "MAIN",
            "label": "Hoveddokument",
            "files": [
                {
                    "id": "b8509a8f-d454-4cb2-b9de-b6d86061278b",
                    "version": 0,
                    "encodingFormat": "text/html",
                    "filename": "hoveddokument.html",
                    "language": "da"
                }
```

```
            ],
            "actions": []
        }
    ]
}
```

The response of the request has the updated CPR recipient:

```
{
    "id": "2ea949fe-ebf3-4035-926c-7dea3dc0c6b5",
    "version": 2,
    "mailboxId": "e000a1d2-2933-44a6-a126-071ccdb4e090",
    "folderId": "a3fe6924-aac1-46e9-bcb4-1138b14a19a0",
    "transactionId": "EyM2UOwkrnoeJtc2DHMBvCMqx3GGEnG0",
    "createdDateTime": "2021-02-10T13:55:06.914Z",
    "lastUpdated": "2021-02-10T14:27:25.962Z",
    "messageType": "REGULAR",
    "label": "Hejsa Lone Defaultssen",
    "forward": false,
    "reply": false,
    "flag": false,
    "legallyNotified": false,
    "read": true,
    "welcomeMessage": false,
    "errorMessage": false,
    "sender": {
        "id": "64bad6cd-9d26-4b6d-866d-4f689c0e0107",
        "version": 0,
        "senderId": "230792****",
        "senderIdType": "CPR",
        "label": "Lone 2076524521 Defaultssen"
    },
    //THE RECIPIENT IS UPDATED WITH THE CPR
    "recipient": {
        "id": "22618fbd-04ff-4e8c-bc71-cfb54ffb5dbe",
        "version": 1,
        "recipientId": "060498****",
        "recipientIdType": "CPR"
        "label": "Flemming Jensen"
    },
    "replyData": [],
    "documents": [
        {
            "id": "4d253d55-ac00-4332-9b2f-fd4b269a164b",
            "version": 0,
            "documentType": "MAIN",
            "label": "Hoveddokument",
            "files": [
                {
                    "id": "b8509a8f-d454-4cb2-b9de-b6d86061278b",
                    "version": 0,
                    "encodingFormat": "text/html",
```

```
                    "filename": "hoveddokument.html",
                    "language": "da"
                }
            ],
            "actions": []
        }
    ]
}
```

6.12 Update a draft with different masked recipient

In order to update a draft with already existing recipient, it is necessary to provide a not masked value of CPR. If the value is masked in the PUT request, it will be ignored and not updated.

Body of the PUT request with new label and new different masked CPR:

```
{
    "id": "2ea949fe-ebf3-4035-926c-7dea3dc0c6b5",
    "version": 2,
    "mailboxId": "e000a1d2-2933-44a6-a126-071ccdb4e090",
    "folderId": "a3fe6924-aac1-46e9-bcb4-1138b14a19a0",
    "transactionId": "EyM2UOwkrnoeJtc2DHMBvCMqx3GGEnG0",
    "createdDateTime": "2021-02-10T13:55:06.914Z",
    "lastUpdated": "2021-02-10T14:27:25.962Z",
    "messageType": "REGULAR",
    //NEW LABEL VALUE
    "label": "New Label",
    "forward": false,
    "reply": false,
    "flag": false,
    "legallyNotified": false,
    "read": true,
    "welcomeMessage": false,
    "errorMessage": false,
    "sender": {
        "id": "64bad6cd-9d26-4b6d-866d-4f689c0e0107",
        "version": 0,
        "senderId": "230792****",
        "senderIdType": "CPR",
        "label": "Lone 2076524521 Defaultssen"
    },
    //THE NEW RECEIPIENT WITH MASKED VALUE
    "recipient": {
        "id": "22618fbd-04ff-4e8c-bc71-cfb54ffb5dbe",
        "version": 1,
        "recipientId": "121099****",
        "recipientIdType": "CPR"
        "label": "Flemming Jensen"
    },
    "replyData": [],
    "documents": [
```

```
        {
            "id": "4d253d55-ac00-4332-9b2f-fd4b269a164b",
            "version": 0,
            "documentType": "MAIN",
            "label": "Hoveddokument",
            "files": [
                {
                    "id": "b8509a8f-d454-4cb2-b9de-b6d86061278b",
                    "version": 0,
                    "encodingFormat": "text/html",
                    "filename": "hoveddokument.html",
                    "language": "da"
                }
            ],
            "actions": []
        }
    ]
}
```


The response has the newly updated label but the same CPR as before the update:


```
{
    "id": "2ea949fe-ebf3-4035-926c-7dea3dc0c6b5",
    "version": 2,
    "mailboxId": "e000a1d2-2933-44a6-a126-071ccdb4e090",
    "folderId": "a3fe6924-aac1-46e9-bcb4-1138b14a19a0",
    "transactionId": "EyM2UOwkrnoeJtc2DHMBvCMqx3GGEnG0",
    "createdDateTime": "2021-02-10T13:55:06.914Z",
    "lastUpdated": "2021-02-10T14:27:25.962Z",
    "messageType": "REGULAR",
    //LABEL IS UPDATED CORRECTLY
    "label": "New Label",
    "forward": false,
    "reply": false,
    "flag": false,
    "legallyNotified": false,
    "read": true,
    "welcomeMessage": false,
    "errorMessage": false,
    "sender": {
        "id": "64bad6cd-9d26-4b6d-866d-4f689c0e0107",
        "version": 0,
        "senderId": "230792****",
        "senderIdType": "CPR",
        "label": "Lone 2076524521 Defaultssen"
    },
    //RECEIPIENT IS NOT UPDATED
    "recipient": {
        "id": "22618fbd-04ff-4e8c-bc71-cfb54ffb5dbe",
        "version": 1,
        "recipientId": "060498****",
        "recipientIdType": "CPR"
```

```
        "label": "Flemming Jensen"
    },
    "replyData": [],
    "documents": [
        {
            "id": "4d253d55-ac00-4332-9b2f-fd4b269a164b",
            "version": 0,
            "documentType": "MAIN",
            "label": "Hoveddokument",
            "files": [
                {
                    "id": "b8509a8f-d454-4cb2-b9de-b6d86061278b",
                    "version": 0,
                    "encodingFormat": "text/html",
                    "filename": "hoveddokument.html",
                    "language": "da"
                }
            ],
            "actions": []
        }
    ]
}
```


6.12.1 Upload content to a file resource

When writing a draft or replying to a Message, you must add the file content to a File resource present in a Document as part of your Message.

It is done using a PUT multipart request. In the below example using the endpoint:

`PUT /mailboxes/{mailboxId}/messages/{messageId}/documents/{documentId}/files/{fileId}/content`

Example:

`PUT /mailboxes/e7c96848-2e39-4cfa-b10e-3347a68ba022/messages/7ebd24ea-78e8-443d-ad1c-8289d3f50c99/documents/5f74111f-e189-4955-99ec-f44e1a60ed6a/files/27337891-de83-4f59-812c-3964c248a14e/content`

With the content file as body:


```
{
    "id": "7ebd24ea-78e8-443d-ad1c-8289d3f50c99",
    "version": 1,
    "mailboxId": "e7c96848-2e39-4cfa-b10e-3347a68ba022",
    "folderId": "f9d69b81-e0ba-4c5c-b959-f9241677b888",
    "transactionId": "Ex3rp8PRW3oMY4RKE8McXiUF1AT8oFqj",
    "createdDateTime": "2021-01-15T14:17:31.579Z",
    "lastUpdated": "2021-01-15T14:22:39.818Z",
    "messageType": "REGULAR",
    "forward": false,
    "reply": false,
```

```
    "flag": false,
    "legallyNotified": false,
    "read": true,
    "welcomeMessage": false,
    "errorMessage": false,
    "sender": {
        "id": "51d1b490-ec9f-4cf1-98ab-f36927aa42ee",
        "version": 0,
        "senderId": "69832541",
        "senderIdType": "CVR",
        "label": "Mejerby Standardbank"
    },
    "replyData": [],
    "documents": [
        {
            "id": "5f74111f-e189-4955-99ec-f44e1a60ed6a",
            "version": 0,
            "documentType": "MAIN",
            "label": "Hoveddokument",
            "files": [
                {
                    "id": "27337891-de83-4f59-812c-3964c248a14e",
                    "version": 1,
                    "encodingFormat": "text/html",
                    "filename": "hoveddokument.html",
                    "language": "da"
                }
            ],
            "actions": []
        }
    ]
}
```


The If-Match header must be set to the version of the File resource.

The name of the form element must be ‘file’. Here is a cURL example:

```
curl --location --request PUT 'https://test.digitalpost.dk/apis/v1/mailboxes/
e7c96848-2e39-4cfa-b10e-3347a68ba022/messages/7ebd24ea-78e8-443d-ad1c-8289d3f50c99/
documents/5f74111f-e189-4955-99ec-f44e1a60ed6a/files/27337891-
de83-4f59-812c-3964c248a14e/content' \
--header 'If-Match: 1' \
--header 'Authorization: Bearer eyJhbG...xmoiA' \
--form 'file=@"/data/documents/hoveddokument.html"'
```

The response is a structure containing the new version to be used for further PUT requests - for instance for use in auto save.


```
{
    "id": "27337891-de83-4f59-812c-3964c248a14e",
    "version": 2,
    "fileSize": 415
```

```
}
```

The client must NOT base64 encode the file content. The maximum size of the file is 10 MB.

6.13 Automatic data scrubbing of the filename

The filename sent by the user will undergo automatic data scrubbing if it contains any invalid characters. The full list of invalid characters can be found under section *Filename validation*.

As an example the name `file>name.pdf` would automatically have the `>` character removed, resulting in the final file being named `filename.pdf`.

6.13.1 Send message

Messages sent from the mailbox has memoVersion 1.2.

When you want to send a message, you need to send a post request to:

`POST /mailboxes/{mailboxId}/messages/{messageId}/send`

Example:

`POST /mailboxes/e000a1d2-2933-44a6-a126-071ccdb4e090/messages/2ea949fe-ebf3-4035-926c-7dea3dc0c6b5/send`

The first UUID after */mailboxes/* should be your mailbox ID and the UUID after */messages/* should be the ID of the message you want to send.

In this example we want to send this message:


```
{
    "id": "2ea949fe-ebf3-4035-926c-7dea3dc0c6b5",
    "version": 3,
    "mailboxId": "e000a1d2-2933-44a6-a126-071ccdb4e090",
    "folderId": "a3fe6924-aac1-46e9-bcb4-1138b14a19a0",
    "transactionId": "EyMBTksE80UnyX61UQMsN1CoJ8TDn7x7",
    "createdDateTime": "2021-02-10T13:55:06.914Z",
    "lastUpdated": "2021-02-10T15:35:09.140Z",
    "messageType": "REGULAR",
    "label": "Hejsa Lone Defaultssen",
    "forward": false,
    "reply": false,
    "flag": false,
    "legallyNotified": false,
    "read": true,
    "welcomeMessage": false,
    "errorMessage": false,
    "sender": {
        "id": "64bad6cd-9d26-4b6d-866d-4f689c0e0107",
        "version": 0,
        "senderId": "230792****",
        "senderIdType": "CPR",
        "label": "Lone 2076524521 Defaultssen"
```

```
    },
    "recipient": {
        "id": "34abe116-424f-478c-ac21-e583c7415945",
        "version": 1,
        "recipientId": "44556679",
        "recipientIdType": "CVR"
    },
    "replyData": [],
    "documents": [
        {
            "id": "4d253d55-ac00-4332-9b2f-fd4b269a164b",
            "version": 0,
            "documentType": "MAIN",
            "label": "Hoveddokument",
            "files": [
                {
                    "id": "b8509a8f-d454-4cb2-b9de-b6d86061278b",
                    "version": 2,
                    "encodingFormat": "text/html",
                    "filename": "hoveddokument.html",
                    "language": "da"
                }
            ],
            "actions": []
        }
    ]
}
```


And the response we got looks like this:


```
{
    "id": "2ea949fe-ebf3-4035-926c-7dea3dc0c6b5",
    "version": 4,
    "mailboxId": "e000a1d2-2933-44a6-a126-071ccdb4e090",
    "folderId": "6e6654b7-3494-4c94-8b82-6905ff601957",
    "transactionId": "EyMFJModgMQZRievm6gd9vVR5CF5usgJ",
    "createdDateTime": "2021-02-10T13:55:06.914Z",
    "lastUpdated": "2021-02-10T16:04:01.369Z",
    "messageType": "REGULAR",
    "label": "Hejsa Lone Defaultssen",
    "forward": false,
    "reply": false,
    "flag": false,
    "legallyNotified": false,
    "read": true,
    "welcomeMessage": false,
    "errorMessage": false,
    "sender": {
        "id": "64bad6cd-9d26-4b6d-866d-4f689c0e0107",
        "version": 0,
        "senderId": "230792****",
        "senderIdType": "CPR",
```

```
        "label": "Lone 2076524521 Defaultssen"
    },
    "recipient": {
        "id": "34abe116-424f-478c-ac21-e583c7415945",
        "version": 1,
        "recipientId": "44556679",
        "recipientIdType": "CVR"
    },
    "replyData": [],
    "sendDateTime": "2021-02-10T16:04:01.329Z",
    "documents": [
        {
            "id": "4d253d55-ac00-4332-9b2f-fd4b269a164b",
            "version": 0,
            "documentType": "MAIN",
            "label": "Hoveddokument",
            "files": [
                {
                    "id": "b8509a8f-d454-4cb2-b9de-b6d86061278b",
                    "version": 2,
                    "encodingFormat": "text/html",
                    "filename": "hoveddokument.html",
                    "language": "da"
                }
            ],
            "actions": []
        }
    ]
}
```


6.13.2 Reply to message

Endpoint for replying to a message:

`POST /mailboxes/{mailboxId}/messages/{messageId}/reply`

Example:

`POST /mailboxes/8deb553b-0536-4671-9c9e-239f202d56e0/messages/8deb553b-0536-4671-9c9e-239f202d56e0/reply`

Will create a draft message with original sender as recipient, filled replyData, and a main document ready for content. Returns:


```
{
  "id": "912a097a-1556-433b-8a2a-19e2f0c4c514",
  "version": 0,
  "mailboxId": "8deb553b-0536-4671-9c9e-239f202d56e0",
  "folderId": "767442c8-9b93-41f0-8ced-509c397ab934",
  "transactionId": "EpYk1qI5rYy74mjDKtBoQpclcBBMP7ik",
  "createdDateTime": "2020-08-17T17:43:13.799Z",
  "lastUpdated": "2020-08-17T17:43:13.799Z",
```

```
"messageType": "REGULAR",
"label": "Sv: Pladsanvisning",
"forward": false,
"reply": false,
"flag": false,
"legallyNotified": false,
"read": false,
"welcomeMessage": false,
"errorMessage": false,
"sender": {
  "id": "ab7f2b42-1fb6-408d-b582-67c7a14eb96c",
  "version": 0,
  "senderId": "010378****",
  "senderIdType": "CPR"
},
"recipient": {
  "id": "294edf0a-d4c2-438f-b6d4-679cd689950e",
  "version": 0,
  "recipientId": "24586369",
  "recipientIdType": "CVR",
  "label": "Kommunen",
  "attentionData": {
    "id": "e5fc3c88-9c87-4723-8e57-cb8c0d40c2db",
    "version": 0,
    "personId": "9000001234",
    "personLabel": "Hans Hansen",
    "productionUnitNumber": "1234567890",
    "productionUnitName": "Produktionsenhed A",
    "globalLocationNumber": "5798000012345",
    "location": "Kommune A",
    "emailAddress": "info@bornehaven.dk",
    "relatedEmailAgent": "Hans Jensen",
    "seNumber": "24586369",
    "seCompanyName": "Kommune",
    "telephoneNumber": "12345678",
    "relatedTelephoneAgent": "Ib Jensen",
    "ridNumber": "CVR:24586369-RID:1234567890123",
    "ridCompanyName": "Virksomhed",
    "contentResponsibleId": "22334455",
    "contentResponsibleLabel": "Børnehaven, rød stue",
    "generatingSystemId": "Sys-1234",
    "generatingSystemLabel": "KommunaltPostSystem",
    "address": {
      "id": "80c31b0f-1369-474c-8b2f-36a148f89a27",
      "version": 0,
      "addressId": "8c2ea15d-61fb-4ba9-9366-42f8b194c852",
      "addressLabel": "Gaden",
      "houseNumber": "7A",
      "door": "th",
      "floor": "3",
      "co": "C/O",
      "zipCode": "9000",
      "city": "Aalborg",
```

```
        "country": "DK",
        "crsIdentifier": "EPSG:25832",
        "geographicEastingMeasure": "557501.23",
        "geographicNorthingMeasure": "6336248.89",
        "geographicHeightMeasure": "0.0"
      }
    }
  },
  "replyData": [
    {
      "id": "e4128215-19aa-4f24-ae8d-c1a91ecccef6",
      "version": 0,
      "childMessageId": "912a097a-1556-433b-8a2a-19e2f0c4c514",
      "parentMessageId": "8deb553b-0536-4671-9c9e-239f202d56e0"
    }
  ],
  "documents": [
    {
      "id": "f4abea17-ef6c-443d-96f9-928a5ee87a36",
      "version": 0,
      "documentType": "MAIN",
      "label": "Hoveddokument",
      "files": [
        {
          "id": "3efe49c3-28b8-4e01-a41f-d84291b25580",
          "version": 0,
          "encodingFormat": "text/html",
          "filename": "hoveddokument.html",
          "language": "da"
        }
      ],
      "actions": []
    }
  ]
}
```


6.13.3 Update one or more of certain predetermined fields of message (PATCH)

The If-Match header must be set to the version of the Message resource.

`PATCH /mailboxes/{mailboxId}/messages/{messageId}`

Example:

`PATCH /mailboxes/8deb553b-0536-4671-9c9e-239f202d56e0/messages/2ea949fe-ebf3-4035-926c-7dea3dc0c6b5`

Example on body:


```
{
  "folderId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "messageCode": "string",
```

```
  "label": "string",
  "flag": true,
  "legallyNotified": false,
  "read": true,
  "recipient": {
    "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "version": 0,
    .... full recipient with attentiondata, contactpoint etc.
    }
  },
  "note": "string"
}
```

If an element is left out or set to null, then that element is not used in the patch.

```
{
    "flag": true
}
```

The above only sets flag and does not alter any of the other possible values.

```
{
    "flag": true
    "legallyNotified": null,
    "read": null
}
```

The above does the exact same thing - only sets flag does not alter any of the other possible values.

6.13.4 Move message between folders

Lets say we have a message in a mailbox with values:

Mailbox id	8deb553b-0536-4671-9c9e-239f202d56e0
Message id	61768cb7-7b46-4f4f-a980-fc1c8206a6ef
Folder id	3fa85f64-5717-4562-b3fc-2c963f66afa6
New folder id	9f790fc1-7d58-4d5e-9af3-11bce02b2308

The message is moved between folders by updating and switching folderId.

`PATCH /mailboxes/{mailboxId}/messages/{messageId}`

Example:

`PATCH /mailboxes/8deb553b-0536-4671-9c9e-239f202d56e0/messages/61768cb7-7b46-4f4f-a980-fc1c8206a6ef`

With body:

```
{
  "folderId": "9f790fc1-7d58-4d5e-9af3-11bce02b2308"
}
```

There are some rules for moving a message based on the messageState:

- A RECEIVED message cannot be moved to SENT, DRAFTS
- A RECEIVED can be moved to INBOX, DELETED, USER_DEFINED
- A DRAFT message cannot be moved to SENT, INBOX
- A DRAFT can be moved to DRAFTS, DELETED, USER_DEFINED
- A SENT message cannot be moved to INBOX, DRAFTS
- A SENT can be moved to SENT, DELETED, USER_DEFINED

If an invalid move is attempted, the request is considered BAD (error code 400) and the following error message is returned:

```
message.update.folderType.not.allowed
```

6.13.5 Limitations for legal representatives

A legal representative is not allowed to move a message to the DELETED folder, and once a message is in the DELETED folder, a legal representative cannot modify it.

6.13.6 Forward message to e-mail address

> **Note:** Already forwarded messages cannot be forwarded again.

Forwarding is done POSTing a forward command to:

`POST /mailboxes/{mailboxId}/messages/{messageId}/forward`

Example:

`POST /mailboxes/3e3a13df-a2ef-47cb-835c-bd527f96d2a6/messages/cb36a59f-ffef-41e7-b2d9-df136b56f07a/forward`

With body:

```
{
 "recipientId": "Hans@netcompany.com",
 "recipientIdType": "EMAIL",
 "comment": "Hi Hans. Check out this message that I forwarded to your email",
 "senderLabel": "Flemming Jensen at Test Test"
```

```
}
```

The response is a 201 Created with the forwarded message:


```
{
    "id": "1a78610e-ca92-4b43-a263-3c35b387995f",
    "version": 0,
    "mailboxId": "3e3a13df-a2ef-47cb-835c-bd527f96d2a6",
    "folderId": "74c64bac-535f-4c43-b12e-d03b7d48aed9",
    "transactionId": "GEfhJ5jOty1Z9y28ZAm8kWUlqumVr50m",
    "createdDateTime": "2025-06-04T12:08:22.491Z",
    "lastUpdated": "2025-06-04T12:08:22.491Z",
    "messageDateTime": "2025-06-04T12:08:22.473Z",
    "messageType": "REGULAR",
    "messageState": "SENT",
    "memoCreatedDateTime": "2025-06-04T12:08:22.401Z",
    "label": "Vs: Message to be forwarded to email",
    "forward": false,
    "reply": false,
    "flag": false,
    "legallyNotified": false,
    "read": false,
    "welcomeMessage": false,
    "errorMessage": false,
    "sender": {
        "id": "2077efe3-5735-42cf-9941-6d5cf74fb0f2",
        "version": 0,
        "senderId": "77227722",
        "senderIdType": "CVR",
        "label": "Flemming Jensen at Test Test"
    },
    "recipient": {
        "id": "aad754e6-654c-47ac-b166-5672c4c47c2e",
        "version": 0,
        "recipientId": "Hans@netcompany.com",
        "recipientIdType": "EMAIL"
    },
    "documents": [
        {
            "id": "bc4f2301-45c1-4d3d-9b3a-bb0be85dab99",
            "version": 0,
            "documentType": "MAIN",
            "label": "Hoveddokument",
            "files": [
                {
                    "id": "4f9d5d8d-9c6b-460e-92d3-8b9722f6880e",
                    "version": 0,
                    "encodingFormat": "text/html",
                    "filename": "hoveddokument.html",
                    "language": "da",
                    "fileSize": 206
                }
```

```
            ],
            "actions": []
        }
    ],
    "replyData": [
        {
            "id": "812b9e1b-2655-4d0c-9ff7-23d04bf2b90e",
            "version": 0,
            "childMessageId": "b58fb2ba-35fa-4be1-a1f3-61114226ba90",
            "additionalReplyData": []
        }
    ],
  "contentData": {
    "id": "cf0cda8f-96e6-4312-824b-23cbc0aae4d3",
    "version": 0,
    "cprDataCprNumber": "251216****",
    "cprDataName": "Emilie",
    "cvrDataCvrNumber": "5493833",
    "cvrDataCompanyName": "Hansen A/S",
    "motorVehicleLicenseNumber": "HJ93851",
    "motorVehicleChassisNumber": "HJ93851HJ93851HJ93851",
    "propertyNumber": "58J",
    "caseId": "TY-3473652222",
    "caseSystem": "E-journal",
    "kleDataSubjectKey": "874kj7d6d82a",
    "kleDataVersion": "12",
    "kleDataActivityFacet": "Visitation",
    "kleDataLabel": "Pladshenvisning",
    "formDataTaskKey": "83798972311d",
    "formDataVersion": "3",
    "formDataActivityFacet": "B",
    "formDataLabel": "Visitation",
    "productionUnitNumber": 10,
    "productionUnitName": "Afdeling A",
    "educationCode": "AB47236",
    "educationName": "RUC",
    "address": {
      "id": "742a5a26-5190-4559-b5b0-7f5301dfb188",
      "version": 0,
      "addressId": "da1c15bb-f74d-4a26-8617-4fbb5ac4f063",
      "addressLabel": "Søen",
      "houseNumber": "45",
      "door": "tv",
      "floor": "0",
      "co": "AB1",
      "zipCode": "5000",
      "city": "Odense",
      "country": "DK"
    },
    "additionalContentData": [
      {
        "id": "3ddd2c4d-c958-429b-9a92-66eed80b62ec",
        "version": 0,
```

```
        "contentDataType": "En type",
        "contentDataName": "Afdeling",
        "contentDataValue": "Inddrivelse"
      }
    ]
  },
    "forwardData": {
        "id": "aa04b794-dce1-4db1-b203-0cb6754e02ff",
        "version": 0,
        "originalMessageId": "b58fb2ba-35fa-4be1-a1f3-61114226ba90",
        "originalSender": "020690****",
        "originalContentResponsible": "Test Test",
        "comment": "Hi Hans. Check out this message that I forwarded to your email",
        "newMemoId": "ff52c804-4f22-402a-acfa-7f209630246b",
        "originalMessageDateTime": "2025-06-04T11:40:36.255Z"
    },
    "sendDateTime": "2025-06-04T12:08:22.473Z",
    "userActivities": []
}
```


6.13.7 Forward message to trusted recipient and authority

Forwarding is done POSTing a forward command to:

`POST /mailboxes/{mailboxId}/messages/{messageId}/forward`

Example:

`POST /mailboxes/3e3a13df-a2ef-47cb-835c-bd527f96d2a6/messages/b58fb2ba-35fa-4be1-a1f3-61114226ba90/forward`

With body:

```
{
    "recipientId": "0206902025",
    "recipientIdType": "CPR",
    "comment": "Hi CPR. This is a forwarded message from trusted recipient",
    "senderLabel": "77227722 Trusted recipient"
}
```

`recipientId` can be a CVR of a trusted recipient or a Danish authority, in which case `recipientIdType` must be set to CVR.

The response is a 201 Created with the forwarded message in the body:


```
{
    "id": "242c5c5f-608c-4438-9319-f2eac372a009",
    "version": 0,
    "mailboxId": "3e3a13df-a2ef-47cb-835c-bd527f96d2a6",
    "folderId": "74c64bac-535f-4c43-b12e-d03b7d48aed9",
    "transactionId": "GEffEECbxtf4uCfxlzcZ92wex2EkIIKv",
```

```
"createdDateTime": "2025-06-04T11:52:43.059Z",
"lastUpdated": "2025-06-04T11:52:43.059Z",
"messageDateTime": "2025-06-04T11:52:43.047Z",
"messageType": "REGULAR",
"messageState": "SENT",
"memoCreatedDateTime": "2025-06-04T11:52:23.042Z",
"label": "Vs: Message to be forwarded to trusted authority",
"forward": false,
"reply": false,
"flag": false,
"legallyNotified": false,
"read": false,
"welcomeMessage": false,
"errorMessage": false,
"sender": {
    "id": "ba7b8c74-6646-4364-8c52-028f1943e590",
    "version": 0,
    "senderId": "77227722",
    "senderIdType": "CVR",
    "label": "77227722 Trusted recipient"
},
"recipient": {
    "id": "fbad615e-ee25-494a-909f-ad2fb60bc6b6",
    "version": 0,
    "recipientId": "020690****",
    "recipientIdType": "CPR",
    "label": "Test Test"
},
"documents": [
    {
        "id": "a0947bc6-9ad1-4a70-83f9-702902b2b106",
        "version": 0,
        "documentType": "MAIN",
        "label": "Hoveddokument",
        "files": [
            {
                "id": "01a35163-5e98-4846-8f8f-f9a836e31c33",
                "version": 0,
                "encodingFormat": "text/html",
                "filename": "hoveddokument.html",
                "language": "da",
                "fileSize": 206
            }
        ],
        "actions": []
    }
],
"replyData": [
    {
        "id": "7cd672ee-f716-44af-afe4-dc3c3a3290d2",
        "version": 0,
        "childMessageId": "b58fb2ba-35fa-4be1-a1f3-61114226ba90",
        "additionalReplyData": []
```

```
        }
    ],
    "forwardData": {
        "id": "b41625e2-39e4-4443-8d86-7359e69aa694",
        "version": 0,
        "originalMessageId": "b58fb2ba-35fa-4be1-a1f3-61114226ba90",
        "originalSender": "020690****",
        "originalContentResponsible": "Test Test",
        "comment": "Hi CPR. This is a forwarded message from trusted recipient",
        "newMemoId": "3f0f0ed6-4be8-4b51-8bf2-49fec6fbffb4",
        "originalMessageDateTime": "2025-06-04T11:40:36.255Z"
    },
  "contentData": {
    "id": "ceec9eca-e06c-4e0c-8621-7f140b4e087a",
    "version": 0,
    "cprDataCprNumber": "251216****",
    "cprDataName": "Emilie",
    "cvrDataCvrNumber": "5493833",
    "cvrDataCompanyName": "Hansen A/S",
    "motorVehicleLicenseNumber": "HJ93851",
    "motorVehicleChassisNumber": "HJ93851HJ93851HJ93851",
    "propertyNumber": "58J",
    "caseId": "TY-3473652222",
    "caseSystem": "E-journal",
    "kleDataSubjectKey": "874kj7d6d82a",
    "kleDataVersion": "12",
    "kleDataActivityFacet": "Visitation",
    "kleDataLabel": "Pladshenvisning",
    "formDataTaskKey": "83798972311d",
    "formDataVersion": "3",
    "formDataActivityFacet": "B",
    "formDataLabel": "Visitation",
    "productionUnitNumber": 10,
    "productionUnitName": "Afdeling A",
    "educationCode": "AB47236",
    "educationName": "RUC",
    "address": {
      "id": "6be967af-8535-4396-b395-beb824e41ecd",
      "version": 0,
      "addressId": "da1c15bb-f74d-4a26-8617-4fbb5ac4f063",
      "addressLabel": "Søen",
      "houseNumber": "45",
      "door": "tv",
      "floor": "0",
      "co": "AB1",
      "zipCode": "5000",
      "city": "Odense",
      "country": "DK"
    },
    "additionalContentData": [
      {
        "id": "916a9be0-cfe9-41d5-b591-0727199a50a2",
        "version": 0,
```

```
        "contentDataType": "En type",
        "contentDataName": "Afdeling",
        "contentDataValue": "Inddrivelse"
      }
    ]
  },
    "sendDateTime": "2025-06-04T11:52:43.047Z",
    "userActivities": []
}
```

When `recipientIdType` or `senderIdType` are CPRs then the last 4 digits of `recipientId` or `senderId` will be masked with **** in the response to the request. When the original sender is a citizen the fields `forwardData.originalSender` , `forwardData.originalContentResponsible` and `forwardData.originalRepresentative` will be masked in the same way.

6.13.8 ReplyData mail threads

The `replyData` of a message contains the entire reply history of the message.

In these examples the UUID is replaced with a letter to better illustrate the structure.

Say we have a message A, if reply = true, the message is enriched with the message ID and the MeMo ID:

```
{
  "id": "A",
  "memoId": "memo-id-A",
  reply: true,
  "replyData": [
    {
      "chiledMessageId": "A"
    }
  ],
}
```

When A is replied to, the reply, B, looks like this:

```
{
  "id": "B",
  "memoId": "memo-id-B",
  "reply": true,
  "replyData": [
    {
      "chiledMessageId": "A"
    },
    {
      "childMessageId": "B",
      "parentMessageId": "A"
    }
  ]
```

```
}
```

When B is replied to, that reply, C, looks like this:

```
{
  "id": "C",
  "memoId": "memo-id-C",
  "reply": true,
  "replyData": [
    {
      "chiledMessageId": "A"
    },
    {
      "childMessageId": "B",
      "parentMessageId": "A"
    },
    {
      "childMessageId": "C",
      "parentMessageId": "B"
    }
  ]
}
```

From this last message C we can traverse the child/parent relationships from C to B and again from B to A, thus ending up with the root message A. If you want to find all messages of A's mail thread you can query like this:

```
/mailboxes/{mailboxId}/messages/?replyData.parentMessageId=A
```

6.13.9 Write to the authorities

When you need to write a message to the authorities you need to take following steps:

- Create draft message
- Update draft with recipient data(CVR number)
- Upload content to a file resource
- Send message

The first step to take is creating a draft message, this can be done by following the example "create draft message" under "Common use case examples".

The second step is the update the newly created draft with recipient data. There is an example called "Update draft", which can be found under "Common use case examples". The draft should be updated with recipient data, such as CVR number and label.

The third step is to upload content file to a file resource as part of your message. This can be done by following the example "Upload content to a file resource", under "Common use case examples". Beware that the maximum file size of the file may not be greater than 10 MB.

The last step is to send the message. Here too you can find an example and its named "send message" and can be found under "Common use case examples".

6.13.10 Examples of error messages

The error codes and error messages that can be returned from the mailbox are documented in the section “Front-end validation and error codes in the Viewclient”. This page shows a few examples of some of these error messages.

6.14 Uploading invalid html to a file

Read the section “Upload content to a file resource” for description of how to upload bytes.

When adding content to a file, the content is validated as html if the file’s encodingFormat is text/html. The html must adhere to a narrow whitelist of allowed html. Read “HTML whitelist for document validation” for description. If the html added is invalid, the response to the PUT request will be 400 BAD REQUEST with a body like in the example below:

```
{
    "code": "digital.post.error",
    "message": "ValidationException: 61d4c728-353f-4b49-bee8-c26d8fe7addd can not be
updated with html content",
    "fieldErrors": [
        {
            "resource": "target",
            "code": "html.sanitizer.rejected.element.attributes",
            "message": "Elementet \"meta\" indeholder attribut \"http-equiv\" der
ikke accepteres af Digital Post."
        },
        {
            "resource": "target",
            "code": "html.sanitizer.rejected.element",
            "message": "Elementet \"a\" indeholder html der ikke accepteres af
Digital Post."
        }
    ]
}
```

The html file looked like this:

```
<html>
  <head>
    <meta charset="utf-8"/>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
    <meta http-equiv="refresh" content="5"/>
    <title>Test page</title>
  </head>
  <body>
     <h1>Hello Digst</h1>
     <img src="" alt="" />
     <h1 style="background-image:url(&#39;&#39;)">Hello</h1>
    <a href="http://www.dr.dk">Danmarks Radio</a>
  </body>
</html>
```

Error codes beginning with “html.sanitizer.“ are all from the html-validation.

The following error message tells us that `<meta http-equiv="refresh" content="5"/>` is not allowed:

`"message": "Elementet \"meta\" indeholder attribut \"http-equiv\" der ikke accepteres af Digital Post."`

The following error message is not as revealing as the whitelist states:

`"message": "Elementet \"a\" indeholder html der ikke accepteres af Digital Post."`

Links	
a	• href ◦ https, mailto

“http“ is not allowed. It must be “https” in this case.

6.15 Trying to send a message without content in the main document


```
{
    "code": "digital.post.error",
    "message": "ValidationException: Unable to send message",
    "fieldErrors": [
        {
            "resource": "message",
            "field": "documents",
            "code": "message.send.documents.files.content.required",
            "message": "Der skal tilføjes indhold til filen ba9d5d8c-a12b-4a78-b670-
d5cbb2d938f8.",
            "rejectedValue": [
                {
                    "id": "b4403d29-e2d7-4e4c-9cea-7e5edc7bd624",
                    "version": 0,
                    "documentType": "MAIN",
                    "label": "Hoveddokument",
                    "files": [
                        {
                            "id": "ba9d5d8c-a12b-4a78-b670-d5cbb2d938f8",
                            "version": 0,
                            "encodingFormat": "text/html",
                            "filename": "hoveddokument.html",
                            "language": "da"
                        }
                    ],
                    "actions": []
                }
            ]
        }
```

```
    ]
}
```

6.16 Replying to an unreplyable message (reply = false)

```
{
    "code": "digital.post.error",
    "message": "ValidationException: A reply draft can not be created for message
with id 5ef34405-60c4-4995-81c5-7cc3956fa165",
    "fieldErrors": [
        {
            "resource": "message",
            "field": "reply",
            "code": "message.create.reply.notAllowed",
            "message": "Denne meddelse kan ikke besvares.",
            "rejectedValue": false
        }
    ]
}
```

6.16.1 Message state action matrix

The following describes the message state and the actions that are possible within that state.

	Message state		
Action	**RECEIVED**	**DRAFT**	**SENT**
Send	⛔	✅	⛔
Forward[1]	✅	⛔	✅
Reply[2]	✅	⛔	⛔
Move[3]	✅	✅	✅
Delete	✅	✅	✅
Update read/flag	✅	✅	✅
Add or update note	✅	✅	✅

Mark legally notified[4]	✅	⛔	⛔
Add, update or remove sender, recipient, documents, files and file content	⛔	✅	⛔
User Activitives[5]	✅	⛔	✅

1. Forward is only allowed if `message.forward` is true.
2. Reply is only allowed if `message.reply` is true.
3. A RECEIVED message, a DRAFT and a SENT message are all able to be moved to other folders. However, there are exceptions, as to which folders each type can be moved to. See the rules in section *Move message between folders*.
4. `message.legallyNotified` Cannot be changed once switched true.
5. The system can append the two activities of replying to or forwarding a message.

7 Event log services - TI

The types of events exposed are listed in “Event Log Index Events” and the rights needed to access those events are listed in the “Required roles“-column.

Service	URL	Usage	Required roles	Open API
Query event log	`GET /events/`	Used to find information changes related to the caller (sender-system or citizen). Common use cases are; • Finding messages where the delivery failed • Finding information about business or technical receipts • Finding info on who changed contact-structure • Seeing what changes to have made to a resource And many more. The events that are exposed in the event-log are filtered by context so that you can only see relevant events.	• Action Log Admin • Authority Recipient System • Authority Sender System • Citizen • Company Recipient System • Company Sender System • Courts Of Denmark • Delegated Support Admin • Full Power Of Attorney • Message Employee • Message Log Admin • Message Write • Organisation Admin • Organisation User Admin • Recall Admin • Search Log Admin • System Manager	Swagger UI

The Event log by default limits your query to the past 3 week. If you need to alter this behavior you must define a custom range. See section “Default and configurable search interval” for details.

7.1 Querying the event-log

For description of common search functionality, please revisit the section **Querying and searching resources**.

Unlike most of the others services in Digital Post, the event-log only exposes a service that is used to search for events.

The result is a paginated `EventSearchResult`, on the form as shown below. Default results per page is 100.

```
{
  "currentPage": 0,
  "next":
"WyAxNjUxMTQ3MjA0NjgzNDkyLCAiRktLVFR0Y2hzdzViY2tXR24zMXJMWm9TeG1sUXo1Um0iIF0=",
  "totalPages": 100,
```

```
  "elementsOnPage": 100,
  "totalElements": 10000,
  "events": [...]
}
```

7.1.1 Querying

Format

The format for querying the event log followed the common format in Digital Post as shown below:

```
https://api.test.digitalpost.dk/apis/v1/events/?<parameter>=<value>
https://api.test.digitalpost.dk/apis/v1/events/?<parameter>=<value>,<value2>,<value3>
https://api.test.digitalpost.dk/apis/v1/events/?
<parameter>=<value>&<parameter2>=<value2>
```

All of the event-properties are searchable, and can therefore be combined as needed. However since the event-log does not contain CPR or CVR for events from before March 2023, it will convert these back and forth between the identity registry as needed, to avoid the user having to deal with identities. This results in the event-log being less robust for searches using a larger than default pagesize from before that date.

Default and configurable search interval

As default events created over the past 3 weeks will be returned when querying the event log.

If another time interval is needed there can be configured a specific interval using the 'dateFrom' and 'dateTo' parameters with a maximum range of 3 months.

Exampel:

```
/events/?dateFrom=2023-05-11T13:52:16.713Z&dateTo=2023-08-11T13:52:16.713Z
```

Using wildcard

Querying the event log using the wildcard functionality is not supported. Instead, the wildcard characters, `*` and `?`, are interpreted as normal text characters and as such it is still possible to find events containing these characters. For example, the query

```
/events/?subject=BUSINESS*
```

will only return events where the value of “subject” exactly matches “BUSINESS*”.

Examples

Example where we specify `subject=BUSINESS_RECEIPT` and a response size of 2:

```
https://api.test.digitalpost.dk/apis/v1/events/?size=2&subject=BUSINESS_RECEIPT
```

Which results in an example output like this:

```
{
```

```
    "currentPage": 0,
    "next":
"WyAxNjUxMTQ3MjA0NjgzNDkyLCAiRktLVFR0Y2hzdzViY2tXR24zMXJMWm9TeG1sUXo1Um0iIF0=",
    "totalPages": 806,
    "elementsOnPage": 2,
    "totalElements": 1611,
    "events": [
        {
            "id": "be015432-78e8-4eaf-9bf2-c1eb8ed255d7",
            "eventId": "d4fc6350-741b-4894-8b07-b8fdab26cfc4",
            "version": 0,
            "transactionId": "F7Uwy9Erm6DTZbm70Bh4n5yEVXQ6k2KB",
            "channel": "output",
            "channels": [
                "output"
            ],
            "subject": "BUSINESS_RECEIPT",
            "type": "FAILED_REST",
            "rootId": "665311dd-4fad-40f4-8911-78aa79989a97",
            "eventTime": "2021-08-13T22:52:37.143770Z",
            "created": "2021-08-13T22:52:37.706Z",
            "owner": "44556682",
            "actor": "eac1856b-d338-450b-91a5-4fba16fcc893",
            "parentId": "7a08a78e-d51e-4c52-ba8b-b98dcdf6a3a0",
            "system": {
                "id": "02f51669-b18b-4c50-a1a9-2f8d7be8770b",
                "name": "distribution-sender-rest",
                "user": "eac1856b-d338-450b-91a5-4fba16fcc893"
            },
            "message": "2dcc1262-13b0-49eb-aaa9-c9dbb152b56b",
            "eventProperties": {
                "owner": "adfe1341-70da-46dd-8efb-4510be6de280",
                "error-message": "I/O error on POST request for \"https://
test.digitalpost.dk\": Connect to test.digitalpost.dk:443 [test.digitalpost.dk/
10.1.77.102] failed: connect timed out; nested exception is
org.apache.http.conn.ConnectTimeoutException: Connect to test.digitalpost.dk:443
[test.digitalpost.dk/10.1.77.102] failed: connect timed out",
                "recipient-system-endpoint": "https://test.digitalpost.dk"
            },
            "metaProperties": {
                "transmissionId": "2964ef1c-27da-416c-bb8c-3a2dd75957f6",
                "size": "2542",
                "messageType": "DIGITALPOST",
                "senderSystem": "7b8323c5-2883-4623-80d4-57f7bb91d181",
                "sender": "44556682",
                "legalNotification": "false",
                "messageUUID": "2dcc1262-13b0-49eb-aaa9-c9dbb152b56b",
                "recipient": "99881128",
                "title": "Test MeMo built with properties:  longMessage: false,
numberOfAdditionalAttachments: null, reply: false, contactPointId: null,
replyByDateTime: null, action: false, legalNotification: false, mandatory: false",
                "mandatory": "false"
            },
```

```
            "eventTag": "BUSINESS_RECEIPT_SENT"
        },
        {
            "id": "09d014db-ccd8-47a6-a8b1-dace1b458f92",
            "eventId": "f5d9e052-68a9-4b7b-955c-6b3e59cdd226",
            "version": 0,
            "transactionId": "F7Uwx8yFgvsjVteuFmDI4F9CEBQlbeIa",
            "channel": "output",
            "channels": [
                "output"
            ],
            "subject": "BUSINESS_RECEIPT",
            "type": "FAILED_REST",
            "rootId": "4a1ce65f-1d74-48cf-a046-d42a9a41113a",
            "eventTime": "2021-08-13T22:51:36.745203Z",
            "created": "2021-08-13T22:51:37.374Z",
            "owner": "44556682",
            "actor": "eac1856b-d338-450b-91a5-4fba16fcc893",
            "parentId": "1565d9e3-584e-4a24-b395-52410dd821cb",
            "system": {
                "id": "02f51669-b18b-4c50-a1a9-2f8d7be8770b",
                "name": "distribution-sender-rest",
                "user": "eac1856b-d338-450b-91a5-4fba16fcc893"
            },
            "message": "9cce7b32-362a-4b72-a2dd-d42af0f6adba",
            "eventProperties": {
                "owner": "adfe1341-70da-46dd-8efb-4510be6de280",
                "error-message": "I/O error on POST request for \"https://
test.digitalpost.dk\": Connect to test.digitalpost.dk:443 [test.digitalpost.dk/
10.1.77.102] failed: connect timed out; nested exception is
org.apache.http.conn.ConnectTimeoutException: Connect to test.digitalpost.dk:443
[test.digitalpost.dk/10.1.77.102] failed: connect timed out",
                "recipient-system-endpoint": "https://test.digitalpost.dk"
            },
            "metaProperties": {
                "transmissionId": "d2289476-3333-4cd8-a048-309330146b81",
                "size": "2542",
                "messageType": "DIGITALPOST",
                "senderSystem": "7b8323c5-2883-4623-80d4-57f7bb91d181",
                "sender": "44556682",
                "legalNotification": "false",
                "messageUUID": "9cce7b32-362a-4b72-a2dd-d42af0f6adba",
                "recipient": "99881128",
                "title": "Test MeMo built with properties:  longMessage: false,
numberOfAdditionalAttachments: null, reply: false, contactPointId: null,
replyByDateTime: null, action: false, legalNotification: false, mandatory: false",
                "mandatory": "false"
            },
            "eventTag": "BUSINESS_RECEIPT_SENT"
        }
    ]
}
```

7.2 Event Log Index Events

> Events are only retained for a limited period. Each event category and type have their own specified retention period, cf. https://digitaliser.dk/digital-post/nyhedsarkiv/2024/maj/hoering-af-slettefrister-i-digital-post.

The event log persists and enriches events, sent from the other components of Digital Post. These events range from simply describing an update of an object in one of the store components, to more business related cases such as if a legal message has been opened for the first time and therefore is “forkyndt”.

Events from different components looks slightly different, they are however always composed of certain elements:

- `id` : The ID of the event resource as it is stored in the database
- `eventId` : The id of the Event
- `version` : The version of the event, this is currently always 0 as events are immutable
- `transactionId` : The transaction ID linked to the transaction
- `channel(s)` : currently all channels are output due to current implementation, this is an internal data point not relevant for external parties
- `subject` : The subject of the event, these are specific keys used for the logic on whether to persist the event.
- `type` : The type of the event, these are specific keys used for the logic on whether to persist the event.
- `organisationIdentityId` : The id of the owning organisation if the actor is an employee/system
- `eventTime` :Time stamp for when the event happened, always in UTC.
- `created` : Time stamp for when the event was saved to the eventlog, always in UTC.
- `owner` : Owner of the event (or resource that the event references), if the event was done by an employee or a system, the owner will be the CVR of the linked organisation or company, or the CPR of the citizen.
- `ownerIdentityId` : Identity Id of the Owner
- `actor` : Who is responsible for the event. In the example below, an employee sends a message, therefore he is the actor. If the actor is an employee in another organisation, then the CVR returned instead. E.g. if a citizen service employee modifies data that belongs to a citizen, the citizen cannot identify which exact employee did the change only the municipality that the employee is employed by.
- `actorIdentityId` Identity Id of the Actor
- `system` : Systeminformation about the component responsible for sending the event.
- `message` : The message is most often an ID. In this case it is an ID of the message being updated.
- `eventProperties` : The event properties are additional context supplied by the creator of the event, in this case a changelog and the ID of the mailbox that the message is in.
    - always contains a version property as shown in the example (subject to change).
- `metaProperties` : metaProperties are additional information that the event log tries to gather when it persists an event. As a default no addational information is added.
- `searchEventProperties` : The field is used to store information about search parameters in a readable format.
- `eventTag` : A tag describing both the subject and type in a single atribute eg. `DRAFT_SAVED`

Examples of element in event log:

```
{
            "id": "[UUID]",
            "eventId": "[UUID]",
            "version": 0,
            "transactionId": "Fcj9pmGuNI4d6xuO5KXzirQEQNUKMOPQ",
            "channel": "output",
            "channels": [
                "output"
            ],
            "subject": "MEMO",
            "type": "VALIDATED",
            "organisationIdentityId": "[UUID]",
            "eventTime": "2023-05-03T11:22:10.744051Z",
            "created": "2023-05-03T11:22:11.032Z",
            "owner": "[CVR/CPR/UUID/RID]",
            "ownerIdentityId": "[UUID]",
            "actor": "[CVR/CPR/UUID/RID]",
            "actorIdentityId": "[UUID]",
            "system": {
                "id": "[UUID]",
                "name": "distribution-validator-single",
                "user": "[UUID]"
            },
            "message": "[UUID]",
            "eventProperties": {
                "owner": "[UUID]",
                "parent-event-message": "[UUID]#[UUID]"
            },
            "metaProperties": {
                "recipientIdentityId": "[UUID]",
                "numberOfAttachments": "2",
                "memoVersion": "[1.1|1.2]"
                "messageId": "[STRING]",
                "title": "[STRING]",
                "mandatory": "false",
                "senderSystemName": "[STRING]",
                "senderIdentityId": "[UUID]",
                "transmissionId": "[UUID]",
                "recipientType": "CITIZEN",
                "senderContactPointId": "[UUID]",
                "size": "244380",
                "messageType": "DIGITALPOST",
                "senderSystem": "[UUID]",
                "sender": "[CVR/CPR/UUID/RID]",
                "legalNotification": "false",
                "messageUUID": "[UUID]",
                "recipient": "[CVR/CPR/UUID/RID]",
                "senderType": "AUTHORITY",
                "recipientType": "AUTHORITY",
                "contentResponsible": "[STRING/CPR/CVR]",
                "contentResponsibleIdentityId": "[UUID]",
                "representativeId": "[CPR/CVR]",
```

```
            "representativeIdentityId": "[UUID]"
        },
        "searchEventProperties": {},
        "eventTag": "MEMO_SEND_VALIDATED"
    }
```



```
{
        "id": "[UUID]",
        "eventId": "[UUID]",
        "version": 0,
        "transactionId": "Fcj09sKuUILiOsgoPBgBsMANSpimyJzr",
        "channel": "output",
        "channels": [
            "output"
        ],
        "subject": "MESSAGE",
        "type": "UPDATED_DRAFT",
        "eventTime": "2023-05-03T10:09:16.572694Z",
        "created": "2023-05-03T10:09:16.763Z",
        "owner": "[CVR/CPR/UUID/RID]",
        "ownerIdentityId": "[UUID]",
        "actor": "[CVR/CPR/UUID/RID]",
        "actorIdentityId": "[UUID]",
        "system": {
            "id": "[UUID]",
            "name": "mailbox-store",
            "user": "[UUID]"
        },
        "message": "[UUID]",
        "eventProperties": {
            "owner": "[UUID]",
            "mailboxId": "[UUID]",
            "message-id": "[STRING]",
            "memo-id": "[UUID]",
            "version": "5",
            "client_id": "borger-dk-web-post-visningsklient-oidc-demo-id"
        },
        "metaProperties": {
            "memoVersion": "[1.1|1.2]"
            "recipientIdentityId": "[UUID]",
            "messageType": "DIGITALPOST",
            "sender": "[CVR/CPR/UUID/RID]",
            "legalNotification": "false",
            "messageUUID": "[UUID]",
            "recipient": "[CVR/CPR/UUID/RID]",
            "messageId": "[UUID]",
            "title": "[STRING]",
            "mandatory": "false",
            "senderIdentityId": "[UUID]",
            "representativeId": "[CPR/CVR]",
            "representativeIdentityId": "[UUID]"
```

```
        },
        "searchEventProperties": {},
        "eventTag": "DRAFT_SAVED"
    }
```

7.2.1 Fields of interest on events grouped on subject.

If the subject of the event matches, the following properties are attempted added to the metaProperties/ eventProperties/searchEventProperties.

Subject: MEMO:

- eventProperties:
  - parent-event-message
  - nem-sms-recipient (IF NEMsms)
  - recipient (if forwarded to email)
- metaProperties
  - senderContactPointId: ID of sender contact point
  - recipientContactPointId: ID of recipient contact point
  - recipientContactPointName: Name of recipient contact point
  - legalNotification: If the receipt is about a “forkyndelse”
  - mandatory: If the related message was mandatory
  - messageId: MeMo message ID
  - messageUUID: Full ID of the MeMo
  - memoVersion: String of memo version
  - numberOfAttachments: number of attachments
  - recipient: ID of final recipient (fx a citizen)
  - sender: ID of the sender (fx. an organisation)
  - size: In bytes
  - title: Title of the message
  - senderSystem: ID of the sender system
  - senderSystemName: Name of the sender system
  - recipientSystem: ID of the recipient system
  - recipientSystemName: Name of the recipient system
  - senderType: CITIZEN/AUTHORITY/COMPANY
  - recipientType: CITIZEN/AUTHORITY/COMPANY
  - contentResponsibleId: Content responsible in sender in memo
  - contentResponsibleIdentityId: Identity Id of content responsible if cpr/cvr
  - representativeId: Is a cpr or cvr-number from mailbox for full access in memo version 1.2 it's the cpr/ cvr that is sending the message
  - representativeIdentityId: Is identity id of representative-id if available

Subject: MESSAGE:

- eventProperties:
  - nem-sms-recipient (IF NEMsms)
  - recipient
  - owner
  - mailboxId
  - message-id
  - transmission-id
  - memo-id

    - client_id
- metaProperties
    - senderContactPointId: ID of sender contact point
    - recipientContactPointId: ID of recipient contact point
    - recipientContactPointName: Name of recipient contact point
    - legalNotification: If the receipt is about a “forkyndelse”
    - mandatory: If the related message was mandatory
    - messageId: MeMo message ID
    - messageUUID: Full ID of the MeMo
    - numberOfAttachments: number of attachments
    - recipient: ID of final recipient (fx a citizen)
    - sender: ID of the sender (fx. an organisation)
    - size: In bytes
    - title: Title of the message
    - senderSystem: ID of the sender system
    - senderSystemName: Name of the sender system
    - recipientSystem: ID of the recipient system
    - recipientSystemName: Name of the recipient system
    - contentResponsibleId: Content responsible in sender in memo
    - contentResponsibleIdentityId: Identity Id of content responsible if cpr/cvr
    - representativeId: Is a cpr or cvr-number from mailbox for full access in memo version 1.2 it's the cpr/cvr that is sending the message
    - representativeIdentityId: Is identity id of representative-id if available

Subject: TECHNICAL_RECEIPT:

Technical Receipts are only sent for SFTP and SMTP memo messages

- eventProperties:
    - receipt-status
    - mime-subject
    - sender-system-smtp-endpoint
- metaProperties
    - senderSystemSmtpEndpoint
    - messageId
    - title
    - receiptStatus

Subject: BUSINESS_RECEIPT:

- eventProperties:
    - error-message
    - sender-system-id
    - receipt-status
    - failure-id
    - parent-event-type
    - parent-event-subject
    - parent-event-message
    - sender-system-receipt-endpoint
- metaProperties
    - numberOfAttachments
    - legalNotification
    - mandatory
    - messageUUID: Full ID of the MeMo
    - receiptStatus: Status of the receipt

- recipient: ID of final recipient
- sender: ID of the sender (fx. an organisation)
- senderSytemId: ID of the sender system
- senderSystemName: Name of the sender system
- senderSystemReceiptEndpoint
- size: In bytes
- title: Title of the message
- transmissionId: generated uuid, is the same for both technical and business receipt

Subject: MAILBOX :

- eventProperties:
  - mailboxId

Subject: ACCESS :

- eventProperties:
  - mailboxId
  - clientId

Subject: ACCESS_REQUEST :

- eventProperties:
  - privileges
  - privilege_end_date
  - access-to
  - version
  - access-request-type
  - target

Subject: FOLDER :

- eventProperties:
  - owner
  - mailboxId
  - version
  - clientId

Subject: NOTIFICATION :

- eventProperties:
  - owner
  - mailboxId
  - recipient
  - messageId

Subject: REGISTRATION_STATUS :

- eventProperties:
  - owner
  - registration-status:
  - version
  - previous-registration-status

Subject: EMAIL_NOTIFICATION_SUBSCRIPTION/
SMS_NOTIFICATION_SUBSCRIPTION/
PUSH_NOTIFICATION_SUBSCRIPTION:

- eventProperties:
  - owner

- mailboxId
- channel (EMAIL/PHONE OF SUBSCRIPTION)
- version
- client_id
- changelog (if changed)

Subject: NEM_SMS:

- eventProperties:
  - owner
  - changelog (if changed)

Subject: CONTACT:

- eventProperties:
  - owner
  - changelog (if changed)
  - result

Subject: SYSTEM:

- eventProperties:
  - owner
  - changelog (if changed)
  - resourceName
  - organisation (id)
  - version

Subject: CONTACT_GROUP:

- eventProperties:
  - changelog (if changed)
  - resourceName (name of group)
  - organisation (id)
  - parentGroupId
  - version

Subject: CONTACT_POINT:

- eventProperties:
  - owner
  - changelog (if changed)
  - contactGroups (list of id's)
  - organisation (id)
  - resourceName (name of point)
  - version

Subject: EVENT_LOG/CONTACT/PRIVILEGE_GROUP/DIRECT_PRIVILEGE:

searchEventProperties contains the search parameters in a more readable format

- eventProperties:
  - query (if search)
  - result (if search)
- searchEventProperties:
  - query (if search)
  - result (if search)

Subject: CPR/CVR:

- eventProperties:
    - owner
    - firstname-provided (if cpr)
    - lastname-provided (if cpr)

Subject: IDENTITY:

- eventProperties:
    - owner
    - query (if search)
    - result (if search)
    - cpr-changed
    - name-changed
- searchEventProperties:
    - query (if search)
    - result (if search)

Subject: EXEMPTION:

- eventProperties:
    - owner
    - zipCode
    - address1Text
    - address2Text
    - address3Text
    - countryCode
    - name
    - version

Subject: CONSENT:

- eventProperties:
    - owner
    - deviceId
    - client_id

Subject:PRIVILEGES/DELEGATED_SUPPORT_ADMIN_PRIVILEGE:

The changedPrivileges field is a list of the privileges granted/revoked in a readable format

- eventProperties:
    - changelog (if changed)
    - granteeId
    - identityGroupType
- searchEventProperties:
    - changedPrivileges

Subject: SYSTEM_FETCH:

- eventProperties:
    - amount-failed
    - owner
    - to-date
    - from-date
    - system-fetch
    - mailboxId
    - contact-point-id
    - amount-total
    - version

    - client_id
    - amount-fetched

Subject: STATISTICAL_REPORT_SUBSCRIPTION:

- eventProperties:
    - owner
    - name
    - version

Subject: STATISTICAL_REPORT:

- eventProperties:
    - owner
    - reportId
    - subscriptionType
    - name
    - version

For a full comprehensive list of all events stored in the event log, see the file "Events In Event-log.xlsx" at https://digitaliser.dk/digital-post/vejledninger/technical-integration. The Excel sheet also includes a column with the target group for the event, this is to make it clearer which events an end user can expect to see. The schema is a separate file for readability.

8 Push notification integration - TI

8.1 Some general info about push notifications via DP

This section is only relevant for view clients.

There is no explicit (active) way for view clients to send push notifications on behalf of DP. In this document, the automated flow is described, as well as the push notification settings component.

- The backend/developers of a view client (from here on forward referred to as a `Tenant`) sets their `Settings` for Apple's *Apple Push Notification service* (APNs) (see https://developer.apple.com/library/archive/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/PayloadKeyReference.html) and Google's *Firebase Cloud Messaging* (FCM) (see https://firebase.google.com/docs/reference/fcm/rest/v1/projects.messages) describing notification sounds, vibration patterns etc.
- A `Tenant` is a push notification provider that is tied to a Digital Post Identity. It's mostly just used to find the correct `Settings` required to send a push notification.
- `Settings` is a store for the settings that both Google and Apple, respectively, expect *every time* a push notification is sent. Since these settings are expected to rarely change, we store them inside DP so that this info is automatically included in every push notification.

- When a message is received in a mailbox, a flow is triggered based on the message and mailbox that:
  - Fetches the `Settings`
  - Using messageId, mailboxId, deviceId, provider (APNS/ FCM) it then
  - Sends a push notification to the device with device ID 'deviceId' via FCM/APN containing:
    - Message ID
    - Mailbox ID
    - Notification title
    - Notification body
      - (at the time of writing this is assumed to be a canned message akin to "You've received a Digital Post message from <Sender>")
    - Any additional OS specific settings (lights, sounds, colors, priority, etc.)

This means that the active step Tenants have to make, is to set their `Settings` for every app (both iOS and Android. In case of launching a newer app in the future, it would need its own `Settings`).

8.2 Registering as a push notification tenant (aka "I want to send push notifications")

To setup as a tenant, contact DP via your usual service desk.

Include the following information:

- Who you are:
  - Either:
    - CVR number
  - or
    - Organisation ID

> ⚠ **If you as an organisation have more than a single view client we also need to know which one by providing the ClientId of the client.**

- Which app(s) you want to support push notifications for
  - App package
    - e.g. Dk.digst.digital.post.DigDPReaderApp
  - App Name
  - We need to know if you have more than a the default of one iOS app + one Android app
  - ***Note that a single Settings object can include one configuration each for APNs and FCM!***

You will then receive:

- A `Tenant`
  - You will most likely not use this for much directly. As it's just for connecting `Settings` to your Digital Post Identity
- One (or more) `Settings`
  - Use these to specify FCM and APNs settings
    See https://developer.apple.com/library/archive/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/PayloadKeyReference.html
    and https://firebase.google.com/docs/reference/fcm/rest/v1/projects.messages
  - Fields also listed on page "*Push notification settings store model*" under "*Settings json*" in DD120 document.

Example application for a setup where there is "one app" but for both Android and iOS:

> *Please setup Tenant for organisation with id: 03117712-034a-4ccd-a863-c2503304e611*
> *We have two apps:*
> ** MyCoolDigitalPostApp for android.*
> *com.appdeveloper.mycooldigitalpostapp*
>
> ** RadDigitalPostRead for iOs.*
> *com.appdeveloper.raddigitalpostread*

Example application for a setup where there are two APNs apps and an Android app:

> *Please setup Tenant for organisation with id: 03117712-034a-4ccd-a863-c2503304e611*
> *We have three apps:*
> ** MyCoolDigitalPostApp for android*
> *com.appdeveloper.mycooldigitalpostapp*
> ** RadDigitalPostRead for iOS*
> *com.appdeveloper.raddigitalpostread*
> ** 123DigitalPost for iPadOs*
> *com.appdeveloper.123digitalpost*

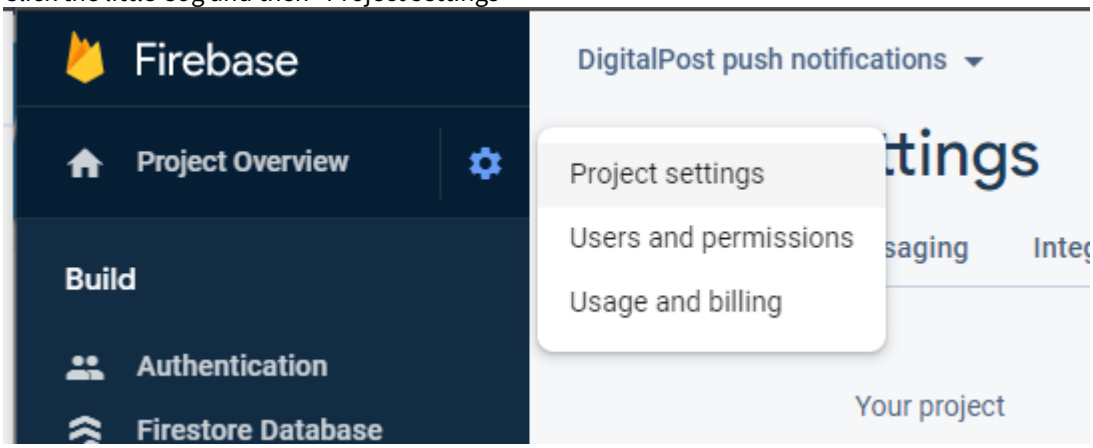
In the latter example two different Settings are required, as they would have different application identifiers for APNs.

8.3 Creating FCM service account + private key

1. If you're from DP, our project is below. If you're a viewclient, simply start at step 2.
   a. If you do not have permissions in our DP ask dvb for them

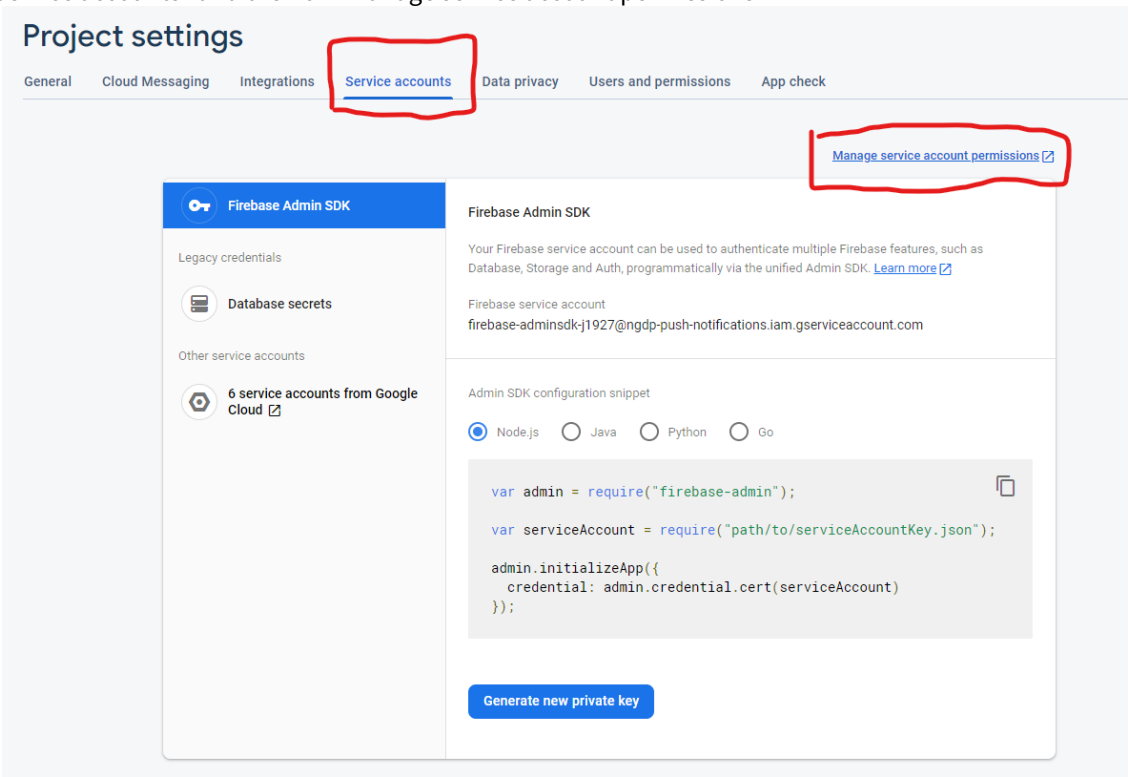
b. Go to https://console.firebase.google.com/project/ngdp-push-notifications/overview

2. Go to Project settings for your FCM app
   a. Click the little Cog and then “Project settings”

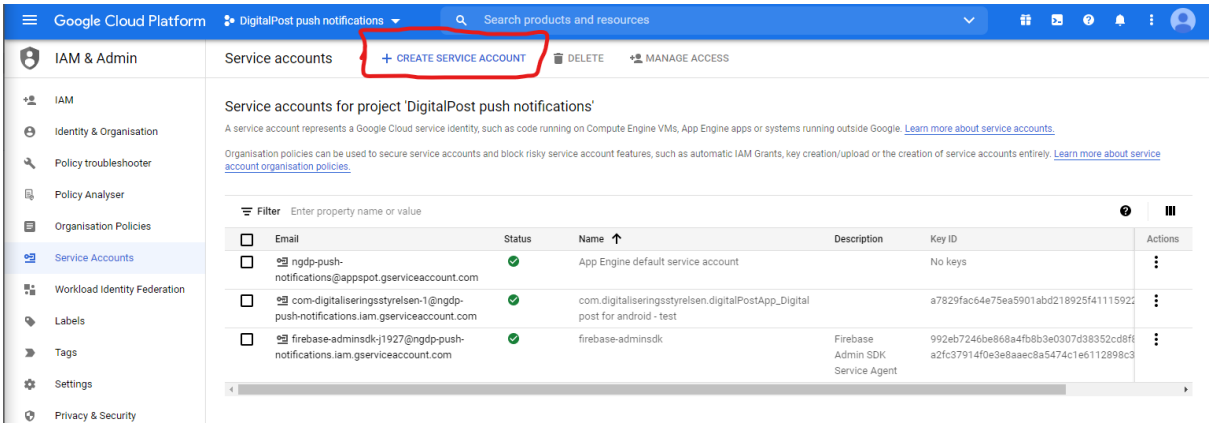


3. Click “Service accounts” and then on “Manage service account permissions”

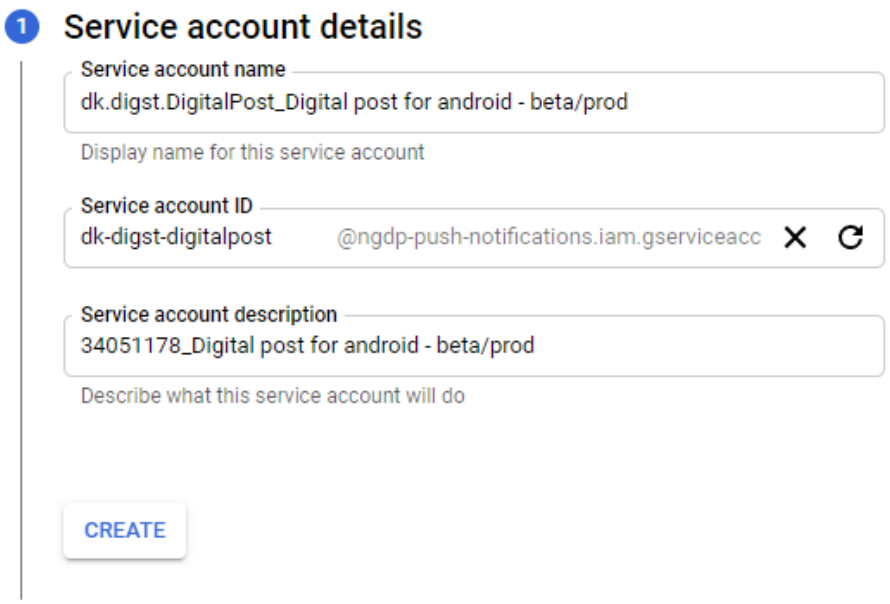
a.



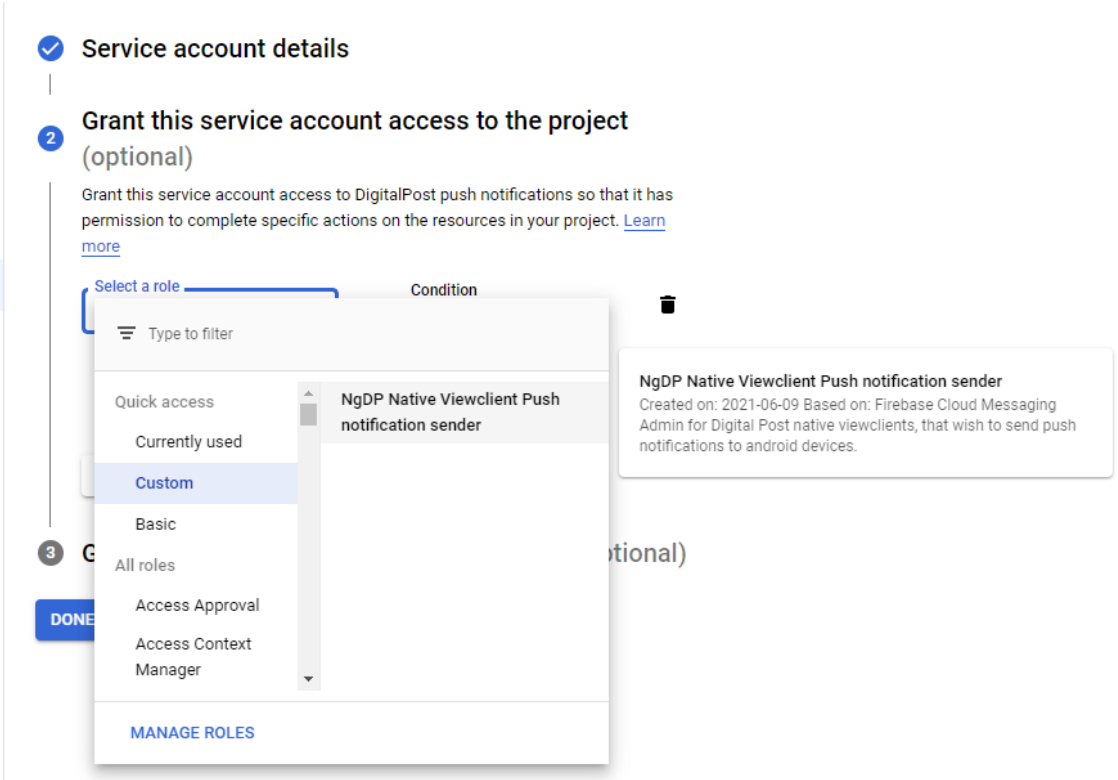
4. Click “Create service account”



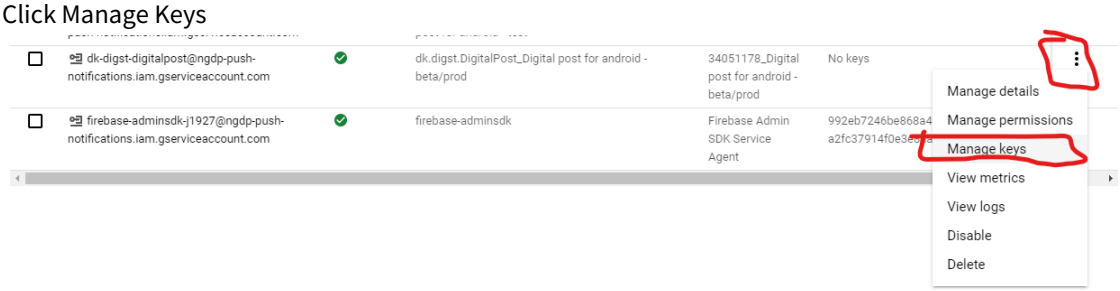
5. For step 1 Fill out the info in the following way:
    a. Service account name:
        i. <package>_<app name>
        ii. Example: dk.digst.DigitalPost_Digital post for android
    b. Service account ID:
        i. <package> (but with dashes instead of dots)
        ii. If possible, use the whole package, otherwise short down the app-specific part and use numbers
        iii. Example: dk-digst-digitalpost
        iv. Example: com-digitaliseringsstyrelsen-1
    c. Service account description
        i. <CVR>_<app name>
        ii. Example: 34051178_Digital post for android - beta/prod



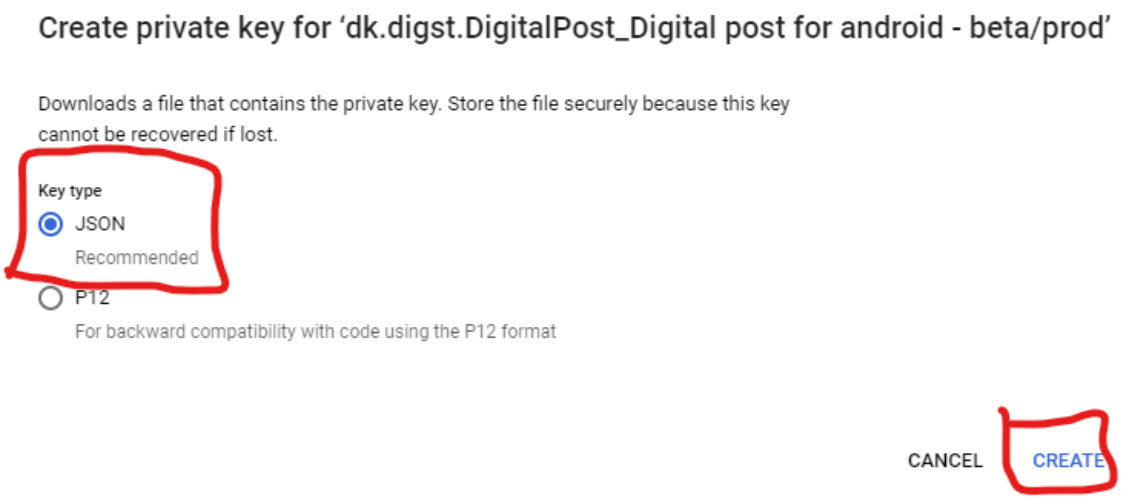
6. For step two of creating the service account, grant it privileges to send push notifications.
    a. For DP FCM project we have setup the role “NgDP Native Viewclient Push notification sender” with these rights.

7. 

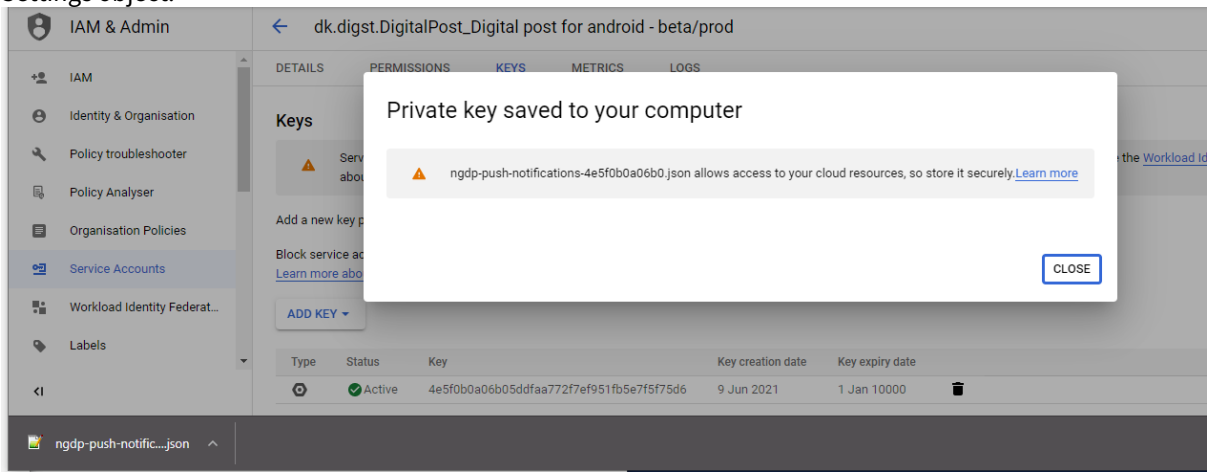
   For step 3 (“Grant users access to this service account (optional)”) **Skip this step**
8. After the service account has been created, add a new key to the account, this key will be what you have to add to the FCM Credentials in the Settings object.
   a. Click the Kebab menu (the 3 dots)
   b. Click Manage Keys



   c. Click Add Key
   d. Click Create new key
   e. Select JSON



9. Done!
   The key has been downloaded to your computer. Add the content of this json to the FCM Credentials in the Settings object.



8.4 Interaction with "push notifications"

After this setup is completed, you can proceed with the following calls.

Get your tenant ID

`/apis/v1/tenants/`

Response:

```
{
    "content": [
        {
            "id": "eb1063f3-e12c-4b29-96f8-e3cb4745357e",
            "version": 0
        }
    ],
```

```
    "number": 0,
    "size": 20
}
```


Get full tenant info

`/apis/v1/tenants/eb1063f3-e12c-4b29-96f8-e3cb4745357e`

Response:


```
{
    "id": "eb1063f3-e12c-4b29-96f8-e3cb4745357e",
    "version": 0,
    "transactionId": "F1mZpUwk6IdykTZxLXTsPs7K0FZVSUrk",
    "tenantName": "Dev01 Test tenant",
    "identityId": "29fb35c5-eb3e-4248-bc50-92df974bdbc6"
}
```


Get list of settings for your tenant

`/apis/v1/settings/`

Response:


```
{
    "content": [
        {
            "id": "6841e177-e795-4d1b-9076-7042b6f32366",
            "version": 0
        }
    ],
    "number": 0,
    "size": 20
}
```


Get full settings

`/apis/v1/settings/6841e177-e795-4d1b-9076-7042b6f32366`

Response (this specific object does not include any APNs settings)

APNs/FCM credentials should be provided by yourself, by getting an APNs/FCM project.

⚠ Note that encoding newlines in fcm.security.credentials as \r\n will mess with the request, which may result in push notifications not working. Instead simply use \n if needed.

⚠ Note that APNs sends private keys in a different format than we expect. (For apn.apnSecurity.privateKey.) This can be fixed with openssl before submitting the private key to your settings object, for example:
`openssl pkcs8 -in AuthKey_432T42ND.p8 -out AuthKey.pem -nocrypt`

Where AuthKey_432T42ND.p8 is a cert issued by APNs, and AuthKey.pem will be a valid private key to upload.


```
{
    "id": "6841e177-e795-4d1b-9076-7042b6f32366",
    "version": 0,
    "transactionId": "F2ZUnp29chA6XA2ILyKktriry5lKNk9H",
    "tenantId": "eb1063f3-e12c-4b29-96f8-e3cb4745357e",
    "instanceId": "5ee1a8d3-c4c3-4430-8fe5-44708f78b1b7",
    "apn": {
        "expiration": "2021-04-20T10:46:37.361Z",
        "sound": {
            "name": "notificationCupcake.caf"
        }
    },
    "fcm": {
        "security": {
            "credentials": "<redacted json object that describes fcm security -
should be included in your original tenant+settings setup request. If not, create a
ticket.>"
        },
        "priority": "HIGH",
        "ttl": 3600000,
        "notification": {
            "defaultSound": true,
            "sticky": true,
            "localOnly": false,
            "priority": "DEFAULT",
            "defaultVibrateTimings": true,
            "vibrateTimings": [
                500,
                500,
                500
            ],
            "notificationCount": 10,
            "defaultLightSettings": true
        }
    }
}
```


Update settings

PUT `/apis/v1/settings/6841e177-e795-4d1b-9076-7042b6f32366`

Body: (change on line 26 from previous result)


```
{
    "id": "6841e177-e795-4d1b-9076-7042b6f32366",
    "version": 0,
    "transactionId": "F2ZUnp29chA6XA2ILyKktriry5lKNk9H",
```

```
    "tenantId": "eb1063f3-e12c-4b29-96f8-e3cb4745357e",
    "instanceId": "5ee1a8d3-c4c3-4430-8fe5-44708f78b1b7",
    "apn": {
        "expiration": "2021-04-20T10:46:37.361Z",
        "sound": {
            "name": "notificationCupcake.caf"
        }
    },
    "fcm": {
        "security": {
            "credentials": "<redacted>"
        },
        "priority": "HIGH",
        "ttl": 3600000,
        "notification": {
            "defaultSound": true,
            "sticky": true,
            "localOnly": false,
            "priority": "DEFAULT",
            "defaultVibrateTimings": true,
            "vibrateTimings": [
                600,
                500,
                500
            ],
            "notificationCount": 10,
            "defaultLightSettings": true
        }
    }
}
```


Response:


```
{
    "id": "6841e177-e795-4d1b-9076-7042b6f32366",
    "version": 1,
    "transactionId": "F2mKSNjPRrKeEBCP3HswE5PgvmSKnZop",
    "tenantId": "eb1063f3-e12c-4b29-96f8-e3cb4745357e",
    "instanceId": "5ee1a8d3-c4c3-4430-8fe5-44708f78b1b7",
    "apn": {
        "expiration": "2021-04-20T10:46:37.361Z",
        "sound": {
            "name": "notificationCupcake.caf"
        }
    },
    "fcm": {
        "security": {
            "credentials": "<redacted>"
        },
        "priority": "HIGH",
        "ttl": 3600000,
        "notification": {
```

```
            "defaultSound": true,
            "sticky": true,
            "localOnly": false,
            "priority": "DEFAULT",
            "defaultVibrateTimings": true,
            "vibrateTimings": [
                600,
                500,
                500
            ],
            "notificationCount": 10,
            "defaultLightSettings": true
        }
    }
}
```

8.5 Subscription to push notification

A push notification subscription is handled in the same way as SMS- and email- notification subscriptions.

To register a mailbox to push notification subscriptions, the app will have to provide the parameters:

- providerType
  - APN for Apple/FCM for Google
- deviceToken
  - which is obtained by the app on the device
- deviceId
- tenantId
  - id of app producer
- instanceId
  - id of the app version
- mailboxId
  - id of the mailbox to register for push notifications

8.5.1 Example

Below is an example of a list of subscriptions. This example mailbox has subscriptions for SMS, e-mail and two push notifications.

`HTTP GET /mailboxes/8deb553b-0536-4671-9c9e-239f202d56e0`

```
{
  "id": "8deb553b-0536-4671-9c9e-239f202d56e0",
  "version": 2,
  "transactionId": "EnIowtB5B5OTHo0GOe5iljXgyaGfXtbO",
  "createdDateTime": "2020-07-03T08:24:14.561Z",
  "lastUpdated": "2020-07-03T08:32:33.155Z",
  "ownerType": "CITIZEN",
  "statusType": "ACTIVE",
  "statusDate": "2020-07-03",
  "recipientSystemAvailable": false,
  "exempt": false,
```

```
  "access": {
    "id": "ce65def2-eb5b-4d4d-86da-4f70ffe9f6e1",
    "version": 3,
    "transactionId": "EnIowtKlvFhKtjc5UJkmsoqiphRf2jyO",
    "createdDateTime": "2020-07-03T08:24:15.230Z",
    "lastUpdated": "2020-07-03T08:32:33.173Z",
    "accessType": "OWNER",
    "mailboxId": "8deb553b-0536-4671-9c9e-239f202d56e0",
    "introductionCompleted": true,
    "smsNotificationSubscription": {
      "id": "8e2b1cf0-1b3c-4db6-950a-663f26209f3d",
      "version": 0,
      "unlistedNumber": false,
      "mobileNumber": "29892630"
    },
    "emailNotificationSubscriptions": [
      {
        "id": "87052e65-67da-4bbc-bbd7-ecd78cfa1928",
        "version": 0,
        "email": "test2@nc.dk"
      }
    ],
    "pushNotificationSubscriptions": [
      {
        "id": "87052e65-67da-4bbc-bbd7-ecd78cfa1928",
        "version": 0,
        "status": "ACTIVE"
        "status_date": "2020-07-10T09:21:51.910Z"
        "providerType": "APN",
        "deviceId": "c7135357-f27b-4c77-b87a-c81567cc4f71",
        "instanceId": "8769de0a-830f-4e13-9a6f-6f757c503862",
        "tenantId": "b062d3ed-a0ec-48c5-ad26-61457b9fd180",
        "deviceToken":
"00fc13adff785122b4ad28809a342098234124142134809787 8e577c991de8f0"
      },
      {
        "id": "e6c6ef98-0765-4723-8f98-cf1957b2a338",
        "version": 0,
        "status": "STALE"
        "status_date": "2021-07-10T09:12:11.250Z"
        "providerType": "FCM"
        "deviceId": "04e2278d-05c8-4346-a3cf-afee406175f3",
        "instanceId": "955f7fd7-14e6-48fe-9541-49075bf25585",
        "tenantId": "b6936f9a-6241-464e-a045-819e311e72cf",
        "deviceToken": "654C4DB3-3F68-4969-8ED2-80EA16B46EB0"
      }
    ],
  }
}
```

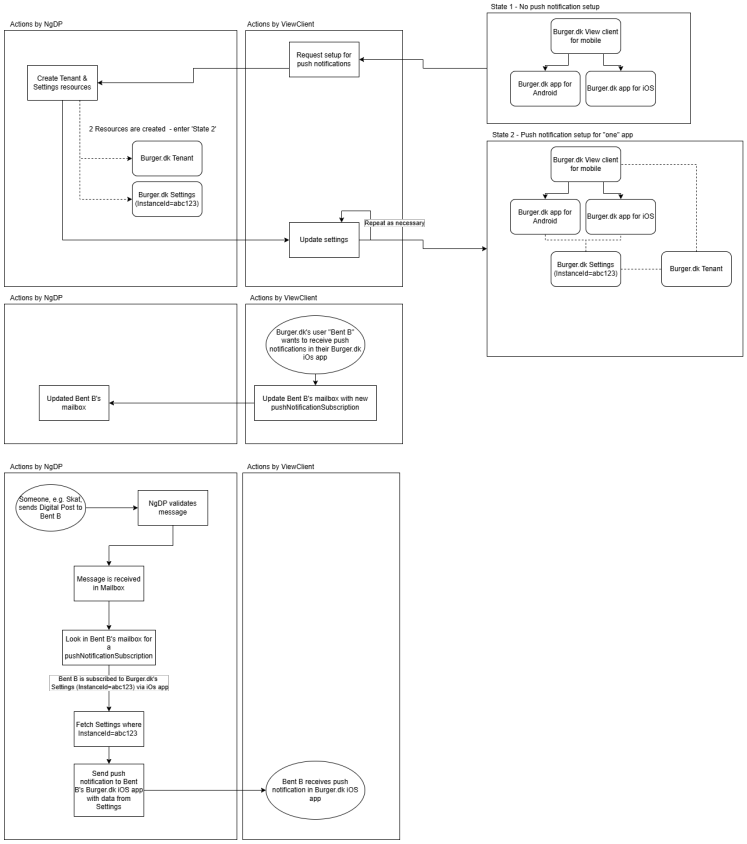

A new push notification subscription is created by adding a new one to the list. An existing is updated by modifying it in the current list. An existing is deleted by removing it from the list.

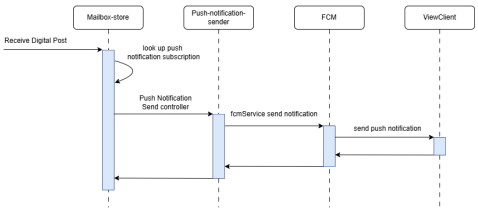
```
PUT /mailboxes/8deb553b-0536-4671-9c9e-239f202d56e0
```

8.6 Visual guide for push notification flows

Here is a visual diagram describing the 3 different flows for push notifications:

- Register as Tenant
- Sign up users for push notifications
- Users receive push notifications when they receive message in their mailbox





9 Identity registry services - TI

The below table shows an overview of all the services that the Identity-registry exposes externally. The table also gives a small description of the common usage patterns that the APIs are intended to support as well as an overview of which roles have permission to call. This overview does not go into details about which Identities the different roles can view or update.

9.1 Identities

Service	URL	Usage	Required roles	Open API
Query identities	`GET /identities/`	Fetching one or multiple Identities by CPR, CVR, type, employee's ID, PID number, SID number, NemLogin's ID and Client ID.	• Citizen • Citizen service employee • Business service employee • Rights administrator • Delegated Support Admin • Support admin • System administrator • Employee • System manager	Swagger UI
Non-caching query of identities	`POST /identities/?isBulkLookup=true`	Fetching one or multiple Identities by CPR, CVR, type, employee's ID, PID number, SID number, NemLogin's ID and Client ID. A request body with specified values (CPR/CVR) is required.	• Citizen • Citizen service employee • Business service employee • Rights administrator • Delegated Support Admin • Support admin • System administrator • Employee • System manager	Swagger UI

Service	URL	Usage	Required roles	Open API
Fetch Identity	`GET /identities/{id}`	Fetching a single identity	• Citizen • Citizen service employee • Business service employee • Rights administrator • Delegated Support Admin • Support admin • System administrator • Employee • System manager	Swagger UI
Update identity	`PUT /identifies/{id}`	Updating the identity, providing a e-mail to user rights administrators or adding alias to employee	• Digital Post rights administrator • System manager	Swagger UI
CPR Validation	`POST /identities/validation/cpr`	Verify the given CPR is matched with the citizen user in the token.	• Citizen	Swagger UI

9.2 Direct privileges

Service	URL	Usage	Required roles	Open API
Query Direct Privilege	`GET /privileges/direct/{grantee-id}`	Used to search Direct privileges of a grantee	• Citizen • Citizen service employee • Business service employee • Rights administrator • Delegated Support Admin • Employee • Support admin • System manager	Swagger UI

Service	URL	Usage	Required roles	Open API
Fetch Direct Privilege	`GET /privileges/direct/{direct-privilege-id}`	Used to fetch the information of a Direct privilege	• Citizen • Citizen service employee • Business service employee • Rights administrator • Delegated Support Admin • Employee • Support admin • System manager	Swagger UI
Create Direct Privilege	`POST /privileges/direct/`	Creating Direct privilege	• Citizen • Citizen service employee • Business service employee • Support admin • Rights administrator • System manager	Swagger UI
Delete Direct Privilege	`DELETE /privileges/direct/{direct-privilege-id}`	Revoke a Direct Privilege	• Citizen • Citizen service employee • Business service employee • Support admin • Rights administrator • System manager	Swagger UI

9.3 Grantees

Service	URL	Usage	Required roles	Open API
Fetch Grantee	`GET identity-groups/{identity_group_id}/grantees/{grantee_id}`	Used to fetch the information of a Grantee	• Citizen • Citizen service employee • Business service employee • Rights administrator • Delegated Support Admin • Employee • Support admin • System manager	Swagger UI

9.4 Identity groups

Service	URL	Usage	Required roles	Open API
Fetch Identity group	`GET /identity-groups/{identity-group-id}`	Fetch identity group information	• Citizen • Citizen service employee • Business service employee • Rights administrator • Delegated Support Admin • Support admin • Employee • System manager	Swagger UI
Create Identity group	`POST /identity-groups`	Creating an Identity group	• Rights administrator	Swagger UI

Service	URL	Usage	Required roles	Open API
Update Identity group	`PUT /identity-groups/{identity-group-id}`	Updating an Identity group	• Citizen • Citizen service employee • Business service employee • Rights administrator • Support admin • System manager	Swagger UI
Delete Identity group	`DELETE /identity-groups/{identity-group-id}`	Deleting an Identity group	• Citizen • Rights administrator	Swagger UI

9.5 Privilege types

Service	URL	Usage	Required roles	Open API
Query privilege type	`GET /privilege-types/?scopeIdentityId={identity-id}&granteeIdentityId={identity-id}`	Get all available privilege types for specific scope and grantee (optional). If the `granteeIdentityId` is not provided, the `scopeIdentityId` must be the id of the parent organisation to which the user is assigned. When querying for privilege-types between employee and parent organisation `granteeIdentityId` **MUST** be omitted.	• Rights administrator • Delegated Support Admin • Legal owner • System manager • Citizen service employee	Swagger UI

9.6 Querying Identities, Direct privileges, Privilege Type

- Identities
  - Fetching identities on ID
- Direct Privileges
  - Fetching Direct privileges on ID
- Identity-group privileges
  - Fetching a specific identity-group
  - Searching for a specific grantee or scope id
- Privilege Types
  - Fetching privilege types with scopeIdentityId and granteeIdentityId:
  - Fetching internal privilege types:

9.7 Identities

For a description of common search functionality, please revisit the section **Querying and searching resources** as well as the https://test.digitalpost.dk/api/swagger-ui/index.html?validatorUrl=none&url=/api/current/json#/ specification.

The following endpoint has been exposed externally from the Identity registry:

GET `/identities/`

- This queries all identities the user is allowed to see.
  Example: Citizen service employees can see the identities of citizens and companies.

The result is an `IdentitySearchResult`, which looks like this in JSON:

```
{
    "currentPage": 0,
    "totalPages": 1,
    "elementsOnPage": 1,
    "totalElements": 1,
    "identities": []
}
```

9.7.1 Fetching identities on ID

```
GET /identities/0b723a6e-c32f-42b6-a124-a79c2cb7d599
```

will fetch the identity with ID `0b723a6e-c32f-42b6-a124-a79c2cb7d599` (Random UUID with no actual identity behind it).

Searching for Identities

Generally, the format is:

```
GET /identities/?<parameter>=<value>
GET /identities/?<parameter>=<value>,<value>,<value>
GET /identities/?<parameter>=<value>&<parameter>=<value>&<parameter>=<value>
```

It's possible to mix and match different parameters. For lines 2 and 3, it's possible to add as many parameters/values as desired.
Example:

```
GET /identities/?cvrNumber=1234567890
```

returns the identity with CVR number 1234567890

```
GET /identities/?cvrNumber=1234567890,2345678901
```

would get us the two identities with the given CVR numbers.

Searching for identities using POST with body

In addition to using GET for searching for identities it is possible to search for a single identity or group of identities using:

```
POST /identities/?isBulkLookup=true
```

and with a body consisting of several (e.g. a list) values in the case of ids or cvr numbers, or of either a single cpr number or several as seen in the respective examples here:

```
{
  "cprNumber": "0204600202"
}
```

or

```
{
  "cprNumber": ["0204600202", "1512440289"]
}
```

If `isBulkLookup` is set to anything but `true` or if the body is empty a bad request message results.

9.8 Direct Privileges

9.8.1 **Fetching Direct privileges on ID**

```
GET /privileges/direct/c77d2e47-bb5d-410d-8ff7-e08c1f971c54
```

will fetch the direct privilege with ID `c77d2e47-bb5d-410d-8ff7-e08c1f971c54` (Random UUID with no actual direct privilege behind it).

Endpoint for creation of Direct privileges:

```
POST /privileges/direct/
```

Example:

```
{
            "granteeId": "2a1e9c38-0fbc-47ac-ab8c-a5ca213e2405",
            "issuerId": "a58e5129-8c21-463f-8355-008eab2957e3",
            "scopeId": "a58e5129-8c21-463f-8355-008eab2957e3",
            "source": "SELF_SERVICE",
            "type": "CITIZEN"
}
```

Create a Direct privilege and Identity group (if not exist)


```
{
    "createdDate": "2021-07-13T06:48:25.063Z",
    "granteeId": "2a1e9c38-0fbc-47ac-ab8c-a5ca213e2405",
    "id": "e009a294-e6cf-4b40-8b80-1eab13f42863",
    "identityGroupId": "9666a80a-cb7b-4289-9de6-c71f810c46e3",
    "issuerId": "a58e5129-8c21-463f-8355-008eab2957e3",
    "lastUpdated": "2021-07-13T06:48:25.063Z",
    "scopeId": "a58e5129-8c21-463f-8355-008eab2957e3",
    "source": "SELF_SERVICE",
    "type": "CITIZEN",
    "version": 0
}
```


9.9 Identity-group privileges

9.9.1 Fetching a specific identity-group

```
GET /identity-groups/UUID
```

This endpoint will fetch the identity-group with the specified UUID, which includes all the grantees and all the privileges

Example


```
{
    "id": "cfd256f3-1ca8-4cdf-befb-16bf9ee2762a",
    "version": 3413,
    "name": "DEFAULT",
    "transactionId": "GICacDnaMoprjmevbx0y7rdwgucP0jzB",
    "createdDate": "2022-03-15T07:53:22.691Z",
    "lastUpdated": "2025-08-14T12:03:04.633Z",
    "issuerId": "516dbeb4-934f-46f7-82ae-79dd159eadcb",
    "ownerId": "516dbeb4-934f-46f7-82ae-79dd159eadcb",
```

```
    "grantees": [
        {
            "id": "2958d254-5425-49a4-9042-c5cb384079f0",
            "version": 0,
            "identityGroupId": "cfd256f3-1ca8-4cdf-befb-16bf9ee2762a",
            "identityId": "516dbeb4-934f-46f7-82ae-79dd159eadcb",
            "issuerId": "516dbeb4-934f-46f7-82ae-79dd159eadcb",
            "createdDate": "2022-03-15T07:53:22.691Z",
            "lastUpdated": "2022-03-15T07:53:22.691Z"
        }
    ],
    "privileges": [
        {
            "id": "7891798b-b649-4e7b-adac-8ef3b8bd980f",
            "version": 0,
            "identityGroupId": "cfd256f3-1ca8-4cdf-befb-16bf9ee2762a",
            "issuerId": "7bf6dcab-2257-4cf0-acdb-243be2e2be03",
            "scopeId": "0b467ca4-9127-4085-a407-ef266e611f24",
            "type": "ORGANISATION_ADMINISTRATOR",
            "source": "APPOINTED",
            "createdDate": "2024-05-24T08:18:11.073Z",
            "lastUpdated": "2024-05-24T08:18:11.073Z"
        }
      ],
     "type": "DEFAULT"
}
```


9.9.2 Searching for a specific grantee or scope id

It is possible to extract all identity-groups that has a grantee with the specified identity ID, or all identity-groups that have privileges associated with it with a specified scope identity ID

Search query for fetching identity-groups where grantee has specified identity ID

```
GET /identity-groups/?grantees.identityId=UUID
```

Example


```
{
    "currentPage": 0,
    "next":
"WyAxNzU1MTcyOTg0NjMzLCAiR0lDYWNEbmFNb3Byam1ldmJ4MHk3cmR3Z3VjUDBqekIiIF0=",
    "totalPages": 1,
    "elementsOnPage": 1,
    "totalElements": 1,
    "identityGroups": [
        {
            "id": "cfd256f3-1ca8-4cdf-befb-16bf9ee2762a",
            "version": 3413,
            "name": "DEFAULT",
            "transactionId": "GICacDnaMoprjmevbx0y7rdwgucP0jzB",
```

```
            "createdDate": "2022-03-15T07:53:22.691Z",
            "lastUpdated": "2025-08-14T12:03:04.633Z",
            "issuerId": "516dbeb4-934f-46f7-82ae-79dd159eadcb",
            "ownerId": "516dbeb4-934f-46f7-82ae-79dd159eadcb",
            "grantees": [
                {
                    "id": "2958d254-5425-49a4-9042-c5cb384079f0",
                    "version": 0,
                    "identityGroupId": "cfd256f3-1ca8-4cdf-befb-16bf9ee2762a",
                    "identityId": "516dbeb4-934f-46f7-82ae-79dd159eadcb",
                    "issuerId": "516dbeb4-934f-46f7-82ae-79dd159eadcb",
                    "createdDate": "2022-03-15T07:53:22.691Z",
                    "lastUpdated": "2022-03-15T07:53:22.691Z"
                }
            ],
            "privileges": [
                {
                    "id": "751c81b3-72e3-4534-a8f0-e5edaaca019e",
                    "version": 0,
                    "identityGroupId": "cfd256f3-1ca8-4cdf-befb-16bf9ee2762a",
                    "issuerId": "7bf6dcab-2257-4cf0-acdb-243be2e2be03",
                    "scopeId": "ee0808dd-076b-486d-8661-c06c26a94309",
                    "type": "ORGANISATION_ADMINISTRATOR",
                    "source": "APPOINTED",
                    "createdDate": "2024-05-24T08:46:48.469Z",
                    "lastUpdated": "2024-05-24T08:46:48.469Z"
                }
              ],
            "type": "DEFAULT"
        }
    ]
}
```


Search query for fetching identity-groups where privilege has specified scope identity ID

```
GET /identity-groups/?privileges.scopeId=UUID
```

Example


```
{
    "currentPage": 0,
    "next":
"WyAxNzM0MzQ5NzYzOTQwLCAiRzZEV1FDSG9lT0lJMVpOaWxPMjE5bkJkdmZ4RTB4V00iIF0=",
    "totalPages": 7,
    "elementsOnPage": 100,
    "totalElements": 655,
    "identityGroups": [
        {
            "id": "571bd62d-b723-4b1b-8451-106a018173e7",
            "version": 38,
            "name": "DEFAULT",
            "transactionId": "GJUiakBspWsyCZ87SDjstwGwLLfEwd6V",
```

```
            "createdDate": "2022-02-21T11:00:42.891Z",
            "lastUpdated": "2025-09-09T11:47:35.174Z",
            "issuerId": "22060615-9e77-4122-bd4a-8ef82d0ad782",
            "ownerId": "22060615-9e77-4122-bd4a-8ef82d0ad782",
            "parentOwnerId": "516dbeb4-934f-46f7-82ae-79dd159eadcb",
            "grantees": [
                {
                    "id": "df63459e-8bb4-4e00-af69-eea45a47c386",
                    "version": 0,
                    "identityGroupId": "571bd62d-b723-4b1b-8451-106a018173e7",
                    "identityId": "22060615-9e77-4122-bd4a-8ef82d0ad782",
                    "issuerId": "22060615-9e77-4122-bd4a-8ef82d0ad782",
                    "createdDate": "2022-02-21T11:00:42.911Z",
                    "lastUpdated": "2022-02-21T11:00:42.911Z"
                }
            ],
            "privileges": [
                {
                    "id": "0fc9b74f-c124-4a24-aba9-2fe7d40d8605",
                    "version": 0,
                    "identityGroupId": "571bd62d-b723-4b1b-8451-106a018173e7",
                    "issuerId": "516dbeb4-934f-46f7-82ae-79dd159eadcb",
                    "scopeId": "21e775c7-d6ee-4d18-b4a4-e5384c8048c3",
                    "type": "DELEGATE_SYSTEM",
                    "source": "SELF_SERVICE",
                    "createdDate": "2025-09-09T11:47:35.174Z",
                    "lastUpdated": "2025-09-09T11:47:35.174Z"
                }
              ],
            "type": "DEFAULT"
        }
    ]
}
```


9.10 Privilege Types

When fetching privilege type a scopeIdentityId is required, while the granteeIdentityId is optional. When querying for privileges available to an employee, the granteeIdentityId parameter must be omitted entirely (not set to null or empty).

9.10.1 Fetching privilege types with scopeIdentityId and granteeIdentityId:

Endpoint for querying of privilege type for specific scope and grantee:

```
GET /privilege-types/?scopeIdentityId=c77d2e47-bb5d-410d-8ff7-
e08c1f971c54&granteeIdentityId=0b723a6e-c32f-42b6-a124-a79c2cb7d599
```

Example reponse:


```
[
```

```
    "MESSAGE_WRITE",
    "ORGANISATION_ADMINISTRATOR",
    "MESSAGE_EMPLOYEE",
    "ACTION_LOG_ADMINISTRATOR",
    "SEARCH_LOG_ADMINISTRATOR",
    "MESSAGE_LOG_ADMINISTRATOR",
    "STATISTICS_ADMINISTRATOR",
    "MESSAGE_BASIC",
    "CONTACT_ADMINISTRATOR",
    "ORGANISATION_USER_ADMINISTRATOR"

]
```

9.10.2 Fetching internal privilege types:

Endpoint for querying privilege type with only scopeIdentityId (internal privileges):

> **Important:** This endpoint returns only internal privileges - those available to employees within the organization they belong to. The scopeIdentityId must correspond to the company of the user's parent organization.

```
GET /privilege-types/?scopeIdentityId=c77d2e47-bb5d-410d-8ff7-e08c1f971c54
```

Example response:

```
[
    "MESSAGE_WRITE",
    "ORGANISATION_ADMINISTRATOR",
    "MESSAGE_EMPLOYEE",
    "ACTION_LOG_ADMINISTRATOR",
    "SEARCH_LOG_ADMINISTRATOR",
    "MESSAGE_LOG_ADMINISTRATOR",
    "STATISTICS_ADMINISTRATOR",
    "MESSAGE_BASIC",
    "CONTACT_ADMINISTRATOR",
    "ORGANISATION_USER_ADMINISTRATOR"

]
```

9.11 Direct privilege

A direct privilege is expected to be a unique combination of; issuer, privilege-type, scope and grantee - the before-mentioned combination is most likely enough to unambiguously identity and revoke a privilege.

A direct privilege is between the following identities:

Tildeler (Issuer)	Recipient (Grantee)
Citizen	Company
Citizen	Citizen
Company	Company
Company	Employee

A Rights Administrator in a company can see all the company's proxies assigned to other identities (other companies, employees in other companies, or citizens). Likewise, he/she can also see all the proxies that have been assigned to my company or an employee in my company.

The employee can see the privileges that are granted to and from their company.
The employee can see only the privileges **only** of their company.

Citizens may only see themselves, as well as the power of attorney relationships. Power of attorney both ways - from and to a citizen.

9.12 Creating Direct Privilege

The creating endpoint will create a direct privilege and an identity group in which the grantee is an owner. The identity group is created only once the first time, the next direct privilege will be assigned to the existing group.

Endpoint for creation of Direct privileges:

```
POST /privileges/direct/
```

Example:

POST `privileges/direct/`

```
{
            "granteeId": "2a1e9c38-0fbc-47ac-ab8c-a5ca213e2405",
            "issuerId": "a58e5129-8c21-463f-8355-008eab2957e3",
            "scopeId": "a58e5129-8c21-463f-8355-008eab2957e3",
            "source": "SELF_SERVICE",
            "type": "CITIZEN"
}
```

After we create the privilege, this is the response we get:

```
{
    "createdDate": "2021-07-13T06:48:25.063Z",
    "granteeId": "2a1e9c38-0fbc-47ac-ab8c-a5ca213e2405",
    "id": "e009a294-e6cf-4b40-8b80-1eab13f42863",
    "identityGroupId": "9666a80a-cb7b-4289-9de6-c71f810c46e3",
```

```
    "issuerId": "a58e5129-8c21-463f-8355-008eab2957e3",
    "lastUpdated": "2021-07-13T06:48:25.063Z",
    "scopeId": "a58e5129-8c21-463f-8355-008eab2957e3",
    "source": "SELF_SERVICE",
    "type": "CITIZEN",
    "version": 0
}
```


9.13 Privilege group

With the introduction of privilege groups, the number of privileges an identity can be granted expands.
I.e. an identity inherits privileges assigned to the privilege group that the identity is included in. Privilege groups grant a user one or more privileges. A privilege is equivalent to having a specific role in the context of a particular company / authority.

There are the following scenarios:

- A user (identity) must be able to see an overview of all privileges:
  - Assigned (user is *grantee*)
  - Distributed (user is *issuer*)
- An administrator must be able to view and manage privilege groups and associated members (identities) and the group's assigned privileges.

9.14 Querying the Privilege Group

Querying privilege groups are done using a GET request to the `/identity-groups/` endpoint.

The result is an `IdentityGroupSearchResult` , which looks like this in JSON:


```
{
  "currentPage": 0,
  "totalPages": 1,
  "elementsOnPage": 1,
  "totalElements": 1,
  "identityGroups": []
}
```


9.14.1 Fetching

Getting a privilege group can be done using a GET request to the `/identity-groups/{id}` endpoint.

The result is the privilege group with the specified ID.

Let us assume that we want to find the privilege group that has the id: `7ee17165-c961-4b61-8212-1b980ae2294f`

GET `https://api.digitalpost.dk/apis/v1/identity-groups/7ee17165-c961-4b61-8212-1b980ae2294f`

Which gives us the following response:

```
{
    "id": "7ee17165-c961-4b61-8212-1b980ae2294f",
    "version": 2,
    "name": "DELEGATED_PRIVILEGE_ed84499b-ef7b-4dc4-8bcd-e31f1c875bfd",
    "transactionId": "F3f3fUzjEORcPKuOIM1rwET9OZoYEevJ",
    "createdDate": "2021-05-28T07:37:30.273Z",
    "lastUpdated": "2021-05-28T07:37:30.371Z",
    "issuerId": "ed84499b-ef7b-4dc4-8bcd-e31f1c875bfd",
    "ownerId": "29ed649b-1a62-44db-8203-9000d8d06596",
    "grantees": [
        {
            "id": "ebb5a669-f9bd-4dfe-933b-7fd3a7be1834",
            "version": 0,
            "identityGroupId": "7ee17165-c961-4b61-8212-1b980ae2294f",
            "identityId": "6e61749d-3b2d-4353-be57-76142ca38342",
            "issuerId": "ed84499b-ef7b-4dc4-8bcd-e31f1c875bfd",
            "createdDate": "2021-05-28T07:37:30.350Z",
            "lastUpdated": "2021-05-28T07:37:30.350Z"
        }
    ],
    "privileges": [
        {
            "id": "eb0c922a-6975-42c9-af93-bdf4ddb01622",
            "version": 0,
            "identityGroupId": "7ee17165-c961-4b61-8212-1b980ae2294f",
            "issuerId": "ed84499b-ef7b-4dc4-8bcd-e31f1c875bfd",
            "scopeId": "ed84499b-ef7b-4dc4-8bcd-e31f1c875bfd",
            "type": "COMPANY_SENDER_SYSTEM",
            "source": "SELF_SERVICE",
            "createdDate": "2021-05-28T07:37:30.367Z",
            "lastUpdated": "2021-05-28T07:37:30.367Z"
        }
    ],
    "type": "DEFAULT"
}
```


9.14.2 Searching

Besides the functionality described above, the Privilege group overrides and offers search using the following parameters:

List of owner	List of identity IDs of owners of the privilege group.
List of issuer	List of identity IDs of the issuer of the privilege.
List of grantee	List of identity IDs of grantees of the privilege.
List of scope	List of identity IDs of who are delegated by the issuer.

Generally, the format is:

```
/identity-groups/?<parameter>=<value>
/identity-groups/?<parameter>=<value>,<value>,<value>
/identity-groups/?<parameter>=<value>&<parameter>=<value>&<parameter>=<value>
```

It is possible to mix and match different parameters. For lines 2 and 3, it is possible to add as many parameters/values as desired.

Examples:

```
/identity-groups/?scope=576f90c8-aecd-492c-9d41-cae3db5c2fe7
```

returns the privilege group/s has scope ID is `576f90c8-aecd-492c-9d41-cae3db5c2fe7`

```
/identity-groups/?scope=576f90c8-aecd-492c-9d41-
cae3db5c2fe7,935877f7-3379-463f-9c75-8fbd715e3702
```

would get us the privilege group/s with the given scope Ids.

9.14.3 Creating

Endpoint for creation of identity-group: POST `/identity-groups/`

Example:

POST `/identity-groups/`

```
{
    "name": "TEST GROUP"
}
```

This creates an Identity group:

```
{
    "id": "28e35a75-857f-490e-a9e3-3be807bc34fb",
    "version": 0,
    "name": "TEST GROUP",
    "transactionId": "F5x1bes9QnqjG9LVrXLilVK7I1MJukiQ",
    "createdDate": "2021-07-13T08:43:49.502Z",
    "lastUpdated": "2021-07-13T08:43:49.502Z",
    "issuerId": "387cdf11-59bf-4016-8afe-be0e0de7e45e",
    "ownerId": "387cdf11-59bf-4016-8afe-be0e0de7e45e",
    "grantees": [],
    "privileges": [],
    "type": "MANUAL"
}
```

9.14.4 Updating

Endpoint for updating an identity-group: PUT `/identity-groups/{id}`.

When updating the `IdentityGroup` through this endpoint, we must first fetch the specific resource (id of the identity-group). Since the search index in only eventually consisted, we must first do a fetch of the resource that we want to update.

```
GET /identity-groups/28e35a75-857f-490e-a9e3-3be807bc34fb
```

Which gives:


```
{
    "id": "28e35a75-857f-490e-a9e3-3be807bc34fb",
    "version": 0,
    "name": "TEST GROUP",
    "transactionId": "F5x1bes9QnqjG9LVrXLilVK7I1MJukiQ",
    "createdDate": "2021-07-13T08:43:49.502Z",
    "lastUpdated": "2021-07-13T08:43:49.502Z",
    "issuerId": "387cdf11-59bf-4016-8afe-be0e0de7e45e",
    "ownerId": "387cdf11-59bf-4016-8afe-be0e0de7e45e",
    "grantees": [],
    "privileges": [],
    "type": "MANUAL"
}
```


Now we can change the properties that we want to. Note that `grantees` and `privileges` are sub-resources, which means they have to be changed separately, e.g. `/identity-groups/{id}/grantees/`. In this example we change the name of the group like so:

```
PUT /identity-groups/28e35a75-857f-490e-a9e3-3be807bc34fb
```

With this request body:

```
{
    "name": "GROUP_V1"
}
```

Note that we must also include the precondition headers.

If the request is successful, the endpoint will return the updated group in the response body:


```
{
    "id": "28e35a75-857f-490e-a9e3-3be807bc34fb",
    "version": 1,
    "name": "GROUP_V1",
    "transactionId": "F5x2Ytx8TYc1uwzxHrnq4Dj2kPH1ww4l",
    "createdDate": "2021-07-13T08:43:49.502Z",
    "lastUpdated": "2021-07-13T08:51:01.391Z",
```

```
    "issuerId": "387cdf11-59bf-4016-8afe-be0e0de7e45e",
    "ownerId": "387cdf11-59bf-4016-8afe-be0e0de7e45e",
    "grantees": [],
    "privileges": [],
    "type": "MANUAL"
}
```

9.15 Querying in Kibana

Get all the identities that have access to an identity e.g. privileges to a company's mailbox:

```
GET /identity-registry-identity-groups-v12/_search
{
  "query": {
    "nested": {
      "path": "privileges",
      "query": {
        "bool": {
          "must": [
            { "match": { "privileges.scopeId":
"xxc921d7-44b5-45d4-9b3a-60f6371151c6" } }
          ]
        }
      },
      "score_mode": "avg"
    }
  }
}
```

9.15.1 Creating Privileges

Creating a privilege is done using a POST request to the `/identity-groups/{groupId}/privileges/` endpoint:

POST `https://api.digitalpost.dk/apis/v1/identity-groups/2e273d28-0fb7-4797-a391-36f5e549e26c/privileges/`

With the body:

```
{
    "scopeId": "35a3e43f-4506-4f74-9cc5-9421d5e80bbe",
    "source": "MANUAL",
    "type": "CITIZEN"
}
```

After the privilege is created, the solution responds with the privilege:

```
{
```

```
    "id": "a9ff0e61-5292-4eec-a906-13fc505dd43a",
    "version": 0,
    "identityGroupId": "2e273d28-0fb7-4797-a391-36f5e549e26c",
    "issuerId": "35a3e43f-4506-4f74-9cc5-9421d5e80bbe",
    "scopeId": "35a3e43f-4506-4f74-9cc5-9421d5e80bbe",
    "type": "CITIZEN",
    "source": "MANUAL",
    "createdDate": "2021-07-26T03:38:44.279Z",
    "lastUpdated": "2021-07-26T03:38:44.279Z"
}
```


9.15.2 Updating Privileges

Using fetched privilege, we can modify certain fields. Let us change the source.

PUT `https://api.digitalpost.dk/apis/v1/identity-groups/2e273d28-0fb7-4797-a391-36f5e549e26c/privileges/a9ff0e61-5292-4eec-a906-13fc505dd43a`

With header corresponding to the current version of privilege (which in this case is 0):

```
If-Match: 0
```

And update body:


```
{
    "id": "a9ff0e61-5292-4eec-a906-13fc505dd43a",
    "version": 0,
    "identityGroupId": "2e273d28-0fb7-4797-a391-36f5e549e26c",
    "issuerId": "35a3e43f-4506-4f74-9cc5-9421d5e80bbe",
    "scopeId": "35a3e43f-4506-4f74-9cc5-9421d5e80bbe",
    "type": "CITIZEN",
    "source": "SELF_SERVICE",
    "createdDate": "2021-07-26T03:38:44.279Z",
    "lastUpdated": "2021-07-26T03:38:44.279Z"
}
```


After we have sent this update request, we get this result:


```
{
    "id": "a9ff0e61-5292-4eec-a906-13fc505dd43a",
    "version": 1,
    "identityGroupId": "2e273d28-0fb7-4797-a391-36f5e549e26c",
    "issuerId": "35a3e43f-4506-4f74-9cc5-9421d5e80bbe",
    "scopeId": "35a3e43f-4506-4f74-9cc5-9421d5e80bbe",
    "type": "CITIZEN",
    "source": "SELF_SERVICE",
    "createdDate": "2021-07-26T03:38:44.279Z",
    "lastUpdated": "2021-07-26T03:46:26.846Z"
}
```

9.15.3 Delegating Privileges

A privilege can in some situations be delegated as a child privilege.

GET `/identity-groups/2e273d28-0fb7-4797-a391-36f5e549e26c/privileges/`

This requires a new field to be added, `parentPrivilegeId`, which points to the original privilege. This can only be done if the parent privilege has source to be one of the following: `APPOINTED`, `MIGRATED`, and `NPTE`.

```
{
    "issuerId": "3a958236-7214-4278-bff6-136e6502e10c",
    "scopeId": "ea63e6de-9d89-4e38-bbbe-2b7e2fac1e69",
    "parentPrivilegeId": "6633ce13-5e7d-4ba2-acb9-30749a2508f3",
    "source": "SELF_SERVICE",
    "type": "LEGAL_REPRESENTATIVE"
}
```

And the response becomes

```
{
    "id": "b8a4a083-2a49-434a-992d-69af4ac7e0fa",
    "version": 0,
    "identityGroupId": "65ad60ed-3a57-4b49-8830-3dafd3a66571",
    "issuerId": "75c31129-8c09-445f-90e8-c39161093f07",
    "parentPrivilegeId": "6633ce13-5e7d-4ba2-acb9-30749a2508f3",
    "scopeId": "ea63e6de-9d89-4e38-bbbe-2b7e2fac1e69",
    "type": "LEGAL_REPRESENTATIVE",
    "source": "SELF_SERVICE",
    "createdDate": "2024-10-21T11:07:11.932Z",
    "lastUpdated": "2024-10-21T11:07:11.932Z"
}
```

9.15.4 Adding Grantee

When we have fetched the identity-group, we can add a grantee to that by using this endpoint:
POST `/identity-groups/{id}/grantees/`

Example

POST `https://api.digitalpost.dk/apis/v1/identity-groups/77464abe-f017-42b4-a278-1f29fc97fd84/grantees/`

With the body:

```
{
    "identityId": "2a1e9c38-0fbc-47ac-ab8c-a5ca213e2405",
    "issuerId": "f31cdacc-9c37-4be2-8e78-cf03bce61ea8"
}
```

After we create the grantee, we get this response:

```
{
    "id": "a8e46333-3d9a-4aee-9773-2ac53658389e",
    "version": 0,
    "identityGroupId": "77464abe-f017-42b4-a278-1f29fc97fd84",
    "identityId": "2a1e9c38-0fbc-47ac-ab8c-a5ca213e2405",
    "issuerId": "f31cdacc-9c37-4be2-8e78-cf03bce61ea8",
    "createdDate": "2021-07-13T07:28:11.448Z",
    "lastUpdated": "2021-07-13T07:28:11.448Z"
}
```


9.15.5 Updating Grantee

After having fetched a grantee, we can modify certain fields of this one. For instance, changing the identity:

PUT `https://api.digitalpost.dk/apis/v1/identity-groups/77464abe-f017-42b4-a278-1f29fc97fd84/grantees/a8e46333-3d9a-4aee-9773-2ac53658389e`

With header corresponding to the current version of privilege (which in this case is 0):

```
If-Match: 0
```

And update body:

```
{
    "identityId": "77a7475b-9f47-4a0b-a4b0-2232a5446a73",
    "issuerId": "f31cdacc-9c37-4be2-8e78-cf03bce61ea8"
}
```

When we send this update request this is the result we get:


```
{
    "id": "a8e46333-3d9a-4aee-9773-2ac53658389e",
    "version": 1,
    "identityGroupId": "77464abe-f017-42b4-a278-1f29fc97fd84",
    "identityId": "77a7475b-9f47-4a0b-a4b0-2232a5446a73",
    "issuerId": "f31cdacc-9c37-4be2-8e78-cf03bce61ea8",
    "createdDate": "2021-07-13T07:28:11.448Z",
    "lastUpdated": "2021-07-13T07:57:02.277Z"
}
```

10 Distribution - TI

10.1 Distribution Services

10.2 MeMo

MeMo (abbreviation for **me**ssage **mo**del) is used for exchanging messages in Digital Post. It is developed and maintained by Agency for Digital Government or in Danish “Digitaliseringsstyrelsen”. The goal of MeMo is to provide a standardized format for Digital Post messages and to better facilitate automated distribution and for messages to be self-contained.

10.2.1 Information

The detailed information about the format, documentation and examples for MeMo:

- MeMo https://digitaliser.dk/digital-post/vejledninger/memo

10.2.2 MeMo library

As MeMo is utilized by Digital Post, senders and recipients; handling (de-/serialization) from XML is encapsulated in a common library that is available for public use. As well as compression utilities for the TAR+LZMA for bulk shipments and conversions between adjacent MeMo versions.

These repositories can be accessed here:

- https://bitbucket.org/nc-dp/memo-lib-dot-net/src/master/
- https://bitbucket.org/nc-dp/memo-lib-java/src/master/

It comes with additional feature of the semantron validation, and contains the XSD is validated inside Digital post when a message is sent.

Note that the repository is using old memo version by default, it has to be explicitly provided what version it should use.

This ensures compatibility, intended to lower the need for support when developing integration with Digital Post.

10.2.3 Size requirements for messages and attachments

Different restrictions apply to a message depending on whether it is sent from a view client or from a sender system. However, it always applies that a message cannot contain more than 1 MainDocument and 10 AdditionalDocuments/TechnicalDocuments (combined), and each of these documents cannot contain more than 10 files (signifying different versions of the same document).

Requirements for messages sent from a view client

A message cannot exceed 70MB when it is sent from a view client. This limit is set to ensure that the entirety of the message does not exceed 99,5MB after the message is encoded. Furthermore, a single attachment cannot exceed 10MB.

Requirements for messages sent from a sender system

A message cannot exceed 99,5 MB when sent from a sender system. There is no limit on a single attachment as long as the entirety of the message does not exceed 99,5 MB.

10.3 Distribution REST services

Note, that services sending and receiving MeMo's messages only support XML (not JSON).

Generally all non-memo responses are JSON, but XML is supported by using the Accept header.

10.4 Inbound services

From the distribution component, the following services are exposed:

Service	URL	Usage	Required roles	Open API
Send MeMo messages	`POST /memos/`	Sending `.tar.lzma` files containing MeMo's, or single xml memo files, to the solution. Returns technical receipt with transmissionId	• Sender system	Swagger UI
Recall delayed MeMo message	`DELETE /memos/{memo-id}`	Recalling MeMos before due date	• Sender system • Public authority administrator	Swagger UI
Send MeMo messages as bulk	`POST /memos-bulk/`	Sending `.tar.lzma` files containing MeMo's, or single xml memo files, to the solution. Handles messages sent as bulk-transmissions always. Returns Technical receipt with transmissionId	• Sender system	Swagger UI
Fetch MeMo	`GET /memos/{memo-id}`	Fetching memo's from DP (publish-subscribe)	• Recipient System	Swagger UI
Fetch list of available MeMos	`GET /memos/`	Fetching a list of all available MeMos for the recipient system (publish subscribe)	• Recipient System	Swagger UI

Service	URL	Usage	Required roles	Open API
Send business receipt to Digital Post	`POST /memos/{memo-id}/receipt`	Sending a business receipt to DP	• Recipient System	Swagger UI
Fetch list of available business receipts	`GET /receipts/`	Fetching a list of available business receipts for the sender system (REST_PULL)	• Sender System	Swagger UI
Fetch business receipt	`GET /receipts/{receipt-id}`	Fetching business receipts from DP (REST_PULL)	• Sender System	Swagger UI
Delete buisness receipt	`DELETE /receipts/{receipt-id}`	Delete receipts	• Sender System	Swagger UI

10.4.1 Send MeMo messages

In the 'Send MeMo messages' request, the MeMos should be sent as a requests body with an accompanying Content-Type header:

- A tar archive compressed with LZMA, the type also used when archiving using MeMo-lib, with the Content-Type header set to `application/x-lzma`

or

- An xml file, which requires the Content-Type header to be set to `application/xml`.

The tar-lzma file can be named as wanted, but the individually files in the archive must be named the same as the `messageUUID` from the MeMo.

The name of the xml file when sending a single MeMo, should be `messageUUID` from MeMo, without the `.xml` file extension.

file type	Content-Type header	filename
tar.lzma	application/x-lzma	any filename
xml	application/xml	messageUUID

Should I send single messages or bulks?

> ⚠ It is important as a sender that you choose the correct way of sending messages as using the wrong method can negatively impact other senders.

This section is for when sending messages via `/memos/`.

Digital Post includes two distinct methods for sending messages via REST, each designed to cater to different communication needs. Messages can either be sent as a single message, or as bulks of multiple messages (a tar.lzma file with only one message is seen as a single message).

Single messages are meant for individual, one-on-one communication. It is intended for when an employee needs to write and send a message to a single recipient, and the message should be sent immediately.

On the other hand, bulk messages are designed for mass communication or mass sending messages. This interface allows for the distribution of messages to a large group of recipients. It's especially useful when sending out general notifications to all members of a particular group, for instance, all citizens in a municipality.
It can also be used for sending single messages, if e.g. the sender system is designed to store all messages that are sent within the day, and then sending all those messages at the same time.

Choosing the appropriate method based on your communication needs ensures efficient messaging and optimal communication, as sending bulks as single messages can negatively impact other senders. Always remember to take into account the number of messages you are sending when deciding which interface to use - If it is more than one, they should be sent as bulks.

In summary, individual messages are delivered shortly after being sent (excelt MeMos sent with delayed delivery), whereas bulk messages are processed with a different priority. If immediate delivery is not crucial or you send a high volume of messages, you should use the `/memos-bulk/` endpoint or properly package the messages as `.tar.lzma`, so as to not negatively affect the Digital Post solution. This endpoint processes all messages in bulk regardless of size and is recommended to alleviate traffic congestion that occurs when using single message transmissions, which can adversely impact other users who depend on quick message delivery.

10.5 Outbound services

From the distribution component, there are the following outbound services. The URLs which are used in the outbound services are the either the `endpoint` or the `receiptEndpoint` which the registered on the relevant sender or receiver system.

Service	Usage	Data sent
Send receipt via REST	Sending business receipts to sender systems	Business receipt with transmissionId
Send message via REST	Sending MeMo messages to recipient systems	MeMo

10.5.1 Sender system respond to received Business Receipt

The sender system is expected to respond with status code **200 OK, 201 CREATED** or **202 ACCEPTED** and without a response body upon successfully receiving a Business Receipt.

Outbound MeMo REST Push request

This page describes how a MeMo is sent by Digital Post via REST Push.

A MeMo is pushed to a REST PUSH recipient system with mutual SSL using HTTP POST to the recipient system's endpoint. The MeMo's messageUUID is appended to the URL, using request parameter name `memo-message-uuid`. This allows the recipient system to identify the MeMo if the body of the message cannot be interpreted, for some reason.

Example:

```
https://dp.skat.dk/modtagersystem/kontaktpunkt-1?memo-message-
uuid=a66fcd7b-3392-4c69-ae2e-48f5c2e5ad98
```

Or if the URL already had parameters:

```
https://dp.skat.dk/modtagersystem?kontaktpunkt=1&memo-message-
uuid=a66fcd7b-3392-4c69-ae2e-48f5c2e5ad98
```

The Content-Type header is: `application/xml`

Example:

```
POST /modtagersystem?kontaktpunkt=1&memo-message-
uuid=b6fcb074-2843-4f66-8481-682715232ac9 HTTP/1.1
Host: dp.skat.dk
Content-Type: application/xml
Content-Length: 18634

<memo:Message xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:memo="https://DigitalPost.dk/MeMo-1" memoVersion="1.2" xmlns:xsd="http://
www.w3.org/2001/XMLSchema#">
    <memo:MessageHeader>
        <memo:messageType>DIGITALPOST</memo:messageType>
        <memo:messageUUID>b6fcb074-2843-4f66-8481-682715232ac9</memo:messageUUID>
        <memo:label>Til forvaltningen</memo:label>
        <memo:reply>true</memo:reply>
        <memo:mandatory>false</memo:mandatory>
        <memo:legalNotification>false</memo:legalNotification>
        <memo:Sender>
            <memo:senderID>0610328534</memo:senderID>
            <memo:idType>CPR</memo:idType>
            <memo:label>Lone Hansen</memo:label>
        </memo:Sender>
        <memo:Recipient>
            <memo:recipientID>63636363</memo:recipientID>
```

```
            <memo:idType>CVR</memo:idType>
        </memo:Recipient>
    </memo:MessageHeader>
    <memo:MessageBody>
        <memo:createdDateTime>2020-06-29T12:00:00Z</memo:createdDateTime>
        <memo:MainDocument>
            <memo:File>
                <memo:encodingFormat>text/html</memo:encodingFormat>
                <memo:filename>Hoveddokument</memo:filename>
                <memo:language>da</memo:language>
                <memo:content>JVBER....</memo:content>
            </memo:File>
        </memo:MainDocument>
    </memo:MessageBody>
</memo:Message>
```

Expected response:

One of:

- 200 OK
- 201 CREATED
- 202 ACCEPTED

Timeouts:

Recipient systems are expected to reply within the following timeout settings:

- connect
    - Max time to use establishing HTTP connection
    - 10 seconds
- read
    - Max time to use during the HTTP POST
    - 10 seconds

10.6 Sending single MeMo

MeMos must be POST'ed using certificate from active sender system or authentication will fail with a 401.

HTTP POST with Content-Type `application/xml` to Digital Post `/memos/` endpoint. `messageUUID` of the MeMo must be added as a request parameter named 'memo-message-uuid' and Content-Length header **must** be correct:

```
POST /apis/v1/memos/?memo-message-uuid=b6fcb074-2843-4f66-8481-682715232ac9 HTTP/1.1
Host: digitalpost.dk
Content-Type: application/xml
Content-Length: 18634

<memo:Message xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:memo="https://DigitalPost.dk/MeMo-1" memoVersion="1.2" xmlns:xsd="http://
www.w3.org/2001/XMLSchema#">
    <memo:MessageHeader>
        <memo:messageType>DIGITALPOST</memo:messageType>
        <memo:messageUUID>b6fcb074-2843-4f66-8481-682715232ac9</memo:messageUUID>
        <memo:label>Til forvaltningen</memo:label>
```

```
        <memo:reply>true</memo:reply>
        <memo:mandatory>false</memo:mandatory>
        <memo:legalNotification>false</memo:legalNotification>
        <memo:Sender>
            <memo:senderID>0610328534</memo:senderID>
            <memo:idType>CPR</memo:idType>
            <memo:label>Lone Hansen</memo:label>
        </memo:Sender>
        <memo:Recipient>
            <memo:recipientID>63636363</memo:recipientID>
            <memo:idType>CVR</memo:idType>
        </memo:Recipient>
    </memo:MessageHeader>
    <memo:MessageBody>
        <memo:createdDateTime>2020-06-29T12:00:00Z</memo:createdDateTime>
        <memo:MainDocument>
            <memo:File>
                <memo:encodingFormat>text/html</memo:encodingFormat>
                <memo:filename>Hoveddokument</memo:filename>
                <memo:language>da</memo:language>
                <memo:content>JVBER....</memo:content>
            </memo:File>
        </memo:MainDocument>
    </memo:MessageBody>
</memo:Message>
```

10.6.1 Sending multiple MeMos

Raw body

```
POST /apis/v1/memos/ HTTP/1.1
Host: digitalpost.dk
Content-Type: application/x-lzma
Content-Length: 2223513431

"<contents here>"
```

Using curl:

```
curl --location --request POST 'https://digitalpost.dk/apis/v1/memos/' \
--header 'Content-Type: application/x-lzma' \
--data-binary '@/home/user/memos/memo_archive.tar.lzma'
```

Multipart

Form element name must be `file`. Form element content type must be `application/x-lzma`.

```
curl --location --request POST 'https://digitalpost.dk/apis/v1/memos/' \
--header 'Content-Type: multipart/form-data'
--header 'Accept: application/json' \
```

```
--form 'file=@"/home/user/memos/memo_archive.tar.lzma"'
```

10.6.2 REST_PULL service protocol

The service protocol REST_PULL can be used for both recipient and sender rest systems.

For recipient system the service protocols REST_PULL and REST_PUBLISH_SUBSCRIBE are quite similar with the only difference being whether a notification will be sent to the systems endpoint, when a MeMo is ready to be fetched, or not. The purpose of the service protocol REST_PULL is for recipient systems to explicitly choose not to have notifications every time a MeMo is ready to be fetched. REST_PULL systems cannot have endpoints.
Sender systems with REST_PULL service protocol do not get business receipts send to a receipt-endpoint. It is the sender systems responsibility to fetch business receipt associated with that system.

Systems with REST_PULL can at any time fetch all available MeMos and/or receipts. See more of how this fetching is done in section 10.7.5 “Distribution use case examples” or section 10.8.2 “Fetching business receipts”.

10.7 Sending large amount of messages

10.7.1 memo-bulk via rest

In the endpoint `/memos/`, messages can be categorized into single memos and in bulk transmissions depending on if they are sent as `xml` or `tar.lzma` files respectively. However with `/memos-bulk/`, all messages are sent as bulk transmissions disregarding the file format. `/memos-bulk/` is the recommended endpoint to use when sending single messages that are not urgent, instead being processed by a lower prioritized queue. This prevents unnecessary blocking of single memos that are urgent and expected to be sent immediately.

10.7.2 Accepted File Formats

Endpoint	Protocol	File Content	Note
`/memos-bulk/`	POST	`.tar.lzma`	Archive file
`/memos-bulk/`	POST	`.xml`	Single memo, will work as if a bulk transmission.

10.7.3 Example

Examples are provided in the following of how to use `/memos-bulk/`.

Sending a bulk archive:

```
curl --location --request POST 'https://digitalpost.dk/apis/v1/memos-bulk/' \
--header 'Content-Type: multipart/form-data'
--header 'Accept: application/json' \
--form 'file=@"/home/user/memos/memo_archive.tar.lzma"'
```

Sending a single memo as a bulk transmission:

```
curl --location --request POST 'https://digitalpost.dk/apis/v1/memos-bulk/?memo-
message-uuid=b6fcb074-2843-4f66-8481-682715232ac9'  \
--header "Content-Type: application/xml" \
--header "Content-Length: 18634" \
--data '<memo:Message xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:memo="https://DigitalPost.dk/MeMo-1" memoVersion="1.2" xmlns:xsd="http://
www.w3.org/2001/XMLSchema#">
    <memo:MessageHeader>
        <memo:messageType>DIGITALPOST</memo:messageType>
        <memo:messageUUID>b6fcb074-2843-4f66-8481-682715232ac9</memo:messageUUID>
        <memo:label>Til forvaltningen</memo:label>
        <memo:reply>true</memo:reply>
        <memo:mandatory>false</memo:mandatory>
        <memo:legalNotification>false</memo:legalNotification>
        <memo:Sender>
            <memo:senderID>0610328534</memo:senderID>
            <memo:idType>CPR</memo:idType>
            <memo:label>Lone Hansen</memo:label>
        </memo:Sender>
        <memo:Recipient>
            <memo:recipientID>63636363</memo:recipientID>
            <memo:idType>CVR</memo:idType>
        </memo:Recipient>
    </memo:MessageHeader>
    <memo:MessageBody>
        <memo:createdDateTime>2020-06-29T12:00:00Z</memo:createdDateTime>
        <memo:MainDocument>
            <memo:File>
                <memo:encodingFormat>text/html</memo:encodingFormat>
                <memo:filename>Hoveddokument</memo:filename>
                <memo:language>da</memo:language>
                <memo:content>JVBER....</memo:content>
            </memo:File>
        </memo:MainDocument>
    </memo:MessageBody>
</memo:Message>'
```

In both cases the return value is a Technical Receipt (for more details about the return value see Distribution Receipt Domain Model).

10.7.4 Bulk MeMo SFTP Service

- SFTP Server folder structure
- Receipt XML

To support bulk memos upload, the SFTP protocol can be used.

Once a SFTP sender system has been setup in the system registry (using Administrative Access (AA)), you use the SSH username from the sender system (it is called “SSH brugernavn” in AA), and the private key associated with the public key you uploaded for the sender system in AA, to connect to Digital Post SFTP server, where you put archives to be sent and pull receipts.

SFTP Server folder structure

Root folder (per user)	Subfolders	User access rights	Description
`/{username}` Note: SFTP server uses chroot functionality to prevent access to filesystem thus real path is different (and is up to server admins)	.	**dir**: RX	Home folder. Contains subfolders defined below. The username is the id of the systems identity id encoded in base62 to ensure that it is always under 32 characters.
	`memos/{bulkMemoTarLzmaUUID}.tar.lzma`	**files:** RW **dir:** WX	Contains bulk memo `.tar.lzma` files uploaded by sender system. File name must end with `.tar.lzma` and be unique identifier (UUID). Files that do not match this naming pattern are ignored. Example file name: `91581881-23cb-40b0-9e50-9c4d500649e3.tar.lzma`
	`memos/tmp/*.*`	**files:** RW **dir:** WX	Sender systems are required to use a pattern of **copy&move**, ie. files are to be uploaded to `memos/tmp` folder first and then moved to `memos`

Root folder (per user)	Subfolders	User access rights	Description
	`receipts/`	**files:** R **dir:** WX	Contains technical receipts and business receipts. Technical receipt status is either `RECEIVED` (technical receipt) or negative (`INVALID` , `NOT_ALLOWED`) - if tar.lzma file is corrupted or incorrect in any other way. The UUID of the technical receipt is taken from the uploaded tar.lzma file and is also the transmission ID of both the technical receipt and the business receipt. Business receipts are generated by DP - one per MeMo Receipts will be available for 14 days, after which they will be automatically deleted.
	`receipts/tmp/*.*`	None	Temporary folder used by system user to write business receipts before they are moved to `receipts` folder

Users SHOULD not create additional folders themselves.

Receipt XML

```
<?xml version='1.0' encoding='UTF-8'?>
<receipt>
    <transmissionId>a5345a15-e5cc-4f6d-b2c7-97d0b036bfdf</transmissionId>
    <messageUUID>e9f3bd3f-11d0-4af2-a72f-01327c5bcc96</messageUUID>
    <messageId>MSG-12345</messageId>
    <errorCode>(optional)123</errorCode>
    <errorMessage>(optional) some error message</errorMessage>
    <timeStamp>2020-06-05T12:00:00Z</timeStamp>
    <receiptStatus>COMPLETED</receiptStatus>
</receipt>
```

(Note: real xml will not be pretty printed, but rather a single line text file, in UTF-8 encoding)

Receipts will only be available for 14 days, after which they will be deleted automatically!

10.7.5 Distribution use case examples

Sending memo messages over REST PUSH

In DP a sender system can send a message (MeMo) using the Distribution API to CREATE/POST a MeMo to the DP system. The sender system is notified using technical and business receipts.

Sending a single message from a sender system using REST PUSH:

```
POST /apis/v1/memos/?memo-message-uuid=e60394cd-1ba9-4ff0-833b-9a05113b3df1
```

Status is CREATED, 201 http status. The response is shown below.

```
{
  "transmissionId": "aa1c5cde-4f52-4ff4-b9c5-d737bf478544",
  "timeStamp": "2020-07-02T08:23:07Z",
  "receiptStatus": "RECEIVED"
}
```

The response entails a technical receipt.

The receipt has a transmissionId to identify the transaction.

When the message have been validated by DP and is ready to either be delivered to a recipients mailbox or send to the recipients receiver system, a business receipt is generated by DP and send to the sender systems receipt endpoint.

An example of a business receipt is shown below:

```
{
  "transmissionId": "238179a2-b1fe-4504-b1b7-6be7856974d3",
  "messageUUID": "e60394cd-1ba9-4ff0-833b-9a05113b3df1",
  "messageId": "MSG-12345",
  "errorCode": null,
  "errorMessage": null,
  "timeStamp": "2020-06-25T12:55:00.262362",
  "receiptStatus": "COMPLETED"
}
```

The business receipt status is completed. In use cases where a negative receipt is sent, a message is shown to identify the problem. In both cases, DP expect the sender system to respond to the business receipt with either 200 OK or 202 ACCEPTED with an empty body.

Fetching a single memo over REST PULL

MeMos can both be fetched individually by the use of the ID for a specific message or a system can fetch all available MeMos for that given system.

In DP a REST_PUBLISH_SUBSCRIBE recipient system is notified when a MeMo message is available for it to fetch. However, a REST_PULL recipient systems do not get this notification and are responsible themselves to fetch the currently available MeMos if any.

Fetching a single MeMo for a REST_PUBLISH_SUBSCRIBE recipient system

When a new MeMo is available, a MeMo notification is sent to a REST_PUBLISH_SUBSCRIBE recipient system endpoint, the expected response is 200 OK.

Recipient system service protocol:
`serviceProtocol: ServiceProtocolType.REST_PUBLISH_SUBSCRIBE`

Recipient system endpoint example:
`https://host:8204/memos/`

The MeMo notification contains an URL that points to the location of the new MeMo.

(url) path example:

`https://digital_post_host:port/memos/e60394cd-1ba9-4ff0-833b-9a05113b3df1`

The path includes the ID for that specific message.

The recipient system then makes a

GET `/memos/e60394cd-1ba9-4ff0-833b-9a05113b3df1`

which fetches the MeMo.

Fetching a list of MeMos for a REST_PULL recipient system

A REST_PULL recipient system is responsible itself to fetch the currently available MeMos. A list of all available MeMos will be returned.

The recipient system has to make a

GET `/memos/`

and a list is returned.

Sending a Receipt for a REST_PULL recipient system

A REST_PULL recipient system can send a business receipt containing a MeMo ID of the message that it has fetched.

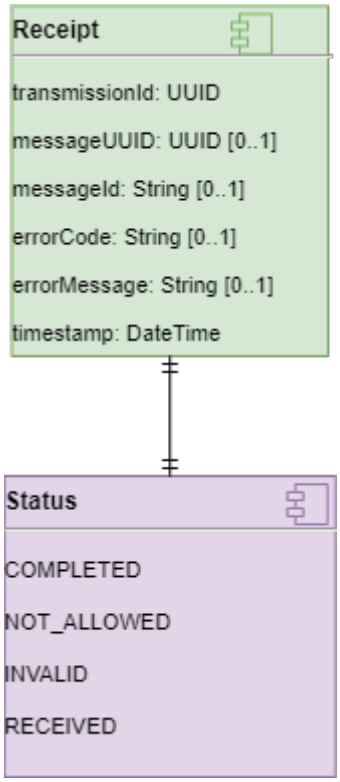
POST `/memos/{memo id}/receipt`

The MeMo with the given ID will be found in the list of available MeMos for the system. The MeMo will firstly be deleted off of the list containing available MeMos, and afterwards from object storage. The MeMo can no longer be fetched.

It is only possible to send a business receipt for a single MeMo at a time.

10.8 Receipts

This section is only relevant for sender systems registered to use MeMo receipts.



The above diagrams shows the domain model for receipts. Fields of type String contains a maximum of 512 characters.

DP operates with two types of receipts, namely Technical Receipts, and Business Receipts. Technical receipts are sent immediately when the message(s) is received by Digital Post, and confirms that Digital Post has received the message. Business receipts are the final state for the sender - This can e.g. be that even though Digital Post was able to receive the message, the memo xml has failed validation, or that the message has been successfully validated by DP, and will be sent.

The meaning of the statuses are:

- `COMPLETED` : Digital Post has validated the messages.
- `NOT_ALLOWED` : The sender is not allowed to send this message
- `INVALID` : There is something invalid in the request - E.g. the MeMo is not in the correct format
- `RECEIVED` : Digital Post has received the message, and will start to validate it. The message is still the responsibility of the sender

Some examples below refer to error codes and error messages in both technical and business receipts. Those are used to indicate issues, that the system encountered during message validation. There are various types, and one can look up the details of them under the section **“Back-end validation and error codes in distribution”**.

The following sections will go through technical and business receipts for each protocol.

10.8.1 REST receipt procedure

Technical receipts

The technical receipt indicates if DP have received the message or not. If the message is received it will be further processed and validated. Please note that the results of the processing and validation will be in the business receipt

and not in the technical receipt. The technical receipt for REST protocol is the REST response. This response will only contain a body if the call is succeeded, and HTTP status 201 is returned. The body will contain a JSON receipt. The field values are:

```
{
    "transmissionId": "86f13750-8068-44c1-93cf-a915998831cf",
    "timeStamp": "2020-12-15T08:23:32.583Z",
    "receiptStatus": "RECEIVED"
}
```

Code snippet above contains correct “Technical Receipt”. The status of request is ‘201 Created’. And below is example of incorrect request status 400.

```
{
    "code": "ValidationException",
    "message": "File type 'null' not allowed. Allowed file types: application/xml,
application/x-lzma",
    "fieldErrors": []
}
```

Short description of fields in receipt:

- `transmissionId` : A UUID generated by Digital Post. This UUID will also be returned in the business receipt, and should be used to chain the two together
- `timeStamp` : Timestamp of when the message was received, which is always now (UTC timezone)
- `receiptStatus` : `RECEIVED` meaning that Digital Post have received the message

Business receipt

The business receipt for the REST protocol are generated when the message has been either rejected or passed the validation. The business receipt will contain the result of the validation, and a positiv result means that the message will be delivered to the recipient.

If a sender system it set up with the service protocol *REST_PUSH* the receipt will be POST’ed to the receipt endpoint of the sender system. If the sender system is set up with the service protocol *REST_PULL* the receipts can be found by the sender system by issuing a GET request to the endpoint */receipts/*. In both cases the body of the receipt will be in the JSON format.

Examples of REST business receipts are shown below:

```
{
  "transmissionId": "86f13750-8068-44c1-93cf-a915998831cf",
  "messageUUID": "182bd6d1-ab9f-48fb-84f6-4f243ace9780",
  "messageId": "MSG-81220",
  "errorCode": null,
  "errorMessage": null,
  "timeStamp": "2020-12-15T08:23:32.583Z",
  "receiptStatus": "COMPLETED"
}
```

Code snippet above shows ‘Business Receipt’ with status ‘ `COMPLETED` ’. And below with status ' `INVALID` '.

```
{
  "transmissionId": "8e62eb2a-8ef7-4034-8bb4-4021ecd9c377",
  "messageUUID": "182bd6d1-ab9f-48fb-84f6-4f243ace9780",
  "messageId": null,
  "errorCode": "message.uuid.not.unique",
  "errorMessage": "The MessageUUID 182bd6d1-ab9f-48fb-84f6-4f243ace9780 is invalid.
MessageUUID must be a unique UUID",
  "timeStamp": "2020-12-15T08:23:32.583Z",
  "receiptStatus": "INVALID"
}
```


Short description of fields in receipt:

- `transmissionId` : A UUID generated by DP. This UUID will also be returned in the technical receipt, and should be used to chain the two together
- `messageUuid` : UUID of MeMo if available, otherwise empty (e.g. if `tar.lzma` extraction fails)
- `messageId` : Read from MeMo if defined, otherwise empty
- `errorCode` : Code of error, that appeared during validation process
- `errorMessage` : Description of error
- `timestamp` : Now (UTC)
- `receiptStatus` : Status of message in a system

Mapping between receipt status and error codes

When an error occurs, different error codes result in different receiptStatus states.

ErrorCodes	**ReceiptStatus**
memo.invalid	INVALID
memo.infected	INVALID
message.uuid.does.not.match.file.name	INVALID
message.document.number.higher.than.allowed	INVALID
memo.document.action.entrypoint.invalid	INVALID
message.file.number.higher.than.allowed	INVALID
message.uuid.does.not.match.file.name	INVALID
recipient.cvr.invalid	INVALID

ErrorCodes	ReceiptStatus
recipient.cpr.invalid	INVALID
recipient.not.found	INVALID
recipient.contact.point.id.required	INVALID
recipient.mailbox.not.found	INVALID
recipient.mailbox.and.default.recipient.system.not.found	INVALID
recipient.nem.sms.subscription.not.found	INVALID
recipient.nem.sms.subscription.mobile.number.not.verified	INVALID
recipient.mailbox.not.found	INVALID
sender.cvr.invalid	INVALID
sender.cpr.invalid	INVALID
sender.not.found	INVALID
id.type.invalid	INVALID
file.empty.not.allowed	NOT_ALLOWED
file.format.not.allowed	NOT_ALLOWED
file.extension.not.allowed	NOT_ALLOWED
Invalid file name found for one or more files in document of type	NOT_ALLOWED
do.not.deliver.until.date.too.late	NOT_ALLOWED
do.not.deliver.until.date.too.early	NOT_ALLOWED
memo.file.size.too.large	NOT_ALLOWED

ErrorCodes	ReceiptStatus
recipient.nem.sms.is.not.allowed	NOT_ALLOWED
recipient.type.cannot.receive.legal.notifications	NOT_ALLOWED
recipient.contact.point.not.allowed.for.id.type	NOT_ALLOWED
recipient.is.exempt	NOT_ALLOWED
recipient.is.closed	NOT_ALLOWED
sender.system.not.found	NOT_ALLOWED
sender.organisation.id.does.not.match	NOT_ALLOWED
sender.mandatory.message.not.allowed	NOT_ALLOWED
sender.legal.notification.not.allowed	NOT_ALLOWED
sender.do.not.deliver.until.date.not.allowed	NOT_ALLOWED
sender.system.is.not.activated	NOT_ALLOWED
sender.system.is.deactivated	NOT_ALLOWED
sender.type.not.allowed	NOT_ALLOWED
sender.system.forward.not.allowed	NOT_ALLOWED

10.8.2 Fetching business receipts

REST_PULL sender systems are responsible for fetching available receipts themselves. A list with UUID's of available receipts for a given system can be fetched, after which the receipts can be fetched individually by the use of the UUID’s. The receipts will be deleted after fetched individually, unless otherwise is specified, or deleted by a schedule cleaner that will delete receipt older than 7 days.

10.8.3 Fetching receipts for a REST_PULL sender system

A REST_PULL sender system is responsible for fetching the currently available receipts.

Fetching a list of available receipts

The sender system has to make a GET as such:

GET `/receipts/`

A list of all available receipts UUID will be returned.
Example of result:

```
{
  "content": [
    "966925f3-569a-4d9a-b688-f49eac9e2c7b",
    "888e528f-1ef0-44ad-ab57-910344cf2003",
    "e93ab749-1b58-4f40-808c-77327ced20bf",
    "c8e94729-b2b9-49bc-99f0-4e7f9863fde0",
    "759e3921-36a7-4353-971e-b0edd93e35ab",
    "c46d1f6a-8947-4758-bb5f-51e356d58a1d"
  ],
  "number": 0,
  "size": 20,
  "totalElements": 6,
  "totalPages": 1
}
```

And a list of UUID's for the individually is returned, which can be used to fetch a single receipt.

Search parameter can be use in order to go through multiple pages. Moreover the size per page can also be change by the usage of search parameter. As standard 20 receipts are returned per page.

Example:

`/receipts/?size=100`

Fetching a single receipt with automatic deletion

GET `/receipts/{receiptId}`

Example of request:

```
GET /receipts/966925f3-569a-4d9a-b688-f49eac9e2c7b
```

Result when using request above:

```
<Receipt>
    <transmissionId>3fa532b9-2b94-4fc3-b979-5d6c3bbf2a3e</transmissionId>
    <messageUUID>c2ec7c7a-f197-4ade-887e-6ce0d3a9267c</messageUUID>
    <timeStamp>2021-07-15T12:08:38.715Z</timeStamp>
    <receiptStatus>COMPLETED</receiptStatus>
</Receipt>
```

Please notice that like in the 'Fetching Memo' flow, the result is in xml format and that the receipt will be deleted after it have be returned.

Fetching a single receipt without automatic deletion

If automatic deletion of a receipt after fetching is not desired, adding the "delete=" queryparam allows one to specify whether the receipt should be deleted after fetching:

GET `/receipts/{receiptId}/?delete=false`

True can also be specified, but then in effect it works the same was as the regular **GET** `/receipts/{receiptId}`

Deleting a single receipt without fetching

It is also possible to delete a single receipt without fetching it.

DELETE `/receipts/{receiptId}`

10.8.4 Bulk-fetching receipts

It is also possible to fetch a list of receipts for the sender system by using **GET** `/receipts-bulk/`. The only supported query parameters are *size* and *page* for paging. After receipts are fetched, each receipt should be deleted individually.

Example request

GET `/receipts-bulk/?size=2&page=3`

gives a response like


```
{
    "currentPage": 3,
    "totalPages": 1004,
    "elementsOnPage": 2,
    "totalElements": 2007,
    "receipts": [
        {
            "transmissionId": "0206bdc8-a254-4b33-8eac-b847afc6afa6",
            "messageUUID": "46f39e7f-b5f4-4bf3-99ba-7651c4c7f5a1",
            "errorCode": "message.uuid.not.unique",
            "errorMessage": "The MessageUUID 46f39e7f-b5f4-4bf3-99ba-7651c4c7f5a1 is
invalid. MessageUUID must be a unique UUID",
            "timeStamp": "2022-12-06T07:59:13.554Z",
            "receiptStatus": "INVALID",
            "id": "726fccc7-8fbc-4464-a836-d80504d36b04"
        },
        {
            "transmissionId": "fdb52007-f5e2-430c-ac92-3c9f95e25ff3",
```

```
            "messageUUID": "46f39e7f-b5f4-4bf3-99ba-7651c4c7f5a1",
            "errorCode": "message.uuid.not.unique",
            "errorMessage": "The MessageUUID 46f39e7f-b5f4-4bf3-99ba-7651c4c7f5a1 is
invalid. MessageUUID must be a unique UUID",
            "timeStamp": "2022-12-06T07:59:14.377Z",
            "receiptStatus": "INVALID",
            "id": "5505867d-8a10-46fa-ad30-4ffefc9431ad"
        }
    ]
}
```


10.8.5 Sending business receipts as a recipient system

When DP sends a MeMo through REST PUSH to a recipient system a Business Receipt is expected. A Business Receipt confirms that the recipient system has successfully been able to handle the received MeMo and that the MeMo will not need to be resent. The format of the business receipt is described in the section **“DP Receipt domain model“**. If DP does not receive a successful Business Receipt when a MeMo has been sent to a recipient system, DP will resend the MeMo according to the flow described in the section **“Flow for resending messages”**.

When a recipient system responds to DP the value of the field *receiptStatus* for Business Receipts does not matter to DP. DP evaluates a Business receipt as successful when the field *errorMessage* is empty (and the field *errorCode* is not "virus.detected").

If the errorCode is virus.detected the message will not be resent. Any other negative receipts will be logged in the event-log, but will not otherwise impact the re-sending of messages.

10.9 Outbound receipt REST push request

This page describes how receipts are sent by Digital Post via REST Push.

A receipt is pushed to a REST PUSH sender system with mutual SSL using HTTP POST to the recipient system’s endpoint. The receipt is sent to the endpoint URL defined for the sender system that sent the MeMo. This is done via the “kvitterings-end point” field in Administrative Access.

REST receipt request to a sender system	
description	Sends a receipt to the sender system
content-type	application/json
encoding	UTF-8
request-type	POST
Accepted responses	200, 201, 202
Data	A digital post distribution receipt - see section 10.8 “Receipts”

If the sender system responds with an unaccepted response (e.g. http 400, 500), the receipt will be sent again at a later time. For details on the retry-flow for receipts, see section 10.11 “Flow for resending messages”.

10.9.1 Temporary blacklisting of receipt endpoints

When a receipt is being sent, a check will be made on the endpoint to see how many times a receipt has previously failed to be sent to that endpoint. If there have been more than the allowed number of failures within a specified timeframe the endpoint is treated as blacklisted and the receipt will not be sent.

Configuration	Value
Number of failures before blacklisting	50
Timeframe (failures must occur inside to trigger blacklist)	60 minutes

These values are configurable and may change in the future. Changes to these values will be announced on Digitaliser.

When a receipt is not sent, the event log will contain an entry for the receipt with the error text “`Business receipt not delivered due to temporary system timeout`”. The receipt will then be attempted to be sent again at a later date, as described in 10.11 “Flow for resending messages”.

To give an example, if the max number of failures is set to 5 and a situation occurs where 6 receipts fails to be send, during the 1 hour time frame, the system is temporally blacklisted and receipts will not be sent. Later when a receipt is attempted to be sent and there is no longer more than 5 failed receipts in the 1 hour time frame, the system will no longer be blacklisted and the receipt will again be sent.

10.10 At-least once principle

Since the delivery of messages to recipients is the number one priority of Digital Post, the solution adheres to an “At least once”-principle when sending out requests to sender systems and recipient systems. This means that Digital post will try to send a message until the first messages succeeds. Under normal operation, Digital Post will send messages through integrations only once, of course. But if there’s a fatal infrastructure disaster, there is a risk that the Solution will resend out any messages that haven’t been processed completely in the near real-time backup. If this situation occurs, the volume is expected to be a few seconds (<20 seconds) worth of processing that would be resent. This holds true for both MeMo-messages, Receipts and Publishing of changes (for systems subscribing on changes of Contacts, for instance).

The different events that will be resent covers:

Event	Description	Components	Referenceses
MeMo Flow	MeMos get delivered to a recipient system	• Distribution-sender-rest • Mailbox	Receiving MeMo Flow for resending messages

Event	Description	Components	Referenceses
Receipt flow	Same as MeMo flow, but a receipt get delivered to a recipient or sender system	• Distribution-sender-rest	Sending MeMo Receiving MeMo Flow for resending messages Distribution Receipt Domain Model
Subscription notifications	Sent to indicate changes or new data being available to be fetched.	• Distribution-sender-rest • Mailbox • System-registry • Contact-store • Statistical Reporting	Receiving MeMo#Delivering-one-MeMo-over-REST-PUBLISH-SUBSCRIBE
Mailbox flow	Mailbox includes notifications flow to inform user that new messages are available.	• Mailbox	Notifications

Duplicate MeMo Handling

When a recipient system receives a duplicate MeMo message, it must send a business receipt, even if one was sent initially. This is crucial to comply with Digital Post's monitoring protocols and prevent system deactivation. Upon receiving a MeMo, the system should check for duplicates and send a receipt regardless. Digital Post monitors receipt compliance and may deactivate systems that fail to send receipts for duplicates consistently after several attempts, see section 10.11 “Flow for resending messages”.

10.11 Flow for resending messages

The following section describes the flow for resending MeMo's to REST recipient systems. Technical- and business receipts or lack thereof triggers the resending flow. These receipts are exchanged between Digital Post and recipient systems to ensure correct delivery and integrity of messages. The flow is terminated as soon as the recipient system returns a positive business receipt or after 7 days.

When the flow is terminated by 7 days passing without a positive business receipt, the memo is redirected, and a mail is sent to the email address contacts registered on the system notifying of the new destination. The redirections is as follows:

- If the failure was to a default recipient system, the memo is saved in the mailbox for the organisation, which can be accessed through http://virk.dk.
- If the failure was to a non-default recipient system, the memo is sent to the default recipient system. If it fails for two days to this system, the message is saved in the mailbox.

If either one of the failure scenarios happens for a recipient system of type `REST_PUBLISH_SUBSCRIBE` or `REST_PULL` and the memo is redirected to the mailbox for the organisation or to a default system of type `REST_PUSH`, then the information about the memo will no longer be returned in the list of available messages.

Technical receipts

(Only for `REST_PUSH` systems and `REST_PUBLISH_SUBSCRIBE`): When a `REST_PUSH` or `REST_PUBLISH_SUBSCRIBE` system gets called by the solution i.e. a REST call is pushed to the system, a technical receipt is created based on the result. The following flow is triggered if there has not been received a technical receipt with "http 200/201/202":

1. The message is resent after **10 minutes**
2. The message is resent after **8 hours**
3. The message is resent after **16 hours**
4. The message is resent after **24 hours**
5. The message is resent every **24 hours for up to 6 days**

Business receipts

If a message is sent from Digital Post to a recipient system and a positive technical receipt is returned, Digital Post will await a business receipt from that recipient system. If a positive business receipt (receiptStatus=COMPLETED) is not returned, the following flow is triggered (excl. returned receipts with error code "`virus.detected`"). In case of `REST_PUBLISH_SUBSCRIBE` and `REST_PULL` recipient system, the retry will make sure the memo is available until it is consumed and a receipt has been created.

1. The message is resent after **8 hours**
2. The message is resent after **16 hours**
3. The message is resent after **24 hours**
4. The message is resent every **24 hours for up to 6 days**

Retrying to default recipient system

After the full technical or business retry flow to a recipient system is completed without a positive business receipt, the message is attempted delivered to the default recipient system, then again after **10 minutes** and **25 hours**. If no delivery can be made to the default recipient system, the message is delivered to the default mailbox, which can be accessed through http://virk.dk or commercial view clients.

System deactivation

If messages to a system fail, the email contacts of the systems will be notified before the message is redirected at day 7. After a message starts failing (no positive business receipt is returned), a notification mail will be sent on day 2 and day 5 after the message was sent for the first time, assuming it is still failing.

The mail will notify that something is failing, which may lead to system deactivation, and will give some info on what system fails and what messages are failing.

To avoid spamming the contacts, the mails are sent based on the earliest failure - i.e. if a memo on day 0 and day 1 are failing, mails are sent on day 2 and 5 for the first message, but not on day 3 and day 6.

When a memo is redirected on day 7, the DP system will check if the recipient system has any message within the last 7 days where it has received both a positive technical receipt and a positive business receipt for any message. If this is not the case, the system will be deemed systematically failing, and will be deactivated so future messages do not get delayed for 7 days of retry before the recipient receives them.
This means that if only a few messages of many fail (e.g. due to an edge case bug in the recipient system), the recipient system will not be deactivated, and only the failing messages are redirected.

Handling of error code `virus.detected`

If a negative receipt is returned with the error "`virus.detected=Virus fundet I modtaget payload for MeMo med id {0}`" the flow is not initiated. The message is handled manually DIGST.

10.12 Flow for resending business receipts - REST Push protocol

When sending MeMos from a sender system with REST_PUSH protocol, Digital Post will retry sending the business receipt if the business receipts fails to deliver to the sender systems provided endpoint.

Digital Post will retry sending the business receipt every 6 hours for 5 days.

If Digital Post is ultimately unable to deliver the business receipt, it will not be lost. The validation status of the message can be found via the event-log. The sender system can do a lookup in the event log to find the validation status.

10.13 MeMo validation

10.13.1 Invalid characters in filename

To ensure the best compatibility with users across multiple operating systems, certain characters are not allowed in the filename (`memo:filename`), see table below.

Invalid special characters

`<`	`>`	`:`
`"`	`/`	`\`
`|`	`?`	`*`
`CR`	`LF`	`CRLF`

Invalid white space characters

Unicode	HTML	Description	Example
U+00A0		No-Break Space	[ ]
U+2000		En Quad	[ ]
U+2001		Em Quad	[ ]
U+2002		En Space	[ ]
U+2003		Em Space	[ ]

Unicode	HTML	Description	Example
U+2004		Three-Per-Em Space	[ ]
U+2005		Four-Per-Em Space	[ ]
U+2006		Six-Per-Em Space	[ ]
U+2007		Figure Space	[ ]
U+2008		Punctuation Space	[ ]
U+2009		Thin Space	[ ]
U+200A		Hair Space	[]
U+2028	&#8232	Line Separator	[]
U+205F		Medium Mathematical Space	[ ]
U+2060	&#8288	Word Joiner	(invisible character that prevents a line from breaking)
U+3000		Ideographic Space	[ ]

10.14 HTML whitelist for document validation

Before messages are delivered, both from sender systems and replying through the mailbox. The HTML of the message is validated to ensure that it can be rendered and does not contain any malicious content. Therefore Digital Post implements two different policies for HTML validation:

- Strict
  - A policy for end users that are writing or attaching HTML to a Mailbox Message from a view client
- Lenient
  - Used when validating incoming MeMos before distribution
  - The lenient white list is a super set of the strict one

10.14.1 Strict

- No comments allowed at all
- Only inline styling on elements

Elements	**Attributes** **Inline styling is allowed on all elements that support styling**
Global All elements	
Elements where attribute is relevant	• role • title • aria-hidden • aria-label • aria-level • aria-orientation • aria-placeholder • aria-sort • aria-relevant • aria-activedescendant • aria-colcount • aria-colindex • aria-colspan • aria-describedby • aria-details • aria-labelledby • aria-posinset • aria-rowcount • aria-rowindex • aria-rowspan
Main Main blocks such as `<html>` , `<body>` , etc.	
html	• xmlns • lang
head	

meta	• charset
	• content
	• name
	• http-equiv
	  • content-security-policy
	  • content-type

title	

Elements	**Attributes** **Inline styling is allowed on all elements that support styling**
body	• lang
Semantic Html 5 semantic elements	
address	
article	
aside	
details	
figcaption	
figure	
footer	
header	
main	
mark	
nav	
section	
summary	
time	
Blocks Allow common block elements including `<p>` , `<h1>` , etc.	

Elements	Attributes Inline styling is allowed on all elements that support styling
p	
div	
h1	
h2	
h3	
h4	
h5	
h6	
hr	
ul	
ol	
li	
blockquote	
dl	
dt	
dd	
hr	

Elements	**Attributes** **Inline styling is allowed on all elements that support styling**
Formatting Allows common formatting elements including `<b>` , `<i>` , etc.	
b	
i	
font	color, face, size, optionally without attributes
s	
u	
o	
sup	
sub	
ins	
del	
strong	
strike	
tt	
code	
big	
small	

Elements	**Attributes** **Inline styling is allowed on all elements that support styling**
br	
span	Optionally without attributes
em	
Tables Allow common table elements	
table	• summary • align • valign
tr	• align • valign
td	• align • valign
th	• align • valign
colgroup	• align • valign
caption	
col	• align • valign
thead	• align • valign
tbody	• align • valign
tfoot	• align • valign

Elements	**Attributes** **Inline styling is allowed on all elements that support styling**
Links	

a	• href
	  • https, mailto
	• target
	  • _blank

Images Allow `<img>` elements from from embedded sources only	

img	• alt
	• src
	  • data:image
	• border
	  • Can only be integer number
	• height
	  • Can only be integer number
	• width
	  • Can only be integer number

Styles Allow certain safe CSS properties in `style="..."` attributes. `<style>` element is not allowed	**Properties**

Elements	Attributes **Inline styling is allowed on all elements that support styling**
• -moz-border-radius • -moz-border-radius-bottomleft • -moz-border-radius-bottomright • -moz-border-radius-topleft • -moz-border-radius-topright • -moz-box-shadow • -moz-outline • -moz-outline-color • -moz-outline-style • -moz-outline-width • -o-text-overflow • -webkit-border-bottom-left-radius • -webkit-border-bottom-right-radius • -webkit-border-radius • -webkit-border-radius-bottom-left • -webkit-border-radius-bottom-right • -webkit-border-radius-top-left • -webkit-border-radius-top-right • -webkit-border-top-left-radius • -webkit-border-top-right-radius • -webkit-box-shadow • azimuth • background • background-attachment • background-color • background-image • background-position • background-repeat • border • border-bottom • border-bottom-color • border-bottom-left-radius • border-bottom-right-radius • border-bottom-style • border-bottom-width • border-collapse • border-color • border-left • border-left-color • border-left-style • border-left-width • border-radius • border-right • border-right-color • border-right-style • border-right-width • border-spacing	• url • data uri only • -moz-inline-box, -moz-inline-stack, -moz-pre-wrap, -o-pre-wrap, -pre-wrap, 100, 200, 300, 400, 500, 600, 700, 800, 900, above, absolute, aliceblue, all-scroll, always, antiquewhite, aqua, aquamarine, armenian, at, auto, avoid, azure, baseline, behind, beige, below, bidi-override, bisque, black, blanchedalmond, blink, block, blue, blueviolet, bold, bolder, border-box, both, bottom, break-word, brown, burlywood, cadetblue, capitalize, caption, center, center-left, center-right, chartreuse, child, chocolate, circle, cjk-decimal, clip, closest-corner, closest-side, code, col-resize, collapse, condensed, contain, content-box, continuous, coral, cornflowerblue, cornsilk, cover, crimson, crosshair, cursive, cyan, darkblue, darkcyan, darkgoldenrod, darkgray, darkgreen, darkkhaki, darkmagenta, darkolivegreen, darkorange, darkorchid, darkred, darksalmon, darkseagreen, darkslateblue, darkslategray, darkturquoise, darkviolet, dashed, decimal, decimal-leading-zero, deeppink, deepskyblue, default, digits, dimgray, disc, disclosure-closed, disclosure-open, dodgerblue, dotted, double, e-resize, ellipse, ellipsis, embed, ethiopic-numeric, expanded, extra-condensed, extra-expanded, fantasy, far-left, far-right, farthest-corner, farthest-side, fast, faster, female, firebrick, fixed, floralwhite, forestgreen, fuchsia, gainsboro, georgian, ghostwhite, gold, goldenrod, gray, green, greenyellow, groove, hand, hebrew, help, hidden, hide, high, higher, hiragana, hiragana-iroha, honeydew, hotpink, icon, indianred, indigo, inherit, inline, inline-block, inline-table, inset, inside, invert, italic, ivory, japanese-formal, japanese-informal, justify, katakana, katakana-iroha, khaki, korean-hangul-formal, korean-hanja-formal, korean-hanja-informal, large, larger, lavender, lavenderblush, lawngreen, left, left-side, leftwards, lemonchiffon, level, lightblue, lightcoral, lightcyan, lighter, lightgoldenrodyellow, lightgreen, lightgrey, lightpink, lightsalmon, lightseagreen,

Elements	Attributes **Inline styling is allowed on all elements that support styling**
• border-style • border-top • border-top-color • border-top-left-radius • border-top-right-radius • border-top-style • border-top-width • border-width • box-shadow • caption-side • color • cue • cue-after • cue-before • direction • elevation • empty-cells • font • font-family • font-size • font-stretch • font-style • font-variant • font-weight • height • image() • letter-spacing • line-height • linear-gradient() • list-style • list-style-image • list-style-position • list-style-type • margin • margin-bottom • margin-left • margin-right • margin-top • max-height • max-width • min-height • min-width • outline • outline-color • outline-style • outline-width • padding	lightskyblue, lightslategray, lightsteelblue, lightyellow, lime, limegreen, line-through, linen, list-item, local, loud, low, lower, lower-alpha, lower-greek, lower-latin, lower-roman, lowercase, ltr, magenta, male, maroon, medium, mediumaquamarine, mediumblue, mediumorchid, mediumpurple, mediumseagreen, mediumslateblue, mediumspringgreen, mediumturquoise, mediumvioletred, menu, message-box, middle, midnightblue, mintcream, mistyrose, mix, moccasin, monospace, move, n-resize, narrower, navajowhite, navy, ne-resize, no-content, no-display, no-drop, no-repeat, none, normal, not-allowed, nowrap, nw-resize, oblique, oldlace, olive, olivedrab, once, orange, orangered, orchid, outset, outside, overline, padding-box, palegoldenrod, palegreen, paleturquoise, palevioletred, papayawhip, peachpuff, peru, pink, plum, pointer, powderblue, pre, pre-line, pre-wrap, progress, purple, red, relative, repeat, repeat-x, repeat-y, ridge, right, right-side, rightwards, rosybrown, round, row-resize, royalblue, rtl, run-in, s-resize, saddlebrown, salmon, sandybrown, sans-serif, scroll, se-resize, seagreen, seashell, semi-condensed, semi-expanded, separate, serif, show, sienna, silent, silver, simp-chinese-formal, simp-chinese-informal, skyblue, slateblue, slategray, slow, slower, small, small-caps, small-caption, smaller, snow, soft, solid, space, spell-out, springgreen, square, static, status-bar, steelblue, sub, super, suppress, sw-resize, table, table-caption, table-cell, table-column, table-column-group, table-footer-group, table-header-group, table-row, table-row-group, tan, teal, text, text-bottom, text-top, thick, thin, thistle, to, tomato, top, trad-chinese-formal, trad-chinese-informal, transparent, turquoise, ultra-condensed, ultra-expanded, underline, unrestricted, upper-alpha, upper-latin, upper-roman, uppercase, vertical-text, violet, visible, w-resize, wait, wheat, white, whitesmoke, wider, x-fast, x-high, x-large, x-loud, x-low, x-slow, x-small, x-soft, xx-large, xx-small, yellow, yellowgreen

Elements	Attributes **Inline styling is allowed on all elements that support styling**
• padding-bottom • padding-left • padding-right • padding-top • pause • pause-after • pause-before • pitch • pitch-range • quotes • radial-gradient() • rect() • repeating-linear-gradient() • repeating-radial-gradient() • rgb() • rgba() • richness • speak • speak-header • speak-numeral • speak-punctuation • speech-rate • stress • table-layout • text-align • text-decoration • text-indent • text-overflow • text-shadow • text-transform • text-wrap • unicode-bidi • vertical-align • voice-family • volume • white-space • width • word-spacing • word-wrap	

10.14.2 Lenient

All allowed in the Strict policy are allowed here, plus the following elements:

- Comments allowed
- Global styling element allowed with no restrictions other than URLs referencing http/https are blocked
- Element style attribute with no restrictions other than URLs referencing http/https are blocked

- Attributes id, class allowed on all supported elements

Elements	Attributes
Global All elements	
Elements where attribute is relevant	• id • class • lang • aria-setsize • aria-busy • aria-atomic • aria-controls • aria-current • aria-description • aria-disabled • aria-errormessage • aria-flowto • aria-haspopup • aria-invalid • aria-keyshortcuts • aria-live • aria-owns • aria-roledescription
Main Further options on elements such as `<html>` , `<body>` , etc.	
html	• xmlns:v • xmlns:o • xmlns:w • xmlns:m
body	• link • vlink
Blocks Further options on block elements	
span	
p	• align

Elements	Attributes
o:p	
div	• align
hr	• size • width • align
picture	

source	• srcset
	  • data:image
	• src
	  • data:image
	• media
	• type

pre	
cite	
ol	• type • start
ul	• type
Links	
a	• name
Tables	
table	• border • cellspacing • cellpadding • width

Elements	Attributes
td	• scope • headers • colspan • width • rowspan • nowrap • height
th	• scope • headers • colspan • width • rowspan • nowrap • height
colgroup	• width
col	• width • height • span
Styles	
style	Global style tag allowed. Local style attribute unrestricted. Only URLs to external resources are blocked.
Pictures	
picture	

source	• scrset
	  • Has to start with `data:image/`
	• src
	  • Has to start with `data:image/`
	• media
	• type

10.14.3 Testing

HTML content can be tested against the white list validator using endpoint:

- `/validations/`
  - Looks like this on the test environment: `curl --location --request POST 'https://api.test.digitalpost.dk/apis/v1/validations/' --header 'Content-Type: text/html'`

POST request with content-type: text/html and request body with HTML, will return 200 OK with a response body containing code `html.validator.approved`, if it finds no validation errors in the HTML:

```
{
    "code": "html.validator.approved",
    "message": "Approved: Html validation using NgDP whitelist - LENIENT policy found
0 errors.",
    "fieldErrors": []
}
```

If it finds validation errors, it returns 400 BAD REQUEST including a response body with code `html.validator.rejected`, and a list of errors:

```
{
    "code": "html.validator.rejected",
    "message": "Rejected: HTML validation using NgDP whitelist - LENIENT policy -
found 1 errors.",
    "fieldErrors": [
        {
            "resource": "errorMessage",
            "code": "html.validator.rejected.element",
            "message": "Filen test indeholder element \"link\", som enten ikke
tilladt eller som indeholder data, der ikke er tilladt."
        }
    ]
}
```

Policy

The default policy used is LENIENT. The policy can be switched using request parameter `policy`:

- `/validations/?policy=STRICT`
  - Looks like this on the test environment: `curl --location --request POST 'https://api.test.digitalpost.dk/apis/v1/validations/?policy=STRICT' --header 'Content-Type: text/html'`

11 Access request registry

See “Access request registry - introduction” for a short description of the different request types.

The following services are exposed from the access-request.

Service	URL	Usage	Required roles	OpenAPI
Query access requests	`GET /access-requests/`	Fetching all access-requests user has access to with optional paging, sorting and filtering.	• Employee • Citizen • System manager	Swagger UI
Create access request	`POST /access-requests/`	Creating access request	• Employee • Citizen • System manager	Swagger UI
Update access request	`PUT /access-requests/{id}`	Updating access request	• Employee • Citizen • System manager	Swagger UI
Delete access request	`DELETE /access-requests/{id}`	Deleting access request	• Employee • Citizen • System manager	Swagger UI
Fetch access request	`GET /access-requests/{id}`	Fetching single access request	• Employee • Citizen • System manager	Swagger UI
Update documentation content	`PUT /access-requests/{accessRequestId}/documentations/{id}/content`	Update the contents of a documentation	• Employee • Citizen • System manager	Swagger UI

Service	URL	Usage	Required roles	OpenAPI
Fetch documentation content	`GET /access-requests/{accessRequestId}/documentations/{id}/content`	Fetching documentation content bytes	• Employee • Citizen • System manager	Swagger UI
Activate access request target	`POST /access-requests/target/activate`	Activating a target using activation code	• Employee	Swagger UI
Fetch organisation lookup	`GET /organisation-lookup/` • `cvrNumber`	Fetching information about the organisation, such as name status active curator, liquidator and executor	• Citizen • Employee • Delegated Support Admin • Citizen Service Employee • ERST Service Employee • First Level Support • Organisation Admin • Oraganisation User Admin • System Manager	Swagger UI
Citizen lookup	`POST /citizen-lookup/` Which uses the following parameters in the body: • `cprNumber` • `firstNameProvided` • `lastNameProvided`	Fetching information about a citizen: identityId, full name,, and if citizen's status is CLOSED	• Citizen • Employee • Delegated Support Admin • Citizen Service Employee • ERST Service Employee • First Level Support • Organisation Admin • Oraganisation User Admin • System Manager	Swagger UI

11.1 Access request registry - introduction

- Purpose of the registry
- Privilege requests
- Delegation requests
- Appointed delegation requests
- Connection agreement requests
- Terms approval requests
- User administrator’s statement of truth privilege requests
- Lost user administrator privilege requests
- Special privilege requests
- Delegated support admin privilege request
- Third party intermediary request
- Concepts

11.2 Purpose of the registry

The access request registry is the request management store of DP. It stores access requests and exposes services to handle these. Eleven different types of access requests are currently supported and they are all represented as one AccessRequest resource but marked as one of these types:

- Privilege requests
- Delegation requests
- Appointed delegation requests
- Connection agreement requests
- Terms approval requests
- User administrator’s statement of truth privilege requests
- Lost user administrator privilege requests
- Special privilege requests
- Legal owner of inactive or closed company privilege request
- Delegated support admin privilege request
- Third party intermediary request

11.3 Privilege requests

A privilege request is filed by either a citizen or an organisation requesting a privilege to a mailbox or an organisation. In certain cases a request can also be triggered by an event elsewhere in DP. A privilege request consists of information about the target of the request and the requested privilege. To assist in the processing, documents may be attached to the request. Examples of scenarios:

- A new employee requesting a privilege (or membership of user group) to own organisation
- A citizen or organisation requesting a ‘executor of estate’ privilege to a deceased citizen's mailbox
- A citizen or organisation requesting party representative privileges to another person's or citizen's mailbox
- A system manager that request on behalf of a citizen or organisation

A privilege request will upon submission end up on a list of incoming requests presented to:

- The citizen owning the mailbox to which access is requested
- The user administrator of the organisation to which access is requested
- A system manager handling the special cases like for instance curators and liquidators etc.

The access can be granted or rejected and the requesting party will be notified of the decision. If access is granted the privileges will be created in the identity registry. If the target of the privilege is unknown to the system an email with activation code and instructions will be sent.

11.4 Delegation requests

Delegation requests are filed by a citizen or an organisation granting one or more privileges to their own mailbox or organisation to another employee or citizen - delegating, so to speak, the privileges. A user group may be used instead of or on combination with privileges. Examples of scenarios:

- A citizen delegating legal representative privileges to another citizen, to an organisation or to a specific employee of that organisation
- A citizen (A) “trusting“ another citizen (B) so messages can be forwarded from B to A
- A user administrator of an organisation delegating a privilege to a new employee of her own organisation
- A user administrator of an organisation delegating a read only access to another organisation or to a specific employee of that organisation

These requests are automatically approved and privileges assigned immediately if target is known, if not an activation code sent instead.

11.5 Appointed delegation requests

The same principle as regular delegation requests, but here a user administrator can delegate privileges to the organisation's employees, that give access to another organisation or a citizen, if those privileges have been appointed to the user administrator’s organisation.

11.6 Connection agreement requests

Request filed by an authority to request access to DP. It requires a physical signature on a document, and manual approval. Upon approval the organisation changes type in the system registry (from COMPANY to AUTHORITY) and is then ready to approve the terms.

11.7 Terms approval requests

Request filed by an authority to request access to DP. It is automatically approved if organisation is correctly registered and an activation code sent to the user administrator organisation by email.

11.8 User administrator’s statement of truth privilege requests

Uses registration type code from system registry to categorize the requester. If within correct category, the request is automatically approved and NPTE privileges are granted to the requester.

11.9 Lost user administrator privilege requests

Can be requested by supporters (with the role FO_SUPPORT) on behalf of an organisation with no access to their user administrator. Automatically approved. The request must always contain rights administrator (`ORGANISATION_USER_ADMINISTRATOR`), but it can also contain two optional privileges:

- Basic access (`MESSAGE_BASIC`)
- Advanced access (`MESSAGE_EMPLOYEE`)

11.10 Special privilege requests

Like a regular privilege request in it’s structure, but only regarding the special privileges:

- Curator
- Executor of estate
- Liquidator

The access can be granted or rejected, by system managers and access request registry administrators and the requesting party will be notified of the decision. If access is granted the privileges will be created in the identity registry.

When a special privilege request includes **curator** privileges for a citizen, a partial mailblock is applied to that citizen upon approval. This means that the Contact object of the citizen is active, but the Mailbox is inactive. For this type of access, the `sentencingDateTime` field on the privilege request must be provided to specify both the start date of the mailblock and the date from which the curator is permitted to read the messages in the citizen's mailbox.

11.11 Delegated support admin privilege request

A delegated support admin privilege request is filed by a DSS support employee requesting privilege to a company or an authority which requests support assistance. The request is auto-approved and results in creation of 2 privileges in the identity registry:

- An appointed privilege a grantee of which is the company Erhvervsstyrelsen (ERST)
- A privilege with DSS support employee as grantee and a privilege appointed to ERST as a parent privilege

The scope of both privileges is a company or authority requesting support assistance and their expiration is set to 30 minutes. The privileges can be revoked earlier when the support employee revokes the privileges manually.

A support employee can have only one open delegated support admin privilege request, so if a support employee requests another one, the currently open one will be revoked.

11.12 Third party intermediary request

A type of access request used for supporting a third party (that can only be created by Citizen Service Employee and System Manager) granting access between two independent parties. When requests of this type are submitted, they are automatically approved and the authorization is immediately granted to the access target. This type of request supports the creation of the privilege for read/read-write access.

When this type of request is approved a Memo/receipt is sent to access target and citizen or organization that the access has been granted to.

11.13 Concepts

A request is a request regarding either privileges or user groups. User groups are only allowed on delegation request. A request can consist of multiple privileges or groups. A request consists of 3 participants:

- requester
    - The participant initiating the request. When delegating, typically a user administrator or a citizen. When requesting privileges typically an employee or a citizen.
    - Although a system manager may create a request on behalf of an organisation, it is still the organisation or citizen that goes here, as they are the actual requester.
- accessTo
    - The scope of the request, which is an organisation or a citizen. If granted the target will get some sort of access to this participant.
- target
    - Who is the recipient of the privileges

In many cases two of these will be the exact same participant, but the registry insists on a complete registration of all three participants. This provides the consistency needed when dealing with a rather flexible API, such as this.

11.14 Access request registry - common use case examples

All request can be created in state DRAFT. In DRAFT state the requests can be worked on and refined until the user deems the request ready for submission. A request will still be validated in DRAFT state and name resolving etc. will also occur.

> DRAFT state can be skipped and a request can be submitted directly in state SUBMITTED, but DRAFT state is required if you need to attach documents to the requests as content cannot be added/changed after submission.

A request in state DRAFT can be read, edited, and deleted following these rules:

Request created by	Users who can read, edit, delete the request
Citizen	Only the exact same citizen
User administrator	User administrators of same organisation
Employee	User administrators of same organisation
System manager and “Special“ request	All system managers

11.15 User administrator delegates privilege to employee identified with an e-mail address

11.15.1 Request

```
{
    "requestType": "DELEGATION_REQUEST",
    "requestState": "DRAFT",
    "requester": {
        "externalId": "30714024",
        "externalIdType": "CVR"
    },
    "accessTo": {
        "externalId": "30714024",
        "externalIdType": "CVR"
    },
    "privileges": [
        "MESSAGE_EMPLOYEE"
    ],
    "userGroups": [],
    "target": {
```

```
        "requestParticipant": {
            "externalId": "30714024",
            "externalIdType": "CVR",
            "emailAddress": "clan@netcompany.com"
        }
    }
}
```

11.15.2 Response

201 CREATED

Notice how external ids have been accompanied by the identity ids, and the user performing the request have been inserted into `employeeIdentityId`.

```
{
    "id": "1ba3a286-bb03-43ac-b874-e21a5bdee636",
    "version": 0,
    "transactionId": "F6DAVnsfzRF0HbqUJ67yJMsUAtV15MoG",
    "createdDateTime": "2021-07-18T14:23:47.805Z",
    "lastUpdated": "2021-07-18T14:23:47.805Z",
    "requestType": "DELEGATION_REQUEST",
    "requestState": "DRAFT",
    "requester": {
        "id": "0ddd0ad1-cc43-45c3-bb9c-2dacd574a4e9",
        "version": 0,
        "createdDateTime": "2021-07-18T14:23:47.813Z",
        "lastUpdated": "2021-07-18T14:23:47.813Z",
        "externalId": "30714024",
        "externalIdType": "CVR",
        "organisationIdentityId": "6337c256-133a-46bb-b85c-d65eb751b37f",
        "employeeIdentityId": "c810f086-5331-4336-bb4b-47dbdc27f3d7"
    },
    "accessTo": {
        "id": "b45b6e83-5086-469f-87f0-dff6875de338",
        "version": 0,
        "createdDateTime": "2021-07-18T14:23:47.807Z",
        "lastUpdated": "2021-07-18T14:23:47.807Z",
        "externalId": "30714024",
        "externalIdType": "CVR",
        "organisationIdentityId": "6337c256-133a-46bb-b85c-d65eb751b37f"
    },
    "privileges": [
        "MESSAGE_EMPLOYEE"
    ],
    "userGroups": [],
    "createdByIdentityId": "c810f086-5331-4336-bb4b-47dbdc27f3d7",
    "createdOnBehalfOf": false,
    "target": {
        "id": "8330cbfe-158d-47f5-9250-d6f47eefbc59",
        "version": 0,
```

```
        "createdDateTime": "2021-07-18T14:23:47.813Z",
        "lastUpdated": "2021-07-18T14:23:47.813Z",
        "requestParticipant": {
            "id": "919a73d9-b27d-4f4d-b309-ed40bf4f9b13",
            "version": 0,
            "createdDateTime": "2021-07-18T14:23:47.814Z",
            "lastUpdated": "2021-07-18T14:23:47.814Z",
            "externalId": "30714024",
            "externalIdType": "CVR",
            "organisationIdentityId": "6337c256-133a-46bb-b85c-d65eb751b37f",
            "emailAddress": "clan@netcompany.com"
        }
    },
    "documentations": [],
    "processedAutomatically": false
}
```


11.16 Usage of generic identity id in access requests

A generic identity id is a part of the requestParticipant in the access request. By using the generic identity the same flow is followed as when one of the identity type specific IDs is used (citizenIdentityId, employeeIdentityId or OrganisationIdentityId), however the solution will determine the type of the identity id. Furthermore, it will provide af field called identityTypeResolved in the response. When the identity type is determine from the generic identity id one or several of the identity type specific IDs will in the response alongside with the generic identity id.

11.16.1 Request

```
{
    "requestType": "DELEGATION_REQUEST",
    "requestState": "DRAFT",
    "requester": {
        "identityId": "7f79954b-ded0-4af3-8a56-ca69791e4685"
    },
    "accessTo": {
        "identityId": "462829eb-26c3-48a5-8020-10166a15c976"
    },
    "privileges": [
        "MESSAGE_EMPLOYEE"
    ],
    "userGroups": [],
    "target": {
        "requestParticipant": {
            "externalId": "30714024",
            "externalIdType": "CVR",
            "emailAddress": "clan@netcompany.com"
        }
    }
}
```

11.16.2 Response


```
{
    "id": "1ba3a286-bb03-43ac-b874-e21a5bdee636",
    "version": 0,
    "transactionId": "F6DAVnsfzRF0HbqUJ67yJMsUAtV15MoG",
    "createdDateTime": "2021-07-18T14:23:47.805Z",
    "lastUpdated": "2021-07-18T14:23:47.805Z",
    "requestType": "DELEGATION_REQUEST",
    "requestState": "DRAFT",
    "requester": {
        "id": "0ddd0ad1-cc43-45c3-bb9c-2dacd574a4e9",
        "version": 0,
        "createdDateTime": "2021-07-18T14:23:47.813Z",
        "lastUpdated": "2021-07-18T14:23:47.813Z",
        "externalId": "30714024",
        "externalIdType": "CVR",
        "organisationIdentityId": "6337c256-133a-46bb-b85c-d65eb751b37f",
        "employeeIdentityId": "7f79954b-ded0-4af3-8a56-ca69791e4685",
        "identityId": "7f79954b-ded0-4af3-8a56-ca69791e4685",
        "identityTypeResolved": "EMPLOYEE"
    },
    "accessTo": {
        "id": "b45b6e83-5086-469f-87f0-dff6875de338",
        "version": 0,
        "createdDateTime": "2021-07-18T14:23:47.807Z",
        "lastUpdated": "2021-07-18T14:23:47.807Z",
        "externalId": "2412001010",
        "externalIdType": "CPR",
        "citizenIdentityId": "462829eb-26c3-48a5-8020-10166a15c976"
        "identityId": "462829eb-26c3-48a5-8020-10166a15c976",
        "identityTypeResolved": "CITIZEN"
    },
    "privileges": [
        "MESSAGE_EMPLOYEE"
    ],
    "userGroups": [],
    "createdByIdentityId": "c810f086-5331-4336-bb4b-47dbdc27f3d7",
    "createdOnBehalfOf": false,
    "target": {
        "id": "8330cbfe-158d-47f5-9250-d6f47eefbc59",
        "version": 0,
        "createdDateTime": "2021-07-18T14:23:47.813Z",
        "lastUpdated": "2021-07-18T14:23:47.813Z",
        "requestParticipant": {
            "id": "919a73d9-b27d-4f4d-b309-ed40bf4f9b13",
            "version": 0,
            "createdDateTime": "2021-07-18T14:23:47.814Z",
            "lastUpdated": "2021-07-18T14:23:47.814Z",
            "externalId": "30714024",
            "externalIdType": "CVR",
            "organisationIdentityId": "6337c256-133a-46bb-b85c-d65eb751b37f",
```

```
            "emailAddress": "clan@netcompany.com"
        }
    },
    "documentations": [],
    "processedAutomatically": false
}
```

11.16.3 Attaching documents to request

Documents can be attached to a DRAFT request and it follows a 2-step approach.

11.17 1. Add documentation element

First we need to tell the resource what documents will be attached. This goes into the `"documentations"` list of the resource. It can be present at creation (POST) or they can be added removed, switched around, or edited using PUT.

When adding a “documentation” element, the only required field is `documentationType.` We can choose from a list of predefined types, that can be seen in the OpenApi documentation:

Current version can be found here: https://test.digitalpost.dk/api

Let us add a BUSINESS_CARD to the request we created.

11.17.1 Request

PUT `/access-requests/1ba3a286-bb03-43ac-b874-e21a5bdee636`

```
{
    "requestType": "DELEGATION_REQUEST",
```

```
    "requestState": "DRAFT",
    "requester": {
        "id": "0ddd0ad1-cc43-45c3-bb9c-2dacd574a4e9",
        "version": 0,
        "createdDateTime": "2021-07-18T14:23:47.813Z",
        "lastUpdated": "2021-07-18T14:23:47.813Z",
        "externalId": "30714024",
        "externalIdType": "CVR",
        "organisationIdentityId": "6337c256-133a-46bb-b85c-d65eb751b37f",
        "employeeIdentityId": "c810f086-5331-4336-bb4b-47dbdc27f3d7"
    },
    "accessTo": {
        "id": "b45b6e83-5086-469f-87f0-dff6875de338",
        "version": 0,
        "createdDateTime": "2021-07-18T14:23:47.807Z",
        "lastUpdated": "2021-07-18T14:23:47.807Z",
        "externalId": "30714024",
        "externalIdType": "CVR",
        "organisationIdentityId": "6337c256-133a-46bb-b85c-d65eb751b37f"
    },
    "privileges": [
        "MESSAGE_EMPLOYEE"
    ],
    "target": {
        "id": "8330cbfe-158d-47f5-9250-d6f47eefbc59",
        "version": 0,
        "createdDateTime": "2021-07-18T14:23:47.813Z",
        "lastUpdated": "2021-07-18T14:23:47.813Z",
        "requestParticipant": {
            "id": "919a73d9-b27d-4f4d-b309-ed40bf4f9b13",
            "version": 0,
            "createdDateTime": "2021-07-18T14:23:47.814Z",
            "lastUpdated": "2021-07-18T14:23:47.814Z",
            "externalId": "30714024",
            "externalIdType": "CVR",
            "organisationIdentityId": "6337c256-133a-46bb-b85c-d65eb751b37f",
            "emailAddress": "clan@netcompany.com"
        }
    },
    "documentations": [
        {
            "documentationType": "BUSINESS_CARD"
        }
    ]
}
```


11.17.2 Response


```
{
    "id": "1ba3a286-bb03-43ac-b874-e21a5bdee636",
    "version": 1,
```

```
"transactionId": "F6DDqxoMf52quFkwgdfTUsLJLurGqHGD",
"createdDateTime": "2021-07-18T14:23:47.805Z",
"lastUpdated": "2021-07-18T14:48:58.049Z",
"requestType": "DELEGATION_REQUEST",
"requestState": "DRAFT",
"requester": {
    "id": "0ddd0ad1-cc43-45c3-bb9c-2dacd574a4e9",
    "version": 0,
    "createdDateTime": "2021-07-18T14:23:47.813Z",
    "lastUpdated": "2021-07-18T14:23:47.813Z",
    "externalId": "30714024",
    "externalIdType": "CVR",
    "organisationIdentityId": "6337c256-133a-46bb-b85c-d65eb751b37f",
    "employeeIdentityId": "c810f086-5331-4336-bb4b-47dbdc27f3d7"
},
"accessTo": {
    "id": "b45b6e83-5086-469f-87f0-dff6875de338",
    "version": 0,
    "createdDateTime": "2021-07-18T14:23:47.807Z",
    "lastUpdated": "2021-07-18T14:23:47.807Z",
    "externalId": "30714024",
    "externalIdType": "CVR",
    "organisationIdentityId": "6337c256-133a-46bb-b85c-d65eb751b37f"
},
"privileges": [
    "MESSAGE_EMPLOYEE"
],
"userGroups": [],
"createdByIdentityId": "c810f086-5331-4336-bb4b-47dbdc27f3d7",
"createdOnBehalfOf": false,
"target": {
    "id": "8330cbfe-158d-47f5-9250-d6f47eefbc59",
    "version": 0,
    "createdDateTime": "2021-07-18T14:23:47.813Z",
    "lastUpdated": "2021-07-18T14:23:47.813Z",
    "requestParticipant": {
        "id": "919a73d9-b27d-4f4d-b309-ed40bf4f9b13",
        "version": 0,
        "createdDateTime": "2021-07-18T14:23:47.814Z",
        "lastUpdated": "2021-07-18T14:23:47.814Z",
        "externalId": "30714024",
        "externalIdType": "CVR",
        "organisationIdentityId": "6337c256-133a-46bb-b85c-d65eb751b37f",
        "emailAddress": "clan@netcompany.com"
    }
},
"documentations": [
    {
        "id": "4a583058-84f4-4491-ba0a-bcc80643713c",
        "version": 0,
        "createdDateTime": "2021-07-18T14:48:58.047Z",
        "lastUpdated": "2021-07-18T14:48:58.047Z",
        "documentationType": "BUSINESS_CARD"
```

```
        }
    ],
    "processedAutomatically": false
}
```

Notice the documentation element:

```
        {
            "id": "4a583058-84f4-4491-ba0a-bcc80643713c",
            "version": 0,
            "createdDateTime": "2021-07-18T14:48:58.047Z",
            "lastUpdated": "2021-07-18T14:48:58.047Z",
            "documentationType": "BUSINESS_CARD"
        }
```

We will need the assigned id to add the byte content to the attachment, which is the second step.

11.18 2. Upload byte content to documentation

It is done using a PUT multipart request.

11.18.1 Request

PUT `/access-requests/1ba3a286-bb03-43ac-b874-e21a5bdee636/documentations/4a583058-84f4-4491-ba0a-bcc80643713c/content`

Notice the documentation id in the URL. The If-Match header must be set to the version of the documentation element.

The name of the multipart form element must be 'file'. Here is a Curl example:

```
curl --location --request PUT 'https://dev.digdp.nchosting.dk/apis/v1/access-
requests/1ba3a286-bb03-43ac-b874-e21a5bdee636/documentations/4a583058-84f4-4491-ba0a-
bcc80643713c/content' \
--header 'If-Match: 0' \
--header 'Authorization: Bearer eyJh...' \
--form 'file=@"../business-card.pdf"'
```

The client must NOT base64 encode the file content. The maximum size of the file is 10 MB.

Allowed file types are: application/pdf, image/png, image/jpeg.

11.18.2 Response

```
{
    "id": "1ba3a286-bb03-43ac-b874-e21a5bdee636",
    "version": 2,
    "transactionId": "F6DGzMEQshbb6c3RXtB7J9DM7zZcWkSd",
    "createdDateTime": "2021-07-18T14:23:47.805Z",
    "lastUpdated": "2021-07-18T15:12:35.222Z",
```

```
"requestType": "DELEGATION_REQUEST",
"requestState": "DRAFT",
"requester": {
    "id": "0ddd0ad1-cc43-45c3-bb9c-2dacd574a4e9",
    "version": 0,
    "createdDateTime": "2021-07-18T14:23:47.813Z",
    "lastUpdated": "2021-07-18T14:23:47.813Z",
    "externalId": "30714024",
    "externalIdType": "CVR",
    "organisationIdentityId": "6337c256-133a-46bb-b85c-d65eb751b37f",
    "employeeIdentityId": "c810f086-5331-4336-bb4b-47dbdc27f3d7"
},
"accessTo": {
    "id": "b45b6e83-5086-469f-87f0-dff6875de338",
    "version": 0,
    "createdDateTime": "2021-07-18T14:23:47.807Z",
    "lastUpdated": "2021-07-18T14:23:47.807Z",
    "externalId": "30714024",
    "externalIdType": "CVR",
    "organisationIdentityId": "6337c256-133a-46bb-b85c-d65eb751b37f"
},
"privileges": [
    "MESSAGE_EMPLOYEE"
],
"userGroups": [],
"createdByIdentityId": "c810f086-5331-4336-bb4b-47dbdc27f3d7",
"createdOnBehalfOf": false,
"target": {
    "id": "8330cbfe-158d-47f5-9250-d6f47eefbc59",
    "version": 0,
    "createdDateTime": "2021-07-18T14:23:47.813Z",
    "lastUpdated": "2021-07-18T14:23:47.813Z",
    "requestParticipant": {
        "id": "919a73d9-b27d-4f4d-b309-ed40bf4f9b13",
        "version": 0,
        "createdDateTime": "2021-07-18T14:23:47.814Z",
        "lastUpdated": "2021-07-18T14:23:47.814Z",
        "externalId": "30714024",
        "externalIdType": "CVR",
        "organisationIdentityId": "6337c256-133a-46bb-b85c-d65eb751b37f",
        "emailAddress": "clan@netcompany.com"
    }
},
"documentations": [
    {
        "id": "4a583058-84f4-4491-ba0a-bcc80643713c",
        "version": 1,
        "createdDateTime": "2021-07-18T14:48:58.047Z",
        "lastUpdated": "2021-07-18T15:12:35.223Z",
        "mediaType": "application/pdf",
        "filename": "business-card.pdf",
        "documentationType": "BUSINESS_CARD",
        "size": 71929
```

```
        }
    ],
    "processedAutomatically": false
}
```

11.18.3 Citizen requesting access to other citizens mailbox

A citizen, A, requesting `LEGAL_REPRESENTATIVE` to another citizen, B. `requester` and `target` is the citizen A who is requesting the privilege. `accessTo` is the citizen B that must approve and grant the request.

11.19 1. Citizen A submits request

11.19.1 Request

POST `/access-requests/`

```
{
    "requestType": "PRIVILEGE_REQUEST",
    "requestState": "SUBMITTED",
    "requester": {
        "citizenIdentityId": "dab86db0-98c0-4062-bb70-39bdb3bd9cab"
    },
    "accessTo": {
        "externalId": "0610749832",
        "externalIdType": "CPR",
        "firstNameProvided": "Lone",
        "lastNameProvided": "Boaikvopg"
    },
    "privileges": [
        "LEGAL_REPRESENTATIVE"
    ],
    "target": {
        "requestParticipant": {
            "citizenIdentityId": "dab86db0-98c0-4062-bb70-39bdb3bd9cab"
        }
    }
}
```

11.19.2 Response

```
{
    "id": "02a307ca-227f-445b-8176-46ad04eb802e",
    "version": 1,
    "transactionId": "F7MjMnGST1vbcsqxf6AZnU9CbWoyEoRh",
    "createdDateTime": "2021-08-10T19:27:17.290Z",
    "lastUpdated": "2021-08-10T19:27:19.538Z",
```

```
"requestType": "PRIVILEGE_REQUEST",
"requestState": "SUBMITTED",
"requester": {
    "id": "5ec74ef8-d22b-40e4-94ec-ef1262dd3d3c",
    "version": 0,
    "createdDateTime": "2021-08-10T19:27:17.291Z",
    "lastUpdated": "2021-08-10T19:27:17.291Z",
    "externalId": "0610742121",
    "externalIdType": "CPR",
    "citizenIdentityId": "dab86db0-98c0-4062-bb70-39bdb3bd9cab"
},
"accessTo": {
    "id": "eeccfad4-e150-4a87-bd77-373eaf5dfc09",
    "version": 0,
    "createdDateTime": "2021-08-10T19:27:17.290Z",
    "lastUpdated": "2021-08-10T19:27:17.290Z",
    "externalId": "0610749832",
    "externalIdType": "CPR",
    "citizenIdentityId": "44ba8c4a-cc01-4f7e-bf4c-8188f9203f28",
    "firstNameProvided": "Lone",
    "lastNameProvided": "Boaikvopg",
    "firstNameResolved": "Lone Boaikvopg"
},
"privileges": [
    "LEGAL_REPRESENTATIVE"
],
"userGroups": [],
"createdByIdentityId": "dab86db0-98c0-4062-bb70-39bdb3bd9cab",
"createdOnBehalfOf": false,
"target": {
    "id": "2bf85948-26c1-40f4-a0fc-f0d22a7e8a90",
    "version": 0,
    "createdDateTime": "2021-08-10T19:27:17.291Z",
    "lastUpdated": "2021-08-10T19:27:17.291Z",
    "requestParticipant": {
        "id": "a6a3291d-ff6d-4b08-8929-61adc0b05c91",
        "version": 0,
        "createdDateTime": "2021-08-10T19:27:17.291Z",
        "lastUpdated": "2021-08-10T19:27:17.291Z",
        "externalId": "0610742121",
        "externalIdType": "CPR",
        "citizenIdentityId": "dab86db0-98c0-4062-bb70-39bdb3bd9cab"
    }
},
"documentations": [
    {
        "id": "b6b27ed1-158b-4a43-9949-b916c4d0fa3f",
        "version": 0,
        "createdDateTime": "2021-08-10T19:27:19.537Z",
        "lastUpdated": "2021-08-10T19:27:19.537Z",
        "mediaType": "application/xml",
        "filename": "generic",
        "documentationType": "ACCESS_REQUEST_MESSAGE_MEMO",
```

```
            "size": 2347,
            "scanState": "NOT_INFECTED"
        },
        {
            "id": "1b1717f8-f707-456d-9fec-7c083d82bf18",
            "version": 0,
            "createdDateTime": "2021-08-10T19:27:19.538Z",
            "lastUpdated": "2021-08-10T19:27:19.538Z",
            "mediaType": "application/xml",
            "filename": "generic",
            "documentationType": "ACCESS_REQUEST_MESSAGE_MEMO",
            "size": 2347,
            "scanState": "NOT_INFECTED"
        }
    ],
    "processedAutomatically": false,
    "submissionDateTime": "2021-08-10T19:27:17.303Z",
    "submittedByIdentityId": "dab86db0-98c0-4062-bb70-39bdb3bd9cab"
}
```


11.20 2. Citizen B queries to see incoming requests

Once submitted it will turn up when citizen B queries.

11.20.1 Request

GET `/access-requests/`

11.20.2 Response


```
{
    "currentPage": 0,
    "totalPages": 1,
    "elementsOnPage": 1,
    "totalElements": 1,
    "accessRequests": [
        {
            "id": "02a307ca-227f-445b-8176-46ad04eb802e",
            "version": 1,
            "transactionId": "F7MjMnGST1vbcsqxf6AZnU9CbWoyEoRh",
            "createdDateTime": "2021-08-10T19:27:17.290Z",
            "lastUpdated": "2021-08-10T19:27:19.538Z",
            "requestType": "PRIVILEGE_REQUEST",
            "requestState": "SUBMITTED",
            "requester": {
                "id": "5ec74ef8-d22b-40e4-94ec-ef1262dd3d3c",
                "version": 0,
                "createdDateTime": "2021-08-10T19:27:17.291Z",
                "lastUpdated": "2021-08-10T19:27:17.291Z",
                "externalId": "0610742121",
```

```
                "externalIdType": "CPR",
                "citizenIdentityId": "dab86db0-98c0-4062-bb70-39bdb3bd9cab"
            },
            "accessTo": {
                "id": "eeccfad4-e150-4a87-bd77-373eaf5dfc09",
                "version": 0,
                "createdDateTime": "2021-08-10T19:27:17.290Z",
                "lastUpdated": "2021-08-10T19:27:17.290Z",
                "externalId": "0610749832",
                "externalIdType": "CPR",
                "citizenIdentityId": "44ba8c4a-cc01-4f7e-bf4c-8188f9203f28",
                "firstNameProvided": "Lone",
                "lastNameProvided": "Boaikvopg",
                "firstNameResolved": "Lone Boaikvopg"
            },
            "privileges": [
                "LEGAL_REPRESENTATIVE"
            ],
            "userGroups": [],
            "createdByIdentityId": "dab86db0-98c0-4062-bb70-39bdb3bd9cab",
            "createdOnBehalfOf": false,
            "target": {
                "id": "2bf85948-26c1-40f4-a0fc-f0d22a7e8a90",
                "version": 0,
                "createdDateTime": "2021-08-10T19:27:17.291Z",
                "lastUpdated": "2021-08-10T19:27:17.291Z",
                "requestParticipant": {
                    "id": "a6a3291d-ff6d-4b08-8929-61adc0b05c91",
                    "version": 0,
                    "createdDateTime": "2021-08-10T19:27:17.291Z",
                    "lastUpdated": "2021-08-10T19:27:17.291Z",
                    "externalId": "0610742121",
                    "externalIdType": "CPR",
                    "citizenIdentityId": "dab86db0-98c0-4062-bb70-39bdb3bd9cab"
                }
            },
            "documentations": [],
            "processedAutomatically": false,
            "submissionDateTime": "2021-08-10T19:27:17.303Z",
            "submittedByIdentityId": "dab86db0-98c0-4062-bb70-39bdb3bd9cab"
        }
    ]
}
```


11.21 3. Citizen B approves request

Citizen B approves the request by changing the `requestState` to `APPROVED`.

11.21.1 Request

PUT `/access-requests/02a307ca-227f-445b-8176-46ad04eb802e`

```
{
    "requestType": "PRIVILEGE_REQUEST",
    "requestState": "APPROVED",
    "requester": {
        "id": "5ec74ef8-d22b-40e4-94ec-ef1262dd3d3c",
        "version": 0,
        "createdDateTime": "2021-08-10T19:27:17.291Z",
        "lastUpdated": "2021-08-10T19:27:17.291Z",
        "externalId": "0610742121",
        "externalIdType": "CPR",
        "citizenIdentityId": "dab86db0-98c0-4062-bb70-39bdb3bd9cab"
    },
    "accessTo": {
        "id": "eeccfad4-e150-4a87-bd77-373eaf5dfc09",
        "version": 0,
        "createdDateTime": "2021-08-10T19:27:17.290Z",
        "lastUpdated": "2021-08-10T19:27:17.290Z",
        "externalId": "0610749832",
        "externalIdType": "CPR",
        "citizenIdentityId": "44ba8c4a-cc01-4f7e-bf4c-8188f9203f28",
        "firstNameProvided": "Lone",
        "lastNameProvided": "Boaikvopg",
        "firstNameResolved": "Lone Boaikvopg"
    },
    "privileges": [
        "LEGAL_REPRESENTATIVE"
    ],
    "userGroups": [],
    "createdByIdentityId": "dab86db0-98c0-4062-bb70-39bdb3bd9cab",
    "createdOnBehalfOf": false,
    "target": {
        "id": "2bf85948-26c1-40f4-a0fc-f0d22a7e8a90",
        "version": 0,
        "createdDateTime": "2021-08-10T19:27:17.291Z",
        "lastUpdated": "2021-08-10T19:27:17.291Z",
        "requestParticipant": {
            "id": "a6a3291d-ff6d-4b08-8929-61adc0b05c91",
            "version": 0,
            "createdDateTime": "2021-08-10T19:27:17.291Z",
            "lastUpdated": "2021-08-10T19:27:17.291Z",
            "externalId": "0610742121",
            "externalIdType": "CPR",
            "citizenIdentityId": "dab86db0-98c0-4062-bb70-39bdb3bd9cab"
        }
    },
    "documentations": [],
    "processedAutomatically": false,
    "submissionDateTime": "2021-08-10T19:27:17.303Z",
    "submittedByIdentityId": "dab86db0-98c0-4062-bb70-39bdb3bd9cab"
}
```

11.21.2 Response


```
{
    "id": "02a307ca-227f-445b-8176-46ad04eb802e",
    "version": 2,
    "transactionId": "F7MlCBdrICUttJJkdYOhv3strTN6vDO8",
    "createdDateTime": "2021-08-10T19:27:17.290Z",
    "lastUpdated": "2021-08-10T19:41:03.957Z",
    "requestType": "PRIVILEGE_REQUEST",
    "requestState": "APPROVED",
    "requester": {
        "id": "5ec74ef8-d22b-40e4-94ec-ef1262dd3d3c",
        "version": 0,
        "createdDateTime": "2021-08-10T19:27:17.291Z",
        "lastUpdated": "2021-08-10T19:27:17.291Z",
        "externalId": "0610742121",
        "externalIdType": "CPR",
        "citizenIdentityId": "dab86db0-98c0-4062-bb70-39bdb3bd9cab"
    },
    "accessTo": {
        "id": "eeccfad4-e150-4a87-bd77-373eaf5dfc09",
        "version": 0,
        "createdDateTime": "2021-08-10T19:27:17.290Z",
        "lastUpdated": "2021-08-10T19:27:17.290Z",
        "externalId": "0610749832",
        "externalIdType": "CPR",
        "citizenIdentityId": "44ba8c4a-cc01-4f7e-bf4c-8188f9203f28",
        "firstNameProvided": "Lone",
        "lastNameProvided": "Boaikvopg",
        "firstNameResolved": "Lone Boaikvopg"
    },
    "privileges": [
        "LEGAL_REPRESENTATIVE"
    ],
    "userGroups": [],
    "createdByIdentityId": "dab86db0-98c0-4062-bb70-39bdb3bd9cab",
    "createdOnBehalfOf": false,
    "target": {
        "id": "2bf85948-26c1-40f4-a0fc-f0d22a7e8a90",
        "version": 1,
        "createdDateTime": "2021-08-10T19:27:17.291Z",
        "lastUpdated": "2021-08-10T19:41:03.958Z",
        "requestParticipant": {
            "id": "a6a3291d-ff6d-4b08-8929-61adc0b05c91",
            "version": 0,
            "createdDateTime": "2021-08-10T19:27:17.291Z",
            "lastUpdated": "2021-08-10T19:27:17.291Z",
            "externalId": "0610742121",
            "externalIdType": "CPR",
            "citizenIdentityId": "dab86db0-98c0-4062-bb70-39bdb3bd9cab"
        },
        "targetState": "GRANTED"
```

```
    },
    "documentations": [],
    "processedAutomatically": false,
    "submissionDateTime": "2021-08-10T19:27:17.303Z",
    "submittedByIdentityId": "dab86db0-98c0-4062-bb70-39bdb3bd9cab",
    "approvalDateTime": "2021-08-10T19:40:54.672Z",
    "approvedByIdentityId": "44ba8c4a-cc01-4f7e-bf4c-8188f9203f28"
}
```

12 Sender-/Receiver Systems

12.1 Rate-limiting

To ensure that Digital Post is operational the API is protected by a rate limiter. The rate limiter ensures that a caller can only perform a set number of requests within a certain period. And if the caller exceeds this limit the call will be rejected by the API and instead get a `HTTP 429 - Too Many Requests`. A set of headers are exposed to the caller which they can use to ensure they are not rate limited. The headers exposed are the following

- `X-RateLimit-Remaining` The number of remaining tokens in the bucket that the caller has available. When this reaches 0 the caller is rejected.
- `X-RateLimit-Requested-Tokens` The number of tokens that the request removed from the bucket when performing the request. This is used to differentiate between 'cheap' and 'expensive' requests. How a request is determined to be either cheap or expensive is internally determined and can therefore change without notice as the system is optimized.
- `X-RateLimit-Burst-Capacity` The burst capacity is the number of tokens which the bucket initially contained.
- `X-RateLimit-Replenish-Rate` The rate of how fast tokens are re-added to the bucket each second.

The way Digital Post determines which bucket to assign callers is internally handled by the Digital Post and may therefore change in the future. However, callers can expect that the implementation follows the identity of the caller. Meaning that if a sender system is calling, Digital Post assigns a bucket to that system. In the case of standard systems, the limiting is based on who the standard system is acting on behalf of. Meaning that if a standard system is serving two different authorities, these will be rate-limited independently.

If Digital Post is unable to determine the identity of the caller the rate limit will fall back to the IP of the caller.

The rate limit configuration is currently as described here

Mutual TLS systems

	TEST	**PROD**
Bucket size	6	60
Replenish rate	3	30

View clients

	TEST	**PROD**
Bucket size	20	20
Replenish rate	10	10

Please note that these limits may change in the future.

12.2 Patterns for integration to Digital Post

This section contains high level architecture of how Sender and Recipient Systems should be structured using the MeMo-lib. Inherently this also applies the reference implementations which shows a practical example of how the architecture is applied. To reduce the complexity the following section only address operations related to handling MeMo messages.

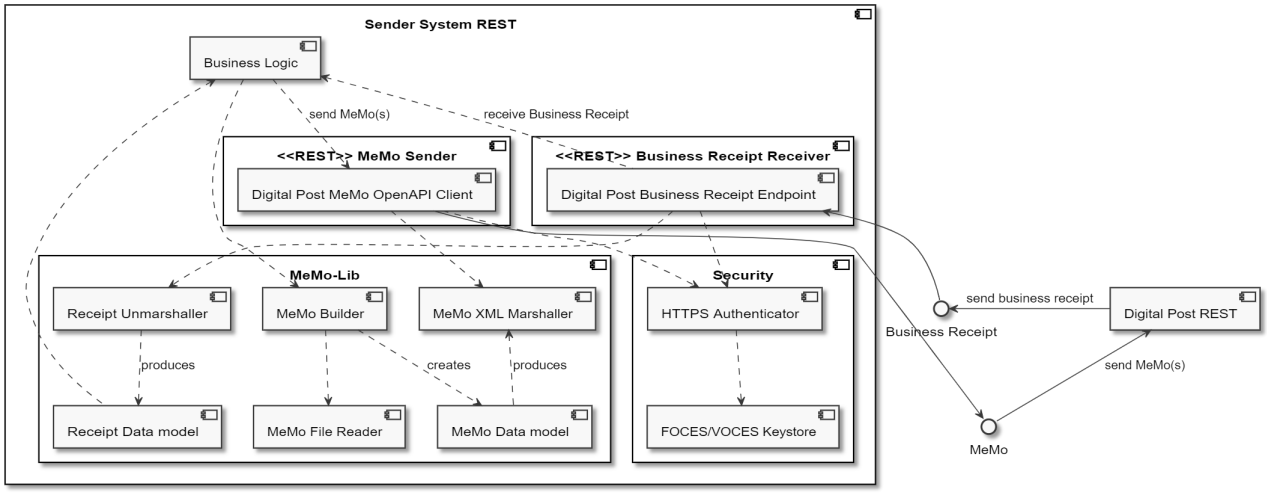
Since Digital Post does not constrain the business logic of when a Sender systems trigger messages and how recipient systems handles received messages all the diagrams contain a “Business Logic” component, which is a representation of where the business related logics could be placed.

12.3 Sender system

The following section gives an overview of how sender systems should be structured.

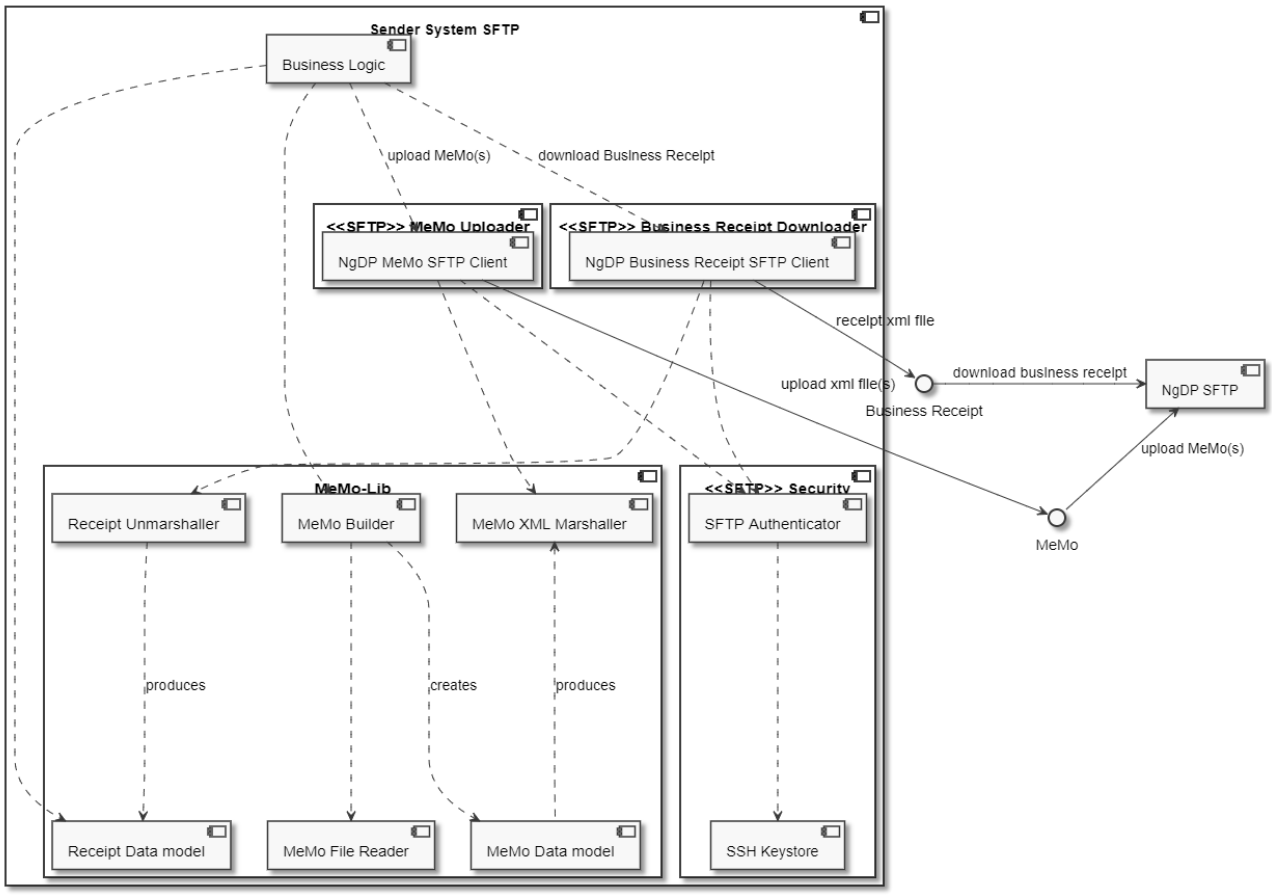
12.3.1 REST Protocol

A Sender System using the REST service protocol is expected to communicate with Digital Post via the HTTP based interface and authenticating using mutual SSL. The REST protocol is ideal to send messages in synchronously to Digital Post and having them distributed immediately.



12.3.2 SFTP Protocol

The SFTP service protocol is ideal when sending large quantities of messages. Which can be delivered asynchronously.

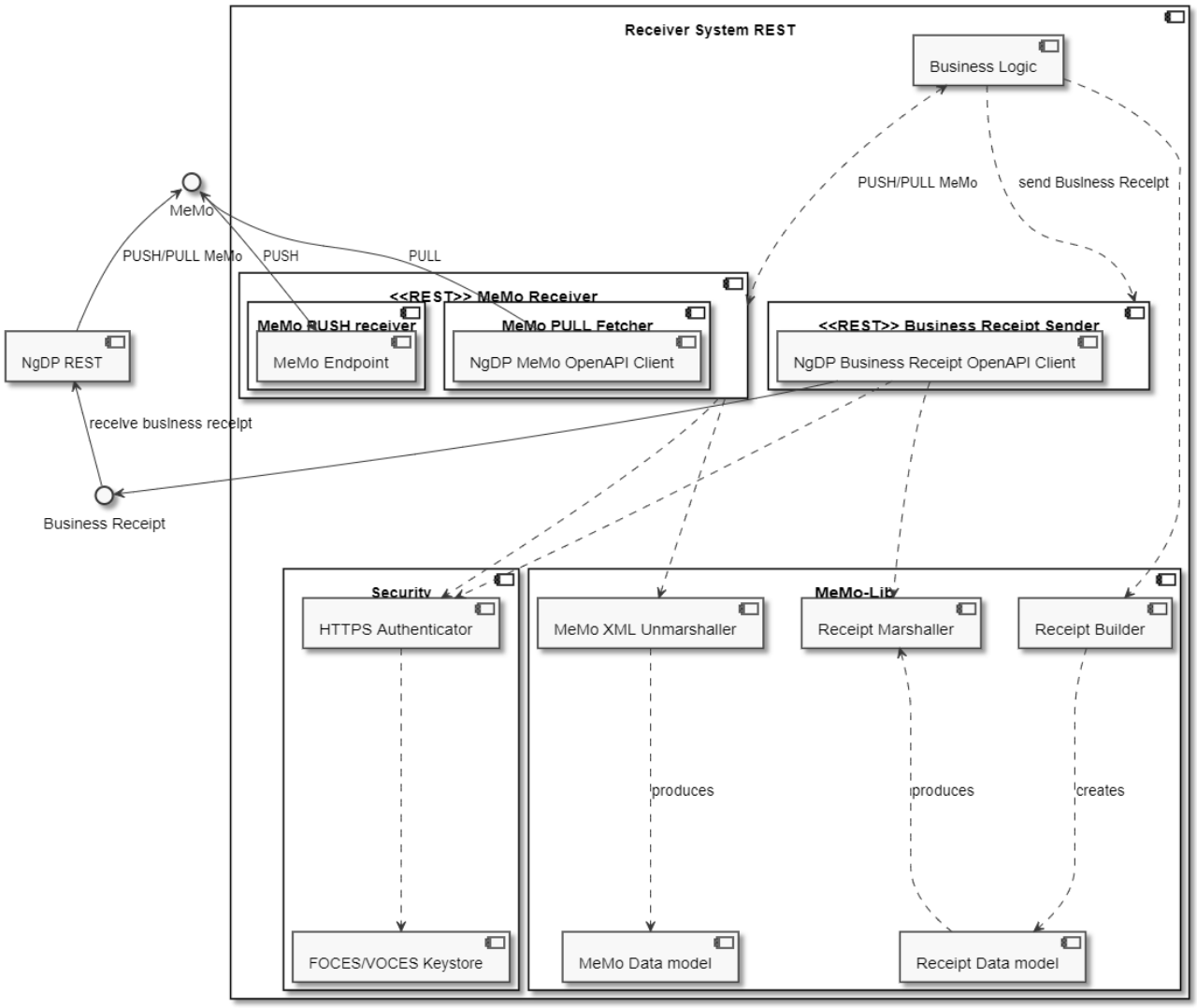


12.4 Recipient system

The following section gives an overview of how recipient systems should be structured.

12.4.1 REST Protocol

The HTTP based REST service protocol can be used both to receive messages as they arrive using the PUSH variant where Digital Post will “push” messages as they are distributed directly to the recipient system. Or as the recipient systems pleases using the pull variant.



12.4.2 Sending Memo

A sender system can deliver MeMos to Digital Post using two different protocols: HTTP, and SFTP.

The term REST is often used instead of HTTP since DP is a rest-full API built on HTTP.

Sending a single MeMo is done just using the XML of the MeMo, but multiple MeMos can be delivered in the same request using tar.lzma packaging and compression.

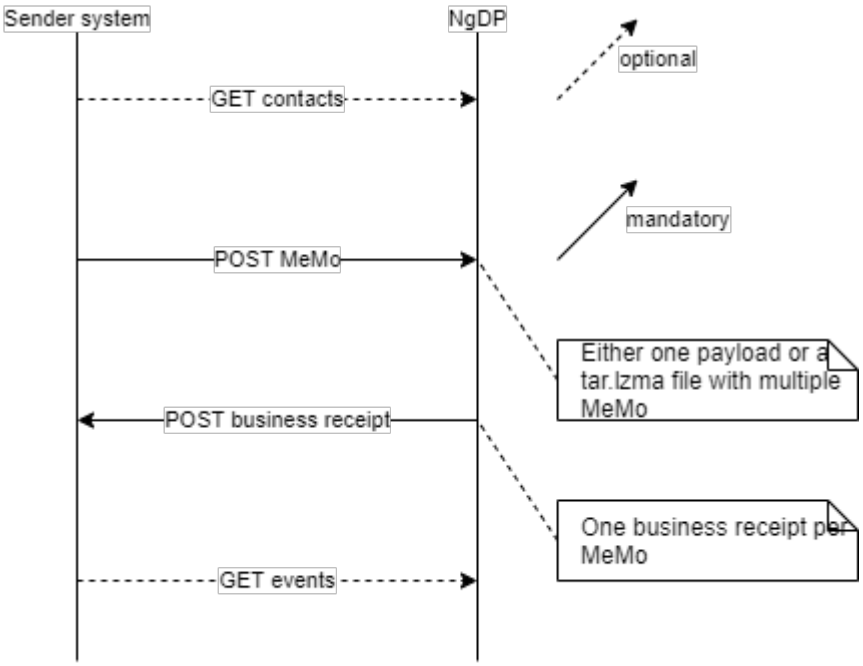
messageUUID - Unique identification of the message UUID is mandatory and must follow the specification for UUID version 4. It is the responsibility of the sender system to ensure that UUID is correctly generated.

Sender system sending one or more MeMo over HTTP (REST)

1. Optional: Sender system lookup if one or more recipients are exempted from Digital Post and/or subscribes to NemSMS
2. Sender system sends either
    a. one MeMo over REST, which must be a <messageUUID> or <messageUUID>.xml

b. Sending one or more MeMos in a tar.lzma file (name of MeMo file inside tar must be <messageUUID>.xml or <messageUUID>)
DP responds with technical receipt in the form of HTTP status code

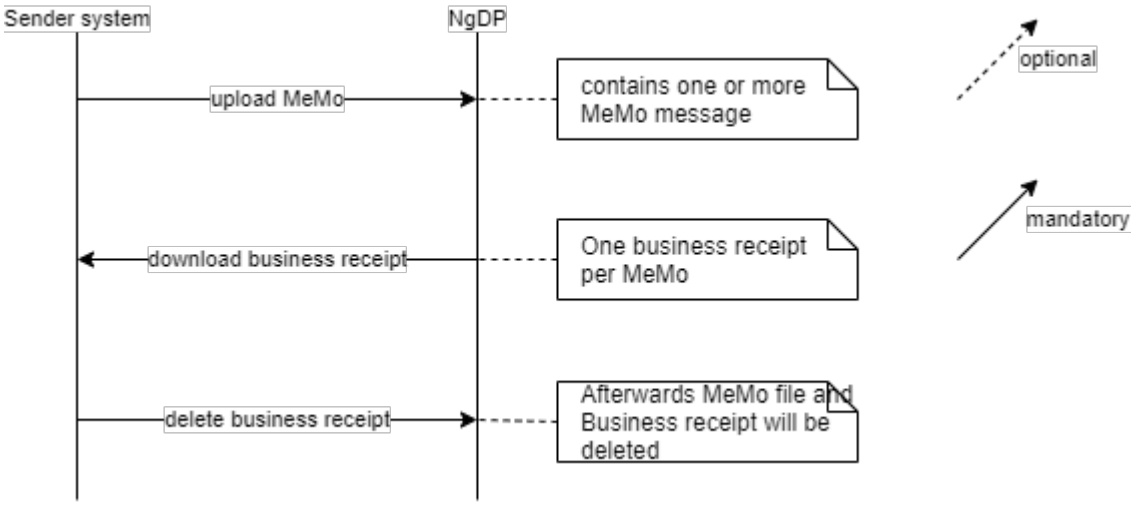
3. DP sends one business receipt per MeMo. Sender system responds with HTTP status code
4. Optional: Sender system can lookup events for the sent MeMo, e.g. to see if messages are waiting for delivery date (valørdato)



Sender system sending file with multiple MeMos over SFTP

DP SFTP support standard SFTP functionality, e.g. compression and transferring/modifying multiple files at-once, either with wildcards or list of files to be addressed.

1. Sender system uploads tar.lzma file with one or more MeMo files (name of MeMo files inside tar must be <messageUUID>.xml or <messageUUID>)
    a. uploading file to temp subfolder
    b. move the file to main folder
2. DP uploads one Business Receipt per MeMo
    a. uploading file to temp subfolder
    b. move the file to main folder
3. Sender system downloads the Business receipt files
4. Sender system deletes the Business receipt files



12.4.3 Receiving Memo

A recipient system can receive MeMos from Digital Post using HTTP.

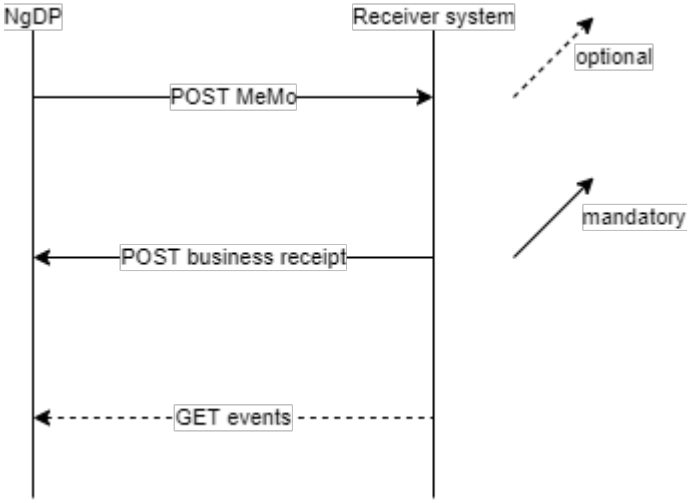
The term REST is often used instead of HTTP since Digital Post is a rest-full API built on HTTP.

Using REST/HTTP is done in one of three ways: REST PUSH, REST PUBLISH SUBSCRIBE, or REST PULL, which is configured when attaching the recipient system.

> If the MeMo is received in a version that is incompatible with the recipient's system, it will be converted to the specified version for the recipient system.

Delivering one MeMo over REST PUSH

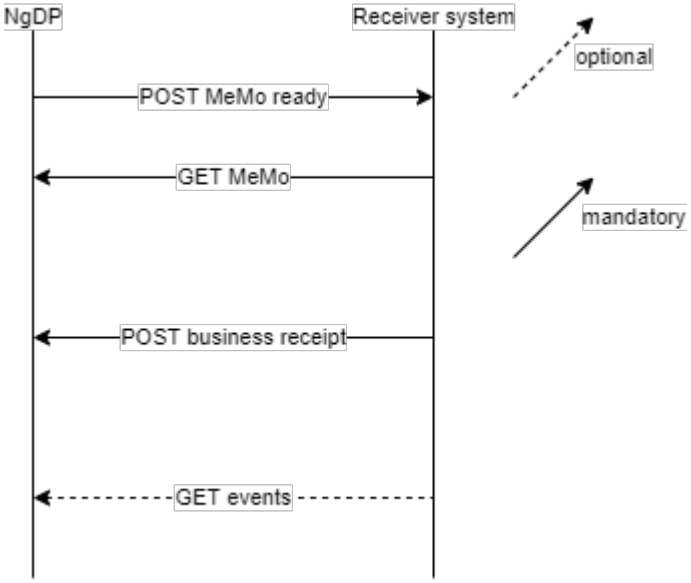
1. Digital Post sends a single MeMo xml, receiver system responds with HTTP status code
2. Recipient system sends a Business Receipt, Digital Post responds with HTTP status code 200



Delivering one MeMo over REST PUBLISH SUBSCRIBE

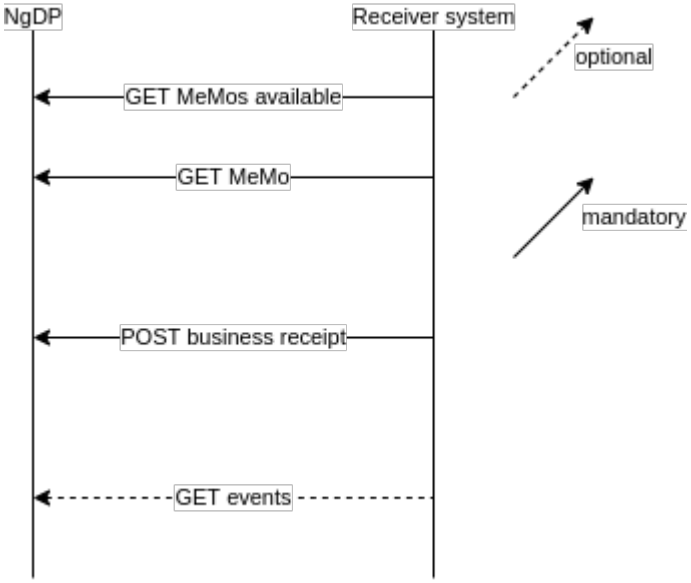
Note: Default recipient systems must use this pattern

1. Digital Post publish new MeMo is ready (one publish call per MeMo), including MeMo id, Receiver system responds with HTTP status code
2. Recipient system fetches MeMo, Digital Post responds with HTTP status code 200 and the MeMo
3. Recipient system sends Business receipt, Digital Post responds with HTTP status code 200



Delivering one MeMo over REST PULL

1. Recipient system fetches list of available MeMos, Digital Post responds with HTTP status code 200 and a list of the MeMos
2. Recipient system fetches MeMo, Digital Post responds with HTTP status code 200 and the MeMo
3. Recipient system sends Business receipt, Digital Post responds with HTTP status code



13 Encoding formats, Environments and Error codes

13.1 Encoding format whitelist for files of documents

The text in purple color is for an upcoming extension in the feature “Udvide liste over tilladte filtyper i Digital Post”.

List of allowed values for the encodingFormat field of the File ressource.

13.1.1 Main document

The following encoding formats and subsequent file extensions are allowed for the main document.

encoding format	file extension
application/pdf	pdf
text/html	html, htm
text/plain	txt

13.1.2 Additional documents

The following encoding formats and subsequent file extensions are allowed for additional documents.

encoding format	file extension
image/heic	heic*, heif*
image/bmp	bmp
text/csv	csv
application/vnd.fujixerox.ddd	ddd
application/msword	doc
application/vnd.openxmlformats-officedocument.wordprocessingml.document	docx
application/x-stata-dta	dta
image/gif	gif

encoding format	file extension
text/html	html, htm
text/calendar	ics, ical
image/jpeg	jpg, jpeg, jfif*
video/quicktime	mov
audio/mpeg	mp3
video/mp4	mp4
application/vnd.oasis.opendocument.spreadsheet	ods
application/vnd.oasis.opendocument.text	odt
application/pdf	pdf
image/png	png
application/rtf	ftf
application/x-spss-sav	sav
image/tiff	tif
text/plain	txt
audio/wav	wav
application/vnd.ms-excel	xls
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	xlsx
application/xml	xml
text/xml	xml

* = only available if `MEMO_ADDITIONAL_FILETYPES_FEATURE_TOGGLE_DATO` is set to true

13.1.3 Technical documents

The following encoding formats and subsequent file extensions are allowed for technical documents.

encoding format	file extension
application/xml	xml
text/xml	xml
application/json	json

13.2 Access to environments

All REST services for an Digital Post environment are exposed from or called from a single IP.

13.2.1 Making requests

When requesting, make sure you include “apis”-path as well as version.

Example: https://api.test.digitalpost.dk/apis/v1/contacts/

13.2.2 TEST environment

Protocol	Inbound/outbound, from Digital Post perspective	Digital Post IP
REST	Inbound	80.198.95.44
SFTP	Inbound	80.198.95.42
REST	Outbound	80.198.95.62

Component	URL	Port	Protocol
Mailserver	http://smtp.test.digitalpost.dk	25	SMTP
SFTP-server	http://sftp.test.digitalpost.dk	22	SFTP
Distribution	https://api.test.digitalpost.dk	443	HTTPS/REST
Kontaktregister	https://api.test.digitalpost.dk	443	HTTPS/REST

Component	URL	Port	Protocol
Systemregister	https://api.test.digitalpost.dk	443	HTTPS/REST
Opbevaring	https://api.test.digitalpost.dk	443	HTTPS/REST
Hændelseslog	https://api.test.digitalpost.dk	443	HTTPS/REST

13.2.3 PROD Environment

Protocol	Inbound/Outbound from Digital Post perspective	Digital Post IP
REST	Inbound	80.198.95.23
SFTP	Inbound	80.198.95.22
REST	Outbound	80.198.95.62

Component	URL	Port	Protocol
Mailserver	smtp.digitalpost.dk	25	SMTP
SFTP-server	sftp.digitalpost.dk	22	SFTP
Distribution	api.digitalpost.dk	443	HTTPS/REST
Kontaktregister	api.digitalpost.dk	443	HTTPS/REST
Systemregister	api.digitalpost.dk	443	HTTPS/REST
Opbevaring	api.digitalpost.dk	443	HTTPS/REST
Hændelseslog	api.digitalpost.dk	443	HTTPS/REST

E-mails from Digital Post

- Unrepliable e-mails (notifications, receipts and rights notifications) are sent from noreply@digitalpost.dk

13.3 SMS character encoding

Digital Post are currently using GSM-7 character encoding standard for SMS. This is true for all SMS that goes through Gateway API (external SMS integration). This includes the following:

- For verification SMS from Contact registry
- For SMS notifications
- For NemSMS messages

13.3.1 Limitations of GSM-7

7-bit Characters: GSM-7 uses 7 bits to represent each character, which is less than the 8 bits used in many other encoding systems. This is designed to maximize the use of the limited bandwidth available for SMS messages.

Basic Character Set: The basic GSM-7 character set includes 128 characters, which cover most (but not all) of the characters used in Western languages. This can result in missing support for some special and non-Latin characters sent as part of an SMS.

SMS Length Limitation: A standard SMS message can contain up to 160 characters when encoded in GSM-7.

Read more about limitations here: https://gatewayapi.com/docs/limitations/#limitations

13.4 Error codes

This section describes all the errors codes returned by DP.

- Front-end validation and error codes in the view client
- Back-end validation and error codes in distribution
- Recipient system error codes

13.4.1 Front-end validation and error codes in the Viewclient

This section describes the front-end errorcodes from DP, i.e. the error codes that may be returned synchronously from the backend via REST CRUD calls. This section presents error codes in a block format, detailing the issues encountered during front-end validation, and the corresponding errors returned by the system.

Access services

```
# Access
access.accessType.notNull=Typen skal udfyldes.
access.mailboxId.notNull=Postkassens id skal udfyldes.
access.maximum.number.of.emails.exceeded=Der kan maksimalt oprettes {0} antal e-mails
til notifikation for denne type adgang.
access.maximum.number.of.sms.exceeded=Der kan maksimalt oprettes {0} antal sms til
notifikation for denne type adgang.
access.emailNotificationSubscriptions.email.required=E-mailadressen skal udfyldes.
access.emailNotificationSubscriptions.email.invalid=E-mailadressen er ikke gyldig.
access.emailNotificationSubscriptions.email.alreadyUsed=E-mailadressen {0} er
allerede angivet én gang.
access.emailNotificationSubscriptions.confirmationTime.verificationRequired=Verifikat
ion skal gennemføres før e-mailadressen kan bekræftes.
```

```
access.smsNotificationSubscription.mobileNumber.required=Mobiltelefonnummeret skal
udfyldes.
access.smsNotificationSubscription.mobileNumber.invalid=Mobiltelefonnummeret {0} er
ikke et gyldigt dansk nummer.
access.smsNotificationSubscription.confirmationTime.verificationRequired=Verifikation
skal gennemføres før mobiltelefonnummeret kan bekræftes.
access.pushNotificationSubscriptions.deviceId.required=Device id skal udfyldes.
access.pushNotificationSubscriptions.deviceId.alreadyUsed=Device id {0} er allerede
angivet én gang.
access.optedOutOfNotificationsDateTime.isInAFuture=datoen må ikke være i fremtiden.
```

Access request services

```
accessRequest.requestType.required=Type (requestType) skal udfyldes.
accessRequest.requestType.invalid=Ugyldig værdi angivet.
accessRequest.privilegeEndDate.invalid=En anmodning må ikke oprettes eller godkendes
med udløbet slutdato.
accessRequest.privilegeEndDateTime.invalid=En anmodning må ikke oprettes eller
godkendes med udløbet slutdato.
accessRequest.privilegeEndDateTime.invalid.after=Anmodningens sluttidspunkt må ikke
være efter {0}.
accessRequest.privilegeEndDate.withPrivilegeEndDateTime=En anmodning kan ikke
indeholde både privilegeEndDate og privilegeEndDateTime.
accessRequest.requestType.modified=Det er ikke tilladt at skifte type.
accessRequest.requestState.required=Tilstand (requestState) skal udfyldes.
accessRequest.requestState.invalid=Ugyldig værdi angivet.
accessRequest.create.requestState.invalid=En anmodning kan kun oprettes med
requestState DRAFT.
accessRequest.requestState.invalidTransition=En anmodning kan ikke skifte fra {0} til
{1}.
accessRequest.requester.required=Anmoderen (requester) skal udfyldes.
accessRequest.accessTo.required=Rettighedsejer (accessTo) skal udfyldes.
accessRequest.target.required=Rettighedsmodtager (target) skal udfyldes.
accessRequest.documentations.mediaType.invalid=Ugyldig type {0} - tilladte typer {1}.
accessRequest.documentations.tooLarge=Ugyldig filstørrelse - maks. størrelse {0}
bytes.
accessRequest.documentations.size=Antallet af vedhæftninger må ikke overstiger {0}.
accessRequest.documentations.documentationType.required=Dokumenttype skal udfyldes.
accessRequest.documentations.documentationType.specific.required=En dokument af typen
{0} skal vedhæftes.
accessRequest.documentations.content.infected=Virus detekteret. Filen kan ikke
tilføjes.
accessRequest.requestParticipant.contact.not.allowed=Det er ikke muligt at lave et
forespørgsel til denne kontakt
accessRequest.requestParticipant.identity.notFound=Deltageren {0} kunne ikke
identificeres.
accessRequest.requestParticipant.identity.identityType=Typen på deltageren {0} kunne
ikke identificeres.
accessRequest.requestParticipant.identity.required=Deltagerens identitet kunne ikke
afgøres. Enten skal externalId og type udfyldes eller også identitets id udfyldes.
```

```
accessRequest.requestParticipant.identity.mismatch=Deltagerens angivne interne
identifikator matcher ikke med den eksterne.
accessRequest.requestParticipant.identityType.mismatch=Deltagerens identitet er en
blanding af borger og virksomhed/medarbejder og dermed ugyldig.
accessRequest.requestParticipant.externalId.pattern=externalId skal indeholde gyldigt
CPR-nummer (10 tegn) eller CVR-nummer (8 tegn).
accessRequest.requestParticipant.externalId.cprLookup.timeout.small=Der er foretaget
for mange fejlende CPR opslag. Yderligere forsøg er blokeret i {0} minutter.
accessRequest.requestParticipant.externalId.cprLookup.timeout.large=Der er foretaget
for mange fejlende CPR opslag. Yderligere forsøg er blokeret i {0} minutter.
accessRequest.requestParticipant.externalIdType.required=Det skal angives om den
eksterne nøgle er CPR-nummer eller CVR-nummer.
accessRequest.requestParticipant.externalIdType.invalid=Det skal angives om den
eksterne nøgle er CPR-nummer eller CVR-nummer.
accessRequest.requestParticipant.firstNameProvided.required=Fornavn skal udfyldes.
accessRequest.requestParticipant.lastNameProvided.required=Efternavn skal udfyldes.
accessRequest.requestParticipant.firstNameProvided.size=Længden af fornavnet må ikke
overstige 256 tegn.
accessRequest.requestParticipant.lastNameProvided.size=Længden af efternavnet må ikke
overstige 256 tegn.
accessRequest.requestParticipant.emailAddress.required=E-mailadresse skal udfyldes
accessRequest.requestParticipant.emailAddress.invalid=Ugyldig e-mailadresse.
accessRequest.requestParticipant.emailAddress.size=Længden af e-mailadressen må ikke
overstige 256 tegn.
accessRequest.requestParticipant.alias.size=Længden af alias må ikke overstige 256
 tegn.
accessRequest.requestParticipant.position.size=Længden af stillingsbetegnelse må ikke
overstige 256 tegn.
accessRequest.requestParticipant.mustBeCitizen=Typen af deltager være en borger i
denne rolle
accessRequest.requestParticipant.mustBeOrganisation=Typen af deltager være en
organisation i denne rolle
accessRequest.target.requestParticipant.required=requestParticipant skal udfyldes på
target.
accessRequest.target.requestParticipant.mustMatchRequester=Target skal matche
requester.
accessRequest.accessTo.full.access.must.be.citizen=AccessTo skal være en borger for
 anmodninger der indeholder læse- og skriveadgang privilegiet.
accessRequest.accessTo.full.access.max.accesses=Adgangsgiver kan højst have 10 læse
og læse- og skriveadgange
accessRequest.accessTo.mustMatchRequester=accessTo skal matche requester.
accessRequest.privileges.npte.required=Ved
USER_ADMIN_STATEMENT_OF_TRUTH_PRIVILEGE_REQUEST skal privilegierne være
foruddefinerede NPTE privilegier: {0}
accessRequest.accessTo.mustMatchAppointments=Ved APPOINTED_DELEGATION_REQUEST skal
accessTo matche allerede tildelte privilegiers scope.
accessRequest.illegalModification=Ugyldig redigering af anmodning foretaget.
accessRequest.rejectionReason.required=Begrundelse (rejectionReason) skal udfyldes.
accessRequest.revocationReason.required=Begrundelse (revocationReason) skal udfyldes.
accessRequest.userGroups.notAllowed=Brugergrupper kan kun benyttes på direkte
tildelinger indenfor samme organisdation.
accessRequest.privilegesOrUserGroups.required=Privilegier og/eller brugergrupper skal
udfyldes.
```

```
accessRequest.privileges.userAdmin.required=Privilegiet skal være
ORGANISATION_USER_ADMINISTRATOR (og kun det).
accessRequest.privileges.special.required=Ved specialanmodninger (CURATOR,
EXECUTOR_OF_ESTATE, LIQUIDATOR) skal privilegiet være én og kun én af de 3.
accessRequest.privileges.special.notAllowed=Special-privilegier (CURATOR,
EXECUTOR_OF_ESTATE, LIQUIDATOR) er kun tilladt på specialanmodninger
(SPECIAL_PRIVILEGIE_REQUEST) eller videre delegeringer
(APPOINTED_DELEGATION_REQUEST).
accessRequest.privileges.required=Privilegier skal udfyldes.
accessRequest.privileges.legalOwnerOfInactiveOrClosedCompany.required=Ved
LEGAL_OWNER_OF_INACTIVE_OR_CLOSED_COMPANY_PRIVILEGE_REQUEST skal privilegierne være
alle foruddefinerede privilegier: {0}
accessRequest.accessTo.legalOwnerOfInactiveOrClosedCompany.hasCurator=Organisationen
har kurator/bobestyrer tilknyttet.
accessRequest.privileges.delegatedSupportAdminPrivilegeRequest.required=Ved
DELEGATED_SUPPORT_ADMIN_PRIVILEGE_REQUEST skal privilegierne være alle
foruddefinerede privilegier: {0}
accessRequest.target.activation.code.required=Aktiveringskode skal udfyldes.
accessRequest.withdrawal.illegal=Anmodningen kan ikke trækkes tilbage når
privilegierne/grupperne er tildelt.
accessRequest.termsApproval.notAuthorityAndNotCorrectIndustryCode=Organisationen skal
være en myndighed eller have branche code 841100.
citizenLookup.identity.notFound=Deltageren {0} kunne ikke identificeres.
citizenLookup.firstNameProvided.required=Fornavn skal udfyldes.
citizenLookup.lastNameProvided.required=Efternavn skal udfyldes.
organisationLookup.organisationId.notFound =Organisationen {0} kunne ikke
identificeres.
citizenLookup.cprLookup.timeout.small=Der er foretaget for mange fejlende CPR opslag.
Yderligere forsøg er blokeret i {0} minutter.
citizenLookup.cprLookup.timeout.large=Der er foretaget for mange fejlende CPR opslag.
Yderligere forsøg er blokeret i {0} minutter.
```

Folder Services

```
# Folder
folder.folderType.illegal=Standardmapper kan ikke oprettes.
folder.folderType.notNull=Mappens type skal udfyldes.
folder.folderType.invalid={0} er ugyldig som mappetype.
folder.name.notBlank=Navn på mappen skal udfyldes.
folder.create.too.many.folders=Der kan maksimalt oprettes {0} antal mapper.
folder.create.standard.folder.exists=En standardmappe af typen {0} eksisterer
allerede.
folder.create.standard.folder.with.parent.not.allowed=En standardmappe må ikke være
undermappe.
folder.createOrUpdate.name.exists.at.level=Der findes allerede en mappe der hedder
{0} på dette niveau.
folder.createOrUpdate.too.many.levels=Antallet af mappe-niveauer må ikke over stige
{0}.
folder.createOrUpdate.non.existing.parent=Overmappen {0} eksistererer ikke i
postkassen.
```

```
folder.createOrUpdate.standard.parent.not.allowed=En standardmappe kan ikke have
undermapper
folder.createOrUpdate.name.size=Antal tegn i feltet name må ikke overstige {0} tegn.
folder.createOrUpdate.name.illegal=Mappens navn må ikke indeholde en af følgende
tegn: {0}
folder.update.standard.folder.may.no.be.updated=En standardmappe må ikke opdateres.
folder.delete.messages.exists=Mappen kan ikke slettes, da den indeholder meddelelser.
folder.delete.subfolders.exists=Mappen kan ikke slettes, da den indeholder
undermapper.
folder.delete.standard.folders.may.not.be.deleted=Mappen kan ikke slettes, da det er
en standardmappe.
```

Message services

```
# Message
message.draft.sendercontactPoint.not.allowed=Afsenders kontakpunkt {0} må ikke have
nogen værdi
message.create.reply.notAllowed=Denne meddelelse kan ikke besvares.
message.update.reply.sender.notNull=Feltet sender må ikke slettes for besvarelser.
message.update.reply.recipient.notNull=Feltet recipient må ikke slettes for
 besvarelser.
message.update.reply.sender.id.not.allowed=Feltet senderId må ikke opdateres for
 besvarelser.
message.update.reply.sender.id.type.not.allowed=Feltet senderIdType må ikke opdateres
for besvarelser.
message.update.reply.recipient.not.allowed=Feltet recipient må ikke opdateres for
 besvarelser.
message.update.reply.label.not.allowed=Feltet label må ikke opdateres for
 besvarelser.
message.update.folderId.notNull=Mappe skal angives.
message.update.folderId.notFound=Den angivne mappe eksisterer ikke i postkassen.
message.update.legallyNotified.not.allowed=Kun forkyndelser kan markeres forkyndt.
message.update.folderType.not.allowed=En meddelelse i tilstand {0} kan ikke flyttes
til mappe af typen {1}
message.create.draft.folderType.not.allowed=Meddelelse skal placeres i DRAFTS-mappen;
ikke i {0}.
message.create.draft.messageType.not.allowed=Meddelelsestypen {0} er ikke tilladt
her. Det skal være {1}.
message.create.draft.folderId.notFound=Den angivne mappe eksisterer ikke i
postkassen.
message.create.forward.email.invalid=Den angivne email {0} er ikke valid.
message.create.forward.notAllowed=Denne meddelelse kan ikke videresendes.
message.create.forward.comment.size=Antal tegn i feltet {0} må ikke overstige {1}
tegn.
message.create.forward.comment.full-access.allowed.size=Der er for mange tegn i
kommentaren når standardteksten påsættes i starten af kommentaren.
message.create.forward.recipientId.notBlank=Modtagers id skal angives
message.create.forward.recipientId.email.invalid=E-mailadressen {0} er ugyldig.
message.create.forward.recipientId.cpr.invalid=CPR-nummeret {0} er ugyldigt
message.create.forward.recipientId.cvr.invalid=CVR-nummeret {0} er ugyldigt
message.create.forward.recipientIdType.notNull=Modtagertypen skal angives
```

```
message.create.forward.recipientIdType.invalid=Ugyldig modtagertype {0}. CPR/CVR/
EMAIL er gyldige værdier.
message.create.forward.senderLabel.size=Antal tegn i feltet {0} må ikke overstige {1}
tegn.
message.create.forward.recipientLabel.size=Antal tegn i feltet {0} må ikke overstige
{1} tegn.
message.create.forward.size.exceeds.allowed.size=Total filstørrelse {0} overskrider
tilladt størrelse til email-videresendelse: {1}.
message.maximum.number.of.additionalContentData.exceeded=Der kan maksimalt oprettes
{0} antal additionalContentData.
message.document.actions.status.of.additionalActionStatusData.exceeded=Der kan
maksimalt oprettes {0} antal additionalActionStatusData.
message.document.draftMessageRequired=Meddelelsen skal være en kladdemeddelelse for
 at kunne tilføje dokument.
message.document.actionStatus.empty = "StatusCode"-feltet inden for "ActionStatus"
 kan ikke være tomt.
message.document.number.higher.than.allowed=Grænsen for antal dokumenter der kan
tilføjes beskeden er oversteget: {0}
message.file.encodingFormat.required=Format skal udfyldes - f.eks. 'text/html'.
message.file.encodingFormat.size=Feltet {0} må ikke overstige 256 tegn.
message.file.filename.required=Filnavn skal udfyldes.
message.file.filename.size=Feltet {0} må ikke overstige 256 tegn.
message.file.content.size=Filen er for stor.
message.file.content.infected=Virus detekteret. Filen kan ikke tilføjes.
message.file.language.required=Sprog skal udfyldes - f.eks. 'da'.
message.file.encoding.format.invalid=Det angivne format {0} er ikke tilladt for
 dokumenter af typen {1}. Tilladte formater: {2}
message.file.encoding.format.invalid.switch.to.html=encodingFormat kan ikke skiftes
til text/html da det eksisterende indhold ikke er validt html.
message.file.name.invalid=Det angivne filtypenavn i {0} er ikke tilladt for filer med
format {1}. Tilladte filtypenavne: {2}
message.file.name.invalid.character=File name contains invalid character: {0}
message.file.draftMessageRequired=Meddelelsen skal være en kladdemeddelelse for at
kunne tilføje fil.
message.file.number.higher.than.allowed=Grænsen for antal filer der kan tilføjes
beskeden er oversteget: {0}
message.full-access.documents.allowed.size=Der kan kun være ét hoveddokument. Der var
{0}
message.full-access.files.allowed.size=Der kan kun være én fil på hoveddokumentet.
Der var {0}
message.full-access.files.wrong.encoding=Filen har en forkert filtype. Det skal være
HTML format
message.language.code.invalid=Sprogkoden {0} matcher ikke ISO 369-1
message.total.file.size.exceeds.allowed.size=Total filstørrelse overskrider tilladt
størrelse: {0}
message.recipient.recipientIdType.invalid=Ugyldig modtagertype {0}. CPR/CVR er
gyldige værdier.
message.recipient.recipientId.cpr.invalid=CPR-nummeret {0} er ugyldigt.
message.recipient.recipientId.cvr.invalid=CVR-nummeret {0} er ugyldigt.
message.recipient.recipientId.email.invalid=E-mailen {0} er ugyldig.
message.recipient.contactPoint.contactInfo.exceeds.max=Der kan højst angives 2
 kontaktinformationer.
```

```
message.recipient.contactPoint.contactInfo.label.notBlank=Informationsfeltets navn
skal angives.
message.recipient.contactPoint.contactInfo.value.notBlank=Informationsfeltets værdi
skal angives.
message.send.label.required=Feltet label skal udfyldes ved afsendelse.
message.send.sender.label.required=Feltet sender.label skal udfyldes ved afsendelse.
message.send.invalid.folder=En meddelelse kan kun sendes fra 'Kladder'-mappen.
message.send.invalid.state=Kun kladder kan sendes. Denne meddelelse er en i
tilstanden {0}.
message.send.documents.required=Der skal tilføjes et hoveddokument.
message.send.documents.main.exceeds.max=Der kan kun tilføjes et hoveddokument.
message.send.documents.files.required=Der skal tilføjes minimum en fil til et
dokument.
message.send.documents.files.content.required=Der skal tilføjes indhold til filen
{0}.
```

System Fetch services

```
# SystemFetch
systemFetch.organisationId.notNull=Organisationens id skal udfyldes
systemFetch.contactPointId.notNull=Kontaktpunkt id skal udfyldes
systemFetch.systemFetchStatusType.invalid=Status typen må kun sættes til STOPPED
systemFetch.currently-running=Systemafhentning er i gang for postkassen
systemFetch.organisationId.notFound=Organisationen kan ikke findes.
systemFetch.organisationId.invalid=Organisation matcher ikke postkassen. Postkassse
CVR:{0}. Organisation CVR: {1}".
systemFetch.organisationId.changed=Organisationen kan ikke skiftes.
systemFetch.systemId.notFound=Systemet kan ikke findes.
systemFetch.systemId.inactive=Systemet er ikke aktivt.
systemFetch.systemId.invalid=Systemet er ikke et modtagersystem.
systemFetch.systemId.not.default=Systemet er ikke angivet som v�rende det prim�re
modtagersystem.
systemFetch.contactPointId.notFound=Kontaktpunktet kan ikke findes.
systemFetch.contactPointId.inactive=Kontaktpunktet er ikke aktivt.
systemFetch.contactPointId.changed=Kontaktpunktet kan ikke skiftes på kørende
Systemafhentning. Stop og start ny.
systemFetch.systemFetchStatusType.cannotStop=En FINISHED systemafhentning kan ikke
stoppes.
```

Asynchronous distribution messages

If a message cannot be distributed, an error message will be generated in senders mailbox. It can contain the following errors:

```
# From distribution-validator
memo.infected=Virus detekteret i et bilag
recipient.not.found=Ukendt modtager
memo.invalid=Intern fejl
recipient.is.closed=Modtageren kan ikke modtage digital post
```

```
recipient.is.exempt=Modtageren er fritaget
recipient.contact.point.not.allowed.for.id.type=Kontaktpunkt kan ikke angives for
 denne modtager
recipient.mailbox.and.default.recipient.system.not.found=Modtageren kan ikke findex.
sender.not.found=Afsenderen (dig) kan ikke findes i systemet.
```

File and document services

Initial validation of file attachments occur while creating and editing a draft through the view client. Files are HTML validated against a whitelist. This entails the following error codes in case of rejection:

```
# Html validator
html.validator.rejected=Filen {0} kunne ikke genkendes som et gyldigt html-dokument
html.validator.rejected.comments=Filen {0} indeholder kommentarer. Kommentarer er
ikke tilladt.
html.validator.rejected.element=Filen {0} indeholder element \"{1}\", som enten ikke
tilladt eller som indeholder data, der ikke er tilladt.
html.validator.rejected.element.attributes=Filen {0} indeholder element \"{1}\" med
attribut \"{2}\", der enten ikke er tilladt attribut, eller som indeholder data, der
ikke er tilladt.
html.validator.rejected.unknown-element=Filen {0} indeholder url i en ikke godkendt
placering. Det er sandsynligvis i en style attribut. Kun data url'er er tilladt.
```

Contact services

```
danish.mobile.number.invalid="{0}" er ugyldigt dansk mobilnummer
danish.mobile.number.needed=Dansk mobilnummer er påkrævet
exemption.end.update.not.allowed=Opdatering af fritagelses slutdato er ikke tilladt
exemption.start.invalid=Ugyldig startdato for fritagelse
exemption.after.voluntary.registration.not.allowed=Det er ikke tilladt at fritage sig
selv efter frivillig tilmelding
invalid.exemption.start.or.end=Fritagelses start- og sluttidspunkt må ikke sættes
id.not.exist=ID findes ikke
registration.status.needed=Registreringsstatus er påkrævet
type.wrong="{0}" er den forkerte type
type.update.not.allowed=Typeopdatering er ikke tilladt
changed.date.update.not.allowed=Opdatering at status.changedDate er ikke tilladt
access.denied=Adgang nægtet
terms.id.missing=Ingen vilkår accepteret
terms.id.invalid=Ugyldige vilkår accepteret
removal.of.confirmedDateTime.not.allowed=Det er ikke tilladt at fjerne en nemSMS
bekræftelse
new.confirmedDateTime.must.be.after.old.confirmedDateTime=En ny bekræftelse: {0} skal
være nyere end den foregående: {1}
confirmedDateTime.can.not.be.in.the.future=Bekræftelse ugyldig:{0} en bekræftelse kan
ikke være i fremtiden
cannot.confirm.nem.sms.for.nonexistent.verification=NemSMS nummeret skal verificeres
før det kan bekræftes
terms.type.missing=Type for vilkår mangler at bliver angivet
```

```
terms.version.missing=Vilkårs version skal angives
voluntary.registration.status.closed=Frivillig registrering ikke tilladt for lukkede
kontakter
voluntary.registration.age=Frivillig registrering ikke tilladt for borgere under 15
voluntary.registration.eligible=Frivillig registrering kun tilladt for borgere som er
berettiget til frivilig registrering
voluntary.registration.active=Frivillig registrering kan ikke udføres for allerede
registreret borger
```

Identity services

```
clientdetails.clientSecretRequired.mandatory=Hemmelighed er påkrævet for klienter!
clientdetails.clientsecret.invalid=Den givne hemmelighed er ugyldig
clientdetails.clientsecret.mangled=Den angivne hemmelighed matcher delvist den
eksisterende hemmelighed! Dette skyldes enten den nye hemmelighed ligner den gamle
for meget, eller input er beskadiget pga. manglende håndtering af specialtegn.

grantee.identity.group.immutable=Gruppe reference er uforanderlig
grantee.identity.group.invalid=Invalid gruppe reference
grantee.identity.group.required=Gruppe reference er påkrævet
grantee.identity.group.default.immutable=DEFAULT-gruppe er uforanderlig
grantee.identity.invalid=Invalid reference til identitet
grantee.identity.required=Reference til identitet er påkrævet
grantee.issuer.invalid=Udstedende identitet er ugyldig
grantee.issuer.required=Udstedende identitet er påkrævet

identifier.empty=Minimum én identifikator er påkrævet
identifier.type.empty=Type er påkrævet
identifier.type.invalid=Type er ugyldig
identifier.type.unambiguous.constraint.violation=Utvetydige type(r): {0} overtræder
maximum antal: 1 for én identitet
identifier.value.empty=Værdien er påkrævet
identifier.value.invalid=Værdien er ugyldig

identity.citizenName.invalid=Person navn er ugyldig
identity.invalid=Invalid identitet
identity.parent.id.invalid=Forældrereference er ugyldig
identity.parent.type.invalid=Forældretype er ugyldig
identity.parent.employee.invalid=Forældre CVR og medarbejder RID kombination er
ugyldig
identity.type.empty=Type er påkrævet
identity.type.invalid=Type er ugyldig
identity.email.invalid=E-mailen er ugyldig
identity.email.empty=Identiteten har ikke registeret en e-mail som kan verificeres

identityGroup.issuer.invalid=Udstedende identitet er ugyldig
identityGroup.issuer.required=Udstedende identitet er påkrævet
identityGroup.name.required=Navn er påkrævet
identityGroup.owner.invalid=Ejer er ugyldig
identityGroup.owner.required=Ejer er påkrævet
identityGroup.type.invalid=Type er ugyldig
```

```
identityPrivilege.delegated.type.update.not.allowed=Delegeret privilegie understøtter
ikke manuel redigering
identityPrivilege.group.immutable=Gruppe reference er uforanderlig
identityPrivilege.group.invalid=Gruppe er ugyldig
identityPrivilege.group.required=Gruppe er påkrævet
grantee.identity.group.parent.invalid=Gruppe skal have fælles parent ejer med parent-
privilegiets gruppe
identityPrivilege.issuer.invalid=Udstedende identitet er ugyldig
identityPrivilege.issuer.required=Udstedende identitet er påkrævet
identityPrivilege.parent.id.invalid=Forældrereference er ugyldig
identityPrivilege.parent.scope.invalid=Forældreafgrænsning afviger fra afgrænsning
identityPrivilege.parent.source.invalid=Forældrekilde afviger fra kilde
identityPrivilege.parent.type.invalid=Forældretype afviger fra type
identityPrivilege.scope.invalid=Identitetsreference er ugyldig for privilegiets
afgrænsning
identityPrivilege.scope.required=Identitetsreference er påkrævet for privilegiets
afgrænsning
identityPrivilege.source.appointed.invalid=Kilde APPOINTED må ikke kombineres med
parentPrivilegeId
identityPrivilege.source.invalid=Kilde er ugyldig
identityPrivilege.source.required=Kilde er påkrævet
identityPrivilege.type.invalid=Type er ugyldig
identityPrivilege.type.required=Type er påkrævet
identityPrivilege.type.invalid.scope.power-of-attorney=Modtager af ægte fuldmagt kan
både være borger og virksomhed
identityPrivilege.type.feature.disabled.full-power-of-attorney=Fuld adgang er ikke
slået til
identityPrivilege.type.invalid.grantee.full-power-of-attorney=Adgangshaver af fuld
adgang kan kun være en borger eller virksomhed
identityPrivilege.type.invalid.type=Adgangsgiver kan højst have 10 læse- eller fulde
adgange
identity.subscription.confirmationTime.removal.not.allowed=Bekræftelsestidspunkt må
ikke fjernes
identity.subscription.confirmationTime.must.be.more.recent=Bekræftelsestidspunkt skal
være nyere end det eksisterende
identity.subscription.confirmationTime.cannot.confirm.unverified.email=Verifikation
skal gennemføres før e-mailadressen kan bekræftes igen

directPrivilege.grantee.scope.invalid=Modtager (grantee) skal være forskellig fra
afgrænsning (scope)
directPrivilege.grantee.invalid=Modtager (grantee) er ugyldig
directPrivilege.grantee.required=Modtager (grantee) er påkrævet
directPrivilege.grantee.parent.invalid=Modtager (grantee) skal have fælles parent med
parent-privilegiets grantee
ulkDirectPrivileges.list.maxBulkSize.exceed=Antallet af direkte privilegier for
 masseopdatering overstiger det tilladte maksimum: {0}

identity.privilege.bulk.query.maxSize.exceeded=Antallet af IDer overstiger det
tilladte maksimum: {0}.
```

Verification services

```
verification.max.attempts=Det maksimale antal forsøg er overskredet.
verification.validity.exceeded=PIN er udløbet.
verification.pin.invalid=Ugyldig PIN
verification.pin.is.present.for.non.verifying.state=Verifikations pinkode må kun være
til stede for verifying tilstand
verification.not.in.verifying.state=Nuværende verifikation er ikke i verificerings-
tilstand
verification.state.invalid=Tilstanden på verifikationen er ugyldig
verification.channel.null.or.blank=Værdien skal være udfyldt
verification.channelType.null.or.unknown=Værdien skal være udfyldt
verification.identityId.null=Værdien skal være udfyldt
verification.already.exists=Verifikation findes allerede
verification.is.not.verified=Verifikation er ikke i verificerings-tilstand
verification.linkToken.can.be.only.present.for.mobile=Link verifikationsflow kan kun
startes for mobil tilmelding
verification.linkToken.is.present.for.non.verifying.state=Link token må kun være til
stede for "verifying" tilstand
verification.invalid.linkToken=Token må ikke være tom
verification.invalid.flowType=Den angivne type af flow er udyldig
verification.link.token.cannot.be.present.for.pin.flow=Verifikations link ikke
tilladt i pin verificerings flow
verification.pin.cannot.be.present.for.link.flow=Verifikations pin ikke tilladt i
link verificerings flow
```

Contact subscription services

```
contact.id.duplicated=Contact'en findes allerede i abonnement
contact.id.invalid=Kontakt id må ikke være blankt

url.invalid=Notifikationsadressen er ikke en gyldig URL
```

System subscription services

```
cvr.duplicated=Et eller flere CVR nummer optræder flere gange
cvr.invalid=CVR nummeret er ugyldigt
url.invalid=Notifikationsadressen er ikke en gyldig URL
```

System registry services

Error codes which can occur when creating or updating ContactPoints, ContactGroups, Systems and Organisations.

```
activefrom.needed.when.activeto.stated="aktiv fra" er påkrævet når "aktiv til" er
angivet
activefrom.invalid="Aktiv fra skal være før {0}"
```

```
activefrom.required="Aktiv fra skal angives"
activeto.not.allowed="Et standard modtagersystem kan ikke inaktiveres, og derfor kan
aktiv til ikke angives."
activeto.invalid={0} er ikke før {1}
active.invalid="Kontakt punktet kan ikke være aktivt da dets associerede modtager
system ikke er aktivt."
cvr.number.not.exist="CVR {0} does not exist"
active.period.invalid={0, choice, 1#Det følgende systems aktive periode sammenfalder
|1< De følgende systemers aktive periode sammenfalder}:{1} kun {2, choice, 1#1 aktivt
modtager system|1<{1, number, integer}aktive modtager systemer} er tilladt
authority.type.not.allowed=Kun myndigheder må have en myndigheds type
authority.type.invalid="{0}" er en ugyldig myndigheds type
contact.groups.circular.dependencies=Kontakt grupper må ikke have cirkulær
afhængighed
cvr.number.invalid="{0}" er et ugyldigt CVR nummer
delegated.cvr.does.not.exist=Fuldmagt kan ikke gives til et cvr-nummer der ikke
findes
danish.phone.number.needed=Dansk mobilnummer er påkrævet
delegated.cvr.cannot.equal.owner.organization.cvr=Et system kan kun deleges til et
andet CVR nummer end den organisation som systemet tilhører
danish.phone.number.invalid="{0}" er et ugyldigt dansk mobilnummer
email.address.needed=Email er påkrævet
email.address.invalid="{0}" er ugyldig email
externalLink.invalid="{0}" er et ugyldig URL
externalLinkText.needed=Eksternt link tekst påkrævet
endpoint.is.invalid="{0}" er et ugyldigt endepunkt
endpointCertificateType.is.invalid="{0}" er en ugyldigt certifikattype
field.value.is.needed=Værdi påkrævet
field.ip.range.not.singular=Når en IP addresseinterval angives må kun ét element
angives
cannot.parse.ip=Kan ikke læse IP addresse
cannot.parse.ip.range=Kan ikke læse IP addresseinterval
ip.list.too.big=For mange IP addresser angivet (over "{0}")
ip.list.not.unique=En eller flere IP addresser er ikke unikke
invalid.ip.range.size=Mængden af IP addresser i addresseintervallet er for stor (over
"{0}")
invalid.parent.group=Gruppen du forsøger at ligge "{0}" under er en ugyldig gruppe
invalid.system.type.delegated.cvr = "Only a sender / receiver system can be able to
be delegated to another CVR number"
oces.public.certificate.validation.failed=Det uploadede certifikat fejlede
valideringen med kode: {0}
oces.public.certificate.not.foces.or.voces=Det uploadede certifikat med subject
serial number {0} er ikke et funktions- eller virksomhedscertifikat
oces.public.certificate.not.issued.by.nets=Det uploadede certifikat er ikke signeret
med NETS rodcertifikat
oces.public.certificate.could.not.be.parsed=Det uploadede certifikat kunne ikke
læses, er det et gyldigt certifikat?
nets.root.cert.not.found=Kunne ikke finde NETS rod certifikat, kan ikke validere
uploadede certifikat
ssh.public.key.too.large=Den uploadede fil er for stor
ssh.public.key.missing=Den uploadede fil er tom
organisation.type.invalid="{0}" er en ugyldig organisations type
receiptEndpoint.is.invalid="{0}" er et ugyldigt kvitterings-endepunkt
```

```
sender.system.endpoint.is.not.null=Afsendersystemer kan ikke have et endepunkt
system.memoTransitionDateTime.is.before.use=Modtagersystemer kan ikke have et
overgangstidspunkt før {0}, da den nye MeMo version ikke er i brug
system.memoTransitionDateTime.is.after.expiration=Modtagersystemer kan ikke have et
overgangstidspunkt efter {0}, da den nye MeMo version skal bruges
recipient.system.receipt.endpoint.not.null=Modtagersystemer kan ikke have et
kvitterings-endepunkt
service.protocol.invalid.for.attaching.certificates=Et system med service protokol
{0} kan ikke have et tilknyttet certifikat
service.protocol.invalid.for.attaching.ssh-keys=Et system med service protokol {0}
kan ikke have en tilknyttet ssh-key
service.protocol.invalid.for.system.type=Et system med typen "{0}" kan ikke have
service protokollen "{1}"
service.protocol.not.supported="{0}" er ikke en understøttet service protokol
technical.contact.required.for.standard.system.template=Teknisk kontaktperson skal
udfyldes
service.protocol.sft.missing.ip=IP er påkrævet ved brug af service protokollen SFTP
targets.type.needed=Målgruppe er påkrævet
logo.organisationType.invalid=Logo er kun tilladt for myndigheder
logo.contentType.invalid="{0}" er ikke en understøttet filtype. Tilladt type er
"image/png". Sørg for at filen ender på .png.
logo.fileSize.invalid=Filstørrelsen på {0} bytes skal være mellem {1} og {2} bytes
logo.dimensions.invalid=Filens dimensioner skal være kvadratiske og minimum {0}px, {1
}px
logo.file.invalid=Filen kan ikke læses. Fejl: {0}
receiptEndpoint.not.empty=Modtagersystemer med service protokol REST_PULL kan ikke
have et kvitterings-endepunkt
endpoint.not.null=Modtagersystemer med service protokol REST_PULL kan ikke have et
endepunkt
code.version.is.needed=Klassifikations version skal angives for kontakt punkt med
kode type: {0}
contact.point.code.type.exist=Kode typen: {0}, eksisterer allerede for dette kontakt
punkt
contact.point.code.type.custom.limit.exceeded=Det er ikke tilladt at have {0} af
kontakt punkt kode typen CUSTOM
system.api.token.renew.not.allowed=Systemer med et certifikat må ikke opdatere deres
API Token
contact.group.delete.subgroup.exists=Kontaktgruppen kan ikke slettes, da den har
undergrupper
contact.group.delete.related.contact.point.exists=Kontaktgruppen kan ikke slettes, da
den har et eller flere kontaktpunker hørende til gruppen
system.has.contact.point=System id: {0} har kontaktpunkt tilknyttet: {1}
notification.email.standard.system.inheritance.property.is.null=Ikke valgt
notification.email.standard.system.closed.subject=Orientering om inaktiv
systemleverandør
notification.email.standard.system.closed.content.system.type.is.default.recipient.an
d.sender=Det betyder, at jeres aktive primærsystem ''{0}'' ikke vil blive
vedligeholdt. I kan dermed risikere, at systemet på et tidspunkt ikke længere kan
sende og modtage jeres post korrekt.
notification.email.standard.system.closed.content.system.type.is.recipient.and.sender
=Det betyder, at jeres aktive system ''{0}'' ikke vil blive vedligeholdt. I kan
dermed risikere, at systemet på et tidspunkt ikke længere kan sende og modtage jeres
post korrekt
```

```
notification.email.standard.system.closed.content.system.type.is.default.recipient=De
t betyder, at jeres aktive primærsystem ''{0}'' ikke vil blive vedligeholdt. I kan
dermed risikere, at systemet på et tidspunkt ikke længere kan modtage jeres post
korrekt.
notification.email.standard.system.closed.content.system.type.is.recipient=Det
betyder, at jeres aktive system ''{0}'' ikke vil blive vedligeholdt. I kan dermed
risikere, at systemet på et tidspunkt ikke længere kan modtage jeres post korrekt.
notification.email.standard.system.closed.content.system.type.is.sender=Det betyder,
at jeres aktive system ''{0}'' ikke vil blive vedligeholdt. I kan dermed risikere, at
systemet på et tidspunkt ikke længere kan sende jeres post korrekt.
```

Push notification settings error codes

These error codes can be encountered when creating/updating `Settings` in the push-notification-settings-store (see *"Push notification integrations"*). They're mostly for a few values for Firebase Cloud Messaging settings, where Google expects values in a certain format / value range. We refer to the official documentation to ensure creating well-formed settings: https://firebase.google.com/docs/reference/fcm/rest/v1/projects.messages

```
access.denied=Adgang nægtet
field.update.not.allowed=Dette felt må ikke opdateres
identity.does.not.exist=Den angivne identitet findes ikke
value.is.not.a.rrggbb.value=Den angivne værdi er ikke på formatet "#rrggbb",
eksempelvis #0fab88
value.is.not.a.rgba.value=Den angivne værdi er ikke ordentligt json objekt med
nøglerne 'red', 'green', 'blue', 'alpha', og værdi mellem 0 og 1
value.is.between.zero.and.one=Den angive værdi er ikke mellem 0 og 1 (inklusiv)
```

13.4.2 Back-end validation and error codes in distribution

This section describes the back-end error codes, that are returned when a message fails to be distributed. These error codes are returned asynchronously, either as receipts to sender systems, or as error messages in the mailbox. If nothing is mentioned, the error codes are returned in business receipts for messages received via Digital Post via all protocols.

13.4.3 Business receipt error codes

This subsection presents error codes in a block format, detailing the issues encountered during message distribution and the corresponding errors returned by the system.

Validation

```
memo.infected=Antivirus scan found threats
memo.invalid={0}
memo.version.not.allowed={0} is currently not a valid version
memo.root.invalid=Invalid XML root
memo.namespace.not.found=Missing memo xml namespace
message.body.not.found=MessageBody does not exist
```

```
do.not.deliver.until.date.too.early='Do not deliver until date' can not be in the
past
do.not.deliver.until.date.too.late='Do not deliver until date' is too late. Maximum
number of days allowed is {0}
message.uuid.does.not.match.file.name=The MessageUUID {0} does not match the UUID in
the filename {1}
message.uuid.not.unique=The MessageUUID {0} is invalid. MessageUUID must be a unique
UUID
file.name.invalid=Filename {0} is invalid. The format of the filename should be
'{'UUID'}' or '{'UUID'}'.xml
file.name.invalid.character=File name contains invalid character: {0}
contact.point.id.format.not.allowed={0} contactPointID {1} invalid. Expected format
UUID
notification.length.over.limit=Notification can not be longer than {0}
empty.notification.not.allowed=Empty notification is not allowed for MeMo of type
NEMSMS
reply.data.message.uuid.not.found=replyData missing message UUID
post.type.invalid=Post type: {0} is incorrect

# Recipient
recipient.is.closed=Recipient with {0} {1} is {2}
recipient.is.exempt=Recipient with {0} {1} is exempt
recipient.nem.sms.is.not.allowed=Recipient of type {0} can not receive nem sms
messages
recipient.contact.point.id.required=Contact point must contain a contact point id
recipient.contact.point.not.allowed.for.id.type=Contact point not allowed for
 recipient with id type {0}
recipient.cpr.invalid=The format of the cpr number: {0} is incorrect
recipient.cvr.invalid=The format of the cvr number: {0} is incorrect
recipient.not.found={0} with {1} {2} does not exist
recipient.nem.sms.subscription.not.found=Recipient with {0} {1} does not have a nem
sms subscription
recipient.nem.sms.subscription.mobile.number.not.verified=Recipient with {0} {1} has
not verified the mobile number {2}
recipient.mailbox.not.found=Recipient with {0} {1} does not have a mailbox
recipient.mailbox.and.default.recipient.system.not.found=Recipient with {0} {1} does
not have a mailbox or default recipient system
recipient.type.cannot.receive.legal.notifications=Recipient is of type {0} and
therefore cannot receive legal notifications.

# Sender
sender.not.found={0} with {1} {2} does not exist
sender.organisation.id.does.not.match=The sender organisation in the message does not
match {0} which was resolved when the message was received
sender.cpr.invalid=The format of the cpr number: {0} is incorrect
sender.cvr.invalid=The format of the cvr number: {0} is incorrect
sender.system.not.found=The sender system {0} which was resolved when the message was
received does not exist on the organisation {1}
sender.system.is.not.activated=The sender system {0} has not been activated yet. The
activation date of the system is {1}
sender.system.is.deactivated=The sender system {0} was deactivated at {1}
sender.mandatory.message.not.allowed=Sender is not allowed to send mandatory messages
```

```
sender.legal.notification.not.allowed=Sender is not allowed to send legal
notifications
sender.type.not.allowed=Only authorities can send messages to recipients of type {0}
sender.do.not.deliver.until.date.not.allowed=Senders of type {0} are not allowed to
send messages with a 'do not deliver until date'
sender.system.forward.not.allowed=Sender systems may not forward messages through
Digital Post
id.type.invalid=Invalid {0} id type {1}

# Representative
representative.cpr.invalid=The format of the cpr number: {0} is incorrect
representative.cvr.invalid=The format of the cvr number: {0} is incorrect

# Documents
memo.document.action.entrypoint.invalid=Invalid EntryPoint URL: [{0}]. HTTPS scheme
and valid uri required.

# Files
file.format.not.allowed=File encodingFormat(s) {0} for one or more files in {1}
document not allowed. Only the following are allowed for this type of document: {2}
file.extension.not.allowed=One or more invalid file exentions in one or more files is
not allowed: {0}
memo.file.size.too.large=File size of memo is too large. Allowed file size is {0}
bytes.
file.empty.not.allowed=One or more of the attachments in the message are empty
file.language.not.allowed=File language(s) {0} for one or more files in {1} document
not allowed. Only ISO 369-1 language codes are allowed.

# Html validator
html.validator.rejected=Filen {0} kunne ikke genkendes som et gyldigt html-dokument
html.validator.rejected.comments=Filen {0} indeholder kommentarer. Kommentarer er
ikke tilladt.
html.validator.rejected.element=Filen {0} indeholder element \"{1}\", som enten ikke
tilladt eller som indeholder data, der ikke er tilladt.
html.validator.rejected.element.attributes=Filen {0} indeholder element \"{1}\" med
attribut \"{2}\", der enten ikke er tilladt attribut, eller som indeholder data, der
ikke er tilladt.
html.validator.rejected.unknown-element=Filen {0} indeholder url i en ikke godkendt
placering. Det er sandsynligvis i en style attribut. Kun data url'er er tilladt.

# File and Document number validator
message.document.number.higher.than.allowed=The limit for the number of documents
that can be added to the message has been exceeded: {0}. Limit is {1}.
message.file.number.higher.than.allowed=The limit for the number of files that can be
added to the document \"{0}\" has been exceeded: {1}. Limit is {2}.
```

Extracting memos from archive

```
archive.processing.failed=An error occurred while processing the archive: {0}
no.archive.entry=No archive entry could be found in the file
```

```
file.name.uuid.is.not.valid=The file name {0} does not contain a valid UUID
```

13.4.4 SFTP error codes

The following error codes can be returned in technical receipts for bulk messages sent via SFTP

```
message.filename.invalid=Message filename {0} is invalid
file.is.empty=Filen {0} har ikke noget indhold.
transmission.uuid.not.unique=TransmissionId {0} er ugyldigt. TransmissionId skal være
entydigt
unknown.error=Der er opstået en ukendt fejl for filen {0}. Prøv at sende filen igen.
```

Recipient-system error codes

This section describes the errorCodes that Recipient-systems may set in the Business Receipt they send to DP upon receiving (REST_PUSH) or fetching (REST_PULL/REST_PUBLISH_SUBSCRIBE) MeMos from the solution.

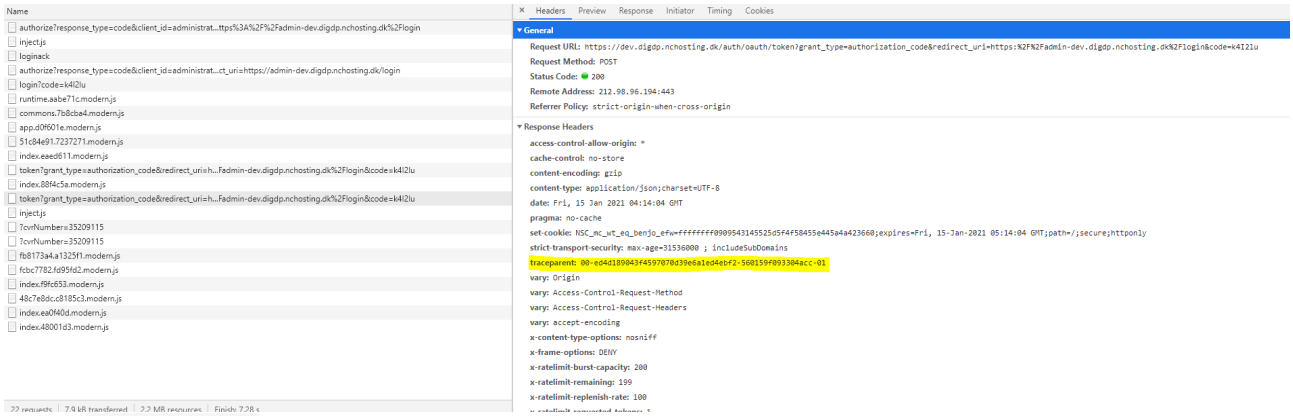
Business receipt errorcodes

```
virus.detected=Virus fundet i modtaget payload for MeMo med id {0}
```

13.4.5 Tracing requests using W3C headers

Digital Post provides support for tracing HTTP requests via W3C Trace Context specification for external parties. In summary, the following HTTP headers are available from the responses returned from request:

- `traceparent` : This HTTP header field identifies the incoming request in a tracing system. See the RFC for details. **This is the header that will be used for external request tracing**. See screenshot for example:



- `tracestate` : This HTTP header is to provide additional vendor-specific trace identification information across different distributed tracing systems and is a companion header for the `traceparent` field. See RFC for details.

The `tracestate` header is currently missing from the current implementation but it is NOT used in tracing requests, therefore, it is safe to ignore this header.

To request support, external parties can provide the `traceparent` header from the response to their requests. The value of this header is in the format similar to this example:
`00-0af7651916cd43dd8448eb211c80319c-b7ad6b7169203331-01`

14 Java/.Net Core, Security perspective, MeMo-lib and Test

14.1 Reference Systems for Java and .Net Core

14.1.1 Overall description and purpose

The purpose of the Reference Systems built for Java and .Net Core is to provide *code by example* by providing reference implementations of sender- and recipient-systems for Digital Post (DP). Authorities and developers can use the examples in the Reference Systems to gain an overall understanding of how these implementations can be programmed, and what systems must be able to handle in the interaction with DP.

The reference implementation contains examples of sending- and/or receiving MeMos as well as receipts across these protocols:

- REST_PUSH and REST_PUBLISH_SUBSCRIBE (publish/subscribe using long-polling)
- SFTP (sender-system only)

14.1.2 Supported platforms

The Reference Systems have been built for both Java and .Net Core, and made available as public Bitbucket repositories. Naturally, there are some differences between the Java and .Net Core versions, but there has been an emphasis on code-reuse and streamlining the implementation as much as possible to reduce the scope of differences. The available versions can be found here:

- Java version (https://bitbucket.org/nc-dp/reference-systems-for-java/src/master/)
- .Net Core version (https://bitbucket.org/nc-dp/reference-systems-for-dotnet/src/master/)

14.1.3 Application architecture

The application architecture for both versions of the Reference Systems involves separating responsibilities and protocols into individual sub-modules.

Each sub-module can act as an independent runnable application to trigger specific flows. (The .Net pendant to sub-module is a “Project” within a “Solution”).

Sub-modules (Java name / .Net Core name)

ssl-client / SSLClient

Is used for authentication and used for handling mutual SSL handshake between the Reference Systems and DP. The client allows one to use a certificate in combination with an API-key to ensure validation.

utility-library / UtilityLibrary

Provides model and service resources for MeMos and receipts, handling the creation, parsing and logging of MeMos as well as the creation, sending and logging of positive or negative receipts.

system-rest-push / RestPush

Reference implementation of REST_PUSH protocol sender- and recipient-system. Showcases how REST requests to DP can be implemented, utilizing the RestClient provided by ssl-client - and how the recipient-systems can provide an endpoint for receiving MeMos from DP - as well as the processing of these received MeMos and the creation and sending of Business Receipts back to DP.

system-rest-publish-subscribe / RestPublishSubscribe

Reference implementation of REST_PUBLISH_SUBSCRIBE protocol recipient-system. Showcases how REST calls can be made to DP to fetch information about MeMos currently available for fetching, as well as fetching these one by one from DP. Also contains example code to create and sends back Business Receipts to DP.

system-sftp / Sftp

Reference implementation of SFTP protocol sender-system. Showcases how created MeMos can be bundled to a TAR.LZMA file and uploaded to DP's SFTP server - as well as downloading available receipts from the SFTP server.

14.1.4 REST protocol examples

Overall description

The Reference Systems REST examples are configured to represent an organisation integrated to Digital Post (DP) with a REST_PUSH protocol sender- and recipient-system (system-rest-push sub-module) and a REST_PUBLISH_SUBSCRIBE recipient-system (system-rest-publish-subscribe sub-module).

For the Reference Systems REST protocol examples, the two endpoints of sender- and recipient-systems are exemplified by exposed RestController endpoints that provide examples of how MeMos and receipts can be received and processed:

- Handling of receipts being sent to the *receiptEndpoint* of a system is exemplified by an endpoint exposed by the Reference Systems application, which can process the receipt and print relevant information to the console.
- Handling of a validated and processed MeMo being sent to the *endpoint* of a system is likewise exemplified by an endpoint exposed by the Reference Systems application, which can process the MeMo and print relevant information to the console.

Purpose

It is the aim of the Reference Systems REST protocol examples to provide insight into the interaction between sender- and recipient-systems and DP, with extensive console logging and in-code commentary utilized to describe each process. This is achieved through example code showcasing how MeMos and receips can be created and sent as well as received and processed.

The examples for REST_PUSH can be found within the Reference Systems repositories here:

- Java: https://bitbucket.org/nc-dp/reference-systems-for-java/src/master/system-rest-push/
- .Net Core: https://bitbucket.org/nc-dp/reference-systems-for-dotnet/src/master/RestPush/

And for REST_PUBLISH_SUBSCRIBE:

- Java: https://bitbucket.org/nc-dp/reference-systems-for-java/src/master/system-rest-publish-subscribe/
- .Net Core: https://bitbucket.org/nc-dp/reference-systems-for-dotnet/src/master/RestPull/

REST flow

The Reference Systems contains example implementations of the 4 primary REST steps, described below.

1. The sender-system sends a MeMo, e.g to a recipient-system. DP receives the MeMo and returns a Technical receipt in the response body.
2. DP validates and processes the MeMo and sends back a business receipt for the sender-system (for REST_PUSH), to notify whether this step was successful. For REST_PULL the business receipt is made available.

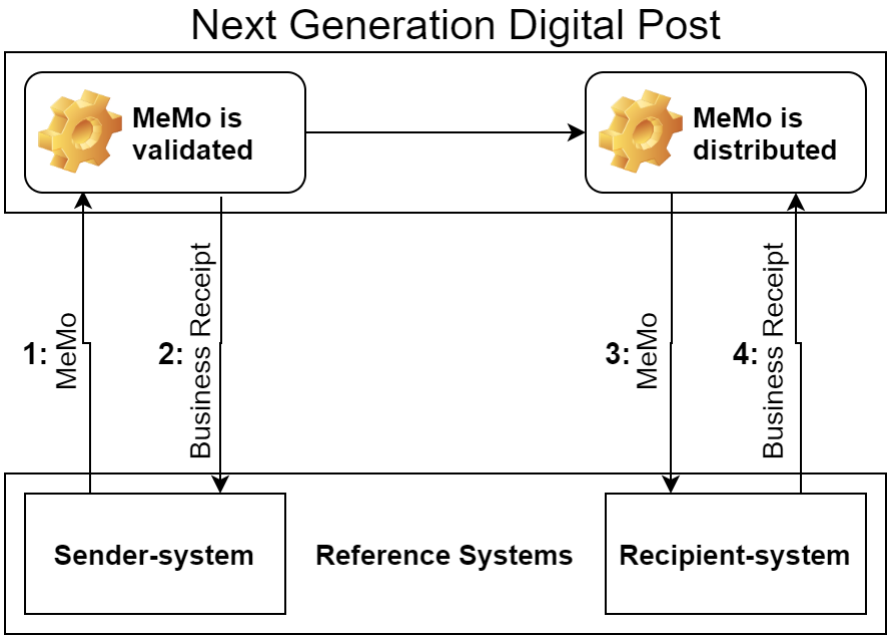
3. If successful, DP distributes the MeMo to the (in this example) recipient-system (REST_PUSH), or the recipient-system fetches the MeMo from DP (REST_PUBLISH_SUBSCRIBE or REST_PULL).
4. Recipient-system creates and sends back a Business Receipt to DP to notify whether it successfully received the MeMo (and DP uses this information to delete the MeMo from internal storage).

In the implementation of the Reference Systems, observing the steps of the REST protocol flows can be initiated by running either the system-rest-push or system-rest-publish-subscribe applications.

Running the system-rest-push application will create a single MeMo as well as a tar.lzma of 3 MeMos, and send these to DP, with the recipient being configurable. This mimics the process of sending MeMos as a sender-system, as DP will handle everything from there onwards (given that the MeMo adheres to the agreed format and that the recipient is reachable).

Running the system-rest-publish-subscribe application will start the process of the REST_PUBLISH_SUBSCRIBE recipient-system fetching a list of available MeMos and attempting to fetch each of these from DP.

A simplified representation of the Reference Systems application and how it showcases the interactions that sender- and recipient-systems will have with DP is presented below. It exemplifies the usage of receipts to communicate whether actions where successful or not.



Reference REST sender-system

The following section will describe the two primary REST protocol interactions between sender-systems and DP. They are:

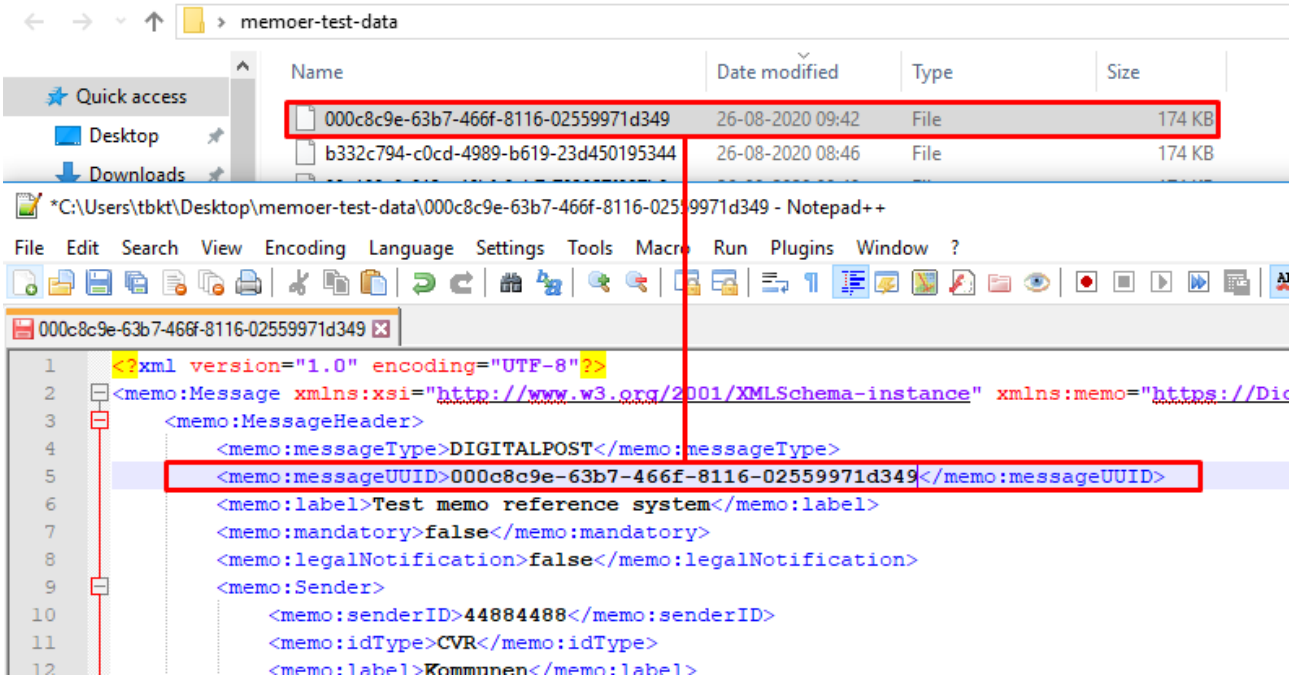
1. Sending MeMo
2. Receiving Receipt

Sending MeMo

The Reference sender-system can send a MeMo intended to the DP test environment. The DP endpoint is {environmentUrl}/memos/, exposed at the following path on e.g. test01: https://api.test.digitalpost.dk/apis/v1/memos/. The endpoint consumes a Resource - in the format of either:

- A tar.lzma file with content-type ‘application/x-lzma’ containing XML files
- An XML file with content-type ‘application/xml’

Additionally, the name of the XML files should be the messageUUID from the MeMo. The .xml extension is optional. See picture below:



In the case of the Reference Systems application, the service is able to create MeMo messages that adhere to these rules, which besides actively being sent by the Reference sender-system, will also give insight into the structure and contents of a MeMo.

For an in-depth overview of inbound REST services, see chapter *Inbound services* in *Digital Post - Technical Integration* for more information.

Initiating flow

Sending a MeMo is doable by running the appropriate application within the Reference System application. Running the system-rest-push application will create a single MeMo separately as an XML as well as a tar.lzma containing 3 MeMos, and proceed to send these. The system-rest-push sub-module utilizes a service in the utility-library sub-module to create MeMos programmatically by utilizing memo-lib. Running this application also exposes the REST_PUSH recipient-system endpoint which provides an example to how MeMos can be received.

Receiving receipts from DP

Immediately upon receiving the MeMo, DP will send back a Technical Receipt. If this step was successful, DP will validate and process the MeMo and send a Business Receipt back to the sender-system at the *receiptEndpoint* URL. The sender-system responds with a HTTP Status code to notify DP of successful or unsuccessful delivery of the receipt. The Reference Systems contain example code showcasing this interaction. This Business Receipt contains information on the status of DP's processing of the MeMo. (see chapter 4.9.3 *Receipt domain model* and chapter 4.9.4 *REST receipts* in *Digital Post - Technical Integration* for more information).

Specifically for the Reference sender-system example, the receipts from DP will be logged to the console for an overview of its content and status. A positive Business Receipt to the sender-system notifies that the MeMo is now considered the responsibility of DP, and this stage marks the end of interaction between the sender-system and DP for the respective MeMo.

Reference REST recipient-system

Receiving MeMo

There are two ways in which a REST protocol recipient-system can receive MeMos.

REST_PUSH

If the recipient of a MeMo is a REST_PUSH recipient-system, DP will send MeMos to the recipient-system endpoint as soon as these have been validated by DP. An example of this is exposed by a RestController in the system-rest-push sub-module, which will initiate the processing of any received MeMo from DP.

REST_PUBLISH_SUBSCRIBE

If the recipient of a MeMo is a REST_PUBLISH_SUBSCRIBE recipient-system, DP will not send MeMos immediately upon validation, but instead send a notification to the recipient-system, allowing the recipient-system to fetch them when needed. An example of this functionality is implemented in the system-rest-publish-subscribe sub-module, where the recipient-system can call two DP endpoints: One for fetching a list of available MeMos - and one for fetching each available MeMo in this list.

Processing MeMo

When the MeMo has been received through either REST_PUSH or REST_PUBLISH_SUBSCRIBE, the following process of parsing it and sending back Business Receipts is identical.

The Reference System examples implements functionality of the publicly available MeMo-lib, by utilizing a parser to parse the MessageHeader from the received MeMo.

Relevant information from the MeMo MessageHeader will then be logged to the console for an overview of its content.

Draft example shown below of MeMo MessageHeader parsed and logged to console:

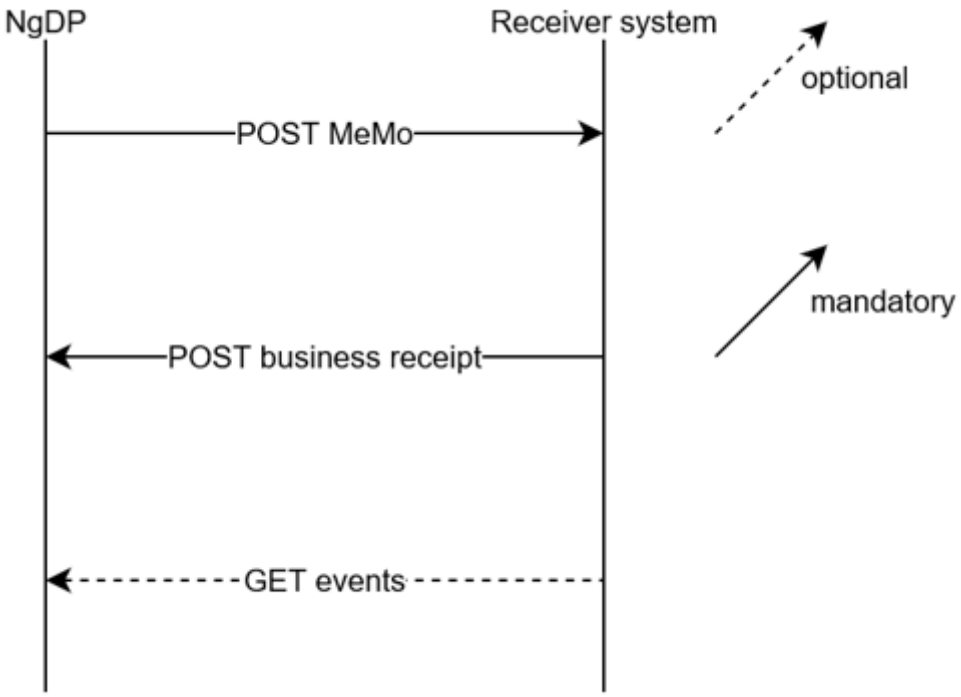
```
2020-09-01 08:19:55.048  INFO 4748 --- [nio-8081-exec-2] d.d.d.p.r.s.r.s.MemoLoggerService
  messageType=DIGITALPOST
  messageUUID=eb03da92-fb62-4880-976e-3f92bd25e2a4
  messageId=<null>
  messageCode=<null>
  label=Lønseddel-Jan
  notification=<null>
  additionalNotification=<null>
  reply=<null>
  replyByDateTime=<null>
  doNotDeliverUntilDate=<null>
  mandatory=false
  legalNotification=false
  sender=dk.digst.digital.post.memolib.v1.model.Sender@78dfc435
  recipient=dk.digst.digital.post.memolib.v1.model.Recipient@8434f8a
  contentData=<null>
  forwardData=<null>
  replyData=<null>
```

Sending Business Receipt to DP

When the Reference recipient-system has parsed the received MeMo, it will send a Business Receipt back to DP. The memoId is a PathVariable, which must be the UUID of the MeMo. This informs DP which specific MeMo the Business Receipt is a response to.

The Reference System will automatically create a Business Receipt if it receives a MeMo, and send it to the correct endpoint. If the Reference System application encounters an error in the parsing of the MeMo MessageHeader, this error will populate the ErrorMessage field of the Business Receipt.

Thus, the Business Receipt as built by the Reference System application can be either positive or negative, dependent on (in the context of the Reference System application) whether the MeMo sent from DP was passable.



Upon receiving a positive Business Receipt from a recipient-system, DP will delete the MeMo from internal storage.

Configuring mutual SSL for REST flow

The examples showcased are from the Java implementation, however the .Net implementation is similar.

There are two main rules for a correct setup with a valid certificate:

- The CVR of the certificate must be the same as the Organisation's CVR that the sender-system is a part of - or alternatively the sender-system must have a "Systemfuldmagt" which points to that CVR.
- The API-key must match the API-key of the sender-system that is to be used for sending.

The certificate and API-key is easily configured in the Reference Systems. In the below example from the Java version, egress-rest-client.properties (located in \reference-systems-for-java\ssl-client\src\main\resources\configI) is setup to allow for plug-n-play of valid certificates and API-keys.

For example, let's say we have an Organisation with CVR 64942212, with a sender-system we want to send with:

```
# keystore and truststore locations should be common to all components
```

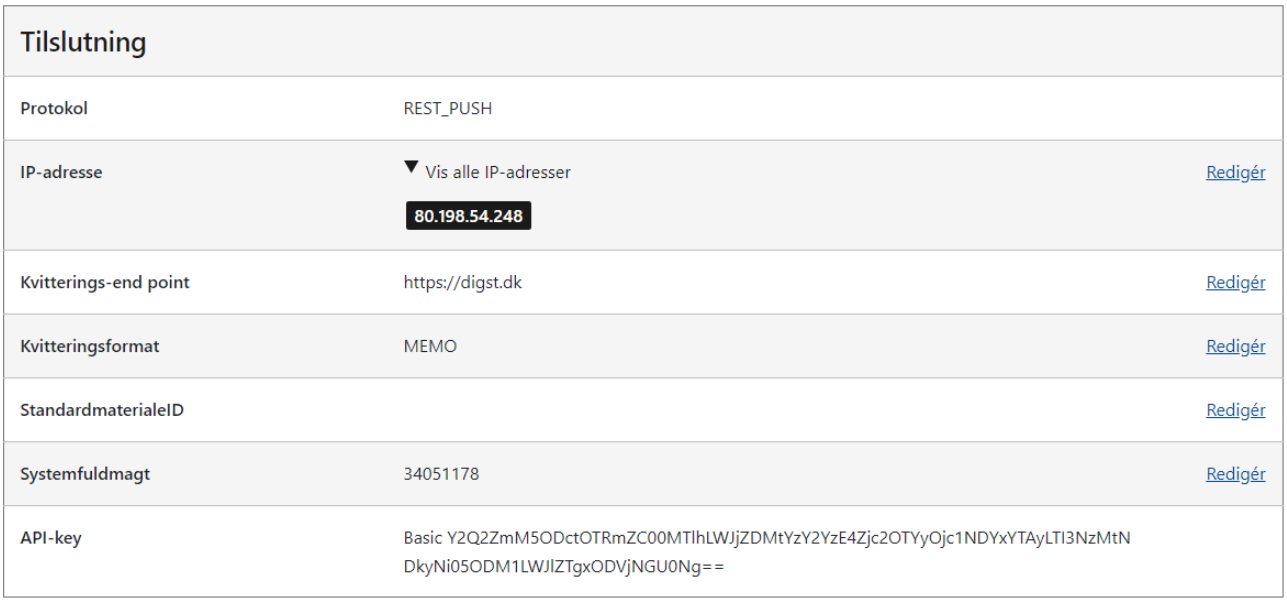
```
dk.digst.digital.post.egress-rest.keyStoreLocation=classpath:/sender-system-keystore/
VOCES_gyldig_2022.p12
dk.digst.digital.post.egress-rest.keyStorePassword=Test1234
dk.digst.digital.post.egress-rest.type=PKCS12
dk.digst.digital.post.egress-rest.alias=VOCES_gyldig_2022.p12
dk.digst.digital.post.egress-rest.trustselfsigned=false
dk.digst.digital.post.egress-
rest.senderSystemApiToken=OTc5MzUxMDEtNGY1Mi00MzE2LTk4YzktYTdjZWQ5NzI5YzdiOmM0MmU5ZTB
hLWZhY2ItNGE0ZS1hYWViLTNlNTE0NzVkOGY5MQ==
dk.digst.digital.post.egress-rest.readtimeout=30000
dk.digst.digital.post.egress-rest.connecttimeout=60000

# proxy is optional
dk.digst.digital.post.egress-rest.proxy=
```

- We must point the SSL client to a valid certificate with the CVR 64942212:
  - **keyStoreLocation**: Location of the certificate.
  - **keyStorePassword**: Password of the certificate.
  - **type**: Type of the certificate.
  - **alias**: Name of the certificate.
- The API-key must match the API-key of the sender-system we want to send as (can be found in Administrative Access).
  - **senderSystemApiToken**: The API-key of the sender-system (without the “Basic “ prefix)

Thus, in the above example, the “VOCES_gyldig_2022.p12” certificate must have the CVR 64942212, and the senderSystemApiToken
(`OTc5MzUxMDEtNGY1Mi00MzE2LTk4YzktYTdjZWQ5NzI5YzdiOmM0MmU5ZTBhLWZhY2ItNGE0ZS1hYWViLTNlNTE0NzVkOGY5MQ==`) must be present on a sender-system on the Organisation.

An exception to the first rule is when “Systemfuldmagt” is used. This allows a sender-system to send on behalf of another CVR, example:



The above pictured sender-system would allow us to send with a certificate that has the CVR 34051178 - even though the Organisation this sender-system is a part of does not have this CVR.

14.1.5 SFTP protocol examples

Overall description

The Reference Systems SFTP application are configured to represent an organisation integrated to Digital Post (DP) with a SFTP protocol sender-system, as recipient-systems can not be SFTP type.

Purpose

It is the aim of the Reference Systems SFTP protocol examples to provide insight into the interaction between sender-systems and DP, with extensive console logging and in-code commentary utilized to describe each process.

The examples for SFTP can be found within the Reference Systems repositories here:

- Java: https://bitbucket.org/nc-dp/reference-systems-for-java/src/master/system-sftp/
- .Net Core: https://bitbucket.org/nc-dp/reference-systems-for-dotnet/src/master/Sftp/

SFTP flow

The Reference Systems contains example implementations of the 3 primary SFTP steps, described below:

1. A sender-system uploads a MeMo to DP's SFTP server.
2. DP fetches the MeMo, and returns a receipt to the SFTP server.
3. The sender-system fetches the receipt from the SFTP server.

In the implementation of the Reference Systems, observing the steps of the SFTP protocol flow can be initiated by running the system-sftp application.

When this application is run, utility-library will be utilized to create 3 MeMos and adding these to a tar.lzma file. This tar.lzma file will automatically be added as the payload to a request to the SFTP server, and the request will be made and the tar.lzma sent.

The application is configured to use a poller which polls the receipt folder on the SFTP server every second for new entries. Whenever a receipt is sent to the SFTP server by DP, the application will shortly thereafter fetch it, log it to console and delete it.

14.2 Security Perspective

14.2.1 Allowed certificate cipher suites

Not all cipher suites are allowed when accessing Digital Post, as many are outdated or insecure. These are the allowed cipher suites:

Suites
TLS1.3-AES256-GCM-SHA384
TLS1.3-AES128-GCM-SHA256
TLS1.2-ECDHE-RSA-AES256-GCM-SHA384
TLS1.2-ECDHE-RSA-AES128-GCM-SHA256

14.3 MeMo-lib

14.3.1 Purpose

The library should provide a simple API to construct and serialize messages in the MeMo format. By utilizing the builder pattern it should be possible to construct a representation of the message and let the library handle the serialization to a zip archive. The file must contain a manifest file and one or more MeMo message file(s) in either xml or json format.

The library should also provide a simple API to access the zip archive. This includes verifying the correctness of the content including the actual MeMo message(s) and the manifest file. The API should also include methods to deserialize the MeMo message.

14.3.2 Repository

MeMo-lib is currently available in Java and .NET. It is accessible at the following repositories:

https://bitbucket.org/nc-dp/memo-lib-java/src/

https://bitbucket.org/nc-dp/memo-lib-dot-net/src

For information and usage, read the readme files in the repositories.

15 Access to Test environments

15.1 Access to the administration portals on the test environment

The administration of your Digital Post solution in the test environment is done through Test Portal and Administrative Access. The Test Portal provides the possibility to create fictive citizens and companies. The test companies can be administered in Administrative Access. Fictive citizens can receive messages sent by your sender system, which can be viewed in the demo client of https://post.demo.borger.dk/.

15.1.1 **Step 1 - Create a MitID simulator identity**

To gain access to the Digital Post Test Portal and create test users, privileges etc., you need to create a test user identity via the MitID-simulator https://mitidsimulator.test-devtest4-nemlog-in.dk/Home/Create

Make sure that Maximum Authentication Assurance level is set to “Substantial”.

Also note, that if you check off “Private MitID” the CPR number entered should exist in the Digital Post test environment.

MitID Simulator

Search identity | Create identity

Identity data

Autofill

Maximum Authentication Assurance Level

Substantial

Username

KimOlsenInc

Password

****

First name

Kim

Middle name

Last name

Olsen

15.1.2 Step 2 - Get the new identity enrolled in DP test environment

To make your new test identity work in the Digital Post solution, you need to create a service request in DP's Servicedesk.

You create a service request by:

1. Access DP's Servicedesk https://digidp.atlassian.net/servicedesk/customer/portal/
2. Create a service request by selecting "Service Request: Access to test environment".
3. Enter the **username** of the MitID identity and **CVR** of the organization the identity should be associated to. Be aware that the CVR number should be an existing CVR number. All real CVR numbers are copied to the test environment, so you are encouraged to choose your real CVR number.
4. Press "send".

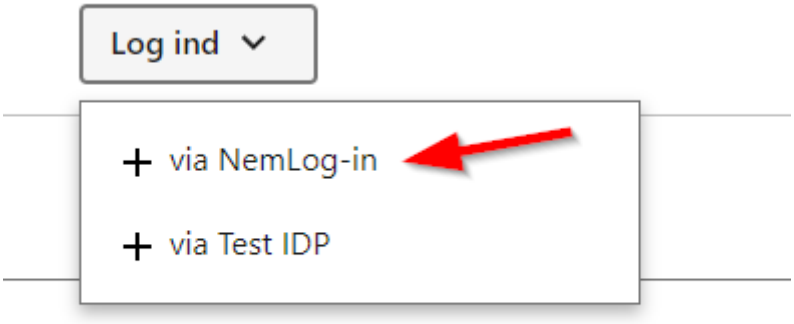
After pressing "Send", Netcompany will process your request and associate the MitID identity to the Digital Post test environment which enables you to log in via NemLog-in.

15.2 Access to Test Portal on the test environment

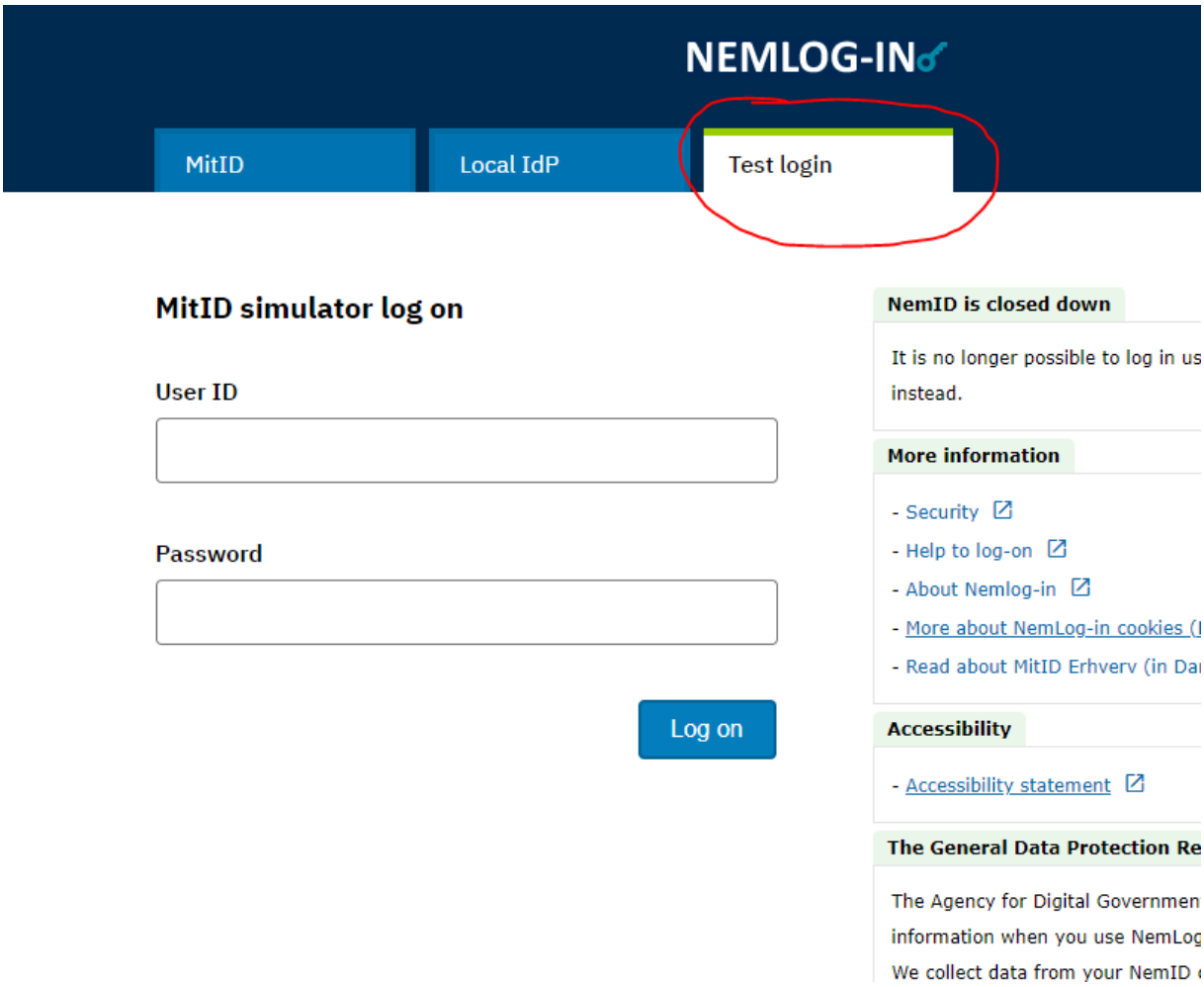
Once step 2 is completed you can login to the Test Portal using NemLog-in -> Test login.

The Test Portal can be accessed using this link: https://testportal.test.digitalpost.dk/login

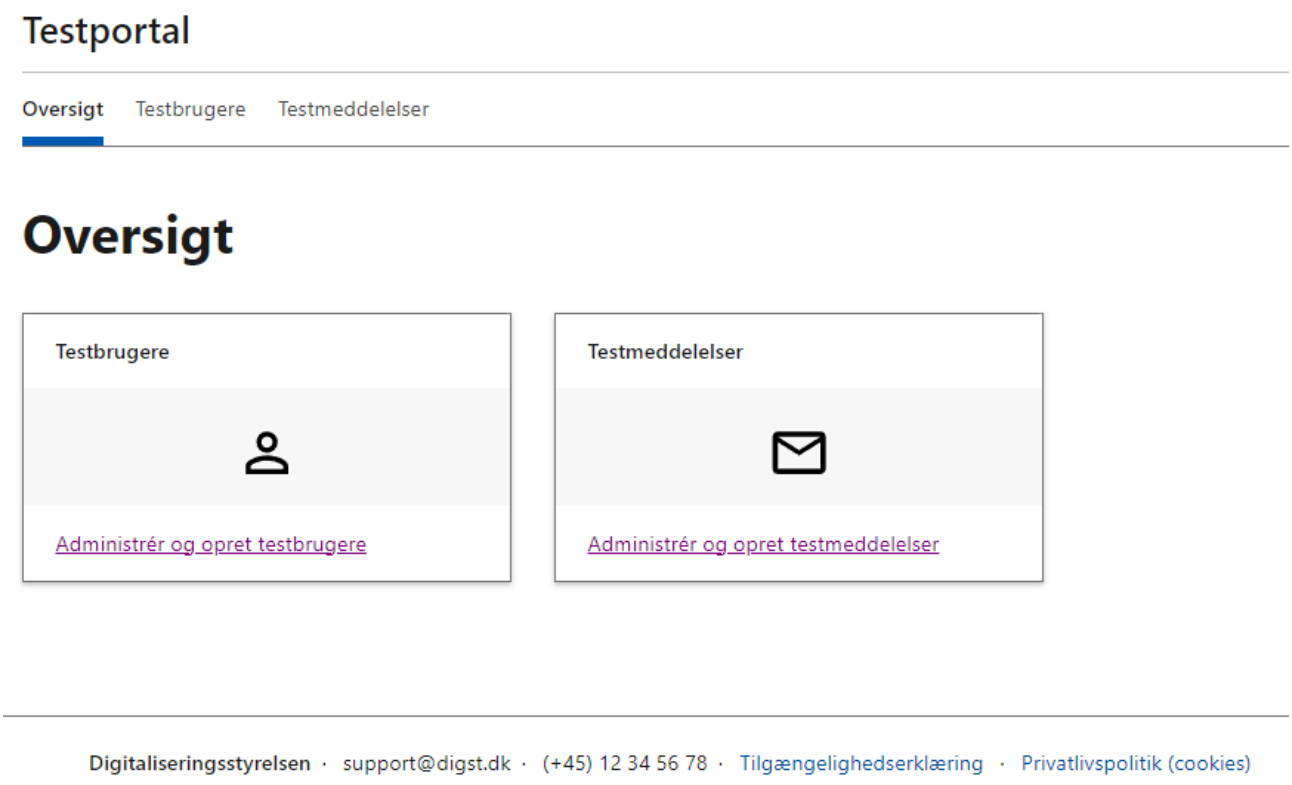
You are asked to login using either Test IDP or NemLog-in. Select “NemLog-in”.



Thereafter you are redirected to Nemlog-in. Select the tab “Test login”. Enter your username from the MitID simulator identity and password.



After a successful authentication, the user should have access to the overview page as shown below. The Test Portal makes it possible to create test users via “Testbrugere” or send Digital Post via “Testmeddelelser”.



The Test Portal can be accessed via https://testportal.test.digitalpost.dk/login.

A guide for the Test Portal can be found via https://digitaliser.dk/digital-post/vejledninger/testportalen.

15.3 Access to Administrative Access on the test environment

In Administrative Access it is possible for companies to create sender- and receiver systems, lookup logs or retrieve statistics. For authorities it is also possible to create contact points and for citizen service employees to exempt contacts or create nem-sms subscriptions. Access to the different functionalities depends on the privileges of the user logged in and whether the organisation associated is registered as a company or an authority. For test idp users, privileges are granted in Test Portal. For nemlog-in users, privileges are granted in Rights Portal.

You will have two login options: via Nemlog-in or via Test IDP.

If you select “Via NemLog-in”, you login using your MitID simulator identity.

If you select “Via Test IDP”, you will can login with the test users created in the Test Portal.

After a successful authentication, the user should have access to the overview page as shown below, where the user has been given the role ‘Kontaktstrukturadministrator'. Different roles give access to different functionality, as described in 'Vejledning til Rettighedsportalen’.

Administrative Access can be accessed via https://admin.test.digitalpost.dk/login

Rights Portal can be accessed via https://rettighedsportal.test.digitalpost.dk/

Virk.dk's test environment can be accessed via https://pilot.virk.dk/

Find guides to the different portals at https://digst.dk/it-loesninger/digital-post/vejledninger-og-begivenheder/

15.3.1 Prerequisites for setting up test systems

To create systems in Administrative Access, the user logged in must have the privilege of a "system manager / systemadministrator". For test IDP users, this privilege can be assigned in Test Portal. For nemlog-in users, the privilege must be assigned in the Rights Portal.

In addition, it requires:

- a **NemLog-in OCES TEST-certificate** for the test environment.
- a modern **TLS-versions (+1.2) and cipher suites (See 'Allowed certificate cipher suites')**

For more information on how to connect to the test environment please see "Connect to Digital Post".

To be able to send Digital Post on the test environment, your system must have a OCES3 TEST-certificate. This is acquired via Nemlog-in. Follow this guide to create a certificate https://www.nemlog-in.dk/vejledningertiltestmiljo/.

Be aware that the test organisations mentioned in this guide cannot be used in Digital Post's test environment. Only the certificates issued in DevTest4 can be used in Digital Post if it is connected to your real CVR number or a fictive CVR number in the Digital Post test environment.

Also note that for being able to send Digital Post to companies or citizens the sender system must be registered to an authority.

16 Troubleshooting, SFTP server, SDLC, OpenID Connect, Connect

16.1 Troubleshooting

These are the typical problems encountered during testing:

- Wrong IP-address.
    - Is it the correct IP address you have added in Administrativ Adgang you are calling from?
- Nemlogin Production certificate used **instead of Nemlogin test certificate**.
    - Has the certificate expired?
- Need to call a specific end point - otherwise you are stuck in the firewall.
- Old ciphers (see 'Prerequisites')
- Non-existent CPR-/CVR number in the dataset.

16.1.1 Certificate policies

Sender and receiver systems must expose *their* endpoints with mutual SSL. Similar to when a sender or receiver system is calling Digital Post, where Digital Post is exposing a “web certificate” as oppose to an OCES certificate. When Digital Post is calling you, Digital Post is using the OCES certificate and you are expected to identify using a “web certificate” as defined in this section.

Format

Web certificates are required to be provided in the X.509 standard, https://en.wikipedia.org/wiki/X.509. As well as the entire trust chain.

Validity

Every certificate has a validity period. A certificate may be either:

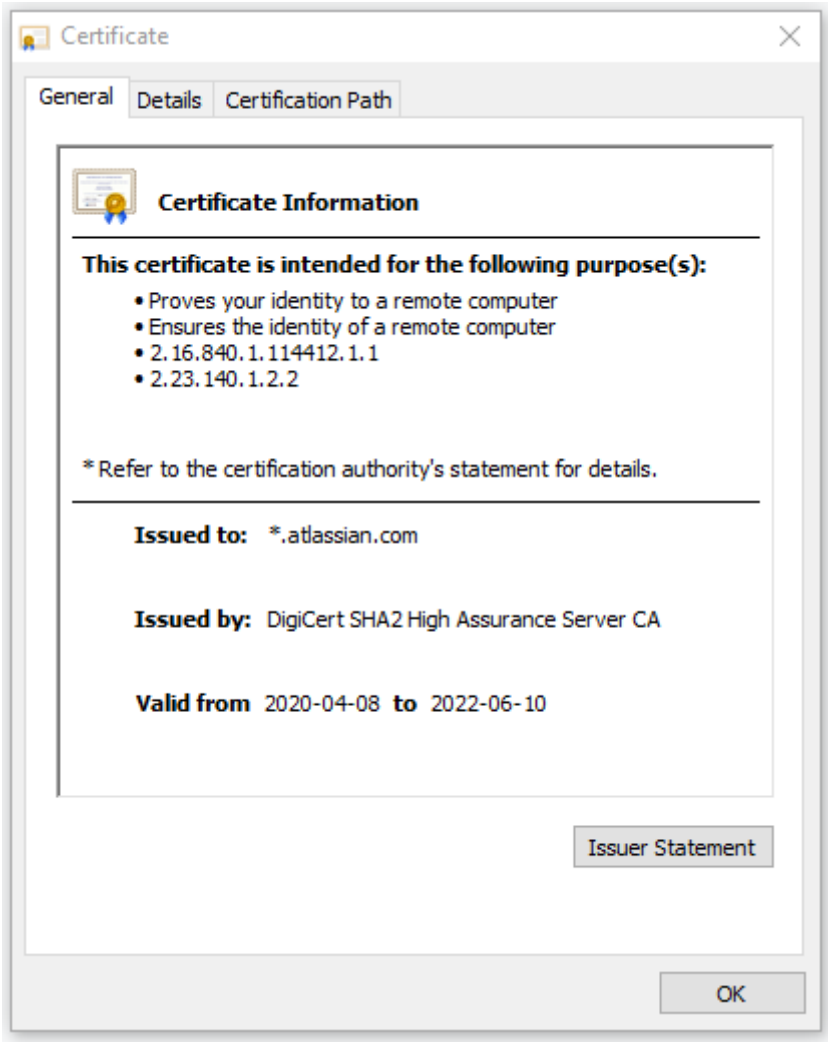
- not yet valid
- **valid**
- expired

Please mind the dates and replace certificates when they are getting close to expiration date.

> It is advisable to have a **defined process of certificate replacement**. A new certificate can have overlapping validity period with the old one, thus certificates can be replaced in real-time without any consequences.

Hostname verification

In order for Web certificate to be verified positively “issued to” field containing domains allowed for the certificate must **match the domain in URL**. An example: a page is hosted on a URL starting with https://test.atlassian.net/… and it’s **valid** certificate is issued for all subdomains of atlassian.net (so called star certificate).



Accepted issuer CAs

When verifying certificates it is crucial to have them issued by CA (Certificate Authority) that is included in the Oracle Java Root Certificate Program since they are included in the distribution Oracle's Java Runtime Environment (JRE). Almost all typical certificate issuers are already included, however, mind that your issuer might is not supported. **It is a client's responsibility to use keys acceptable by Java.** See https://www.oracle.com/java/technologies/javase/carootcertsprogram.html for details.

	issuer name	fingerprint (SHA-1)	valid from	valid to
1	AAA Certificate Services	D1:EB:23:A4:6D:17:D6:8F:D9:25:64:C2:F1:F1:60:17:64:D8:E3:49	2004-01-01T00:00Z	2028-12-31T23:59Z

	issuer name	fingerprint (SHA-1)	valid from	valid to
2	AC RAIZ FNMT-RCM	EC:50:35:07:B2:15:C4:95:62:19:E2:A8:9A:5B:42:99:2C:4C:2C:20	2008-10-29T15:59Z	2030-01-01T00:00Z
3	AC RAIZ FNMT-RCM SERVIDORES SEGUROS	62:FF:D9:9E:C0:65:0D:03:CE:75:93:D2:ED:3F:2D:32:C9:E3:E5:4A	2018-12-20T09:37Z	2043-12-20T09:37Z
4	Actalis Authentication Root CA	F3:73:B3:87:06:5A:28:84:8A:F2:F3:4A:CE:19:2B:DD:C7:8E:9C:AC	2011-09-22T11:22Z	2030-09-22T11:22Z
5	AddTrust External CA Root	02:FA:F3:E2:91:43:54:68:60:78:57:69:4D:F5:E4:5B:68:85:18:68	2000-05-30T10:48Z	2020-05-30T10:48Z
6	AddTrust Qualified CA Root	4D:23:78:EC:91:95:39:B5:00:7F:75:8F:03:3B:21:1E:C5:4D:8B:CF	2000-05-30T10:44Z	2020-05-30T10:44Z
7	AffirmTrust Commercial	F9:B5:B6:32:45:5F:9C:BE:EC:57:5F:80:DC:E9:6E:2C:C7:B2:78:B7	2010-01-29T14:06Z	2030-12-31T14:06Z
8	AffirmTrust Networking	29:36:21:02:8B:20:ED:02:F5:66:C5:32:D1:D6:ED:90:9F:45:00:2F	2010-01-29T14:08Z	2030-12-31T14:08Z
9	AffirmTrust Premium	D8:A6:33:2C:E0:03:6F:B1:85:F6:63:4F:7D:6A:06:65:26:32:28:27	2010-01-29T14:10Z	2040-12-31T14:10Z
10	AffirmTrust Premium ECC	B8:23:6B:00:2F:1D:16:86:53:01:55:6C:11:A4:37:CA:EB:FF:C3:BB	2010-01-29T14:20Z	2040-12-31T14:20Z
11	Amazon Root CA 1	8D:A7:F9:65:EC:5E:FC:37:91:0F:1C:6E:59:FD:C1:CC:6A:6E:DE:16	2015-05-26T00:00Z	2038-01-17T00:00Z
12	Amazon Root CA 2	5A:8C:EF:45:D7:A6:98:59:76:7A:8C:8B:44:96:B5:78:CF:47:4B:1A	2015-05-26T00:00Z	2040-05-26T00:00Z

	issuer name	fingerprint (SHA-1)	valid from	valid to
13	Amazon Root CA 3	0D:44:DD:8C:3C:8C:1A:1A:58:75:64:81:E9:0F:2E:2A:FF:B3:D2:6E	2015-05-26T00:00Z	2040-05-26T00:00Z
14	Amazon Root CA 4	F6:10:84:07:D6:F8:BB:67:98:0C:C2:E2:44:C2:EB:AE:1C:EF:63:BE	2015-05-26T00:00Z	2040-05-26T00:00Z
15	ANF Secure Server Root CA	5B:6E:68:D0:CC:15:B6:A0:5F:1E:C1:5F:AE:02:FC:6B:2F:5D:6F:74	2019-09-04T10:00Z	2039-08-30T10:00Z
16	Autoridad de Certificacion Firmaprofesional CIF A62634068	AE:C5:FB:3F:C8:E1:BF:C4:E5:4F:03:07:5A:9A:E8:00:B7:F7:B6:FA	2009-05-20T08:38Z	2030-12-31T08:38Z
17	Baltimore CyberTrust Root	D4:DE:20:D0:5E:66:FC:53:FE:1A:50:88:2C:78:DB:28:52:CA:E4:74	2000-05-12T18:46Z	2025-05-12T23:59Z
18	Buypass Class 2 Root CA	49:0A:75:74:DE:87:0A:47:FE:58:EE:F6:C7:6B:EB:C6:0B:12:40:99	2010-10-26T08:38Z	2040-10-26T08:38Z
19	Buypass Class 3 Root CA	DA:FA:F7:FA:66:84:EC:06:8F:14:50:BD:C7:C2:81:A5:BC:A9:64:57	2010-10-26T08:28Z	2040-10-26T08:28Z
20	CA Disig Root R2	B5:61:EB:EA:A4:DE:E4:25:4B:69:1A:98:A5:57:47:C2:34:C7:D9:71	2012-07-19T09:15Z	2042-07-19T09:15Z
21	Certigna	B1:2E:13:63:45:86:A4:6F:1A:B2:60:68:37:58:2D:C4:AC:FD:94:97	2007-06-29T15:13Z	2027-06-29T15:13Z
22	Certigna Root CA	2D:0D:52:14:FF:9E:AD:99:24:01:74:20:47:6E:6C:85:27:27:F5:43	2013-10-01T08:32Z	2033-10-01T08:32Z
23	certSIGN ROOT CA	FA:B7:EE:36:97:26:62:FB:2D:B0:2A:F6:BF:03:FD:E8:7C:4B:2F:9B	2006-07-04T17:20Z	2031-07-04T17:20Z

	issuer name	fingerprint (SHA-1)	valid from	valid to
24	certSIGN ROOT CA G2	26:F9:93:B4:ED:3D:28:27:B0:B9:4B:A7:E9:15:1D:A3:8D:92:E5:32	2017-02-06T09:27Z	2042-02-06T09:27Z
25	Certum CA	62:52:DC:40:F7:11:43:A2:2F:DE:9E:F7:34:8E:06:42:51:B1:81:18	2002-06-11T10:46Z	2027-06-11T10:46Z
26	Certum EC-384 CA	F3:3E:78:3C:AC:DF:F4:A2:CC:AC:67:55:69:56:D7:E5:16:3C:E1:ED	2018-03-26T07:24Z	2043-03-26T07:24Z
27	Certum Trusted Network CA	07:E0:32:E0:20:B7:2C:3F:19:2F:06:28:A2:59:3A:19:A7:0F:06:9E	2008-10-22T12:07Z	2029-12-31T12:07Z
28	Certum Trusted Network CA 2	D3:DD:48:3E:2B:BF:4C:05:E8:AF:10:F5:FA:76:26:CF:D3:DC:30:92	2011-10-06T08:39Z	2046-10-06T08:39Z
29	Certum Trusted Root CA	C8:83:44:C0:18:AE:9F:CC:F1:87:B7:8F:22:D1:C5:D7:45:84:BA:E5	2018-03-16T12:10Z	2043-03-16T12:10Z
30	CFCA EV ROOT	E2:B8:29:4B:55:84:AB:6B:58:C2:90:46:6C:AC:3F:B8:39:8F:84:83	2012-08-08T03:07Z	2029-12-31T03:07Z
31	Chambers of Commerce Root	6E:3A:55:A4:19:0C:19:5C:93:84:3C:C0:DB:72:2E:31:30:61:F0:B1	2003-09-30T16:13Z	2037-09-30T16:13Z
32	Chambers of Commerce Root - 2008	78:6A:74:AC:76:AB:14:7F:9C:6A:30:50:BA:9E:A8:7E:FE:9A:CE:3C	2008-08-01T12:29Z	2038-07-31T12:29Z
33	COMODO Certification Authority	66:31:BF:9E:F7:4F:9E:B6:C9:D5:A6:0C:BA:6A:BE:D1:F7:BD:EF:7B	2006-12-01T00:00Z	2029-12-31T23:59Z
34	COMODO ECC Certification Authority	9F:74:4E:9F:2B:4D:BA:EC:0F:31:2C:50:B6:56:3B:8E:2D:93:C3:11	2008-03-06T00:00Z	2038-01-18T23:59Z

	issuer name	fingerprint (SHA-1)	valid from	valid to
35	COMODO RSA Certification Authority	AF:E5:D2:44:A8:D1:19:42:30:FF:47:9F:E2:F8:97:BB:CD:7A:8C:B4	2010-01-19T00:00Z	2038-01-18T23:59Z
36	DigiCert Assured ID Root CA	05:63:B8:63:0D:62:D7:5A:BB:C8:AB:1E:4B:DF:B5:A8:99:B2:4D:43	2006-11-10T00:00Z	2031-11-10T00:00Z
37	DigiCert Assured ID Root G2	A1:4B:48:D9:43:EE:0A:0E:40:90:4F:3C:E0:A4:C0:91:93:51:5D:3F	2013-08-01T12:00Z	2038-01-15T12:00Z
38	DigiCert Assured ID Root G3	F5:17:A2:4F:9A:48:C6:C9:F8:A2:00:26:9F:DC:0F:48:2C:AB:30:89	2013-08-01T12:00Z	2038-01-15T12:00Z
39	DigiCert Global Root CA	A8:98:5D:3A:65:E5:E5:C4:B2:D7:D6:6D:40:C6:DD:2F:B1:9C:54:36	2006-11-10T00:00Z	2031-11-10T00:00Z
40	DigiCert Global Root G2	DF:3C:24:F9:BF:D6:66:76:1B:26:80:73:FE:06:D1:CC:8D:4F:82:A4	2013-08-01T12:00Z	2038-01-15T12:00Z
41	DigiCert Global Root G3	7E:04:DE:89:6A:3E:66:6D:00:E6:87:D3:3F:FA:D9:3B:E8:3D:34:9E	2013-08-01T12:00Z	2038-01-15T12:00Z
42	DigiCert High Assurance EV Root CA	5F:B7:EE:06:33:E2:59:DB:AD:0C:4C:9A:E6:D3:8F:1A:61:C7:DC:25	2006-11-10T00:00Z	2031-11-10T00:00Z
43	DigiCert Trusted Root G4	DD:FB:16:CD:49:31:C9:73:A2:03:7D:3F:C8:3A:4D:7D:77:5D:05:E4	2013-08-01T12:00Z	2038-01-15T12:00Z
44	D-TRUST Root Class 3 CA 2 2009	58:E8:AB:B0:36:15:33:FB:80:F7:9B:1B:6D:29:D3:FF:8D:5F:00:F0	2009-11-05T08:35Z	2029-11-05T08:35Z
45	D-TRUST Root Class 3 CA 2 EV 2009	96:C9:1B:0B:95:B4:10:98:42:FA:D0:D8:22:79:FE:60:FA:B9:16:83	2009-11-05T08:50Z	2029-11-05T08:50Z

	issuer name	fingerprint (SHA-1)	valid from	valid to
46	emSign ECC Root CA - C3	B6:AF:43:C2:9B:81:53:7D:F6:EF:6B:C3:1F:1F:60:15:0C:EE:48:66	2018-02-18T18:30Z	2043-02-18T18:30Z
47	emSign ECC Root CA - G3	30:43:FA:4F:F2:57:DC:A0:C3:80:EE:2E:58:EA:78:B2:3F:E6:BB:C1	2018-02-18T18:30Z	2043-02-18T18:30Z
48	emSign Root CA - C1	E7:2E:F1:DF:FC:B2:09:28:CF:5D:D4:D5:67:37:B1:51:CB:86:4F:01	2018-02-18T18:30Z	2043-02-18T18:30Z
49	emSign Root CA - G1	8A:C7:AD:8F:73:AC:4E:C1:B5:75:4D:A5:40:F4:FC:CF:7C:B5:8E:8C	2018-02-18T18:30Z	2043-02-18T18:30Z
50	Entrust Root Certification Authority	B3:1E:B1:B7:40:E3:6C:84:02:DA:DC:37:D4:4D:F5:D4:67:49:52:F9	2006-11-27T20:23Z	2026-11-27T20:53Z
51	Entrust Root Certification Authority - EC1	20:D8:06:40:DF:9B:25:F5:12:25:3A:11:EA:F7:59:8A:EB:14:B5:47	2012-12-18T15:25Z	2037-12-18T15:55Z
52	Entrust Root Certification Authority - G2	8C:F4:27:FD:79:0C:3A:D1:66:06:8D:E8:1E:57:EF:BB:93:22:72:D4	2009-07-07T17:25Z	2030-12-07T17:55Z
53	Entrust Root Certification Authority - G4	14:88:4E:86:26:37:B0:26:AF:59:62:5C:40:77:EC:35:29:BA:96:01	2015-05-27T11:11Z	2037-12-27T11:41Z
54	Entrust.net Certification Authority (2048)	50:30:06:09:1D:97:D4:F5:AE:39:F7:CB:E7:92:7D:7D:65:2D:34:31	1999-12-24T17:50Z	2029-07-24T14:15Z
55	ePKI Root Certification Authority	67:65:0D:F1:7E:8E:7E:5B:82:40:A4:F4:56:4B:CF:E2:3D:69:C6:F0	2004-12-20T02:31Z	2034-12-20T02:31Z
56	e-Szigno Root CA 2017	89:D4:83:03:4F:9E:9A:48:80:5F:72:37:D4:A9:A6:EF:CB:7C:1F:D1	2017-08-22T12:07Z	2042-08-22T12:07Z

	issuer name	fingerprint (SHA-1)	valid from	valid to
57	E-Tugra Certification Authority	51:C6:E7:08:49:06:6E:F3:92:D4:5C:A0:0D:6D:A3:62:8F:C3:52:39	2013-03-05T12:09Z	2023-03-03T12:09Z
58	GDCA TrustAUTH R5 ROOT	0F:36:38:5B:81:1A:25:C3:9B:31:4E:83:CA:E9:34:66:70:CC:74:B4	2014-11-26T05:13Z	2040-12-31T15:59Z
59	GeoTrust Global CA	DE:28:F4:A4:FF:E5:B9:2F:A3:C5:03:D1:A3:49:A7:F9:96:2A:82:12	2002-05-21T04:00Z	2022-05-21T04:00Z
60	GeoTrust Primary Certification Authority	32:3C:11:8E:1B:F7:B8:B6:52:54:E2:E2:10:0D:D6:02:90:37:F0:96	2006-11-27T00:00Z	2036-07-16T23:59Z
61	GeoTrust Primary Certification Authority - G2	8D:17:84:D5:37:F3:03:7D:EC:70:FE:57:8B:51:9A:99:E6:10:D7:B0	2007-11-05T00:00Z	2038-01-18T23:59Z
62	GeoTrust Primary Certification Authority - G3	03:9E:ED:B8:0B:E7:A0:3C:69:53:89:3B:20:D2:D9:32:3A:4C:2A:FD	2008-04-02T00:00Z	2037-12-01T23:59Z
63	GeoTrust Universal CA	E6:21:F3:35:43:79:05:9A:4B:68:30:9D:8A:2F:74:22:15:87:EC:79	2004-03-04T05:00Z	2029-03-04T05:00Z
64	Global Chambersign Root - 2008	4A:BD:EE:EC:95:0D:35:9C:89:AE:C7:52:A1:2C:5B:29:F6:D6:AA:0C	2008-08-01T12:31Z	2038-07-31T12:31Z
65	GlobalSign	D6:9B:56:11:48:F0:1C:77:C5:45:78:C1:09:26:DF:5B:85:69:76:AD	2009-03-18T10:00Z	2029-03-18T10:00Z
66	GlobalSign	1F:24:C6:30:CD:A4:18:EF:20:69:FF:AD:4F:DD:5F:46:3A:1B:69:AA	2012-11-13T00:00Z	2038-01-19T03:14Z
67	GlobalSign	80:94:64:0E:B5:A7:A1:CA:11:9C:1F:DD:D5:9F:81:02:63:A7:FB:D1	2014-12-10T00:00Z	2034-12-10T00:00Z

	issuer name	fingerprint (SHA-1)	valid from	valid to
68	GlobalSign	69:69:56:2E:40:80:F4:24:A1:E7:19:9F:14:BA:F3:EE:58:AB:6A:BB	2012-11-13T00:00Z	2038-01-19T03:14Z
69	GlobalSign Root CA	B1:BC:96:8B:D4:F4:9D:62:2A:A8:9A:81:F2:15:01:52:A4:1D:82:9C	1998-09-01T12:00Z	2028-01-28T12:00Z
70	GlobalSign Root E46	39:B4:6C:D5:FE:80:06:EB:E2:2F:4A:BB:08:33:A0:AF:DB:B9:DD:84	2019-03-20T00:00Z	2046-03-20T00:00Z
71	GlobalSign Root R46	53:A2:B0:4B:CA:6B:D6:45:E6:39:8A:8E:C4:0D:D2:BF:77:C3:A2:90	2019-03-20T00:00Z	2046-03-20T00:00Z
72	GLOBALTRUST 2020	D0:67:C1:13:51:01:0C:AA:D0:C7:6A:65:37:31:16:26:4F:53:71:A2	2020-02-10T00:00Z	2040-06-10T00:00Z
73	Go Daddy Class 2 Certification Authority	27:96:BA:E6:3F:18:01:E2:77:26:1B:A0:D7:77:70:02:8F:20:EE:E4	2004-06-29T17:06Z	2034-06-29T17:06Z
74	Go Daddy Root Certificate Authority - G2	47:BE:AB:C9:22:EA:E8:0E:78:78:34:62:A7:9F:45:C2:54:FD:E6:8B	2009-09-01T00:00Z	2037-12-31T23:59Z
75	GTS Root R1	E5:8C:1C:C4:91:3B:38:63:4B:E9:10:6E:E3:AD:8E:6B:9D:D9:81:4A	2016-06-22T00:00Z	2036-06-22T00:00Z
76	GTS Root R2	9A:44:49:76:32:DB:DE:FA:D0:BC:FB:5A:7B:17:BD:9E:56:09:24:94	2016-06-22T00:00Z	2036-06-22T00:00Z
77	GTS Root R3	ED:E5:71:80:2B:C8:92:B9:5B:83:3C:D2:32:68:3F:09:CD:A0:1E:46	2016-06-22T00:00Z	2036-06-22T00:00Z
78	GTS Root R4	77:D3:03:67:B5:E0:0C:15:F6:0C:38:61:DF:7C:E1:3B:92:46:4D:47	2016-06-22T00:00Z	2036-06-22T00:00Z

	issuer name	fingerprint (SHA-1)	valid from	valid to
79	Hellenic Academic and Research Institutions ECC RootCA 2015	9F:F1:71:8D:92:D5:9A:F3:7D:74:97:B4:BC:6F:84:68:0B:BA:B6:66	2015-07-07T10:37Z	2040-06-30T10:37Z
80	Hellenic Academic and Research Institutions RootCA 2015	01:0C:06:95:A6:98:19:14:FF:BF:5F:C6:B0:B6:95:EA:29:E9:12:A6	2015-07-07T10:11Z	2040-06-30T10:11Z
81	Hongkong Post Root CA 1	D6:DA:A8:20:8D:09:D2:15:4D:24:B5:2F:CB:34:6E:B2:58:B2:8A:58	2003-05-15T05:13Z	2023-05-15T04:52Z
82	Hongkong Post Root CA 3	58:A2:D0:EC:20:52:81:5B:C1:F3:F8:64:02:24:4E:C2:8E:02:4B:02	2017-06-03T02:29Z	2042-06-03T02:29Z
83	IdenTrust Commercial Root CA 1	DF:71:7E:AA:4A:D9:4E:C9:55:84:99:60:2D:48:DE:5F:BC:F0:3A:25	2014-01-16T18:12Z	2034-01-16T18:12Z
84	IdenTrust Public Sector Root CA 1	BA:29:41:60:77:98:3F:F4:F3:EF:F2:31:05:3B:2E:EA:6D:4D:45:FD	2014-01-16T17:53Z	2034-01-16T17:53Z
85	ISRG Root X1	CA:BD:2A:79:A1:07:6A:31:F2:1D:25:36:35:CB:03:9D:43:29:A5:E8	2015-06-04T11:04Z	2035-06-04T11:04Z
86	Izenpe.com	2F:78:3D:25:52:18:A7:4A:65:39:71:B5:2C:A2:9C:45:15:6F:E9:19	2007-12-13T13:08Z	2037-12-13T08:27Z
87	LuxTrust Global Root	C9:3C:34:EA:90:D9:13:0C:0F:03:00:4B:98:BD:8B:35:70:91:56:11	2011-03-17T09:51Z	2021-03-17T09:51Z
88	LuxTrust Global Root 2	1E:0E:56:19:0A:D1:8B:25:98:B2:04:44:FF:66:8A:04:17:99:5F:3F	2015-03-05T13:21Z	2035-03-05T13:21Z
89	Microsoft ECC Root Certificate Authority 2017	99:9A:64:C3:7F:F4:7D:9F:AB:95:F1:47:69:89:14:60:EE:C4:C3:C5	2019-12-18T23:06Z	2042-07-18T23:16Z

	issuer name	fingerprint (SHA-1)	valid from	valid to
90	Microsoft RSA Root Certificate Authority 2017	73:A5:E6:4A:3B:FF:83:16:FF:0E:DC:CC:61:8A:90:6E:4E:AE:4D:74	2019-12-18T22:51Z	2042-07-18T23:00Z
91	NAVER Global Root Certification Authority	8F:6B:F2:A9:27:4A:DA:14:A0:C4:F4:8E:61:27:F9:C0:1E:78:5D:D1	2017-08-18T08:58Z	2037-08-18T23:59Z
92	NetLock Arany (Class Gold) F? tanúsítvány	06:08:3F:59:3F:15:A1:04:A0:69:A4:6B:A9:03:D0:06:B7:97:09:91	2008-12-11T15:08Z	2028-12-06T15:08Z
93	OISTE WISeKey Global Root GB CA	0F:F9:40:76:18:D3:D7:6A:4B:98:F0:A8:35:9E:0C:FD:27:AC:CC:ED	2014-12-01T15:00Z	2039-12-01T15:10Z
94	OISTE WISeKey Global Root GC CA	E0:11:84:5E:34:DE:BE:88:81:B9:9C:F6:16:26:D1:96:1F:C3:B9:31	2017-05-09T09:48Z	2042-05-09T09:58Z
95	QuoVadis Root CA 1 G3	1B:8E:EA:57:96:29:1A:C9:39:EA:B8:0A:81:1A:73:73:C0:93:79:67	2012-01-12T17:27Z	2042-01-12T17:27Z
96	QuoVadis Root CA 2	CA:3A:FB:CF:12:40:36:4B:44:B2:16:20:88:80:48:39:19:93:7C:F7	2006-11-24T18:27Z	2031-11-24T18:23Z
97	QuoVadis Root CA 2 G3	09:3C:61:F3:8B:8B:DC:7D:55:DF:75:38:02:05:00:E1:25:F5:C8:36	2012-01-12T18:59Z	2042-01-12T18:59Z
98	QuoVadis Root CA 3	1F:49:14:F7:D8:74:95:1D:DD:AE:02:C0:BE:FD:3A:2D:82:75:51:85	2006-11-24T19:11Z	2031-11-24T19:06Z
99	QuoVadis Root CA 3 G3	48:12:BD:92:3C:A8:C4:39:06:E7:30:6D:27:96:E6:A4:CF:22:2E:7D	2012-01-12T20:26Z	2042-01-12T20:26Z
100	QuoVadis Root Certification Authority	DE:3F:40:BD:50:93:D3:9B:6C:60:F6:DA:BC:07:62:01:00:89:76:C9	2001-03-19T18:33Z	2021-03-17T18:33Z

	issuer name	fingerprint (SHA-1)	valid from	valid to
101	Secure Global CA	3A:44:73:5A:E5:81:90:1F:24:86:61:46:1E:3B:9C:C4:5F:F5:3A:1B	2006-11-07T19:42Z	2029-12-31T19:52Z
102	SecureSign RootCA11	3B:C4:9F:48:F8:F3:73:A0:9C:1E:BD:F8:5B:B1:C3:65:C7:D8:11:B3	2009-04-08T04:56Z	2029-04-08T04:56Z
103	SecureTrust CA	87:82:C6:C3:04:35:3B:CF:D2:96:92:D2:59:3E:7D:44:D9:34:FF:11	2006-11-07T19:31Z	2029-12-31T19:40Z
104	Security Communication RootCA1	36:B1:2B:49:F9:81:9E:D7:4C:9E:BC:38:0F:C6:56:8F:5D:AC:B2:F7	2003-09-30T04:20Z	2023-09-30T04:20Z
105	Security Communication RootCA2	5F:3B:8C:F2:F8:10:B3:7D:78:B4:CE:EC:19:19:C3:73:34:B9:C7:74	2009-05-29T05:00Z	2029-05-29T05:00Z
106	SSL.com EV Root Certification Authority ECC	4C:DD:51:A3:D1:F5:20:32:14:B0:C6:C5:32:23:03:91:C7:46:42:6D	2016-02-12T18:15Z	2041-02-12T18:15Z
107	SSL.com EV Root Certification Authority RSA R2	74:3A:F0:52:9B:D0:32:A0:F4:4A:83:CD:D4:BA:A9:7B:7C:2E:C4:9A	2017-05-31T18:14Z	2042-05-30T18:14Z
108	SSL.com Root Certification Authority ECC	C3:19:7C:39:24:E6:54:AF:1B:C4:AB:20:95:7A:E2:C3:0E:13:02:6A	2016-02-12T18:14Z	2041-02-12T18:14Z
109	SSL.com Root Certification Authority RSA	B7:AB:33:08:D1:EA:44:77:BA:14:80:12:5A:6F:BD:A9:36:49:0C:BB	2016-02-12T17:39Z	2041-02-12T17:39Z
110	Starfield Class 2 Certification Authority	AD:7E:1C:28:B0:64:EF:8F:60:03:40:20:14:C3:D0:E3:37:0E:B5:8A	2004-06-29T17:39Z	2034-06-29T17:39Z
111	Starfield Root Certificate Authority - G2	B5:1C:06:7C:EE:2B:0C:3D:F8:55:AB:2D:92:F4:FE:39:D4:E7:0F:0E	2009-09-01T00:00Z	2037-12-31T23:59Z

	issuer name	fingerprint (SHA-1)	valid from	valid to
112	Starfield Services Root Certificate Authority - G2	92:5A:8F:8D:2C:6D:04:E0:66:5F:59:6A:FF:22:D8:63:E8:25:6F:3F	2009-09-01T00:00Z	2037-12-31T23:59Z
113	SwissSign Gold CA - G2	D8:C5:38:8A:B7:30:1B:1B:6E:D4:7A:E6:45:25:3A:6F:9F:1A:27:61	2006-10-25T08:30Z	2036-10-25T08:30Z
114	SwissSign Platinum CA - G2	56:E0:FA:C0:3B:8F:18:23:55:18:E5:D3:11:CA:E8:C2:43:31:AB:66	2006-10-25T08:36Z	2036-10-25T08:36Z
115	SwissSign Silver CA - G2	9B:AA:E5:9F:56:EE:21:CB:43:5A:BE:25:93:DF:A7:F0:40:D1:1D:CB	2006-10-25T08:32Z	2036-10-25T08:32Z
116	SZAFIR ROOT CA2	E2:52:FA:95:3F:ED:DB:24:60:BD:6E:28:F3:9C:CC:CF:5E:B3:3F:DE	2015-10-19T07:43Z	2035-10-19T07:43Z
117	TeliaSonera Root CA v1	43:13:BB:96:F1:D5:86:9B:C1:4E:6A:92:F6:CF:F6:34:69:87:82:37	2007-10-18T12:00Z	2032-10-18T12:00Z
118	thawte Primary Root CA	91:C6:D6:EE:3E:8A:C8:63:84:E5:48:C2:99:29:5C:75:6C:81:7B:81	2006-11-17T00:00Z	2036-07-16T23:59Z
119	thawte Primary Root CA - G2	AA:DB:BC:22:23:8F:C4:01:A1:27:BB:38:DD:F4:1D:DB:08:9E:F0:12	2007-11-05T00:00Z	2038-01-18T23:59Z
120	thawte Primary Root CA - G3	F1:8B:53:8D:1B:E9:03:B6:A6:F0:56:43:5B:17:15:89:CA:F3:6B:F2	2008-04-02T00:00Z	2037-12-01T23:59Z
121	Trustwave Global Certification Authority	2F:8F:36:4F:E1:58:97:44:21:59:87:A5:2A:9A:D0:69:95:26:7F:B5	2017-08-23T19:34Z	2042-08-23T19:34Z
122	Trustwave Global ECC P256 Certification Authority	B4:90:82:DD:45:0C:BE:8B:5B:B1:66:D3:E2:A4:08:26:CD:ED:42:CF	2017-08-23T19:35Z	2042-08-23T19:35Z

	issuer name	fingerprint (SHA-1)	valid from	valid to
123	Trustwave Global ECC P384 Certification Authority	E7:F3:A3:C8:CF:6F:C3:04:2E:6D:0E:67:32:C5:9E:68:95:0D:5E:D2	2017-08-23T19:36Z	2042-08-23T19:36Z
124	T-TeleSec GlobalRoot Class 2	59:0D:2D:7D:88:4F:40:2E:61:7E:A5:62:32:17:65:CF:17:D8:94:E9	2008-10-01T10:40Z	2033-10-01T23:59Z
125	T-TeleSec GlobalRoot Class 3	55:A6:72:3E:CB:F2:EC:CD:C3:23:74:70:19:9D:2A:BE:11:E3:81:D1	2008-10-01T10:29Z	2033-10-01T23:59Z
126	TUBITAK Kamu SM SSL Kok Sertifikasi - Surum 1	31:43:64:9B:EC:CE:27:EC:ED:3A:3F:0B:8F:0D:E4:E8:91:DD:EE:CA	2013-11-25T08:25Z	2043-10-25T08:25Z
127	TWCA Global Root CA	9C:BB:48:53:F6:A4:F6:D3:52:A4:E8:32:52:55:60:13:F5:AD:AF:65	2012-06-27T06:28Z	2030-12-31T15:59Z
128	TWCA Root Certification Authority	CF:9E:87:6D:D3:EB:FC:42:26:97:A3:B5:A3:7A:A0:76:A9:06:23:48	2008-08-28T07:24Z	2030-12-31T15:59Z
129	UCA Extended Validation Root	A3:A1:B0:6F:24:61:23:4A:E3:36:A5:C2:37:FC:A6:FF:DD:F0:D7:3A	2015-03-13T00:00Z	2038-12-31T00:00Z
130	UCA Global G2 Root	28:F9:78:16:19:7A:FF:18:25:18:AA:44:FE:C1:A0:CE:5C:B6:4C:8A	2016-03-11T00:00Z	2040-12-31T00:00Z
131	USERTrust ECC Certification Authority	D1:CB:CA:5D:B2:D5:2A:7F:69:3B:67:4D:E5:F0:5A:1D:0C:95:7D:F0	2010-02-01T00:00Z	2038-01-18T23:59Z
132	USERTrust RSA Certification Authority	2B:8F:1B:57:33:0D:BB:A2:D0:7A:6C:51:F7:0E:E9:0D:DA:B9:AD:8E	2010-02-01T00:00Z	2038-01-18T23:59Z
133	UTN-USERFirst-Object	E1:2D:FB:4B:41:D7:D9:C3:2B:30:51:4B:AC:1D:81:D8:38:5E:2D:46	1999-07-09T18:31Z	2019-07-09T18:40Z

	issuer name	fingerprint (SHA-1)	valid from	valid to
134	VeriSign Class 3 Public Primary Certification Authority - G3	13:2D:0D:45:53:4B:69:97:CD:B2:D5:C3:39:E2:55:76:60:9B:5C:C6	1999-10-01T00:00Z	2036-07-16T23:59Z
135	VeriSign Class 3 Public Primary Certification Authority - G4	22:D5:D8:DF:8F:02:31:D1:8D:F7:9D:B7:CF:8A:2D:64:C9:3F:6C:3A	2007-11-05T00:00Z	2038-01-18T23:59Z
136	VeriSign Class 3 Public Primary Certification Authority - G5	4E:B6:D5:78:49:9B:1C:CF:5F:58:1E:AD:56:BE:3D:9B:67:44:A5:E5	2006-11-08T00:00Z	2036-07-16T23:59Z
137	VeriSign Universal Root Certification Authority	36:79:CA:35:66:87:72:30:4D:30:A5:FB:87:3B:0F:A7:7B:B7:0D:54	2008-04-02T00:00Z	2037-12-01T23:59Z
138	XRamp Global Certification Authority	B8:01:86:D1:EB:9C:86:A5:41:04:CF:30:54:F3:4C:52:B7:E5:58:C6	2004-11-01T17:14Z	2035-01-01T05:37Z

A list changes in time, depending on Java version used

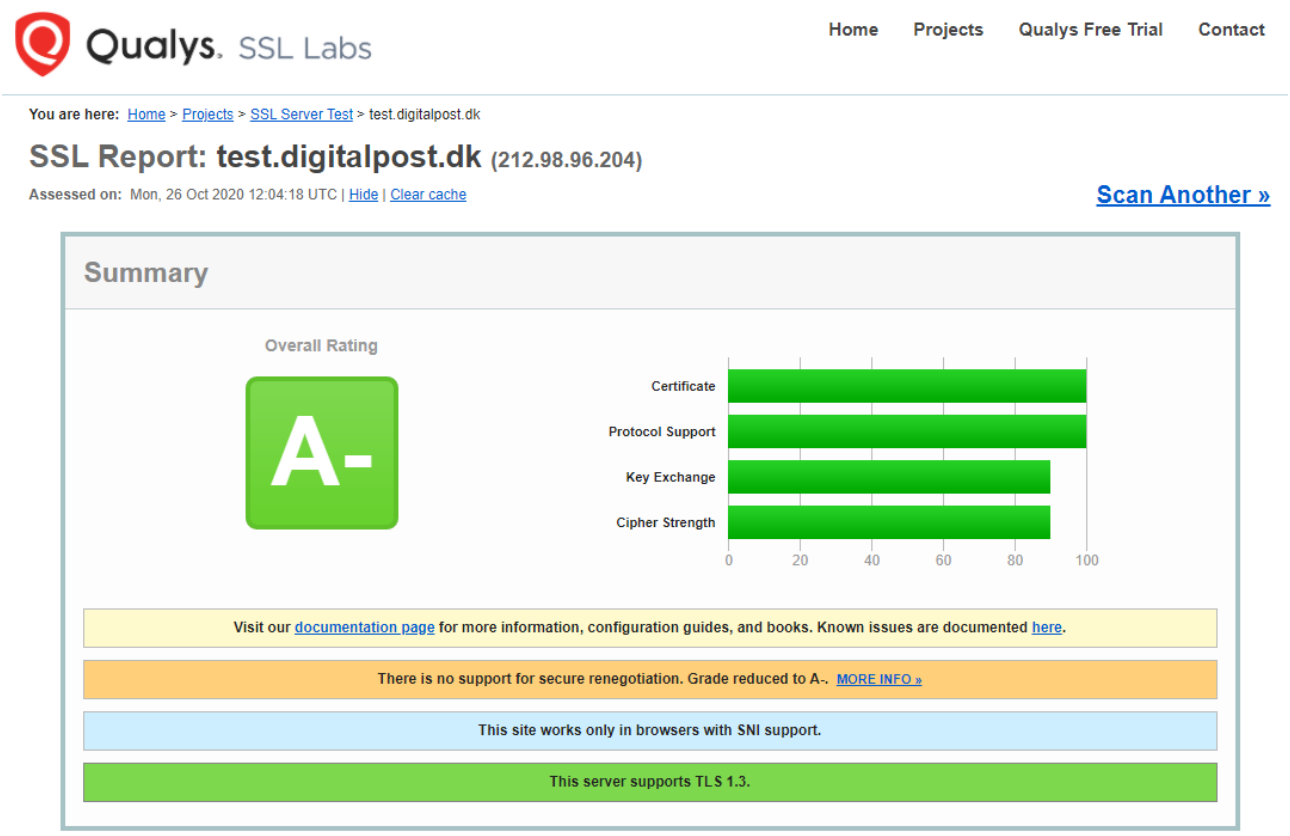
16.1.2 Allowed ciphers

The following Ciphers are allowed when trying to connect to application endpoints, the list of allowed ciphers will change over time when new secure ciphers are supported, and older ciphers are getting insecure:

TLS1.3-AES256-GCM-SHA384
TLS1.2-ECDHE-RSA-AES256-GCM-SHA384
TLS1.2-ECDHE-RSA-AES128-GCM-SHA256
TLS1.3-AES128-GCM-SHA256

16.1.3 Rating

There are many factors that affect certificate quality. To simplify this assessment process one can use so called rating tools. We are using SSL Labs rating, for example: https://www.ssllabs.com/ssltest/analyze.html?d=test.digitalpost.dk:



Rating is based on multiple factors (see https://github.com/ssllabs/research/wiki/SSL-Server-Rating-Guide), and as a rule of thumb:

A+, **A, A-** ratings are mandatory

16.1.4 Troubleshooting

Full certificate chain

We have observed that one of the functionalities that is frequently incorrectly configured is a **full certificate chain** being reported by server - this drops the rating to below A.

16.1.5 Ensuring the authenticity of Digital Posts through the OCES certificate

HTTP calls with mutual SSL

When systems are calling Digital Post through mutual SSL, Digital Post ensures the validity and identity of the caller by performing a series of checks as described in “Mutual SSL authentication using API key”.

Similarly, receiver systems are expected to verify the identity of Digital Post, to ensure that the services are not misused by bad actors. One way of doing this is by pinning the certificate of Digital Post. Meaning that your receiver system *only* allows calls where the caller authenticate using Digital Post’s OCES certificate.

Finding the certificate

To ensure that integrators can do certificate pinning and encryption/signature validation, Digital Post provides the public part of its OCES certificate.

The certificate can be found on https://digitaliser.dk/digital-post/vejledninger/oces-certifikater. Please note that different certificates are used in test and production.

16.1.6 Point of attention regarding certificates

In relation to the initial testing of the Digital Post test environment, there has been issues regarding the test users Trust Chain in their main certificates.
The following documentation and description are intended to indicate points of attention in relation to obtaining an intact Trust Chain and thereby a functional test certificate and ultimately access to DP's administration portal.

Two types of certificates

It is important that you **always use a test certificate (OCES) when accessing the test environment** - and a production certificate (OCES) when accessing the production environment.

The Trust Chain for test- and production certificates are not identical and therefore using a production certificate may cause errors in the Trust Chain, when using it on the test environment.

Furthermore, production certificates can contain confidential data. By default, Netcompany does not have access to the test users certificate chains, so it is only in cases of issues that it may become necessary.

Certificates must be traceable

If an error in the Trust Chain occurs, it is the test user's responsibility to be able to trace the certificate and thereby the associated certificate chain (part of the Trust Chain). When uploading a test certificate, the test environments frontend (Administrativ Access) does not check if the certificate chain is correct, therefore it is very important that the whole chain of the certificate can be traced back as, if an error occurs.

Furthermore, it is also very important, to be able to access the entire chain and not just the end of the certificate.

Check validity of a OCES certificate for test and production

In general

The description of OCES certificates can be found in this document (in Danish) https://digitaliseringskataloget.dk/files/integration-files/020920201531/Kom%20godt%20i%20gang%20-%20certifikater.pdf (Have a look at page 12)

Test certificates

For a test certificate, the OCES certificate must contain the primary (certificate #1) and secondary issuer (certificate #2)

The first certificate must contain the Issuer (CN) e.g:
Issuer: CN=**TRUST2408 Systemtest VII Primary CA**, O=TRUST2408, C=DK

The second certificate must contain the Issuer (CN) e.g:
Issuer: CN=**TRUST2408 Systemtest XXII CA**, O=TRUST2408, C=DK

Production Certificate

The first certificate must contain the Issuer (CN) e.g:
TRUST2408 OCES Primary CA

Tool for checking validity of certificate

In order to check if a certificate (.pem, p12, .cer, .crt) is for production or test, different tools can be used. This example is a Windows tool:

```
certutil -dump <path to cert>
```

https://superuser.com/questions/580697/how-do-i-view-the-contents-of-a-pfx-file-on-windows

This will dump the content of the certificate and you need to check that the information from the dump is correct.

A dump from a valid .p12 (pkcs12) test certificate:

```
================ Certificate 0 ================
================ Begin Nesting Level 1 ================
Element 0:
Serial Number: 4bea6e94
Issuer: CN=TRUST2408 Systemtest VII Primary CA, O=TRUST2408, C=DK
 NotBefore: 12-05-2010 09:32
 NotAfter: 12-01-2037 10:02
Subject: CN=TRUST2408 Systemtest VII Primary CA, O=TRUST2408, C=DK
Signature matches Public Key
Root Certificate: Subject matches Issuer
Cert Hash(sha1): d6b1f3e9319f68d36f1c71c48e47468130543bce
----------------  End Nesting Level 1  ----------------
No key provider information
Cannot find the certificate and private key for decryption.

================ Certificate 1 ================
================ Begin Nesting Level 1 ================
Element 1:
Serial Number: 58187e74
Issuer: CN=TRUST2408 Systemtest VII Primary CA, O=TRUST2408, C=DK
 NotBefore: 04-07-2017 07:18
 NotAfter: 04-07-2032 07:48
Subject: CN=TRUST2408 Systemtest XXII CA, O=TRUST2408, C=DK
Non-root Certificate
Cert Hash(sha1): 896f3cdbdc384eba6b13105b2ca1654bc1b97437
----------------  End Nesting Level 1  ----------------
No key provider information
Cannot find the certificate and private key for decryption.

================ Certificate 2 ================
================ Begin Nesting Level 1 ================
Element 2:
Serial Number: 5bad3b1a
Issuer: CN=TRUST2408 Systemtest XXII CA, O=TRUST2408, C=DK
 NotBefore: 16-12-2019 15:31
 NotAfter: 16-12-2022 15:31
Subject: SERIALNUMBER=CVR:30808460-FID:94731315 + CN=TU GENEREL FOCES gyldig
(funktionscertifikat), O=NETS DANID A/S // CVR:30808460, C=DK
Non-root Certificate
Cert Hash(sha1): 21ad7d2d4280765bfe113b7dd5d62736c34e37bd
```

```
----------------  End Nesting Level 1  ----------------
  Provider = Microsoft Enhanced RSA and AES Cryptographic Provider
Encryption test passed
CertUtil: -dump command completed successfully.
```

Certificates must be unique and not reused

It is very important that you do not reuse the test certificates, because the test environment cannot distinguish reused test certificates from each other, and this may cause a system error.
Therefore, it is important that you use a unique test certificate, when creating a new system in Administrative Access.

Certificate expiration date

When creating or receiving a test certificate it is important that you keep track of when the certificate expires. Before it expires it must be renewed and uploaded again.

If it is possible for your organization, then create a test certificate that do not expire before the end of 2023, then you do not have to worry about renewing it.

16.1.7 Digital Post Service Desk

If you are in the process of integrating to Digital Post and are experiencing issues with the solution - or lacking information regarding the interfaces which is not available in this document - it is possible to create a ticket via the Servicedesk. Use the following link and provide as much information as possible:

https://digidp.atlassian.net/servicedesk/customer/portal/

16.2 SFTP server

16.2.1 Generate an SSH-key to DP

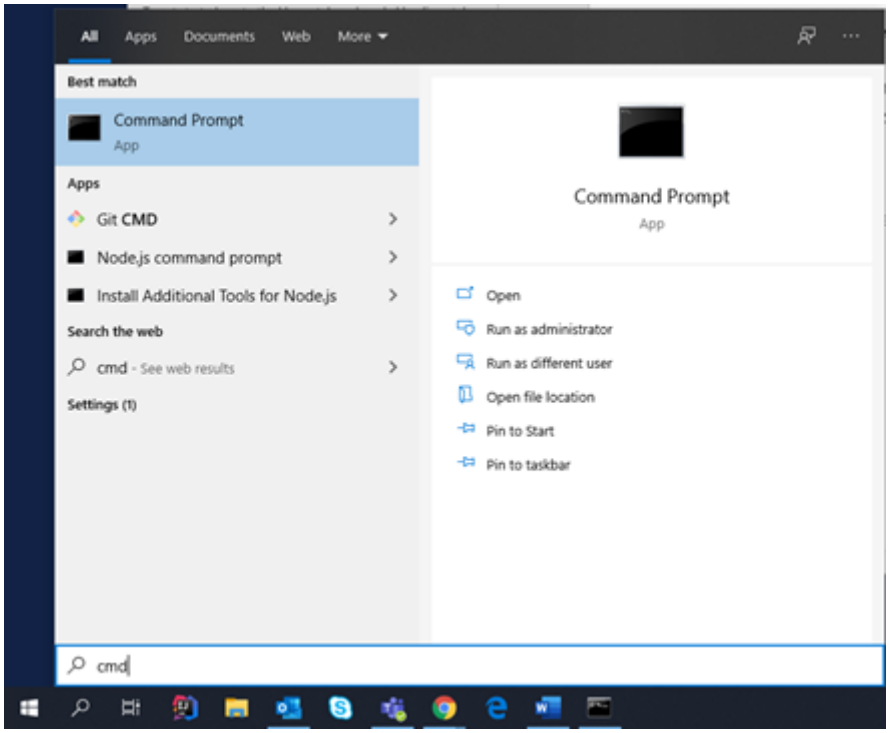
In the DP test environment SSH-key pairs are used to integrate your sender system, if you use the SFTP protocol.

In the following there are two examples of how you can make an SSH-key. Both examples will generate both a Private and a Public SSH-key. However, you should only upload the Public SSH-key id_rsa.pub and not the Private SSH-key, that are generated.

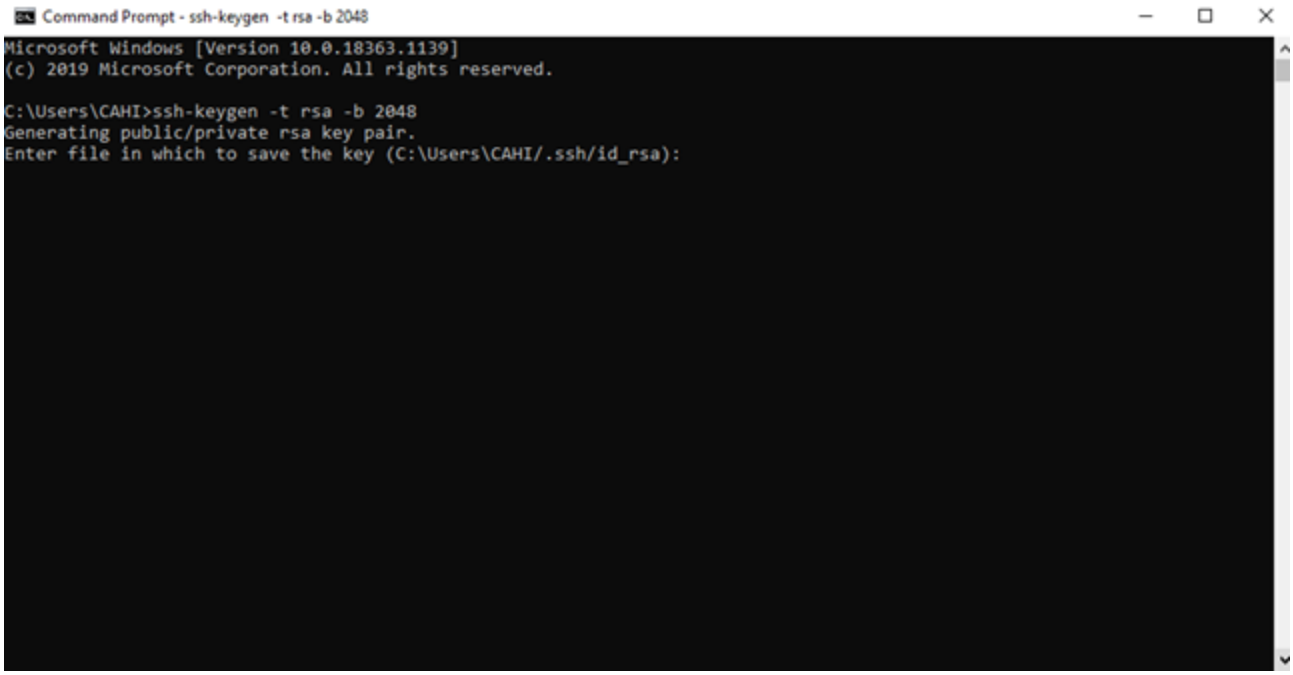
The first example is generated via Command Prompt/The Terminal and the second example is generated via program called PuTTY. Command Prompt works on Windows, Linux and MacOS and PuTTY is a Windows program.

Command Prompt:

To open Command Prompt, you must search “CMD” in the Windows search box and click on Command Prompt.



When Command Prompt is open, you must type “SSH-keygen -t rsa -b 2048” to generate an SSH-key. -t defines what type of SSH-key it should be and -b defines how many bits it is. In the DP test environment we use rsa which support 2048 bits. Once you have written it in the Command Prompt, it will ask where you want to save the file nd what it should be called, which is shown in the image below.



As you can see in the image above, Command Prompt suggest a place to save it an the name of the file. You can separate were it saves the file from the filename on backslash (\) and slash (/). Backslash specifies the file path (C:\Users\CAHI) and is both a new directory (/.ssh) and the filename (/id_rsa). Here you must press “enter”, then the file will be saved in the user’s directory (C:\Users\CAHI), in a directory it creates (/.ssh/).

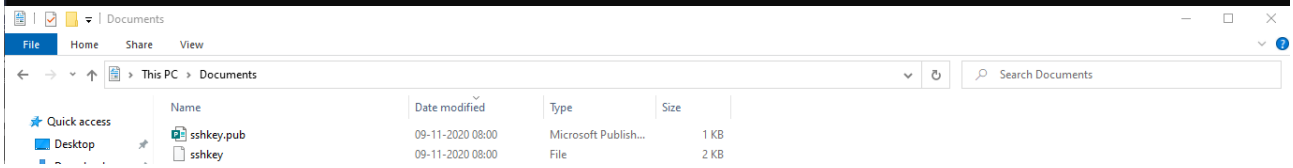
However, it can be specifies where the file is to be saved. An example of this could be “C:\Users\SAHI\Documents\sshkey”. In this example it will be saved in the Documents directory and the file will be named sshkey.

After pressing “enter” or specifying another location to save the file, the Command Prompt will ask for a password for the files. It is possible not to specify a password, however, it is recommended to enter a password. Please note that you must enter the password twice, which is shown in the image below:

```
C:\Users\CAHI>ssh-keygen -t rsa -b 2048
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\CAHI/.ssh/id_rsa): C:\Users\CAHI\Documents\sshkey
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

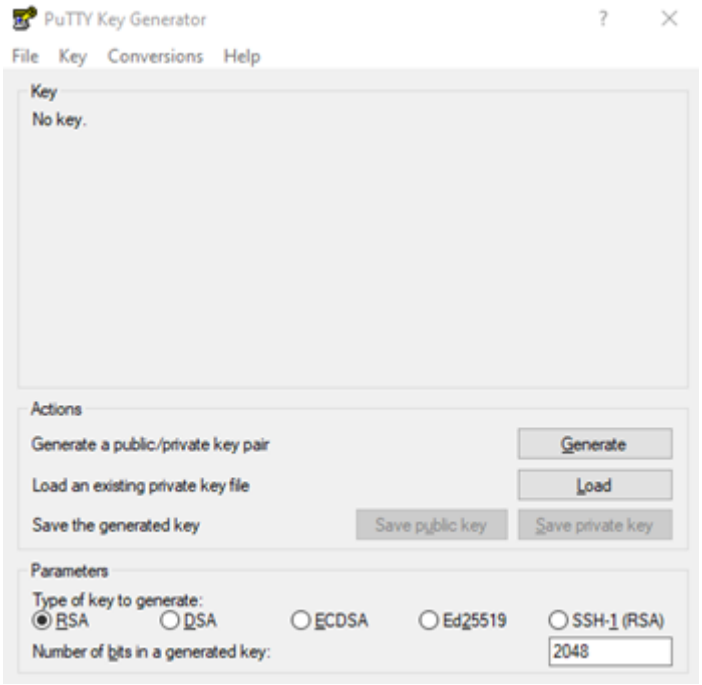
Once you have entered the password, the SSH-key will be generated and shown in the file directory, the SSH-key should look like the image below.

```
C:\Users\CAHI>ssh-keygen -t rsa -b 2048
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\CAHI/.ssh/id_rsa): C:\Users\CAHI\Documents\sshkey
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\CAHI\Documents\sshkey.
Your public key has been saved in C:\Users\CAHI\Documents\sshkey.pub.
The key fingerprint is:
SHA256:pDbSEB1mn/LkKTfmzd5s9rntthsq9JbKxQFdii7dxF0 nclan\cahi@PF1WSVDM
The key's randomart image is:
+---[RSA 2048]----+
|     ..+.       .E|
|      +.. .  + + .|
|     . . =  o = . |
|      o B .o +    |
|     . * S. o o   |
|      o * +... .  |
|         . + .o.. |
|          ..+++.oo|
|           .=*o+*=|
+----[SHA256]-----+
```



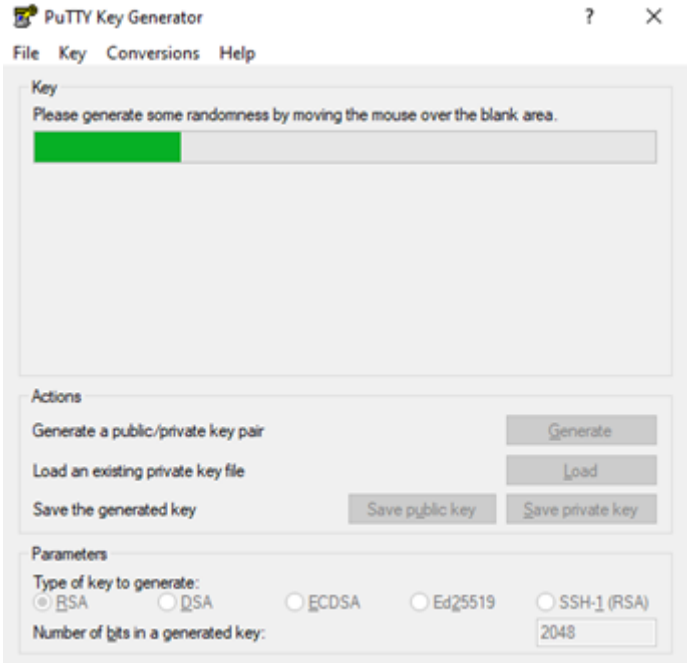
PuTTY:

Another way to generate an SSH-key is by using PuTTY. PuTTY can be downloaded from: https://www.putty.org/ and should look like the image below, when you open it.

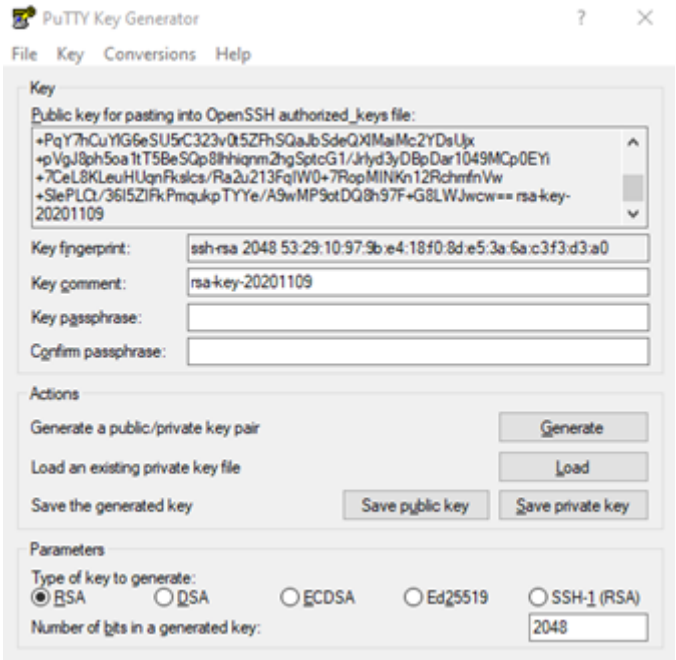


At the bottom of the PuTTY program, we can define different parameters. You must choose “rsa” and make sure the number of “bits” are 2048, which is shown in the image below.

To generate a SSH-key with these parameters, you must press “Generate”. After you have pressed “Generate” you must move the mouse over the empty field in the program. It will then use the mouse input to generate the SSH-key, which is shown in the following image:



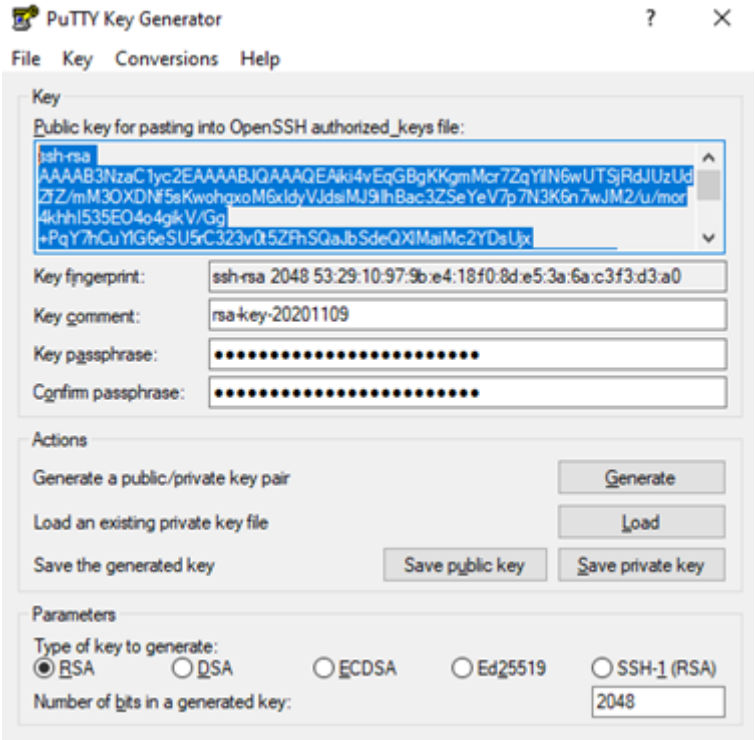
Once you have generate the SSH-key, the “save pblic key” and “save private key” will no longer be disabled and you will have the option to save the files. In addition it is possible to enter a password for the files, which is recommended.



When you have saved the files, both the Public and Private, you must open the Public SSH-key in Notepad or and other text application. Here you must change the text that is displayed in the Public SSH-key, to the SSH-key that is displayed at the top of PuTTY.

You can copy the text by highlighting it and then copy (Ctrl + C) + paste (Ctrl + V) it into the text program. This key should then replace the text in the file, so only the SSH-key from PuTTY is in the file. Remember to copy to copy all the text from PuTTY (Ctrl + A), as you can scroll through the window in PuTTY, which contains the whole key.

In the image below you can se, what it should look like, however it should just contain the SSH-key, that your program has generated.



Please note: The examples above are just two examples out of several, of how to make an SSH-key. If you are familiar with other programs, these can easily be used.

16.2.2 Access the SFTP server

To access the SFTP server the following is needed:

1. A sender system with service protocol SFTP needs to be set up through Administrative Access (AA)
    a. A valid SSH key-pair: a public key and a private key. How to generate this can be found here.
        i. **Note:** the easiest method is to use the Command Prompt, as the PuTTy option may generate the private key in a format that is too new to work.
        ii. **Note:** when creating the sender system in Administrative Access the value of IP field does not matter for non-prod environments when you are connected to the Netcompany VPN as it is whitelisted.

When the sender system has been set up the SFTP server can be accessed. You will need:

1. The SSH username from the sender system. It is called “SSH brugernavn” in AA.
2. The private key asociated with the public key you uploaded for the sender system in AA.

The easiest way to access the SFTP server is through the “links” folder in the devops/tools/kitty

1. Run WinSCP.exe
2. Create a new site with:
    a. File protocol = SFTP
    b. Host name = sftp.test.digitalpost.dk (for non-prod environments)
    c. Port number = 22
    d. User name = the “SSH brugernavn” from your sender system
    e. Password = Blank
        i. Press “Advanced” and go to SSH → Authentication
        ii. Upload the SSH private key matching the public key of your sender system. Press OK (if your private key file is not .ppk format it will not show up when choosing a private key. Use *puttygen <private-key> -o <private-key>.ppk in WSL* to convert the private key to a .ppk file)
    f. Press login

16.3 OIO OpenID Connect to Digital Post

> ⚠ This section not relevant for sender- and recipient systems as they are expected to use mutual SSL when integration with the Digital Post REST API

In-order to integrate with Digital Post, front-end clients must utilize OpenID Connect (OIDC) when accessing the Digital Post REST API. Digital Post as a OpenID Connect provider (OP) adheres to the OIO OpenID Connect profile for OP, specifically a subset of the OIO OIDC profile as outlined below.

16.3.1 OIO OIDC profile - Digital Post subset

Background

The OIO OIDC profile contains the outline for full-fledged OIDC multi-tenant support for NemLog-in, with support slated for sometime in the future. This is intended to support all OIDC needs across the public business-domain for all types of clients.

Implementing 1-1 OIO OIDC support into Digital Post is considered out-of-scope, instead Digital Post supports a subset of the OIO OIDC profile, simplified and tailored around the Digital Post use-cases and clients. Unless noted below, Digital Post OIDC adheres to the OIO OIDC profile.

Client types

Digital Post supports OIO "Native app" and "Web/JS app with a Backend" client types, as these clients are expected to be able to protect issued refresh tokens.

"Web/JS app without a Backend" (aka. refresh-token-rotation) is **unsupported**, as the security compromises to achieve a satisfactory user experience was deemed unacceptable by the Digital Post client forum.

"Service token"

The opaque "access token" and "service token" exchange outlined in the OIO OIDC profile is unsupported by Digital Post.
Digital Post provides the OIDC access token (JWT), id token (JWT) and refresh token (opaque), in order to support common OIDC frameworks utilized by clients.

Scopes

Digital Post supports multiple auto-approved scopes dependent on the view client type. These are

Scope	Description
`openid`	Encompasses access to Digital Post via JWT
`native`	Encompasses native (mobile) clients
`cpr_number`	Encompasses access to sensitive userinfo claims for citizens

Digital Post retains full control over the scopes a view client is approved for upon registration.

OIO JWT profile

The Digital Post JWT doesn't include the Assurance Level claims `aal` and `ial`.

When the Assurance Level is provided by an NSIS-compliant identity-provider, a service-provider is not allowed to repackage that information (source: NemLog-in).

The `priv` claim is omitted as privileges are administrated exclusively in Digital Post (separately from NemLog-in), and the verbose structure generates unnecessarily large values. Instead, the privileges specific to the Digital Post domain are present in the `dppriv` claim. For users granted 55 or more privileges the `dpprivref` supports requesting the full set of privileges granted a user.

Besides the OIO JWT profile claims, Digital Post includes claims specific to the Digital Post domain.

- The `authorities` claim contains the unscoped authorities (aka roles) granted an identity (`dpiid`), specific to the Digital Post domain. The authorities also contain the privilege types, in order to support simplified access evaluation for the domain.
- The `client_id` claim contains the OIO OIDC client ID.

- The `dpiid` claim contains the Digital Post identity ID, an opaque reference specific to the Digital Post domain principal.
- The `dpprivref` claim contains a URI that represents a reference to the Digital Post privileges granted an identity (`dpiid`). If this claim is present, the user is granted more claims than is supported by the header limits imposed by various user agents & backend interfaces. To avoid the header limit and excessive I/O the `dpprivref` claim is triggered for users granted 55 or more privileges. The `dppriv` claim can then be requested using the URI value ~ utilizing the `/userinfo` endpoint.
    - Example:

```
{ .. "dpprivref" : "https://test.digitalpost.dk/auth/oauth/userinfo" .. }
```

- The `dppriv` claim contains Digital Post privileges granted an identity (`dpiid`) in a compact format. For users granted <= 54 privileges the claim is included in the JWT, otherwise see `dpprivref` .
- The `scope` claim contains the scopes granted the client.
- The `sub` claim is reserved for future use by NemLog-in3, currently the `sub` claim mirrors the `dpiid` claim to adhere to OAuth 2.0 JWT profile.

16.3.2 OIDC client enrollment in Digital Post

In-order to register your client with Digital Post, OP requires the following information before a client is authorized with OP.

Information	Description
Client name	Human readable name for OIDC client
Type	OIDC client type to register. Must be `native app` or `Web/JS app with a Backend`
Client ID	Unique ID for the view client - can be human readable or opaque
Client secret	Minimum 32 characters
Redirect URL	Redirect URIs can be any valid URI, i.e. both custom URL schemes (`myapp://`) as well as HTTP/S schemes are allowed. Note that the full redirect URI must be supplied. A test-client is allowed to register multiple URLs, in-order to facilitate development & testing. Test clients are also permitted the use of HTTP-only schemes.

Information	Description
Post logout redirect URL (Digital Post)	Post logout redirect URLs used for logging a user out of Digital Post. Redirect URIs for logging out of Digital Post can be any valid URI, i.e. both custom URL schemes (`myapp://`) as well as HTTP/S schemes are allowed. Note that the full redirect URI must be supplied. A test-client is allowed to register multiple URLs, in-order to facilitate development & testing. Test clients are also permitted the use of HTTP-only schemes.
Post logout redirect URL (NemLog-in)	Post logout redirect URLs for ending a user session in NemLog-in. Redirect URIs for logging out of NemLog-in are limited to HTTP/S-only schemes. Note that the full redirect URI must be supplied. A test-client is allowed to register multiple URLs, in-order to facilitate development & testing. Test clients are also permitted the use of HTTP-only schemes.
AppSwitch return URI (only relevant for native app client types)	The return URI the view client wishes to use for the MitID AppSwitch functionality. The same URI can be used across platforms.

For details about logout, please see the section below regarding “RP-initiated Logout”.

An `openid-configuration` is available in each environment under the `.well-known` endpoint, e.g. https://test.digitalpost.dk/auth/v2/oauth/.well-known/openid-configuration.

16.3.3 RP-initiated Logout

Digital Post supports OpenID Connect RP-Initiated Logout 1.0, unless an exception is specified here.

Session termination

As the RP-initiated specification doesn’t specify how the OpenID Connect provider is expected to recognize a given session, Digital Post has chosen the access token to represent an authorized session.

In order to logout using the `end_session_endpoint` the request MUST include the ID token through the `id_token_hint` parameter, which is linked to the Digital Post session. The access token MUST be included as a `Bearer` authorization header. The access token provided can be expired. Furthermore, the request MAY include a `post_logout_redirect_uri` for redirecting the user after a successful logout of Digital Post as well as a `state` parameter if the view client wishes to maintain state as outlined in the RP-initiated specification.

Terminating sessions by providing the (expired) access token is deprecated and view clients should migrate to terminating sessions using the `id_token_hint` instead.

Logout

> ⚠ The following section, detailing separate logout logic depending on the view-client type is no longer relevant after the view client moves to the `/auth/v2` endpoint. All sessions are now terminated by Digital Post when calling the `end_session_endpoint` as outlined above.

A login in Digital Post generates two "sessions" - one in NemLog-in and one in Digital Post. To ensure that all sessions are terminated when a user logs out it is important that a logout to both providers are initiated.

Logout in NemLog-in

View clients of the type "Web/JS app with a Backend" MUST terminate the users NemLog-in session when they log out of the view client.

To logout of NemLog-in requires calling the `/auth/s9/multi-realm/logout` endpoint exposed by Digital Post. Each view client have their own realm they MUST use when logging in and out of NemLog-in through Digital Post provided via the `idp` query parameter, see the section "NemLog-in Realms" below for details.

Logout in Digital Post

View clients of the type "Native app" MUST revoke the current refresh token if a new user is enrolling their device for Digital Post.

To logout of Digital Post, i.e. revoke the refresh token and any access tokens issued using the refresh token, the `end_session_endpoint` MUST be called. The request MUST include an `id_token_hint` to ensure the issued refresh token can be identified and revoked. Furthermore, the request MAY include a `post_logout_redirect_uri` for redirecting the user after a successful logout of Digital Post as well as a `state` parameter if the view client wishes to maintain state as outlined in the RP-initiated specification.

Logout notes

Please note that while the OIDC specification outlines that a logout can be initiated by redirecting to the OP's logout endpoint, Digital Post also supports GET/POST directly to the logout endpoint as a redirect request may exceed URI length limitation imposed by common browsers.

Note also that Digital Post delegates the responsibility for prompting the user to the client, in-line with NemLog-in's current behavior and as such ignores any `ui_locales` parameters.

16.3.4 NemLog-in Realms

Each view client has their own realm they MUST use when authenticating with NemLog-in, as indicated below. The realms MUST be used during login and logout of NemLog-in. Login is handled through the authorization code grant flow and logout is done through the `/auth/s9/multi-realm/logout` endpoint.

Each view client has their own realm they MUST use when authenticating with Digital Post, as indicated below.

View client	Realm (idp)
Public view clients (borger.dk, virk)	nemlogin
mit.dk	mit-dk-nemlogin

View client	Realm (idp)
e-Boks	e-boks-nemlogin

The realm is indicated by supplying the `idp` query parameter during the authorization code grant flow or logout. It is not necessary to supply the `idp` query parameter when requesting an access token after login or during the refresh token flow.

Examples

Authorization code grant flow

For a public view client to log a user into Digital Post the authorization request would look as follows

```
GET https://test.digitalpost.dk/auth/v2/oauth/authorize?
idp=nemlogin&client_id=...
```

Termination of NemLog-in session (logout)

For a public view client to log a user out of NemLog-in the endpoint `/auth/s9/multi-realm/logout` must be used together with the `idp` query parameter

```
GET https://test.digitapost.dk/auth/s9/multi-realm/logout?idp=nemlogin&f=...
```

In this instance the `f` query parameter is the `post logout redirect URI (NemLog-in)` provided during the client enrollment and is restricted to HTTP/S scheme only as outlined above.

OneTimeUse condition ~ “sessionless” authentication with NemLog-in

Digital Post supports “sessionless” authentication with NemLog-in through the use of the OneTimeUse functionality. A sessionless login refers to a login in NemLog-in that is not considered part of the federation and as such does not support SSO to other public selfservice portals. To perform a sessionless login through Digital Post, the view client must specify a specific value for `idp`, related to the `idp` value for a regular login.

Supported OneTimeUse `idp` Values

Each view client has their own OneTimeUse, mirroring the regular realm, they MUST use when performing a OneTimeUse login through Digital Post.

View client	OneTimeUse idp value
Public view clients (borger.dk, virk)	nemloginOneTimeUse
mit.dk	nemloginMitDkOneTimeUse
e-Boks	nemloginEboksOneTimeUse

Examples

```
GET https://test.digitalpost.dk/auth/v2/oauth/authorize?
idp=nemloginOneTimeUse&client_id=...
```

16.3.5 MitID AppSwitch support

Digital Post supports MitID AppSwitch. To make use of the AppSwitch functionality, two query parameters must be supplied to Digital Post during the authorization code grant flow as follows.

Query parameter	Description
nemloginAppswitchReturnURI	The full return URI used by the MitID app to return the user back to the app which initiated the login.
nemloginAppswitchPlatform	An enum indication which platform the user is on. Must be either `iOS` or `Android`.

16.3.6 eIDAS eID Gateway

Similar to NemLog-in, Digital Post exposes a realm for authentication with the eIDAS eID Gateway. The eID realm can be used by supplying the `idp` query parameter with the value `eid`.

16.4 Connect to Digital Post: Test and Prod

Digital Post have a test and a production environment. Therefore, you need to be aware of your OCES certificates: They depend on whether you are trying to connect to the test- or prod environment. Test certificates for test environment and prod certificates for prod environment.

Note: Connecting and configuring your sender- and receiver systems for Digital Post is done via the administration portal 'Administrativ adgang'. The following steps are only from a technical perspective for preparing your systems. For more information on how to set up your systems in the administration portal look under "References".

Step	System		Protocol		Where?	DP assistance
	Sender system	Receiver system	REST	SFTP		
Get NemLog-In OCES certificate	X	X	X		NemLog-in: https://tu.nemlog-in.dk/oprettelse-og-administration-af-it-systemer/oces3-certifikater/	NA

Generate SSH keys	X			X	Own servers. See section “Generate an SSH-key to DP”	NA
Integration to Contact registry and System registry	X		X		Own system	OpenAPI
Prepare integration to DP SFTP	X			X	Own system	Reference-Sender-system
Integration to Distribution	X	X	X	X	Own system	Reference-Sender-/ receiver-system
Administrate systems and technical contact person	X	X	X	X	Test: https://admin.test.digitalpost.dk/login Prod: https://admin.digitalpost.dk/login	Manuals (see ”Reference”).
Administrate access to prod					Rights management portal (Rettighedsportalen) https://rettighedsportal.digitalpost.dk/home	Manuals (see “Reference”).
Establish contact structure		X	X	X	Administration Portal: ’Administrativ Adgang’ under Kontaktstruktur	Manuals (see ”Reference”).