SignaturGruppen

Nets eID BROKER

Identity Providers

Version 1.2.12 2023

Table of Contents

Terminology

Term	Description
Nets eID Broker (**NEB**)	Nets eID Broker. Certified MitID Broker and general broker and identity provider for enterprise services.
Nets eID Broker Administration web-interface (**ADM-UI**)	Nets eID Broker Administration web-interface. Interface allowing configuration and administration of the integration

API reference

Swagger endpoint URL	Description
[Authority URL]/swagger/index.html	The swagger description of the available Nets eID Broker API

Nets eID Broker documentation documents

This document services as the primary source of technical documentation. The collected list of technical documents for NEB is listed here.

All documents can be found at https://broker.signaturgruppen.dk/

Title	Description
Nets eID Broker Technical Reference **[NEB-TECHREF]**	The primary source of technical documentation for the integration to Nets eID Broker (this document).
Nets eID Broker Frontend Guide **[NEB-FRONTEND]**	Examples and guides for the frontend integration to Nets eID Broker and OpenID Connect in general. Also covering mobile apps.
Nets eID Broker Sessions **[NEB-SESSIONS]**	Technical information about sessions in the NEB platform. Contains information about how NEB handles session and the various ways service providers can utilize and handle this in their integration.
Nets eID Broker OpenID Connect Intro **[NEB-INTRO]**	
Nets eID Broker Identity Providers **[NEB-IDP]**	Contains technical information about the available identity providers supported by NEB.

Changelog

Version 1.2.12

- Updated the section "International identity providers by Nets E-Ident"

Version 1.2.11

- Added MitID section "Supported browsers"

Version 1.2.10

- Updated incorrect MitID parameter transaction_text and transaction_text_type, was previously sign_text in this document.

Version 1.2.9

- Added MitdID Erhverv mitid_erhverv_user_aborted error code

Version 1.2.8

Added MitID Erhverv

Version 1.2.7

Removed description of MitID app-switch. Information moved to separate document.
Updated description of NemID error codes.

Version 1.2.6

- Added MitID identity provider parameter "action_text" description

Version 1.2.5

Added Nets eID Broker documentation documents section
Added MitID identity provider parameter "cpr_hint" description.

Version 1.2.4

- Added mitid_invalid_transfer_token_code error (MitID).

Version 1.2.3

- Added mitid_identity_not_found error (MitID).

Version 1.2.2

- Updated MitID identity provider scope nemid.pid description (MitID)
- Added nemid.pid_status claim for nemid.pid scope flows (MitID)
- Added description of transaction actions for MitID
- Added description of transaction actions for NemID
- Changed mitid.transaction_id from a transaction specific claim to be included as a session claim. This makes the mitid.transaction_id claim available via the Userinfo endpoint, in the userinfo token and included in the transaction token, if the "mitid" scope is requested.
- Added description of the NemID PID for MitID flows API

Version 1.2.1

- Updated description of NemID Private to Business
- Updated description of MitID error codes

Version 1.2

- Updated description of identity provider parameters for MitID and NemID with examples
- Removed unsupported NemID parameter "remember_userid" from the document.

Version 1.1

- Added description of MitID parameter enable_app_switch.

Introduction

This document describes the available identity providers supported as predefined services in NEB.

The intended audiences are IT developers and IT architects.

Business functionality specified in this document may be subject to different commercial agreement requirements.

General information, online demonstration, documentation (including newest version of this document) and example code is found at https://broker.signaturgruppen.dk.

Danish identity providers

This section covers the available national identity providers available via NEB.

Supported Danish identity providers

MitID
MitID Erhverv
NemID

MitID

The MitID identity provider is the official Danish national electronic identity, replacing NemID.

More information is found here: https://digst.dk/it-loesninger/mitid/.

MitID follows the "National Standarder for Identiteters Sikringsniveauer" (NSIS) and all MitID flows is mapped to one of authentication Level of Assurance's (LoA) found in the NSIS specification: https://digst.dk/it-loesninger/nemlog-in/det-kommende-nemlog-in/vejledninger-og-standarder/nsis-standarden/.

Supported browsers

MitID only test and support the browsers that have at least 2% market share and thus it is a requirement, that app switch integrations utilize the default browsers of the respective OS in question.

It is a security requirement that the end-user is presented with the address bar from the browser to allow the end-user to verify the https://mitid.dk domain during the MitID flow.

Further it is a requirement that the browser utilized supports all the MitID authenticator options, including the MitID chip which is based on Web Authentication.

Embedded browsers are not supported and not allowed (* they are allowed in iframe flow variants) – thus integrations must adhere to the Android Custom Tabs or iOS SFSafariViewController browser integrations and it not allowed to use a standard embedded browser for the integration to MitID.

Supported OIDC parameters

Request parameter	Description
idp_values	**mitid**

Supported identity provider parameters (idp_params -> mitid)

Identity Provider parameters (mitid)	Description
reference_text Note: **This text will be displayed to the user in all MitID flows inside the MitID client, but only at the last authenticator shown.**	Type: Base64 encoded string. The reference text containing the transaction content (e.g., “Transfer <amount> to <ac-count>”). It will be shown to the user in the MitID App. It is limited to 130 characters.
transaction_text *This parameter will be ignored for unsigned requests.*	Type: Base64 encoded string. The transaction text is presented to the end-user as part of the MitID flow and allows service providers to provide a transactional context for the MitID flow. This can be of the types valid for the transasction_text_type parameter.
transaction_text_type	Type: String Specifies the type of the transaction_text parameter. One of • text • html If text is specified, the transaction_text will be displayed “as-is” without any rendering, as a plain text value. If html is specified, the content will be displayed and rendered as HTML. The allowed HTML is restricted as specified in this document.
uuid_hint	Type: String If this parameter is set with the end-user MitID UUID, the MitID flow will be automatically started for the MitID identity with the specified MitID UUID.
cpr_hint *It is required encrypt requests when sending cpr_hint, see **[NEB-TECHREF]** for reference.*	Type: String If this parameter is set with the end-user CPR, the MitID flow will be automatically started for the MitID identity with the specified CPR.
require_psd2	Type: Boolean If this parameter is set to true, the individual authentication flow will be PSD2 compliant by restricting what information can be revealed to the end-user during authentication.
loa_value	Specifies the requested Level of Assurance level for the MitID flow.

	One of - low - substantial - high if both loa_value and aal_value is undefined in the request, the default will be set to loa_value=substantial.
aal_value	Specifies the requested Authenticator Assurance Level for the MitID flow. One of - low - substantial - high if loa_value is set, aal_value will be ignored. This enables to request a MitID flow where the aal level might be higher than the (expected) Level of Assurance (loa), i.e. ial=low, aal=substantial => loa=low.
enable_step_up *Requires prompt=login* *Requires loa_value or aal_value higher than the existing session from which to step-up from.*	Type: bool Specifies that the MitID step-up functionality is requested. To activate the MitID step-up flow the prompt=login must be specified to instruct the flow to reauthenticate. Then if the enable_step_up is set to true and if the requested loa og aal value is higher than the existing session for the end-user the MitID authentication will start in step-up mode.
action_text	Type: string, default: null Allows to specify one of the accepted “action” header texts for the MitID client flow. Must be one of the following - "LOG_ON" - "APPROVE" - "CONFIRM" - "ACCEPT" - "SIGN"

Example JSON for identity providers

```
{“mitid”:{“loa_value”:”substantial”, “enable_step_up”:true, “uuid_hint”: “efc7ffb4-e086-4f5f-a1d5-b3c7227db629”}}


idp_params=%7B%E2%80%9Cmitid%E2%80%9D%3A%7B%E2%80%9Cloa_value%E2%80%9D%3A%E2%80%9Dsubstantial%E2%80%9D%2C%20%E2%80%9Cenable_step_up%E2%80%9D%3Atrue%2C%20%E2%80%9Cuuid_hint%E2%80%9D%3A%20%E2%80%9Cefc7ffb4-e086-4f5f-a1d5-b3c7227db629%E2%80%9D%7D%7D
```

Supported scope values

Scope	Description
mitid	List of claims: - mitid.uuid - mitid.date_of_birth - mitid.age - mitid.identity_name - mitid.ial_identity_assurance_level - mitid.transaction_id
transaction_token	The transaction_token scope requests the transaction token from the Token endpoint including the following claims. These claims are not shared in a SSO with other services. transaction_id mitid.transaction_text mitid.transaction_text_type mitid.reference_text
ssn	Social Security Number. List of claims: - dk.cpr Will trigger MitID CPR user-interaction.

ID Token identity claims

Claim value	Possible values
identity_type	private
idp	**mitid**
loa	Level of Assurance One of - https://data.gov.dk/concept/core/nsis/Low - https://data.gov.dk/concept/core/nsis/Substantial - https://data.gov.dk/concept/core/nsis/High
ial	Identity Assurance Level One of - https://data.gov.dk/concept/core/nsis/Low - https://data.gov.dk/concept/core/nsis/Substantial - https://data.gov.dk/concept/core/nsis/High
aal	Authenticator Assurance Level One of - https://data.gov.dk/concept/core/nsis/Low - https://data.gov.dk/concept/core/nsis/Substantial - https://data.gov.dk/concept/core/nsis/High
amr	The list of authenticators used to achieve the resulting LoA. Possible values are: - password

	- code_token - code_reader - code_app - code_app_enchanced - u2f_token
mitid.psd2	The mitid.psd2 claim is only issued as an ID Token identity claim if the authentication of an end-user is PSD2 compliant.

Transaction token MitID specific claims

Claim value	**Possible values**
mitid.uuid	Same value as for ID token.
mitid.reference_text	Passthrough of the MitID **reference_text** identity provider parameter.
mitid.transaction_text_sha256	Base64 encoded SHA256 digest of the MitID **transactiontext** identity provider parameter.
mitid.transaction_text_type	Passthrough of the MitID **transactiontexttype** identity provider parameter
mitid.psd2	The mitid.psd2 claim is always issued as a transaction token MitID specific claim.
transaction_actions	Type: string (single value) or JSON list Only set, if one or more of the following transaction actions where performed: - mitid.login (Login completed) - mitid.reuse_jwt (Automatic reuse of existing session) - mitid.sso_login (SSO login completed) - mitid.controlled_transfer (Controlled Transfer completed) - mitid.transaction_signing (Transaction signing) - mitid.cpr_match (CPR match completed) - mitid.cpr_lookup (Automatic CPR lookup completed)

MitiD Transaction signing

Nets eID Broker supports a transaction signing flow which enables the end-user to approve a transaction text based on text or HTML, as part of the MitID authentication. Transaction signing flow is limited to only signed requests.

This is done by setting the **transaction_text** and **transaction_text_type** MitID identity provider parameters.

The end-user will be shown the text/HTML and will have to approve the text to complete the transaction.

MitID natively supports the **reference_text** (130 characters) parameter which enables a limited size and format to present the end-user with detailed information about the transaction.

If the transaction token is requested, a NEB sealed record of the transaction is returned including all the relevant parameters used to complete the transaction.

If **transaction_text_type** is set to *html*, the HTML content of the **transaction_text** is restricted to a set of qualified tags and parsed to protect against possible malicious content and flow breakage.

Allowed HTML tags:

> *html, body, head, style, title, div, p, ul, li, h1, h2, h3, h4, h5, h6, table, font, tr, th, td, i, u, b, center, a, q, small*

Disallowed expressions and attributes:

CSS expressions and embedded script links for *style* tag.
CSS expression attributes.
Any on- attributes, such as *onload, onclick etc.*
Script link attributes, such as *src, dynsrc, lowsrc, javascript:* etc.

MitID CPR flow

CPR is available from MitID flows if you are a public service provider. In this scenario, NEB will set **dk.cpr** in the result, if requested via the **ssn** scope.

If you are a private service provider, the user's CPR will not be available from the MitID system. In this scenario, a CPR Match service is provided (see MitID CPR Match API), available for MitID Brokers making it possible to match an active MitID session and CPR and verify if the supplied CPR matches the authenticated MitID identity and thus making it possible to verify if a MitID identity has the given CPR.xxxxxx.

NEB implements this as a natural part of the MitID flow and will ask the user for CPR when the service provider requests CPR with the **ssn** scope.

Note, *that it is supported to request CPR via the CPR flow by reauthenticating a user with the additional* ***ssn*** *scope. In this case, NEB will reuse the active MitID session and ask the user for CPR (but will not ask for login), do the required CPR Match verification, and return the CPR to the service. This enables services to only ask for CPR using the CPR flow when needed for specific users.*

Note that MitID only allows MitID Match for 15 minutes after the MitID session was issued.

MitID CPR Match API

See the swagger API reference for details.

The Broker API supports a "MitID CPR Match API" that allows services to match a CPR with a MitID authentication from NEB.

In this way, services can ask the user for CPR and then call the API with the access token retrieved from NEB for the user authentication as authorization header.

This also allows services to verify that an already known CPR matches the MitID identity in question.

Note that MitID only allows MitID Match for 15 minutes after the MitID session was issued.

If "cprNumberMatch" returns false, it means that it is not possible to match the "cpr" input parameter with the CPR-number of the user. MitID restricts CPR-matching to a maximum of three tries per session, in which case the endpoint will return the following response:

To reset the CPR matching exceeded limitation, the client must prompt the user for reauthentication using MitID.

MitID error codes

Possible error codes from the MitID identity provider flow.

Error (error)	Error description (error_description)	Description
access_denied	mitid_user_aborted	The end-user aborted the MitID flow. This can happen during the MitID login process and during the MitID CPR Match flow.

access_denied	mitid_no_ctx	Some or all the required flow context was missing. This might happen if the end-user visits the MitID flow URLs (via a bookmark or similar) at a time where no session or context of the flow exists for the end-user. In some circumstances we can redirect the end-user back to the service provider with this errorcode. The recommended approach is to guide the end-user to a normal front-page.
access_denied	mitid_internal_error	An internal error in the MitID flow.
access_denied	mitid_unexpected_error	An unexpected error happened. We continually monitor for these to identity better ways to handle the observed issues.
access_denied	mitid_core_client_error	Error in the frontend package from MitID. NEB tries to remedy these errors with automatic retries, but multiple consecutive errors within the MitID client might result in this error.
access_denied	mitid_cpr_match_failed	The end-user CPR number could not be validated. This can happen if the end-user has exceeded allowed number of CPR validation attempts or if the end-user does not have a CPR number associated with the selected MitID identity.
access_denied	mitid_timeout	MitID did not respond in time. This signals that the official MitID APIs timed out during the flow (not the NEB APIs).
access_denied	mitid_auth_code_already_used	Attempt made to reuse a MitID authentication code.

		Internal error which typically is due to an API retry on top of an assumed timeout. This might also happen if the end-user re-posts the specific step of the MitID flow.
access_denied	mitid_transfer_token_exchange_code_already_used	Invalid attempt made to reuse MitID transfer token exchange code more than once.
access_denied	mitid_transaction_text_invalid	Invalid transaction signing text. The input is malformed or has invalid content.
access_denied	mitid_transaction_signing_flow_limited_to_signed_request	Transaction signing flow was not possible due to signed request limitation. The authentication request must be signed to utilize the transaction signing functionality.
access_denied	mitid_anti_forgery_validation_error	The MitID flow encountered an invalid anti forgery cookie validation. This can happen if the end-user re-visits an earlier page in the flow ex. by clicking back in the browser.
access_denied	mitid_transaction_text_missing	Missing the transaction text in the signing flow. Invalid request.
access_denied	mitid_identity_not_found	When provided uuid_hint is not a valid or active uuid.
access_denied	mitid_invalid_transfer_token_code	Controlled Tranfer code is already used or expired.

MitID SSO

NEB implements and supports MitID SSO and allows integrating services to utilize MitID SSO for automatic sharing MitID sessions in a SSO defined and managed by participating MitID Brokers.

Any client, service or service provider can be member of up to one MitID SSO Group and if so, will automatically map all MitID sessions to this MitID SSO and automatically reuse an existing session if already active within this MitID SSO.

It is possible to setup MitID SSO groups with other MitID Brokers and other MitID Broker services and service providers.

NEB will handle the required end-user consent, which is required when automatically reusing an active MitID session.

Note, contact Signaturgruppen if you plan to use this feature.

MitID SSO logout

It is required by all clients who utilize a MitID SSO to inform their MitID broker of logout events from the end-user.

Logout is handled either by sending the end-user to the End session endpoint with the original issued ID token or by calling the Logout endpoint with the original issued ID token. See general section about logout in this document for more details.

MitID SSO requires that all logout events are handled using Back-channel endpoints and thus enabling the termination of MitID SSO sessions without the need of the end-user browser. MitID SSO groups are by this forced to be Back-channel SSO groups.

The only way participating service providers can receive logout events is by registering a valid Back-channel endpoint for their integration at NEB.

Participating service providers will be able to get the session status from the Userinfo endpoint.

MitID Controlled Transfer

With MitID Controlled Transfer, a requesting service provider can request and retrieve a “MitID Controlled Transfer Token Exchange Code” (MitID CT Exchange Code) from their MitID Broker. Then, another service provider can exchange this to a MitID authentication from their respective MitID Broker.

Flow:

- Service provider A requests a MitID CT Exchange Code from Broker A and specifies a Transfer Token Text
- Service Provider A redirects end-user to Service Provider B with the MitID CT Exchange Code and Transfer Token Text
- Service Provider B uses Broker B and the implementation provided by Broker B to exchange the MitID CT Exchange Code token and the Transfer Token Text to a MitID authentication enabling the end-user login at Service Provider B.

The protocol for exchanging MitID CT Exchange Code between service providers are up to the two exchanging service providers to define.

The protocol for retrieving or exchanging MitID CT Exchange Code between service providers and MitID brokers are up to each broker to define.

The specification for retrieval and exchange towards the NEB interface is specified here. Note that it is up to each agreement with other service providers, how the MitID CT Exchange Code is exchanged – this is not specified nor handled by NEB.

NOTE: The requesting service provider has a mandatory requirement of getting the correct consent from the end-user, before sending the end-user to another service provider with a MitID CT Exchange Code. The official description of the requirement is:

> *When the user is performing a controlled transfer from service provider A to service provider B, it is the responsibility of service provider A to get a consent from the end user, regarding eID attributes which service provider A has requested on initial request, since these attributes will be available to service provider B.*

Requesting a Controlled Transfer Token Exchange Code

A Controlled Transfer Token Exchange Code is retrieved by calling the MitID Controlled Transfer Token Exchange Code endpoint (see the swagger API reference for details).

Parameters	**Description**
targetBrokerId	MitID Broker ID of receiving MitID broker
targetServiceProviderId	MitID Service Provider ID of receiving service provider
transferTokenText	Type: String The calling service provider must specify a Transfer Token Text and hand this out to the receiving service provider. The Transfer Token Text is restricted by MitID to a maximum of 130 characters. Any length above 130 will result in an unsuccessful request.

Exchanging a Controlled Transfer Token Exchange Code to a MitID login

As a service provider integrating to NEB, it is possible to exchange a MitID CT Exchange Code received from another service provider for a MitID login.

The MitID CT Exchange Code is used by NEB in exchange of a MitID authentication token from the MitID APIs and as such enables NEB to automatically login an end-user in exchange for the MitID CT Exchange Code.

The MitID CT Exchange Code is passed as a parameter to the MitID identity provider in the NEB OIDC specification, alongside the received Transfer Token Text, and will be used to login the end-user based on the MitID session used to create the MitID CT Exchange Code in the first place.

NemID PID claim for MitID flows

It is possible to request the NemID PID claim (nemid.pid) for MitID flows.

This is done by specifying the scope nemid.pid which will trigger the relevant end-user flow required to return the nemid.pid claim along with the MitID login.

NemID PID scope	**Claims**
nemid.pid	• nemid.pid • nemid.pid_status If set, the nemid.pid claim contains the NemID PID. nemid.pid_status can have one of the following values • success

	- unable_to_lookup

To retrieve the PID from a MitID login Nets eID Broker will have to lookup PID from a NemLog-In3 supplied supporting service using the end-user Danish CPR number. The end-user will be guided through the needed steps automatically when the nemid.pid scope is specified.

It is recommended, that the nemid.pid scope is only specified when the returned MitID identity is unknown to the service in question, as the end-user will have to enter his Danish CPR number when this scope is specified.

The nemid.pid scope can be used as a reauthentication scope (see section about reauthentication) and thus NEB will automatically reuse an available user session to ensure that the end-user does not need to authenticate with MitID additional times.

Note, that the scopes “nemid.pid” and “ssn” can be used together or separately. If ssn is specified that dk.cpr claim will be issued to the service provider, but both nemid.pid and ssn scopes will trigger CPR matching for the end-user. It is thus possible to minimize the data handed back to the service provider by controlling it this way.

If the NemID PID could not be retrieved the nemid.pid_status claim will indicate the result.

NemID PID API for MitID flows

Endpoint: /mitid/nemidPidLookup (see the swagger API reference for details)

Allows a service provider to lookup the NemID PID based on a successful MitID session for the end-user.

The endpoint allows for an optional CPR parameter, which is used to complete the lookup.

If the ssn flow was used for the authentication flow the CPR is known by NEB for the current session and thus CPR is not needed for the lookup.

If the ssn flow was not used, the service provider must provide the correct CPR for the end-user as part of the lookup.

MitID service provider parameters

Identity Provider parameters (mitid)	**Description**
transfer_token_exchange_code	Type: string The issued MitID Controlled Transfer Token Exchange Code Example: `”38e195d6-14ad-4ed9-9f0b-d72a26c8ed95”`
transfer_token_text	Type: string (non-empty) A Controlled Transfer Token Text, which is up to the calling Service Provider to define and set. This text must be handed out to the receiving service provider together with the MitID Controlled Transfer Token Exchange Code.

Note: See general section on identity provider parameters for reference on how to set the parameters.

MitID Erhverv

MitID Erhverv is the official digital business solution for companies, organizations, and authorities in Denmark.

More information is found here: https://digst.dk/it-loesninger/mitid-erhverv/.

Supported OIDC parameters

Request parameter	Description
idp_values	**mitid_erhverv**

Supported identity provider parameters (idp_params -> mitid_erhverv)

Note here, that if the user enters the MitID flow for authentication, the specific MitID identity provider parameters is applied to the MitID flow, see more in the MitID identity provider section for information on MitID.

Identity Provider parameters (mitid_erhverv)	Description
allow_private	Type: Boolean, default: false If set to true, then the user will be allowed to select to continue with this private MitID eID. If no available professional identities are available for the user and the **allow_true** is set to true, then the private MitID identity is automatically selected.

Example JSON for identity providers (JSON URL encoded)

```
{“mitid_erhverv”:{“allow_private”:true}}


idp_params=%7B%E2%80%9Cmitid_erhverv%E2%80%9D%3A%7B%E2%80%9Callow_priv
ate%E2%80%9D%3Atrue%7D%7D
```

Supported scope values

Scope	Description
nemlogin	List of claims: • nemlogin.date_of_birth • nemlogin.email • nemlogin.name • nemlogin.family_name • nemlogin.given_name • nemlogin.nemid.rid • nemlogin.org_name • nemlogin.persistent_professional_id • nemlogin.cvr • nemlogin.se_number • nemlogin.p_number • nemlogin.cpr_uuid (for private service providers) • nemlogin.cpr (for public service providers)

ID Token identity claims

Claim value	Possible values
identity_type	professional
idp	**mitid_erhverv**
loa	Level of Assurance One of • https://data.gov.dk/concept/core/nsis/Low • https://data.gov.dk/concept/core/nsis/Substantial The Nets eID Broker is a registered NSIS Substantial broker and thus cannot issue higher than Substantial.
ial	Identity Assurance Level One of • https://data.gov.dk/concept/core/nsis/Low • https://data.gov.dk/concept/core/nsis/Substantial • https://data.gov.dk/concept/core/nsis/High
aal	Authenticator Assurance Level One of • https://data.gov.dk/concept/core/nsis/Low • https://data.gov.dk/concept/core/nsis/Substantial • https://data.gov.dk/concept/core/nsis/High
amr	The list of authenticators used to achieve the resulting AAL/LoA. Possible values are: • mitid:password • mitid:code_token • mitid:code_reader • mitid:code_app • mitid:code_app_enchanced • mitid:u2f_token

Transaction token MitID specific claims

Claim value	Possible values
mitid.reference_text	Passthrough of the MitID **reference_text** identity provider parameter.
mitid.psd2	The mitid.psd2 claim is always issued as a transaction token MitID specific claim.
transaction_actions	Type: string (single value) or JSON list Only set, if one or more of the following transaction actions where performed: • mitid.login (Login completed) • mitid_erhverv.identity_selected

MitID Erhverv error codes

Possible error codes from the MitID Erhverv identity provider flow.

Error (error)	Error description (error_description)	Description
access_denied	mitid_erhverv.no_identities	No professional identities available for the user based on the used MitID identity selected by the user.
access_denied	mitid_erhverv_user_aborted	User clicked abort

NemID

The NemID identity provider is the official Danish national electronic identity, being replaced by MitID.

Supported parameters

Request parameter	Description
idp_values	**nemid**

Supported identity provider parameters

Identity Provider parameters (nemid)	Description
amr_values	Optional space separated list of of one or more of • nemid.otp • nemid.keyfile If not specified the amr_values will default to nemid.otp only. NemID has built-in two clients: the NemID OTP and the NemID Keyfile client. NemID OTP is the default client used by most private NemID logins enabling the NemID App and Code-Card. NemID Keyfile is the default client used by most professional NemID logins enabling most of the enterprise login-variants used by business-employess for authenticating with NemID.
code_app_trans_ctx	Type: Base64 encoded string. Up to 100 characters. Showed in the NemID Code App if the end-user choses the NemID Code App when authenticating.
transaction_text	Type: Base64 encoded string. NemID Sign Text. If set, invokes the NemID Sign flow. The supported types of input, is specified by the **transaction_text_type** parameter.
transaction_text_type	Type: string Specifies the type of the sign_text parameter. The supported values are

	- html - text
private_to_business	Type: bool (default: false) If set to true, the end-user will be guided through the NemID Private to Business flow. This will guide the end-user to select one of the companies (CVR) for which the end-user is authorized to represent the company alone. The selected CVR-number will be set in the **nemid.auth_to_repr** claim.

Example JSON for identity providers

```
{“nemid”:{“amr_values”:”nemid.keyfile”}}


idp_params=%7B%E2%80%9Cnemid%E2%80%9D%3A%7B%E2%80%9Camr_values%E2%
80%9D%3A%E2%80%9Dnemid.keyfile%E2%80%9D%7D%7D
```

Supported scope values

Scope	**Description**
nemid	List of claims: - nemid.common_name - nemid.pid - nemid.rid - nemid.dn - nemid.ssn - nemid.email - nemid.cvr - nemid.auth_to_repr
transaction_token	The transaction_token scope requests the transaction token from the Token endpoint including the following claims. These claims are not shared in a SSO with other services. transaction_id nemid.code_app_trans_ctx nemid.sign_text nemid.xmldsig
ssn	Social Security Number. List of claims: - dk.cpr Will trigger MitID Demo CPR user-interaction.

ID Token identity claims

Claim value	Possible values
Identity_type	• private • professional Private identities are identifiable by their global NemID PID, found in the **nemid.pid** claim. Professionals are employees, and have a unique NemID RID, found in the **nemid.rid** claim. The NemID RID paired with the CVR from the employer forms the primary identifier for NemID professional identities.
idp	**nemid**
amr	One of the following: • nemid.otp • nemid.keyfile
idp_environment	One of the following • test • production

Transaction token NemID specific claims

Claim value	Possible values
nemid.ssn	Same value as for ID token.
nemid.code_app_trans_ctx	Passthrough of the NemID **code_app_trans_ctx** identity provider parameter.
nemid.sign_text	Passthrough of the NemID **sign_text** identity provider parameter.
dk.cpr	Danish CPR. Included if and only if configured as a requirement for the transaction token for the client in the administration interface. Will trigger the same user interaction as setting the **ssn** scope.
transaction_actions	Type: string (single value) or JSON list Only set, if one or more of the following transaction actions where performed: • nemid.login (Login completed) • nemid.signed (Signature flow completed) • nemid.cpr_match (CPR match completed)

NemID CPR

In the current version of NemID, CPR is available from NemID flows if you are a public service provider. In this scenario, NEB will set **dk.cpr** in the result, if requested via the **ssn** scope.

If you are a private service provider, the user's CPR will not be available from the MitID system. In this scenario, a CPR Match service is provided (using the service providers NemID agreement) making it possible to match a NemID PID and CPR and verify if the supplied PID and CPR matches and thus making it possible to verify if a NemID identity has the given CPR.

NEB implements this as a natural part of the MitID flow and will ask the user for CPR when the service provider requests CPR with the **ssn** scope.

If the user has accepted that the CPR is stored for later use (user consent) and returned to the service provider, the user will not have to enter CPR for subsequent MitID flows for the same service provider.

NemID CPR Match API

See the swagger API reference for details.

The Broker API supports a “NemID CPR Match API” that allows services to match a CPR with a NemID authentication from the NEB.

In this way, services can ask the user for CPR and then call the API with the access token retrieved from NEB for the user authentication as authorization header.

This also allows services to verify that an already known CPR matches the NemID identity in question.

NemID Privat til Erhverv (authorized to represent)

The “NemID Privat til Erhverv” service is available via NEB for NemID flows, see the “private_to_business” NemID idp_params option for reference.

Tt will allow the end-user to select between the available CVR-numbers available for which his or hers private NemID is “Allowed to Represent” the specified company.

This requires that the user enters his CPR number (or that the calling service is allowed for automatic CPR retrieval) after which the CVR-list can be retrieved from the Danish “Erhvervsstyrelsen” and the flow can be completed.

NemID error codes

Possible error codes from the MitID identity provider flow.

Error (error)	**Error description (error_description)**	**Description**
access_denied	nemid_user_aborted	The end-user aborted the NemID flow. This can happen during the NemID login process or during the NemID CPR Match flow.
access_denied	nemid_flow_error_TIMESTAMP_EXPIRED	The signed timestamp of the NemID parameters has expired. If the end-user has taken too long to complete the NemID flow, this code is returned.
access_denied	nemid_flow_error_[error code]	A general error code that signals an error that should be handled as a technical error and nothing relevant to tell the end-user. In general, handle this as a generic error. The “error code” signals a specific error, which can be used if contacting Nets eID Broker support.

access_denied	nemid_unspecified_error	Unknown cause of error.
access_denied	nemid_client_flow_error_+[NemID error code]	For a specific NemID related errorcode (returned by the NemID client) the error could look like “**nemid_srv001**” A reference for NemID documentation is found in [NEMID-TU]

International identity providers by Nets E-Ident

This section describes the identity providers that are available via Nets E-Ident.

NEB will help with the setup and take care of the integration to E-Ident and provide a flexible usage of the services provided by the E-Ident service.

This allows your integrations to utilize all the identity providers support by both NEB and E-Ident while utilizing all other features from the NEB platform.

Supported Nets E-Ident identity providers

- BankID Norway
- BankID on mobile Norway
- Buypass Norway
- BankID Sweden

All eIDs available from E-Ident, found at https://www.nets.eu/developer/e-ident/eids/Pages/default.aspx.

References

[NEMID-TU] “NemID TU documentation” https://www.nets.eu/dk-da/kundeservice/nemid-tjenesteudbyder/NemID-tjenesteudbyderpakken/Pages/dokumentation.aspx