



## Identity Providers

Version 1.4 2024

### Table of Contents

Terminology .....	3
API reference .....	3
Nets eID Broker documentation documents .....	3
Changelog .....	4
Version 1.4 .....	4
Version 1.3 .....	4
Version 1.2.16 .....	4
Version 1.2.14 .....	4
Version 1.2.14 .....	4
Version 1.2.13 .....	4
Version 1.2.12 .....	4
Version 1.2.11 .....	4
Version 1.2.10 .....	4
Introduction .....	5



Danish identity providers.....	5
Supported Danish identity providers .....	5
MitID .....	5
Supported browsers.....	5
Supported OIDC parameters .....	6
Supported identity provider parameters (idp_params -> mitid).....	6
Example JSON for identity providers .....	8
Supported scope values .....	8
ID Token identity claims.....	8
Transaction token MitID specific claims.....	9
MitID Transaction signing.....	9
MitID CPR flow.....	10
MitID CPR Match API .....	10
MitID SSO .....	11
MitID Controlled Transfer.....	11
Exchanging a Controlled Transfer Token Exchange Code to a MitID login.....	12
NemID PID claim for MitID flows.....	12
NemID PID API for MitID flows .....	13
SSN details claims for MitID flows .....	13
MitID service provider parameters .....	14
MitID Erhverv.....	15
Supported OIDC parameters .....	15
Supported identity provider parameters (idp_params -> mitid_erhverv).....	15
Example JSON for identity providers (JSON URL encoded) .....	15
Supported scope values .....	15
ID Token identity claims.....	16
Transaction token MitID specific claims.....	16
References.....	17



## Terminology

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

## API reference

Swagger endpoint URL	Description
[ <b>Authority URL</b> ]/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 [ <b>NEB-TECHREF</b> ]	The primary source of technical documentation for the integration to Nets eID Broker (this document).
Nets eID Broker Frontend Guide [ <b>NEB-FRONTEND</b> ]	Examples and guides for the frontend integration to Nets eID Broker and OpenID Connect in general. Also covering mobile apps.
Nets eID Broker Sessions [ <b>NEB-SESSIONS</b> ]	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 [ <b>NEB-INTRO</b> ]	
Nets eID Broker Identity Providers [ <b>NEB-IDP</b> ]	Contains technical information about the available identity providers supported by NEB.



# Changelog

## Version 1.4

- Added specification of retrieving information from the Danish national social security system, see section “SSN details claims for MitID flows”.

## Version 1.3

- Removed description of error codes. See **[NEB-TECHREF]** for general documentation on error codes. A new documentation endpoint has been provided.

## Version 1.2.16

- Added MitID Erhverv errorcode: mitid\_erhverv\_api\_error

## Version 1.2.14

- Added errorcode “mitid\_loa\_aal\_invalid\_parameter”.

## Version 1.2.14

- Added new errorcodes to the document

## Version 1.2.13

- Added MitID error code: “mitid\_core\_client\_load\_error”.

## 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.



## 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 Identiteter 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 -&gt; mitid)

Identity Provider parameters (mitid)	Description
reference_text  <b>Note:</b> 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  <i>This parameter will be ignored for unsigned requests.</i>	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 transaction_text_type parameter.
transaction_text_type	Type: String  Specifies the type of the transaction_text parameter.  One of <ul style="list-style-type: none"><li>• text</li><li>• html</li></ul> 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  <i>It is required encrypt requests when sending cpr_hint, see [NEB-TECHREF] for reference.</i>	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	<p>Specifies the requested Level of Assurance level for the MitID flow.</p> <p>One of</p> <ul style="list-style-type: none"> <li>• low</li> <li>• substantial</li> <li>• high</li> </ul> <p>if both loa_value and aal_value is undefined in the request, the default will be set to loa_value=substantial.</p>
aal_value	<p>Specifies the requested Authenticator Assurance Level for the MitID flow.</p> <p>One of</p> <ul style="list-style-type: none"> <li>• low</li> <li>• substantial</li> <li>• high</li> </ul> <p>if loa_value is set, aal_value will be ignored.</p> <p>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 =&gt; loa=low.</p>
enable_step_up <i>Requires prompt=login</i> <i>Requires loa_value or aal_value higher than the existing session from which to step-up from.</i>	<p>Type: bool</p> <p>Specifies that the MitID step-up functionality is requested.</p> <p>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.</p>
action_text	<p>Type: string, default: null</p> <p>Allows to specify one of the accepted "action" header texts for the MitID client flow.</p> <p>Must be one of the following</p> <ul style="list-style-type: none"> <li>• "LOG_ON"</li> <li>• "APPROVE"</li> <li>• "CONFIRM"</li> <li>• "ACCEPT"</li> <li>• "SIGN"</li> </ul>

### Example JSON for identity providers

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

```
idp_params=%7B%22mitid%22%3A%7B%22loa_value%22%3A%22substantial%22%2C%22enable_step_up%22%3Atrue%2C%22uuid_hint%22%3A%22efc7ffb4-e086-4f5f-a1d5-b3c7227db629%22%7D%7D
```

**Supported scope values**

Scope	Description
mitid	List of claims: <ul style="list-style-type: none"><li>• mitid.uuid</li><li>• mitid.date_of_birth</li><li>• mitid.age</li><li>• mitid.identity_name</li><li>• mitid.ial_identity_assurance_level</li><li>• mitid.transaction_id</li></ul>
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: <ul style="list-style-type: none"><li>• dk.cpr</li></ul> Will trigger MitID CPR user-interaction.

**ID Token identity claims**

Claim value	Possible values
identity_type	private
idp	<b>mitid</b>
loa	Level of Assurance  One of <ul style="list-style-type: none"><li>• https://data.gov.dk/concept/core/nsis/Low</li><li>• https://data.gov.dk/concept/core/nsis/Substantial</li><li>• https://data.gov.dk/concept/core/nsis/High</li></ul>
ial	Identity Assurance Level  One of <ul style="list-style-type: none"><li>• https://data.gov.dk/concept/core/nsis/Low</li><li>• https://data.gov.dk/concept/core/nsis/Substantial</li><li>• https://data.gov.dk/concept/core/nsis/High</li></ul>
aal	Authenticator Assurance Level  One of <ul style="list-style-type: none"><li>• https://data.gov.dk/concept/core/nsis/Low</li><li>• https://data.gov.dk/concept/core/nsis/Substantial</li><li>• https://data.gov.dk/concept/core/nsis/High</li></ul>
amr	The list of authenticators used to achieve the resulting LoA.





	Possible values are: <ul style="list-style-type: none"> <li>• password</li> <li>• code_token</li> <li>• code_reader</li> <li>• code_app</li> <li>• code_app_enhanced</li> <li>• u2f_token</li> </ul>
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 <b>reference_text</b> identity provider parameter.
mitid.transaction_text_sha256	Base64 encoded SHA256 digest of the MitID <b>transactiontext</b> identity provider parameter.
mitid.transaction_text_type	Passthrough of the MitID <b>transactiontexttype</b> 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: <ul style="list-style-type: none"> <li>• mitid.login (Login completed)</li> <li>• mitid.reuse_jwt (Automatic reuse of existing session)</li> <li>• mitid.sso_login (SSO login completed)</li> <li>• mitid.controlled_transfer (Controlled Transfer completed)</li> <li>• mitid.transaction_signing (Transaction signing)</li> <li>• mitid.cpr_match (CPR match completed)</li> <li>• mitid.cpr_lookup (Automatic CPR lookup completed)</li> </ul>

### 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 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	<ul style="list-style-type: none"><li>nemid.pid</li><li>nemid.pid_status</li></ul> <p>If set, the nemid.pid claim contains the NemID PID. nemid.pid_status can have one of the following values</p> <ul style="list-style-type: none"><li>success</li><li>unable_to_lookup</li></ul>

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.

### SSN details claims for MitID flows

It is possible to request detailed information about the person based of the Danish social security system for MitID flows. Service providers can request name and/or address details for the person. To request the details the scopes ssn.details\_name and ssn.details\_address will trigger the relevant end-user flow required to return the detailed information for the name and address respectively along with the MitID login.

SSN details scope	Claims
ssn.details_name ssn.details_address	<p>Shared claims:</p> <ul style="list-style-type: none"><li>ssn.details.status</li><li>ssn.details.name_address_protected</li></ul> <p>The ssn.details.status-claim can have one of the following to values:</p> <ul style="list-style-type: none"><li>success</li><li>unable_to_lookup</li></ul> <p>If the value of the ssn.details.status-claim is unable_to_lookup, then no other claims are returned.</p> <p>The ssn.details.name_address_protected-claim contains a boolean value. If the claim value is "true" then no other claims from ssn.details-scopes are returned.</p> <p>The shared claims are only returned once even if both ssn.details-scopes are requested.</p>
ssn.details_name	<p>List of claims:</p> <ul style="list-style-type: none"><li>ssn.details.given_name</li><li>ssn.details.middle_name</li></ul>



	<ul style="list-style-type: none"><li>• <code>ssn.details.surname</code></li></ul>
<code>ssn.details_address</code>	List of claims: <ul style="list-style-type: none"><li>• <code>ssn.details.care_of_name</code></li><li>• <code>ssn.details.city_name</code></li><li>• <code>ssn.details.floor</code></li><li>• <code>ssn.details.street_name</code></li><li>• <code>ssn.details.street_number</code></li><li>• <code>ssn.details.unit</code></li><li>• <code>ssn.details.zip_code</code></li><li>• <code>ssn.details.zip_name</code></li></ul>

Please note that the transaction token will contain a `transaction_action`-claim with the value `"ssn.details_lookup"`, when one or both of the `ssn.details`-scopes are requested as outlined in the section "Transaction token MitID specific claims".

To retrieve the detailed information for the person from a MitID login Nets eID Broker will have to look up the information from the Danish national social security system using the Danish CPR number. The end-user will be guided through the needed steps automatically when one of or both the `ssn.details`-scopes are specified.

It is recommended, that the `ssn.details`-scopes only are 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 these scopes are specified.

Note, that the scopes `"ssn.details_name"`, `"ssn.details_address"` and `"ssn"` can be used together or separately. If `ssn` is specified that `dk.cpr` claim will be issued to the service provider, but all 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.

### MitID service provider parameters

Identity Provider parameters (mitid)	Description
<code>transfer_token_exchange_code</code>	Type: string The issued MitID Controlled Transfer Token Exchange Code Example: <code>"38e195d6-14ad-4ed9-9f0b-d72a26c8ed95"</code>
<code>transfer_token_text</code>	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 <b>allow_true</b> 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_private%E2%80%9D%3Atrue%7D%7D
```

### Supported scope values

Scope	Description
nemlogin	List of claims: <ul style="list-style-type: none"><li>• nemlogin.date_of_birth</li><li>• nemlogin.email</li><li>• nemlogin.name</li><li>• nemlogin.family_name</li><li>• nemlogin.given_name</li><li>• nemlogin.nemid.rid</li><li>• nemlogin.org_name</li><li>• nemlogin.persistent_professional_id</li><li>• nemlogin.cvr</li><li>• nemlogin.se_number</li><li>• nemlogin.p_number</li><li>• nemlogin.cpr_uuid (for private service providers)</li><li>• nemlogin.cpr (for public service providers)</li></ul>

### ID Token identity claims

Claim value	Possible values
-------------	-----------------



identity_type	professional
idp	<b>mitid_erhverv</b>
loa	<p>Level of Assurance</p> <p>One of</p> <ul style="list-style-type: none"> <li>• <a href="https://data.gov.dk/concept/core/nsis/Low">https://data.gov.dk/concept/core/nsis/Low</a></li> <li>• <a href="https://data.gov.dk/concept/core/nsis/Substantial">https://data.gov.dk/concept/core/nsis/Substantial</a></li> </ul> <p>The Nets eID Broker is a registered NSIS Substantial broker and thus cannot issue higher than Substantial.</p>
ial	<p>Identity Assurance Level</p> <p>One of</p> <ul style="list-style-type: none"> <li>• <a href="https://data.gov.dk/concept/core/nsis/Low">https://data.gov.dk/concept/core/nsis/Low</a></li> <li>• <a href="https://data.gov.dk/concept/core/nsis/Substantial">https://data.gov.dk/concept/core/nsis/Substantial</a></li> <li>• <a href="https://data.gov.dk/concept/core/nsis/High">https://data.gov.dk/concept/core/nsis/High</a></li> </ul>
aal	<p>Authenticator Assurance Level</p> <p>One of</p> <ul style="list-style-type: none"> <li>• <a href="https://data.gov.dk/concept/core/nsis/Low">https://data.gov.dk/concept/core/nsis/Low</a></li> <li>• <a href="https://data.gov.dk/concept/core/nsis/Substantial">https://data.gov.dk/concept/core/nsis/Substantial</a></li> <li>• <a href="https://data.gov.dk/concept/core/nsis/High">https://data.gov.dk/concept/core/nsis/High</a></li> </ul>
amr	<p>The list of authenticators used to achieve the resulting AAL/LoA.</p> <p>Possible values are:</p> <ul style="list-style-type: none"> <li>• mitid:password</li> <li>• mitid:code_token</li> <li>• mitid:code_reader</li> <li>• mitid:code_app</li> <li>• mitid:code_app_enhanced</li> <li>• mitid:u2f_token</li> </ul>

### Transaction token MitID specific claims

Claim value	Possible values
mitid.reference_text	Passthrough of the MitID <b>reference_text</b> identity provider parameter.
mitid.psd2	The mitid.psd2 claim is always issued as a transaction token MitID specific claim.
transaction_actions	<p>Type: string (single value) or JSON list</p> <p>Only set, if one or more of the following transaction actions were performed:</p> <ul style="list-style-type: none"> <li>• mitid.login (Login completed)</li> <li>• mitid_erhverv.identity_selected</li> </ul>

## References

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