DIGITALISERINGSSTYRELSEN

NemLog-in

Supporting services

Contents

References

Term	Reference
[IDWSREST]	OIO IDWS REST v.1.0 profile, available at https://www.digitaliser.dk/resource/3457606
[Manual]	"Brugermanual til NemLog-in administration v3.0." Available at https://www.nemlog-in.dk/media/kgeliazh/25-brugermanual-til-nemlog-in-administration-v-3.pdf
[DF]	Datafordeleren. https://datafordeler.dk/
[OIOSAML]	OIOSAML 3.0.2 profile, available at https://digst.dk/it-loesninger/nemlog-in/det-kommende-nemlog-in/vejledninger-og-standarder/oiosaml-302/
[OCES2]	https://digst.dk/it-loesninger/nemid/om-loesningen/oces-standarden/ Examples: https://www.nets.eu/dk-da/kundeservice/nemid-tjenesteudbyder/NemID-tjenesteudbyderpakken/Pages/OCES-II-certifikat-eksempler.aspx
[CERTGOV]	https://certifikat.gov.dk/

1 Introduction

This documentation package contains the technical information needed for accessing and querying the NemLog-in Supporting services for service providers.

This functionality is aimed at service providers with particular needs. Note, that the basic SAML functionality of NemLog-in does not require the use of these service, as these instead form a set of supporting services required to suit special needs for some service providers and/or NemLog-in Brokers.

The services described here are considered either Lookup services where one piece of information about an identity is exchanged for another or Match services. The match-services will, given two identifiers return true or false, indicating if the two identifies refer to the same person or employee identity, or not.

The services only consider person and employee identities.

1.1 Identifiers

The services described here use identifiers for digital identities, either persons or employees. The identifiers are briefly described in this section.

1.1.1 Certificate identifiers

In OCES2 certificates [OCES2], global identifiers (RID, PID, etc) for the identities holding certificates enter the certificate subject serial number.

In the new OCES and qualified certificates [CERTGOV], a similar convention is used to qualify the identifiers wrt.

- the identity type and
- the *persistence level* of the identifier

The *persistence level* of an identifier indicates the interpretation scope of the identifier.

Three types of identities are supported:

Identity type	Acronym	Description
Person	P	Physical person – usually Danish citizens
Employee	E	Employee identity – a person associated to an organization, and acting in this context.
Organization	O	Organization identity – the identity representing the organization itself.

The following persistence levels are used in NemLog-in:

Persistence level	Acronym	Description
Global	G	The identifiers described in the section “Global identifiers” below are used, i.e. Persistent Identifier in employee certificates and CPR UUID

Persistence level	Acronym	Description
		in person certificates. All actors receive the same global identifiers. The - now deprecated - PID and RID identifiers are also global.
Certificate	C	For long-term certificates, the user organization may choose to issue a new identifier for each user certificate. In this case, when the employee renews his/her certificate, a new identifier will be used.
Session	S	Identifiers specific to a given session. Every session (authentication or signing) will produce a new identifier, even if the same identity performs the authentication/signing.

Not all combinations of identity type and persistence level are supported for every kind of certificate.

The following combinations are possible in NemLog-in:

Identity type/Certificate type	Long term (OCES/qualified)	Short term (qualified)
Person	-	G/S
Employee	G/C	G/S
Organization	G	G

Note, that long term certificates are never issued to private identities as indicated by a '-' in the table. For example, short term qualified certificates issued to employees may contain either a global (G) or a session (S) specific identifier.

When UUID identifiers enter certificates, they become part of the subject serial number.

The subject serial number has the following syntax:

```
UI:DK-<identity type acronym>:<persistence level acronym>:<uuid>
```

Examples:

Employee certificate with session persistence level:

```
UI:DK-E:S:cdc78da8-c295-4693-bc69-da2d799bcb19
```

Employee (long term) certificate with certificate persistence level:

```
UI:DK-E:C:a33f79cd-42b2-4203-aa2d-e526157985ce
```

Organization certificate – persistence level is always global for these:

```
UI:DK-O:G:184c3849-7acd-4a76-98fd-4db60de9d7cc
```

1.1.2 User certificate identifiers

As evident from the table above, the UUID entering the subject serial number of user certificates (long term certificates issued to employees) is either global or certificate specific, at the user organization's discretion:

Persistence level	Description
Global (G)	Employee UUID: Identical to the Persistent Identifier in the OIOSAML profile. Each issued certificate will have the same subject serial number. Note, that this identifier does not coincide with the CPR UUID, even if a CPR UUID is registered for the employee identity.
Certificate (C)	A new UUID is generated and used in each issued user certificate. Each issued certificate (also renewed certificates) will have a different subject serial number.

Usually (and by default) user certificates are issued with the global identifier. But the user organization may select the more privacy friendly option and employ certificate specific identifiers.

1.1.3 Signer certificate identifiers

When the new NemLog-in signing solution is used, signer certificates are issued with the same serial number syntax as user certificates but with different semantics of the identifiers.

When the service provider initializes a signing process he must choose the persistence level of the issued signer certificate identifiers. As shown in the table above, two options are available:

Persistence level	Description
Global (G)	The service provider may select that global identifiers are to be used in the signer certificates. The result is that the Persistent Identifier (EmployeeUUID) is used in employee certificates and CPR UUID in person certificates.
Session (S)	The service provider may also decide that a new UUID is generated for each signer certificate issued, i.e. employing session specific identifiers. In that case, the service provider must use the UUID-Match service to confirm signer's identity.

1.1.4 Identifier correlation

Some of the described global identifiers for private and employee identities are used both in certificates and in OIOSAML assertions.

A summary of these relationships is shown in the table below.

Persistence level	Identity	SAML identifier	Certificate SSN prefix	IdM API identifier[1]
Global	Person	CPRUUID attribute	UI:DK-P:G:	N/A

[1] See https://migrering.nemlog-in.dk/nemlog-in-erhvervslosning/avanceret-setup/integration-med-idm/

Persistence level	Identity	SAML identifier	Certificate SSN prefix	IdM API identifier[1]
Global	Employee	Persistent Identifier attribute[2]	UI:DK-E:G:	EmployeeIdentity.UUID

For rows in the table above, where both a SAML use and certificate prefix is shown, a service provider may directly compare UUID's from SAML assertions with UUID's from certificates to determine if they refer to the same identity.

Note, that although it is possible to request session specific identifiers in SAML assertions (Transient Subject NameID) these cannot be matched with or correlated to session specific certificate identifiers. For this reason, the Transient Subject NameID identifiers are not mentioned in the table above.

EXAMPLE A:

A service provider authenticates a private person and asks for the CPR-UUID attribute.

The service provider also receives a document signed using NemLog-in signing service.

The certificate in the signed document has a subject serial number with prefix 'UI:DK-P:G'.

In that case the service provider may perform a string comparison of the CPR-UUID attribute and the UUID from the certificate: If they are identical, the signer is the same person as the authenticated user.

If, on the other hand, the ssn had a prefix 'UI:DK-P:S' the service provider would have to invoke the UUID-match-service (CprUUIDMatchesSigner, see section 3.3.4) to verify the signer's identity.

EXAMPLE B:

A service provider authenticates an employee and requests the Persistent Identifier attribute. The service provider also provides a webservice, where end users authenticate by presenting a signed ticket.

The ticket is signed using an OCES user certificate, with an SSN prefix 'UI:DK-E:G'.

The service provider may in that case directly determine, if the authenticated user is the same as the user signing the ticket by (string) comparing the Persistent Identifier attribute value with the UUID in the certificate used for signing the ticket.

If, on the other hand, the certificate ssn had prefix 'UI:DK-E:C' the service provider would have to invoke the UUID-match-service (PersistentIdentifierMatchesSigner, see section 3.3.3) to verify the signer's identity.

EXAMPLE C:

A user organisation uses the IdM API to create an employee identity. The *POST /api/administration/identity/employee* method returns the EmployeeIdentity.UUID in the response when creation is successful.

When the identity has been activated, the EmployeeIdentity.UUID is returned in the Persistent Identifier attribute to any public or private service provider requesting that attribute, when an authentication is performed for that identity.

[2] The OIOSAML 3 attribute: https://data.gov.dk/model/core/eid/professional/uuid/persistent

2 Lookup services

The lookup services provide specific methods for gathering information about private and employee identities. Access to these services is limited and the service provider must have a legitimate reason for requesting access. Some services are restricted for use by public service providers exclusively.

There are 6 service endpoints:

Endpoint	Access for public service providers only
PID-CPR	Yes
CPR-PID	No
RID-CPR	Yes
SubjectSerialNumber-RID	No
SubjectSerialNumber-CPRUUID	No
SubjectSerialNumber-CPR	Yes

The service endpoints are described invidually below.

2.1 Requesting access

Prior to accessing the lookup services in production, the service provider must register as Web Service Consumer (WSC) system-user in NemLog-in (https://administration.nemlog-in.dk) and apply for access to the NemLog-in Lookup Service.

This is done by adding an organization- or systemcertificate (VOCES or FOCES), one for production and one for integration test as shown above. In the betatest-environment, you can add the same (test) certificate for both production and integration test.

When the system user is registered, you must request access to the lookup service and apply for the relevant privilege/privileges. Each service endpoint has its own privilege. When requesting access in production you must justify your request by explaining your reason for requesting access to the service.

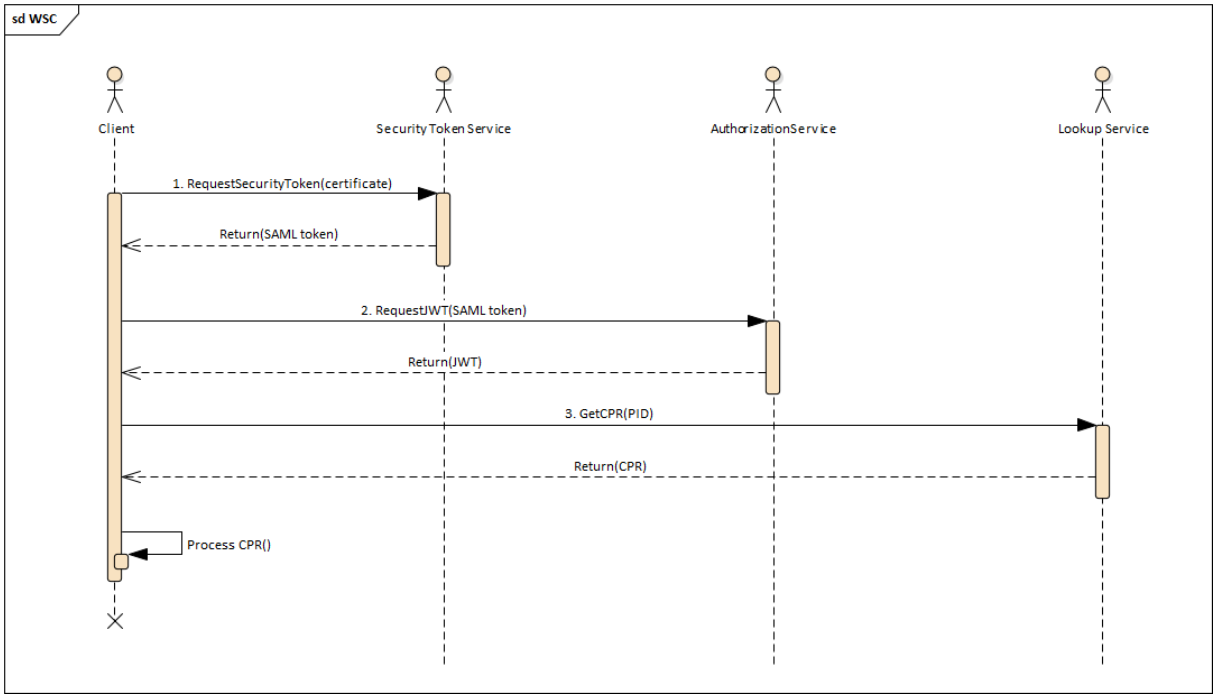
The steps are described in detail in [Manual], sections 7.22 and 7.27.

2.2 Authentication and authorization

When the WSC is registered in NemLog-in, and access is granted to the requested methods, the client is ready to invoke service methods.

Access to service methods is obtained according to the OIO IDWS REST profile [IDWSREST] where SAML assertions are used as Bearer tokens.

The process is shown below.



1. The client (WSC) invokes the STS using the WS-Trust protocol. The WSC certificate is used to sign the request. The STS returns a SAML assertion which contains the granted privileges. We refer to Appendix A – endpoints for Lookup Services for details.
2. The client invokes the Authorization Service endpoint. Here the SAML assertion is validated, and an access token (JWT) is issued. The JWT token is signed - this also holds the list of privileges granted to the WSC.
3. The issued JWT token is passed as bearer token for accessing the different business methods in the API.

The JWT token response (step 2) takes the following form:

```
{
  "accessToken":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJFSUE6UHJpdmlsZWdlcyI6Intc...",
  "tokenType": "Bearer",
  "expiresIn": 1800
}
```

The JWT Access Token is provided in accessToken and expiresIn tells us that the token will expire in 1800 seconds. Note, that this value is provided merely as a hint on how long the client should attempt to use the token. Use this value to determine how long the JWT should be cached.

2.2.1 Recommended usage

For performance reasons, it is recommended that clients reuse the provided Access Token throughout the token lifetime.

To prevent complicated expiration detection it is recommended that clients cache the Access Token for a period somewhat shorter that the token lifetime, e.g. for 90% of the token lifetime provided in the `expiresIn´ attribute above.

2.3 Service interface

2.3.1 PID-CPR

Scope: Method can be used for private identities only.

Path: /api/lookup/pidcpr

Method: POST

Request parameters:

- *string pid*: The PID for which a CPR is requested. E.g. 9208-2002-2-130462414956.

Response parameters:

- *string cpr*: The CPR number, if a CPR is registered for the PID, null otherwise.

2.3.2 CPR-PID

Scope: Method can be used for private identities only.

Path: /api/lookup/cprpid

Method: POST

Request parameters:

- *string cpr*: The 10-digit CPR number for which a PID is requested

Response parameters:

- *string pid*: The PID e.g. 9208-2002-2-130462414956 , if a PID is registered for the CPR, null otherwise.

This endpoint is not available yet.

2.3.3 RID-CPR

Scope: Method can be used for employee identities only.

Path: /api/lookup/ridcpr

Method: POST

Request parameters:

- *string cvr*: 8-digit CVR
- *string rid*: The RID for which a CPR is requested.

Response parameters:

- *string cpr: 10-digit* CPR registered for requested {CVR, RID}, if present.

2.3.4 SubjectSerialNumber-RID

Scope: Method can be used for employee identities only.

Path: /api/lookup/subjectserialnumberrid

Method: POST

Request parameters:

- *string subjectSerialNumber*: Subject serial number including UUID from a user certificate (OCES or qualified) for which a RID is requested. The subject serial number is part of the certificate subject distinguished name.
  The subject serial number has the following syntax:
  `UI:DK-E:<qualifier>:4da9c339-a2c0-47cb-b26d-2419da6e04dc`
  where <qualifier> is either "C" og "G".

Response parameters:

- *string rid:* RID registered in NemLog-in for requested UUID, if present. RID format varies, but is usually 8 digits, e.g. "19822376"

This endpoint is not available yet.

2.3.5 SubjectSerialNumber-CPRUUID

Scope: Method can be used for employee identities only.

Path: /api/lookup/subjectserialnumbercpruuid

Method: POST

Request parameters:

- *string subjectSerialNumber*: Subject serial number including UUID from a user certificate (OCES or qualified) for which a CPR UUID is requested. The subject serial number is part of the certificate's subject distinguished name.
  The subject serial number has the following syntax:
  `UI:DK-E:<qualifier>:4da9c339-a2c0-47cb-b26d-2419da6e04dc`
  where <qualifier> is either "C" og "G".

  Requests using global qualifier will return CPRUUID even if no certificate has been issued for the identity.

Response parameters:

- *string cpruuid:* 36-character CPR UUID registered in NemLog-in for requested employee, if available.

This endpoint is not available yet.

2.3.6 SubjectSerialNumber-CPR

Scope: Method can be used for employee identities only.

Path: /api/lookup/subjectserialnumbercpr

Method: POST

Request parameters:

- *string subjectSerialNumber*: Subject serial number including UUID from a user certificate (OCES or qualified) for which a CPR is requested. The subject serial number is part of the certificate's subject distinguished name.
  The subject serial number has the following syntax:
  `UI:DK-E:<qualifier>:4da9c339-a2c0-47cb-b26d-2419da6e04dc`
  where <qualifier> is either "C" og "G".

  Requests using global qualifier will return CPR even if no certificate has been issued for the identity.

Response parameters:

- *string cpr:* 10-digit CPR registered in NemLog-in for requested employee, if available.

This endpoint is not available yet.

3 UUID-Match service

The UUID-Match service consists of the following endpoints:

- SubjectMatchesSigner
- SubjectMatchesCertificate
- PersistentIdentifierMatchesSigner
- CprUUIDMatchesSigner

The endpoints are described in detail below.

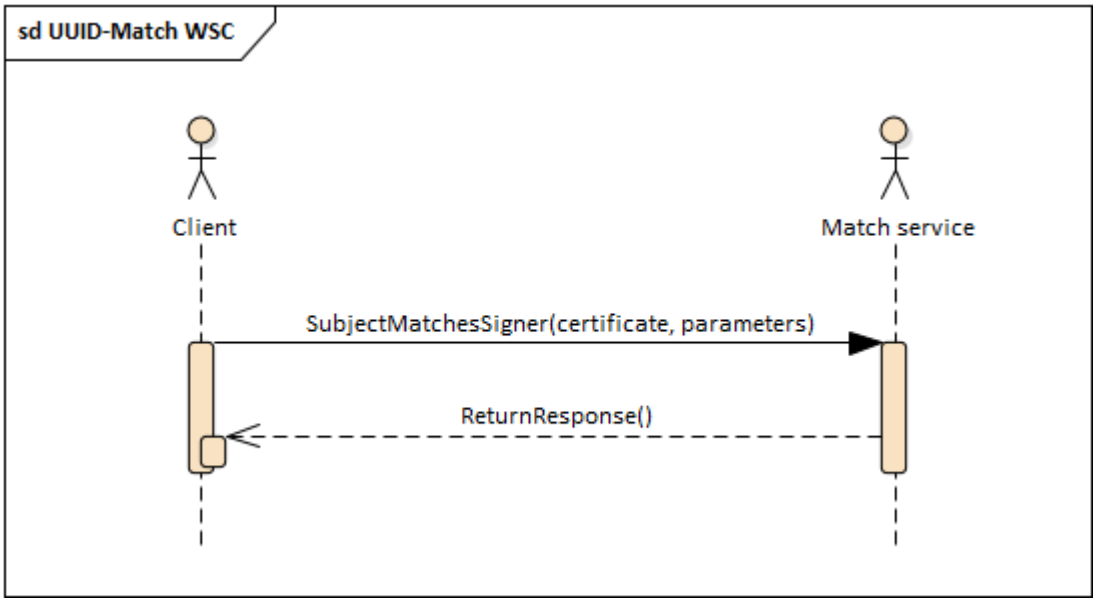
3.1 Requesting access

When a service provider is connected to NemLog-in the service provider must decide the type of service to connect. See [Manual] section 6.4. If the connected system makes use of the new signing service – either as a 'Web SSO' system (select 'Web SSO' type and check 'Kvalificeret signeringstjeneste') or by direct, exclusive connection to the signing service (select 'Kvalificeret signeringstjeneste' radio button) – the UUID-Match service will automatically be available.

Note, that public service providers are also allowed to select 'Signeringstjeneste' in the user interface, selecting this (older) service will not provide access to the UUID-Match service.

3.2 Authentication and authorization

The client must authenticate using one of the signer certificates associated to the connected service as TLS client certificate. The endpoints are used directly, as depicted below.



3.3 Service interface

3.3.1 SubjectMatchesSigner

Description: When a signed document produced by the new NemLog-in Signing Service is received, the signer certificate contains a UUID identifier for signer in its subject serial number.

This service will allow the recipient of such a document to verify if the signer is the same private or employee identity identified by a previous authentication in NemLog-in.

The endpoint thus supports the common scenario, where an authenticated user signs a document.

Scope: Endpoint be used for both person and employee identities.

Path: /api/uuidmatch/subjectMatchesSigner

Method: POST

Request parameters:

- *string subjectNameID*: Persistent Subject NameID value received in SAML assertion according to [OIOSAML] for either person or professional (employee). For example: `https://data.gov.dk/model/core/eid/person/uuid/123e4567-e89b-12d3-a456-426655440000`
- *string signerSubjectSerialNumber:* Subject serial number (SSN) from qualified certificate received as part of a qualified signature. The UUID must be session specific (SSN prefix 'UI:DK-P:S' or 'UI:DK-E:S'), otherwise an error message is returned.
  The full serial number string must be passed, for example:
  `UI:DK-P:S:4da9c339-a2c0-47cb-b26d-2419da6e04dc`
- *string serviceProviderEntityID:* EntityID for the service that requests the match operation and for which the Subject NameID was issued. The EntityID must be that registered for the service in NemLog-in Administration.

Response parameters:

- *enum status:* Enumeration of possible results:
  - *Match* – returned if subject matches signer
  - *NoMatch* – returned if the valid input does not refer the same identity
  - *SubjectNotFound* – returned if passed Subject NameID could not be found
  - *SerialNotFound* – returned if passed subject serial number could not be found.

3.3.2 SubjectMatchesCertificate

Description: This service will allow the recipient of a signature or authentication performed with a private key associated to a user certificate (OCES or qualified) to verify if the holder of that certificate is the same employee identified by authentication in NemLog-in.

Qualified long term certificates can be distinguished from the qualified short term certificates used by NemLog-in signing service by inspecting the time span between NotAfter and NotBefore timestamps in the certificates. If this time span is longer than one week (7*24 hours), the certificate is a long term certificate.

Scope: Endpoint can only be used for employee identities.

Path: /api/uuidmatch/subjectMatchesCertificate

Method: POST

Request parameters:

- *string subjectNameID*: Persistent Subject NameID value received in SAML assertion according to [OIOSAML] for professional (employee). For example: `https://data.gov.dk/model/core/eid/professional/uuid/123e4567-e89b-12d3-a456-426655440000`

- *string subjectSerialNumber:* Subject serial number from long-term user (employee) certificate (either OCES or qualified). The full serial number string must be passed, for example:
  `UI:DK-E:G:4da9c339-a2c0-47cb-b26d-2419da6e04dc`
- *string serviceProviderEntityID:* EntityID for the service that requests the match operation and for which the Subject NameID was issued. The EntityID must be that registered for the service in NemLog-in Administration.

Response parameters:

- *enum status:* Enumeration of possible results:
  - *Match* – returned if subject matches signer
  - *NoMatch* – returned if the valid input does not refer the same identity
  - *SubjectNotFound* – returned if passed Subject NameID could not be found
  - *SerialNotFound* – returned if passed subject serial number could not be found.

3.3.3 PersistentIdentifierMatchesSigner

Scope: Endpoint can only be used for employee identities.

Path: /api/uuidmatch/persistentIdentifierMatchesSigner

Method: POST

Request parameters:

- *string persistentIdentifier*: Persistent Identifier attribute value received in SAML assertion according to [OIOSAML] for professional (employee). For example:
  `urn:uuid:323e4567-e89b-12d3-a456-426655440000`
- *string signerSubjectSerialNumber:* Subject serial number from qualified certificate received as part of a qualified signature. The embedded UUID must be session (S) specific, otherwise an error message is returned. The full serial number string must be passed, for example:
  `UI:DK-E:S:4da9c339-a2c0-47cb-b26d-2419da6e04dc`
- *string serviceProviderEntityID:* EntityID for the service that requests the match operation and for which the Subject NameID was issued. The EntityID must be that registered for the service in NemLog-in Administration.

Response parameters:

- *enum status:* Enumeration of possible results:
  - *Match* – returned if subject matches signer
  - *NoMatch* – returned if the valid input does not refer the same identity
  - *PersistentIdentifierNotFound* – returned if passed Persistent Identifier could not be found
  - *SerialNotFound* – returned if passed subject serial number could not be found or is not session specific.

3.3.4 CprUUIDMatchesSigner

Description: This service allows caller to verify if a document signer, identified by the signing certificate subject serial number, possesses a given CPR UUID.

Scope: Endpoint be used for both person and employee identities.

Path: /api/uuidmatch/cpruuidmatchessigner

Method: POST

Request parameters:

- *string cprUUID*: CPR UUID attribute value received in SAML assertion according to [OIOSAML] for person or professional (employee). For example: `urn:uuid:423e4567-e01b-12d3-a456-426655444321`
- *string signerSubjectSerialNumber:* Subject serial number from qualified certificate received as part of a qualified signature. The embedded UUID must be session specific, otherwise an error status is returned. The full serial number string must be passed, for example: `UI:DK-P:S:4da9c339-a2c0-47cb-b26d-2419da6e04dc`
- *string serviceProviderEntityID:* EntityID for the service that requests the match operation and for which the Subject NameID was issued. The EntityID must be that registered for the service in NemLog-in Administration.

Response parameters:

- *enum status:* Enumeration of possible results:
  - *Match* – returned if subject matches signer
  - *NoMatch* – returned if the valid input does not refer the same identity
  - *CprUuidNotFound* – returned if passed CPR UUID could not be found
  - *SerialNotFound* – returned if passed subject serial number could not be found or is not session specific.

3.3.5 CPRMatchesSigner

Description: This service allows caller to verify if a document signer, identified by the signing certificate subject serial number, possesses a given CPR number.

Scope: Endpoint can be used for both person and employee identities.

Path: /api/uuidmatch/cprmatchessigner

Method: POST

Request parameters:

- *string signerSubjectSerialNumber:* Subject serial number (SSN) from certificate re-ceived as part of a signed document. The subject serial number must be session specific or global and must belong to a person or an employee.
  The full serial number string must be passed, for example:
  `UI:DK-P:S:4da9c339-a2c0-47cb-b26d-2419da6e04dc`
- *string serviceProviderEntityID*: EntityID for the service that requests the match operation. The EntityID must be that registered for the service in NemLog-in Administration.
- *string cpr:* CPR number to be matched.

Response parameters:

- *enum status*: Enumeration of possible results:
  - *Match* – returned if subject matches CPR
  - *NoMatch* – returned if the subject does not match the given CPR

This endpoint is not available yet.

4 Appendix A – endpoints for Lookup Services

4.1 STS and Authorization Service endpoint

4.1.1 Beta-test environment

The NemLog-in STS service in devtest4 (integrationtest part) is located at

- `https://securetokenservice.test-devtest4-nemlog-in.dk/SecurityTokenService.svc`

The NemLog-in STS service in devtest4 (production part) is located at

- `https://securetokenservice.devtest4-nemlog-in.dk/SecurityTokenService.svc`

The Lookup Service Authorization Service endpoint is located at

- `https://lookupservice.test-devtest4-nemlog-in.dk/api/accesstoken/issue`
- `https://lookupservice.devtest4-nemlog-in.dk/api/accesstoken/issue`

4.1.2 Integrationtest environment

The NemLog-in STS service is located at

- `https://securetokenservice.test-nemlog-in.dk/SecurityTokenService.svc`

The Lookup Service Authorization Service endpoint is located at

- `https://lookupservice.test-nemlog-in.dk/api/accesstoken/issue`

4.1.3 Production environment

The NemLog-in STS service is located at

- `https://securetokenservice.nemlog-in.dk/SecurityTokenService.svc`

The Lookup Service Authorization Service endpoint is located at

- `https://lookupservice.nemlog-in.dk/api/accesstoken/issue`

4.2 Service endpoints

The service endpoints are generally available at

- `https://lookupservice.<hostname>/<path>`

For the beta-test environment (devtest4), the production and integrationtest part of the PID-CPR lookup are available at

- `https://lookupservice.devtest4-nemlog-in.dk/api/lookup/pidcpr`
- `https://lookupservice.test-devtest4-nemlog-in.dk/api/lookup/pidcpr`

respectively.

5 Appendix B – endpoints for UUID-Match service

5.1 Beta-test environment

The Swagger endpoint is available at

- `https://services.test-devtest4-nemlog-in.dk/swagger/ui/index#/UuidMatchService`

The YAML service definition for UUID-Match can be retrieved at

- `https://services.test-devtest4-nemlog-in.dk/docs/swagger-broker.yaml`

5.2 Integrationstest environment

The UUID-Match service is located at

- `https://services.test-nemlog-in.dk/`

The YAML service definition for the UUID-Match is not exposed by the environment. Use the YAML service definition from the beta-test environment.

5.3 Production environment

The UUID-Match service is located at

- https://services.nemlog-in.dk/

The YAML service definition for the UUID-Match is not exposed by the environment. Use the YAML service definition from the beta-test environment.