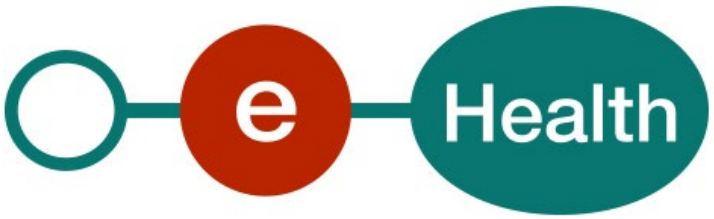


Identity & Authorization Management (IAM)
SSO from fat to thin client
Technical specifications
Version 1.3

This document is provided to you, free of charge, by the

eHealth platform

Willebroekkaai 38 – 1000 Brussel

38, Quai de Willebroeck – 1000 Bruxelles

Table of contents

To the attention of: “IT expert” willing to integrate this web service.

1. Document management

1.1 Document history

Version	Date	Author	Description of changes / remarks
1.0	30/04/2015	eHealth platform	Initial version
1.1	30/10/2018	eHealth platform	Update linked to major release R2018.2
1.2	06/04/2022	eHealth platform	Review IDP screens (new look and feel)
1.3	15/07/2024	eHealth platform	SSO to IAMConnect JWT

2. Introduction

2.1 Context

A user (physical person), in possession of a sessionToken received from the eHealth SecureTokenService (STS), needs to continue his work in a remote web application / client that requires authentication with the eHealth platform. Therefore, he needs to identify himself at the eHealth IdentityProvider (IDP) or eHealth IAM Connect. As the user already has a sessionToken from the STS, he can reuse it to authenticate at the IDP or IAM Connect.

2.2 Goal of the document

This document describes in detail how to use a secure token, delivered by the STS, as identity proof to start a web browser Single Sign on Session.

It does not contain details on how to obtain a secure token from the STS in the first place. For this information, please refer to the cookbook STS available on the portal of the eHealth platform.
(***https://www.ehealth.fgov.be/ehealthplatform/nl/service-iam-identity-access-management***)

Important note : this approach is not available for secure token delivered by STS with eHealth institution/organization certificates.

2.3 eHealth platform document references

All the document references can be found on the portal of the eHealth platform[1]. These versions or any following versions can be used for the services of the eHealth platform.

ID	Title	Version	Date	Author
1	IAM Connect - Mobile integration - Technical specifications	1.9	29/05/2024	eHealth platform
2	SOA – Error guide	1.0	10/06/2021	eHealth platform
3	Secure Token Service - WS Trust	1.1	12/01/2024	eHealth platform

2.4 External document references

All documents can be found on the internet. They are available to the public, but not supported by the eHealth platform.

ID	Title	Source	Date	Author
1	JSON Web Token (JWT)	**https://tools.ietf.org/html/rfc7519**	May 2015	M. Jones (Microsoft) J. Bradley (Ping Identity) N. Sakimura (NRI)

[1] *Https://www.ehealth.fgov.be/ehealthplatform*

3. Support

3.1 Helpdesk eHealth platform

3.1.1 Certificates

To access the secured eHealth platform environment you have to obtain an eHealth platform certificate which is used to identify the initiator of the request. In case you do not have one, please consult the chapter about the eHealth Certificates on the portal of the eHealth platform.

- ***https://www.ehealth.fgov.be/ehealthplatform/nl/ehealth-certificaten***
- ***https://www.ehealth.fgov.be/ehealthplatform/fr/certificats-ehealth***

For technical issues regarding eHealth platform certificates

- Acceptance: ***acceptance-certificates@ehealth.fgov.be***
- Production: ***support@ehealth.fgov.be***

3.1.2 For issues in production

eHealth platform contact centre:

- Phone: 02 788 51 55 (on working days from 7 am till 8 pm)
- Mail: ***support@ehealth.fgov.be***
- *Contact Form :*
  - ***https://www.ehealth.fgov.be/ehealthplatform/nl/contact*** (Dutch)
  - ***https://www.ehealth.fgov.be/ehealthplatform/fr/contact*** (French)

3.1.3 For issues in acceptance

Integration-support@ehealth.fgov.be

3.1.4 For business issues

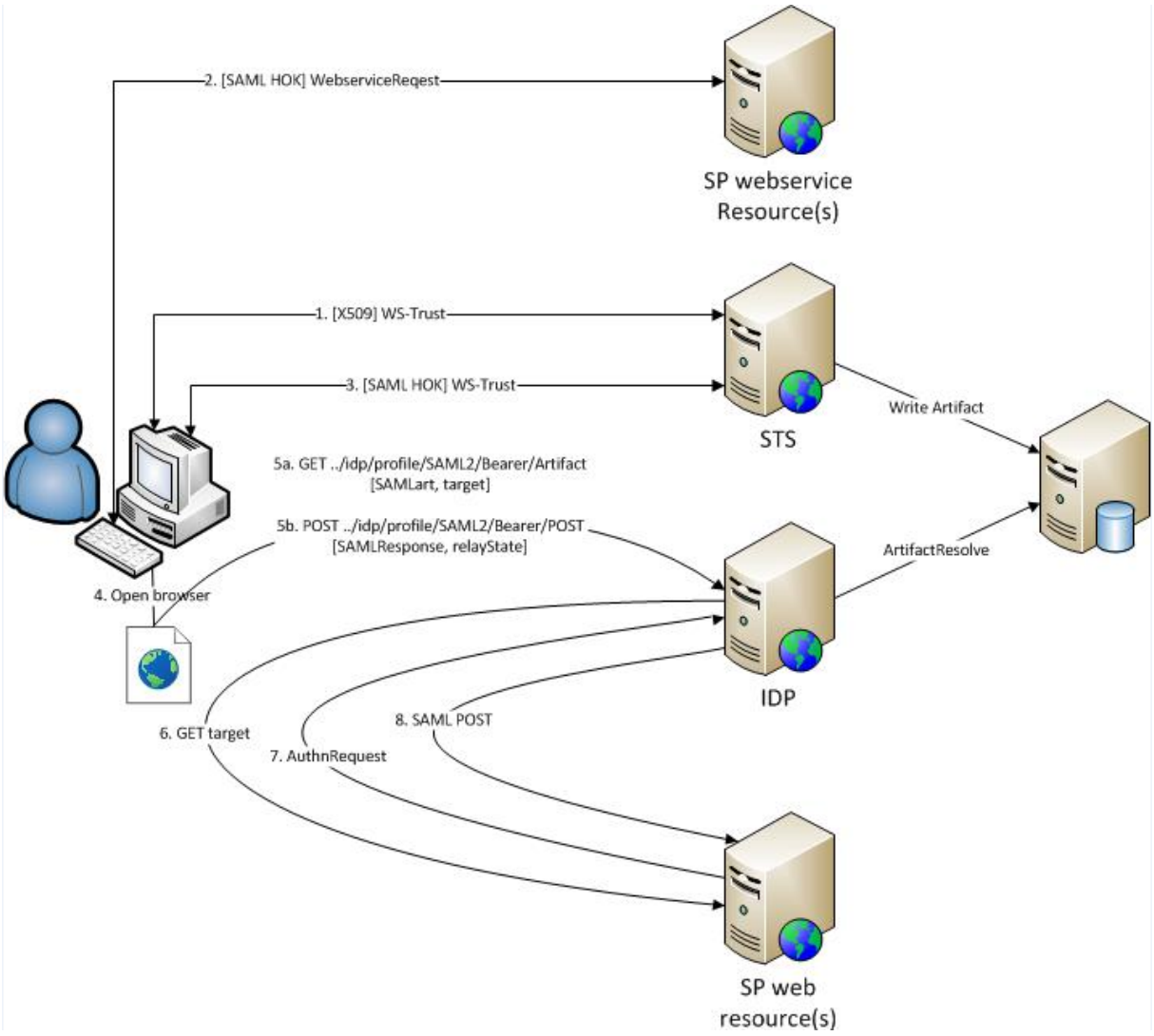
- regarding an existing project: the project manager in charge of the application or service
- regarding a new project or other business issues: ***info@ehealth.fgov.be***For issues in production

3.2 Status

The website ***https://status.ehealth.fgov.be*** is the monitoring and information tool for the ICT functioning of the eHealth services that are partners of the Belgian eHealth system.

4. Global overview

4.1 STS to IDP

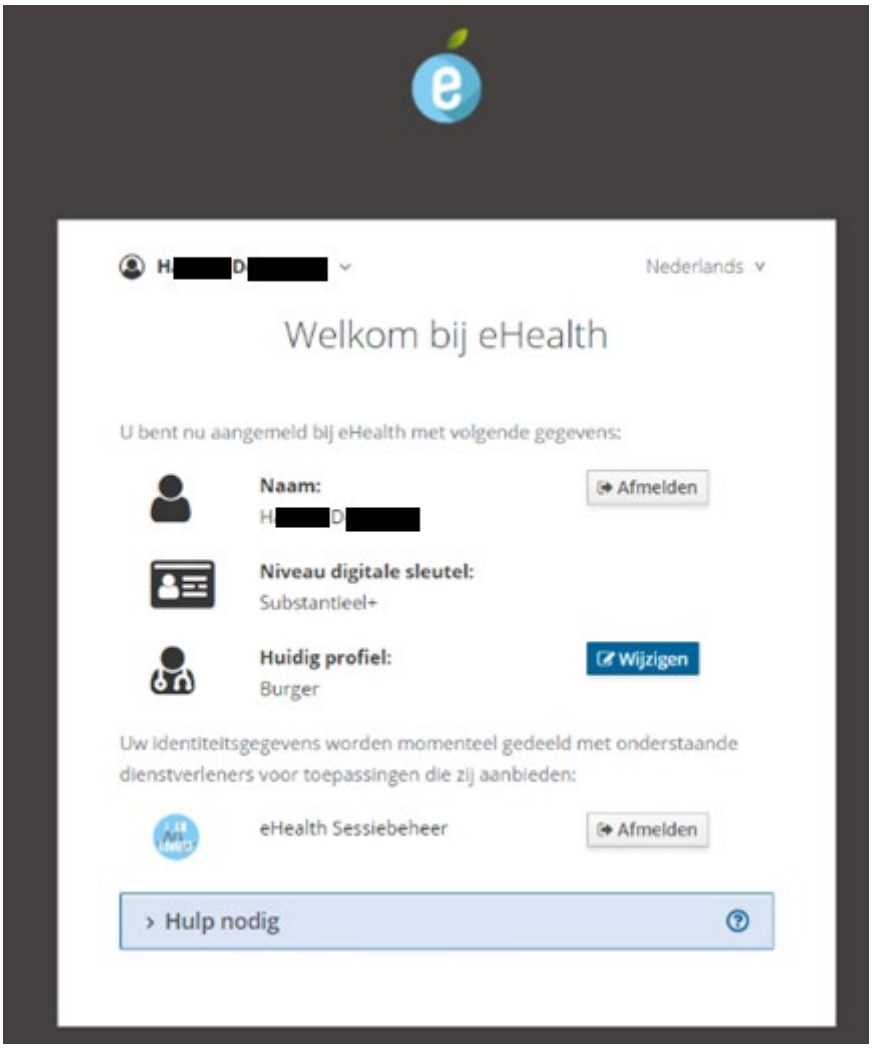


Note

All Request/Response messages are sent over SSL/TLS.

*The client application **must** verify that direct communication with the eHealth Platform is setup, by verifying the certificate used by eHealth server in the SSL/TLS handshake.*

After validation of the Bearer Assertion, the IDP constructs the user's profile based on the contents of the Assertion and activates a session on his behalf.



As of this moment, the user will be able to access web applications in the eHealth IAM Federation without the need to authenticate again.

If an HTTPRequest parameter 'RelayState' is present and points to a trusted location, the browser will not show the above page and automatically redirect the user to the given location.

Before the IDP will actually send authentication information on behalf of the user to a requested application, a notification will be shown - only once - on the 'confirm profile' page so the user can acknowledge that a browser session was started with his identity.

After confirmation, he is sent to the requested application (this is part of the normal process flow of browser SSO).

For more information, go to section 5 of this document.

4.2 STS to IAM Connect

This flow describes how to exchange a SAML HOK token, issued by IAM STS, to a jwt in IAM Connect.

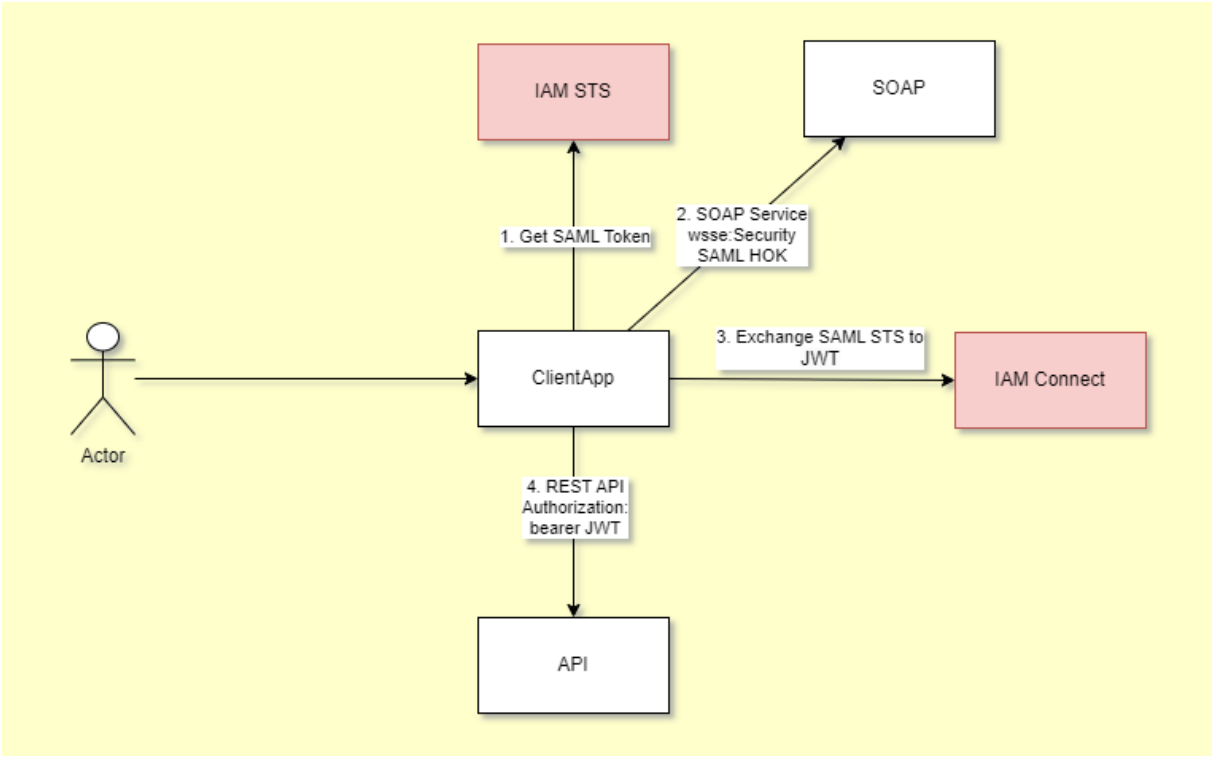
This can also be used to start a web session, based on a SOAP session (see section 4.2.2).

Exchanging tokens from eHealth IAM STS is allowed and supported with following limitations :

1. A client needs to be granted permission to exchange tokens from the given external provider
2. The end user (actor) to which the external token was issued must be registered in the IAM Connect realm (for any ClientApp).

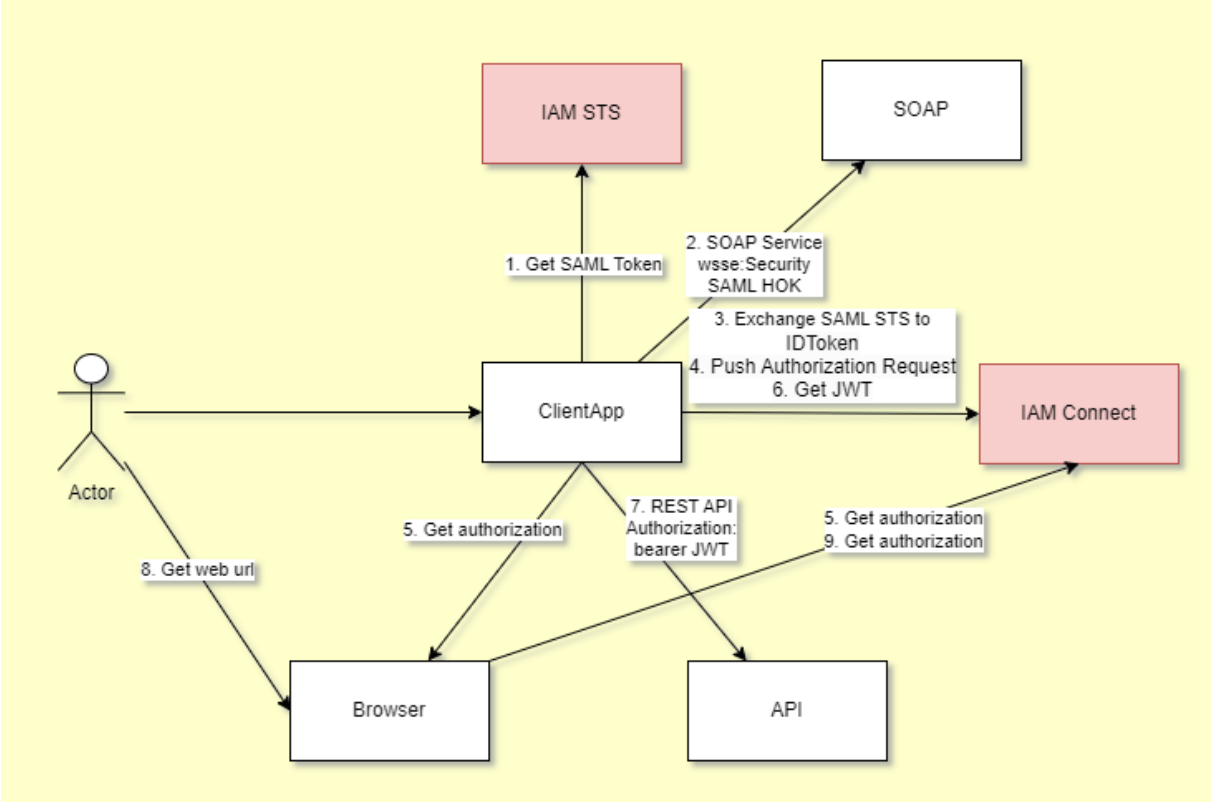
Before reading this section, please ensure that you have read IAM Mobile integration Technical specifications.

4.2.1 SAML HOK Login REST



For more information, go to section 6 of this document.

4.2.2 SAML HOK Login WEB



For more information, go to section 7 of this document.

5. Step-by-step for IDP

5.1 Steps 1-2: SSO web services (STS)

SSO is setup between WS hosted by the eHealth Platform or one of its trusted partners with eHealth IAM STS as Security Token Service.

5.2 Steps 3-5: SSO Web services – Web App (STS → IDP)

SSO session for WS is transferred from fat to thin client.

5.2.1 Request Bearer Token from STS

The client sends a request to the STS to generate a new bearer token, based on the existing holder-of-key (HOK) token. The newly generated token will be valid for a short period (max 10 minutes).

There are multiple options to request this token which will influence the result and remainder of the process flow.

See section 'Detailed solution steps' for details.

5.2.2 Open browser

A new or existing browser window is opened to setup SSO between the client's browser as thin client and eHealth IDP.

5.2.3 Authenticate at IDP

Depending on the result of step 3, the client uses his browser to send his authentication details to the eHealth IDP.

5.3 Steps 6-8: SSO Web App (IDP)

SSO is setup between web applications in the eHealth IAM Federation with eHealth IAM IDP as Identity Provider.

5.4 Detailed solution steps (Web services – Web App)

The eHealth platform offers two solutions to setup SSO from a fat to a thin client. Clients are free to implement one or the other.

5.4.1 Solution: GET Artifact

Using a fat client application and an active HOK Assertion (received from the STS at the start of the web service session), the user requests a reference to a Bearer Assertion in return. He will need to get the reference to the eHealth IDP to get authenticated.

5.4.1.1 Step 3: Request Bearer Token from STS

The user sends a request to eHealth while using an active HOK Assertion as secure authentication token.

5.4.1.1.1 Endpoint

- *PROD*: ***https://services.ehealth.fgov.be/IAM/SingleSignOnService/v1***
- *ACC*: ***https://services-acpt.ehealth.fgov.be/IAM/SingleSignOnService/v1***
- *INT*: ***https://services-int.ehealth.fgov.be/IAM/SingleSignOnService/v1***

5.4.1.1.2 Inbound

- Header

WS-Security 1.1 SOAPHeader with following elements:

- *Timestamp*
- *SAML 1.1 Assertion*: active STS SAMLToken
- *Signature*: on timestamp and assertion

This is the same WS-SecurityPolicy as the one used in all eHealth Services protected with a samlToken.

- Body

RequestSecurityToken (WS-Trust 1.3) with following info:

- *TokenType*: ***http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV2.0***
- *RequestType*: ***http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue***
- *KeyType*: ***http://docs.oasis-open.org/ws-sx/wstrust/200512/Bearer***
- *AppliesTo*: eHealth IDP endpoint that will act as the resolver of the referenced Artifact ([ehealth environment]/idp/profile/SAML2/Bearer/Artifact)
  - *PROD*: ***https://www.ehealth.fgov.be/idp/profile/SAML2/Bearer/Artifact***
  - *ACC*: ***https://wwwacc.ehealth.fgov.be/idp/profile/SAML2/Bearer/Artifact***
  - *INT*: ***https://wwwint.ehealth.fgov.be/idp/profile/SAML2/Bearer/Artifact***

- Example (extract for readability)

```
<soapenv:Envelope xmlns:ns="http://docs.oasis-open.org/ws-sx/ws-trust/200512" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
            <Assertion AssertionID=" 477052849c6a7fbe6fd12540b761ad5a" IssueInstant="2018-10-19T14:17:27.923Z" Issuer="urn:be:fgov:ehealth:sts:1_0" MajorVersion="1" Min
            <ds:Signature Id="SIG-D411B7B5889E507A3D153995865027532" xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                <ds:SignedInfo>
                    <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                    <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
                    <ds:Reference URI="#id-D411B7B5889E507A3D153995865027531">
                    <ds:Reference URI="#TS-D411B7B5889E507A3D153995865027128">
                </ds:SignedInfo>
                <ds:SignatureValue>QtdB516KaxcPasp6bxJDBiExtdBBxggMqC23CIxdVyN4QJksNjhxge8P+gDp1j1QhHdFwMOuHeef
                <ds:KeyInfo Id="KI-D411B7B5889E507A3D153995865027529">
            </ds:Signature>
            <wsu:Timestamp wsu:Id="TS-D411B7B5889E507A3D153995865027128">
        </wsse:Security>
    </soapenv:Header>
    <soapenv:Body wsu:Id="id-D411B7B5889E507A3D153995865027531" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
        <wst:RequestSecurityToken Context="RC-3ea9c474-9539-4d0f-b0af-ea282fec1cb9" xmlns:auth="http://docs.oasis-open.org/wsfed/authorization/200706" xmlns:ds="http://
            <wst:TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV2.0</wst:TokenType>
            <wst:RequestType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue</wst:RequestType>
            <wst:KeyType>http://docs.oasis-open.org/ws-sx/wstrust/200512/Bearer</wst:KeyType>
            <wsp:AppliesTo>
                <wsa:EndpointReference>
                    <wsa:Address>https://wwwacc.ehealth.fgov.be/idp/profile/SAML2/Bearer/Artifact</wsa:Address>
                </wsa:EndpointReference>
            </wsp:AppliesTo>
        </wst:RequestSecurityToken>
    </soapenv:Body>
</soapenv:Envelope>
```

Figure 1: Example SSO Artifact Request

5.4.1.1.3 Outbound

- Body

  RequestSecurityToken Response(WS-Trust 1.3) containing a RequestedUnattachedReference with following info:

  - *Reference*: URL that can be used in a browser to resolve the artifact.

- Example

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <wst:RequestSecurityTokenResponse Context="RC-3ea9c474-9539-4d0f-b0af-ea282fec1cb9" xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
            <wst:RequestedUnattachedReference>
                <wsse:SecurityTokenReference xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
                    <wsse:Reference URI="https://wwwacc.ehealth.fgov.be/idp/profile/SAML2/Bearer/Artifact?SAMLart=AAQABPL...3A" ValueType="http://docs.oa
                </wsse:SecurityTokenReference>
            </wst:RequestedUnattachedReference>
        </wst:RequestSecurityTokenResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Figure 2: Example SSO Artifact Response

5.4.1.2 Step 4-5: Open browser and authenticate at IDP

Note

The artifact is a hard-to-forge, single-use, short-term reference.

This means the user has a limited time frame (a few minutes) to use it. If he attempts to use it a second time, an error will be generated.

The client application opens the user's favorite browser with the URL from the reference received from the STS.

An optional HTTPRequest parameter 'RelayState' can be added to let the IDP redirect to a protected web application if authentication succeeds. To enable SSO from the STS token to web applications on a mobile device, it could also be shown as a QR-code. Since the reference is a full URL, the mobile device will automatically open up a browser and navigate to the URL.

- Example

5.4.2 Solution: POST Assertion

Using a fat client application and an active HOK Assertion (received from the STS at the start of the WS session), the user requests a Bearer Assertion in return. He will need to POST this Assertion to the eHealth IDP to get authenticated.

5.4.2.1 Step 3: Request Bearer Token from STS

The user sends a request to eHealth while using an active HOK Assertion as secure authentication token.

5.4.2.1.1 Endpoint

- *PROD*: ***https://services.ehealth.fgov.be/IAM/SingleSignOnService/v1***
- *ACC*: ***https://services-acpt.ehealth.fgov.be/IAM/SingleSignOnService/v1***
- *INT*: ***https://services-int.ehealth.fgov.be/IAM/SingleSignOnService/v1***

5.4.2.1.2 Inbound

- Header

  WS-Security 1.1 SOAPHeader with following elements:

  - *Timestamp*
  - *SAML 1.1 Assertion: active STS SAMLToken*
  - *Signature: on timestamp and assertion*

This is the same WS-SecurityPolicy as used in all eHealth Services protected with a samlToken.

- Body

  RequestSecurityToken (WS-Trust 1.3) with following info:

  - *TokenType:* ***http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV2.0***
  - *RequestType:* ***http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue***
  - *KeyType:* ***http://docs.oasis-open.org/ws-sx/wstrust/200512/Bearer***
  - *AppliesTo: eHealth IDP endpoint that will act as the consumer of the requested Assertion ([ehealth environment]/idp/profile/SAML2/Bearer/POST)*
    - *PROD*: ***https://www.ehealth.fgov.be/idp/profile/SAML2/Bearer/POST***
    - *ACC*: ***https://wwwacc.ehealth.fgov.be/idp/profile/SAML2/Bearer/POST***
    - *INT*: ***https://wwwint.ehealth.fgov.be/idp/profile/SAML2/Bearer/POST***

Example (extract for readability)

```
<soapenv:Envelope xmlns:ns="http://docs.oasis-open.org/ws-sx/ws-trust/200512" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs
            <Assertion AssertionID=" 477052849c6a7fbe6fd12540b761ad5a" IssueInstant="2018-10-19T14:17:27.923Z" Issuer="urn:be:fgov:ehealth:s
            <ds:Signature Id="SIG-D411B7B5889E507A3D153995889619137" xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                <ds:SignedInfo>
                <ds:SignatureValue>dWm0pOuNxdgvRmTpptI7fz0bQKviCdDhnvJ3QhT7jlMV+F2IOAQSpWpqrKrjkE8XWohLLqverCDg
                <ds:KeyInfo Id="KI-D411B7B5889E507A3D153995889619134">
            </ds:Signature>
            <wsu:Timestamp wsu:Id="TS-D411B7B5889E507A3D153995889618833">
        </wsse:Security>
    </soapenv:Header>
    <soapenv:Body wsu:Id="id-D411B7B5889E507A3D153995889619136" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurit
        <wst:RequestSecurityToken Context="RC-0f43d8be-322e-4a9c-8473-7308781f733e" xmlns:auth="http://docs.oasis-open.org/wsfed/authorizati
            <wst:TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV2.0</wst:TokenType>
            <wst:RequestType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue</wst:RequestType>
            <wst:KeyType>http://docs.oasis-open.org/ws-sx/wstrust/200512/Bearer</wst:KeyType>
            <wsp:AppliesTo>
                <wsa:EndpointReference>
                    <wsa:Address>https://wwwacc.ehealth.fgov.be/idp/profile/SAML2/Bearer/POST</wsa:Address>
                </wsa:EndpointReference>
            </wsp:AppliesTo>
        </wst:RequestSecurityToken>
    </soapenv:Body>
</soapenv:Envelope>
```

Figure 3: Example SSO POST Request

5.4.2.1.3 Outbound

- Body

  RequestSecurityToken Response(WS-Trust 1.3) containing an Assertion with following info:

  - *Issuer*: eHealth IAM STS
  - *Signature*: full Assertion is signed with eHealth IAM certificate
  - *Subject Confirmation*
    - *Method*: bearer
    - *NotOnOrAfter*: expiration time
    - *Recipient*: URL of consumer of this Assertion
  - *Conditions*: short period of validity
  - *AuthnStatement*: context used to authenticate to the STS at start of SSO Session
  - *AttributeStatement*: attributes present in secure authentication token that was used to authenticate to the STS.

- Example (extract for readability)

Figure 4: Example SSO POST Response

5.4.2.2 Step 4-5: Open browser and authenticate at IDP

Note

The Bearer Assertion is a short-term token.

This means that the user has limited time to use it (a few minutes).

To send the received Assertion to eHealth, it must be wrapped in a SAML 2.0 Response (SAML Web Browser SSO Profile).

```
<saml2p:Response ID="[SAMLResponseID]" IssueInstant="[SAMLResponseIssueInstant]"
Version="2.0" xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol">
  <saml2p:Status>
    <saml2p:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success" />
  </saml2p:Status>
  [SAMLAssertion]
</saml2p:Response>
```

The client application must dynamically fill in the [] fields with following data:

- [*SAMLResponseID*]: unique id for tracing and to prevent certain attacks. Type of this XSD attribute is ID. An xsd:ID value must be an NCName meaning it must start with a letter or underscore, and can only contain letters, digits, underscores, hyphens, and periods.
- [*SAMLResponseIssueInstant*]: time the Response was generated to prevent certain attacks. Type of this attribute is dateTime. The type xsd:dateTime represents a specific date and time in the format CCYY-MM-DDThh:mm:ss.sss, which is a concatenation of the date and time forms, separated by a literal letter "T". All of the same rules that apply to the date and time types also apply to xsd:dateTime. An optional time zone expression may be added at the end of the value.
- [*SAMLAssertion*]: the Assertion received in previous step. As the Assertion is signed by eHealth, do not alter this element in any way when inserting it in the Response element or the signature will break, rendering the Assertion invalid.

After the SAML Response is generated, the client application opens with the user's favorite browser a local html file containing a form.

```
<form method="post" action="[ehealth environment]/idp/profile/SAML2/Bearer/POST">
   <input type="hidden" name="RelayState" value="[targetId]" />
   <input type="hidden" name="SAMLResponse" value="[response]" />
   <input type="submit" value="Submit" />
 </form>
```

The client application must automatically fill in the fields before opening the file in the browser with the following data:

- [*ehealth environment*]: URL of IDP Assertion Consumer in same (!) environment where the Assertion was requested from the STS.
  - *PROD*: ***https://www.ehealth.fgov.be/idp/profile/SAML2/Bearer/POST***
  - *ACC*: ***https://wwwacc.ehealth.fgov.be/idp/profile/SAML2/Bearer/POST***
  - *INT*: ***https://wwwint.ehealth.fgov.be/idp/profile/SAML2/Bearer/POST***
- [*targetId*]: target URL of the protected application the user wants to access (optional)
- [*response*]: SAMLResponse containing the received Assertion, encoded in Base64.

A typical implementation of html file should have an auto submit function, an explanation and a backup solution in case the browser did not submit it automatically.

- Example

SAML 2.0 Response (extract for readability)

```
<saml2p:Response ID="_81e23b6b-c6e8-4810-87da-6379c60a1261" Version="2.0" IssueInstant="2015-04-15T09:47:27.312+02:00"
    <saml2p:Status>
        <saml2p:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success" />
    </saml2p:Status>
    <saml2:Assertion xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion" ID="_9c295dc3845b488488b759b070eb781a" IssueIr
</saml2p:Response>
```

HTMLForm

```
<form method="post" action="https://www.ehealth.fgov.be/idp/profile/SAML2/Bearer/POST">
   <input type="hidden" name="RelayState" value="https://www.partner.fgov.be/secure" />
   <input type="hidden" name="SAMLResponse" value="PHNhb...zZT4=" />
   <input type="submit" value="Submit" />
 </form>
```

6. Step-by-step for Login REST

6.1 Step 1 : SSO web services (STS)

SSO is setup between WS hosted by the eHealth Platform or one of its trusted partners with eHealth IAM STS as STS.

6.2 Step 2 : call to SOAP services

The client App can now send SOAP requests with the SAML HOK token in the ws-security header.

6.3 Step 3 : exchange token

The client App requests an exchange of the SAML HOK token to a JWT.

Endpoint: Token endpoint (in realm healthcare)

- *PROD*: ***https://api.ehealth.fgov.be/auth/realms/healthcare/protocol/openid-connect/token***
- *ACC*: ***https://api-acpt.ehealth.fgov.be/auth/realms/healthcare/protocol/openid-connect/token***
- *INT*: ***https://api-int.ehealth.fgov.be/auth/realms/healthcare/protocol/openid-connect/token***

HTTP method: POST.

It accepts form parameters (application/x-www-form-urlencoded) as input.

Field name	Description
grant_type	MANDATORY. Value MUST be set to “urn:ietf:params:oauth:grant-type:token-exchange”
requested_token_type	MANDATORY. Value must be set to “urn:ietf:params:oauth:token-type:access_token”
subject_token_type	MANDATORY. Value must be set to “urn:ietf:params:oauth:token-type:saml1”
subject_token	MANDATORY. A security token (active) that represents an active SAML HOK token, received from external provider (STS), base64 url encoded.
actor_token_type	MANDATORY. Value must be set to “urn:ietf:params:oauth:token-type:jwt”
actor_token	MANDATORY. This parameter must contain a signed jwt, base64 url encoded, compliant with following specs. See section 8 for more information.
client_id	MANDATORY. This parameter specifies the clientId of the requesting client. The client must be known and active in the targeted realm and the client needs to be granted permission to exchange tokens from the given external provider.

The JWT will contain the user's active profile. It is required that the user is already registered.

6.4 Step 4 : call to REST API

The client app can now send API requests with the eHealth bearer token in the Authorization header

7. Step-by-step for Login Web

7.1 Step 1 : SSO web services (STS)

SSO is setup between WS hosted by the eHealth Platform or one of its trusted partners with eHealth IAM STS as Security Token Service.

7.2 Step 2 : call to SOAP services

The client app can now send SOAP requests with the SAML HOK token in the ws-security header.

7.3 Step 3 : exchange token to an IDToken

The client app requests an exchange of the SAML HOK token to an IDToken

Endpoint: Token endpoint (in realm healthcare)

- *PROD*: ***https://api.ehealth.fgov.be/auth/realms/healthcare/protocol/openid-connect/token***
- *ACC*: ***https://api-acpt.ehealth.fgov.be/auth/realms/healthcare/protocol/openid-connect/token***
- *INT*: ***https://api-int.ehealth.fgov.be/auth/realms/healthcare/protocol/openid-connect/token***

HTTP method: POST.

It accepts form parameters (application/x-www-form-urlencoded) as input.

Field name	Description
grant_type	MANDATORY. Value MUST be set to “urn:ietf:params:oauth:grant-type:token-exchange”
requested_token_type	MANDATORY. Value must be set to “urn:ietf:params:oauth:token-type:access_token”
subject_token_type	MANDATORY. Value must be set to “urn:ietf:params:oauth:token-type:saml1”
subject_token	MANDATORY. A security token (active) that represents the an active SAML HOK token, received from external provider (STS), base64 url encoded.
actor_token_type	MANDATORY. Value must be set to “urn:ietf:params:oauth:token-type:jwt”
actor_token	MANDATORY. This parameter must contain a signed jwt, base64 url encoded, compliant with following specs. See section 8 for more information.
client_id	MANDATORY. This parameter specifies the clientId of the requesting client. The client must be known and active in the targeted realm and the client needs to be granted permission to exchange tokens from the given external provider.

audience	MANDATORY. This parameter contains the issuer of the IAM Connect realm in which the client intends to use the IDToken as authentication token. • *PROD*: ***https://api.ehealth.fgov.be/auth/realms/healthcare*** • *ACC*: ***https://api-acpt.ehealth.fgov.be/auth/realms/healthcare*** • *INT*: ***https://api-int.ehealth.fgov.be/auth/realms/healthcare***
scope	MANDATORY. Value MUST only contain “openid”

It is required that the user is already registered.

7.4 Step 4 : PAR request

The client app sends a PAR (Pushed Authorization Request) to IAM Connect, including the IDToken from previous step as id_token_hint.

IAM Connect generates a unique reference and returns this.

Endpoint: pushed_authorization_request_endpoint (in realm healthcare)

- *PROD*: ***https://api.ehealth.fgov.be/auth/realms/healthcare/protocol/openid-connect/ext/par/request***
- *ACC*: ***https://api-acpt.ehealth.fgov.be/auth/realms/healthcare/protocol/openid-connect/ext/par/request***
- *INT*: ***https://api-int.ehealth.fgov.be/auth/realms/healthcare/protocol/openid-connect/ext/par/request***

HTTP method: POST.

It accepts form parameters (application/x-www-form-urlencoded) as input.

Field name	Description
client_id	MANDATORY. This parameter specifies the clientId of the requesting client. The client must be known and active in the targeted realm and the client needs to be granted permission to exchange tokens from the given external provider.
redirect_uri	MANDATORY. It contains valid redirect_uri, registered for the requesting client.
response_type	MANDATORY. Value MUST contain “code”.
scope	MANDATORY. Value MUST contain “openid” at least.
prompt	MANDATORY. Value MUST be “none” for IAM Connect not to show the user any login options.
id_token_hint	MANDATORY. It MUST contain the IDToken, returned by IAM Connect in previous step. Only valid IDTokens, returned in the context of this token exchange will be accepted. If not acceptable, the authorization server will return an error "login required".

7.5 Step 5-6 : call for web authorization

The client App opens a browser which it sends the unique reference from previous step to the authorization endpoint of IAM Connect. IAM Connect will authenticate the user based on the id_token_hint and return an authorization_code to the redirect_uri of the client app.

Endpoint: Authentication endpoint (in realm healthcare)

- *PROD*: ***https://api.ehealth.fgov.be/auth/realms/healthcare/protocol/openid-connect/auth***
- *ACC*: ***https://api-acpt.ehealth.fgov.be/auth/realms/healthcare/protocol/openid-connect/auth***
- *INT*: ***https://api-int.ehealth.fgov.be/auth/realms/healthcare/protocol/openid-connect/auth***

HTTP method: GET.

Field name	Description
client_id	MANDATORY. This parameter specifies the clientId of the requesting client.
request_uri	MANDATORY. It contains the uri, received in previous step from IAM connect as response on the PAR request. It must be sent within 60 seconds

The client App gets the token from the authorization_code (step 6) on the token endpoint of IAM Connect

7.6 Step 7-9 : call to REST API with client/browser

The client App can now send API requests with the eHealth bearer token in the Authorization header. This is optional and not related to the start of the web authentication in the browser.

When receiving the request on the redirect_uri endpoint, the client App can send a new request in the browser to redirect the user to the url of a webapp in which he wants to work (step 8).

The webApp, which has no active session for the user's browser, will redirect the user to IAM Connect for authorization. IAM Connect will already have an active session, based on previous steps, and will generate a token for that application without requesting the user for re-authentication.

8. Actor token generation

For the exchange operation (Sections 6.3 and 7.3), the client MUST generate an actor token.

The actor token is a signed JWT.

The signed JWT is composed of 3 parts :

- Header
- Payload
- Signature

Signing MUST be done with the key, set as holder-of-key in SAML HOK token, as proof of possession.

8.1 header

The header contains the following claims :

Field name	Description
typ	MANDATORY. Value MUST be set to “JWT”
alg	MANDATORY. The value must correspond to the algorithm used (usually RS256).
kid	OPTIONAL. The id of the key used.

8.2 payload

The payload contains the following claims :

Field name	Description
iss	MANDATORY. Value MUST correspond to the clientID
sub	MANDATORY. Value MUST correspond to the SSIN of the end user (identified in the SAML HOK token)
aud	MANDATORY Value must correspond to the issuer of the external provider who issued the SAML HOK token. Value MUST be *urn:be:fgov:ehealth:sts:1_0*
jti	OPTIONAL. Unique identifier for the JWT. The identifier value MUST be assigned in a way that ensures there is a negligible probability of accidental assignment to a different data object.
iat	MANDATORY. *Issued At* claim identifies the time at which the JWT was issued. The JWT must not be issued more than a few minutes in the past.

9. Test and release procedure

9.1 Procedure

This chapter explains the procedures for testing and releasing an application in acceptation or production.

9.1.1 Initiation

If you intend to use the eHealth service, please contact ***info@ehealth.fgov.be***. The Project department will provide you with the necessary information and mandatory documents.

9.1.2 Development and test procedure

You have to develop a client to connect to our web service. Most of the integration info is available on the eHealth portal.

9.1.3 Release procedure

When development tests are successful, you can request to access the eHealth acceptance environment.

From this moment, you start integration and acceptance tests. The eHealth platform suggests testing during a minimum of one month.

After successful acceptance tests, the partner sends his test results and performance results with samples of "eHealth request" and "eHealth answer" to the eHealth point of contact by email.

Then eHealth and the partner agree on a release date. The eHealth platform prepares the connection to the production environment and provides the partner with the necessary information. On the release day, the partner provides the eHealth platform with feedback on the test and performance tests.

For further information and instructions, please contact: ***integration-support@ehealth.fgov.be***.

9.1.4 Operational follow-up

Once in production, the partner using the eHealth service for one of his applications will always test first in the acceptance environment before releasing any adaptations of its application in production. Additionally he will inform the eHealth platform on the progress and test period.

9.1.5 The use of username, password and token

The username, password, and token are strictly personal.
Every user is responsible for maintaining the confidentiality of his username, password, and token. It is prohibited to transfer them to partners and clients. Until inactivation, every user is responsible for any use, including use by a third party.

10. Error and failure messages

The WSC/client MUST manage errors and MUST display in a correct way (when necessary) the error(s) to the end user.

10.1 STS to IDP

If an error occurs during the operation *Request Bearer Token from STS, you should* first verify your request.

A SOAP fault exception should be returned.

SystemError

SOA error guide (on eHealth portal) provides an overview of the possible error messages and their solution.

Example of SystemError

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-trust/200512">wst:RequestFailed</faultcode>
      <faultstring>The specified request failed</faultstring>
      <detail>
        <urn:SystemError Id="Id-60cf86662119699260bac763" xmlns:urn="urn:be:fgov:ehealth:errors:soa:v1">
          <Origin>Consumer</Origin>
          <Code>SOA-01001</Code>
          <Message xml:lang="en">Service call not authenticated.</Message>
          <urn:Environment>Acceptation</urn:Environment>
        </urn:SystemError>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

BusinessError

BusinessError might also be returned. Following table contains a list of common business error codes for the service /IAM/SingleSignOnService/v1.

Error code	Error message	Error description
urn:oasis:names:tc:SAML:2.0:status:RequestDenied	Message did not meet security requirements X.509 Attribute Mismatch	The SAML HOK token (certificate holder) used does not match the SSIN provided in the payload. Verify the SSIN provided in the claim with Uri *urn:be:fgov:person:ssin*.
wst:InvalidRequest	Message not properly encoded Extracting TokenType [%string%] failed	The TokenType in the request is probably incorrect. The value should be ***http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV2.0***
wst:InvalidRequest	Message not properly encoded Extracting RequestType [%string%] failed	The RequestType is probably incorrect.. The value should be ***http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue***
wst:InvalidRequest	Message not properly encoded Extracting KeyType [%string%] failed	The KeyType in the request is probably incorrect. The value should be ***http://docs.oasis-open.org/ws-sx/wstrust/200512/Bearer***

urn:be:fgov:ehealth:1.0:status:MetadataInvalid	Failure validating Endpoint	The element Address (in RequestSecurityToken.AppliesTo.EndpointReference) is probably incorrect. Check the values in sections 5.4.1.1.2 (or 5.4.2.1.2).

Example of BusinessError:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-trust/200512">wst:InvalidRequest</faultcode>
      <faultstring>The request was invalid or malformed</faultstring>
      <detail>
        <urn:BusinessError Id="_6b8ded7bfde4493801a2a18e8a6da7da" xmlns:urn="urn:be:fgov:ehealth:errors:soa:v1">
          <Origin>Client</Origin>
          <Code>wst:InvalidRequest</Code>
          <Message xml:lang="en">Message not properly encoded</Message>
          <Message xml:lang="en">Extracting TokenType [http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV2.]
failed</Message>
          <urn:Environment>Integration</urn:Environment>
        </urn:BusinessError>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

The HTTP header contains a X-CorrelationID . This ID is important for the tracking of the errors so when available, please provide it when requesting support.

10.2 STS to IAM Connect

Token endpoint

If an error occurs on token endpoint, first please verify your request.

Following table contains a list of common error codes

HTTP status code	Error code	Error description	Recommendation
400	invalid_scope	client no longer has requested consent from user	The client is requesting a client scope for which the end user has not granted consent. The end user should be redirected (browser flow) to the login/consent page.
400	invalid_token	invalid subject_token	The subject token provided is not valid : • It may not contain a base64 encoded SAML HOK token. • The SAML HOK assertion signature cannot be verified. • The SAML HOK assertion may be expired. • The SAML HOK assertion might come from another environment.
400	invalid_token	invalid actor_token	The actor token generated is not valid : • The SAML HOK token certificate holder attribute does not match the claim sub. • The claim aud is not *urn:be:fgov:ehealth:sts:1_0*
400	invalid_request	requested_token_type unsupported	Verify the value of the parameter. It should be *urn:ietf:params:oauth:token-type:access_token*
400	Invalid_token	Invalid token	Verify the value of the parameter subject_token_type. It should be *urn:ietf:params:oauth:token-type:saml1*
400	Invalid_request	invalid actor_token_type	Verify the value of the parameter actor_token_type. It should be *urn:ietf:params:oauth:token-type:jwt*

Pushed Authorization Request endpoint

If an error occurs on PAR endpoint, first please verify your request.

Following table contains a list of common error codes

HTTP status code	Error code	Error description	Recommendation
401	unauthorized_client	Client is not allowed to initiate browser login with given response_type. Implicit flow is disabled for the client.	Response_type cannot contain id_token or token. It should be code.
400	Invalid_request	Invalid parameter: redirect_uri	The redirect_uri must match a redirect_ui configured in IAMConnect for the requesting client.
400	invalid_request	Authentication failed.	Client must be registered with given id.
400	Invalid_request	Invalid scopes: …	Requested scopes must exist and be configured for requesting client.

Authorization endpoint

If an error occurred on the authorization endpoint, check the parameters:

- client_id
- request_uri

If the client_id is incorrect, you get this error :

If the request_uri is incorrect/invalid, you get this error :

Check that :

- the request_uri was sent within 60 seconds after receiving it from the PAR endpoint;
- the request_uri was URL encoded.

Following table contains a list of common error codes

HTTP status code	Error code	Error description	Recommendation
200	login_required		Only valid IDTokens returned in the context of this token exchange are accepted.
200	interaction_required	An error is returned if an End-User is not already authenticated or the Client does not have pre-configured consent for the requested Claims or does not fulfil other conditions for processing the request.	Retry the call from PAR request with prompt = consent.