TimeStamping Authority V2.0
Cookbook
Version 1.8

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	13/10/2014	eHealth platform	Initial version
1.1	27/10/2016	eHealth platform	Updated endpoint for TSA V1
1.2	05/07/2017	eHealth platform	p.m.
1.3	31/05/2018	eHealth platform	Update
1.4	24/04/2021	eHealth platform	§ 5.1.2 WS-I Compliance § 5.1.3 Tracing
1.5	11/01/2022	eHealth platform	§ 5.3.2.2 Document hash
1.6	11/07/2022	eHealth platform	§ 2.3 eHealth platform document references (updated) § 3.2 Status (added) § 5.1.3 Tracing (updated)
1.7	24/01/2023	eHealth platform	Remove support for SHA-1
1.8	11/06/2024	eHealth platform	Remove TSA v1

2 Introduction

Within the Belgian hospitals, more ICT systems are being introduced to support the daily operations. Consequently, the hospitals are migrating towards a paperless environment where information is electronically stored.

This makes it more difficult for the supervising authorities to fulfil their duties, as paper documents are currently the most important source of proof of activities. However, migrating to an ICT-based environment also introduces new opportunities, such as the timestamp service.

2.1 Goal of the service

This document provides technical information on calling the web service (WS) timestamping authority of the eHealth platform. The goal of this WS is to offer the possibility of providing a proof that a document existed on a given date by means of a trusted third party timestamp. A hash code of the timestamped document and the timestamp can be archived on the eHealth server. This archive can be consulted through the timestamp consult service (see Cookbook Timestamp Consult).

Currently, there are two versions of the services TSA. The second version includes some bug fixes and some improvements compared to the initial version of the service:

- In TSA V2, the <RFC3161TimeStampToken> returned in response is encoded only once in base64 while it is base 64 double encoded in TSA V1
- In TSA V2, the timestamp token itself is returned instead of timestamp response (wrapper around timestamp token)

However, the first version is still used by many partners.

2.2 Goal of the document

This document is not a development or programming guide for internal applications. Instead, it provides functional and technical information and allows an organization to integrate and use the eHealth service.

However, to ensure smooth, consistent and risk-controlled interactions with as many partners as possible, eHealth partners must commit to complying with the specifications, data formats and release processes described in this document.

Both technical and business requirements must be met to enable the integration and validation of the eHealth service in the client application.

2.3 eHealth document references

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

ID	Title	Version	Date	Author
1	Timestamping Consult V2 Cookbook	1.5	11/07/2022	eHealth platform

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

2.4 External document

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

ID	Title	Source	Date	Author
1	RFC3161	***http://www.ietf.org/rfc/rfc3161.txt***	22/09/2010	Network Working Group
2	XML Timestamping Profile of the OASIS Digital Signature Services	***http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles-timestamping-spec-v1.0-os.pdf***	22/09/2010	OASIS
3	Basic Profile Version 1.1	***http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html***	24/08/2004	Web Services Interoperability Organization

3 Support

3.1 Helpdesk eHealth platform

3.1.1 Certificates

To access the secured eHealth platform environment you must obtain an eHealth platform certificate, which is used to identify the initiator of the request. If 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***

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

Web services

The timestamping project is composed by two web services:

1. **Timestamping Authority (TSA)**
   This service contains one operation called **SignRequest** which performs the classical operations of a timestamping authority on a document:
   There are two possible ways to send documents to the timestamp server, either as a simple document or as a hash code. In both cases, the client must send the value in Base64. The server checks if it has received a hash code. If not, it calculates the hash code of the document received. Next, it concatenates a timestamp to the hash of the document and calculates the hash of this result. Then this hash is digitally signed with the private key of eHealth and a timestamp token is created and returned to the user. The timestamp token contains the information the client application will need to verify the timestamp later. The timestamp authority service implements the OASIS-DSS protocol but does not support multiple document requests. The response token is defined by the RFC 3161 protocol.

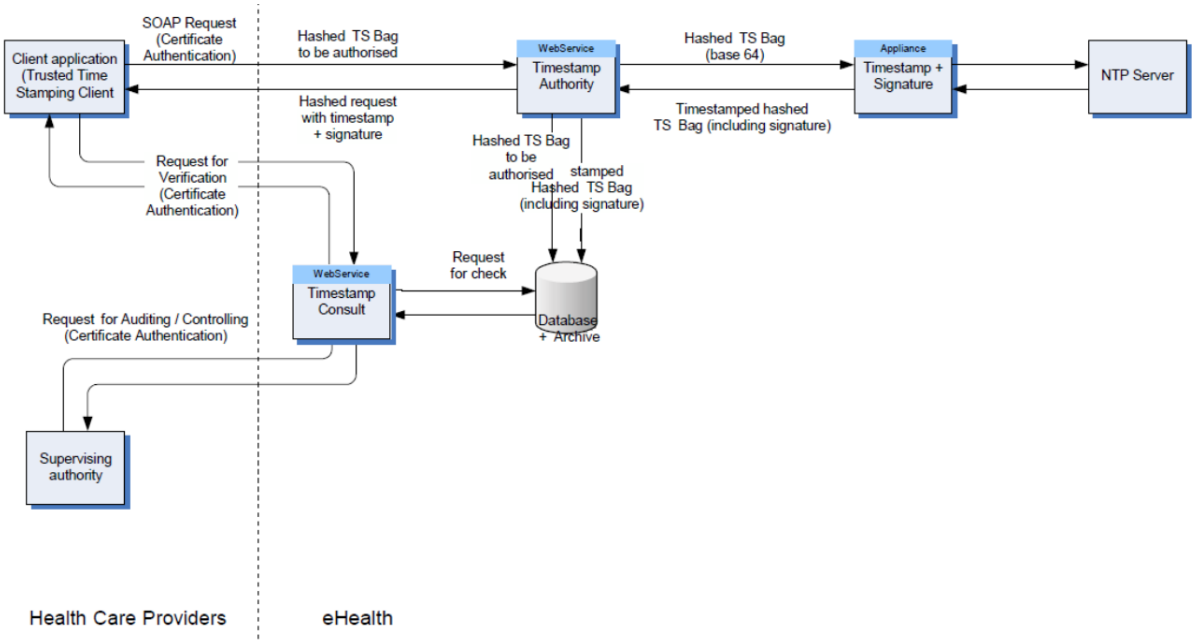
   Moreover, this service offers the possibility to store the received document in the eHealth archive to verify the integrity of the user's archive later. By default, the archiving is only activated for the hospitals involved in the project Electronic Prescription in Hospitals. For the other projects, the archiving is not activated and users have to make a special request to ***support@ehealth.fgov.be*** if they want to activate it (see more information in point 5.3.1.1).

2. **Timestamping Consult (TSC)**
   As the name suggests, this service allows to consult the archive of the time stamp server (for more details, see the cookbook Timestamp Consult service). It contains two operations:
   - **TSConsultTSBagRequest**: Consultation by sending a sequence number and a time (in milliseconds expressed as UTC time). The server will respond with the corresponding timestamp.
   - **TSConsultRequest**: Consultation by sending a period and a list of sequences numbers and times. The server will respond with a list of missing timestamps. It offers the client the possibility to check if his own archive is complete.

   Obviously, the user cannot use the consultation service if archiving is not configured.

Electronic Prescription in Hospitals

In the context of the electronic prescription in hospitals, the use of the timestamping services is governed by a legal context.



The hospital doctor issues an electronic prescription (the document to be timestamped) which is hashed. Every five minutes, the hashed prescriptions are collected into a package called “TimeStampBag”. This bag is sent to the eHealth Platform through the web service *Timestamp Authority* for timestamping. The timestamp token generated from the bag is sent back to the hospital for preservation in its archive. Additionally, a copy of the bag and its timestamp is stored in the eHealth archives.

When desired, the hospital can consult the eHealth archives through the web service Timestamp Consult to compare their content and the content of the hospital’s archives. All bags contained in the hospital’s archives of the hospital should be stored in the eHealth archives. Naturally, a hospital can consult only its own bags in the eHealth archives; it cannot consult the bags of another hospital.

Moreover, the NIHDI can also consult the eHealth Platform archives (linked to any hospital) through the web service Timestamp Consult for auditing and control as a supervising authority.

5 Step-by-step

5.1 Technical requirements

The client must have a certificate to use the service (see Chap 3.1.1)

There are two main differences between the two possible ways to send documents:

Document	Document Hash
The client must provide the Base64 code of the document.	The client must provide: • The Base64 code of the hash of the document. • The id of the document. • The digest method used to calculate the hash code.
The hash code of the document is calculated by the server.	The hash code of the document is calculated by the client.

The following digest methods are supported:

- SHA-256 (V1) OR ***http://www.w3.org/2001/04/xmlenc#sha256*** (V2)
- SHA-384 (V1) OR ***http://www.w3.org/2001/04/xmldsig-more#sha384*** (V2)
- SHA-512 (V1) OR ***http://www.w3.org/2001/04/xmlenc#sha512*** (V2)

The digest method used by the TS server for a document is SHA-256.

The response token (Base 64) contains:

- the hash code of the document,
- the digest method used to calculate the hash code,
- the timestamp generated by the timestamp server (in milliseconds expressed as UTC time),
- the sequence number of the timestamp generated by the timestamp server.

(For more information about the response token, see the specifications of the RFC 3161 protocol – Par 2.4).

5.1.1 Security policies to apply

You must use SSL one- way for the transport layer.

As WS security policy, we expect:

- A timestamp (the date of the request), with a time-to-live of one minute (if the message does not arrive during this minute, it shall not be treated).
- The signature with the certificate of
  - the timestamp (the one mentioned above),
  - the body (the message itself),
  - and the binary security token: an eHealth certificate.

  This will allow eHealth to verify the integrity of the message and the identity of the message author.

A document explaining how to implement this security policy can be obtained at the eHealth platform.

5.1.2 WS-I Basic Profile 1.1

Your request must be WS-I compliant (See Chap 2.4 External document references).

5.1.3 Tracing

To use this service, the request SHOULD contain the following two http header values (see RFC

https://datatracker.ietf.org/doc/html/rfc7231#section-5.5.3):

1. User-Agent: information identifying the software product and underlying technical stack/platform. It MUST include the minimal identification information of the software such that the emergency contact (see below) can uniquely identify the component.
   a. Pattern: {minimal software information}/{version} {minimal connector information}/{connector-package-version}
   b. Regular expression for each subset (separated by a space) of the pattern: [[a-zA-Z0-9-\/]*\/[0-9azA-Z-_.]*
   c. Examples:
      User-Agent: myProduct/62.310.4 Technical/3.19.0
      User-Agent: Topaz-XXXX/123.23.X freeconnector/XXXXX.XXX
2. From: email-address that can be used for emergency contact in case of an operational problem.
   Examples:
   From: ***info@mycompany.be***

5.2 Process overview

The partner wanting to use the timestamping service(s) must first contact eHealth (***info@ehealth.fgov.be***) in order to specify clearly the context and the purpose of his project. If his request is validated, he must obtain a certificate (see paragraph 3.4) and contact the eHealth platform to configure his profile. Then, he can start using the service.

NB: As part of the electronic prescription in hospitals, the hospital must fill and sign the protocol found here ***https://www.ehealth.fgov.be/ehealthplatform/fr/service-datation-electronique-timestamping***.

5.3 Web service

The WS TSA V2 contains one operation called “SignRequest” and it has the following endpoints:

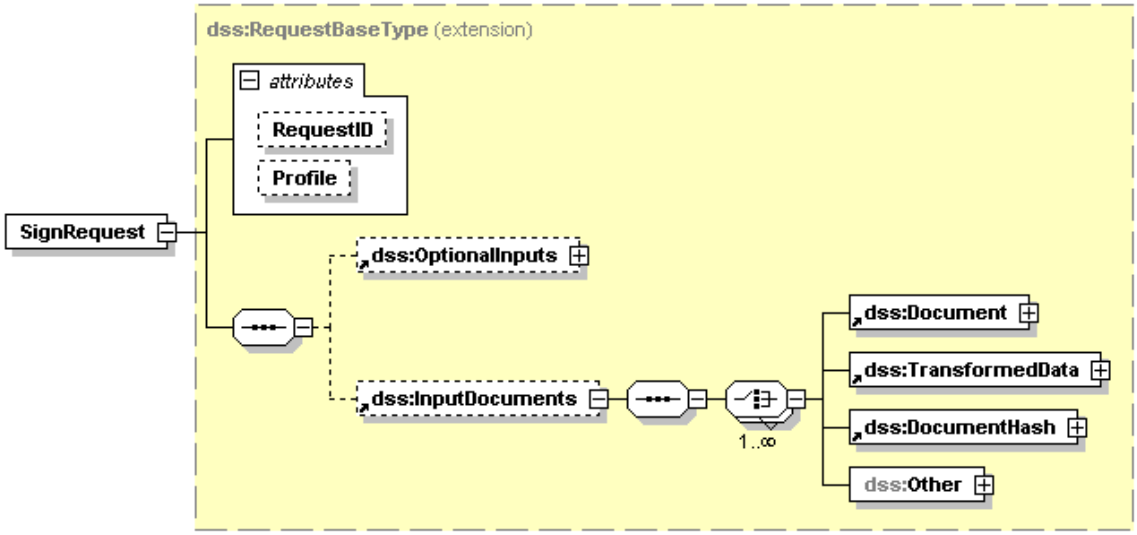
- Integration environment : ***https://services-int.ehealth.fgov.be/TimestampAuthority/v2***
- Acceptance environment : ***https://services-acpt.ehealth.fgov.be/TimestampAuthority/v2***
- Production environment : ***https://services.ehealth.fgov.be/TimestampAuthority/v2***

Note: Fields that are not described may be ignored.

5.3.1 SignRequest

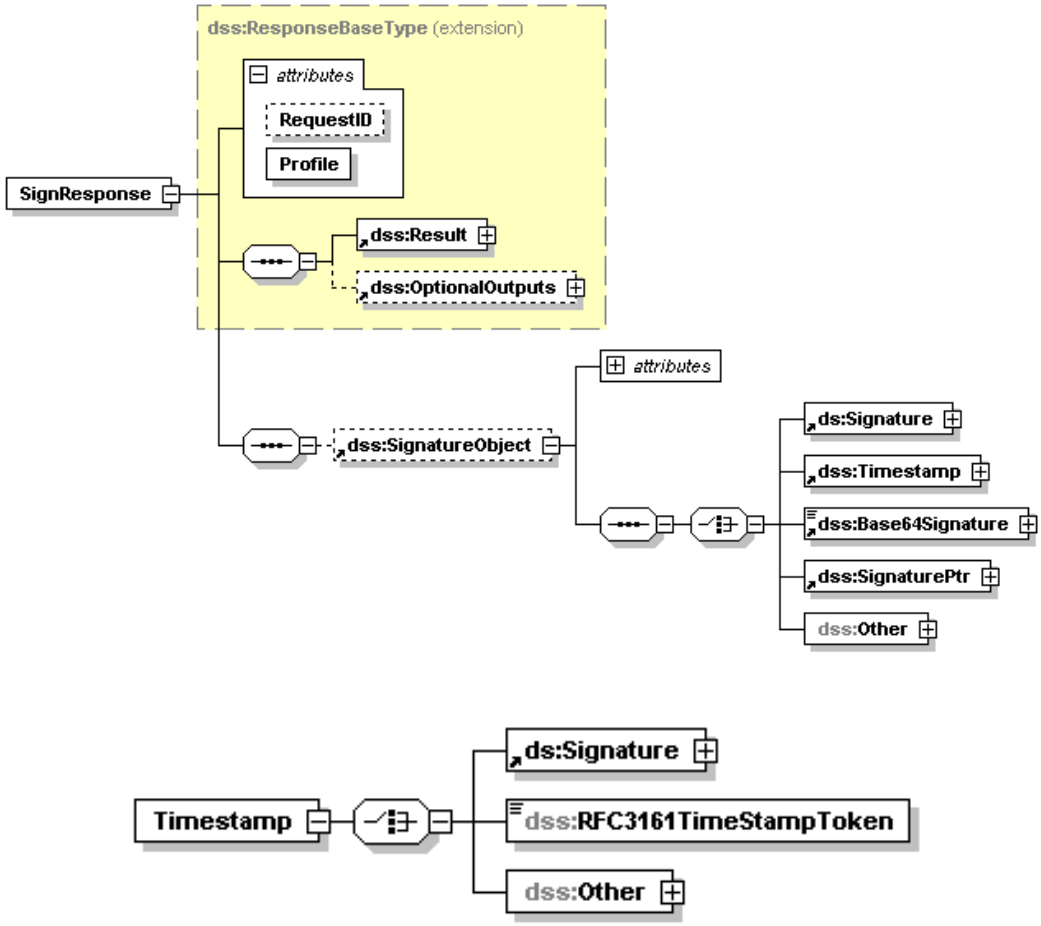
If it is a multiple-document request, only the first one is processed. The two available InputDocuments types are Document and DocumentHash.

5.3.1.1 Input arguments



Field name	Description
RequestID	Unique request identifier
Profile	**urn:ehealth:profiles:timestamping:1.1** for a Document in input (the output will be a token without TSA certificate) **urn:ehealth:profiles:timestamping:2.1** for a DocumentHash in input (the output will be a token without TSA certificate) **urn:ehealth:profiles:timestamping:1.1-cert** for a Document in input (the output will be a token with TSA certificate) **urn:ehealth:profiles:timestamping:2.1-cert** for a DocumentHash in input (the output will be a token with TSA certificate) ***Note:*** Currently, the default configuration for timestamp clients is **TSA v2** **Hospitals**: 1.1, 1.1-cert with archiving; 2.1 and 2.1-cert, without archiving **Other projects**: 2.1 and 2.1-cert, without archiving Therefore, if you want to use another configuration (by example profile 2.0 with archiving), you have to make a special requirement to ***support@ehealth.fgov.be***
InputDocuments	Document (Base64)

5.3.1.2 Output arguments



Field name	Description
Profile	Same value as the corresponding field of the request
RequestID	Same value as the corresponding field of the request
Result	Contains the result message (success, error description, …)
RFC3161TimeStampToken	Timestamp Token

5.3.1.3 Example

Request:

```
<urn:SignRequest RequestID="unique_request_identifier" Profile="urn:ehealth:profiles:timestamping:1.1">
      <urn:InputDocuments>
        <urn:Document>
            <urn:Base64Data>ZUhlYWx0aCBUaW1lc3RhbXBpbmcgU2VydmljZQ==</urn:Base64Data>
        </urn:Document>
      </urn:InputDocuments>
</urn:SignRequest>
```

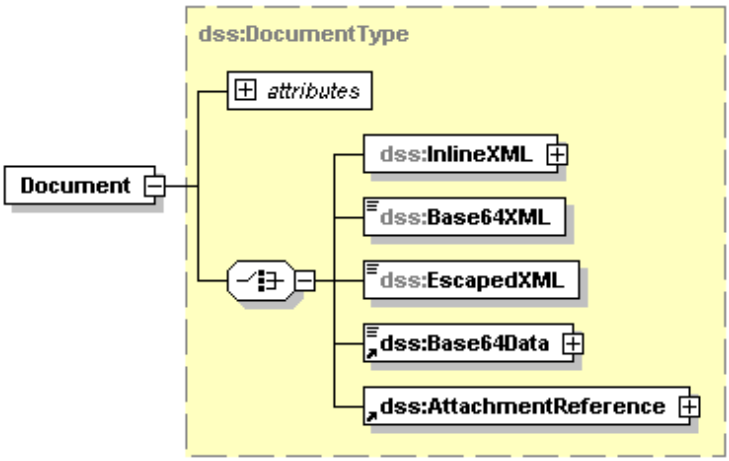
Response:

<urn:SignResponse RequestID="unique_request_identifier" Profile="urn:ehealth:profiles:timestamping:1.1" xmlns:urn="urn:oasis:names:tc:dss:1.0:core:schema">
<urn:Result>
<urn:ResultMajor>urn:oasis:names:tc:dss:1.0:resultmajor:Success</urn:ResultMajor>
<urn:ResultMinor>urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:OnAllDocuments
</urn:ResultMinor>
<urn:ResultMessage xsi:nil="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
</urn:Result>
<urn:SignatureObject>
<urn:Timestamp>
<urn:RFC3161TimeStampToken>
MIAGCSqGSIb3DQEHAqCAMIIEAAIBAzEPMA0GCWCGSAFlAwQCAQUAMIIBGQYLKoZIhvcNAQkQAQSgggEIBIIBB
DCCAQACAQEGCisGAQQBhFkKAwEwMTANBglghkgBZQMEAgEFAAQgDDAoP6nrSGc5DOLBQo9Id6bH2niNcKm5
ESvRpoFDJ68CBlN5v9BkjRgTMjAxNDEwMTMxMTI2MTkuMTYzWjAEgAIB9KCBmKSBlTCBkjEbMBkGA1UEAxMSU
21hbHMgLSAwNDA2Nzk4MDA2MQswCQYDVQQGEwJCRTEOMAwGA1UEChMFU21hbHMxFzAVBgNVBAsTDlRvZ
WdhbmdzYmVoZWVyMRIwEAYDVQQHEwlTVCBHSUxMRVMxKTAnBgkqhkiG9w0BCQEWGnNhcmEudmFuZGVyb
WVlcmVuQHNtYWxzLmJlMYICyzCCAscCAQEwfzBwMQswCQYDVQQGEwJCRTEcMBoGA1UEChMTQ2VydGlwb3N
0IHMuYS4vbi52LjFDMEEGA1UEAxM6Q2VydGlwb3N0IEUtVHJ1c3QgU2Vjb25kYXJ5IFF1YWxpZmllZCBDQSBmb3I
gTGVnYWwgUGVyc29ucwILAQAAAAABLsNdpM4wDQYJYIZIAWUDBAIBBQCgggEdMBoGCSqGSIb3DQEJAzENBgs
qhkiG9w0BCRABBDAvBgkqhkiG9w0BCQQxIgQgbaX0q6y+A5h3xaNaHiwiJMh/cazBaGhA4iTJaVSPp4Qwgc0GCyq
GSIb3DQEJEAIMMYG9MIG6MIG3MIGcBBTNzqr/i3BsScdKaFfwqYZ0gK9FEDCBgzB0pHIwcDELMAkGA1UEBhMCQ
kUxHDAaBgNVBAoTE0NlcnRpcG9zdCBzLmEuL24udi4xQzBBBgNVBAMTOkNlcnRpcG9zdCBFLVRydXN0IFNlY29uZ
GFyeSBRdWFsaWZpZWQgQ0EgZm9yIExlZ2FsIFBlcnNvbnMCCwEAAAAAAS7DXaTOMBYEFGVGylaF5LExe0OMx9
Mb9NB9IIOkMA0GCSqGSIb3DQEBCwUABIIBAJGVL1QH7+JpxohApuL/mrU1f5ZqYEhUfK1c3ku1zs9PPKfW0ab4ug
C5OYsm6Ai2ukumsxgDDz2+wYZi2kVAFJ5G879EGWIMvQVe1jX5M5x0iLZp44emb34hsW3pMOAj0vaW3LfgHQzj
3JfkuBf/g5IESM2ncKk4LgDCIFabOdMm2FDkeyb4P+96yC2XGGfH2eCQsYD9nMx0RFo5B2RtRDxte+DnodQtn4Y3p
yoto+AzpacLWhKdG0PBZydCaMJg0TFRLo090g5ilhWnCamlcPzPjw1KJOBwHbOacYSR3UxaUzRFmiaaAcYlj2KuEc5
OELc+B/KlEa3Z2oySQ1A/4tYAAAAA</urn:RFC3161TimeStampToken>
</urn:Timestamp>
</urn:SignatureObject>
</urn:SignResponse>

5.3.2 Used Types

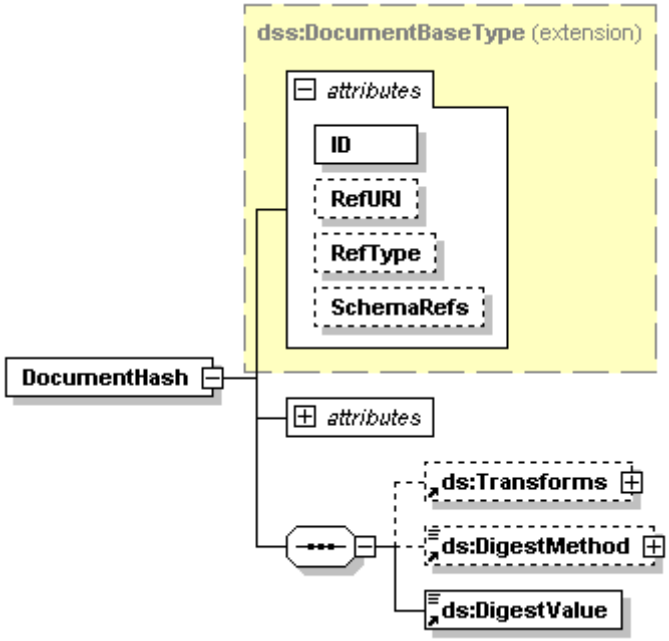
Fields which have not been described may be ignored.

5.3.2.1 Document



Field name	Descriptions
Base64Data	Base64 of the document

5.3.2.2 Documenthash



Field name	Descriptions
ID	Unique identifier for a document (not for the hash) Note that this attribute is defined as an 'xs:NCName'. That definition forbids – among other limitations – that this element starts with a number, dot or minus character.
DigestMethod	Method used to calculate the hash (see point 5.1)
DigestValue	Hash of the document

5.3.2.3 Example

Request:

<urn:SignRequest RequestID="unique_request_identifier" Profile="urn:ehealth:profiles:timestamping:2.1">
<urn:InputDocuments>
<urn:DocumentHash ID="unique_document_identifier">
<xd:DigestMethod Algorithm="***http://www.w3.org/2001/04/xmlenc#sha256***"></xd:DigestMethod>
<xd:DigestValue>5dem9pzA0SRil7f87op5CKY6YPqhLo0tOAsBgKmmw5c=</xd:DigestValue>
</urn:DocumentHash>
</urn:InputDocuments>
</urn:SignRequest>

Response:

<ns3:SignResponse Profile="urn:ehealth:profiles:timestamping:2.1" RequestID="unique_request_identifier" xmlns="urn:oasis:names:tc:SAML:1.0:assertion" xmlns:ns2="http://www.w3.org/2000/09/xmldsig#" xmlns:ns3="urn:oasis:names:tc:dss:1.0:core:schema">
<ns3:Result>
<ns3:ResultMajor>urn:oasis:names:tc:dss:1.0:resultmajor:Success</ns3:ResultMajor>
<ns3:ResultMinor>urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:OnAllDocuments
</ns3:ResultMinor>
</ns3:Result>
<ns3:SignatureObject>
<ns3:Timestamp>
<ns3:RFC3161TimeStampToken>

MIAGCSqGSIb3DQEHAqCAMIICkAIBAzEPMA0GCWCGSAFlAwQCAQUAMFUGCyqGSIb3DQEJEAEEoEYERDBCAgEBBgEqMCEwCQYFKw4DAhoFAAQU54+O8pPrvaDVRIGuZqTRAYmFIPoCBgFJCUaaTxgPMjAxNDEwMTMxMTMxMDJaMYICITCCAh0CAQEwWzBSMRMwEQYDVQQDEwplSGVhbHRoIENBMQswCQYDVQQGEwJCRTEZMBcGA1UEChMQZUhlYWx0aC1wbGF0Zm9ybTETMBEGA1UECxQKU21hbHMgVE8mUAIFExqifZwwDQYJYIZIAWUDBAIBBQCggZgwGgYJKoZIhvcNAQkDMQ0GCyqGSIb3DQEJEAEEMBwGCSqGSIb3DQEJBTEPFw0xNDEwMTMxMTMxMDJaMCsGCyqGSIb3DQEJEAIMMRwwGjAYMBYEFH37sdGJrA+x08UUeHgZ5n4uDgejMC8GCSqGSIb3DQEJBDEiBCBje+SvaynEbmMAvuHkFkQIn5REjMMymFS2N92ShMk4wzANBgkqhkiG9w0BAQEFAASCAQCDF/8Vp5TWftUOPvXCNWWDcm/ZUjUYleUXbLRBJzdK0aGdlxgyKs9DqyMLCngrOZY8+P6oqWNkR6FFyMbJnzo4KxOqVux9TcwwW9ySBLOQ+fnUppLDFke2vlusQ4foW4DMxVhKgIL4bNmLwJi2Nknx0QnQds0ihhZ1c6/cJdrYHICKLUwThW2LUcjZMW5JJrC5JepetXJIybH3tRxytveoo91gXNYlnhLKDtMFhucky+3o4dAFRsWshSDFfbeiIbWd7OWNC/mfuNL1XfQIBU3f7bvIOGf+1U2zvvp2QHjK49a2L51/5+Ww+Q32gJbBP6EeYEipDgOsABnHmdy6vBOBAAAAAA==</ns3:RFC3161TimeStampToken>

</ns3:Timestamp>
</ns3:SignatureObject>
</ns3:SignResponse>

6 Risks and security

6.1 Security

6.1.1 Business security

In case the development adds an additional use case based on an existing integration, the eHealth platform must be informed at least one month in advance with a detailed estimate of the expected load. This will ensure an effective capacity management.

In case of technical issues on the WS, the partner may obtain support from the contact center (see Chap 3)

If the eHealth platform finds a bug or vulnerability in its software, we advise the partner to update his application with the newest version of the software within 10 business days.

If the partner finds a bug or vulnerability in the software or web service provided by the eHealth platform, he is obliged to contact and inform us immediately. The partner is not allowed to publish this bug or vulnerability under any circumstances.

6.1.2 Web service

WS security used in this manner is in accordance with the common standards. Your request will provide:

- SSL one way
- time-to-live of the message: one minute.
- signature of the timestamp, body and binary security token. This will allow the eHealth platform to verify the integrity of the message and the identity of the message author.
- no encryption on the message.

6.1.3 The use of username, password and token

The username, password, and token are strictly personal and must not be shared.

Every user is responsible for maintaining the confidentiality of their username, password. Every user is also accountable for all activities conducted with their credentials, including the use by a third party, until the credentials are deactivated. .

7 Test and release procedure

7.1 Procedure

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

7.1.1 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.

7.1.2 Development and test procedure

You have to develop a client in order to connect to our WS. Most of the required info to integrate is published in the Support section on the eHealth portal.

In some cases, the eHealth platform provides you with a mock-up service or test cases in order for you to test your client before releasing it in the acceptance environment.

7.1.3 Release procedure

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

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

After successful acceptance tests, the partner sends the test results and performance results along with a sample of “eHealth request” and “eHealth answer” to the eHealth point of contact by email.

Subsequently, the eHealth platform 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 tests and performance results.

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

7.1.4 Operational follow-up

Once in production, the partner using the eHealth service for one of his applications will always test any adaptations in the acceptance environment before releasing them to production. Additionally, the partner will inform the eHealth platform about the progress and the testing period.

8 Error and failure messages

8.1 Error codes originating from the eHealth platform:

These error codes first indicate a problem in the arguments sent, or a technical error.

Error code	Description
OASIS_RESULT_MAJOR_REQUESTER_ERROR	The request is invalid
OASIS_RESULT_MAJOR_RESPONDER_ERROR	An error occurred on the server side

When an error occurred, a detailed message is sent.

8.2 WS-I Basic Profile 1.1 - Errors

SOA-03001	***Malformed message***	*Consumer*	*This is the default error for content related errors in case no more details are known.*
SOA-03002	***Message must be SOAP***	*Consumer*	*Message does not respect the SOAP standard.*
SOA-03003	***Message must contain SOAP body***	*Consumer*	*Message respects the SOAP standard, but body is missing.*