Emergency Medical Service Registry WS (EMSR)
Consultation
Cookbook
Version 2.12

eHealth platform

Willebroekkaai 38 – 1000 Brussel

38, Quai de Willebroek – 1000 Bruxelles

Table of contents

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	28/07/2016	SMALS	First version
1.1	19/09/2016	eHealth-platform	Fine tuning Consultation & RegistrationAdded: chapter 5.1.1. (SSO)
1.2	20/09/2016	eHealth-platform	Added: 5.1.1.2 (SSO)
1.3	06/10/2016	SMALS	GetSheetResponse: individual result encrypted
1.4	18/10/2016	SMALS	Change certificate identifier
2.0	31/08/2017	SMALS	Change UAM Configuration and KMEHR messagestructure
2.1	20/09/2017	SMALS	Corrections after feedback from integrator
2.2	24/10/2017	SMALS	Correction of the operation name getSds (previously getSDS in the documentation)
2.3	12/12/2017	SMALS	Correction of small errors in document
2.4	19/01/2018	SMALS	Adaptation of error messages (v.2.0.13 of EMSR).
2.5	22/05/2018	SMALS	Minor change in the KMEHR (v.2.0.15 of EMSR)
2.6	21/06/2018	SMALS	Main change in this version is the public key (ETK) thathas to be used to encrypt requests to EMSR. This corresponds with version v2.0.16 of EMSR.
2.7	02/10/2018	SMALS	Minor changes related to release 2.0.20.
2.8	12/10/2018	SMALS	Changes in the KMEHR Cookbook related to release 2.0.22 and TLS configuration.
2.9	5/09/2019	eHealth platform	Correct link to KMEHR information
2.10	25/09/2019	eHealth platform	Update
2.11	18/01/2021	eHealth platform	Correction (boolean)
2.12	29/07/2022	eHealth platform	§ 2.3 eHealth document references (updated) § 3.2 Status (added) § 5.1.5 WS-I Basic Profile (added) § 5.1.6 Tracing (added)

2. Introduction

2.1 Goal of the service

The purpose of this service is to provide authenticated ambulances services and hospitals with a set of methods for registering and consulting EMSR (Emergency Medical Service Registry) sheets and consulting SDS data.

GetSheet method will be available for hospitals and ambulances services. RegisterPartA, RegisterPartB, GetSds method will be available for ambulances services.

The registering of the sheet is done in two steps: the first part A is sent when the patient arrives at the hospital with available patient information and transaction I. Hereafter, within 5 days, part B is sent with all patient information and transaction II.

On the sheet consultation the system concatenates transaction I, II and adds SDS data (in particular timings) as transaction III. Patient information is always retrieved from part B.It is possible to separately get SDS data with the GetSds method in order to retrieve address for example.

2.2 Goal of the document

In this cookbook, we explain the structure and content aspects of the possible requests and the replies of EMSR web service. An example illustrates each of those messages. In addition, a list of possible errors can be found in this document.

This information should allow (the IT department of) an organization to develop and use the web service call.

Some technical and legal requirements must be met in order to allow the integration of EMSR web service in client applications.

This document is neither a development nor a programming guide for internal applications; eHealth partners always keep a total freedom within those fields. Nevertheless, in order to interact in a smooth, homogeneous and risk controlled way with a maximum of partners, eHealth partners must commit to comply with specifications, data format, and release processes described within this document. In addition, our partners in the health sector must also comply with the business rules of validation and integration of data within their own applications in order to minimize errors and incidents.

2.3 eHealth platform document references

On the portal of the eHealth platform, you can find all the referenced documents.[1]. These versions, or any following ones, can be used for the eHealth platform service.

ID	Title	Version	Date	Author
1	EMSR - KMEHR Message CookBook	3.0	25/03/2021	HFCSE
2	Glossary.pdf	1.0	01/01/2010	eHealth platform
3	STS HolderofKey - Cookbook	1.5	13/07/2022	eHealth platform
4	SOA – Error guide	1.0	10/06/2021	eHealth platform
4	Cookbook ETEE voor bekende bestemmeling / Cookbook ETEE destinataire connu	2.9	18/07/2022	eHealth platform

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

2.4 External document references

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	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
2	OASIS – Web services security – SAML Token Profile 1.1	***https://www.oasis-open.org/committees/download.php/16768/wssv1.1-spec-os-SAMLTokenProfile.pdf***	01/02/2006	OASIS Standard
3	Lijst met de AMBUREG-variabelen Liste des variables AMBUREG	***https://www.health.belgium.be/nl/richtlijnen-ambureg*** ***https://www.health.belgium.be/fr/directives-ambureg***	20/05/2019	FOD Volksgezondheid, Veiligheid van de Voedselketenen Leefmilieu SPF Santé Publique, Sécurité de la chaîne alimentaire et Environnement

3. Support

3.1 Helpdesk eHealth platform

3.1.1 Certificates

In order to access the secured eHealth platform environment you have to obtain an eHealth platform certificate, 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***

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.

3.3 End-To-End Encryption

In order to secure the information exchanged between hospitals and EMSR WS, most of requests and responses contain encrypted data. Encryption is performed with a public encryption key belonging to the recipient. This means that a request sent to EMSR WS has to use EMSR public encryption token and the response will use the public encryption key of the hospital that sent the request to encrypt the data (The public encryption token key is required in the request). The eHealth platform delivers the procedures to create a pair of private/public keys and the public encryption token.

Process of encryption upon sending of data (RegisterPartA, RegisterPartB):

1) Sign the KMEHR data to encrypt with your personal private key
2) Get the public key of The Federal Public Service of Health (EHP: 1990003302) with application id 'EMSR' from the ETK Depot.

The GetEtk Request should contain following search criteria:

```
<urn:Identifier>
    <urn:Type>EHP</urn:Type>
    <urn:Value>1990003302</urn:Value>
    <urn:ApplicationID>EMSR</urn:ApplicationID>
</urn:Identifier>
```

3) Encrypt the data with the public key above and using eHealth encryption libraries.
4) Sign again with your personal private key.

Process of decryption upon consulting of data (GetSheet):
1) Verify validity of signature
2) Decrypt KMEHR data using your private key.

Therefore, the GetSheet request has to contain your public key used to encrypt the GetSheet response.

In XML, the encrypted data is represented in base64 binary format, translating each byte of binary data into an ASCII string format. Thus from XML data, this representation has first to be decoded into byte using the base64 encoding scheme before being decrypted.

Note: While the base64 files are in ASCII format, they should still be encoded using the UTF-8 format.

3.4 TLS configuration of eHealth services

Since 18/11/2018 the SSL configuration of the eHealth services has been modified in production.

- The supporting infrastructure should comply to following technology :
  - TLS 1.2 (RFC5246)

This version of TLS is supported as of version 1.6.0 v111 of Java (published on 20-01-2016). To verify if your infrastructure support TLS 1.2 you can try connecting to following URL, and if a connection is established, this means your infrastructure can support TLS 1.2.

- ***https://etee.int.pub.ehealth.fgov.be/EtkDepot/v1/soap (integration)***

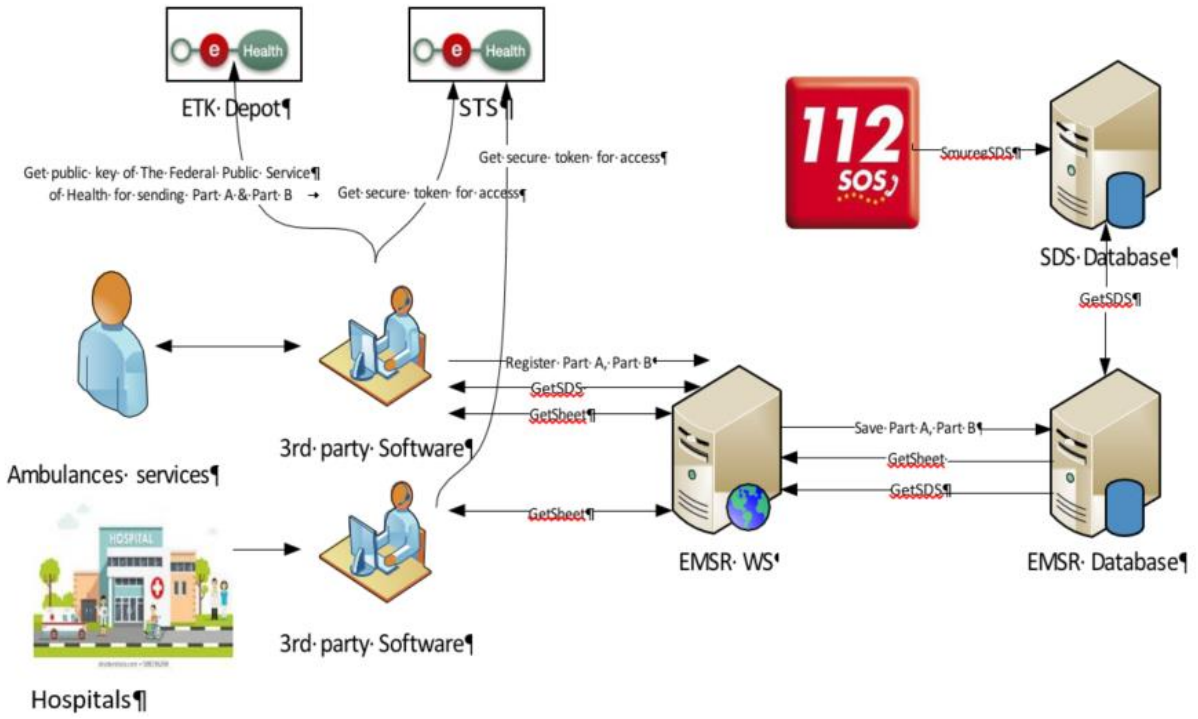
For the users of the .NET connectors, you can activate TLS 1.2 by setting following property:

- java.lang.System.setProperty ("https.protocols", "TLSv1,TLSv1.1,TLSv1.2");

```
class Program
{
    static void Main(string[] args)
    {
        java.lang.System.setProperty("https.protocols", "TLSv1,TLSv1.1,TLSv1.2");
```

For all questions regarding this configuration, please contact support@ehealth.fgov.be

4. Global overview



This global overview aims to show how the consultation web service is used.

Step 1. To use the Consultation WS, you have to contact the WS STS to get a secure token containing theidentification of the user (see 5.1.1 Use of the eHealth SSO solution and the STS CookBook).

Step 2. Once you have your secure token, you are able to use and contact the Consultation WS to call GetSdsor GetSheet method.

Step 3. When you call GetSheet method, you have to provide your public key in the request, in order for theresponse to be encrypted to you.

When your call has been sent, the system will respond to you with a response message (encrypted sheet or SDS info).

5. Step-by-step

The content of KMEHR part is described in the file “Lijst met de AMBUREG-variabelen”/”Liste des variables AMBUREG” (see section 2.4 ID 2). All validations of the KMEHR data are written in Schematron language. An archive with a small validation application can be downloaded from the eHealth EMSR support page.

5.1 Technical requirements

All the xml requests that are submitted to the WS must be encoded in the UTF-8 form.

5.1.1 Use of the eHealth SSO solution

For each WS accessed on the eHealth platform, authentication ensures that the requester is allowed access. eHealth certificates are used to trust the requester. In order to use EMSR Consultation, prior authentication has to be made on STS with the use of the eHealth Certificate and with specific parameters. An assertion will be generated that can then be used to make a call and access the EMSR Consultation WS.

The complete overview of the profile and a systematic implementation to start protecting a new application with SSO @ eHealth is described in the eHealth STS cookbook.

In order to implement a call to the eHealth STS you can reuse the implementation as provided in the "eHealth technical connector":

- ***https://www.ehealth.fgov.be/ehealthplatform/nl/service-ehealth-platform-services-connectors***
- ***https://www.ehealth.fgov.be/ehealthplatform/fr/service-ehealth-platform-services-connectors***

Nevertheless, eHealth implementations use standards and any other compatible technology (WS stack for the client implementation) can be used instead.

The attributes that need to be provided and the attributes that should be certified by the eHealth platform in order to obtain a token valid for EMSR Consultation services are described in section 5.1.2 (if EMSR Consultation is one by an ambulance service) or section 5.1.3 (if EMSR Consultation is done by a Hospital).

To access the EMSR Consultation WS, the response token must contain “true” for all of the certification attributes. If you obtain “false”, contact eHealth to verify that the requested test cases were correctly configured (See section 3).

5.1.2 EMSR Consultation by ambulance service

The SAML token request is secured with the eHealth certificate of the ambulance service. The certificate used by the Holder-Of-Key (HOK) verification mechanism is the same eHealth certificate. The needed attributes are the following (AttributeNamespace="urn:be:fgov:identification-namespace"):

- The NIHII number of the ambulance service:

  urn:be:fgov:ehealth:1.0:certificateholder:ambulanceservice:nihii-number and
  urn:be:fgov:ehealth:1.0:ambulanceservice:nihii-number

You must also specify which information by the eHealth platform has to assert:

- The NIHII number of the ambulance service (AttributeNamespace="urn:be:fgov:identification-namespace"): *urn:be:fgov:ehealth:1.0:certificateholder:ambulanceservice:nihii-number* and *urn:be:fgov:ehealth:1.0:ambulanceservice:nihii-number*
- The ambulance service must be a recognized ambulance service (AttributeNameSpace=”urn:be:fgov:certifiednamespace:ehealth”): *urn:be:fgov:ehealth:1.0:certificateholder:ambulanceservice:nihii-number:recognisedambulanceservice:boolean*

5.1.3 EMSR Consultation by hospital

The SAML token request is secured with the eHealth certificate of the hospital. The certificate used by the HOKverification mechanism is the same eHealth certificate. The needed attributes

are the following (AttributeNamespace="urn:be:fgov:identification-namespace"):

- The NIHII number of the hospital: *urn:be:fgov:ehealth:1.0:certificateholder:hospital:nihii-number* and *urn:be:fgov:ehealth:1.0:hospital:nihii-number*

You must also specify which information must be asserted by eHealth:

- The NIHII number of the hospital (AttributeNamespace="urn:be:fgov:identification-namespace"):*urn:be:fgov:ehealth:1.0:certificateholder:hospital:nihii-number* and *urn:be:fgov:ehealth:1.0:hospital:nihii-number*

The hospital must be a recognized hospital (AttributeNameSpace=”urn:be:fgov:certified-namespace:ehealth”): *urn:be:fgov:ehealth:1.0:certificateholder:hospital:nihii- number:recognisedhospital:boolean*

5.1.4 Security policies to apply

We expect that you 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 doesn’t arriveduring 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 or a SAML token issued by STS

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

The STS cookbook, explaining how to implement this security policy, can be found on the eHealth portal.

https://www.ehealth.fgov.be/ehealthplatform/nl/service-iam-identity-access-management (Dutch)

https://www.ehealth.fgov.be/ehealthplatform/fr/service-iam-identity-access-management (French)

5.1.5 WS-I Basic Profile 1.1

Your request must be WS-I compliant (See Chap 2.4 - External Document Ref).

5.1.6 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 Web Service

The WS presented in this cookbook is the Consultation WS of EMSR. The Consultation WS has one method: GetSheet. The GetSheet method returns KMEHR content.

The content of KMEHR part is described in the file “ambureg_variables_vxx.xlsx”. The EMSR Consultation webservice has the following endpoints:

- Integration environment: ***https://services-int.ehealth.fgov.be/EMSR/Consultation/v1***
- Acceptance environment: ***https://services-acpt.ehealth.fgov.be/EMSR/Consultation/v1***
- Production environment: ***https://services.ehealth.fgov.be/EMSR/Consultation/v1***

5.2.1 Method GetSheet

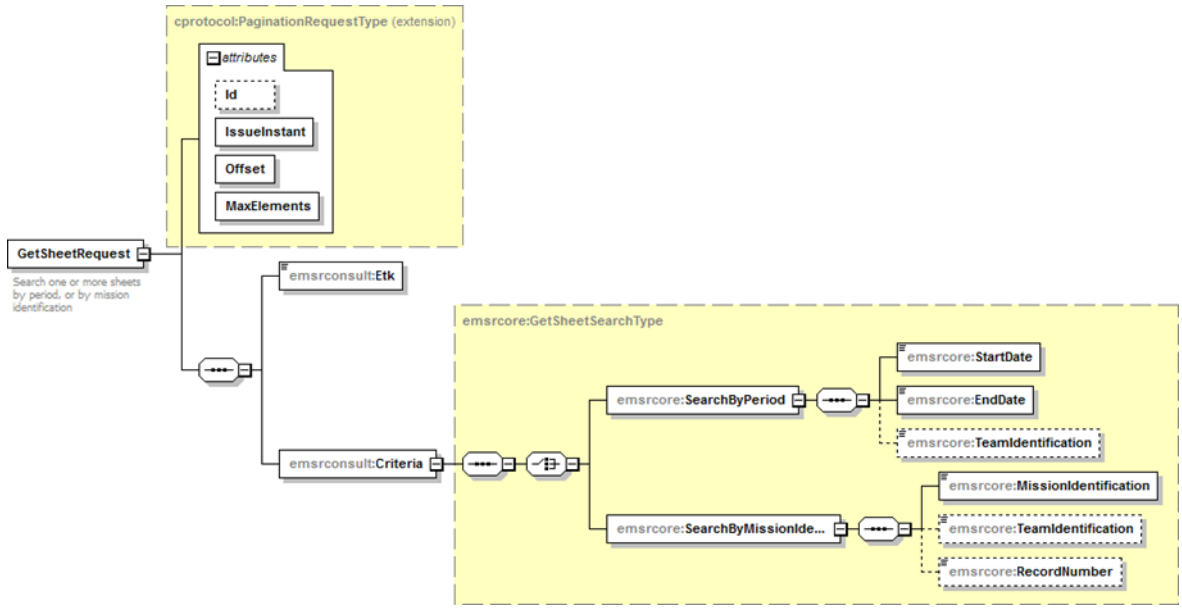
This WS returns all available information about one more EMSR sheet, which the user may access.

5.2.1.1 Request

It is possible to make a search of sheets by period between two dates or directly by MissionIdentification if this one is known. Please note that the MissionIdentification is not a mandatory field when registering the part A of a sheet, and therefore not all sheets have a MissionIdentification number.

Pagination

In the request Offset and MaxElements, attributes must be specified. Offset is set to 0, and MaxElements can be maximum 100. When 100 elements are returned, this can means more results are present. A second call can be made with Offset set to 100 and MaxElements to 100 to gather other results. This can also mean that search criteria were not correctly selected.

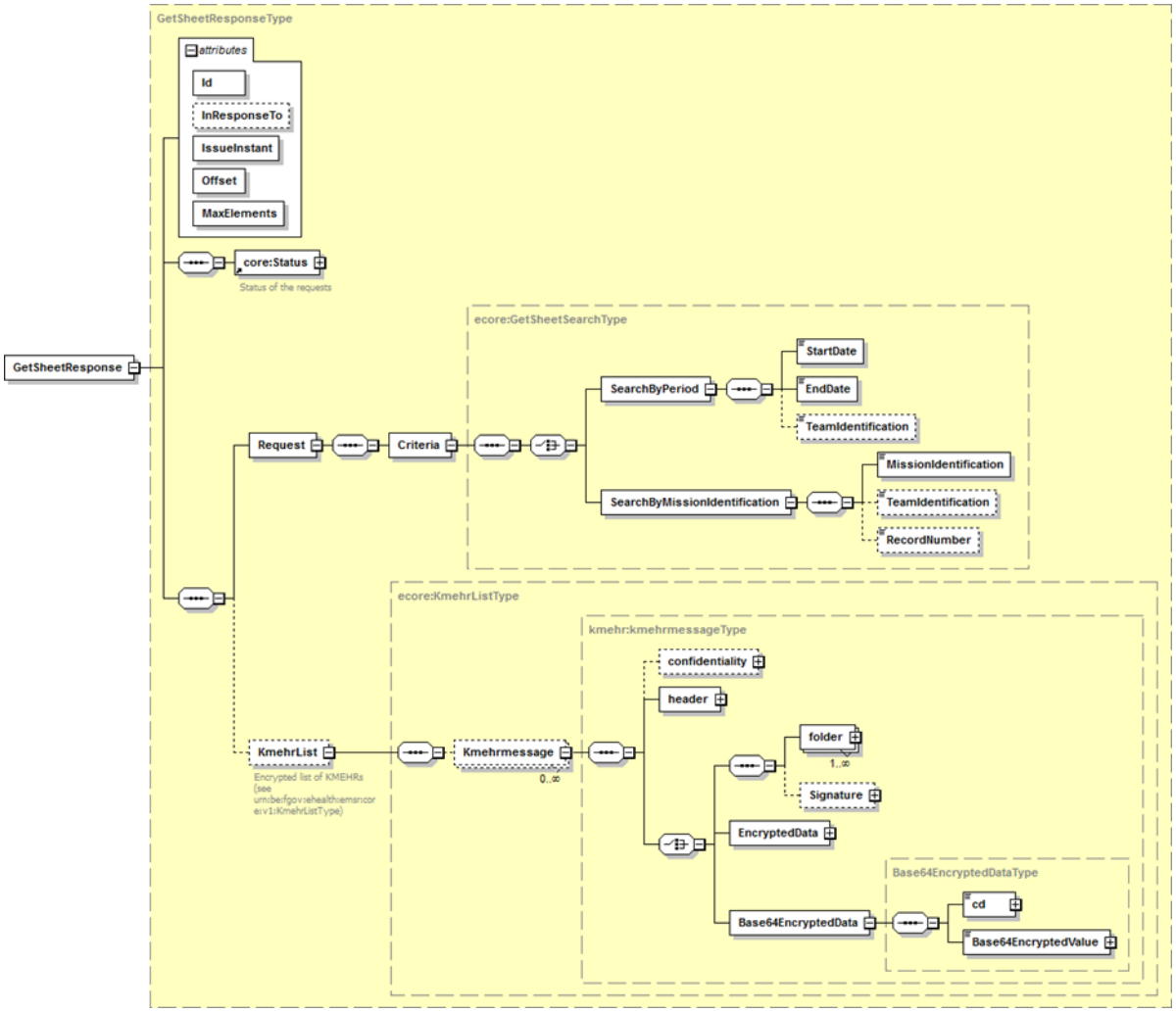


Field Name	Description	Attribute	Required
Id	Identifier of the request within the caller system.	Yes	No
IssueInstant	Date and time of the request.	Yes	Yes
Offset	Index of first element of response	Yes	Yes
MaxElements	Number of result per request (max 100)	Yes	Yes
Etk	The ETK which should be used to encrypt the response	No	Yes

StartDate	Beginning of the date range search	No	Yes
EndDate	End of the date range search	No	Yes
TeamIdentification	This is the unique identification number of the team in 9 characters.	No	No
MissionIdentification	Id of the mission	No	Yes
RecordNumber	This is the sheet number. It consists of max 20 characters.	No	No

5.2.1.2 Response

The status element description is detailed in 5.2.2.2 StatusType.



The KMEHR part is encrypted and described in detail in a separate document.

The response contains the submitted search criteria (for their description see above).

Field Name	Description	Attribute	Required
Id	Identifier of the response.	Yes	Yes
InResponseTo	Id attribute of the request	Yes	No
IssueInstant	Date and time of the response.	Yes	Yes
Offset	Index of first element of response	Yes	Yes

MaxElements	Number of result per request (max 100)	Yes	Yes
Status	See section 5.2.2.2 StatusType	No	Yes
KmehrList	Corresponds to a set of results for the requested search criteria. Each individual result is encrypted using ETEE and encoded base64. See section 5.2.2.1 KmehrListType.	No	No

5.2.1.3 Example

Request

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
      <soapenv:Header/>
      <soap:Body>
            <emsrcons:GetSheetRequest Id="bdc38ae62-3e7f-4f80-80f7-
c3e745500fa3" IssueInstant="2001-12-17T09:30:47Z" Offset="0"
MaxElements="100"...>
      <emsrcons:Etk>UjBsR09EbGhjZ0dTQUxNQUFBUUNBRU1tQ1p0dU1GUXhEUzhi</ems
rcons:Etk>
                  <emsrcons:Criteria>
                        <ecore:SearchByPeriod>
                  <ecore:StartDate>2001-12-17T09:30:47Z</ecore:StartDate>
                  <ecore:EndDate>2001-12-19T09:30:47Z</ecore:EndDate>

      <ecore:TeamIdentification>MAHOBO101</ecore:TeamIdentification>
                        </ecore:SearchByPeriod>
                  </emsrcons:Criteria>
            </emsrcons:GetSheetRequest>
      </soap:Body>
</soap:Envelope>
```

Response

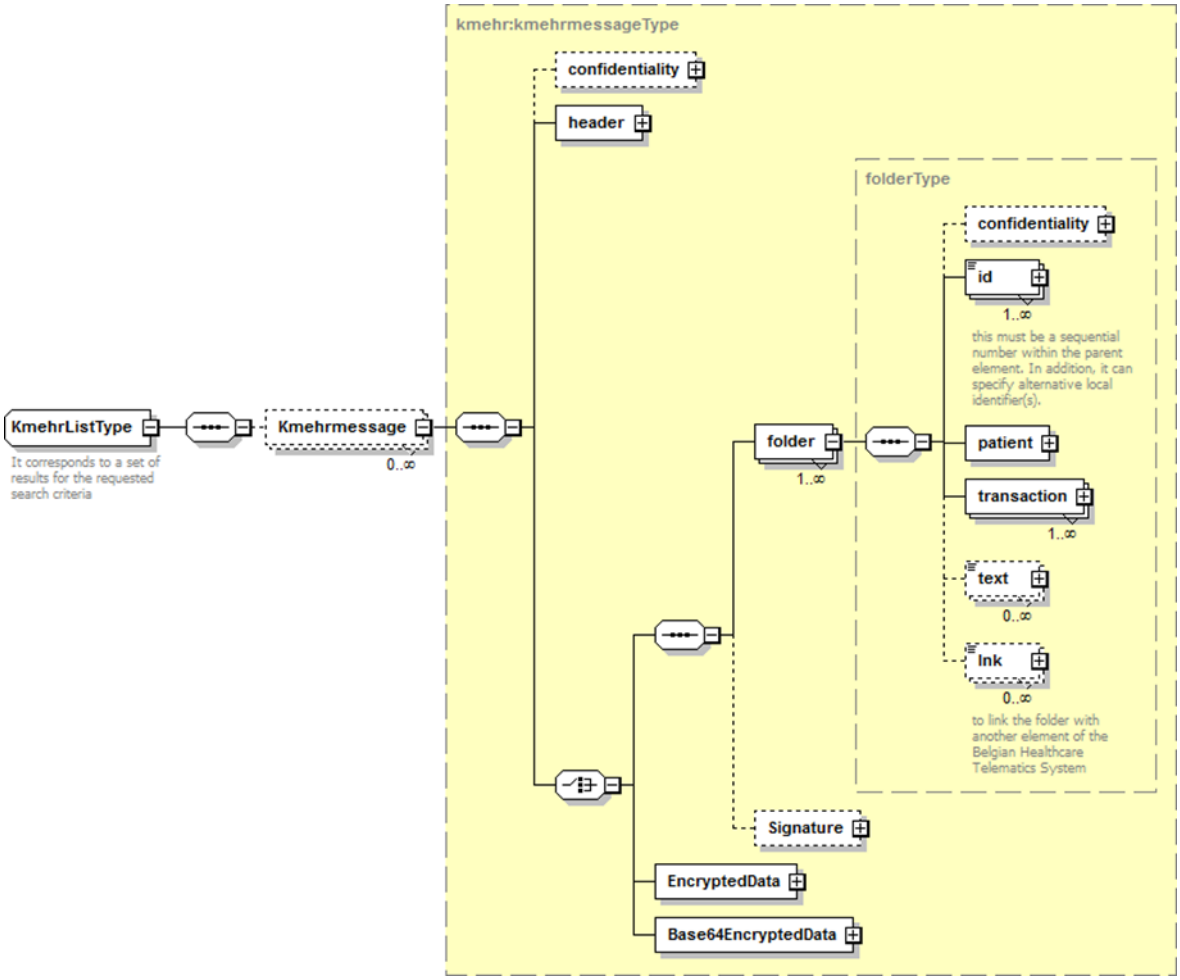
```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    • <soap:Body>
        <emsrcons:GetSheetResponse Offset="0"
MaxElements="100" Id="_de2754ca-83fe-41ce-9c72-9c3a7f586b38"
InResponseTo="bdc38ae62-3e7f- 4f80-80f7-c3e745500fa3"
IssueInstant="2016-04-07T10:09:48.288+02:00" ...>
            • <core:Status>
                • <core:StatusCode
Value="urn:be:fgov:ehealth:2.0:status:Success"/>
            • </core:Status>
            • <emsrcons:Request>
                • <emsrcons:Criteria>
                    • <ecore:SearchByPeriod>
            <ecore:StartDate>2001-12-
            17T09:30:47Z</ecore:StartDate>
            <ecore:EndDate>2001-12-19T09:30:47Z</ecore:EndDate>

    • <ecore:TeamIdentification>MAHOBO101</ecore:TeamIdentification>
                    • </ecore:SearchByPeriod>
                • </emsrcons:Criteria>
            • </emsrcons:Request>
            • <emsrcons:KmehrList>
            • <ecore:Kmehrmessage>
            <kmehr:header> see KMEHR CookBook </kmehr:header>
            • <kmehr:Base64EncryptedData>
            <kmehr:cd S="CD-ENCRYPTION-METHOD" SV="1.0">CMS</cd>
<kmehr:Base64EncryptedValue>UjBsR09EbGhjZ0dTQUxNQUFBUUNBRU1tQ1p0dU1GUXhEU
zhi<kmehr:Base64EncryptedValue>
            • </kmehr:Base64EncryptedData>
            • </ecore:Kmehrmessage>
            • <ecore:Kmehrmessage>
            <kmehr:header> see KMEHR CookBook </kmehr:header>
            • <kmehr:Base64EncryptedData>
            <kmehr:cd S="CD-ENCRYPTION-METHOD" SV="1.0">CMS</cd>
<kmehr:Base64EncryptedValue>xNQUFBUUNBRU1tQ1p0dU1GUXhEUzhi<kmehr:Base64En
cryptedValue>
            </kmehr:Base64EncryptedData>
            </ecore:Kmehrmessage>
            </emsrcons:KmehrList>
        </emsrcons:GetSheetResponse>
    </soap:Body>
</soap:Envelope>
```

5.2.2 Used types

5.2.2.1 KmehrListType

Corresponds to a set of results for the requested search criteria.

Each element of the list is a KMEHR message of standard KMEHR type: kmehrmessageType. Each element corresponds to one sheet. Each sheet individually is encrypted and the KMEHR content can be found under <Base64EncryptedData>. It is encrypted with the ETK provided by you in the GetSheetRequest.



5.2.2.2 StatusType

eHealth SOA service response is composed of an Status element. This element is used to indicate the status of the completion of the request. The status is represented by a StatusCode and optionally, the message describing the status. Additional detail gives extra information on the encountered business errors returned by the target service.

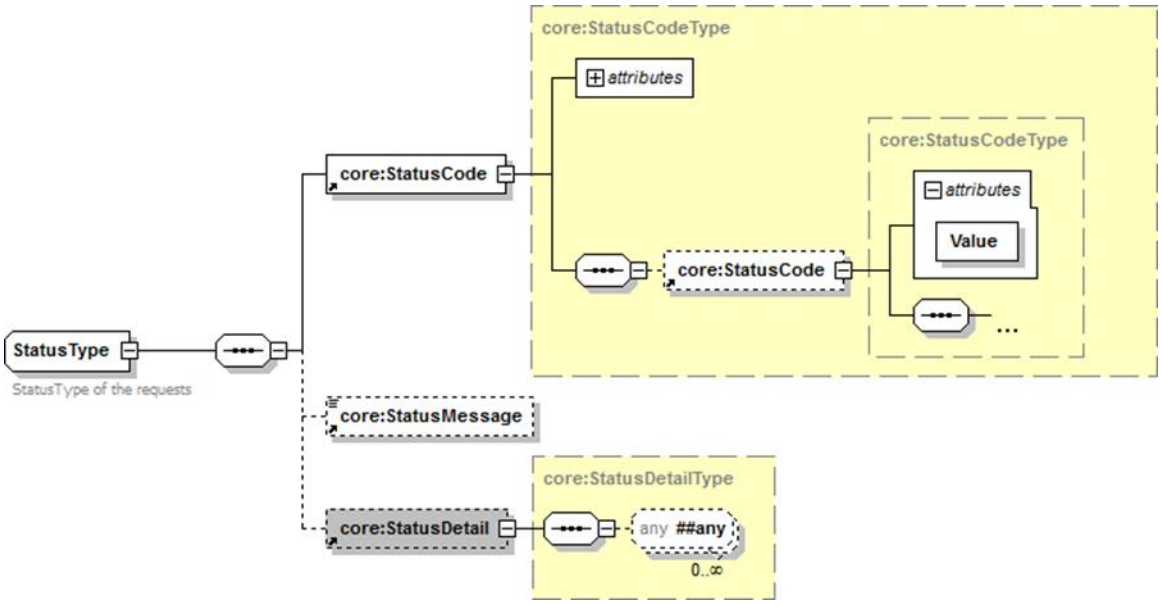


Figure1 – Status Type

Field Name	Description	Attribute	Required
StatusCode	See table further below for a list of possible values	No	Yes
StatusMessage	An optional message describing the error.	No	Yes
StatusDetail	The StatusDetail is defined as a free type, available for service to put any element in it to give extra information on the encounteredbusiness errors returned by the target service.	No	No

StatusCode is recursive; therefore, *StatusCode* (level 1) could be embedded by an optional sub *StatusCode* (sub level). Each *StatusCode* must have a value attribute and there must be at least a level 1 *StatusCode*.

The response returns at least Level 1 *StatusCode* with one of the following values:

URI	Description
'urn:be:fgov:ehealth:2.0:status:Success'	Completion of the request without errors.
'urn:be:fgov:ehealth:2.0:status:Requester'	Completion of the request with errors caused by the WS consumer.
'urn:be:fgov:ehealth:2.0:status:Responder'	Completion of the request with errors caused by the WS provider.

The optional Level 2 *StatusCode,* if returned, may have different values indicating specific cause of the error such as invalid input, missing input, and data not found etc.

URI	Description
'urn:be:fgov:ehealth:2.0:status:Intermediate'	Unknown error.
'urn:be:fgov:ehealth:2.0:status:InvalidInput'	Invalid input error.
'urn:be:fgov:ehealth:2.0:statusMissingInput'	Missing input.
'urn:be:fgov:ehealth:2.0:status:DataNotFound'	No results for the request.
'urn:be:fgov:ehealth:2.0:status:RequestDenied'	Unauthorized request (business level).
'urn:be:fgov:ehealth:2.0:status:RequestUnsupported'	Service does not support the request.

Example

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns8:GetSheet ... Offset="0" MaxElements="100" Id="_14cc837e-de41-
4b38-b23a-f19a91148a83" InResponseTo="bb16782e9-9cea-4af4-8ce4-
e1abe70a9687" IssueInstant="2016-04-07T10:40:57.881+02:00">
      <ns4:Status>
        <ns4:StatusCode Value="urn:be:fgov:ehealth:2.0:status:Success">
          <ns4:StatusCode
Value="urn:be:fgov:ehealth:2.0:status:InvalidInput"/>
        </ns4:StatusCode>
        <ns4:StatusMessage>KMEHR rule 22.3 validation
error.</ns4:StatusMessage>
      </ns4:Status>
    </ns8:GetSheet>
  </soap:Body>
</soap:Envelope>
```

See "Section 8 Error and failure messages" for further description of StatusCode used in this service.

6. Risks and security

6.1 Security

6.1.1 Business security

In case the development adds a use case based on an existing integration, the eHealth platform must be informed at least one month in advance. A detailed estimate of the expected load is necessary to be able to ensure an effective capacity management.

When technical issues occur on the WS, the partner can obtain support from the contact centre (see Chap 3)

If the eHealth platform should find a bug or vulnerability in its software, the partner must update his application with the latest version of the software, within ten (10) business days.

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

6.1.2 Web service

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

- SSL one way
- "Time-to-live" of the message: one minute.
- Signature of the timestamp, body and binary security token. This allows the eHealth platform to verify the integrity of the message and the identity of its author.
- No encryption on the message.

6.1.3 The use of username, password and token

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

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 platform 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 information needed to integrate is published on the portal of the eHealth platform.

Upon request and depending on the case, the eHealth platform provides you with a **test case** 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 to access the acceptance environment of the eHealth platform. From this moment, you start the integration and acceptance tests. The eHealth platform suggests testing during minimum one month.

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

Once a release date has been agreed on, the eHealth platform prepares the connection to the production environment and provides the partner with the necessary information. During 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***.

7.1.4 Operational follow-up

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

7.2 Test cases

The eHealth platform recommends performing tests for all of the following cases:

- GetSheet search by date
- GetSheet search by MissionIdentification

Most of the time it is not useful to retry a call when something went wrong. You should rather change your search criteria.

Loops and load tests are strictly prohibited!

8. Error and failure messages

There are three different possible types of response:

- If there are no technical or business errors, a business response is returned.
- If a business error occurred, it is contained in a business response that undergoes a regular transformation[2] 2 (see chapter 8.1 Business errors).
- In the case of a technical error, a SOAP fault exception is returned (see chapter 8.2).

8.1 Business errors

See 5.2.2.2 StatusType for description of the StatusCode mechanism.

Business errors are forwarded without any transformation (they are treated as regular business responses). These error codes first indicate a problem in the arguments sent.

Error codes originating from the eHealth platform:

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

StatusCode	Message	Solution
urn:be:fgov:ehealth:2.0:status:Success (level 1) urn:be:fgov:ehealth:2.0:status:DataNotFound (level 2)	No results for the request	Change one of the search criteria. Reduce the number of search criteria.
urn:be:fgov:ehealth:2.0:status:Requester (level 1) urn:be:fgov:ehealth:2.0:status:InvalidInput (level 2)	The Offset attribute cannot be negative	Offset must be >= 0. It can be higher than MaxElements
urn:be:fgov:ehealth:2.0:status:Requester (level 1) urn:be:fgov:ehealth:2.0:status:InvalidInput (level 2)	The MaxElements attribute is too high	MaxElements may not exceed 100 items per request
urn:be:fgov:ehealth:2.0:status:Requester (level 1) urn:be:fgov:ehealth:2.0:status:InvalidInput (level 2)	The MaxElements attribute cannot be negative or zero	MaxElements must be >=0
urn:be:fgov:ehealth:2.0:status:Requester (level 1) urn:be:fgov:ehealth:2.0:status:RequestDenied (level 2)	Combination of team identification and intervention number does not exist in SDS-database	There is no SDS sheet matching the search criteria entered.

[2] Please refer to the paragraph 5.2.2.2

urn:be:fgov:ehealth:2.0:status:Requester (level 1) urn:be:fgov:ehealth:2.0:status:InvalidInput (level 2)	The content of the Base64EncryptedData cannot be decrypted	The encrypted part of the request (i.e. the folder part of the KMEHR) could not be decrypted because it is not encrypted correctly for EMSR (see section 3.1).

8.2 Technical errors

Technical errors are errors inherent to the internal working of a web service. They are returned as SOAP Faults.

The SOA Standard for Error handling specifies a structure for SystemError and BusinessError, thrown as SOAP Faults.

A SystemError MUST be thrown when a system failure occurred. It is not related to the business of the service. The SOA system error structure is as follows:

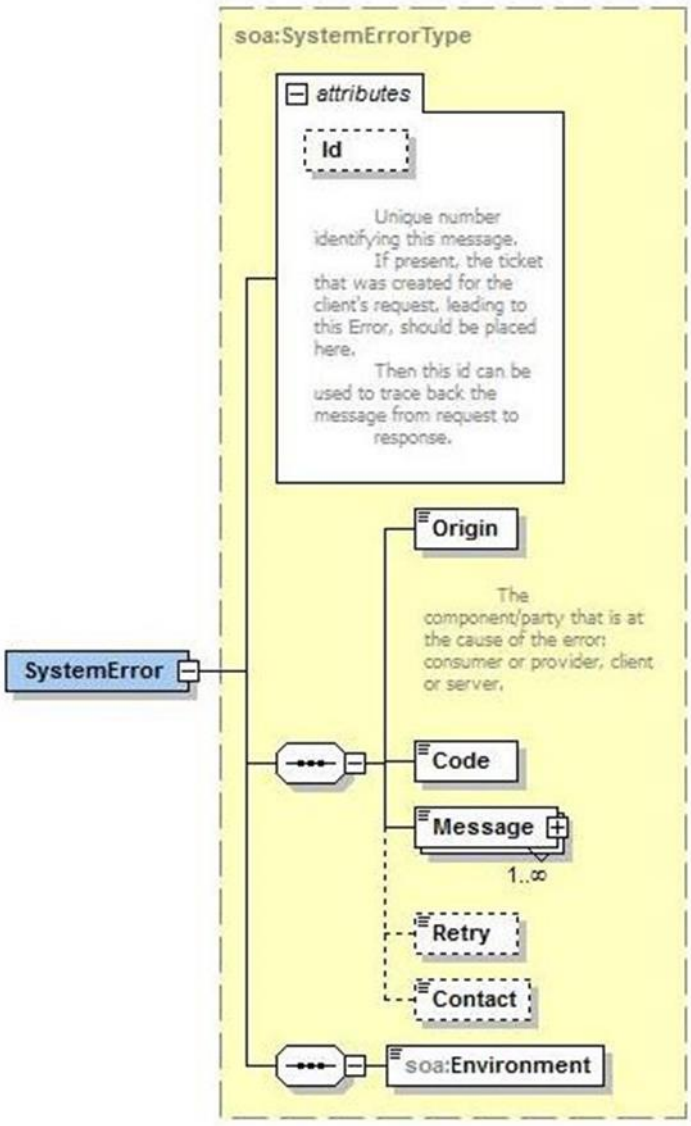


Figure 3 SystemError

The SystemError element MUST contain a unique Id attribute for tracing. The Origin MUST be set to Server or Provider.

Retry SHOULD be set to true if the consumer can try again immediately without interventions.

Example

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
   <soapenv:Body>
      <soapenv:Fault>
         <faultcode>soapenv:Server</faultcode>
         <faultstring>SOA-02002</faultstring>
         <detail>
            <soa:SystemError Id="9E0-00000P1-00-C" xmlns:soa="urn:be:fgov:ehealth:errors:soa:v1">
               <Origin>Server</Origin>
               <Code>SOA-02002</Code>
               <Message xml:lang="en">Service is temporarily not available. Please contact service
desk.</Message>
               <Retry>true</Retry>
               <soa:Environment>Test</soa:Environment>
            </soa:SystemError>
         </detail>
      </S:Fault>
   </soapenv:Body>
</soapenv:Envelope>
```

The SOAP Fault element has the following sub elements:

Element name	Descriptions	Required
faultcode	A code for identifying the fault	Yes
faultstring	A human readable explanation of the fault	Yes
faultactor	Information about who caused the fault to happen (the origin)	No
detail	Holds application specific error information related to the Body element. For example, it could include a java stack trace or any other kind of trace, used internally, to document on the cause ofthis error.	No

The default SOAP faultcode values are defined in an extensible manner that allows for new SOAP fault code values to be defined while maintaining backwards compatibility with existing fault code values.

Element name	Descriptions
versionMismatch	Found an invalid namespace for the SOAP Envelope element.
mustUnderstand	An immediate child element of the Header element, with the mustUnderstand attribute set to "1", was not understood.
client	The message was incorrectly formed or contained incorrect information.
server	There was a problem with the server so the message could not proceed.

Description of the possible SOAP fault exceptions:

Error code	Component	Description	Solution/Explanation
SOA-00001	Undefined	Service error	This is the default error sent to the consumer in case more details are unknown.
SOA-01001	Consumer	Service call not authenticated	From the security information provided: • or the consumer could not be identified • or the credentials provided are not correct
SOA-01002	Consumer	Service call not authorized	The consumer is identified and authenticated but isnot allowed to call the given service.
SOA-02001	Provider	Service not available. Please contact servicedesk	An unexpected error has occurred: • Retries will not work • Service desk may help with root cause analysis
SOA-02002	Provider	Service temporarily not available. Please try later	An unexpected error has occurred: • Retries should work • If the problem persists service desk may help
SOA-03001	Consumer	Malformed message	This is default error for content related errors in caseno more details are known.
SOA-03002	Consumer	Message must be SOAP	Message does not respect the SOAP standard
SOA-03003	Consumer	Message must containSOAP body	Message respects the SOAP standard, but body is missing
SOA-03004	Consumer	WS-I compliance failure	Message does not respect the WS-I standard
SOA-03005	Consumer	WSDL compliance failure	Message is not compliant with WSDL inRegistry/Repository
SOA-03006	Consumer	XSD compliance failure	Message is not compliant with XSD inRegistry/Repository
SOA-03007	Consumer	Message content validation failure	From the message content (conform XSD): • Extended checks on the element format failed • Cross-checks between fields failed