# EMSR v2 Registration
# Cookbook
# 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 | 05/01/2021 | SMALS | First version |
| 1.1 | 24/03/2021 | SMALS | Deletion of method getSDS from cookbook |
| 1.2 | 03/08/2022 | eHealth platform | § 2.3 eHealth document references (updated)<br>§ 3.1 Support (updated)<br>§ 5.3.3 WS-I Basic Profile<br>§ 5.3.4 Tracing (updated) |
| 1.3 | 22/01/2024 | SMALS | Register part A & B shorter registration deadline |

# 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 Emergency Medical Service Registry (EMSR) sheets and consulting SDS data.

GetSheetList and GetSheet method will be available for hospitals and ambulances services.

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

The sheet registering is a 2-step process.

1 PartA is sent when a patient arrives at an hospital. It contains any information about the patient available at the time of arrival.

2. PartB is sent within 5 days. This step will provide the complete identification information about the patient.

On sheet consultation the system concatenates partA, partB and SDS data (most recent SDS data) to generate the intervention sheet. Patient information from partB will replace the patient information of part B.

It is possible to separately get SDS data with the GetSds method (method of EMSR consultation) 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 WS. 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 WS call.

Some technical and legal requirements must be met in order to allow the integration of EMSR WS 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 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 | EMSR v2 - KMEHR Message CookBook | 3.1 | 25/03/2021 | HFCSE |
| 2 | Glossary.pdf | 1.0 | 01/01/2010 | eHealth |
| 3 | STS HolderofKey - Cookbook | 1.6 | 25/01/2023 | eHealth |
| 4 | Cookbook end-to-end vercijfering voor bekende bestemmeling / encryption end-to-end vers destinataire connu | 2.9 | 18/07/2022 | eHealth |

## 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 | 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 |
| 2 | Lijst met de AMBUREG-variabelen<br>Liste des variables AMBUREG | ***https://www.health.belgium.be/nl/richtlijnen-ambureg***<br>***https://www.health.belgium.be/fr/directives-ambureg*** | 20/05/2019 | FOD Volksgezondheid, Veiligheid van de Voedselketen en Leefmilieu<br>SPF Santé Publique, Sécurité de la chaîne alimentaire et Environnement |
| 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 |

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

# 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 ambulances/hospitals and EMSR WS, most 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 ambulance/hospital that sent the request to encrypt the data (The public encryption token key is required in the request). eHealth platform provides the procedures to create a pair of private/public keys and the public encryption token.

<urn:Value>1990003302</urn:Value>

- 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 librairies.

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.

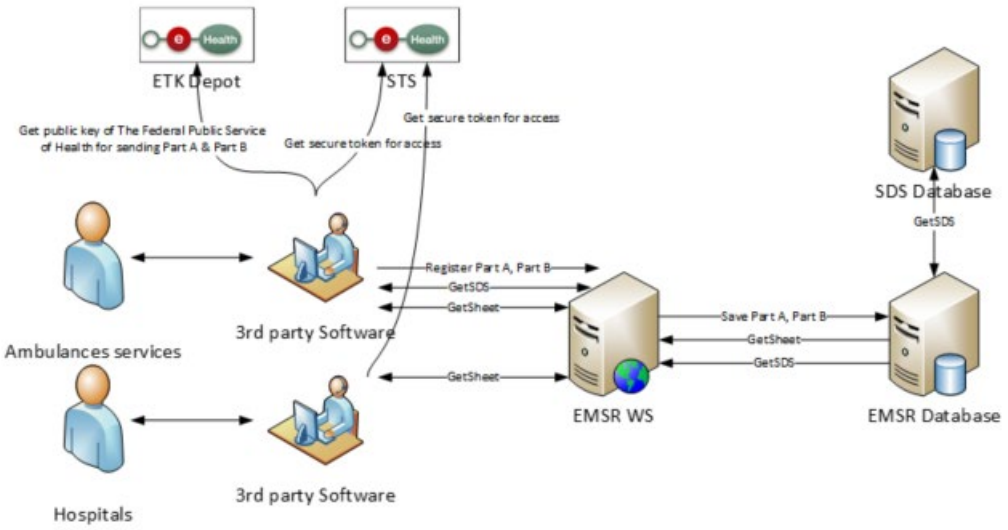
Therefore, the GetSheet request has to contain your public key that will be used to encrypt the GetSheet response.

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.

# 4. Global overview

To encrypt the content of your message, you have to call method Get ETK of the ETK Depot WS.




This global overview aims to show how the registration WS is used.

Step 1. To use the Registration WS, you have to contact the WS STS to get a secure token containing the identification of the user (see 5.2 Use of the eHealth SSO solution and the STS Cookbook).

Step 2. To encrypt the content of your message, you have to call method Get ETK of the ETK Depot Web Service (synonymous to the ETEE Webservice, **see ETEE for known recipient Cookbook on the eHealth Portal**) in order to get the public key of the recipient and use the Crypto Library.

See 3.3 End-to-End Encryption for more detailed information.

Step 3. Once you have your secure token, you are able to use and contact the Registration service to register part A or part B.

Step 4. When your call has been sent, the system will respond with a response message.

# 5. Step-by-step

## 5.1 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 provides 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, using eHealth encryption libraries.
4) Sign again with your personal private key

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 bytes 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.

## 5.2 Technical requirements

All the xml requests submitted to the WS must be encoded in the UTF-8 format.

## 5.3 Use of the eHealth SSO solution

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

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

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

The attributes to be provided and certified by eHealth in order to obtain a token valid for EMSR Registration services are described in following section. To access the EMSR Registration WS, the response token must contain “true” for all of the certification attributes. If you obtain “false”, contact eHealth to verify whether the requested test cases were correctly configured (See section 3).

### 5.3.1 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 required 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 must be asserted by the eHealth platform:

- 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*

### 5.3.2 Security policies to apply

*See section 3.2 for the update in the TLS configuration.*

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 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 or a SAML token issued by STS

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

The STS cookbook explains how to implement this security policy can be obtained. It can be found on the eHealth portal.

### 5.3.3 WS-I Basic Profile 1.1

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

### 5.3.4 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***

## 5.4 Web service

This WS handles the collection of data for an intervention. There are four different methods on the Registration WS. Two of them will be used to register the part A and the part B of the intervention, one will allow users to get one or more SDS records and finally one will allow the user to delete an existing intervention.

The content of KMEHR part is described in the file “ambureg_variables_vxx.xlsx”.

The EMSR Registration webservice has the following endpoints:

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

### 5.4.1 Method RegisterPartA & RegisterPartB

This method is a collecting method for the first and the second part of an intervention. It will take a KMEHR as input and will acknowledge the reception of the message. If the KMHEHR is not correct, an error will be returned with the invalided rule and an explanation of the rule.

When ‘Registering’ the different parts, always integrate the variables of the **patient** (under folder element) already sent in part A in the KMEHR message header. The MissionIdentification number is not mandatory when registering the part A of the sheet. In the part B however, it is mandatory and the mission must be assigned to the team that registers the part B.

**Once the part B has been sent, the sheet is considered complete.** Only the patient and the hospital of destination can be changed after PartB is sent. Beware that changing the hospital of destination can only be done once.

The registration of part A or part B can only be done up to 3 months after the call date (call date can be found in the SDS).

It is accomplished by correcting the following fields in a registerPartB request:

- 29.1 patient.patientNationalNumber
- 29.2 patientName
- 30 patientFirstname
- 31.1 patient.patientInterventionAddrFlag
- 31.4 patient.patientAddressAbroadFlag
- 31.4.1 patient.patientAddressCountryCode
- 31.5 patient.patientZipCode
- 31.6 patient.patientLocality
- 31.8 patient.patientStreet
- 31.9 patient.patientStreetNo
- 31.10 patient.patientbox

- 31.10.1 patientPatientForeignAdress/FreeText
- 31.11 invoicingaddresspatientaddressFlag
- 31.12 invoicing.name
- 31.13 invoicing.Firstname
- 31.14 invoicing.Street
- 31.15 invoicing.Streetno
- 31.16 invoicing.Zipcode
- 31.17 invoicing.Locality
- 31.18 invoicing.box
- 31.19 invoicing.AbroadFlag
- 31.19.1 invoicing.CountryCode
- 31.20 invoicing.abroadFreeText
- 32.1 patient.patientDob
- 33 patient.patientSexCode

For the hospital of destination, the following fields are editable once:

- 41.1 hospitalcode
- 41.1.1 hospitalsiteas

**The folder also needs to contain the correct namespaces:**

- xmlns="http://www.ehealth.fgov.be/standards/kmehr/schema/v1"
- xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
- xsi:schemaLocation="http://www.ehealth.fgov.be/standards/kmehr/schema/v1 xsd-kmehr-1.17.1/ehealth-kmehr/XSD/kmehr_elements-1_17.xsd">

```
<folder xmlns="http://www.ehealth.fgov.be/standards/kmehr/schema/v1"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.ehealth.fgov.be/standards/kmehr/schema/v1 xs
d-kmehr-1.17.1/ehealth-kmehr/XSD/kmehr_elements-1_17.xsd">
    <id S="ID-KMEHR" SV="1.0">1</id>
    <patient>
        <id S="INSS" SV="1.0">65406500000</id>
        ...
    </patient>
</folder>
```

### 5.4.1.1 Request

The request mainly consists of one element: the encrypted KMEHR message.

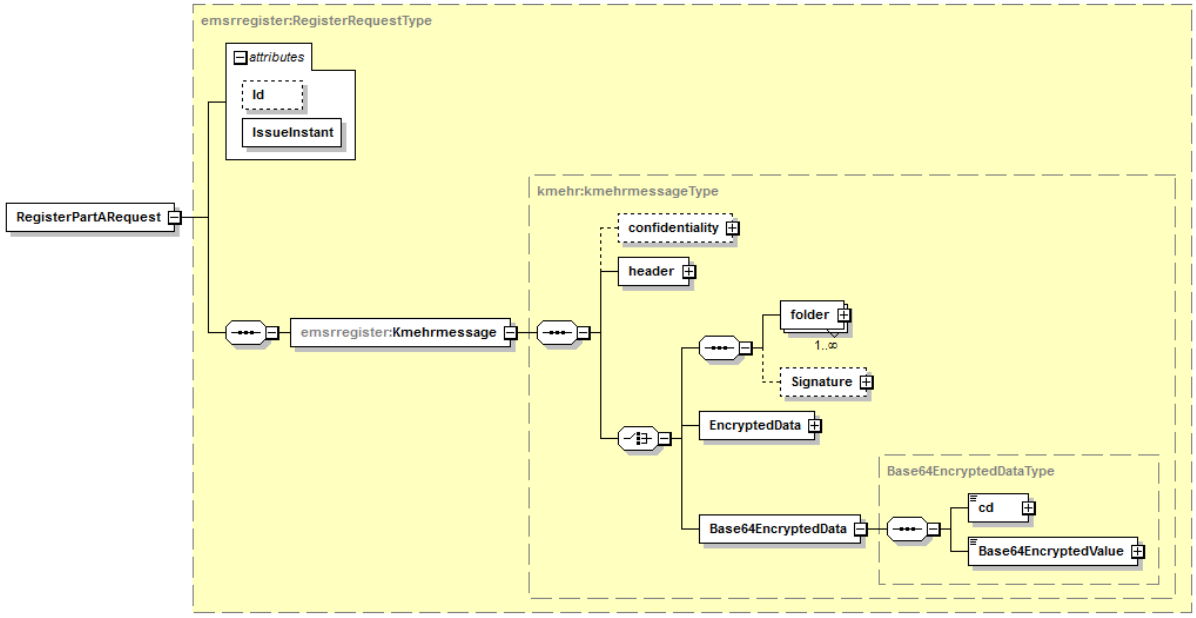



**Figure 1: RegisterPartARequest**

| 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 |
| Kmehrmessage | The KMEHR message contains at least header and folder. **The folder is contained in the Base64EncryptedData,** while the header is not encrypted in the KMEHR message. | No | Yes |
| Base64EncryptedValue | This element contains a KMEHR message, encrypted using ETEE (see ETEE CookBook) and encoded Base64. **The Base64EncryptedValue contains only the folder element of the KMEHR message.** | No | Yes |

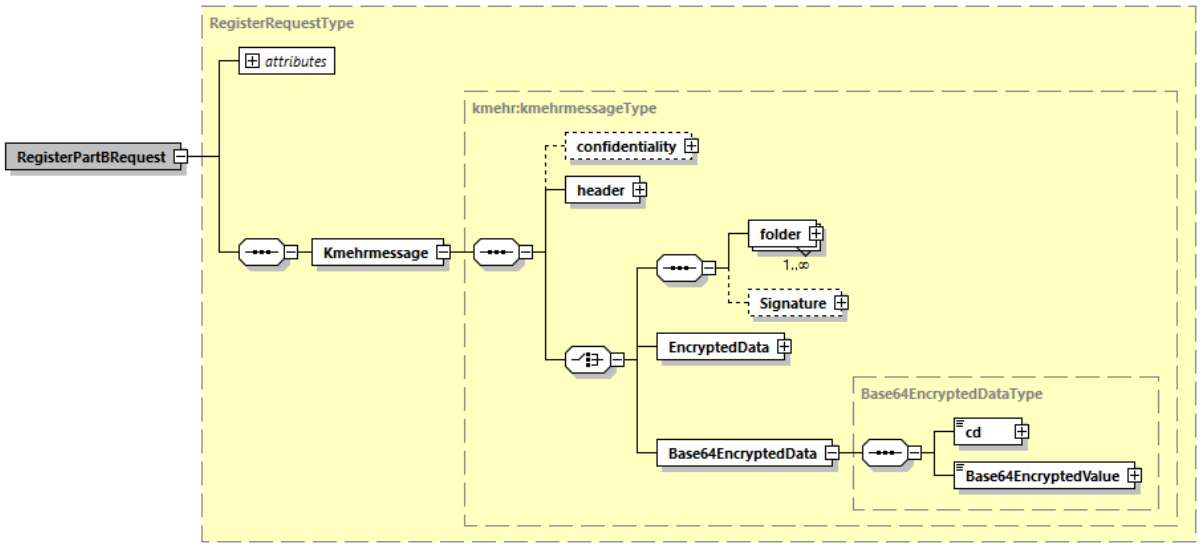



**Figure 2: RegisterPartBRequest**

| Id | Identifier of the request within the caller system. | Yes | No |
|---|---|---|---|
| IssueInstant | Date and time of the request. | Yes | Yes |
| Kmehrmessage | The KMEHR message contains at least header and folder. **The folder is contained in the Base64EncryptedData,** while the header is not encrypted in the KMEHR message. | No | Yes |
| Base64EncryptedValue | This element contains a KMEHR message, encrypted using ETEE (see ETEE CookBook) and encoded Base64. **The Base64EncryptedValue contains only the folder element of the KMEHR message.** | No | Yes |

### *5.4.1.2 Response*

The response is an acknowledgement. The status element description is detailed in 5.4.2.1 StatusType.

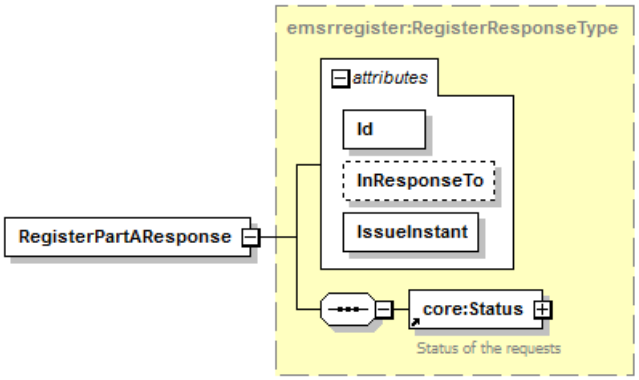



**Figure 3: RegisterPartAResponse**

| 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 |
| Status | | See section 5.4.2.1 StatusType | No | Yes |

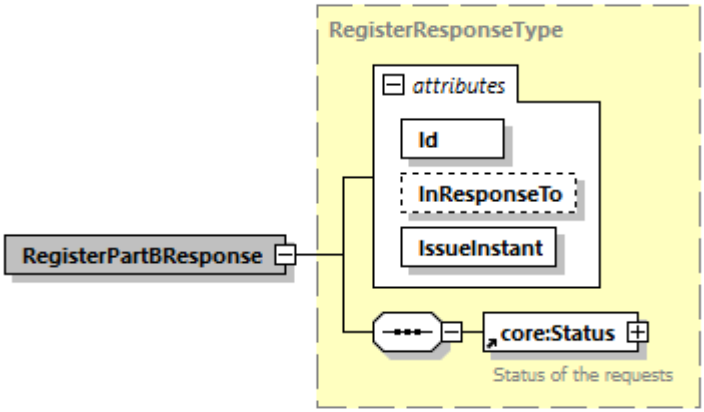



**Figure 4: RegisterPartAResponse**

| 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 |
| Status | See section 5.4.2.1 StatusType | No | Yes |

### *5.4.1.3 Example*

Request:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Header/>
    <soap:Body>
        <emsrregister:RegisterPartARequest Id="bdc38ae62-3e7f-4f80-80f7-
c3e745500fa3" IssueInstant="2001-12-17T09:30:47Z"...>
            <ecore:Kmehrmessage>
            <!-
- Unecrypted part of the KMEHR (contains the header of the KMEHR) -->
                <kmehr:header> see KMEHR CookBook </kmehr:header>
            <!-
- Encrypted part of the KMEHR (contains the folder of the KMEHR)-->
                <kmehr:Base64EncryptedData>
                    <kmehr:cd S="CD-ENCRYPTION-METHOD" SV="1.0">CMS</cd>
```

```
        <kmehr:Base64EncryptedValue>fAkEnUmBeRrrrrrjBsR09EbGhjZ0dTQUxNQUFBUUNB
RU1tQ1p0dU1GUXhEUzhi<kmehr:Base64EncryptedValue>
                </kmehr:Base64EncryptedData>
            </ecore:Kmehrmessage>
        </emsrregister:RegisterPartARequest>
    </soap:Body>
</soap:Envelope>
```

Response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <emsrregister:RegisterPartAResponse 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>
      </emsrregister:RegisterPartAResponse>
   </soap:Body>
</soap:Envelope>
```

### 5.4.2 Used Types

#### *5.4.2.1 StatusType*

eHealth SOA service response contains a *Status* element which 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 details give extra information on the encountered business errors returned by the target service.

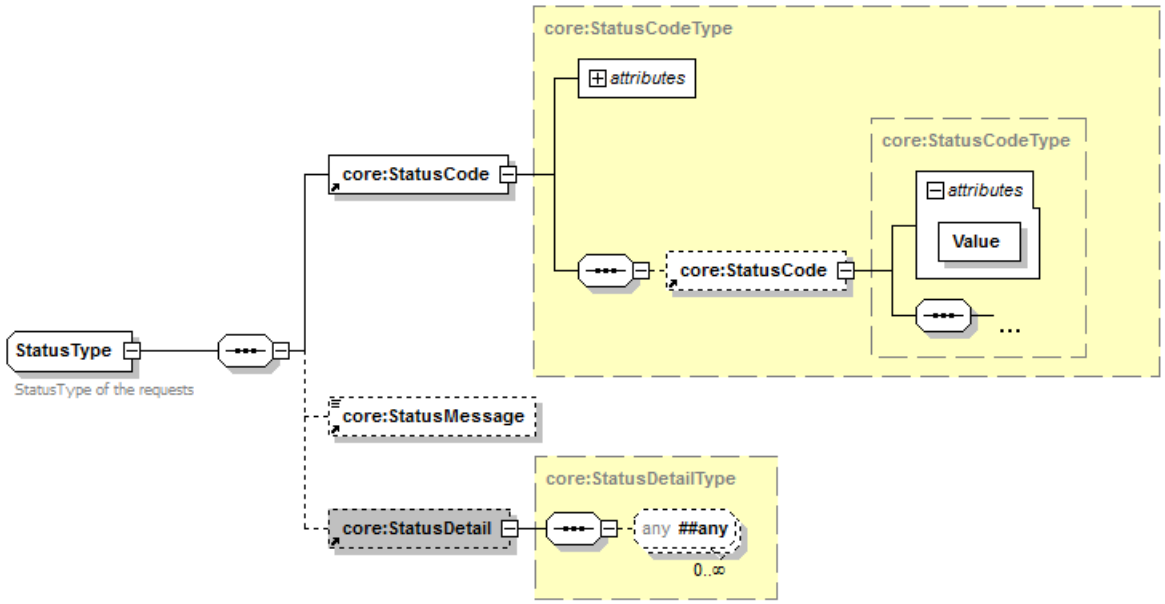



**Figure 5: StatusType**

| Field Name | Description | Attribute | Required |
|---|---|---|---|
| StatusCode | See table 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 add any element to give extra information on the encountered business 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, 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:StatusMessag
e>
      </ns4:Status>
    </ns8:GetSheet>
  </soap:Body>
</soap:Envelope>
```

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

### 5.4.3 DeleteSheet

#### *5.4.3.1 Request*

The DeleteSheet request is executed using the RecordNumber of a sheet. It can be used on a sheet part A but also on a sheet with part A and B.

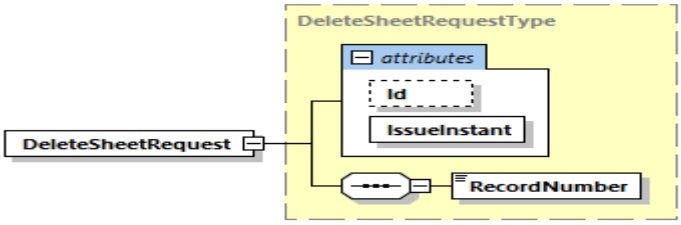



Figure 6: DeleteSheetRequest

| 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 |
| RecordNumber | Sheet number which consists of max 20 characters. | No | Yes |

### *5.4.3.2 Response*

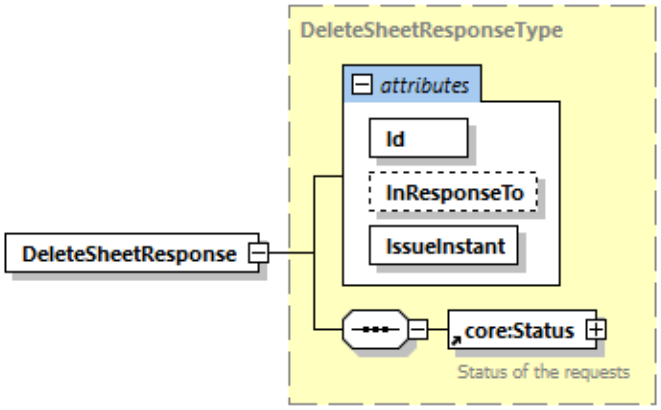



**Figure 7: DeleteSheetResponse**

| 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 |
| Status | See section 5.4.2.1 StatusType | No | Yes |

#### *5.4.3.3 Example*

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xm
lns:urn="urn:be:fgov:ehealth:emsr:registration:v2">
   <soapenv:Header/>
   <soapenv:Body>
      <urn:DeleteSheetRequest Id="id-
FBFE19453D50A4BFE61491476862891174" IssueInstant="2020-07-
31T09:18:48.288+02:00">
         <urn:RecordNumber>U777RPUM0GUQ6</urn:RecordNumber>
      </urn:DeleteSheetRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns4:DeleteSheetResponse Id="_5f9f8671-d4d9-4a0f-9e88-
6d26d1b29482" InResponseTo="id-
FBFE19453D50A4BFE61491476862891174" IssueInstant="2020-07-
31T09:18:20.875+02:00" xmlns:ns10="urn:be:fgov:ehealth:monitoring:protocol:v2"
 xmlns:ns2="http://www.w3.org/2000/09/xmldsig#" xmlns:ns3="http://www.w3.org/2
001/04/xmlenc#" xmlns:ns4="urn:be:fgov:ehealth:emsr:registration:v1" xmlns:ns5
="urn:be:fgov:ehealth:commons:core:v2" xmlns:ns6="urn:be:fgov:ehealth:commons:
protocol:v2" xmlns:ns7="http://www.ehealth.fgov.be/standards/kmehr/schema/v1"
xmlns:ns8="urn:be:fgov:ehealth:emsr:core:v1" xmlns:ns9="urn:be:fgov:ehealth:er
rors:service:v1">
         <ns5:Status>
            <ns5:StatusCode Value="urn:be:fgov:ehealth:2.0:status:Success"/>
         </ns5:Status>
      </ns4:DeleteSheetResponse>
   </soap:Body>
</soap:Envelope>
```

### 5.4.4 AnonymizePatientData

The AnonymizePatientData request erases the identity information of a patient from a sheet.

The following information are erased from the sheet (for more information about the fields, please check external document 4, “Lijst met de AMBUREG-variabelen”) :

| | |
|---|---|
| 29.1 | patient.patientNationalNumber |
| 29.2 | patientName |
| 30 | patientFirstname |
| 31.1 | patient.patientInterventionAddrFlag |
| 31.4 | patient.patientAddressAbroadFlag |
| 31.4.1 | patient.patientAddressCountryCode |

31.5 patient.patientZipCode
31.6 patient.patientLocality
31.8 patient.patientStreet
31.9 patient.patientStreetNo
31.10 patient.patientbox
31.10.1 patientPatientForeignAdress/FreeText
31.11 invoicingaddresspatientaddressFlag
31.12 invoicing.name
31.13 invoicing.Firstname
31.14 invoicing.Street
31.15 invoicing.Streetno
31.16 invoicing.Zipcode
31.17 invoicing.Locality
31.18 invoicing.box
31.19 invoicing.AbroadFlag
31.19.1 invoicing.CountryCode
31.20 invoicing.abroadFreeText
32.1 patient.patientDob
33 patient.patientSexCode

After an execution of AnonymizePatientData, no further RequestPartB or getSheet for that sheet number will be accepted.

#### *5.4.4.1 Request*

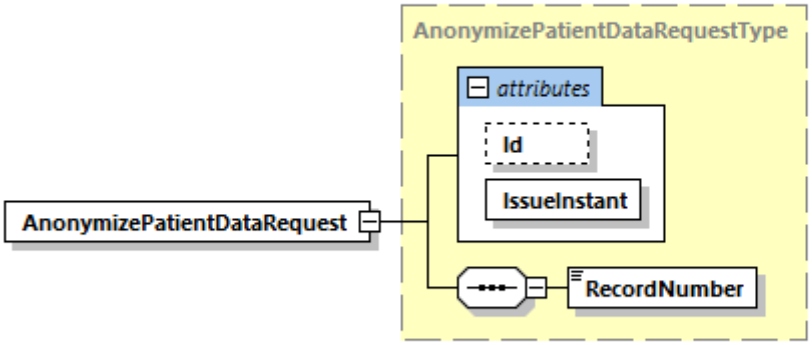



**Figure 8: AnonymizePatientDataRequest**

| 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 |

| | | | |
|---|---|---|---|
| RecordNumber | Sheet number which consists of max 20 characters. | No | Yes |

### 5.4.4.2 Response

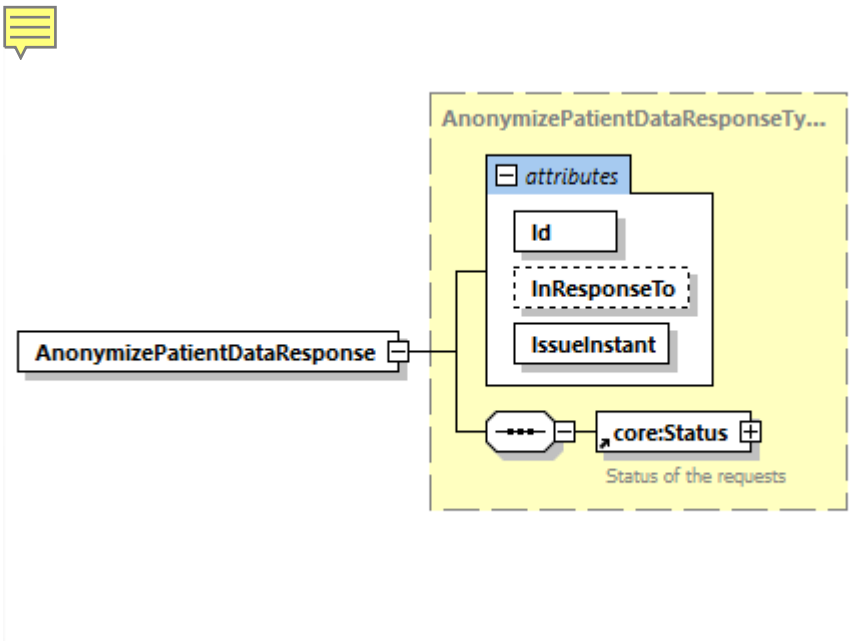



**Figure 9: AnonymizePatientDataResponse**

| 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 |
| Status | See section 5.4.2.1 StatusType | No | Yes |

### 5.4.4.3 Example

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xm
lns:urn="urn:be:fgov:ehealth:emsr:registration:v2">
   <soapenv:Header/>
   <soapenv:Body>
      <urn:AnonymizePatientDataRequest Id="id-
FBFE19453D50A4BFE61491476862891174" IssueInstant="2020-07-
31T10:58:48.288+02:00">
         <urn:RecordNumber>20121912360638224101</urn:RecordNumber>
      </urn:AnonymizePatientDataRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xm
lns:urn="urn:be:fgov:ehealth:emsr:registration:v2">
   <soapenv:Header/>
   <soapenv:Body>
      <urn:AnonymizePatientDataRequest Id="id-
FBFE19453D50A4BFE61491476862891174" IssueInstant="2020-07-
31T10:58:48.288+02:00">
         <urn:RecordNumber>20121912360638224101</urn:RecordNumber>
      </urn:AnonymizePatientDataRequest>
   </soapenv:Body>
```

# 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.

In case eHealth finds a bug or vulnerability in its software, the partner is advised to update his application with the newest version of the software within 10 business days.

In case the partner finds a bug or vulnerability in the software or WS that eHealth delivered, he is obliged to contact and inform eHealth immediately and he is not allowed to publish this bug or vulnerability in any case.

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

### 6.1.3 The use of username, password and token

The username, password, and token are strictly personal and are not allowed to transfer.
Every user takes care of his username, password, and token and is forced to confidentiality of it. Every user is also responsible of every use, which includes the use by a third party, until the inactivation.

# 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 integration info to integrate is published 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 to access the eHealth acceptance environment.

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

After successful acceptance tests, the partner sends his test results and performance results with a sample 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. During the release day, the partner provides eHealth 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 service for one of its applications will always test first in the acceptance environment before releasing any adaptations of its application in production. In addition, he will inform eHealth on the progress and test period.

## 7.2 Test cases

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

- Publish multiple different part A and part B, and check correct handling of error message received.
- Delete a sheet.
- Anonymize a sheet.

In addition, the organization should also run negative test cases:

- Publish a part A or part B already present.
- Publish part B when part A is not already there.
- Delete a sheet that does not exist or has already been deleted.
- Anonymize a sheet that already has been anonymized.

# 8. Error and failure messages

There are three different possible types of response:

- If there are no technical or business errors, business response is returned.
- If a business error occurred, a business error is returned (see chapter 8.1 Business errors).
- In the case of a technical error, a SOAP fault exception is returned (see chapter 8.1.2).

## 8.1 Business errors

### 8.1.1 Status Code

See 5.4.2.1 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.

| StatusCode | Message | Solution |
|---|---|---|
| urn:be:fgov:ehealth:2.0:status:Success (level 1)<br>urn:be:fgov:ehealth:2.0:status:DataNotFound (level 2) | No results for the request | Change one of the search criteria.<br>Diminish the number of search criteria. |
| urn:be:fgov:ehealth:2.0:status:Requester (level 1)<br>urn:be:fgov:ehealth:2.0:status:InvalidInput (level 2) | KMEHR message validation error | Correct data in sheet. |
| urn:be:fgov:ehealth:2.0:status:Requester (level 1)<br>urn:be:fgov:ehealth:2.0:status:RequestDenied (level 2) | KMEHR part A does not exist | First send part A before part B |
| urn:be:fgov:ehealth:2.0:status:Requester (level 1)<br>urn:be:fgov:ehealth:2.0:status:RequestDenied (level 2) | KMEHR part A already exist | Part A cannot be sent again |
| urn:be:fgov:ehealth:2.0:status:Requester (level 1)<br>urn:be:fgov:ehealth:2.0:status:RequestDenied (level 2) | KMEHR part B already exist | Part B cannot be sent again |
| urn:be:fgov:ehealth:2.0:status:Requester (level 1)<br>urn:be:fgov:ehealth:2.0:status:RequestUnsupported (level 2) | KMEHR message decryption error | The transmitted data cannot be decrypted |
| urn:be:fgov:ehealth:2.0:status:Requester (level 1)<br>urn:be:fgov:ehealth:2.0:status:RequestUnsupported (level 2) | KMEHR message XSD validation error | The transmitted data cannot be desterilised |
| urn:be:fgov:ehealth:2.0:status:Requester (level 1)<br>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)<br>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)<br>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)<br>urn:be:fgov:ehealth:2.0:status:InvalidInput (level 2) | The content of the Base64EncryptedData can't 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). |
| urn:be:fgov:ehealth:2.0:status:Requester (level 1)<br>urn:be:fgov:ehealth:2.0:status:RequestDenied (level 2) | The sheet is already anonymized | The sheet has already been anonymized and cannot be anonymized a second time. |
| urn:be:fgov:ehealth:2.0:status:Requester (level 1)<br>urn:be:fgov:ehealth:2.0:status:RequestDenied (level 2) | The sheet is not yet exported and can't be anonymized | To anonymize a sheet, it has to be already exported in the daily SPF Health export. Wait 24h before redoing the request. |
| urn:be:fgov:ehealth:2.0:status:Requester (level 1)<br>urn:be:fgov:ehealth:2.0:status:DataNotFound (level 2) | Sheet not found | The RecordNumber is not correct. Please check the RecordNumber. |
| urn:be:fgov:ehealth:2.0:status:Requester (level 1)<br>urn:be:fgov:ehealth:2.0:status:Forbidden (level 2) | The team is not a member of your service | Check the TeamIdentification in the request, it does not make part of the ambulance service. |

### 8.1.2 Status Detail

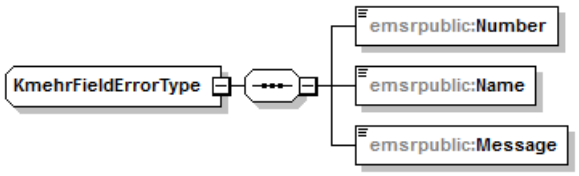



**Figure 10: StatusDetail structure**

If there was a validation error with the encrypted KMEHR content, a set of 'KmehrFieldError's will be returned. Each one corresponds to one rule violation with an explanation as described below.

| Field Name | Description | Attribute | Required |
|---|---|---|---|
| Number | Rules number as in the excel sheet 'ambureg_variables that is being violated | No | Yes |
| Name | Enunciation of the failing rule | No | Yes |
| Message | This describes why the rule fails. | No | Yes |

Example of rule validation errors:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
```

```
        <emsr:RegisterPartAResponse 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:Request
or">
                    <core:StatusCode Value="urn:be:fgov:ehealth:2.0:status:Inv
alidInput"/>
                </core:StatusCode>
                <core:StatusMessage>KMEHR message validation error</core:Statu
sMessage>
                <core:StatusDetail>
                    <emsrregister:KmehrFieldError>
                        <Number>4.1</Number>
                        <Name>missionIdentification</Name>
                        <Message>the value should match "^\d{11}$"</Message>
                    </emsrregister:KmehrFieldError>
                    <emsrregister:KmehrFieldError>
                        <Number>4.2</Number>
                        <Name>interHospitalTransportFlag</Name>
                        <Message>the value must be a boolean</Message>
                    </emsrregister:KmehrFieldError>
                </core:StatusDetail>
            </core:Status>
        </emsr:RegisterPartAResponse>
    </soap:Body>
</soap:Envelope>
```

Example KMEHR message not compliant to KMEHR XSD:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <emsr:RegisterPartAResponse 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:Request
or">
                    <core:StatusCode Value="urn:be:fgov:ehealth:2.0:status:Req
uestUnsupported"/>
                </core:StatusCode>
                <core:StatusMessage>KMEHR message XSD validation error</core:S
tatusMessage>
            </core:Status>
        </emsr:RegisterPartAResponse>
```

```
    </soap:Body>
</soap:Envelope>
```

Example part A missing when sending part B:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <emsr:RegisterPartAResponse 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:Request
or">
                    <core:StatusCode Value="urn:be:fgov:ehealth:2.0:status:Req
uestDenied"/>
                </core:StatusCode>
                <core:StatusMessage>KMEHR part A does not exist</core:StatusMe
ssage>
            </core:Status>
        </emsr:RegisterPartAResponse>
    </soap:Body>
</soap:Envelope>
```

Example when part A already exists:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <emsr:RegisterPartAResponse 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:Request
or">
                    <core:StatusCode Value="urn:be:fgov:ehealth:2.0:status:Req
uestDenied"/>
                </core:StatusCode>
                <core:StatusMessage>KMEHR part A already exists</core:StatusMe
ssage>
            </core:Status>
        </emsr:RegisterPartAResponse>
    </soap:Body>
</soap:Envelope>
```

## 8.2 Technical errors

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

The SOA Standard for Errorhandling 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:

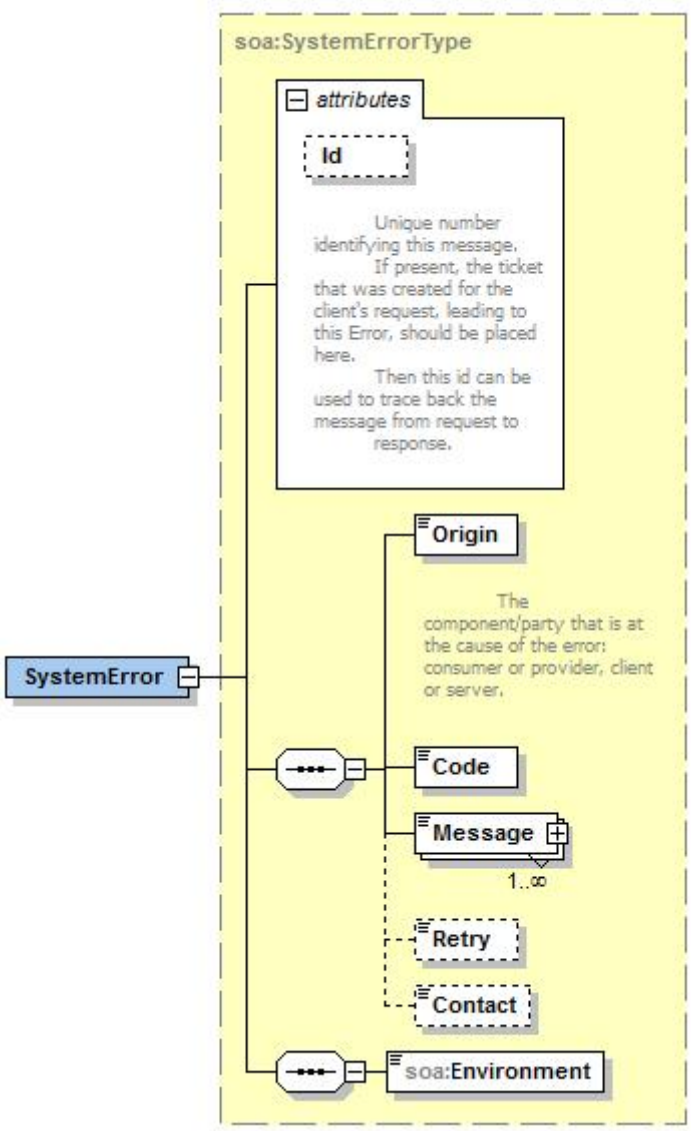



**Figure 11 SystemError**

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-
02001: Service is not available. Please contact service desk.</faultstring>
         <detail>
            <urn:SystemError Id="Id-
0ab63c6044370e219bb557dd" xmlns:urn="urn:be:fgov:ehealth:errors:soa:v1">
               <Origin>Server</Origin>
               <Code>SOA-02001</Code>
               <Message xml:lang="en">Service is not available. Please contact
 service desk.</Message>
               <urn:Environment>Acceptation</urn:Environment>
            </urn:SystemError>
         </detail>
      </soapenv: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/what 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 of this error. | No |

The default SOAP fault code 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 | Incorrectly formed message or containing 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 | Default error sent to the consumer if more details are missing.. |

| | | | |
|---|---|---|---|
| SOA-01001 | Consumer | Service call not authenticated | From the security information provided:<br>• or the consumer could not be identified;<br>• or the credentials provided are not correct. |
| SOA-01002 | Consumer | Service call not authorized | The consumer is identified and authenticated but is not allowed to call the given service. |
| SOA-02001 | Provider | Service not available. Please contact service desk | An unexpected error has occurred:<br>• Retries will not work.<br>• Service desk may help with root cause analysis. |
| SOA-02002 | Provider | Service temporarily not available. Please try later | An unexpected error has occurred:<br>• Retries should work.<br>• If the problem persists service desk may help. |
| SOA-03001 | Consumer | Malformed message | Default error for content related errors if more details are missing. |
| SOA-03002 | Consumer | Message must be SOAP | Message does not respect the SOAP standard. |
| SOA-03003 | Consumer | Message must contain SOAP 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 in Registry/Repository. |
| SOA-03006 | Consumer | XSD compliance failure | Message is not compliant with XSD in Registry/Repository. |
| SOA-03007 | Consumer | Message content validation failure | From the message content (conform XSD):<br>• Extended checks on the element format failed<br>• Cross-checks between fields failed |