<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Document management</td>
<td>4</td>
</tr>
<tr>
<td>1.1 Document history</td>
<td>4</td>
</tr>
<tr>
<td>2. Introduction</td>
<td>5</td>
</tr>
<tr>
<td>2.1 Goal of the service</td>
<td>5</td>
</tr>
<tr>
<td>2.2 Goal of the document</td>
<td>5</td>
</tr>
<tr>
<td>2.3 New in version 2.0</td>
<td>5</td>
</tr>
<tr>
<td>2.4 eHealth platform document references</td>
<td>6</td>
</tr>
<tr>
<td>2.5 External document references</td>
<td>6</td>
</tr>
<tr>
<td>2.6 Service History</td>
<td>6</td>
</tr>
<tr>
<td>3. Support</td>
<td>7</td>
</tr>
<tr>
<td>3.1 For issues in production</td>
<td>7</td>
</tr>
<tr>
<td>3.2 For issues in acceptance</td>
<td>7</td>
</tr>
<tr>
<td>3.3 For business issues</td>
<td>7</td>
</tr>
<tr>
<td>3.4 Certificates</td>
<td>7</td>
</tr>
<tr>
<td>4. Global overview</td>
<td>8</td>
</tr>
<tr>
<td>5. Step-by-step</td>
<td>9</td>
</tr>
<tr>
<td>5.1 Technical requirements</td>
<td>9</td>
</tr>
<tr>
<td>5.2 Use of the eHealth SSO solution</td>
<td>9</td>
</tr>
<tr>
<td>5.3 Encryption for known recipient</td>
<td>9</td>
</tr>
<tr>
<td>5.4 Process overview</td>
<td>9</td>
</tr>
<tr>
<td>5.5 eHealthBox Consultation Web Service</td>
<td>10</td>
</tr>
<tr>
<td>5.5.1 Lifetime of a message</td>
<td>10</td>
</tr>
<tr>
<td>5.5.2 getBoxInfo Method</td>
<td>11</td>
</tr>
<tr>
<td>5.5.3 GetMessagesList Method</td>
<td>14</td>
</tr>
<tr>
<td>5.5.4 GetMessageList Request</td>
<td>14</td>
</tr>
<tr>
<td>5.5.5 GetAllEhboxesMessagesList Method</td>
<td>19</td>
</tr>
<tr>
<td>5.5.6 getFullMessage Method</td>
<td>23</td>
</tr>
<tr>
<td>5.5.7 MoveMessage Method</td>
<td>26</td>
</tr>
<tr>
<td>5.5.8 DeleteMessage Method</td>
<td>29</td>
</tr>
<tr>
<td>5.5.9 getHistory Method</td>
<td>31</td>
</tr>
<tr>
<td>5.5.10 GetMessageAcknowledgmentsStatus Method</td>
<td>34</td>
</tr>
<tr>
<td>5.5.11 Used types</td>
<td>37</td>
</tr>
<tr>
<td>6. Risks and security</td>
<td>51</td>
</tr>
<tr>
<td>6.1 MTOM Policy</td>
<td>51</td>
</tr>
<tr>
<td>6.2 Security</td>
<td>51</td>
</tr>
<tr>
<td>6.2.1 Business security</td>
<td>51</td>
</tr>
<tr>
<td>6.2.3 Security policies to apply</td>
<td>51</td>
</tr>
<tr>
<td>7. Test and release</td>
<td>53</td>
</tr>
</tbody>
</table>
To the attention of: “IT expert” willing to integrate this web service.
# 1. Document management

## 1.1 Document history

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author</th>
<th>Description of changes / remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2</td>
<td>06/05/2013</td>
<td>eHealth platform</td>
<td>DeleteMessage, Annex v1.0 20171026.docx</td>
</tr>
<tr>
<td>2.3</td>
<td>21/09/2017</td>
<td>eHealth platform</td>
<td>Update list of qualities, Annex v1.0 20171026.docx</td>
</tr>
<tr>
<td>2.4</td>
<td>23/04/2018</td>
<td>eHealth platform</td>
<td>Externalize SSO and quality specification</td>
</tr>
</tbody>
</table>
2. Introduction

2.1 Goal of the service

The eHealthBox Consultation Web Service (WS) allows an authenticated user to consult information about the content associated with his eHealthBox.

A user can get general information on his eHealthBox, a list of messages for a specific folder and the content of a specific message. He can also move a message to his inbox.

Fields indicated as ‘obsolete’ are old fields still in use by some systems and kept for backward compatibility. They are out-of-date and must not be used by new partners for they provide no ‘extra’ feature.

The size of a message is currently limited to 10MB. The global size of an eHealthBox is also limited to 10mb (including inbox and trash bin folder). Note that an encrypted message weighs more due to the encryption overhead.

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 service called “eHealthBox” as provided by the eHealth platform.

We will explain the structure and content aspects of the possible requests, as well as the replies of the eHealth WS. An example illustrates each of those messages. In this document, you will also find a list of possible errors.

However, in order to interact in a smooth, homogeneous and risk controlled way with a maximum of partners, these partners must commit to comply with the requirements of specifications, data format and release processes of the eHealth platform as described in this document.

Technical and business requirements must be met in order to allow the integration and validation of the eHealth platform service in the client application.

2.3 New in version 2.0

- GetMessagesList performance has been improved and now returns a maximum of 100 messages by request. Start & End indexes are needed and allow specifying which messages are returned.
- GetMessagesList can return a list of messages from your inbox, sent box, bininbox, binsentbox.
- GetMessagesList and GetFullMessage now return more information under Sender (Id, Type, SubType) in order to reply to the sender.
- GetFullMessage now returns all the other recipients. This will enable you to make a “Reply to All”. Note that you are included with the recipients.
- Get the content of a message from your inbox or sent box (by specifying Source).
- Find out if a sent message has been successfully sent to a group of recipients, was received and read by those recipients with GetMessageAcknowledgmentsStatus.
- BoxId allow you to specify which of your eHealthBoxes you want to use.
- A single request GetAllEhboxesMessagesList enables you to receive all messages (100 by request) from all you accessible eHealthBoxes.
- CustomMeta are returned in responses from GetMessagesList, GetAllEhboxesMessagesList and GetFullMessage for early treatment. These are Meta data freely defined and added to the message by the sender.
- GetFullMessage no longer moves a message to the bin after it has been read. You will have to manually move it to your bin by using MoveMessage.
- MoveToInbox became MoveMessage (Source, Destination). Move a message from & to your Inbox, sent box or bin.
•  **DeleteMessage** enables the user to physically and definitely delete one or more messages from the inbox, sent box or bin. This can be used to clean up the eHealthBox when reaching the size limit.

### 2.4 eHealth platform document references

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

<table>
<thead>
<tr>
<th>ID</th>
<th>Title</th>
<th>Version</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Glossary.pdf</td>
<td>1.0</td>
<td>01/01/2010</td>
<td>eHealth platform</td>
</tr>
<tr>
<td>2</td>
<td>Cookbook STS</td>
<td>1.0</td>
<td>31/08/2010</td>
<td>eHealth platform</td>
</tr>
<tr>
<td>3</td>
<td>Cookbook ETEE: Bekende bestemmelings/Destinataire connu</td>
<td>2.3</td>
<td>06/05/2011</td>
<td>eHealth platform</td>
</tr>
<tr>
<td>4</td>
<td>Cookbook eHealthBox Publication</td>
<td>2.4</td>
<td>23/04/2018</td>
<td>eHealth platform</td>
</tr>
<tr>
<td>5</td>
<td>eHBox_Quality v1.01</td>
<td>1.01</td>
<td>23/04/2018</td>
<td>eHealth platform</td>
</tr>
<tr>
<td>6</td>
<td>eHBox_SSO v1.1</td>
<td>1.01</td>
<td>23/04/2018</td>
<td>eHealth platform</td>
</tr>
</tbody>
</table>

### 2.5 External document references

This list is a summary of all links used in the document referring to internet resources. The eHealth platform is not responsible for these links.

<table>
<thead>
<tr>
<th>Title</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.w3.org/standards/xml/">http://www.w3.org/standards/xml/</a></td>
<td>24/03/11</td>
</tr>
<tr>
<td><a href="http://www.oasis-open.org/specs/">http://www.oasis-open.org/specs/</a></td>
<td>24/03/11</td>
</tr>
<tr>
<td><a href="http://www.w3.org/TR/soap12-mtom/">http://www.w3.org/TR/soap12-mtom/</a></td>
<td>30/09/11</td>
</tr>
</tbody>
</table>

### 2.6 Service History

This chapter contains the list of changes made to the service with respect to the previous version.

<table>
<thead>
<tr>
<th>Previous version</th>
<th>Previous release date</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>03/02/2011</td>
<td>Major changes: Encryption, Multi-box, Publication WebApplication, Consultation WebApplication reviewed, general reliability ...</td>
</tr>
</tbody>
</table>

Remark: = “None” when the major version = 1

---

1 [https://ehealth.fgov.be/ehealthplatform](https://ehealth.fgov.be/ehealthplatform)
3. Support

3.1 For issues in production

eHealth platform contact center:

- Phone: 02/788 51 55
- Mail: support@ehealth.fgov.be
- Contact Form:
  - [https://www.ehealth.fgov.be/ehealthplatform/nl/contact](https://www.ehealth.fgov.be/ehealthplatform/nl/contact) (Dutch)
  - [https://www.ehealth.fgov.be/ehealthplatform/fr/contact](https://www.ehealth.fgov.be/ehealthplatform/fr/contact) (French)

3.2 For issues in acceptance

Integration-support@ehealth.fgov.be

3.3 For business issues

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

3.4 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:
  - Dutch version: [https://www.ehealth.fgov.be/ehealthplatform/nl/ehealth-certificaten](https://www.ehealth.fgov.be/ehealthplatform/nl/ehealth-certificaten)
- For technical issues regarding eHealth platform certificates
  - Acceptance: acceptance-certificates@ehealth.fgov.be
  - Production: support@ehealth.fgov.be
4. Global overview

This global overview aims to show the different steps needed to use the Consultation WS.

Step 1. The first required step for the user to use the Consultation WS is to contact the STS Service to acquire his proper token containing his data. (see 5.2 and Cookbook STS)

Step 2. Secondly, thanks to his token, the user can use the different “Consultation WS” to retrieve his message(s), manage his eHealthBox or get information on it.

Step 3. Depending on the request of the user, the “Consultation WS” will provide the user with an answer concerning his eHealthBox or one of his eHealthBoxes.

Step 4. Finally, if the message was encrypted for the recipient, the client deciphers his message with his private key and the Crypto Library (see the CookBook: “ETEE for unknown recipient” on the portal of the eHealth platform).

If the user wants to publish any message or reply, he should use the eHealthBox publication process.
5. Step-by-step

5.1 Technical requirements

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

5.2 Use of the eHealth SSO solution

This section specifies how the call to the Secure Token Service (STS) must be done in order to access the WS. You must precise several attributes in the request. The details on the identification attributes and the certification attributes can be found in the separate document eHealth eHBox_SSO.

To access the eHealth WS, the response token must contain “true” for the ‘boolean’ certification attribute.
If you obtain “false”, contact the eHealth contact center to verify that the requested test cases were correctly configured.

5.3 Encryption for known recipient

If an encrypted message was received, it has to be deciphered first. See Global overview.
In order to decipher the content of a message and the various fields, you have to use the local stored private key and the Crypto Libraries. Each field must be deciphered separately (one at a time).
All the information about the use of the encryption libraries is described in the ETEE cookbook available on on the website of the eHealth platform.

**Encrypted message convention:** If an encrypted message is received, ALL “Encryptable” fields contain (one and all) encrypted content. You can detect if the message has been encrypted thanks to the element IsEncrypted in the responses from GetMessagesList, GetAllEhboxesMessagesList and GetFullMessage

5.4 Process overview

Technical information is to be found on the Registry website of the eHealth platform:

https://services.ehealth.fgov.be/registry/uddi/bsc/web
The important sections of the WSDL (Web Service Definition Language) of the Consultation WS are:

- The applicable Policies, which cover the MTOM (file download) and security aspects.
- The different methods: getBoxInfo, getMessagesList, GetAllEhboxesMessagesList, getFullMessage, moveMessage, getMessageHistory, getMessageAcknowledgmentsStatus. (The ping method is only used for the monitoring of the web service)
- The types that are used by the methods.
- The fault messages are also defined for each method.

5.5 eHealthBox Consultation Web Service

5.5.1 Lifetime of a message

- When a message reaches the expiration date AND already has been put in the recycle bin after being read, it is definitely removed from the application.
When a message is older than 1 year (counted from publication date), it is definitely removed from the application, even if it has not been read.

5.5.2 getBoxInfo Method

The getBoxInfo method allows an authenticated user to receive general information about his mailbox, such as the number of messages, the current used size of his mailbox, the maximum allowed size of his mailbox, and the number of messages, which could not be received because the mailbox was full. These messages are still waiting to be placed in the mailbox. You need to clean your mailbox until the current size is lower than the max size. The messages will then enter into your mailbox.

5.5.2.1 getBoxInfo Request

You can optionally request information of another of your mailboxes by specifying it via BoxId.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoxId</td>
<td>If the client wants to use another of his eHealthBoxes, he can specify it here (see section 5.5.11.2). This avoids the client having to re-authenticate himself each time.</td>
</tr>
<tr>
<td>Mandator</td>
<td>Obsolete, do not use.</td>
</tr>
</tbody>
</table>

5.5.2.2 getBoxInfo Response

The response contains a success status code and general information on the eHealthBox as explained below. The BoxId element enables you to discover information on the current eHealthBox if you currently do not have any. Attention should be paid to NbrMessagesInStandBy larger than zero and if CurrentSize > MaxSize.
**Field name** | **Description**  
--- | ---  
Id | The ticket number (Id) attributed to the exchange request/response by the eHealth platform, is used to identify the session.
| Status | The Status block contains a code and a message. If no error has occurred during transaction, the Code will be ‘100’ and the Message ‘SUCCESS’. 

In case of a business error:
- The Code is an error code that unequivocally identifies the problem (see chapter 7 for the possible values).
- The Message will be a description of the error. Each Message has a Lang (language) characteristic:
  - "FR": French
  - "NL": Dutch
  - "EN": English
  - "DE": German
  - "NA": Not applicable

In case of technical errors, you will receive a Soap Fault message (see chapter 8). |
|---|---|
| BoxId | The identification number (Id) and the Type of the eHealthBox are provided. If applicable, the Subtype is also returned.
- The Quality of the owner from the eHealthBox (see eHBox_Quality v1.01 20180227.docx - List of qualities) |
| NbrMessagesInStandBy | The number of messages that are in standby because the eHealthBox is full. To consult these messages, the user has to delete others. |

### 5.5.2.3 Example

The following example does NOT contain the SAML assertion.

**Request**

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:urn="urn:be:fgov:ehealth:ehbox:consultation:protocol:v2">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:GetBoxInfoRequest/>
  </soapenv:Body>
</soapenv:Envelope>
```

**Response:**

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns3="urn:be:fgov:ehealth:ehbox:consultation:protocol:v2"
  xmlns:ns2="urn:be:fgov:ehealth:errors:service:v1"
  xmlns:ns4="urn:be:fgov:ehealth:ehbox:monitoring:protocol:v2">
  <Status>
    <Code>100</Code>
    <Message Lang="EN">SUCCESS</Message>
  </Status>
  <BoxId>
    <Id>9999999964</Id>
    <Type>INSS</Type>
  </BoxId>
```
5.5.3 GetMessagesList Method

The getMessagesList method provides a list of messages for a specific folder of your eHealthBox listed in order by date (most recent first, index “1”). A consequence could be that if a new message arrives between two consecutive queries, a message will be shown two times (message “1” becomes “2”, “2” becomes “3”, etc.).

E.g.: If you requested the messages between “1” and “100” and then the messages between “101” and “200”, then the message “100” would be the same as message “101”.

The content of the message is not yet returned by this method, but all the information needed is there in order to treat, filter, sort the messages. The sender, recipient, title message, publication date, message size, custom metas are all displayed for example.

5.5.4 GetMessageList Request

You can optionally request information of another of your mailboxes by specifying it via BoxId. This method can only return 100 messages at a time; consequently, you must use it multiple times if necessary.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoxId</td>
<td>If the client wants to use another of his eHealthBoxes, he can specify it here (see section 5.5.11.2). This avoids the client having to re-authenticate himself each time.</td>
</tr>
<tr>
<td>Source</td>
<td>You can specify the folder specific to your request via the Source parameter. The possible values are:</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>• “INBOX“ for the inbox folder.</td>
</tr>
<tr>
<td></td>
<td>• “SENTBOX“ for the sent box folder.</td>
</tr>
<tr>
<td></td>
<td>• “BININBOX“ for messages deleted from the inbox folder.</td>
</tr>
<tr>
<td></td>
<td>• “BINSENTBOX“ for messages deleted from the sent box folder.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>StartIndex</th>
<th>Index of the first message (minimum 1).</th>
</tr>
</thead>
<tbody>
<tr>
<td>EndIndex</td>
<td>Index of the last message (minimum 1). A maximum of 100 messages can be returned at once. EndIndex &lt; StartIndex + 100</td>
</tr>
<tr>
<td>Mandator</td>
<td>Obsolete, do not use.</td>
</tr>
</tbody>
</table>

### 5.5.4.1 getMessagesList Response

The response contains a success status code and as many Message elements as there are messages in the considered eHealthBox. Each element contains all necessary information to treat the message without downloading each individual message.
The ticket number (Id) attributed to the exchange request/response by the eHealth platform, is used to identify the eHealth session.
Status

The Status block contains a code and a message. If no error has occurred during the transaction, the Code will be ‘100’ and the Message ‘SUCCESS’.

In case of a business error:
- The Code is an error code that unequivocally identifies the problem (see chapter 7 for the possible values).
- The Message will be a description of the error. Each Message has a Lang (language) characteristic:
  - “FR” : French
  - “NL” : Dutch
  - “EN” : English
  - “DE” : German
  - “NA” : Not applicable

In case of technical errors, you will receive a Soap Fault message (see chapter 8).

Source

You can specify the folder specific to your request via the Source parameter. The possible values are:
- “INBOX” for the inbox folder.
- “SENTBOX” for the sent box folder.
- “BININBOX” for messages deleted from the inbox folder.
- “BINSENTBOX” for messages deleted from the sent box folder.

Message

0-to-more Message tag(s) describe(s) the eHealthBox Message(s). Each Message is defined by the following:
- The MessageId that represents a unique message identification generated by the system and returned during publication and when calling the getMessagesList. String of 13 digits.
- The Destination of the Message (see section 5.5.11.8).
- The Sender of the Message (see section 5.5.11.16).
- The Mandatee of the Message (see section 5.5.11.16). In case an authority was used during the publication of the message, the real sender will be found here. The name of the principal on whose behalf the message was sent, can then be found in Sender.
- The MessageInfo that contains additional information about the Message such as publication date, size ... (see section 5.5.11.13).
- The ContentInfo of the Message such as title, mime type ... (see section 5.5.11.5).
- The ContentSpecification of the Message that contains information such as importance ... (see section 5.5.11.6).
- The CustomMeta of the Message that contains free Meta data specified by the user (see section 5.5.11.7)

5.5.4.2 Example

The following example does not contain the SAML assertion.

Request:

```xml
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:urn="urn:be:fgov:ehealth:ehbox:consultation:protocol:v2">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:GetMessagesListRequest>
      <Source>INBOX</Source>
  ```
<StartIndex>1</StartIndex>
<EndIndex>100</EndIndex>
</urn:GetMessagesListRequest>
</soapenv:Body>
</soapenv:Envelope>

Response
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns4:GetMessagesListResponse
      xmlns:ns2="urn:be:fgov:ehealth:errors:service:v1"
      xmlns:ns3="urn:be:fgov:ehealth:ehbox:monitoring:protocol:v2"
      xmlns:ns4="urn:be:fgov:ehealth:ehbox:consultation:protocol:v2">
      <Status>
        <Code>100</Code>
        <Message Lang="EN">SUCCESS</Message>
      </Status>
      <Source>INBOX</Source>
      <Message>
        <MessageId>9Y0002LKM100K</MessageId>
        <Destination>
          <Id>77012824158</Id>
          <Type>INSS</Type>
          <Quality>DOCTOR</Quality>
        </Destination>
        <Sender>
          <Id>71000139</Id>
          <Type>NIHII</Type>
          <Quality>HOSPITAL</Quality>
          <Name>Doe</Name>
          <FirstName>John</FirstName>
        </Sender>
        <MessageInfo>
          <PublicationDate>2011-06-28+02:00</PublicationDate>
          <ExpirationDate>2011-12-31+01:00</ExpirationDate>
          <Size>46</Size>
        </MessageInfo>
        <ContentInfo>
          <EncryptableINSSPatient>OTgwNTMwNDU3NDYyMQ0K</EncryptableINSSPatient>
          <ContentType>NEWS</ContentType>
          <Title>News in eHealthBox</Title>
          <MimeType>text/plain</MimeType>
          <HasFreeInformations>true</HasFreeInformations>
          <HasAnnex>false</HasAnnex>
        </ContentInfo>
        <ContentSpecification>
          <IsImportant>false</IsImportant>
          <IsEncrypted>false</IsEncrypted>
        </ContentSpecification>
      </Message>
    </ns4:GetMessagesListResponse>
  </S:Body>
</S:Envelope>
5.5.5 GetAllEhboxesMessagesList Method

The GetAllEhboxesMessagesList method provides a list of all the messages from all the eHealthBoxes of a user (personal and enterprise eHealthBoxes) for a specific folder. This method repeatedly calls upon the GetMessagesList method for all known eHealthBoxes of the currently connected user.

5.5.5.1 GetAllEhboxesMessagesList Request

This method can only return 100 messages at a time; consequently, you must call upon it multiple times if necessary.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>You can specify the folder specific to your request via the Source parameter. The possible values are:</td>
</tr>
<tr>
<td></td>
<td>• “INBOX” for the inbox folder.</td>
</tr>
<tr>
<td></td>
<td>• “SENTBOX” for the sent box folder.</td>
</tr>
<tr>
<td></td>
<td>• “BININBOX” for messages deleted from the inbox folder.</td>
</tr>
<tr>
<td></td>
<td>• “BINSENTBOX” for messages deleted from the sent box folder.</td>
</tr>
<tr>
<td>StartIndex</td>
<td>Index of the first message (minimum 1).</td>
</tr>
<tr>
<td>EndIndex</td>
<td>Index of the last message (minimum 1). A maximum of 100 messages can be returned at a time. EndIndex &lt; StartIndex + 100</td>
</tr>
<tr>
<td>Mandator</td>
<td>Obsolete, do not use.</td>
</tr>
</tbody>
</table>

5.5.5.2 GetAllEhboxesMessagesList Response

The response contains a success status code and as many Message elements as there are messages in all the eHealthBoxes of the currently connected user (personal and enterprise eHealthBoxes). Each element contains all necessary information to treat the message without downloading each individual message. You can identify which eHealthBox received the message via the Destination element.
<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>The ticket number (Id) attributed to the exchange request/response by the eHealth platform is used to identify the eHealth session.</td>
</tr>
<tr>
<td>Status</td>
<td>The <em>Status</em> block contains a code and a message. If no error has occurred during the transaction, the <em>Code</em> will be '100' and the <em>Message</em> ‘SUCCESS’. Otherwise:</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>In case of a business error:</td>
</tr>
<tr>
<td></td>
<td>• The <em>Code</em> is an error code that unequivocally identifies the problem (see chapter 7 for the possible values).</td>
</tr>
<tr>
<td></td>
<td>• The Message will be a description of the error. Each Message has a <em>Lang</em> (language) characteristic:</td>
</tr>
<tr>
<td></td>
<td>• “FR”: French</td>
</tr>
<tr>
<td></td>
<td>• “NL”: Dutch</td>
</tr>
<tr>
<td></td>
<td>• “EN”: English</td>
</tr>
<tr>
<td></td>
<td>• “DE”: German</td>
</tr>
<tr>
<td></td>
<td>• “NA”: Not applicable</td>
</tr>
<tr>
<td></td>
<td>In case of technical errors, you will receive a Soap Fault message (see chapter 8).</td>
</tr>
<tr>
<td>Source</td>
<td>You can specify the folder, concerned by your request via the <em>Source</em> parameter.</td>
</tr>
<tr>
<td></td>
<td>The possible values are:</td>
</tr>
<tr>
<td></td>
<td>• “INBOX” for the inbox folder.</td>
</tr>
<tr>
<td></td>
<td>• “SENTBOX” for the sent box folder.</td>
</tr>
<tr>
<td></td>
<td>• “BININBOX” for messages deleted from the inbox folder.</td>
</tr>
<tr>
<td></td>
<td>• “BINSENTBOX” for messages deleted from the sent box folder.</td>
</tr>
<tr>
<td>Message</td>
<td>0-to-more Message tag(s) describe(s) the eHealthBox Message(s). Each Message is defined with the following:</td>
</tr>
<tr>
<td></td>
<td>• The MessageId that represents a unique message identification generated by the system and returned during the publication and when calling upon the getMessagesList.</td>
</tr>
<tr>
<td></td>
<td>• The Destination of the Message (see section 5.5.11.8).</td>
</tr>
<tr>
<td></td>
<td>• The <em>Sender</em> of the Message (see section 5.5.11.16).</td>
</tr>
<tr>
<td></td>
<td>• The <em>Mandatee</em> of the Message (see section 5.5.11.16). In case an authority was used during publication of the message, the real sender will be found here. The name of the <em>Principal</em> on whose behalf the message was sent will then be found in <em>Sender</em>.</td>
</tr>
<tr>
<td></td>
<td>• The <em>MessageInfo</em> that contains additional information about the Message such as publication date, size ... (see section 5.5.11.13).</td>
</tr>
<tr>
<td></td>
<td>• The <em>ContentType</em> of the Message such as title, mime type ... (see section 5.5.11.5).</td>
</tr>
<tr>
<td></td>
<td>• The <em>ContentTypeSpecification</em> of the Message that contains information such as importance ... (see section 5.5.11.6).</td>
</tr>
<tr>
<td></td>
<td>• The <em>CustomMeta</em> of the Message that contains free Meta data specified by the user (see section 5.5.11.7)</td>
</tr>
</tbody>
</table>

**5.5.5.3 Example**

The following example does not contain the SAML assertion.

**Request:**

```xml
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
  xmlns:urn="urn:be:fgov:ehealth:ehbox:consultation:protocol:v2">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:GetAllEhboxesMessagesListRequest>
```
5.5.6 getFullMessage Method

The getFullMessage method is used to get the corresponding complete message and its content to a provided MessageId.

5.5.6.1 getFullMessage Request

You can optionally request a message from your mailbox by specifying it via BoxId.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoxId</td>
<td>If the client wants to use another of his eHealthBoxes, he can specify it here (see section 5.5.11.2). This avoids the client having to re-authenticate himself each time.</td>
</tr>
</tbody>
</table>
| Source     | You can specify the folder specific to your request via the Source parameter. The possible values are:  
• “INBOX” for the inbox folder.  
• “SENTBOX” for the sent box folder.  |
| MessageId  | The MessageId is a unique message identification generated by the system and returned during the publication and when calling upon the getMessagesList. String of 13 digits. |
| Mandator   | Obsolete, do not use. |

5.5.6.2 getFullMessage Response

The response contains the same information as already returned by GetMessagesList plus the Message content in the element Message.
### Field Name | Description
--- | ---
**Id** | The ticket number (Id) attributed to the exchange request/response by the eHealth platform, is used to identify the eHealth session.

**Status** | The Status block contains a code and a message. If no error has occurred during the transaction, the Code will be ‘100’ and the Message ‘SUCCESS’.

- **In case of a business error:**
  - The Code is an error code that unequivocally identifies the problem (see chapter 7 for the possible values).
  - The Message will be a description of the error. Each Message has a Lang (language) characteristic:
    - “FR” : French
    - “NL” : Dutch
    - “EN” : English
    - “DE” : German
    - “NA” : Not applicable

- In case of technical errors, you will receive a Soap Fault message (see chapter 8).

**Sender** | The Sender of the Message (see section 5.5.11.16).

**Mandate** | In case an authority was used during the publication of the message, the real sender will be found under agent (see section 5.5.11.16). The name of the principal on whose behalf the message was sent will then be found in **Sender** (see section 5.5.11.16).
<table>
<thead>
<tr>
<th>Message</th>
<th>The Message itself (see section 5.5.11.12;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageInfo</td>
<td>Additional information about the Message (see section 5.5.11.13)</td>
</tr>
</tbody>
</table>

### 5.5.6.3 Example

The following example does NOT contain the SAML assertion.

**Request:**

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:be:fgov:ehealth:ehbox:consultation:protocol:v2">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:GetFullMessageRequest>
      <Source>INBOX</Source>
      <MessageId>9Y0002LKLP004</MessageId>
    </urn:GetFullMessageRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

**Response:**

```xml
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns4:GetFullMessageResponse xmlns:ns2="urn:be:fgov:ehealth:errors:service:v1"
xmlns:ns3="urn:be:fgov:ehealth:ehbox:monitoring:protocol:v2"
xmlns:ns4="urn:be:fgov:ehealth:ehbox:consultation:protocol:v2">
      <Status>
        <Code>100</Code>
        <Message Lang="EN">SUCCESS</Message>
      </Status>
      <Sender>
        <Id>71000139</Id>
        <Type>NIHII</Type>
        <Quality>HOSPITAL</Quality>
        <Name>Balduino</Name>
        <FirstName>Anil</FirstName>
      </Sender>
      <Mandatee>
        <Id>0422674827</Id>
        <Type>CBB</Type>
        <Quality>INSTITUTION</Quality>
        <Name>Njord</Name>
        <FirstName>Eduard</FirstName>
      </Mandatee>
      <Message MessageId="9Y0002LKLP004">
        <PublicationId>InitialDoc</PublicationId>
        <DestinationContext>
          <Id>99999999964</Id>
          <Type>INSS</Type>
          <Quality>DOCTOR</Quality>
        </DestinationContext>
      </Message>
    </ns4:GetFullMessageResponse>
  </S:Body>
</S:Envelope>
```
<ContentContext>
  <Content>
    <Document>
      <Title>Document in eHealthBox</Title>
      <EncryptableBinaryContent/>
      <DownloadFileName>test.txt</DownloadFileName>
      <MimeType>text/plain</MimeType>
    </Document>
    <FreeInformations>
      <EncryptableFreeText/>
    </FreeInformations>
    </Content>
    <ContentSpecification>
      <IsImportant>false</IsImportant>
      <IsEncrypted>false</IsEncrypted>
    </ContentSpecification>
    <CustomMeta>
      <Key>CategoryID</Key>
      <Value>2</Value>
    </CustomMeta>
    <CustomMeta>
      <Key>DocumentType</Key>
      <Value>Scan</Value>
    </CustomMeta>
  </ContentContext>
</Message>
<MessageInfo>
  <PublicationDate>2011-06-28+02:00</PublicationDate>
  <ExpirationDate>2011-12-31+01:00</ExpirationDate>
  <Size>12</Size>
</MessageInfo>
</ns4:GetFullMessageResponse>
</S:Body>
</S:Envelope>

5.5.7 MoveMessage Method

The MoveMessage method enables the user to move a message from a Source ("INBOX", "SENTBOX", "BININBOX", "BSENTBOX") to a Destination ("INBOX", "SENTBOX", "BININBOX", "BSENTBOX"). Only some combinations are allowed as explained below. You need to indicate if a message was received or sent by the concerned eHealthBox. You can do this by looking in which folder the message is situated or thanks to a comparison between the currently connected user and the Sender and/ or Destination element.

Allowed combinations

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>INBOX</td>
<td>BININBOX</td>
</tr>
<tr>
<td>SENT BOX</td>
<td>BSENTBOX</td>
</tr>
<tr>
<td>BININBOX</td>
<td>INBOX</td>
</tr>
<tr>
<td>BSENTBOX</td>
<td>SENTBOX</td>
</tr>
</tbody>
</table>
Limitation: Max 100 messages/request

### 5.5.7.1 MoveMessage Request

In Source specify where the message is currently situated and in Destination where the message must be moved to. In MessageId specify as many elements as there are messages to be moved. This method can only move 100 messages at a time; consequently, you must use it multiple times if necessary.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoxId</td>
<td>If the client wants to use another of his eHealthBoxes, he can specify it here (see section 5.5.11.2). This avoids the client having to re-authenticate himself each time.</td>
</tr>
</tbody>
</table>
| Source     | You can specify the folder specific to your request via the Source parameter. The possible values are:  
  - “INBOX” for the inbox folder.  
  - “SENTBOX” for the sent box folder.  
  - “BININBOX” for messages deleted from the inbox folder.  
  - “BINSENTBOX” for messages deleted from the sent box folder. |
| Destination| You can specify the folder specific to your request via the Destination parameter. The possible values are:  
  - “INBOX” for the inbox folder.  
  - “SENTBOX” for the sent box folder.  
  - “BININBOX” for messages deleted from the inbox folder.  
  - “BINSENTBOX” for messages deleted from the sent box folder. |
| MessageId  | The MessageId’s corresponding to the message(s) to move. |
| Mandator   | Obsolete, do not use. |
5.5.7.2 MoveMessage Response

The response contains a success status code or a Business Error as defined in chapter 8. The Business Error enables you to identify which messages where not successfully moved, even though all other have been moved successfully.

![Diagram of MoveMessageResponse]

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>The ticket number (Id) attributed to the exchange request/response by the eHealth platform, is used to identify the eHealth session.</td>
</tr>
<tr>
<td>Status</td>
<td>The Status block contains a code and a message. If no error has occurred during the transaction, the Code will be ‘100’ and the Message ‘SUCCESS’. In case of a business error:</td>
</tr>
<tr>
<td></td>
<td>• The Code is an error code that unequivocally identifies the problem (see chapter 7 for the possible values).</td>
</tr>
<tr>
<td></td>
<td>• The Message will be a description of the error. Each Message has a Lang (language) characteristic:</td>
</tr>
<tr>
<td></td>
<td>o “FR” : French</td>
</tr>
<tr>
<td></td>
<td>o “NL” : Dutch</td>
</tr>
<tr>
<td></td>
<td>o “EN” : English</td>
</tr>
<tr>
<td></td>
<td>o “DE” : German</td>
</tr>
<tr>
<td></td>
<td>o “NA” : Not applicable</td>
</tr>
<tr>
<td></td>
<td>In case of technical errors, you will receive a Soap Fault message (see chapter 8).</td>
</tr>
</tbody>
</table>

5.5.7.3 Example

The following example does NOT contain the SAML assertion.

Request

```xml
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
  xmlns:urn="urn:be:fgov:ehealth:ehbox:consultation:protocol:v2">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:MoveMessageRequest>
```
<Source>INBOX</Source>
<Destination>BININBOX</Destination>
<MessageId>9Y0002LM3006</MessageId>
</urn:MoveMessageRequest>
</soapenv:Body>
</soapenv:Envelope>

Response
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"

  <S:Body>
      <ns4:MoveMessageResponse ns2="urn:be:fgov:ehealth:errors:service:v1"
      xmlns:ns3="urn:be:fgov:ehealth:ehbox:monitoring:protocol:v2"
      xmlns:ns4="urn:be:fgov:ehealth:ehbox:consultation:protocol:v2">
          <Status>
              <Code>100</Code>
              <Message Lang="EN">SUCCESS</Message>
          </Status>
      </ns4:MoveMessageResponse>
  </S:Body>
</S:Envelope>

5.5.8 DeleteMessage Method

The DeleteMessage method enables the user to delete physically and definitely one or more messages from the inbox, sent box or bin. This can be used to clean up the eHealthBox when reaching the size limit. Be cautious when using this method. Best is to show a warning message to the user before deleting the messages.

Limitation: Max 100 messages/request

5.5.8.1 DeleteMessage Request

In MessageId you specify where the messages to delete from the bin or directly from the inbox or sent box. This method can only delete 100 messages at a time; consequently, you must use it multiple times if necessary.
<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BoxId</strong></td>
<td>If the client wants to use another of his eHealthBoxes, he can specify it here (see section 5.5.11.2). This avoids the client having to re-authenticate himself each time.</td>
</tr>
</tbody>
</table>
| **Sources** | You can specify the folder specific to your request via the Source parameter. The possible values are:  
- “INBOX” for the inbox folder.  
- “SENTBOX” for the sent box folder.  
- “BININBOX” for messages moved from the inbox folder.  
- “BINSENTBOX” for messages moved from the sent box folder. |
| **MessageId** | The MessageId's corresponding to the message(s) to delete. |

### 5.5.8.2 DeleteMessage Response

The response contains a success status code or a Business Error as defined in chapter 8. The Business Error enables you to identify which messages where not successfully deleted, even though all other have been deleted successfully.

![DeleteMessageResponse Diagram](https://via.placeholder.com/150)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Id</strong></td>
<td>The ticket number (Id) attributed to the exchange request/response by the eHealth platform, is used to identify the eHealth session.</td>
</tr>
</tbody>
</table>
| **Status** | The Status block contains a code and a message. If no error has occurred during the transaction, the Code will be ‘100’ and the Message ‘SUCCESS’. In case of a business error:  
- The Code is an error code that unequivocally identifies the problem (see chapter 7 for the possible values).  
- The Message will be a description of the error. Each Message has a Lang (language) characteristic:  
  o “FR”: French  
  o “NL”: Dutch  
  o “EN”: English  
  o “DE”: German  
  o “NA”: Not applicable  
In case of technical errors, you will receive a Soap Fault message (see chapter 8). |
| **MessageId** | List of MessageId that could not be deleted. However, the other messages were successfully deleted. |
5.5.8.3 Example

The following example does NOT contain the SAML assertion.

Request

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:urn="urn:be:fgov:ehealth:ehbox:consultation:protocol:v2">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:DeleteMessageRequest>
      <MessageId>INBOX</MessageId>
      <MessageId>9Y0002LKM3006</MessageId>
      <MessageId>9Y0002LKM3007</MessageId>
    </urn:DeleteMessageRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Response

```xml
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns4:DeleteMessageResponse
      xmlns:ns2="urn:be:fgov:ehealth:errors:service:v1"
      xmlns:ns4="urn:be:fgov:ehealth:ehbox:consultation:protocol:v2">
      <Status>
        <Code>815</Code>
        <Message Lang="EN">One or more messages couldn’t be deleted. All other messages were successfully deleted. Please verify for each message that the MessageId is correct, and it is in the Inbox or in the recycle bin.</Message>
      </Status>
      <MessageId>9Y0002LKM3006</MessageId>
    </ns4:DeleteMessageResponse>
  </S:Body>
</S:Envelope>
```

5.5.9 getHistory Method

When a new message is sent or updates an old News item (by using the same PublicationId), the old news item is archived and replaced by the newer news item. This method enables you to request a list of the old version of that news item by using the MessageId attributed to the newer news item.

The getHistory method is used to get the older message versions of a “news” type message. The method returns a list of MessageId’s corresponding to the previous version of a “news item”, which enables the user to enter a getFullMessage on those MessageId’s.

The getHistory method cannot retrieve a history of a document.

5.5.9.1 getHistory Request

You can request the list of MessageId of a news item from you Inbox or from your Sent Box. You can optionally request information of another of your mailboxes by specifying it via BoxId, In MessageId, you can place the MessageId of the newer News item, or the MessageId of an old version of the same news item.
### 5.5.9.2 getHistory Response

The response gives you a group of MessageId’s which concern the same news item. You can then enter a GetFullMessage in order to retrieve the old news item if necessary.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoxId</td>
<td>If the client wants to use another of his eHealthBoxes, he can specify it here (see section 5.5.11.2). This avoids the client having to re-authenticate himself each time.</td>
</tr>
</tbody>
</table>
| Source     | You can specify the folder specific to your request via the Source parameter. The possible values are:  
- “INBOX” for the inbox folder.  
- “SENTBOX” for the sent box folder. |
| MessageId  | The MessageId of the message to consult. The MessageId is a unique message identification generated by the system and returned during the publication and when calling upon getMessagesList. String of 13 digits. |
| Mandator   | Obsolete, do not use. |

User provides ID of his mandator and the quality of the mandator in use.
### 5.5.9.3 Example

The following example does NOT contain the SAML assertion.

**Request**

```xml
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
  xmlns:urn="urn:be:fgov:ehealth:ehbox:consultation:protocol:v2"

  <soapenv:Header/>
  <soapenv:Body>
    <urn:GetHistoryRequest>
```

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>The ticket number (Id) attributed to the exchange request/response by the eHealth platform, is used to identify the eHealth session.</td>
</tr>
</tbody>
</table>
| Status     | The Status block contains a code and a message. If no error has occurred during the transaction, the Code will be ‘100’ and the Message ‘SUCCESS’. In case of a business error:
  - The Code is an error code that unequivocally identifies the problem (see chapter 7 for the possible values).
  - The Message will be a description of the error. Each Message has a Lang (language) characteristic:
    - “FR”: French
    - “NL”: Dutch
    - “EN”: English
    - “DE”: German
    - “NA”: Not applicable
  In case of technical errors, you will receive a Soap Fault message (see chapter 8). |
| MessageId  | List of MessageId’s from the older message versions. |
5.5.10 GetMessageAcknowledgmentsStatus Method

The GetMessageAcknowledgmentsStatus method is used to find out for a message that the user has sent which recipients have received, viewed or read the message and at what time.

5.5.10.1 GetMessageAcknowledgmentsStatus Request

This method can only return 100 acknowledgements at a time; consequently, you must call upon it multiple times if necessary.
**5.5.10.2 GetMessageAcknowledgmentsStatus Response**

The response gives you information about your sent message: who received and read your message and at what time.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoxId</td>
<td>If the client wants to use another of his eHealthBoxes, he can specify it here (see section 5.5.11.2). This avoids the client having to re-authenticate himself each time.</td>
</tr>
<tr>
<td>MessageId</td>
<td>The <em>MessageId</em> is the message to consult. The <em>MessageId</em> is a unique message identification generated by the system and returned during the publication and when calling upon the getMessagesList. String of 13 digits.</td>
</tr>
<tr>
<td>StartIndex</td>
<td>Index of the first acknowledgment (minimum 1).</td>
</tr>
<tr>
<td>EndIndex</td>
<td>Index of the last acknowledgment (minimum 1). A maximum of 100 acknowledgments can be returned at once. EndIndex &lt; StartIndex + 100.</td>
</tr>
<tr>
<td>Mandator</td>
<td>Obsolete, do not use.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>The ticket number (<em>Id</em>) attributed to the exchange request/response by the eHealth platform, is used to identify the eHealth session.</td>
</tr>
</tbody>
</table>
The **Status** block contains a code and a message. If no error has occurred during the transaction, the **Code** will be ‘100’ and the **Message** ‘SUCCESS’. In case of a business error:
- The Code is an error code that unequivocally identifies the problem (see chapter 7 for the possible values).
- The Message will be a description of the error. Each Message has a Lang (language) characteristic:
  - “FR”: French
  - “NL”: Dutch
  - “EN”: English
  - “DE”: German
  - “NA”: Not applicable
In case of technical errors, you will receive a Soap Fault message (see chapter 8)

| AcknowledgmentsStatus | Contains a Row for each different Recipient of the message. Each Row contains the identification of the Recipient (Type SenderType, see section 5.5.11.16), the time the message was published, the time the message was received (viewed) by that Recipient, and the time the message was read by that Recipient. |

### 5.5.10.3 Example

The following example does not contain the SAML assertion.

**Request**

```xml
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
  xmlns:urn="urn:be:fgov:ehealth:ehbox:consultation:protocol:v2">
  <soapenv:Header />
  <soapenv:Body>
    <urn:GetMessageAcknowledgmentsStatusRequest>
      <MessageId>9Y0002LKH020J</MessageId>
      <StartIndex>1</StartIndex>
      <EndIndex>100</EndIndex>
    </urn:GetMessageAcknowledgmentsStatusRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

**Response**

```xml
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns4="urn:be:fgov:ehealth:ehbox:consultation:protocol:v2">
  <S:Body>
    <ns4:GetMessageAcknowledgmentsStatusResponse
      xmlns:ns2="urn:be:fgov:ehealth:errors:service:v1"
      xmlns:ns3="urn:be:fgov:ehealth:ehbox:monitoring:protocol:v2"
      xmlns:ns4="urn:be:fgov:ehealth:ehbox:consultation:protocol:v2">
      <Status>
        <Code>100</Code>
        <Message Lang="EN">SUCCESS</Message>
      </Status>
      <AcknowledgmentsStatus>
```

---

*Columns are not aligned correctly in the table.*

---
5.5.11 Used types

5.5.11.1 Annex
### Field name | Descriptions
--- | ---
EncryptableTitle | An *Annex* has an *EncryptableTitle*, a human readable description of its purpose (string minimum 1, maximum 400). If *IsEncrypted* is true (see Section 5.5.11.6), the content must be encrypted before being converted to xs:base64Binary (see section 5.3).
EncryptableBinaryContent | A base64-encoded binary content. If *IsEncrypted* is true (see Section 5.5.11.6), the content must be encrypted before being converted to xs:base64Binary (see section 5.3).
EncryptableTextContent | A base64-encoded text content. If *IsEncrypted* is true (see Section 5.5.11.6), the content must be encrypted before being converted to xs:base64Binary (see section 5.3).
DownloadFileName | E.g. “principal.pdf” (string minimum 1, maximum 80).
Mime Type | Represents the mime type of the content. E.g. “application/pdf”, “text/plain”, “application/octet-stream” (string minimum 1, maximum 50).
Signing | See section 5.5.11.17

### 5.5.11.2 BoxID
A *BoxID* contains all the information on the eHealthBox the client wants to use for the request. You will find all mandatory information about the allowed combinations Id-Type-SubType-Quality in the eHBox_Quality v1.01 20180227.docx – List of qualities.

![BoxID Diagram](image)

### Field name | Descriptions
--- | ---
ID | Your eHealthBox’s identification number. This is a digital number representing an INSS, NIHII, FAMPH, or CBE. String.
Type | Your eHealthBox’s ID type (“INSS”, “NIHII”, “FAMPH” or “CBE”). String.
Subtype (obsolete) | If the recipient is an organization, the *Subtype* allows (if necessary) further specification (such as “HOSPITAL” *SubType* for a Hospital *Quality*, or “GROUP” *SubType* for a Group *Quality*). String.
Quality | Your eHealthBox’s *Quality*. String (see eHBox_Quality v1.01 20180227.docx).
5.5.11.3 Content

A **Content** contains the message content (a document or a news item) and optionally zero-or-more free information, a Patient INSS and zero-or-more annexes.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document</td>
<td>See section 5.5.11.10</td>
</tr>
<tr>
<td>News</td>
<td>See section 5.5.11.15</td>
</tr>
<tr>
<td>FreeInformations</td>
<td>See section 5.5.11.11</td>
</tr>
<tr>
<td>EncryptableINSSPatient</td>
<td>This optional field allows specifying an INSS number of a patient concerned by the message content. xs:base64Binary. If <strong>IsEncrypted</strong> is true (see Section 5.5.11.6), the content must be encrypted before being converted to xs:base64Binary (see section 5.3).</td>
</tr>
<tr>
<td>Annex</td>
<td>See section 5.5.11.1</td>
</tr>
</tbody>
</table>

5.5.11.4 ContentContext

A **ContentContext** contains the message content and message details, as well as zero-or-more (50 maximum) free **CustomMetas**. These **CustomMetas** can freely be specified by the user for internal usage. You can define a Key and a value for each **CustomMeta** (see 5.5.11.7).

<table>
<thead>
<tr>
<th>Field name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content</td>
<td>See section 5.5.11.3.</td>
</tr>
</tbody>
</table>
5.5.11.5 ContentInfo

<table>
<thead>
<tr>
<th>Field name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>EncryptableINSSPatient</td>
<td>This optional field allows specifying an INSS number of a patient concerned by the message content. If IsEncrypted is true (see Section 5.5.11.6), the content must be encrypted before being converted to xs:base64Binary (see section 5.3).</td>
</tr>
<tr>
<td>ContentType</td>
<td>The Content Type of the message (“DOCUMENT”, “NEWS”).</td>
</tr>
<tr>
<td>Title</td>
<td>The Title of the message, a human readable description of its purpose (string minimum 1, maximum 400).</td>
</tr>
<tr>
<td>MimeType</td>
<td>Represents the mime type of the content. E.g. “application/pdf”, “text/plain”, “application/octet-stream” (string minimum 1, maximum 50).</td>
</tr>
<tr>
<td>HasFreeInformation</td>
<td>A flag (true or false) that indicates if the message has any Free Information.</td>
</tr>
<tr>
<td>HasAnnex</td>
<td>A flag (true or false) that indicates if the message has an Annex.</td>
</tr>
</tbody>
</table>
5.5.11.6 ContentSpecification

<table>
<thead>
<tr>
<th>Field name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>ApplicationName</td>
<td>The Application sending the message (optional, string minimum 1, maximum 25).</td>
</tr>
<tr>
<td>IsImportant</td>
<td>Boolean (true or false) that indicates if the message is to be considered as important.</td>
</tr>
<tr>
<td>IsEncrypted</td>
<td>Boolean (true or false) that indicates if the content has been encrypted.</td>
</tr>
</tbody>
</table>

5.5.11.7 CustomMeta

CustomMeta was introduced in order to enable the client to transport any Meta information relative to the message he wants. You can specify a maximum of 50 different pairs (key, value). The fields are limited each to 250 characters. Those CustomMetas will be transported from the sender to the recipient. You can for example add a CustomMeta for internal usage as “CategoryId, 17”, or “MessageContent, Blood analysis”.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>Alphanumeric string used as a key (string minimum 1, maximum 250).</td>
</tr>
<tr>
<td>Value</td>
<td>Alphanumeric string value corresponding to the Key (string minimum 1, maximum 250).</td>
</tr>
</tbody>
</table>

5.5.11.8 Destination

A Destination contains all the information relative to your eHealthBox that received the message. This is very useful when you call upon the GetAllEhboxesMessagesList to identify which message was received by which eHealthBox. After all, two different eHealthBoxes could receive the same message. The message would then appear two times in GetAllEhboxesMessagesList.
<table>
<thead>
<tr>
<th>Field name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>The recipient’s identification number. This is a digital number representing an INSS, NIHII, FAMPH, or CBE. String.</td>
</tr>
<tr>
<td>Type</td>
<td>The recipient’s ID type (“INSS”, “NIHII”, “FAMPH” or “CBE”). String.</td>
</tr>
<tr>
<td>Subtype (obsolete)</td>
<td>If the recipient is an organization, the Subtype allows (if necessary) further specification (such as &quot;HOSPITAL&quot; SubType for a Hospital Quality, or &quot;GROUP&quot; SubType for a Group Quality). String.</td>
</tr>
<tr>
<td>Quality</td>
<td>A Quality defines the recipient’s eHealthBox. String (see eHBox_Quality v1.01 20180227.docx).</td>
</tr>
<tr>
<td>User</td>
<td>An optional User (FirstName and LastName) can be added in the destination context. In case of a publication to an organization, this field is used to specify a member of this organization (e.g. a doctor working in a hospital), (string minimum 1, maximum 100).</td>
</tr>
</tbody>
</table>

### 5.5.11.9 DestinationContext

A DestinationContext contains all the information on the recipient.

You will find all mandatory information about the allowed combinations Id-Type-SubType-Quality in the eHBox_Quality v1.01 20180227.docx – List of qualities.
<table>
<thead>
<tr>
<th>Field name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>The recipient’s identification number. This is a digital number representing an INSS, NIHII, FAMPH, or CBE. String.</td>
</tr>
<tr>
<td>Type</td>
<td>The recipient’s ID type (“INSS”, “NIHII”, “FAMPH” or “CBE”). String.</td>
</tr>
<tr>
<td>Subtype (obsolete)</td>
<td>If the recipient is an organization, the Subtype allows (if necessary) further specification (such as &quot;HOSPITAL&quot; SubType for a Hospital Quality, or &quot;GROUP&quot; SubType for a Group Quality). String.</td>
</tr>
<tr>
<td>Quality</td>
<td>A Quality defines the recipient’s eHealthBox. String (see eHBox_Quality v1.01 20180227.docx)</td>
</tr>
<tr>
<td>User</td>
<td>An optional User (FirstName and LastName) can be added in the destination context. In case of a publication to an organization, this field is used to specify a member of this organization (e.g. a doctor working in a hospital), (string minimum 1, maximum 100). String.</td>
</tr>
<tr>
<td>Mandate (obsolete)</td>
<td>Optional authority information will be added if the recipient has been granted an authority. The constituent’s identification number (Id) and Type are requested. If the constituent is an organization, the Subtype allows (if necessary) further specification (such as &quot;HOSPITAL&quot; SubType for a Hospital Quality, or &quot;GROUP&quot; SubType for a Group Quality). The recipient’s name may be specified.</td>
</tr>
</tbody>
</table>
5.5.11.10 Document

Please note that a message will contain either a News item or a Document, not both.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>A Document has a Title, a human readable description of its intent (string minimum 1, maximum 400).</td>
</tr>
<tr>
<td>EncryptableBinaryContent</td>
<td>A base64-encoded binary content. If IsEncrypted is true (see Section 5.5.11.6), the content must be encrypted before being converted to xs:base64Binary (see section 5.3).</td>
</tr>
<tr>
<td>EncryptableTextContent</td>
<td>A base64-encoded text content. If IsEncrypted is true (see Section 5.5.11.6), the content must be encrypted before being converted to xs:base64Binary (see section 5.3).</td>
</tr>
<tr>
<td>DownloadFileName</td>
<td>E.g. “principal.pdf” (string minimum 1, maximum 80).</td>
</tr>
<tr>
<td>MimeType</td>
<td>Represents the mime type of the content. E.g. “application/pdf”, “text/plain”, “application/octet-stream” (string minimum 1, maximum 50).</td>
</tr>
<tr>
<td>Signing</td>
<td>See section 5.5.11.17</td>
</tr>
</tbody>
</table>

5.5.11.11 FreelInformations

The sender is free to add more information via the FreelInformations field. These FreelInformations will be transparently provided to the recipient(s).
### Field name | Descriptions
---|---
EncryptableFreeText | Contains any alphanumeric string. If `IsEncrypted` is true (see Section 5.5.11.6), the content must be encrypted before being converted to xs:base64Binary (see section 5.3).

Table | *Title* (the title of the table) and 1 or more Row(s) (each Row has a `EncryptableLeftCell` and a `EncryptableRightCell` as xs:base64Binary.). If `IsEncrypted` is true (see Section 5.5.11.6), the content must be encrypted before being converted to xs:base64Binary (see section 5.3).

EncryptableOldFreeInformation (obsolete) | A base64-encoded content and a Render attribute (as a string). If `IsEncrypted` is true (see Section 5.5.11.6), the content must be encrypted before being converted to xs:base64Binary (see section 5.3).
### 5.5.11.12 Message

**ehboxc:MessageType**

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageId</td>
<td>The <em>MessageId</em> is a unique message identification generated by the system and returned during the publication and when calling upon the getMessagesList. String of 13 digits.</td>
</tr>
<tr>
<td>PublicationId</td>
<td>The Id that the sender has used to publish the message. String, minimum 1, maximum 13.</td>
</tr>
</tbody>
</table>
| DestinationContext | The *DestinationContext* is a complex type that contains information about the recipients.  
|                 | See details in section 5.5.11.9.  
|                 | A *Message* can have numerous *DestinationContext* (numerous recipients).                                                                  |
| ContentContext | The *ContentContext* is a complex type that contains the message content. See details in section 5.5.11.4.                                    |
| Meta           | **Currently, no meta information is defined.**  
|                 | Additional system meta information: can be defined by the eHealth platform and used in convention with the client (for future needs). The type of meta information must be defined in the eHealthBox system before it can be used (see section 5.5.11.14).                                       |
| CopyMailTo     | One or more email address(es) that will receive a notification when the message has been published (optional, string minimum 1, maximum 80). If you would like to notify |
more than one recipient, you can add each e-mail address in a separate `CopyMailTo` element.
By default, a notification will be sent to the hospital’s security manager (registered in the user management of the social security) or in case of a publication to an individual person (doctor, citizen...): the person will receive a notification if he has updated his email address on the web application UPPAD. ([https://www.ehealth.fgov.be/fr/esante/professionnels-de-la-sante/uppad](https://www.ehealth.fgov.be/fr/esante/professionnels-de-la-sante/uppad))

### 5.5.11.13 MessageInfo

A `MessageInfo` contains short details about the message.

![MessageInfo Diagram](image)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>PublicationDate</td>
<td>The <em>Date</em> the message is published. (format: yyyy-mm-dd+hh:mm).</td>
</tr>
<tr>
<td>ExpirationDate</td>
<td>The <em>Expiration Date</em> of the message (format: yyyy-mm-dd+hh:mm). More information about the Expiration Date in Lifetime of a message.</td>
</tr>
<tr>
<td>Size</td>
<td>The <em>Size</em> of the message in bytes.</td>
</tr>
</tbody>
</table>

### 5.5.11.14 Meta

![Meta Diagram](image)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>The type of the meta information (string minimum 1, maximum 250).</td>
</tr>
<tr>
<td>Value</td>
<td>A list of <em>Values</em> for this <em>Type</em> (string minimum 1, maximum 250).</td>
</tr>
</tbody>
</table>

### 5.5.11.15 News

Please note that a message will contain a News item or a Document, not both.
<table>
<thead>
<tr>
<th>Field name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>A Document has a Title, a human readable description of its intent (string minimum 1, maximum 400).</td>
</tr>
<tr>
<td>EncryptableBinaryContent</td>
<td>A base64-encoded binary content. If IsEncrypted is true (see Section 5.5.11.6), the content must be encrypted before being converted to xs:base64Binary (see section 5.3).</td>
</tr>
<tr>
<td>EncryptableTextContent</td>
<td>A base64-encoded text content. If IsEncrypted is true (see Section 5.5.11.6), the content must be encrypted before being converted to xs:base64Binary (see section 5.3).</td>
</tr>
<tr>
<td>MimeType</td>
<td>Represents the mime type of the content. E.g. “application/pdf”, “text/plain”, “application/octet-stream” (string minimum 1, maximum 50).</td>
</tr>
</tbody>
</table>
### 5.5.11.16 Sender/Recipient

A *Sender / Recipient* element contains all the information relative to the *Sender* or *Recipient* of the message.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>The Sender / Recipient ID. This is a digital number representing an INSS, NIHII, FAMPH, or CBE. String.</td>
</tr>
<tr>
<td>Type</td>
<td>The Sender / Recipient ID type (&quot;INSS&quot;, &quot;NIHII&quot;, &quot;FAMPH&quot; or &quot;CBE&quot;). String.</td>
</tr>
<tr>
<td>Subtype (obsolete)</td>
<td>If the recipient is an organization, the <em>Subtype</em> allows (if necessary) further specification (such as &quot;HOSPITAL&quot; <em>SubType</em> for a Hospital <em>Quality</em>, or &quot;GROUP&quot; <em>SubType</em> for a Group <em>Quality</em>). String.</td>
</tr>
<tr>
<td>Quality</td>
<td>A <em>Quality</em> defines the Sender / Recipient / Mandatee eHealthBox. String (see eHBox_Quality v1.01 20180227.docx).</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the Sender/Recipient. String.</td>
</tr>
<tr>
<td>FirstName</td>
<td>FirstName of the Sender/Recipient (optional). String.</td>
</tr>
<tr>
<td>PersonInOrganisation</td>
<td>If the message was sent by an organization like a hospital, the name of this organization will be stored in &quot;Name&quot; above and the INSS of the person really sending the message will be stored in &quot;PersonInOrganisation&quot; for information. This is in String format.</td>
</tr>
</tbody>
</table>
### 5.5.11.17 Signing

To ensure data integrity, the sender can sign the content and provide the following security `Signing` information.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>SigningType</code></td>
<td>The type of signature used; can be any string you want. This is a required attribute as soon as you define the element <code>Signing</code>. E.g. “PKCS”, “sha256” (required, string minimum 1, maximum 50).</td>
</tr>
<tr>
<td><code>SigningDownloadFileName</code></td>
<td>The name of the signing file. E.g. “signature.sha” (string minimum 1, maximum 80).</td>
</tr>
<tr>
<td><code>BinarySigningContent</code></td>
<td>The signature of the BinaryContent. xs:base64Binary.</td>
</tr>
<tr>
<td><code>TextSigningContent</code></td>
<td>The signature of the TextContent. xs:base64Binary.</td>
</tr>
</tbody>
</table>
6. Risks and security

6.1 MTOM Policy

For binary content sending, the “Message Transmission Optimization Mechanism” (MTOM/XOP) should be used. 

See http://www.w3.org/TR/soap12-mtom/ for the technical specification

6.2 Security

6.2.1 Business security

In case the development adds an additional use case based on an existing integration, the eHealth platform (info@eHealth.fgov.be) 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 responsible for this service. 

In case 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. 

In case the partner finds a bug or vulnerability in the software or WS that the eHealth platform has delivered, he is obliged to contact and inform the eHealth platform immediately. He is prohibited to publish this bug or vulnerability.

6.2.2 Web Service Security Policy

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

- that the request is authenticated with the SAML security profile policy. 
  See also eIBox_SSO v1.01 20180228.docx for a more detailed description of the SSO Access in the case of the eHealth platform.
- SSL one way
- an X.509 certificate. This certificate will contain the identifiers of the caller: INSS or NIHII number or CBE enterprise number. More information on how to obtain a certificate:
  Dutch version: https://www.ehealth.fgov.be/ehealthplatform/nl/ehealth-certificaten
  French version: https://www.ehealth.fgov.be/ehealthplatform/fr/certificats-ehealth
- the time-to-live of the message: one minute
- the 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.

In order to use the web services, an authorization from the eHealth platform is required.

6.2.3 Security policies to apply

We expect that you use SSL one way for the transport layer. As a 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 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)
  - the binary security token: an eHealth certificate or a SAML token issued by STS

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 on the website of the eHealth platform. This STS cookbook can be found on the portal of the eHealth platform.
7. Test and release

7.1 Procedure

This chapter explains the procedures for testing and releasing an application in acceptance 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. On the portal of the eHealth platform, you will find most of the required info to integrate.

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

7.1.2.1 Create test cases

Rules to access the Publication services are the same in test and in production.

Access rules:

- to use the Consultation services, the user must be part of one of the following profiles: hospital, nurse, group, institution, doctor, and laboratory...
- authentication with a certificate

All test cases have to be configured by the eHealth development team.

7.1.2.2 Request a certificate

Prior to requesting the certificate, you need to have installed the latest version of Java 1.6 and the Belgium eID middleware. You also need a smart-card reader and a Belgian eID. You can request the test certificate through one of the following URLs:

- Dutch version: https://www.ehealth.fgov.be/ehealthplatform/nl/ehealth-certificaten
- French version: https://www.ehealth.fgov.be/ehealthplatform/fr/certificats-ehealth

The process is described in the “How to request an eHealth test certificate”.

Depending on the user, you will need NIHII, INSS or CBE identification numbers in order to request the certificate.

7.1.2.3 Obtain SAML token

The usage of the STS and the structure of the exchanged xml-messages are described in the eHealth STS cookbook.

In the case of eHealthBox Publication web service, see eHealthBox – SSO v1.1.
7.1.3 Release procedure

When the development tests are successful, you can request to access the acceptance environment of the eHealth platform. From the moment you start the integration and acceptance tests, the eHealth platform suggests testing during at least one month.

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

Then 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. During the release day, the partner provides the eHealth platform with feedback on the test and on the performance tests.

For further information and instructions, please contact: info@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 perform test first in the acceptance environment before releasing any adaptations of its application in production. In addition, he will inform the eHealth platform on the progress and test period.

7.2 Test cases

This section describes a systematic process to test the Consultation service. The eHealth platform recommends performing tests for all of the following cases:

1. Consult your eHealthbox information with the method “getBoxInfo()”.
2. Based on your tests cases defined previously for the Publication service, get the list of messages contained in your eHealthbox with the method “getMessagesList()”. Execute this request on your inbox, sent box and on your bin.
3. Pick one of the “MessageIds” returned from your messages list and use it to get the full message with the method “getFullMessage()”. Execute this request on a message from your inbox and sent box.
4. Use again your “MessageID” to move your received message from your inbox to your recycle bin with the method “moveMessage()”.
5. Always consult the previous history of your message through a “MessageID” with the method “getMessagseHistory()”.
6. Use a “MessageID” from your sent box to consult the state of that message with “getMessageAcknowledgmentsStatus()”.
7. Consult all your messages from all your eHealthboxes with “GetAllEhboxesMessagesList()”. Execute this request on your inbox, sent box and on your bin.
8. Error and failure messages

8.1 Send Message Response Status Codes

Error codes originating from the eHealth platform:

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

<table>
<thead>
<tr>
<th>Error code</th>
<th>Component</th>
<th>Description</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>SendMessage</td>
<td>SUCCESS</td>
<td></td>
</tr>
<tr>
<td>806</td>
<td>* / MessageId</td>
<td>The specified MessageId is invalid; please verify that the Source and the MessageId are correct and that you can access it.</td>
<td>Is the MessageId correct? Is the message present in “Source”?</td>
</tr>
<tr>
<td>807</td>
<td>* / Start &amp; EndIndex</td>
<td>EndIndex must be larger or equal to StartIndex; please correct StartIndex and EndIndex.</td>
<td>Is “EndIndex &gt; StartIndex”?</td>
</tr>
<tr>
<td>808</td>
<td>* / Start &amp; EndIndex</td>
<td>A maximum of 100 messages can be returned by request; please correct StartIndex and EndIndex.</td>
<td>Is “EndIndex - StartIndex + 1 &lt;= 100”?</td>
</tr>
<tr>
<td>809</td>
<td>GetMessage-AcknowledgmentsStatus/MessageId</td>
<td>The specified MessageId is invalid; please verify that the MessageId is correct and that you are the sender.</td>
<td>Are you the sender of the message? Is the MessageId correct?</td>
</tr>
<tr>
<td>810</td>
<td>* / BoxId</td>
<td>The specified BoxId is invalid; please verify the data and that you can access it.</td>
<td>Can you normally access that eHealthBox?</td>
</tr>
<tr>
<td>812</td>
<td>MoveMessage/Source &amp; Destination</td>
<td>You cannot move a message from your Inbox to your Sent box (even via recycle bin) and vice versa.</td>
<td>Is the message present in “Source”?</td>
</tr>
<tr>
<td>813</td>
<td>MoveMessage/Source &amp; Destination</td>
<td>“7/10 messages were moved successfully. The following messages could not be moved: M1...Mx...Mn. Please verify for each message Mx that the Source and the MessageId are correct. Also pay attention that a message in the recycle bin which was deleted from the Inbox cannot be restored back to the Sent box and vice versa.”</td>
<td>Some messages where not found in the folder specified in “Source”. Some messages cannot be moved to “Destination”.</td>
</tr>
</tbody>
</table>
8.2 Soap Fault Error Codes

They contain the following attributes.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>Unique number identifying this message. If present, the ticket that was created for the client's request, leading to this error. Placed here, this Id can be used to trace back the message from the request.</td>
</tr>
<tr>
<td>Origin</td>
<td>The component/party causing the error: consumer or provider, client or server.</td>
</tr>
<tr>
<td>Code</td>
<td>The Error Code</td>
</tr>
<tr>
<td>Message</td>
<td>A human readable message</td>
</tr>
<tr>
<td>Retry</td>
<td>An optional Boolean that indicates if it is worth resending the same Request.</td>
</tr>
<tr>
<td>Contact</td>
<td>An optional field specifying a contact description.</td>
</tr>
<tr>
<td>Environment</td>
<td>The eHealth environment in which the error occurs: integration, acceptance or production.</td>
</tr>
</tbody>
</table>
**Attributes**

- **Id**
  - Unique number identifying this message.
  - If present, the ticket that was created for the client's request, leading to this Error, should be placed here.
  - Then this id can be used to trace back the message from request to response.

- **Origin**
  - The component/party that is at the cause of the error: consumer or provider, client or server.

- **Code**

- **Message**
  - 1.x

- **Retry**

- **Contact**

- **core:LocalisedString**
  - Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility. See RFC 3066 at http://www.iana.org/assignments/urn-parameters/urn-parameters-xmi.png and the IANA registry at http://www.iana.org/assignments/urn-parameters/urn-parameters-xmi.png for further information.
  - The union allows for the 'in-declaration' of xmldat with the empty string.
8.2.1 Schema Validation Errors

When invoking the WS, you must provide a valid XML. Before executing any action, the eHealthBox system verifies if the XML is valid by running a validation check towards the SendMessageRequest XSD.

If the validation fails, a SOAP Fault is returned with the following code and message:

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOA-03006</td>
<td>XSD compliance failure</td>
</tr>
</tbody>
</table>

Example

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="id-6">
    <soapenv:Fault>
      <faultcode>soapenv:Client</faultcode>
      <faultstring>SOA-03006</faultstring>
      <detail>
        <soa:SystemError xmlns:soa="urn:be:fgov:ehealth:errors:soa:v1" id="5bbd8a2a-bb21-4cf8-99bc-8d52c18e2801">
          <Origin>Consumer</Origin>
          <Code>SOA-03006</Code>
          <Message xml:lang="en">XSD compliance failure</Message>
          <soa:Environment>Production</soa:Environment>
        </soa:SystemError>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

8.2.2 Technical Errors

Technical errors are errors inherent to the internal working of the eHealth WS. These errors can also occur if the token used to call the WS is not valid.

They contain the standard SOAP Fault attributes.

The table provides the different codes and messages returned in a SOAP fault message:

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOA-00001</td>
<td>An internal error has occurred. Please contact the Contact Center.</td>
</tr>
</tbody>
</table>

This list can evolve.

Example

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
    <env:Fault>
      <faultcode>soapenv:Server</faultcode>
      <faultstring>SOA-00001</faultstring>
      <detail>
        <soa:SystemError Id="ec582704-d623-4b05-ab7f-98d5c9706dd1" xmlns:soa="urn:be:fgov:ehealth:errors:soa:v1">
          <Origin>Server</Origin>
          <Code>SOA-00001</Code>
          <Message xml:lang="en">An internal error has occurred. Please contact service desk.</Message>
        </soa:SystemError>
      </detail>
    </env:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```
<soa:Environment>Production</soa:Environment>
</soa:SystemError>
</detail>
</env:Fault>
</env:Body>
</soapenv:Envelope>