**Cookbook  
eJustBox Publication Web Service  
Version 1.0**

This document is provided to you free of charge by  
**Smals**

**Fonsnylaan 20 – Av. Fonsny 20**

**1000 BRUSSELS**

All are free to circulate this document with reference to the URL source.

# Table of contents

Table of contents 2

1 Document management 4

1.1 Document history 4

2 Introduction 5

2.1 Goal of the service 5

2.2 Various highlights 5

2.3 Goal of the document 5

2.4 eJustice document references 5

2.5 Service history 6

3 Business and privacy requirements 7

4 Global overview 8

5 Step-by-step 9

5.1 Technical requirements 9

5.2 Process overview 9

5.2.1 Web Service WSDL (Web Service Definition Language) 9

5.3 eJustBox Publication Web Service 9

5.3.1 Lifetime of a message (To be further defined) 9

5.3.2 Out-of-Office system 9

5.3.3 Mass mailings (not available in first phase) 10

5.3.4 SendMessage Method 10

5.3.4.1 SendMessage Request 10

5.3.4.2 SendMessage Response 11

5.3.4.3 Example 13

5.3.5 Used Types 14

5.3.5.1 Annex 14

5.3.5.2 Actor 15

5.3.5.3 Content 15

5.3.5.4 ContentContext 16

5.3.5.5 ContentSpecification 17

5.3.5.6 CustomMeta 17

5.3.5.7 Sender 18

5.3.5.8 Substitute 18

5.3.5.9 DestinationContext 19

5.3.5.10 Document 20

5.3.5.11 Meta 20

5.3.5.12 Status 21

6 Risks and security 22

6.1 SOAP with Attachments 22

6.2 Security 22

6.2.1 Business security 22

7 Test and release procedure 23

7.1 Procedure 23

7.1.1 Initiation 23

7.1.2 Development and test procedure 23

7.1.3 Create test cases 23

7.1.4 Release procedure 23

7.1.5 Operational follow-up 23

7.2 Test cases 23

8 Error and failure messages 24

8.1 Send Message Response Status Codes 24

8.2 Soap Fault Error Codes 25

8.2.1 Schema Validation Errors 26

8.2.2 Technical Errors 27

9 Annex 2 – List of Qualities 29

10 List of sources 30

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

# Document management

## Document history

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description of changes / remarks** |
| 1.0 | 29/04/2016 | Malik Alimoekhamedov | Initial draft |

# Introduction

## Goal of the service

The eJustBox Publication Web Service allows an authenticated user to publish an eJustBox message (Document or News) for different addressees, with optional elements such as annexes and metadata.

The publication request is received by the eJustBox central systems and processed ***asynchronously***.

This means that a successful response does not guarantee that the message will be correctly published at the end of the process. A publication failure can occur later on due to the behavior of external systems. In this case a structured notification message (an eJustBox message) is sent back to the sender.

A successful response message only guarantees that the message will be processed.

The size of a message and of an eJustBox is currently limited to 10MB, **on inbox and trash bin folder**. Note that an encrypted message weighs more due to the encryption overhead.

## Various highlights

* An “Out-of-Office” system enables the sender to know if one of the recipients is absent and to send his message to a substitute or substitutes, so the sender’s message can be treated. To do this, they can automatically transfer their messages to another colleague responsible during their holidays thanks to the “Out-of-Office” system.
* If a recipient is absent, a special *SendMessageResponse* is sent back, which will contain the substitutes. This will be further detailed below.
* SOAP with Attachments (SwA) standard is being used.
* eJustBoxes are limited to 10MB on inbox and trash bin folder.

## Goal of the document

This document provides functional and technical information about calling the , as provided by the eJustice platform.

In this service specification document, we will explain the structure and content aspects of the possible requests, as well as the replies of the eJustice web service. An example illustrates each of those messages. A list of possible errors can also be found in this document.

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

Some technical and legal requirements must be met in order for the eJustice web services to be integrated in client applications; this document is meant to provide you with an overview of these requirements.

This document is neither a development nor a programming guide for internal applications; eJustice partners always have a total freedom within those fields. Nevertheless, in order to interact in a smooth, homogeneous and risk controlled way with a maximum of partners, eJustice partners must commit to comply with specifications, data format, and release processes described in 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.

## eJustice document references

All the document references can be found in the technical library on the eJustice portal. These versions or any following versions can be used for the eJustice service.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Title** | **Version** | **Date** | **Author** |
| 1 | Cookbook eJustBox Consultation | 0.4 | 12/04/2016 | Malik Alimoekhamedov |

## Service history

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

|  |  |  |
| --- | --- | --- |
| **Previous version** | **Previous release date** | **changes** |
| 1.0 | 12/04/2016 | Initial version. |

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

# Business and privacy requirements

For problems with applications working in the acceptance / integration environment, please contact:

Integration Support: [contactcenter@eranova.fgov.be](mailto:contactcenter@eranova.fgov.be)

Available from 9:00 am till 16:30 pm

For problems with applications in the production environment, please contact:

Production Support: [contactcenter@eranova.fgov.be](mailto:integration-support@ehealth.fgov)

Available from 7:00 am till 20:00 pm

# Global overview

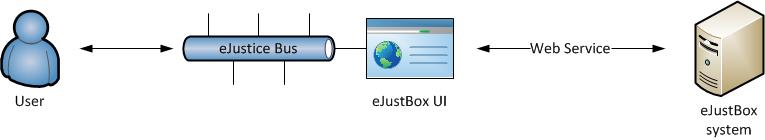


Figure 1: eJustBox Concept Diagram

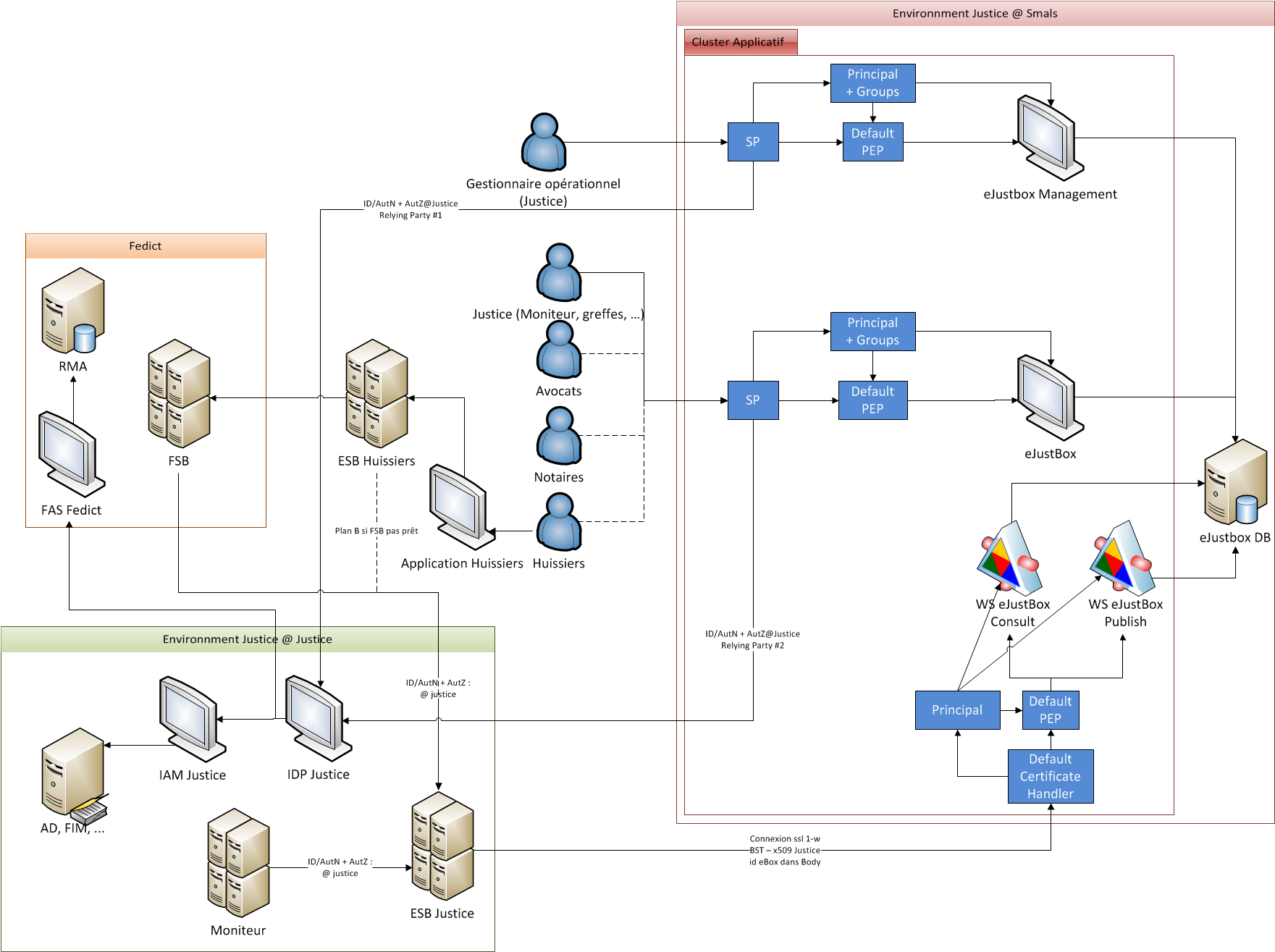


Figure 2: eJustBox interactions

There will be no STS authentication process implemented (at least in the initial phase) for users of eJustBox. eJustice Bus will act as a trusted filter allowing only authorized users to access eJustBox’s user interface. This user interface will be wired to consultation and publication (this document) Web services. No message encryption will be provided by the system in first phase.

# Step-by-step

## Technical requirements

All the xml requests that are submitted to the web service must be encoded in the UTF-8 format.

## Process overview

### Web Service WSDL (Web Service Definition Language)

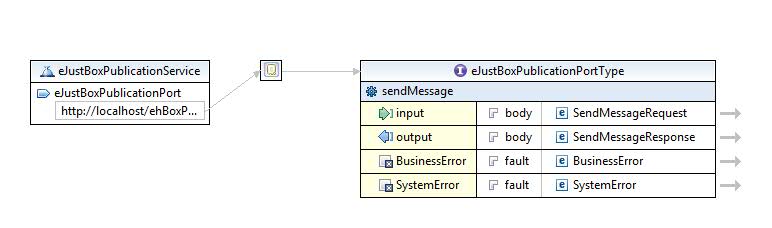


Figure 3: eJustBox Publication WSDL

The important sections of the WSDL (Web Service Definition Language) of the Publication Web Service are:

* The types (**SendMessageRequest** and **SendMessageResponse**) that are used by the **sendMessage** method. The fault message is also defined.
* The **sendMessage** method.

## eJustBox Publication Web Service

### Lifetime of a message (To be further defined)

* When the expiration date of a message is reached AND it has already been placed in the recycle bin AND it has been read, the message 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 hasn’t been read.

### Out-of-Office system

This system enables the sender to know if one of the recipients is absent and to send his message to a substitute or substitutes, so the sender’s message can be treated. Users can automatically transfer their messages to another colleague responsible during their holidays thanks to the “Out-of-Office” system. If a recipient is absent, a special *SendMessageResponse* is sent back, which will contain the substitutes

When sending a message, the system checks if some of the recipients of the message have an active Out-of-Office. If this is the case, a specific OoO Business error is returned (in SendMessageResponse). This response contains the period during which the original recipient is absent, the substitutes of the recipient (if specified) and their qualities.

It is possible that no replacement was introduced by the absent person. In this case, the same OoO Business error will be returned, but no substitute will be offered.

**SendMessageRequest must then be called again and substitutes should be added next to the original recipient. *OoOProcessed* under *DestinationContext* is set to true for the original recipient. For this recipient then, there will be no more OoO verification during the second request. If the message is to be sent encrypted, the message must be encrypted again for the added substitute(s).**

If the substitute itself is absent, a new OoO Business error will be returned, and so on.

The OoO activates automatically when the time period is reached.

OoO system is inactive for a mass mailing (described below). No notification will be thrown thus.

### Mass mailings (not available in first phase)

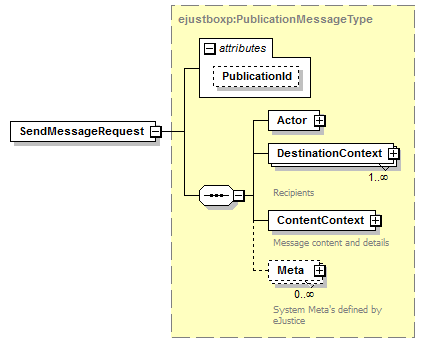
You can send a message to a particular person or to multiple persons by specifying one *DestinationContext* per person. You can also send a message to a list of persons sharing the same “Quality”. Please refer to **Error! Reference source not found.**. In order to be able to send a message to all persons sharing the same quality, prior authorization from eJustice (info@eJustice.fgov.be) is required. eJustice will investigate your request, and then allow you to publish this type of message.

### SendMessage Method

The request that must be sent in order to publish the eJustBox content for one or more recipients will be discussed here. You can send a message of the type “Document” or “News item”. A document contains a PublicationId that you define and that cannot be used twice. A News item, on the other hand, can be updated and this is done by sending a new news item with the same PublicationId. The only difference between a “Document” and “News item” message type is the fact that you can update it.

#### SendMessage Request

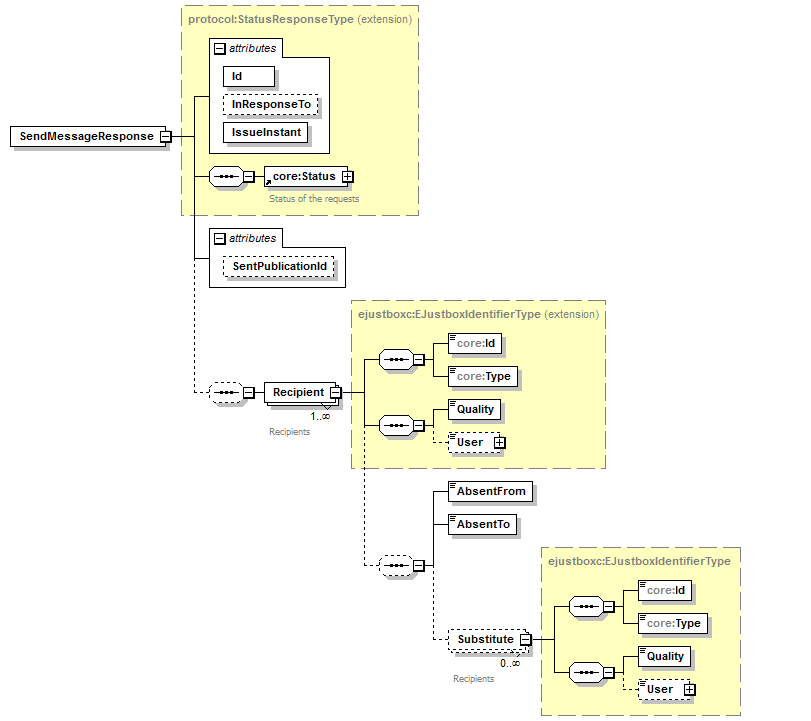
You will find all mandatory information about the allowed combinations Id-Type- Quality in Annex 2 – List of Qualities.



|  |  |
| --- | --- |
| **Field name** | **Description** |
| PublicationId | The identification of the publication as defined to the client.  Any alphanumeric string (minimum 1, maximum 13) can be used.  The use of the PublicationId depends on the content type: document or news item (see further).   * If the content is a “document”, the PubicationId must be unique. If you try to publish 2 documents with the same PublicationId, the second won’t be published. * If the content is a “news item”, you are allowed to use the same PublicationId to update your news content. When you publish 2 news with the same PublicationId, the first will be added as history and replaced by the second in the inbox folder. You have to be careful, if you first send a news item to x recipients, and then update your news item by sending a new content with the same PublicationId, all the recipients will receive the updated content, even if you specify other recipients. |
| Actor | See section 5.3.5.2 |
| Sender | See section 5.3.5.7 |
| DestinationContext | The *DestinationContext* is a complex type that contains information about the recipients.  This type is detailed in section 5.3.5.8  A *SendMessageRequest* can have numerous *DestinationContext*s(numerous recipients). |
| ContentContext | The *ContentContext* is a complex type that contains the message content.  This type is detailed in section 5.3.5.4. |
| Meta | **Currently, no meta information is defined.**  Additional system meta information can be defined by eJustice and used in convention with the client (for future needs). The type of meta information must be defined in the eJustBox system before it can be used (see section 5.3.5.11). |

#### SendMessage Response

We describe here how the response returns for the request as described above. The response contains a success status code if no major error could be found in the request. Still, r**emember that the system is asynchronous: a successful response does not guarantee that the message will be correctly published at the end of the process.** Possible Business Errors returned by the method are described in chapter 8.



|  |  |
| --- | --- |
| **Field name** | **Description** |
| Id | The *Id* is attributed to the eJustBox message by the eJustBox system. This is a unique number that is used to identify the message in the eJustBox system. This Id will be provided to the recipient by the consultation web service. |
| InResponseTo | Id attribute of the request |
| IssueInstant | Date and time of the response. |
| Status | Described in the Status section |
| SentPublicationId | The PublicationId provided in the SendMessageRequest message allows the grouping of the requests/responses. String 13 digits. |
| Recipient | This optional *Recipient* element is a complex type that contains information about the absent recipients (if any). This type extends *EJustboxIdentifierType*.  Detailed information about the content can be found in section 5.3.5.3 |
| AbsentFrom/To | This element contains the period of absence of the recipient. The start and end date are inclusive. The element type is xs:date. |
| Substitute | This optional *Substitute* element is a complex type that contains information about the colleagues replacing the absent person (if any defined by the absent person).  Detailed information about the content can be found in section 5.3.5.8 |

#### Example

Request:

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:be:fgov:eJustice:ejustbox:publication:protocol:v1">

<soapenv:Header/>

<soapenv:Body>

<urn:SendMessageRequest PublicationId="1009927801393">

<Actor>

<Id>982374923273</Id>

<Type>Type A</Type>

<Quality>Quality A</Quality>

<Name>Name</Name>

</Actor>

<DestinationContext>

<Id>10099278</Id>

<Type>Type B</Type>

<Quality>Quality B</Quality>

</DestinationContext>

<DestinationContext>

<Id>23022211</Id>

<Type>Type B</Type>

<Quality>Quality C</Quality>

</DestinationContext>

<ContentContext>

<Content>

<Document>

<Title>DocumentTitle</Title>

<EncryptableContent> [cid:4906633928</EncryptableContent](cid:4906633928%3c/EncryptableContent)>

</Document>

<EncryptableINSSPerson>ODQxMjA4MjI3NjI=</EncryptableINSSPerson>

<Annex>

<EncryptableTitle>YW5uZXggMQ==</EncryptableTitle>

<EncryptableTextContent>YW5uZXg=</EncryptableTextContent>

<DownloadFileName>annexname.txt</DownloadFileName>

<MimeType>text/plain</MimeType>

</Annex>

</Content>

<ContentSpecification>

<ContentType>DOCUMENT</ContentType>

<IsImportant>true</IsImportant>

<IsEncrypted>true</IsEncrypted>

<PublicationReceipt>true</PublicationReceipt>

<ReceivedReceipt>true</ReceivedReceipt>

<ReadReceipt>true</ReadReceipt>

</ContentSpecification>

</ContentContext>

</urn:SendMessageRequest>

</soapenv:Body>

</soapenv:Envelope>

Response:

<ehboxp:SendMessageResponse Id="5H0111043267W" SentPublicationId="1009927801393" xsi:schemaLocation="urn:be:fgov:eJustice:ejustbox:publication:protocol:v1 eJustice-eJustBox-publication-schema-protocol-1\_0.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ehboxp="urn:be:fgov:eJustice:ejustbox:publication:protocol:v1">

<Status>

<Code>826</Code>

<Message Lang="EN">One or more recipients have an Out-Of-Office active.</Message>

</Status>

<Recipient>

<Id>23022211</Id>

<Type>Type A</Type>

<Quality>Quality A</Quality>

<AbsentFrom>2012-01-01+01:00</AbsentFrom>

<AbsentTo>2012-01-04+01:00</AbsentTo>

<Substitute>

<Id>81021542380</Id>

<Type>Type B</Type>

<Quality>Quality B</Quality>

</Substitute>

<Substitute>

<Id>97949989164</Id>

<Type>Type B</Type>

<Quality>Quality C</Quality>

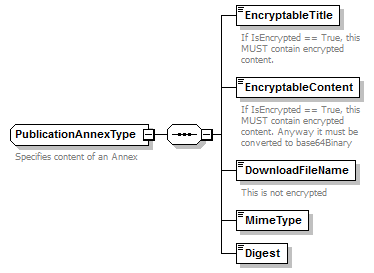
</Substitute>

</Recipient>

</ehboxp:SendMessageResponse>

### Used Types

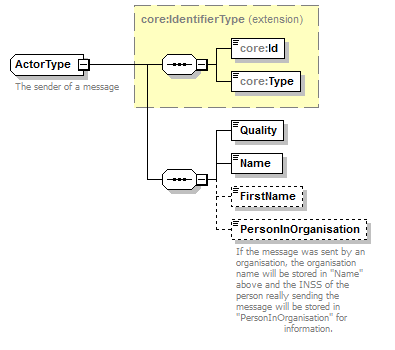
#### Annex



|  |  |
| --- | --- |
| **Field name** | **Descriptions** |
| EncryptableTitle | An *Annex* has an *EncryptableTitle*, a human readable description of its intent (minimum 1, maximum 400 characters in binary form).  If *IsEncrypted* is true (see Section 5.3.5.5), the title must be encrypted. |
| EncryptableContent | Contains a SOAP reference to the attachment Eg. <EncryptableContent> [cid:4906633928</EncryptableContent](cid:4906633928%3c/EncryptableBinaryContent)>. Type is xsd:anyURI.  If *IsEncrypted* is true (see Section 5.3.5.5), the attachment must be encrypted. |
| DownloadFileName | E.g. “principal.pdf” (string minimum 1, maximum 255). |
| MimeType | Represents the mime type of the content. E.g. “application/pdf”,” text/plain”, “application/octet-stream” (string minimum 1, maximum 255). |
| Digest | Mandatory element. The SHA-256 hash of the content after encryption (if encrypted) in binary form. Eg. <Digest>w1+mJdkfliLwJeo4ySUHDu/2nwuXJwBpj9atfTDZZig=</Digest> |

#### Actor

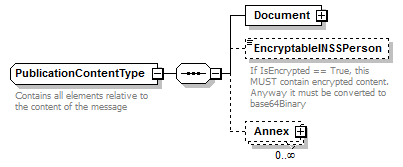
An *Actor* contains all the information on the eJustBox used for the request.



|  |  |
| --- | --- |
| **Field name** | **Descriptions** |
| ID | Your eJustBox’s identification number. This is a digital number representing an organization |
| Type | Your eJustBox’s ID type. String. |
| Quality | Your eJustBox’s *Quality*. String. |
| Name | Actor’s Last Name |
| FirstName | Actor’s First Name |
| PersonInOrganisation | If the message was sent by an organization, the organization name will be stored in “Name” above and the INSS of the person really sending the message will be stored in “PersonInOrganisation” for information purposes |

#### Content

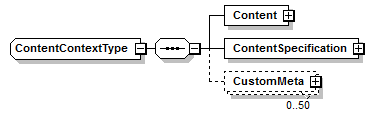
A *Content* contains the message content (a document or a news item) and optionally some free information, a Patient INSS and some annexes.



|  |  |
| --- | --- |
| **Field name** | **Descriptions** |
| Document | A news item is now a special case of a Document. See section 5.3.5.10 |
| EncryptableINSSPerson | This optional field allows specifying an INSS number of a patient concerned by the message content.  If *IsEncrypted* is true (see Section 5.3.5.5), the INSS number must be encrypted. |
| Annex | See section 5.3.5.1 |

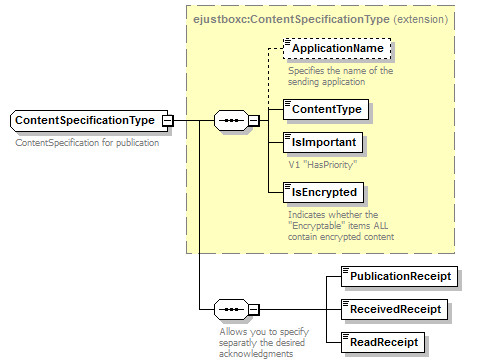
#### ContentContext

A *ContentContext* contains the message content and message details, as well as zero-or-more (50 maximum) free *CustomMeta*s. These *CustomMetas* can be freely specified by the user for internal usage. You can define a Key and a value for each *CustomMeta* (see 5.3.5.6).



|  |  |
| --- | --- |
| **Field name** | **Descriptions** |
| Content | See section 5.3.5.3 |
| ContentSpecification | See section 5.3.5.5 |
| CustomMeta | See section 5.3.5.6 |

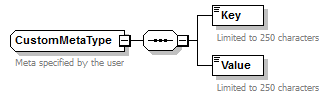
#### ContentSpecification



|  |  |
| --- | --- |
| **Field name** | **Descriptions** |
| ApplicationName | The Application sending the message (optional, string minimum 1, maximum 25). |
| ContentType | The *Content Type* of the message (“DOCUMENT”, “NEWS”). |
| IsImportant | Boolean (true or false) that indicates if the message is to be considered as important. |
| IsEncrypted | Boolean (true or false) that indicates if the content has been encrypted. |
| PublicationReceipt | Boolean (true or false) that indicates if a publication receipt is requested. A message is returned to the sender’s eJustBox when the message is persisted in database. Still, the recipient may not be able to view it if his eJustBox is full. |
| ReceivedReceipt | Boolean (true or false) that indicates if a received receipt is requested. A message is returned to the sender’s eJustBox when the message is viewed by the recipient (when he calls *GetMessagesList*). |
| ReadReceipt | Boolean (true or false) that indicates if a read receipt is requested. A message is returned to the sender’s eJustBox when the message is read by the recipient (when he calls *GetFullMessage*). |

#### CustomMeta

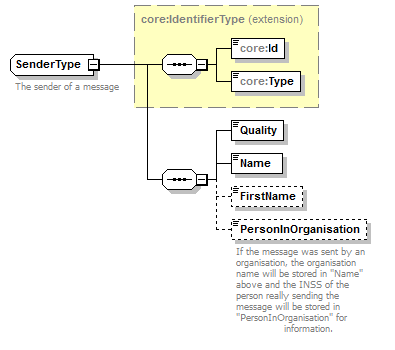
*CustomMeta* was invented in order to enable the client to transport any Meta information relative to the message he wants. You can specify a maximum of 50 different pairs (key, value). The fields are limited each to 250 characters. Those *CustomMetas* will be transported from the sender to the recipient.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



|  |  |
| --- | --- |
| **Field name** | **Descriptions** |
| Key | Alphanumeric string used as a key (string minimum 1, maximum 250). |
| Value | Alphanumeric string value corresponding to the *Key* (string minimum 1, maximum 250). |

#### Sender

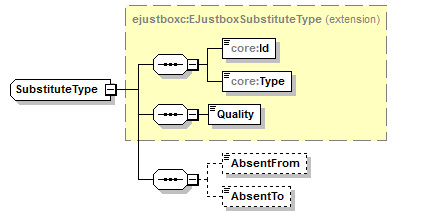
A *Sender* element contains all the information relative to the *Sender* of the message.



|  |  |
| --- | --- |
| **Field name** | **Descriptions** |
| ID | This is a digital number representing an organization. String. |
| Type | The Sender ID type. String. |
| Quality | A *Quality* defines the Sender eJustBox. String. |
| Name | Name of the Sender. String. |
| FirstName | FirstName of the Sender (optional). String. |
| PersonInOrganisation | If the message was sent by an organization, the organization name will be stored in "Name" above and the INSS of the person really sending the message will be stored in "PersonInOrganisation" for information. String. |

#### Substitute

A *Substitute* element contains all the information relative to the *Substitute* designated for a person.

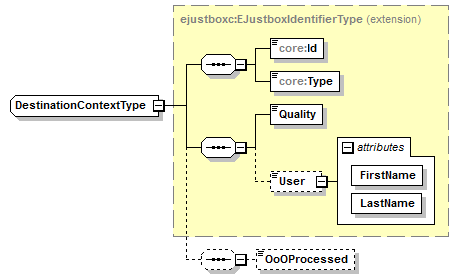


|  |  |
| --- | --- |
| **Field name** | **Descriptions** |
| ID | This is a digital number representing an organization. String. |
| Type | The Sender ID type. String. |
| Quality | A *Quality* defines the Sender eJustBox. String. |
| AbsentFrom | The *AbsentFrom* of the Out-of-Office. This date is inclusive. Ex. 2013-07-03+02:00 |
| AbsentTo | The *AbsentTo* of the Out-of-Office. This date is inclusive. Ex. 2013-07-17+02:00 |

#### DestinationContext

A *DestinationContext* contains all the information on the recipient.

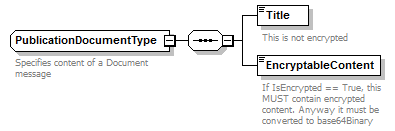
You will find all mandatory information about the allowed combinations Id-Type- Quality see Annex 2 – List of Qualities.



|  |  |
| --- | --- |
| **Field name** | **Descriptions** |
| ID | The recipient’s identification number or “ALL” (publication to a “*Quality*”). This is a digital number representing an organization. String. |
| Type | The recipient’s ID type. String. |
| Quality | A *Quality* defines the recipient’s eJustBox and is mandatory. String. |
| User | 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). |
| OoOProcessed | Optional Boolean, set to false by default. If during a previous call to SendMessageRequest an Out-of-Office notification was received and this notification has been processed, OoOProcessed should be set to true in order to avoid receiving the same notification again. Although, a new OoO notification could be received for one of the substitutes inserted in recipients. |

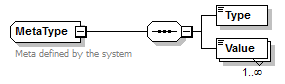
#### Document

Please note that a message will either contain a News item or a Document, not both. News items are “updatable documents”. You can indicate that the document can be updated by specifying it (“NEWS”) in the element *ContentType* under *ContentSpecification*, see section 5.3.5.5.



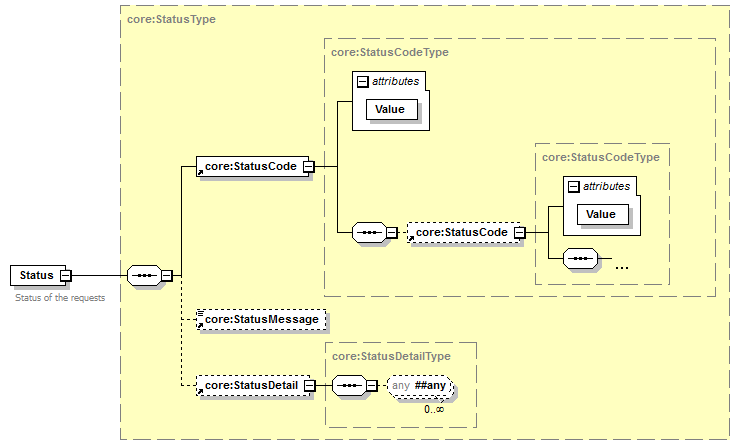
|  |  |
| --- | --- |
| **Field name** | **Descriptions** |
| Title | A Document has a *Title,* a human readable description of its intent (string minimum 1, maximum 400). |
| EncryptableContent | Contains a SOAP reference to the attachment Eg. <EncryptableContent> [cid:4906633928</EncryptableContent](cid:4906633928%3c/EncryptableContent)>. Type is xsd:anyURI.  If *IsEncrypted* is true (see Section 5.3.5.5), the attachment must be encrypted. |

#### Meta



|  |  |
| --- | --- |
| **Field name** | **Descriptions** |
| Type | The type of the meta information (string minimum 1, maximum 250). |
| Value | A list of *Value*s for this *Type* (string minimum 1, maximum 250). |

#### Status



| **Field name** | **Descriptions** |
| --- | --- |
| Status | Status of the request |
| StatusCode | If no error has occurred during the transaction, the *Code* will be ‘100’.  In case of a business error:   * The *Code* is an error code that unequivocally identifies the problem (see chapter 7 for the possible values). |
| StatusMessage | If no error has occurred during the transaction, the *Message* ‘SUCCESS’.  In case of a business error:   * 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). |
| StatusDetail | Detailes regarding the status |

# Risks and security

## SOAP with Attachments

The standard is used to transfer binary data via SOAP web services. SOAP with Attachments is an older alternative to MTOM. SwA is used in combination with WS-Security.

The SwA specification describes how to send binary data separated from the SOAP body using multipart MIME messages.

[*SOAP Messages with Attachments*](http://www.w3.org/TR/SOAP-attachments) is a link to the W3C specification of the standard:

<http://www.w3.org/TR/SOAP-attachments>

*Attachments Profile Version 1.0* is a link to the WS-I profile with clarifications and amendments to the SwA specification:

<http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html>

To keep a reference to the attachment from within the message payload, WS-I defined the simpleType swaRef:

<http://ws-i.org/profiles/basic/1.1/swaref.xsd>

The attachment data is no part of the message payload. So when you sign the SOAP body for secure transfer, the attachment data itself is not signed.

## Security

### Business security

In case the development adds an additional use case based on an existing integration, eJustice (info@eJustice.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 web service, the partner may obtain support from the contact center that is responsible for this service (see [eJustice contact](#_eHealth_contact)).

In case eJustice 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 web service that eJustice has delivered, he is obliged to contact and inform eJustice immediately and he is prohibited to publish this bug or vulnerability in any case.

# Test and release procedure

## Procedure

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

### Initiation

If you intend to use the eJustice service, please contact contactcenter@eranova.fgov.be. The Project department will provide you with the necessary information and mandatory documents.

### Development and test procedure

You have to develop a client (service consumer) in order to connect to our web service. Most of the required information to integrate is published in the technical library on the eJustice portal site.

In some cases eJustice provides you with a mock-up service or test cases in order for you to test your client (service consumer) before releasing it in the acceptance environment.

### Create test cases

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

Access rules:

* + To use the Publication services, the user must be part of one of the following profiles: **Quality A**

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

### Release procedure

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

From the moment you start the integration and acceptance tests, eJustice 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 “eJustice request” and “eJustice answer” to the eJustice point of contact by email.

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

For further information and instructions, please contact: contactcenter@eranova.fgov.be

### Operational follow-up

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

## Test cases

This section describes a step-by-step process to test the Publication service.

eJustice recommends performing tests for all of the following cases:

* Publish a document message to the sender eJustBox and get a successful response.
* Publish a news item message to the sender eJustBox and get a successful response.

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

* Publish a wrong xml input and get an error response message: *XSD compliance failure* (see section 8)*.*

# Error and failure messages

## Send Message Response Status Codes

Error codes originating from the eJustice platform:

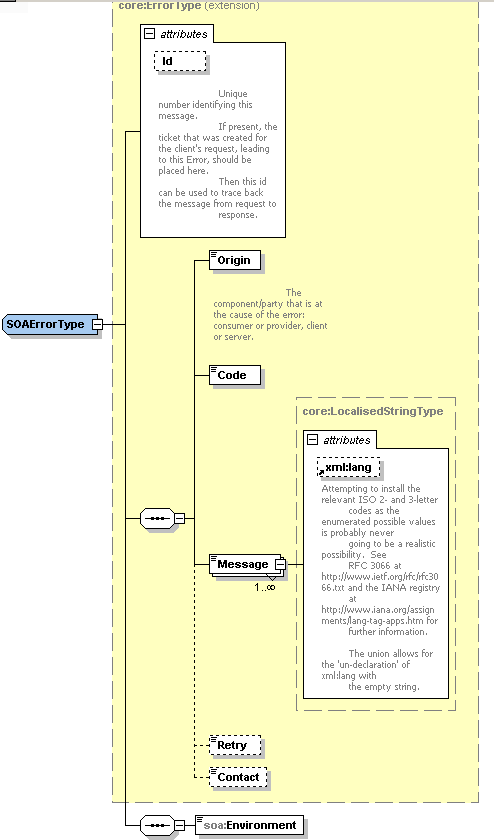
These error codes first indicate a problem in the sent arguments. This table can evolve.

|  |  |  |  |
| --- | --- | --- | --- |
| **Error code** | **Component** | **Description** | **Solution** |
| 100 | SendMessage | SUCCESS |  |
| 801 | SendMessage | The message exceeds the maximum authorized size. | Reduce appendixes size.  Split the message and send multiple messages. |
| 802 | SendMessage/ DestinationContext | The specified Identifier is invalid; please verify the ID. | Is the identifier valid and known by eJustice? |
| 803 | SendMessage/ DestinationContext | The specified Quality is invalid; please verify that Quality is a quality recognized by the system. | Verify allowed combination in Annex 2 – List of Qualities |
| 804 | SendMessage/ DestinationContext | The specified Type is invalid; please verify that Type is a type recognized by the system. | Verify allowed combination in Annex 2 – List of Qualities |
| 805 | SendMessage/ DestinationContext | The specified SubType is invalid; please verify that SubType is a subtype recognized by the system. | Verify allowed combination in Annex 2 – List of Qualities |
| 810 | SendMessage/ BoxId | The specified BoxId is invalid; please verify the data and that you can access it. | Can you normally access that eJustBox? |
| 814 | SendMessage/ DestinationContext | You are not authorized to publish to this Quality. | Contact eJustice to get authorization. |
| 816 | SendMessage / Document or Annex | The digest of the streamed document doesn't match with the supplied one. | Recalculate the SHA-256 hash on the content. |
| 826 | SendMessage | One or more recipients have an Out-Of-Office active. | Check the returned Substitutes for the absent recipient. |

## Soap Fault Error Codes

They contain the following attributes:

|  |  |
| --- | --- |
| **Field name** | **Descriptions** |
| 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 the request. |
| Origin | The component/party causing the error: consumer or provider, client or server. |
| Code | The Error Code |
| Message | A human readable message |
| Retry | An optional Boolean that indicates if it is worth resending the same Request. |
| Contact | An optional field specifying a contact description. |
| Environment | The eJustice environment in which the error occurs: integration, acceptation or production. |



### Schema Validation Errors

When invoking the Web Service, you must provide a valid XML.

Before executing any action, the eJustBox 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:

|  |  |
| --- | --- |
| **Code** | **Message** |
| SOA-03006 | XSD compliance failure |

Example:

<?xml version="1.0" encoding="UTF-8"?>

<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:eJustice: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>

### Technical Errors

Technical errors are errors inherent to the internal working of the eJustice Web Service. These errors can also occur if the token used to call the web service is not valid.

They contain the standard SOAP Fault attributes.

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

|  |  |
| --- | --- |
| **Code** | **Message** |
| SOA-00001 | An internal error has occurred. Please contact the Contact Centers. |

This list can evolve.

Example:

<?xml version="1.0" encoding="UTF-8"?>

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

<env: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:eJustice:errors:soa:v1">

<Origin>Server</Origin>

<Code>SOA-00001</Code>

<Message xml:lang="en">An internal error has occured. Please contact service desk.</Message>

<soa:Environment>Production</soa:Environment>

</soa:SystemError>

</detail>

</env:Fault>

</env:Body>

</soapenv:Envelope>

# Annex 2 – List of Qualities

Currently, the allowed combinations Type – Quality for the owner of an eJustBox are:

**Special Note:** It is not possible anymore to send a message to or as a CITIZEN.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Quality name | Quality eHbox | Type | Certificate attribute | Identification attribute | Certification attribute |
|  |  |  |  |  |  |

This list is prone to change.

|  |  |  |  |
| --- | --- | --- | --- |
| Quality name | Quality eHbox | Type | Certification attribute |
|  |  |  |  |

This list is prone to change.

# List of sources

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

|  |  |
| --- | --- |
|  | **Date** |
| http://www.w3.org/standards/xml/ | 24/03/11 |
| http://www.oasis-open.org/specs/ | 24/03/11 |
| <http://www.w3.org/TR/soap12-mtom/> | 30/09/11 |