

For assistance on filling the meta data fields of this document, just activate the Show/Hide (¶) function

OneGate(CSSR)

HTTPS Entrypoints

End User Manual

PSD:	CRS
Project code:	CRS
Department Group:	PRSM
Author:	PRSM
Date:	10/01/2011
Version :	2.3
Status:	Draft
Authorized by:	Ronny Martin
Reviewers:	PRSM, DQ
Commentators:	PRSM, DQDQ, External partners
Approvers:	Cellule Research and Development (DQ)
Location:	http://teamn.prd.nbb/sites/CRS/Shared Documents/DQ07CSSRINT/User's manuals - External/OneGate(CSSR) - End User Manual - HTTPS entrypoints.docx

Abstract: This document template is to be used for writing the End User Manual for a developed business service or product. It forms part of the System Development Life Cycle (SDLC). There are separate user manuals for operators/administrators and for developers of the service.

© National Bank of Belgium, Brussels

All rights reserved.
Reproduction for educational and non-commercial purposes is permitted provided that the source is acknowledged.

Table of Contents

1. Introduction	4
1.1 Document history	4
1.2 References	4
1.3 Overview of document	5
2. Product features and capabilities	6
2.1 Product purpose	6
2.2 Scope of intended use	7
2.3 Scenario	7
3. Generalities	9
3.1 Communication protocol	9
3.2 Authentication	9
3.3 Authorization	9
3.4 Secure data	10
3.5 Certificate	10
3.6 Binary data	10
3.7 Volume	10
3.8 Useful tools	11
4. HTTPS Entrypoints description	12
4.1 Overview	12
4.2 Upload file	12
4.3 Request list of available feedback	14
4.4 Request a feedback	17
4.5 GET an attachment of a feedback	20
4.6 Request list of available messages	20
4.7 Request a message	23
4.8 GET an attachment of a message	25
5. Error codes	26
5.1 Client error	26
5.2 Server error	26
6. Appendix - Sample using cURL	27
6.1 Prerequisites	27
6.2 Command cURL	28
7. Definition of terms and abbreviations	32

1. Introduction

This document contains everything you need to automate the data exchange with the application OneGate(CSSR).

This document describes the prerequisites of the usage of the OneGate(CSSR) HTTPS Entrypoints, the goal of each one of them, its input, output and possible errors.

Target audience for this document is the external partners who wish to automate the data exchange with the application OneGate(CSSR) using the HTTPS Entrypoints.

Note that all of those services are also available in a Web Service version using SOAP.

1.1 Document history

Date	Version	Author	Description of change
01/06/2010	- Draft	PRSM	Initial version
10/09/2010	2.0	PRSM	Entrypoints v2.0 (feedback with url)
24/09/2010	2.1	PRSM	Correction of some schemas and element definitions
22/10/2010	2.2	PRSM	Additional information about the http 302 redirection

1.2 References

Ref.	Title	Author	Location
[1]	OneGate(CSSR) - End user manual	DQ ¹	French Dutch
[2]	OneGate(CSSR) - End User manual - Web Services	PRSM ²	English
[3]	OneGate(CSSR) - XML Protocol	PRSM	[Not available]
[4]	File Exchange Mechanism - S/MIMEv2 Specifications	SYAS ³	[Not available]
[5]	NBB Certificate policy	DSM ⁴	French Dutch
[6]	NBB Certificate Practice Statement for External Counterparties (CPS)	DSM	French Dutch
[7]	Certificate management	DSM	English
[8]	Enrollment procedure v3.2	DSM	[Not available]

¹ DQ: Department General Statistics

² PRSM: IT Applications

³ SYAS: Network & Application Security

⁴ DSM: Data Security Management

1.3 Overview of document

This manual is structured as follows:

Section	Title	Main Purpose
1	Introduction	Specifies the document version, lists other, related documents and summarises the contents of this manual
2	Product description and environment:	Explains why the product exists, its scope and the scenario analysed
3	Generalities	Describes the generic points about the usage of the Web services.
4	HTTPS endpoints description	Explains how to use the product
5	Error codes	Lists and explains product error codes and associated corrective actions
6	Appendix - Sample using cURL	Some examples how to use the product using cURL
7	Definition of terms and abbreviations	Lists and explains any special terms used in the document.

2. Product features and capabilities

2.1 Product purpose

Today, the need for information grows continuously. Administrative and statistical institutes interview the citizens and companies regularly to collect this information. With the dematerialization, the paper form has given way to electronic form sent via Internet. The goal of the application One Gate (CSSR) is to become a unique point of data collection and data exchange for the National Bank of Belgium.

The data collection can be done either manually by filling an online form either (semi) automatically by sending an XML file via a manual file upload, a secured email or a web service call. The current document is limited to the description of the fully automated data exchange using the HTTPS endpoints (drawn in Figure 1 under point 3). You can also fully automated the data exchange using the Web Services instead of the HTTPS endpoints. The Webservice entry points are described in document [2].

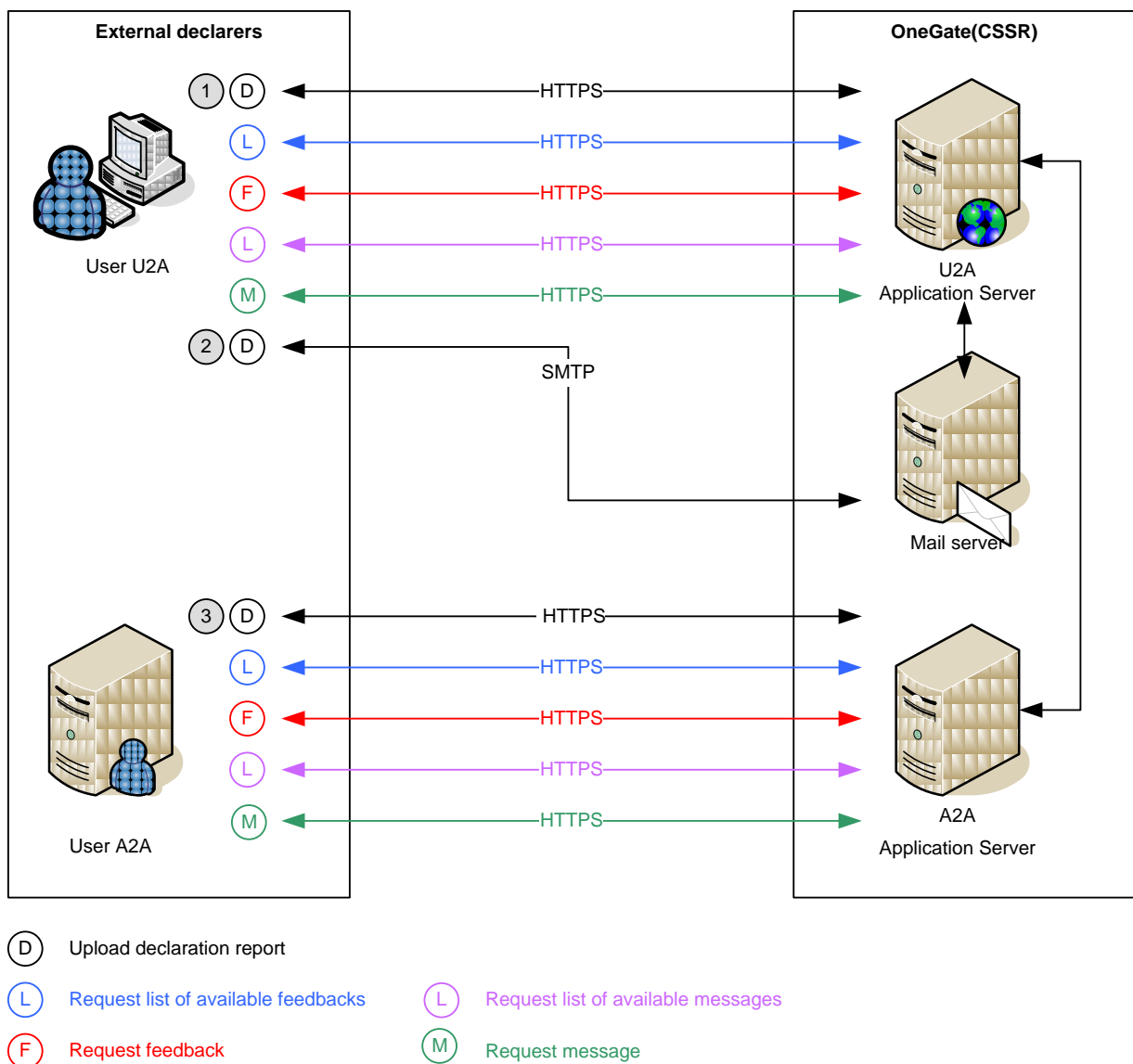


Figure 1 - General overview

2.2 Scope of intended use

OneGate(CSSR) can be used for all business domain where data must be collected via Internet. Only the authorized users can use the OneGate(CSSR) Entrypoints.

Once you have done the technical effort to automate the exchange for a certain business domain, you can reuse this implementation for all other business domains.

2.3 Scenario

For the data collection process, the data exchange with the application OneGate(CSSR) consists of **four** chronological activities:

- The declarer sends a file containing one declaration report.
- The declarer requests the list of the available feedbacks.
- The declarer requests a specific feedback.
- **The declarer submits an HTTPS request with the URL of the different attachments in the specific feedback and the server responds with the content of the attachment.**

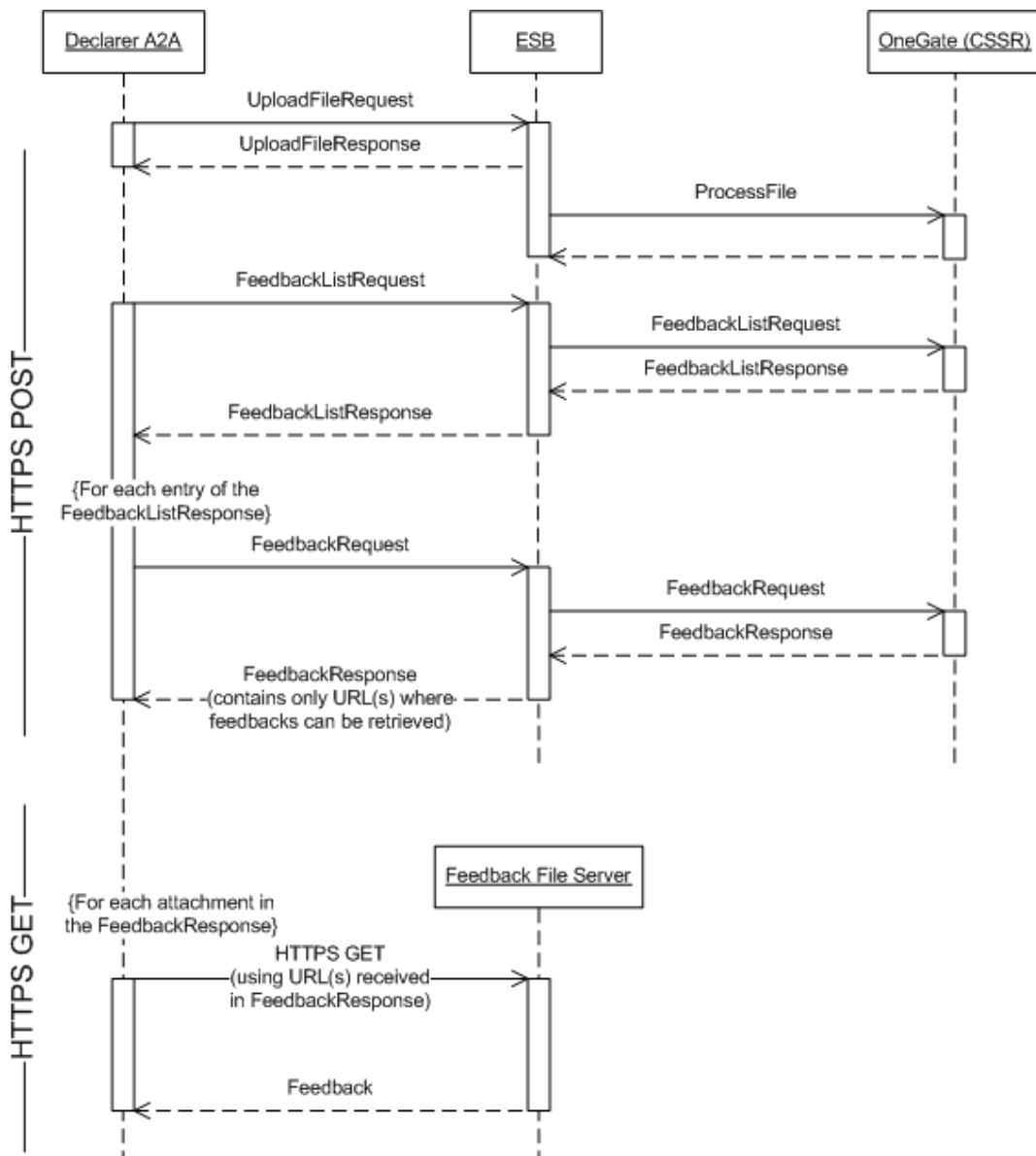


Figure 2 - Scenario of the fully automated data exchange with OneGate(CSSR) **for data collection**

For the message consultation process, the data exchange with the application OneGate(CSSR) consists of three chronological activities:

- The declarer requests the list of the available messages.
- The declarer requests a specific message.
- The declarer submits an HTTPS request with the URL of the different attachments in the specific message and the server responds with the content of the attachment.

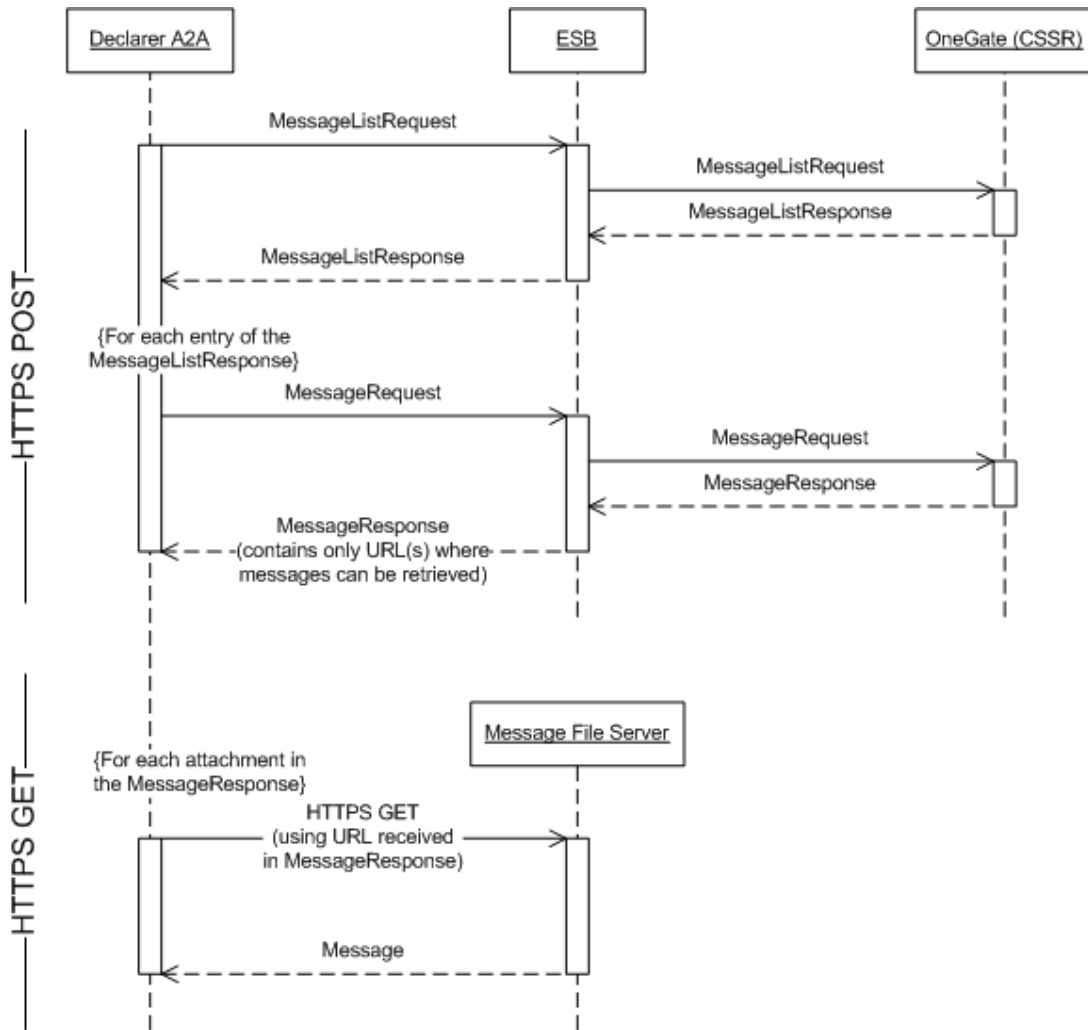


Figure 3 - Scenario of the fully automated data exchange with OneGate(CSSR) for message consultation

The content of the FeedbackResponse/MessageResponse is one (or several) URL(s) where the feedback/message can be retrieved. The feedback/message can be retrieved by using an https-request (GET request) to this URL.

3. Generalities

This chapter describes the generalities about the OneGate(CSSR) Entrypoints: the communication protocol, the authentication, the authorization and the binary data format.

3.1 Communication protocol

The communication protocol used is HTTPS with SSLv3 certificate based authentication.

The method invocation is POST, **except for the 'GetFeedback' (using the url received in the FeedbackResponse), which will be a HTTPS GET.**

Important : The client must allow a http 302 redirect for authentication ! After authentication, a http 302 is received and the request is redirected to the service. Take care :

- there are two redirects (http 302)
 - a redirect to the NBB login proxy (/nbbloginproxy/authenticationbackend)
 - after the NBB login proxy decided you have access to the requested url, you are redirected again to the original url (ex : <https://bcdpack-test.nbb.be/soap/nbb>)
- the client has to follow these redirections with a POST. Be carefull not to transform the request from a POST to a GET ! The backend server is waiting for a POST
- after each redirection, we continue using the same protocol (always HTTPS)
- (persistency) cookies are necessary to handle these redirections. Otherwise the request will be redirected to the login proxy again, and loop.

Some languages provide a library that manages these redirections for you. Java developers can use the Apache HttpClient library.

3.2 Authentication

The communication protocol used is HTTPS with SSLv3 certificate based authentication. The certificate based authentication means that remote users get authenticated using a X509 certificate.

The application OneGate(CSSR) recognizes the certificate from NBB, Global Sign, Certipost and Isabel. If you never used your certificate to access one of the NBB applications, you need to register your certificate. If you try to consume one of the HTTPS endpoints without having registered your certificate before, you will receive an HTTPS 401 error code.

You can find more information about the certificate policy and registration in the "OneGate(CSSR) - End user manual" (Ref.[1]) ; and about the NBB certificates in the referred documents [5] to [8].

3.3 Authorization

When your certificate is registered at the NBB, you must request access with this certificate to the application OneGate(CSSR). The authorization process is based on security role. To request the role that will give you access to the application OneGate(CSSR), you need to follow the procedure described in document [1]. Once the authority approved your access request, you will be able to use your certificate to automate the data exchange and have access to the online application.

If you try to consume one of the HTTPS Entrypoints without having requested access before, you will receive an HTTPS 403 error code.

3.4 Secure data

The communication protocol between the client and OneGate is always HTTPS which guarantees confidentiality of the data exchange. However, for some business domains additional security measures can be taken.

OneGate(CSSR) supports the exchange of secure data. By secure, we mean signed data or signed and encrypted data. Depending of the data sensitivity, the business will be required to send the data:

- without additional signing or encryption
- signed: to authenticate the sender and guarantee the data integrity
- signed and encrypted:
 - authenticate the sender and guarantee the data integrity
 - guarantee that only the receiver can read the data

Exchange of signed and/or encrypted files from/to the NBB will occur with files which comply with the S/MIMEv2 standard described in document referenced by [4].

3.5 Certificate

User certificate: read 3.1 Communication protocol.

The communication protocol used is HTTPS with SSLv3 certificate based authentication.

The method invocation is POST, except for the "GetFeedback" (using the url received in the FeedbackResponse), which will be an HTTPS GET.

Important: the client must allow http 302 redirect for authentication! After authentication, an http 302 is received and the request is redirected to the service.

Authentication to know which certificate can be used to sign data.

Server certificate: the public key of the server certificate used to encrypt/sign data is available on the OneGate(CSSR) web site.

3.6 Binary data

The data exchange between the declarer and the application OneGate(CSSR) will be done using different file formats (XML, PDF, Word document, ...) and different levels of security.

In native HTTPS the payload can be passed without encoding or in base64 encoded format.

3.7 Volume

The size of the request cannot exceed 10 MB. The size of the payload of the HTTPS GET reply is limited depending on the specific Institute/Business Domain. To limit this size, payload can be compressed. If zip is used, the content-type of the attachment will be "application/zip".

3.8 Useful tools

Before automated the implementation of the usage of the HTTPS Entrypoints, you can easily test it using cURL (<http://curl.haxx.se>):

cURL is a command line tool for transferring files with URL syntax, supporting FTP, FTPS, HTTP, HTTPS, SCP, SFTP, TFTP, TELNET, DICT, LDAP, LDAPS and FILE. cURL supports SSL certificates, HTTPS POST, HTTPS PUT, FTP uploading, HTTPS form based upload, proxies, cookies, user+password authentication (Basic, Digest, NTLM, Negotiate, kerberos...), file transfer resume, proxy tunneling and a busload of other useful tricks.

4. HTTPS Entrypoints description

4.1 Overview

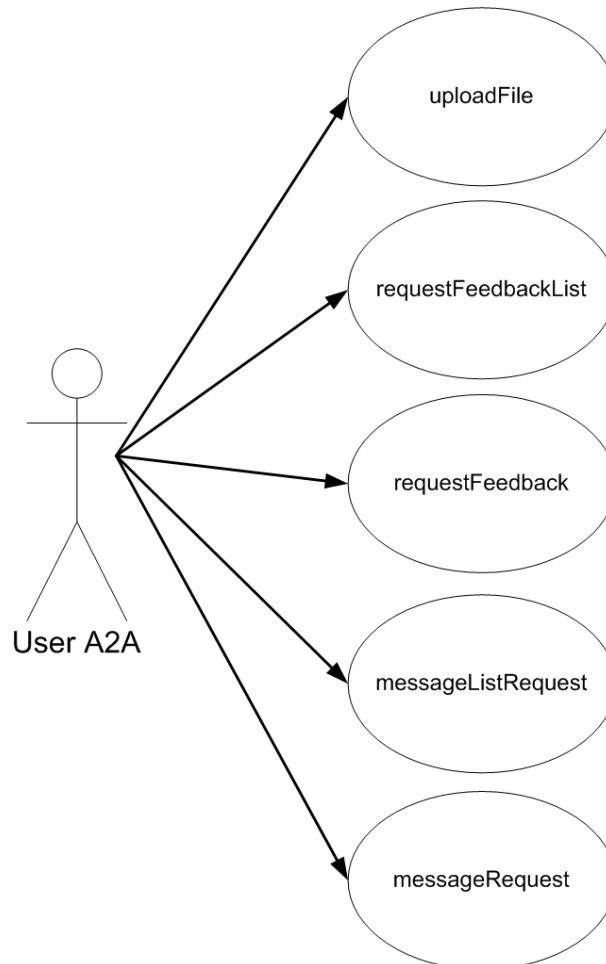


Figure 4 - HTTPS Entrypoints services

Figure 4 gives an overview of the HTTPS Entrypoints used by the declarer to fully automate the data exchange with the application OneGate(CSSR). As a reminder, the data exchange consists of sending data and retrieving the associated feedback that contains the result of the validation of the imported data.

4.2 Upload file

4.2.1 Description

The declarer uses the service "Upload file" to send a file with a declaration report to be processed by the application OneGate(CSSR) and receives a ticket in return. This ticket is used as an acknowledgement of the file receipt and will be used to uniquely identify the file transfer. The processing of the file is done asynchronously following the FIFO principle. Due to the asynchronous process, the validation report will not be available immediately and there is a delay between the file upload and the registration of the file reception in OneGate(CSSR).

Note that the processing of the declaration can be also followed via the online OneGate(CSSR) application using the function "Exchanges > File exchange log".

The sent file contains a declaration report that must follow the data exchange protocol published by the business where the XML schema and the security level will be specified.

List of the supported XML protocols:

- Declaration Report (cfr. [3]) ; all new reporting
- XML protocol of the old CSSR application
- Other specific protocols like for the CKO 2

Supported security level to apply to the sending file:

- None
- Signed
- Signed and encrypted

4.2.2 URL

Test: <https://onagate-a2a-test.nbb.be/crs-esb/invoke/uploadFile-v2-0>

Production: <https://onagate-a2a.nbb.be/crs-esb/invoke/uploadFile-v2-0>

These are general URLs for OneGate. Each Business Domain has his one's own URL. Please consult the URL document for the desired Institute / Business Domain !

4.2.3 HTTPS Request

4.2.3.1 Body

The payload – body of the request – contains the declaration report to upload to OneGate(CSSR).

Before being sent, the payload must fulfil the following requirements:

- The declaration report must be valid against the XML protocol fixed by the business.
- Only one file can be uploaded by request.
- The file can be compressed. In this case, the zip file can contains only one file.
- Following the business requirements, the file must be signed or signed and encrypted

So to generate the payload, you need to execute the following steps:

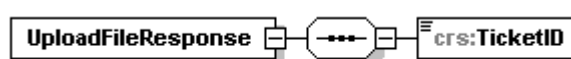
- Validate the file against the XML schema ; *optional but recommended*
- Zip file ; *optional but recommended*
- Sign the file with your private key ; *required or not by the business requirement*
- Encrypt the file with the OneGate(CSSR) public key ; *required or not by the business requirement*

4.2.3.2 Header

You can give a name to the file uploaded by specifying the HTTPS header "filename". This header is optional.

4.2.4 HTTPS Response

If the OneGate (CSSR) has successfully received the payload, the response returns a status code 200 and the ticket ID associated to the file upload. The response is sent in the XML format described below.



UploadFileResponse	
Description	Contains the ticket id associated with the file exchange.
XML format	ComplexType
Children	TicketID

TicketID	
Description	Identifies the file transfer uniquely. The ticket ID is used as an acknowledgement of the file receipt and will be requested in case of problem by the Service Desk to be able to detect the problem.
XML format	xs:string whitespace = collapse
Validation	Required

4.2.5 Error

Status code – Reason phrase	Corrective action
400 range	Client error. The request contains bad syntax or cannot be fulfilled. An overview of HTTP error codes can be found in http://www.rfc-editor.org/rfc/rfc2616.txt .
500 range	Server error. The server failed to fulfil request. Please try again later and if problem persists, contact

4.3 Request list of available feedback

4.3.1 Description

The service "requestFeedbackList" is used to request the list of feedback identifiers available. Only the identifier of the feedback associated with a file sent with this user will be sent back. The feedback associated with files sent by another user but for a common declarer will not be sent back.

You can choose between requesting a list of either new feedbacks or feedbacks associated to files sent during a specified time frame. The second option offers you the possibility to request feedbacks that have been retrieved earlier.

4.3.2 URL

Test: <https://onagate-a2a-test.nbb.be/crs-esb/invoke/requestFeedbackList-v2-0>

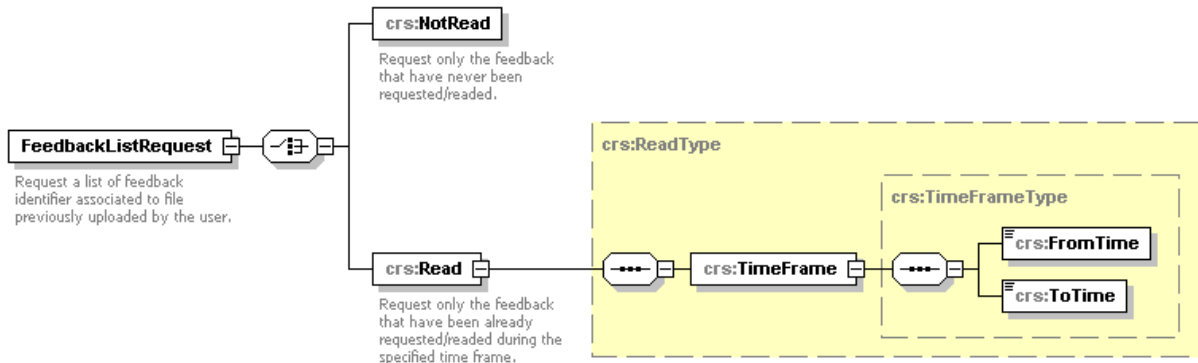
Production: <https://onagate-a2a.nbb.be/crs-esb/invoke/requestFeedbackList-v2-0>

These are general URLs for OneGate. Each Business Domain has his one's own URL. Please consult the URL document for the desired Institute/Business Domain !

4.3.3 HTTPS Request

4.3.3.1 Body

The request of the feedback list must be fulfill the XML format described below.



FeedbackListRequest	
Description	Contains the attributes to request the list of feedback identifiers available for the requester.
XML format	ComplexType
Children	NotRead Read

NotRead	
Description	Used to request the identifiers of new feedback.
XML format	ComplexType
Validation	Empty element

Read	
Description	Used to request the identifiers of feedback requested earlier during the specified time frame.
XML format	ComplexType
Children	TimeFrame

TimeFrame	
Description	Used to specify the time frame during which the file has been retrieved for the first time from the application OneGate(CSSR).
XML format	ComplexType
Children	FromTime ToTime

FromTime	
Description	Timestamp identifying the start of the time frame.
XML format	xs:dateTime
Validation	Required

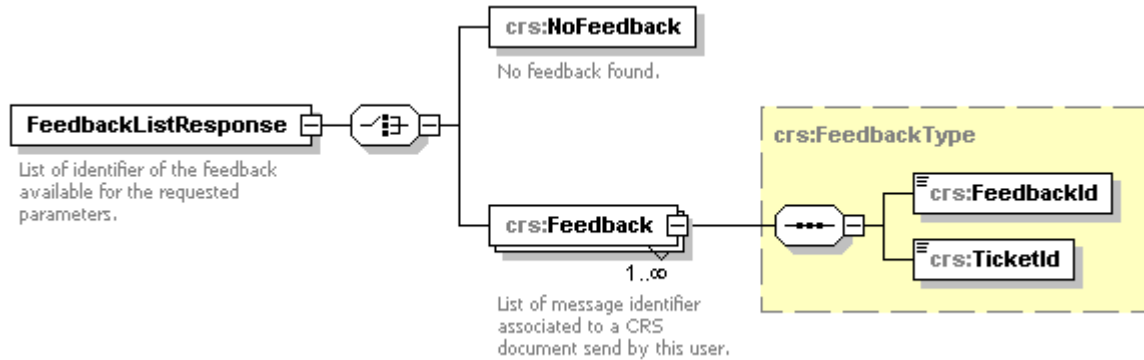
ToTime	
Description	Timestamp identifying the end of the time frame.
XML format	xs:dateTime
Validation	Required

4.3.3.2 Header

No specific header.

4.3.4 HTTPS Response

If OneGate (CSSR) has successfully processed the request, the response returns a status code 200 and the list of the available feedbacks in the XML format described below.



FeedbackListResponse	
Description	Contains the list of the available feedback identifiers.
XML format	ComplexType
Children	NoFeedback Feedback

NoFeedback	
Description	No feedback found for the specified search criteria. <ul style="list-style-type: none"> • Not read: no new feedback • Read: no feedback retrieved during the specified time frame
XML format	ComplexType
Validation	Empty element

Feedback	
Description	Contains the information about the feedback available for download.
XML format	ComplexType Minimum occurrence: 1 Maximum occurrence: unbounded
Children	FeedbackId TicketId

FeedbackId	
Description	Identifier of the feedback
XML format	xs:nonNegativeInteger. minExclusive: 0
Validation	Required

TicketID	
Description	Identifier of the file transfer to whom the feedback is associated.
XML format	xs:string whitespace: collapse
Validation	Required

4.3.5 Error

Status code – Reason phrase	Corrective action
400 Identification with code 'userId' not found	You have access to OneGate(CSSR) but not to the requested data. Please check, in the specific business documentation, if the URL used to send your request is the right one. If the right URL is used, please contact the access manager to request to verify the access rights linked to your credentials.
4xx range	Client error. The request contains bad syntax or cannot be fulfilled. An overview of HTTP error codes can be found in http://www.rfc-editor.org/rfc/rfc2616.txt .
5xx range	Server error. The server failed to fulfil request. Please try again later and if problem persists, contact the service desk.

4.4 Request a feedback

4.4.1 Description

The service "feedbackRequest" is used to request a specific feedback by providing its identifier. The format of the feedback must follow the business requirement in terms of format and security level applied.

If the business requires that you send your file signed and/or encrypted, the feedback will follow the same requirement and will be signed and/or encrypted.

4.4.2 URL

Test: <https://onagate-a2a-test.nbb.be/crs-esb/invoke/requestFeedback-v2-0>

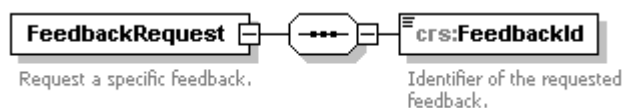
Production: <https://onagate-a2a.nbb.be/crs-esb/invoke/requestFeedback-v2-0>

These are general URLs for OneGate. Each Business Domain has his proper own URL. Please consult the URL document for the desired Institute/Business Domain !

4.4.3 HTTPS Request

4.4.3.1 Body

The request of a specific feedback must be fulfill the XML format described below.



FeedbackRequest	
Description	Contains the information about the requested feedback
XML format	ComplexType
Children	FeedbackId

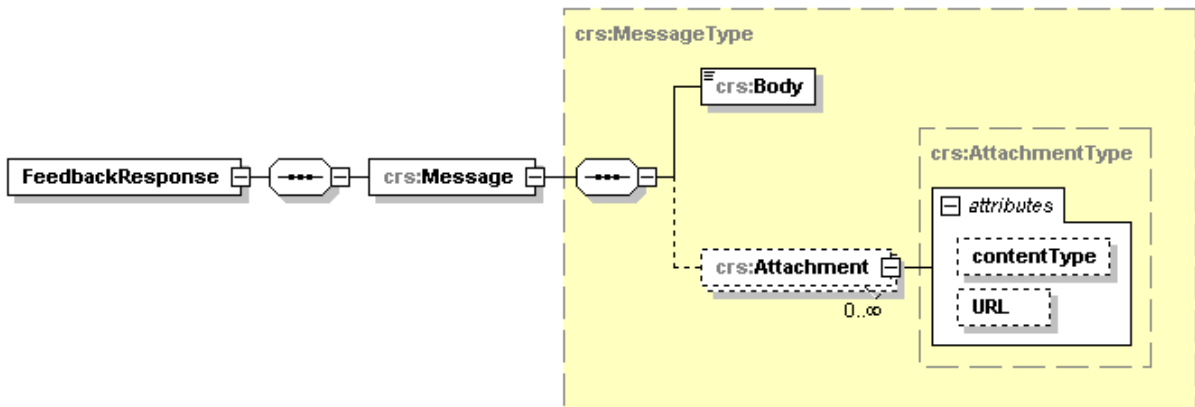
FeedbackId	
Description	Identifier of the requested feedback
XML format	xs:nonNegativeInteger minExclusive: 0 ⁵
Validation	Required

4.4.3.2 Header

No specific header.

4.4.4 HTTPS Response

If OneGate (CSSR) has successfully processed the request, the response returns a status code 200 and the feedback in the XML format described below.



FeedbackResponse	
Description	Contains the requested feedback
XML format	ComplexType
Children	Message

Message	
Description	The message is the feedback which contains at least one body with optional attachment
XML format	ComplexType
Children	Body Attachment

⁵ Max value: 2⁶³ - 1 = 9223372036854775807

Body	
Description	Body of the message in plain text (can be empty). Example: Validation report for ticket number [480]
XML format	xs:string
Validation	Required

Attachment			
Description	Attachment of the message <ul style="list-style-type: none"> • @contentType specifies the type of the content using the Internet media type. • @url specifies the URL where the feedback can be retrieved using an HTTPS GET request 		
XML format	text/xml minOccurs: 0 maxOccurs: unbounded		
Attributes	Name	Type	Value
	contentType	xs:string	"text/xml" or "application/zip"
	URL	xs:string	
Validation	Empty element, Optional		

4.4.5 Error message

Status code – Reason phrase	Corrective action
400 - Identification with code 'userId' not found	You have access to OneGate(CSSR) but not to the requested data. Please check, in the specific business documentation, if the URL used to send your request is the right one. If the right URL is used, please contact the access manager to request to verify the access rights linked to your credentials.
400 - Feedback with id '<FeedbackId>' not found.	This error can occur when <ul style="list-style-type: none"> • The element "TicketId" from the FeedbackListResponse is used instead of the "FeedbackId". • the requested feedback doesn't exist. • a different certificate is used to request the list of feedbacks and the feedback. Note that the same certificate must be used to upload a file and request the associated feedbacks.
4xx range	Client error. The request contains bad syntax or cannot be fulfilled. An overview of HTTP error codes can be found in http://www.rfc-editor.org/rfc/rfc2616.txt .
5xx range	Server error. The server failed to fulfil request. Please try again later and if problem persists, contact the service desk.

4.5 GET an attachment of a feedback

4.5.1 Description

Each attachment of a feedbackResponse should be retrieved by its unique url (HTTPS GET). This implies that a A2A authenticated user, that has access to the HTTP-entrypoints, will also have the ability to get access to the requested URL. The application will verify that the A2A user has the authority to access the specified URL. This is done to prevent a participant from accessing data of another participant.

4.5.2 URL

Test: URL found in the FeedbackResponse

Production: URL found in the FeedbackResponse

4.5.3 HTTPS Request

The attachment can be accessed automatically (in the client application) or semi-automatically (by using cURL) to perform a "GET" operation on the URL.

4.5.4 Error message

Status code – Reason phrase	Corrective action
4xx range	Client error. The request contains bad syntax or cannot be fulfilled. An overview of HTTP error codes can be found in http://www.rfc-editor.org/rfc/rfc2616.txt .
5xx range	Server error. The server failed to fulfil the request. Please try again later and if problem persists, contact the service desk.

4.6 Request list of available messages

4.6.1 Description

The service "messageListRequest" is used to request the list of message identifiers available. Only the identifier of the message destined to this user will be sent back. The message associated with the userId of another user but for a common declarer will not be sent back.

You can choose between requesting a list of either new messages or messages already consulted during a specified time frame. The second option offers you the possibility to request messages that have been retrieved earlier.

4.6.2 URL

Test: <https://onagate-a2a-test.nbb.be/crs-esb/invoke/requestMessageList-v2-0>

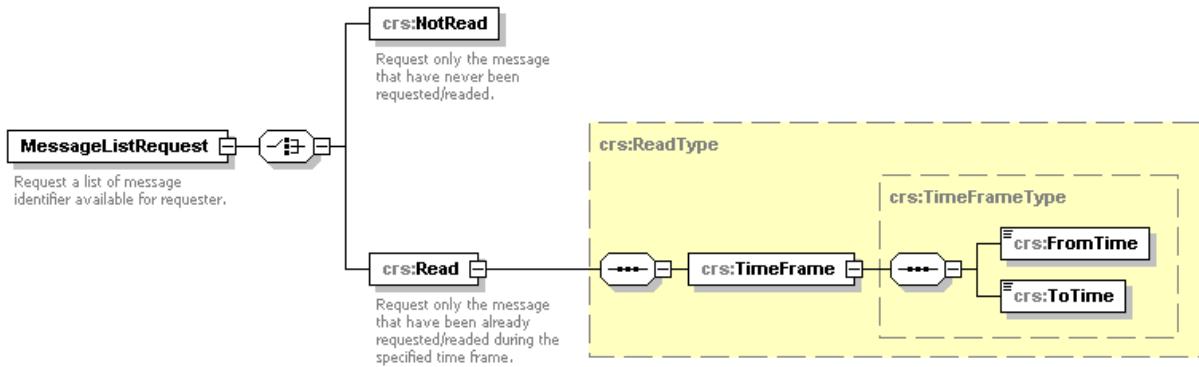
Production: <https://onagate-a2a.nbb.be/crs-esb/invoke/requestMessageList-v2-0>

These are general URLs for OneGate. Each Business Domain has his one's own URL. Please consult the URL document for the desired Institute/Business Domain !

4.6.3 HTTPS Request

4.6.3.1 Body

The request of a specific feedback must be fulfill the XML format described below.



MessageListRequest	
Description	Contains the attributes to request the list of message identifiers available for the requester.
XML format	ComplexType
Children	NotRead Read

NotRead	
Description	Used to request the identifiers of new message.
XML format	ComplexType
Validation	Empty element

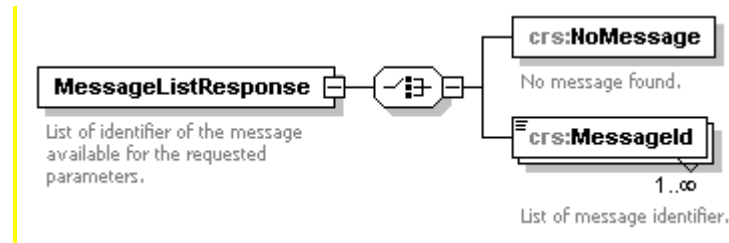
Read	
Description	Used to request the identifiers of message requested earlier during the specified time frame.
XML format	ComplexType
Children	TimeFrame

TimeFrame	
Description	Specified the time frame during which the message has been retrieved the first time.
XML format	ComplexType
Children	FromTime ToTime

FromTime	
Description	Timestamp identifying the start of the time frame.
XML format	xs:dateTime
Validation	Required

ToTime	
Description	Timestamp identifying the end of the time frame.
XML format	xs:dateTime
Validation	Required

4.6.4 HTTPS Response



MessageListResponse	
Description	Contains the list of the available message identifiers.
XML format	ComplexType
Children	NoMessage MessageId

NoMessage	
Description	No message found for the specified search criteria. <ul style="list-style-type: none"> Not read: no new message Read: no message read during the specified time frame
XML format	ComplexType
Validation	Empty element

MessageId	
Description	Identifier of the message
XML format	xs:nonNegativeInteger. minExclusive: 0 ⁶
Validation	Required

4.6.5 Error message

Status code – Reason phrase	Corrective action
400 Identification with code 'userId' not found	You have access to OneGate(CSSR) but not to the requested institute. Please check, in the specific business documentation, if the URL used to send your request is the right one. If the right URL is used, please contact the access manager to request to verify the access rights linked to your credentials.
4xx range	Client error. The request contains bad syntax or cannot be fulfilled.

⁶ Max value: 2⁶³ - 1 = 9223372036854775807

	An overview of HTTP error codes can be found in http://www.rfc-editor.org/rfc/rfc2616.txt .
5xx range	Server error. The server failed to fulfil the request. Please try again later and if problem persists, contact the service desk.

4.7 Request a message

4.7.1 Description

The service "messageRequest" is used to request a specific message by providing its identifier. The format of the message must follow the business requirement in terms of format and security level applied.

If the business required that you send your file signed and/or encrypted, the message will follow the same requirement and will be signed and/or encrypted.

4.7.2 URL

Test: <https://onagate-a2a-test.nbb.be/crs-esb/invoke/requestMessage-v2-0>

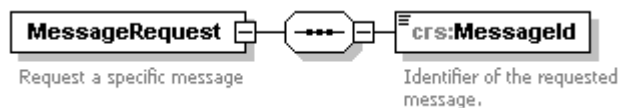
Production: <https://onagate-a2a.nbb.be/crs-esb/invoke/requestMessage-v2-0>

These are general URLs for OneGate. Each Business Domain has his proper own URL. Please consult the URL document for the desired Institute/Business Domain !

4.7.3 HTTPS Request

4.7.3.1 Body

The request of a specific feedback must be fulfill the XML format described below.

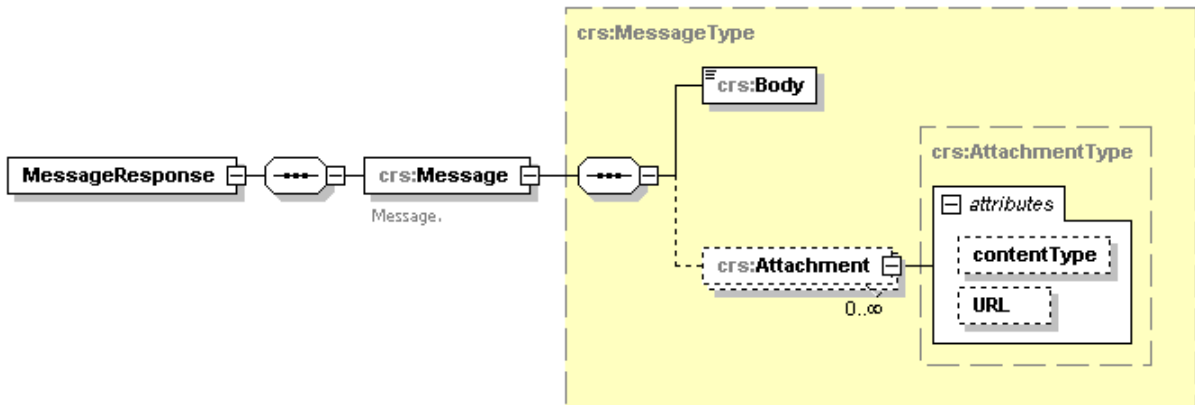


MessageRequest	
Description	Contains the information about the requested message
XML format	ComplexType
Children	MessageId

MessageId	
Description	Identifier of the requested message
XML format	xs:nonNegativeInteger minExclusive: 0 ⁷
Validation	Required

⁷ Max value: 2⁶³ - 1 = 9223372036854775807

4.7.4 HTTPS Response



MessageResponse	
Description	Contains the requested message
XML format	ComplexType
Children	Message

Message	
Description	The message contains at least one body and optional attachments
XML format	ComplexType
Children	Body Attachment

Body	
Description	Body of the message in plain text (can be empty). Example: Message [142].
XML format	xs:string
Validation	Required

Attachment			
Description	Attachment of the message <ul style="list-style-type: none"> • @contentType specifies the type of the message using the Internet media type. • @URL specifies the URL where the message can be retrieved using an https-request (GET). 		
XML format	ComplexType		
Attributes	Name	Type	Default
	contentType	xs:string	text/xml ⁸
	URL	xs:string	
Validation	Empty element, Optional		

⁸ contentType can also have the following values: "application/zip", "application/pdf" or "text/csv"

4.7.5 Error message

Status code – Reason phrase	Corrective action
400 - Identification with code 'userId' not found	You have access to OneGate(CSSR) but not to the requested institute. Please check, in the specific business documentation, if the URL used to send your request is the right one. If the right URL is used, please contact the access manager to request to verify the access rights linked to your credentials.
400 - Message with id '<MessageId>' not found.	The requested message doesn't exist.
4xx range	Client error. The request contains bad syntax or cannot be fulfilled. An overview of HTTP error codes can be found in http://www.rfc-editor.org/rfc/rfc2616.txt .
5xx range	Server error. The server failed to fulfil request. Please try again later and if problem persists, contact the service desk.

4.8 GET an attachment of a message

4.8.1 Description

Each attachment of a MessageResponse should be retrieved by its unique url (HTTPS GET). This implies that a A2A authenticated user, that has access to the HTTP-entrpoints, will also have the ability to get access to the requested URL. The application will verify that the A2A user has the authority to access the specified URL. This is done to prevent a participant from accessing data of another participant.

4.8.2 URL

Test: URL found in the MessageResponse

Production: URL found in the MessageResponse

4.8.3 HTTPS Request

The attachment can be accessed automatically (in the client application) or semi-automatically (by using cURL) to perform a "GET" operation on the URL.

4.8.4 Error message

Status code – Reason phrase	Corrective action
4xx range	Client error. The request contains bad syntax or cannot be fulfilled. An overview of HTTP error codes can be found in http://www.rfc-editor.org/rfc/rfc2616.txt .
5xx range	Server error. The server failed to fulfil the request. Please try again later and if problem persists, contact the service desk.

5. Error codes

If the request could not be processed due to a client error or a server error, the response will contain **respectively** a status code **in 400 or 500 range**.

An overview of HTTP error codes can be found in

5.1 Client error

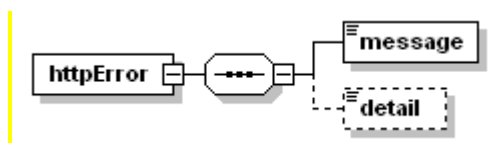
If your request contains something wrong, the response returns a status code **in the 400 range** and a reason phrase given an indication that the problem is located at the client side. In this case, please correct your request before send it again.

5.1.1 Header

Status code - Reason phrase	Detail
400 – Bad request	Consult the body of the response for more details.
403	Different problems can cause this error code: <ol style="list-style-type: none"> " User certificate missing or rejected !" <ul style="list-style-type: none"> No certificate found in your request You need to register your certificate or a new one. Client certificate has expired or is not yet valid <ul style="list-style-type: none"> You need to request a new certificate and restart the registration procedure The request exceeds the max allowed content length <ul style="list-style-type: none"> Your request exceeds the max size specified in Volume.
Other status code	Please consult http://www.rfc-editor.org/rfc/rfc2616.txt .

5.1.2 Body

If the HTTP error code is 400, the body can contain more details about the error cause. This details will be presented using the following XML structure:



where "message" contains the error message and "details" can contains more informations about this error.

5.2 Server error

If the server cannot process your request due to a technical problem, the response returns a status code **500 (or in the 500 range)** and a reason phrase given an indication of the problem occurred on the server side. In this case, please retry to send your request and if the problem persists, please contact the NBB IT Servicedesk (+ 32 2 221 40 60 ; servicedesk@nbb.be).

6. Appendix - Sample using cURL

6.1 Prerequisites

6.1.1 Client certificate

cURL uses only the pem certificate file format. So if your certificate has not a .pem extension, you need to convert your certificate to a pem.

How to convert certificate .pfx to .pem ?⁹

If you have requested and installed a certificate onto a Windows server using the *Internet Information Service (IIS)* certificate wizard, you can export that certificate with its private key to a *Personal Information Exchange (PFX)* file. To import this certificate onto the Access Gateway, you must convert the PFX file to the unencrypted PEM format.

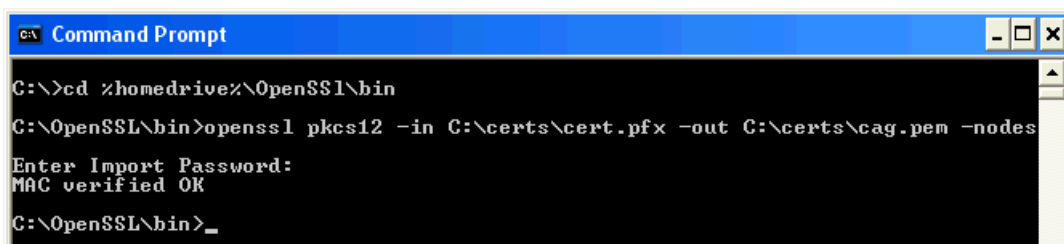
You may use the open-source utility **OpenSSL** to perform the conversion from PFX to PEM. You can download a Win32 distribution of OpenSSL here:

<http://www.slproweb.com/products/Win32OpenSSL.html>

You might also need C++ re-distributable files if you want to use OpenSSL which can be obtained at the following URL: <http://www.microsoft.com/downloads/details.aspx?FamilyID=9B2DA534-3E03-4391-8A4D-074B9F2BC1BF&displaylang=en>

To convert a PFX file to a PEM file, follow these steps on a Windows machine:

1. Download and install the Win32 OpenSSL (Win32 OpenSSL v0.9.8i) package from <http://www.slproweb.com/products/Win32OpenSSL.html>
2. Create a folder **c:\certs** and copy the file **yourcert.pfx** into the **c:\certs** folder.
3. Open a command prompt and change into the **OpenSSL\bin** directory:
cd %homedrive%\OpenSSL\bin
4. Type the following command to convert the PFX file to an unencrypted PEM file (all on one line):
openssl pkcs12 -in c:\certs\yourcert.pfx -out c:\certs\cag.pem -nodes
5. When prompted for the import password, enter the password you used when exporting the certificate to a PFX file. You should receive a message that says **MAC verified OK**.



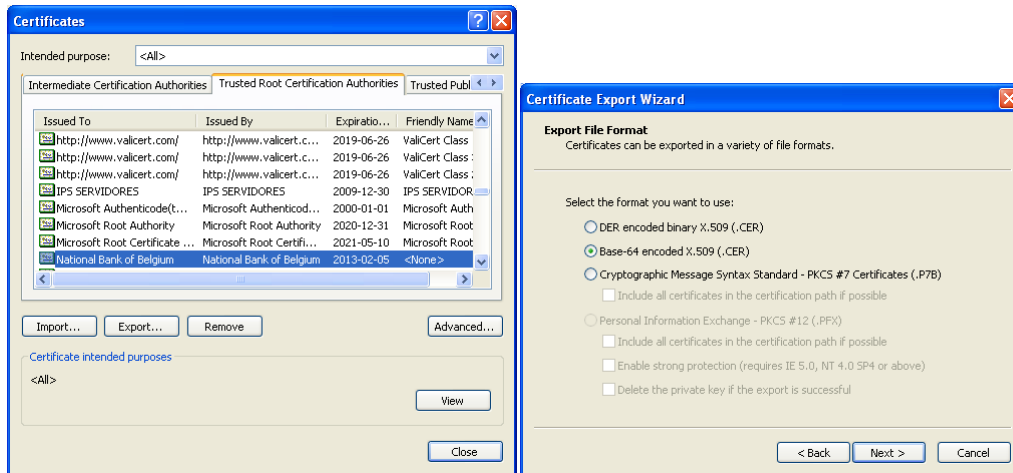
```

C:\>cd %homedrive%\OpenSSL\bin
C:\OpenSSL\bin>openssl pkcs12 -in C:\certs\cert.pfx -out C:\certs\cag.pem -nodes
Enter Import Password:
MAC verified OK
C:\OpenSSL\bin>_
  
```

6. You can find the converted certificate in the specified location **c:\certs\cag.pem**

⁹ Procedure described by Citrix in the following article: <http://support.citrix.com/article/CTX106028>

6.1.2 Server certificate



6.2 Command cURL

6.2.1 Parameters

-b/--cookie <name=data>

(HTTP) Pass the data to the HTTP server as a cookie. It is supposedly the data previously received from the server in a "Set-Cookie:" line. The data should be in the format "NAME1=VALUE1; NAME2=VALUE2". If no '=' symbol is used in the line, it is treated as a filename to use to read previously stored cookie lines from, which should be used in this session if they match. Using this method also activates the "cookie parser" which will make curl record incoming cookies too, which may be handy if you're using this in combination with the -L/--location option. The file format of the file to read cookies from should be plain HTTP headers or the Netscape/Mozilla cookie file format.

NOTE that the file specified with -b/--cookie is only used as input. No cookies will be stored in the file. To store cookies, use the -c/--cookie-jar option or you could even save the HTTP headers to a file using -D/--dump-header!

If this option is set more than once, the last one will be the one that's used.

--data-binary <data>

(HTTP) This posts data exactly as specified with no extra processing whatsoever. If you start the data with the letter @, the rest should be a filename. Data is posted in a similar manner as --data-ascii does, except that newlines are preserved and conversions are never done.

-E/--cert <certificate[:password]>

(SSL) Tells curl to use the specified certificate file when getting a file with HTTPS or FTPS. The certificate must be in PEM format. If the optional password isn't specified, it will be queried for on the terminal. Note that this option assumes a "certificate" file that is the private key and the private certificate concatenated! See --cert and --key to specify them independently.

-H/--header <header>

(HTTP) Extra header to use when getting a web page. You may specify any number of extra headers. Note that if you should add a custom header that has the same name as one of the internal ones curl would use, your

externally set header will be used instead of the internal one. This allows you to make even trickier stuff than curl would normally do. You should not replace internally set headers without knowing perfectly well what you're doing. Remove an internal header by giving a replacement without content on the right side of the colon, as in: `-H "Host:"`. Curl will make sure that each header you add/replace is sent with the proper end-of-line marker, you should thus not add that as a part of the header content: do not add newlines or carriage returns, they will only mess things up for you.

-L/--location

(HTTP/HTTPS) If the server reports that the requested page has moved to a different location (indicated with a Location: header and a 3XX response code), this option will make curl redo the request on the new place. If used together with `-i/--include` or `-I/--head`, headers from all requested pages will be shown. When authentication is used, curl only sends its credentials to the initial host. If a redirect takes curl to a different host, it won't be able to intercept the user+password. See also `--location-trusted` on how to change this. You can limit the amount of redirects to follow by using the `--max-redirs` option.

When curl follows a redirect and the request is not a plain GET (for example POST or PUT), it will do the following request with a GET if the HTTP response was 301, 302, or 303. If the response code was any other 3xx code, curl will re-send the following request using the same unmodified method.

--post302

Tells curl to respect RFC 2616/10.3.2 and not convert POST requests into GET requests when following a 302 redirection. The non-RFC behaviour is ubiquitous in web browsers, so curl does the conversion by default to maintain consistency. However, a server may require a POST to remain a POST after such a redirection. This option is meaningful only when using `-L/--location` (Added in 7.19.1)

-v/--verbose

Makes the fetching more verbose/talkative. Mostly useful for debugging. A line starting with `>` means "header data" sent by curl, `<` means "header data" received by curl that is hidden in normal cases, and a line starting with `*` means additional info provided by curl. Note that if you only want HTTPS headers in the output, `-i/--include` might be the option you're looking for.

6.2.2 Upload file

6.2.2.1 Command cURL

With file name:

```
curl -v -L -b cookies --post302 -E "<certificate.pem:password>" --data-binary
"@<filepath>"
-H " filename: <filename>"
-k "https://onagate-a2a.nbb.be/crs-esb/invoke/uploadFile-v2-0"
```

Without file name:

```
curl -v -L -b cookies --post302 -E "<certificate.pem:password>" --data-binary
"@<filepath>"
-k " https://onagate-a2a.nbb.be/crs-esb/invoke/uploadFile-v2-0"
```

6.2.2.2 HTTPS Request

Sample of declarationReport.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<DeclarationReport xmlns="http://www.onagate.eu/2010-01-01">
  <Administration>
    <From declarerType="KBO">0100200300</From>
    <To>NBB</To>
```

```

<Domain>FRO</Domain>
<Response feedback="true">
  <Email>frank.osaer@nbb.be</Email>
  <Language>EN</Language>
</Response>
</Administration>
<Report close="true" date="2009-10-01" code="FROS_RPT_2">
  <Data action="replace" form="FRO_REP_FORM_2">
    <Item>
      <Dim prop="AMOUNT">100</Dim>
      <Dim prop="IMPEXT">IMPORT</Dim>
      <Dim prop="INTEXT2">INTRA</Dim>
    </Item>
    <Item>
      <Dim prop="AMOUNT">200</Dim>
      <Dim prop="IMPEXT">IMPORT</Dim>
      <Dim prop="INTEXT2">EXTRA</Dim>
    </Item>
  </Data>
</Report>
</DeclarationReport>

```

6.2.2.3 HTTPS Response

```

<?xml version="1.0"?>
<crs:UploadFileResponse
  xmlns:crs="http://www.onegate.eu/2010-09-01/esb">
  <crs:TicketID>632</crs:TicketID>
</crs:UploadFileResponse>

```

6.2.3 Request list of feedback

6.2.3.1 Command cURL

```

curl -v -L -b cookies --post302 -E "<certificate.pem:password>" --data-binary
"@requestFeedbackList.xml"
-k "https://onegate-a2a.nbb.be/crs-esb/invoke/ requestFeedbackList-v2-0"

```

6.2.3.2 HTTPS Request

Sample of requestFeedbackList.xml:

```

<?xml version="1.0"?>
<crs:FeedbackListRequest xmlns:crs="http://www.onegate.eu/2010-09-01/esb">
  <crs:NotRead/>
</crs:FeedbackListRequest>

```

6.2.3.3 HTTPS Response

```

<?xml version="1.0"?>
<crs:FeedbackListResponse xmlns:crs="http://www.onegate.eu/2010-09-01/esb">
  <crs:Feedback>
    <crs:FeedbackId>60163</crs:FeedbackId>
    <crs:TicketId>490</crs:TicketId>
  </crs:Feedback>
  <crs:Feedback>
    <crs:FeedbackId>60168</crs:FeedbackId>
    <crs:TicketId>492</crs:TicketId>
  </crs:Feedback>
</crs:FeedbackListResponse>

```

6.2.4 Request feedback

6.2.4.1 Command cURL

```
curl -v -L -b cookies --post302 -E "<certificate.pem:password>" --data-binary
"@requestFeedback.xml"
-k " https://onagate-a2a.nbb.be/crs-esb/invoke/requestFeedback-v2-0"
```

6.2.4.2 HTTPS Request

Sample of requestFeedback.xml:

```
<?xml version="1.0"?>
<crs:FeedbackRequest xmlns:crs="http://www.onagate.eu/2010-09-01/esb">
  <crs:FeedbackId>60163</crs:FeedbackId>
</crs:FeedbackRequest>
```

6.2.4.3 HTTPS Response

```
<?xml version="1.0"?>
<crs:FeedbackResponse xmlns:crs="http://www.onagate.eu/2010-09-01/esb">
  <esb:Message>
    <esb:Body>Next feedback for {TICKET_ID:123}
      {REPORT_REFERENCE_TEXT:AFE457e8sd4} {STEP;3}</esb:Body>
    <esb:Attachment contentType="text/xml"
URL="https://someurl.be/dummy=142254" />
    <esb:Attachment contentType="text/xml"
URL="https://someurl.be/dummy=128564" />
  </esb:Message>
</crs:FeedbackResponse>
```

6.2.5 GET attachment for feedback

6.2.5.1 Command cURL

```
curl -v -L -b cookies --post302 -E "<certificate.pem:password>"
-o "https://someurl.be/dummy=142254"
```

6.2.5.2 HTTPS Request

Sample of the HTTPS url:

```
https://someurl.be/dummy=142254
```

6.2.5.3 HTTPS Response

The xml file.

7. Definition of terms and abbreviations

Abbreviation	Description
A2A	Application to Application ; refers to the interaction between two applications.
Acknowledgment of file receipt	This acknowledgment of file receipt indicates that a file was received, that the file transfer is recorded under a identification number (Tickeld). This acknowledgment does not contain any information about the validity of the document.
Feedback OneGate(CSSR)	Feedback OneGate is the validation report generated automatically by OneGate(CSSR) when the sending file is processed. The feedback format is fixed (XML following the protocol used for the FeedbackReport).
Feedback Back-Office	The feedback back-office is a report generated manually or automatically by the specific business application that will exploit the data collected via OneGate(CSSR). This report can contain information related to a second level of validation of the reported data or other types of information such as complementary question or various information. The file format of the business report is free (HTML, xls, doc, PDF,...)
FIFO	First In First Out can be translated as "First-come, First-served". This expression describes the principle of a queue processing where what comes in first is handled first, what comes in next waits until the first is finished before being handled.
HTTPS	Hypertext Transfer Protocol Secure is a combination of the Hypertext Transfer Protocol with the SSL/TLS protocol to provide encryption and secure identification of the server.
Internet Media Type	An Internet Media Type, originally called "MIME type" or "Content-type", is a two-part identifier for file formats on the Internet. A media type is composed of at least two parts: a type, a subtype and one or more optional parameters (e.g. "image/png").
NBB	National Bank of Belgium
SOAP	Simple Object Access Protocol SOAP is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. It uses XML technologies to define an extensible messaging framework providing a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation specific semantics. (Definition from http://www.w3.org/TR/soap/)
SSL	Secure Socket Layer are cryptographic protocols that provide security for communications over networks such as the Internet.
U2A	User to Application ; refers to the interaction between an user and an application.
Web Service	A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTPS with an XML serialization in conjunction with other Web-related standards. (Definition issue from the W3C)
WSDL	Web Services Definition Language WSDL is an XML-based language for describing Web services and how to access them.