



Social Security
KSZ

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

Service Specification:

Integration and use of the service

RegisteredMail Version 1.0

To the attention of: "IT expert" aiming to integrate this Service

Table of Contents

| | | |
|------|---|-------------------------------------|
| 1 | Goal of the service | 4 |
| 1.1 | Description of the service..... | 4 |
| 1.2 | Service consumers..... | 4 |
| 2 | Document management..... | 4 |
| 2.1 | Document history | 4 |
| 2.2 | Validation | 4 |
| 2.3 | Document references..... | 4 |
| 2.4 | Goal of the document..... | 5 |
| 3 | Service history..... | 5 |
| 4 | Global overview of the Service Group..... | 5 |
| 5 | Prerequisites | 6 |
| 5.1 | Business prerequisites | 6 |
| 5.2 | Technical prerequisites | 6 |
| 6 | Test, release and exploitation procedures | 6 |
| 7 | Description of the service operations | 7 |
| 7.1 | Operation SendDocument | 7 |
| 7.3 | Operation ConfirmDelivery - <i>deprecated</i> | 10 |
| 8 | Description of the common types..... | 13 |
| 8.1 | Simple Types | 13 |
| 8.2 | ComplexType DocumentRequest..... | 16 |
| 8.3 | ComplexType RegisteredMailItem..... | 17 |
| 8.4 | ComplexType RegisteredMail..... | Error! Bookmark not defined. |
| 8.5 | ComplexType File | 21 |
| 8.6 | ComplexType FileDescription | 19 |
| 8.7 | ComplexType FileItem | 22 |
| 8.8 | ComplexType Digest..... | 23 |
| 8.9 | ComplexType Enterprise | 23 |
| 8.10 | ComplexType DocumentResponse..... | 25 |
| 8.11 | ComplexType Receipt | 25 |
| 8.13 | ComplexType ReceiptItem | 26 |
| 8.14 | ComplexType RegisteredMailDescription | Error! Bookmark not defined. |
| 8.15 | ComplexType DeliveryRequest..... | Error! Bookmark not defined. |
| 8.16 | ComplexType DeliveryResponse | Error! Bookmark not defined. |
| 9 | Common error codes..... | 28 |
| 10 | Annex..... | 31 |

1 Goal of the service

1.1 Description of the service

The service RegisteredMail will allow the partners in the Social Security sector to send digital documents as registered mail. This electronic version of registered mail has the same juridical value as its paper equivalent.

1.2 Service consumers

The institutions of the Belgian Social Security that need to send documents as registered mail items to institutions, companies, professionals, citizens ...

2 Document management

2.1 Document history

| Version | Date | Author | Description of changes / remarks |
|---------|------------|----------|----------------------------------|
| 0.1 | 20/11/2013 | J. Buyse | First Draft |
| 0.2 | 25/11/2013 | J. Buyse | Modification after scope change |
| 2.0 | 12/02/2014 | J.Buyse | Final version |

2.2 Validation

| Reviewers | Name | Version Reviewed | Remarks |
|---------------------------------------|-------------|------------------|---------|
| Institution | F. Lastname | x.y | |
| Institution | F. Lastname | x.y | |
| Smals Project manager | F. Lastname | x.y | |
| Smals functional analyst or architect | F. Lastname | x.y | |
| IntegrationSOA | F. Lastname | x.y | |

2.3 Document references

| ID | Title | Version | Date | Author |
|----|--|---------|-------------------|--------------------------------------|
| 1. | SAD http://intranet.smals.be/mayafirdav/nodes/6467012/ePost%20%2D%20Software%20Architecture%20Document.doc | | <i>DD/MM/YYYY</i> | Pierre Herremans, Stéphane Flamme |
| 2. | Web Services Security X.509 Certificate Token Profile http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-x509TokenProfile.pdf | 1.1 | 01/02/2006 | OASIS |
| 3. | Web Services Security SAML Token Profile http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-errata-os-SAMLTOKENProfile.pdf | 1.1 | 01/11/2006 | OASIS |
| 4. | OASIS Standard WS-Trust http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.pdf | 1.3 | 19/03/2007 | OASIS |

2.4 Goal of the document

This document provides functional and technical information on calling the Service RegisteredMail, as provided by the KSZ. This information should allow (the IT department of) an organization to integrate and use the service.

In this service specification document, we explain the structure and content aspects of the possible service requests and replies. An example illustrates each of those messages. Also, the list of possible errors is included in this document.

Some technical and legal requirements must be satisfied in order to allow the integration of the services in client applications; this document was written in order to provide you with an overview of requirements which have to be met in order to integrate correctly with the Services offered by KSZ.

3 Service history

This chapter contains the list of changes to the service since the previous publication.

Remark: If only the minor(y) number has changed, the service is backward compatible with the previous version. Existing consumers with no need to use the new functionality do not have to change their implementation.

Previous version number: None

Previous release date: None

List of changes:

- 01/04/2014 v1.0: First version
- 12/02/2014 v2.0: Phase 2

4 Global overview of the Service Group

Not applicable

5 Prerequisites

5.1 Business prerequisites

Comment: This chapter contains information about the privacy rules and the preconditions governing the use of the Service.

5.2 Technical prerequisites

To call the WS you must sign your soap request with a Fedict X509 certificate.
This certificate identifies the institution and the application.

| X509 | Value | Comment |
|------|--|---|
| C | BE | - |
| O | Federal Government | - |
| OU | <i>Institution Name</i> | Example OU=National Social Security Office |
| OU | OU=urn:be:fgov:kbo-bce:organization:cbe-number:\${CBE} | Example for institution with CBE 12345676890 this OU must be OU=urn:be:fgov:kbo-bce:organization:cbe-number:1234567890 |
| CN | <i>Application identifier</i> | - |

Figure 1 - X.509 certificate: parameters

5.2.1 Security

Transport security

- For transport security one way SSL applies. The SSL connection will provide encryption to the data transported, as well as server authentication of the service provider.

Message security

- Message security using the OASIS WS-Security standards provides authentication of the service consumer and protects the message integrity end-to-end. See section 2.3 for links to the specifications.
- **Direct authentication (OASIS Standard X.509 Certificate Token Profile):** The consumer includes its X.509 certificate (wsse:BinarySecurityToken) and a timestamp (wsu:Timestamp) within the wsse:Security header. The consumer signs the entire SOAP body, timestamp and BST headers.

6 Test, release and exploitation procedures

/

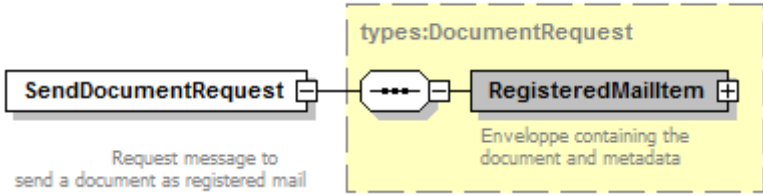
7 Description of the service operations

7.1 Operation SendDocument

7.1.1 Functional description

This operation allows a requester to send a document as registered mail.

7.1.2 Request message construction

| | |
|---------|---|
| diagram |  |
| source | <pre><xsd:element name="SendDocumentRequest" type="types:DocumentRequest"> <xsd:annotation> <xsd:documentation xml:lang="en"> Request message to send a document as registered mail </xsd:documentation> </xsd:annotation> </xsd:element></pre> |

Example:

```
<soapenv:Envelope xmlns:soapenv=http://schemas.xmlsoap.org/soap/envelope/
  xmlns:v1="http://socialsecurity.be/registeredmail/v1"
  xmlns:v11="http://socialsecurity.be/registeredmail/types/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header/>
  <soapenv:Body>
    <v1:SendDocumentRequest>
      <v11:RegisteredMailItem>
        <v11:Token>testToken_sha-256</v11:Token>
        <v11:Recipient xsi:type="v11:Enterprise">
          <v11:CompanyID>0861023666</v11:CompanyID>
          <v11:QualityCode>QUAL_FSC</v11:QualityCode>
        </v11:Recipient>
        <v11:Document>
          <v11:Title>RegMail_T1</v11:Title>
          <v11:Digest>
            <v11:DigestMethod>SHA-256</v11:DigestMethod>
            <v11:DigestValue>Y5/G6JbrPF3sXafd8qS/s51b5vdDVHgmZVX2H+5q0t8=</v11:DigestValue>
          </v11:Digest>
          <v11:DownloadFileName>FileName</v11:DownloadFileName>
          <!--Optional:-->
          <!--<v11:MIME>?</v11:MIME>-->
          <v11:Type>EDER501</v11:Type>
          <v11:Language>en</v11:Language>
          <v11:Content>cid:472982604608</v11:Content>
        </v11:Document>
      </v11:RegisteredMailItem>
    </v1:SendDocumentRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

7.1.3 Reply message interpretation

| | |
|---------|--|
| diagram | <p>The diagram illustrates the relationship between two message types. On the left, a box labeled 'SendDocumentResponse' is connected to a box labeled 'Receipt'. A dashed box labeled 'types:DocumentResponse' encloses the 'Receipt' box. Below 'SendDocumentResponse' is the text: 'Response message containing the receipt for the sent registered mail item'. Below 'Receipt' is the text: 'The receipt confirming that the document was sent'.</p> |
| source | <pre><xsd:element name="GetStatusRequest" type="types:StatusRequest"> <xsd:annotation> <xsd:documentation xml:lang="en"> Request message to obtain the status and optionally the signed slip for a certain registered mail item </xsd:documentation> </xsd:annotation> </xsd:element></pre> |

Example:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns5:SendDocumentResponse xmlns:ns2="http://socialsecurity.be/errors/v1"
      xmlns:ns3="http://socialsecurity.be/registeredmail/types/v1"
      xmlns:ns4="http://socialsecurity.be/errors/serviceprovider/v1"
      xmlns:ns5="http://socialsecurity.be/registeredmail/v1">
      <ns3:Receipt>
        <ns3:Ticket>1ZG0000036L3Z</ns3:Ticket>
        <ns3:Token>testToken_sha-256</ns3:Token>
        <ns3:Document>
          <ns3:Title>Receipt</ns3:Title>
          <ns3:Digest>
            <ns3:DigestMethod>SHA-256</ns3:DigestMethod>
            <ns3:DigestValue>8C3Q59LOG0Mx52pzCJXqjoBcFwpim30FaNlEMfvR6Ys=</ns3:DigestValue>
          </ns3:Digest>
          <ns3:DownloadFileName>Receipt.xml</ns3:DownloadFileName>
          <ns3:MIME>text/xml</ns3:MIME>
          <ns3:Type>RECEIPT</ns3:Type>
          <ns3:Language>en</ns3:Language>
          <ns3:Content>cid:18ee6a91-a70a-4cae-b0a3-3156129b7719@example.jaxws.sun.com</ns3:Content>
          <ns3:Signature>MIAGCSqGSIB3DQEHAqCAMIACAQExDzANBgglghkgBZQMEAgEFADCAgkqhkiG9w0BBwEA
            AKCAMIIDDCCafSgAwIBAgIJAK+Akl6e2pm8MA0GCSqGSIB3DQEBBQUAMCUxFjAUBgNVBAMTDVntYWxzIFJ
            vb3QgQ0ExCzAJBgNVBAYTAKJFMB4XDTEyMTAyOTE0MjAwNjAwNjAwNjAwNjAwNjAwNjAwNjAwNjAwNjAwNjAw
            MNU21hbHMgUm9vdCBDQTELMAGALUEBhMCQkUwggEiMA0GCSqGSIB3DQEBAQUAA4IBDwAwggEKAoIBAQBdBF
            OUUq12wCXbS7jqtuoQig6/F13Vm8nYFi2l6in7X3QJdH5HynOd4RsV2GP8sacvMkMZ4tM57yIq32KPK69Li
            rXDvHHp95je+l7JNh6zKFZQQH4BYHADvyXyM2qjAkDjJf/4kCqzmFLWM++HM+fb9693hCWzxtjmTwbC6pgV
            CYXlOpaTrI0HzYKw7L0Bj9wNyPT0qnxQ/H8TV129UYxp+20M1pjkMcnWI5aNoMwOlbgI6DbvgsGMFoAodEh
            1Bcf5Dnyo4Hhu0FgaQH+fIy5pH65D1EmsBl9EzVO7qrV8Gy5odCLsAlV4b6yl+hCQHRsY/SROR6bp3pDul
            kWwN+ClAgMBAAGjPzA9MB0GALUdDgQWBBQM7/gRiNIQ/p2A3vI3y0dh07FGyzAPBgNVHRMBAf8EBTADAQH/
            MASGALUdDwQEAwIBBJANBgkqhkiG9w0BAQUFAAOCAQEALW/pfvF1vq/P9j9Zfd94IAVjbiCTYUN4bHnjox/
            vLs7dVrBrennN+zu+IjJK3VSBaFFyZwLnChOLyQP1WVhBDHLq+/yBq3quFzHPLaDX9sSlqy9CfZ12Bo1/p
            4u+g/LCq0SojcfVxGlgfFrDkyE4fFSEL3jmmcpwrq5kFVM6WLUZfb5UBYd1NTqsLDsQW38lXXSGGafA09D
            X6dLwQqbA0h5JXSMsvlC452BEVGRJcon7r7DxSO4c6KSuWzX5hAE2dKqX0OvsNfv8alA12nv9KwhMNYfBu
            ytFiARqvFpocIQS/VyYtB6aUn2BV8aLE+Qq7pRPrYzYigbY+Q8cGOTCCA5MwggJ7oAMCAQICCCm1Ff84mT
            2RTANBgkqhkiG9w0BAQUFADALMRyWfAYDVQDEw1TbWFscysBSb290IENBMQswCQYDVQQGEWJCRTAeFw0xMz
            ExMDYwOTQ2MjhaFw0xODExMDYwOTQ2MjhaMIIGuMRMwEQYDVQQDDApLU1ogLSBCQ1NTMRswGQYDVQQKDBJGZ
            WRlcmF5IEEdvdmVybml1bnQxGDAWBgNVBASMD1NvY21hbCBTZW51cm10eTEtTMBEgALUECwwKs1NaIC0gQkNT
            UzE+MDwGALUECww1dXJuOmJlOmZnb3Y6a2JvLWljZTpvcmRhdml6YXRpb246Y2JlLW51bWJlcjoyNDQ2ND
            A2MzExCzAJBgNVBAYTAKJFMiIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEApnoU9sUwMCz64MXy7
            d0mZIDAI0tE2LLh/nyHMxvIMGsBS51PtbrxCGT2gnRZHMjIaoFplipd1/x8/TQfxnER4M+fuv0ELZ9RyGDz
```



```

1Dz6pSOCURKUDYaLclTdfavSoObBAC9tVmwIymqQa9UQMZRYw1QpeO/QVZ70f66Usa129PyNWSQArwb3IjY
uNOQvKvFj+mmdoPuAL3v0Y6kKmAmUXnJ7BJFsRbhtT6y24f03hEphMBrUVsKkn7j+3jdoh5alaJv4UZEk7
Ckmz2DaETHoiuYfDPRC9scjgx+9fgywoHfFSsgd3JqA2FlbPqq9aqI+DiOc2XK+MM+u0BQnWdNwIDAQABoz
wwOjAdBgNVHQ4EFgQUtmyX7/XfX46iCuGoex5MLCZw8bIwDAYDVR0TAQH/BAIwADALBGNVHQ8EBAMCBsAwD
QYJKoZIhvcNAQEFBQADggEBAEaCEl+PUnLRzkkLHAD+odZz4B5EGLUrFGvYkyJbmxy4wMzPuhEMCnzCrqsU
20efTu9QXkvN3hPwzQzrfNs0CYPQodX7dZaTkKfVRmB9jRUzgc9r81yoFReysXbjIL77xDMVDHyBy+lNAus
OQMF1ARb0+DF0GE1PFfe+7PYHa0lIAib7up8t04jY/wvMp3bgEIVeh0hlRDxPo0VWylzX5PRuP0/EhTfIhJL
iNA4dF1F2rS2RXjD83yK7dISHBCjdeH2XCjSxXdAU6FZmowrSnaAcMY8kNCynmBk+RT/Z5lzIyITN8MGCiX
QlAoHaW6zs6n0FTIRVEcGo80sIwiqRDukAAADGCBpgwggaUAqEBMDIwJTEWMBQGA1UEAxMNU21hbHMgUm9v
dCBDQTELMakGA1UEBhMCQkUCCQcm1Ff84mT2RTANBglghkgBZQMEAgEFAKCBkzAYBgkqhkiG9w0BCQMxCwY
JKoZIhvcNAQcBMC8GCSqGSIb3DQEJJBDEiBCDwLdDn0s4bQzHnanMIleqOgFwXCmKbc4Vo3Uqx+9HpiZGBBg
sqhkiG9w0BCRAclZ3MDUwMzAxMA0GCWCGSFA1AwQCAQUABCbtrWDLGd0v250nBNK8NK0Fv5e90k5Wrci7d
l9KEHqKHDANBgkqhkiG9w0BAQEFAASCAQBG2+mdkM7EuAhN5M5Qm745pPk1oQ2SrhaaPuzjh0781VyWgcu3q
X7IRK7S7E//gS55KXIJYf1m0BoDSY4CdNXgP6m572vzHjeIocEzGvv2G6VCvGSEnXZV+eVBwYr5Uew08gL
K/EPWADqI1CZErVXG81Yxaq+QvK8CgHu3d+IyDU+pWCYpfq17iRgaa/tQcED6YUCwAW/15uXRuW9KcAx0iK
JPuaC5d41lvMqzilzMYHgO+7lOML7+xSePE7ZMF6KdqaJV7NyIlgIhwJVx512Eh51jt144anNwzVSq5a1Ho
FqRaWrUMg7Vzhjs7ngso2a5Q4IdJdGJS+kQ45jW6y40YIEoTCCBJ0GCyqGSIB3DQEJEAIOYIEjDCCBIgG
CSqGSIb3DQEHAqCCBHkwwgR1AgEDMq8wDQYJYIZIAWUDBAIBBQAwggG0BgsqhkiG9w0BCRABBKCCAX0EggF
5MIIBDQIBAQQKkYBbBAGEWQoDATAxMA0GCWCGSFA1AwQCAQUABCcm2R6taUR7yp08Vw2u5E7jq+w/5FKWbx
hb7b2IB9qjhgIQUzNALwDTGBMyMDE0MDMzMTA2MzEwOS41OTFaMASAAgH0AgpEoD9l7E3AgCbeoIIBAKSB/
TCB+jE/MD0GA1UEAww2QmFucXVlIENhcnJlZm91ciBkZSBsYSBz6w1jdXJpdMOpIHNvY21hbGUzLSAwNDA2
Nzk4MDA2MqswCQYDVQGEwJCRTEyMDAGA1UECgwpQmFucXVlIENhcnJlZm91ciBkZSBsYSBz6w1jdXJpdMO
pIHNvY21hbGUzNTAzBgNVBASMLEF1ZG10IGludGVybmUgZXQgc8OpY3VyaXTDqSBkZSBsJ2luZm9ybWFOaW
9uMRlWEAYDVQHEw1CULVYRUxMRVMxKzApBgkqhkiG9w0BCQEWHHBhdHJpY2suYm9jaGFyZEBiY3NzLmZnb
3YuYmUxggLLMIICXwIBATB/MHAXCzAJBgNVBAYTAKJFMRwwGgYDVQKExNDZlXj0aXBvc3Qgcy5hLi9uLnYu
MUMwQQYDVQDEzpDZXJ0aXBvc3QgRS1UcnVzdCBTZWNvbmlRcnkGUHVhbG1maWVkaWVkbWVkbWVkbWVkbWVkbW
QZzXzb25zAgSBAAAAAEvl1PfdjANBglghkgBZQMEAgEFAKCCAR0wGgYJKoZIhvcNAQkDMQ0GCyqGSIB3DQ
EJEAEMC8GCSqGSIb3DQEJJBDEiBCDU4WcgKDNnZjZJL65wDiQzUhVcOqMT2DqyXEWL8njmZTCBzQYLKozIh
vcNAQKQAgwXgb0wgbowgbcwZwEFDDX4K6QE19m3SLNduJb9ytPN1YSMIGDMHskcjbWmqswCQYDVQGEwJCR
TECMBoGA1UEChMTQ2Vydg1wb3N0IHMuYS4vbi52LjFDMEEGA1UEAxM6Q2Vydg1wb3N0IEUUVHJlc3QgU2V
jb25kYXJ5Xj5lFF1YwXpZml1ZCBDDQSBmb3IgtGVnYWwGUUVyc29ucwILAQAAAAABL5d33YwFgQUw17+Vathmc
FmB/HHE7gMdrLjFokwDQYJKoZIhvcNAQELBQAEEggEApqTyWSgDBRlHT+plTqsBKEWaoW97Qy8xscsXDittv
wZsLkuYE5HY2jYvbuCgIcqsWsbANI8nhi0yn7mqm09x09L5vPF2T7yF+oJNT07EuxZMNPqmHUvNrDYuHoLz
7TYM+rdhatFN7KECKdvjIj5TQTy2NNOwDNih0fyrzDYF3M4hFU2jKqB3eIF/azXHW9FOF0f0f3FbJasF2Ga
NOs+a64DWgIELS90CU25mKs+q8Z2pxQvMRgoujPuy33LeKc6VDUBiSsHLtIcQ7gL+231tj4PV4KBSBwFxfM
V95SvCwGqcoKRIJTYAF/yCMgKoJLwsB/mFUY0eY5zyV7BbvsYB2QAAAAAAA==</ns3:Signature>
</ns3:Document>
</ns3:Receipt>
</ns5:SendDocumentResponse>
</S:Body>
</S:Envelope>

```

7.1.4 Error codes

The error codes listed here define possible business errors whilst processing the request. These error codes indicate a problem in the content of the request, including format errors.

| Error code | Description |
|-----------------|---|
| EPOST_T01 | Internal Server Error |
| EPOST_T02 | Unauthorized |
| EPOST_T03 | Cannot capture soap request |
| EPOST_T04 | Unsupported principal type |
| EPOST_B01 | Business Error |
| | |
| GENERIC_XML_T01 | Cannot generate GenericXmlDocument from xml |
| GENERIC_XML_T02 | Cannot build Request from GenericXmlDocument |
| GENERIC_XML_T03 | Cannot generate xml from GenericXmlDocument |
| GENERIC_XML_B01 | GenericXml does not contain all required data |
| | |

| | |
|-----------|--|
| REPO_T01 | Repository error |
| REPO_T02 | Repository validation error |
| | |
| FILE_T01 | No such digest method algorithm |
| FILE_T02 | Cannot persist file content |
| FILE_T03 | Cannot read file content |
| FILE_B01 | Digest doesn't match document. |
| FILE_B02 | Digest method not accepted (only SHA-1 |
| | |
| SEPIA_T01 | Sepia system error |
| SEPIA_T02 | Could not sign receipt |
| | |
| TNG_T01 | TNG unreachable - system error |
| TNG_T02 | Cannot create ticket |
| | |
| JMS_T01 | Cannot send JMS message |
| JMS_T02 | Problem getting JMS queue |
| JMS_T03 | Problem getting JMS connection factory |
| JMS_T04 | Unable to set JMS message properties |
| JMS_T05 | Unable to get JMS message properties |
| JMS_T06 | Cannot read JMS message |
| JMS_T07 | Cannot build JMS message |
| JMS_B01 | JMS message properties are missing |

7.3 Operation GetStatus

7.3.1 Functional description

This function allows the sender to obtain the status of his registered mail item..

7.3.2 Request message construction

| | |
|---------|--|
| diagram | <p>Request message to obtain the status and optionally the signed slip for a certain registered mail item</p> <p>The ticket number that references a registered mail item internally within RegisteredMail</p> |
| source | <pre><xsd:element name="GetStatusRequest" type="types:StatusRequest"> <xsd:annotation> <xsd:documentation xml:lang="en"> Request message to obtain the status and optionally the signed slip for a certain registered mail item </xsd:documentation> </xsd:annotation></pre> |

| | |
|--|----------------|
| | </xsd:element> |
|--|----------------|

Example:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v1="http://socialsecurity.be/registeredmail/v1"
xmlns:v11="http://socialsecurity.be/registeredmail/types/v1">
  <soapenv:Header/>
  <soapenv:Body>
    <v1:GetStatusRequest>
      <v11:Ticket>001000KG4GECZ</v11:Ticket>
    </v1:GetStatusRequest >
  </soapenv:Body>
</soapenv:Envelope>

```

7.3.3 Reply message interpretation

| | |
|---------|---|
| diagram | |
| source | <pre> <xsd:element name="GetStatusResponse" type="types:StatusResponse"> <xsd:annotation> <xsd:documentation xml:lang="en"> Response message containing the status for the requested registered mail item and optionally the signed slip </xsd:documentation> </xsd:annotation> </xsd:element> </pre> |

Example:

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns5:GetStatusResponse xmlns:ns2="http://socialsecurity.be/errors/v1"
xmlns:ns3="http://socialsecurity.be/registeredmail/types/v1"
xmlns:ns4="http://socialsecurity.be/errors/serviceprovider/v1"
xmlns:ns5="http://socialsecurity.be/registeredmail/v1">
      <ns3:Ticket>1ZG0000036L3Z</ns3:Ticket>
      <ns3:Token>testToken_sha-256</ns3:Token>
      <ns3:Status>PENDING</ns3:Status>
    </ns5:GetStatusResponse>
  </S:Body>
</S:Envelope>

```

7.3.4 Error codes

The error codes listed here define possible business errors whilst processing the request. These error codes indicate a problem in the content of the request, including format errors.

| Error code | Description |
|------------|---|
| EPOST_T01 | Internal Server Error |
| EPOST_T02 | Unauthorized |
| EPOST_B01 | Business Error |
| | |
| REPO_T01 | Repository error |
| REPO_T02 | Repository validation error |
| REPO_B01 | Ticket provided does not match Registered Mail Item |

8 Description of the common types

8.1 Simple Types

8.1.1 simpleType CBE

| | |
|--------|--|
| source | <pre><xsd:simpleType name="CBE"> <xsd:annotation> <xsd:documentation xml:lang="en">The KBO number of the enterprise</xsd:documentation> </xsd:annotation> <xsd:restriction base="xsd:string"> <xsd:pattern value="[0-9]?[0-9]{9}"/> </xsd:restriction> </xsd:simpleType></pre> |
|--------|--|

8.1.2 simpleType DigestMethod

| | |
|--------|---|
| source | <pre><xsd:simpleType name="DigestMethod"> <xsd:annotation> <xsd:documentation xml:lang="en">Digest Algorithm (SHA-1, SHA-256, SHA-512). </xsd:documentation> </xsd:annotation> <xsd:restriction base="xsd:string"> <xsd:enumeration value="SHA-1"/> <xsd:enumeration value="SHA-256"/> <xsd:enumeration value="SHA-512"/> </xsd:restriction> </xsd:simpleType></pre> |
|--------|---|

8.1.3 simpleType DigestValue

| | |
|--------|---|
| source | <pre><xsd:simpleType name="DigestValue"> <xsd:annotation> <xsd:documentation xml:lang="en">Value of digest</xsd:documentation> </xsd:annotation> <xsd:restriction base="xsd:base64Binary"> <xsd:minLength value="1"/> <xsd:maxLength value="150"/> </xsd:restriction> </xsd:simpleType></pre> |
|--------|---|

8.1.4 simpleType FileName

| | |
|--------|--|
| source | <pre><xsd:simpleType name="FileName"> <xsd:annotation> <xsd:documentation xml:lang="en">Name to download the file.</xsd:documentation> </xsd:annotation> <xsd:restriction base="xsd:string"> <xsd:minLength value="1"/> <xsd:maxLength value="80"/> </xsd:restriction> </xsd:simpleType></pre> |
|--------|--|

8.1.5 simpleType Language

| | |
|--------|---|
| source | <pre><xsd:simpleType name="Language"> <xsd:annotation> <xsd:documentation>Language, restricted to the values FR for french, NL for dutch, DE for German </xsd:documentation> </xsd:annotation> <xsd:restriction base="xsd:string"> <xsd:enumeration value="FR"> <xsd:annotation> <xsd:documentation>french</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="NL"> <xsd:annotation> <xsd:documentation>Dutch</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="DE"> <xsd:annotation> <xsd:documentation>German</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="EN"> <xsd:annotation> <xsd:documentation>English</xsd:documentation> </xsd:annotation> </xsd:enumeration> </xsd:restriction> </xsd:simpleType></pre> |
|--------|---|

8.1.6 simpleType MIME

| | |
|--------|---|
| source | <pre><xsd:simpleType name="MIME"> <xsd:annotation> <xsd:documentation xml:lang="en">MIME supported by e-Box. </xsd:documentation> </xsd:annotation> <xsd:restriction base="xsd:string"> <xsd:enumeration value="application/pdf"/> <xsd:enumeration value="application/zip"/> <xsd:enumeration value="text/csv"/> <xsd:enumeration value="text/html"/> <xsd:enumeration value="text/plain"/> <xsd:enumeration value="text/xml"/> <xsd:enumeration value="application/vnd.oasis.opendocument.text"/> <xsd:enumeration value="application/vnd.oasis.opendocument.spreadsheet"/> <xsd:enumeration value="application/vnd.oasis.opendocument.presentation"/> <xsd:enumeration value="application/vnd.oasis.opendocument.graphics"/> <xsd:enumeration value="application/vnd.ms-excel"/> <xsd:enumeration value="application/vnd.openxmlformats-officedocument.spreadsheetml.sheet"/> <xsd:enumeration value="application/vnd.ms-powerpoint"/> <xsd:enumeration value="application/msword"/> <xsd:enumeration value="application/vnd.openxmlformats-officedocument.wordprocessingml.document"/> </xsd:restriction> </xsd:simpleType></pre> |
|--------|---|

8.1.7 simpleType QualityCode

| | |
|--------|---|
| source | <pre><xsd:simpleType name="QualityCode"> <xsd:annotation> <xsd:documentation xml:lang="en">Quality code from the userManagement of social security. Ex : QUAL_EMP_NOSS, QUAL_SSC, QUAL_SP_IND, ...</xsd:documentation> </xsd:annotation> <xsd:restriction base="xsd:string"> <xsd:minLength value="1"/> <xsd:maxLength value="125"/> </xsd:restriction> </xsd:simpleType></pre> |
|--------|---|

8.1.8 simpleType Sender

| | |
|--------|---|
| source | <pre><xsd:simpleType name="Sender"> <xsd:annotation> <xsd:documentation xml:lang="en">The KBO of the sender certificate</xsd:documentation> </xsd:annotation> <xsd:restriction base="xsd:string"> <xsd:maxLength value="600"/> </xsd:restriction> </xsd:simpleType></pre> |
|--------|---|

8.1.9 simpleType ShortString

| | |
|--------|---|
| source | <pre><xsd:simpleType name="ShortString"> <xsd:annotation> <xsd:documentation xml:lang="en">Short string limited to 25 chars</xsd:documentation> </xsd:annotation> <xsd:restriction base="xsd:string"> <xsd:minLength value="1"/> <xsd:maxLength value="25"/> </xsd:restriction> </xsd:simpleType></pre> |
|--------|---|

8.1.10 simpleType Status

| | |
|--------|---|
| source | <pre><xsd:simpleType name="Status"> <xsd:annotation> <xsd:documentation xml:lang="en">The status of the registered mail item</xsd:documentation> </xsd:annotation> <xsd:restriction base="xsd:string"> <xsd:enumeration value="PENDING"/> <xsd:enumeration value="SENT"/> <xsd:enumeration value="ERROR"/> </xsd:restriction> </xsd:simpleType></pre> |
|--------|---|

8.1.11 simpleType Ticket

| | |
|--------|--|
| source | <pre><xsd:simpleType name="Ticket"> <xsd:annotation> <xsd:documentation xml:lang="en">A ticket number as provided by the</pre> |
|--------|--|

| | |
|--|--|
| | <pre>TNG</xsd:documentation> </xsd:annotation> <xsd:restriction base="xsd:string"> <xsd:length value="13"/> <xsd:pattern value="[0-9A-HJ-NP-Z]{13}"/> </xsd:restriction> </xsd:simpleType></pre> |
|--|--|

8.1.12 simpleType Title

| | |
|--------|---|
| source | <pre><xsd:simpleType name="Title"> <xsd:annotation> <xsd:documentation xml:lang="en">Same as subject of mail</xsd:documentation> </xsd:annotation> <xsd:restriction base="xsd:string"> <xsd:minLength value="1"/> <xsd:maxLength value="400"/> </xsd:restriction> </xsd:simpleType></pre> |
|--------|---|

8.1.13 simpleType Token

| | |
|--------|--|
| source | <pre><xsd:simpleType name="Token"> <xsd:annotation> <xsd:documentation xml:lang="en">A generic token</xsd:documentation> </xsd:annotation> <xsd:restriction base="xsd:string"> <xsd:length value="50"/> </xsd:restriction> </xsd:simpleType></pre> |
|--------|--|

8.1.14 simpleType swaRef

| | |
|--------|---|
| source | <pre><xsd:simpleType name="swaRef"> <xsd:restriction base="xs:anyURI"/> </xsd:simpleType></pre> |
|--------|---|

8.1.15 simpleType Type

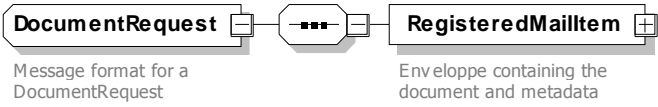
| | |
|--------|--|
| source | <pre><xsd:simpleType name="Type"> <xsd:annotation> <xsd:documentation xml:lang="en">The document type.</xsd:documentation> </xsd:annotation> <xsd:restriction base="xsd:string"> <xsd:minLength value="1"/> <xsd:maxLength value="80"/> </xsd:restriction> </xsd:simpleType></pre> |
|--------|--|

8.2 ComplexType DocumentRequest

8.2.1 Functional description

The DocumentRequest type contains the information necessary to make a sendDocument request. It only has one child item; the RegisteredMailItem that must be sent.

8.2.2 Example

| | |
|---------|---|
| diagram |  <p>Message format for a DocumentRequest</p> <p>Envelope containing the document and metadata</p> |
| source | <pre> <xsd:complexType name="DocumentRequest"> <xsd:annotation> <xsd:documentation xml:lang="en">Message format for a DocumentRequest</xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="RegisteredMailItem" type="RegisteredMailItem"> <xsd:annotation> <xsd:documentation xml:lang="en">Envelope containing the document and metadata</xsd:documentation> </xsd:annotation> </xsd:element> </xsd:sequence> </xsd:complexType> </pre> |

8.3 ComplexType RegisteredMailItem

8.3.1 Functional description

The RegisteredMailItem type identifies a registered mail item. It has three child items:

- Token: The token that was included in the sendDocument request which serves as the identifier for the user
- Document: The document to send as a registered mail item (*cf* 8.4 File)
- Recipient: The recipient of the document (*cf* 8.6 Enterprise)
- Meta: Metadata linked to the document

8.3.2 Example

| | |
|----------------|--|
| <p>diagram</p> | |
| <p>source</p> | <pre> <xsd:complexType name="RegisteredMailItem"> <xsd:annotation> <xsd:documentation xml:lang="en">Envelope containing the document and metadata</xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="Token" type="Token"> <xsd:annotation> <xsd:documentation xml:lang="en">A token, permitting the sender to identify the document</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="Recipient" type="Entity"> <xsd:annotation> <xsd:documentation xml:lang="en">The recipient of the document</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="Document" type="File"> <xsd:annotation> <xsd:documentation xml:lang="en">The document to be sent as registered mail</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="Meta" type="Meta" minOccurs="0" maxOccurs="100"> <xsd:annotation> <xsd:documentation xml:lang="en">Meta data linked to registered mail</xsd:documentation> </xsd:annotation> </xsd:element> </xsd:sequence> </xsd:complexType> </pre> |

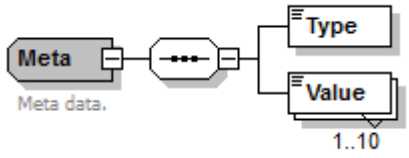
8.4 ComplexType Meta

8.4.1 Functional description

The meta type permits to send further specific information about the document under the form of type-value pairs

- Type: the type of the meta data
- Value: the values of the metadata (up to 10)

8.4.2 Example

| | |
|---------|--|
| diagram |  |
| source | <pre> <xsd:complexType name="Meta"> <xsd:annotation> <xsd:documentation xml:lang="en">Meta data. </xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="Type" type="ShortString"/> <xsd:element name="Value" type="ShortString" maxOccurs="10"/> </xsd:sequence> </xsd:complexType> </pre> |

8.5 ComplexType FileDescription

8.5.1 Functional description

The FileDescription type contains all the information of a document without containing the actual contents

- Title: Title of the sent document.
- Digest: In order to check the integrity of the message content, the sender must add the hash of the document (DigestValue). The WS will then verify that the provided digest is the same as the document digest computed according to the provided method (SHA-1; SHA-256 or SHA-512).
- DownloadFileName: name used to open or save the document.
- MIME [optional]: MIME type of the document
- Type: The document type
- Language: The language of the document

8.5.2 Example

| | |
|----------------|---|
| <p>diagram</p> | <p>A file must have a title, a content and a "name" to download file, and save it on disk. Mime is optional</p> |
| <p>source</p> | <pre> <xsd:complexType name="FileDescription"> <xsd:annotation> <xsd:documentation xml:lang="en">A file must have a title, a content and a "name" to download file, and save it on disk. Mime is optional </xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="Title" type="Title"/> <xsd:element name="Digest" type="Digest"/> <xsd:element name="DownloadFileName" type="FileName"/> <xsd:element name="MIME" type="MIME" minOccurs="0"/> <xsd:element name="Type" type="Type"/> <xsd:element name="Language" type="Language"/> </xsd:sequence> </xsd:complexType> </pre> |

8.6 ComplexType File

8.6.1 Functional description

The File complexType extends the FileDescription complexType. In addition, it contains a content element, which is a URI that references the document in attachment (using swaRef standard).

- Content: Contents of the document in an attachment

8.6.2 Example

| | |
|----------------|---|
| <p>diagram</p> | <p>The diagram illustrates the structure of the File complexType. It is an extension of the FileDescription complexType. FileDescription contains the following elements: Title, Digest, DownloadFileName, MIME (optional), Type, and Language. File extends FileDescription and includes a Content element. Annotations provide additional information: 'A file must have a title, a content and a "name" to download file, and save it on disk. Mime is optional' for File, and 'Reference to the contents in the attachment' for Content.</p> |
| <p>source</p> | <pre> <xsd:complexType name="File"> <xsd:annotation> <xsd:documentation xml:lang="en">A file must have a title, a content and a "name" to download file, and save it on disk. Mime is optional </xsd:documentation> </xsd:annotation> <xsd:complexContent> <xsd:extension base="FileDescription"> <xsd:sequence> <xsd:element name="Content" type="wsi:swaRef"> <xsd:annotation> <xsd:documentation xml:lang="en">Reference to the contents in the attachment</xsd:documentation> </xsd:annotation> </xsd:element> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre> |

8.7 ComplexType SignedFile

8.7.1 Functional description

The SignedFile complexType extends the File complexType. In addition, it contains a signature element, which is the base64 encoded signature produced by Sepia for the document in Content

8.7.2 Example

| | |
|----------------|--|
| <p>diagram</p> | <p>The diagram illustrates the structure of the SignedFile complexType. It is an extension of the File complexType. The File complexType contains the following elements: Title, Digest, DownloadFileName, MIME (indicated by a dashed box), Type, and Language. The SignedFile complexType extends File and adds a Signature element. The Content element is also shown as part of the File complexType structure, with a note: "Reference to the contents in the attachment". The Signature element is described as: "The signature generated by SepiaDocumentSigner for the File content".</p> |
| <p>source</p> | <pre> <xsd:complexType name="SignedFile"> <xsd:annotation> <xsd:documentation xml:lang="en">Extension of file containing a signature by SepiaDocumentSigner</xsd:documentation> </xsd:annotation> <xsd:complexContent> <xsd:extension base="File"> <xsd:sequence> <xsd:element name="Signature" type="xsd:base64Binary"> <xsd:annotation> <xsd:documentation xml:lang="en">The signature generated by SepiaDocumentSigner for the File content</xsd:documentation> </xsd:annotation> </xsd:element> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre> |

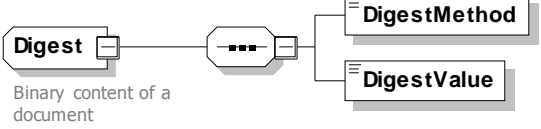
8.8 ComplexType Digest

8.8.1 Functional description

The Digest type contains all necessary information to verify the digest of a document. It has two child items:

- DigestMethod: The algorithm that was used to calculate the DigestValue:
- DigestValue: The digest calculated with the DigestMethod

8.8.2 Example


| | |
|---------|---|
| diagram |  <p>Binary content of a document</p> |
| source | <pre><xsd:complexType name="Digest"> <xsd:annotation> <xsd:documentation xml:lang="en">Binary content of a document</xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="DigestMethod" type="DigestMethod"/> <xsd:element name="DigestValue" type="DigestValue"/> </xsd:sequence> </xsd:complexType></pre> |

8.9 ComplexType Entity

8.9.1 Functional description

The Entity type defines a generic recipient

8.9.2 Example

| | |
|---------|---|
| diagram |  <p>An abstract entity. This can be a person, enterprise, ...</p> |
| source | <pre><xsd:complexType name="Entity" abstract="true"> <xsd:annotation> <xsd:documentation xml:lang="en">An abstract entity. This can be a person, enterprise, ...</xsd:documentation> </xsd:annotation> </xsd:complexType></pre> |

8.10 ComplexType Enterprise

8.10.1 Functional description

The Enterprise type defines a recipient who is identified as an enterprise. It has two child items:

- CompanyID: The CBE number of the enterprise
- QualityCode: The quality under which the destinationary

8.10.2 Example

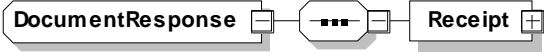
| | |
|---------|---|
| diagram | |
| source | <pre> <xsd:complexType name="Enterprise"> <xsd:annotation> <xsd:documentation xml:lang="en">An enterprise in UMan</xsd:documentation> </xsd:annotation> <xsd:complexContent> <xsd:extension base="Entity"> <xsd:sequence> <xsd:element name="CompanyID" type="CBE"> <xsd:annotation> <xsd:documentation xml:lang="en">The KBO number of the enterprise</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="QualityCode" type="QualityCode"> <xsd:annotation> <xsd:documentation xml:lang="en">The quality under which the entity is known</xsd:documentation> </xsd:annotation> </xsd:element> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre> |

8.11 ComplexType DocumentResponse

8.11.1 Functional description

The DocumentResponse type contains the response message the sender receives after making a sendDocument request. It only has one child item, the receipt for the registered mail item (*cfr. 8.8 Receipt*).

8.11.2 Example

| | |
|---------|---|
| diagram |  <p>Response message that should be either an echo of the token with a ticket or an error</p> <p>The receipt confirming that the document was sent</p> |
| source | <pre><xsd:complexType name="DocumentResponse"> <xsd:annotation> <xsd:documentation xml:lang="en">Response message that should be either an echo of the token with a ticket or an error</xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="Receipt" type="Receipt"> <xsd:annotation> <xsd:documentation xml:lang="en">The receipt confirming that the document was sent</xsd:documentation> </xsd:annotation> </xsd:element> </xsd:sequence> </xsd:complexType></pre> |

8.12 ComplexType Receipt

8.12.1 Functional description

The Receipt type contains the receipt the sender gets after calling the sendDocument operation. It contains three child items:

- Token: The token that was included in the sendDocument request which serves as the identifier for the user
- Ticket: The ticket number of this request which identifies this registered mail item and can serve for future reference (e.g. to make a status request)
- Document: The receipt which serves as the proof for the sender that he in fact has sent this registered mail item (*cfr. 8.4 File*)

8.12.2 Example

| | |
|---------|---|
| diagram | |
| source | <pre> <xsd:complexType name="Receipt"> <xsd:annotation> <xsd:documentation xml:lang="en">The receipt confirming that the document was sent</xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="Ticket" type="Ticket"> <xsd:annotation> <xsd:documentation xml:lang="en">The ticket number that will reference the registered mail item internally within RegisteredMail</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="Token" type="Token"> <xsd:annotation> <xsd:documentation xml:lang="en">The token sent by the user in the send request, which serves as a user provided identifier</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="Document" type="SignedFile"/> </xsd:sequence> </xsd:complexType> </pre> |

8.14 ComplexType ReceiptContent

8.14.1 Functional description

The ReceiptContent type contains the information contained in the content element of the Receipt type.

- Ticket: The ticket number of this request which identifies this registered mail item and can serve for future reference (e.g. to make a status request)
- Token: The token that was included in the sendDocument request which serves as the identifier for the user
- Recipient: The recipient of the registered mail item
- Sender: The subject of the x509 certificate of the sender
- Document: The FileDescription of the RegisteredMailItem that was sent

8.14.2 Example

| | |
|----------------|--|
| <p>diagram</p> | <pre> classDiagram class ReceiptContent { Ticket Token Recipient Sender Document } class Ticket { "The ticket number that will reference the registered mail item internally within RegisteredMail" } class Token { "The token sent by the user in the send request, which serves as a user provided identifier" } class Recipient { "The recipient of the document" } class Sender { "The sender of the document" } class Document { "FileDescription of the document that was sent as registered mail" } ReceiptContent --> Ticket ReceiptContent --> Token ReceiptContent --> Recipient ReceiptContent --> Sender ReceiptContent --> Document </pre> |
| <p>source</p> | <pre> <xsd:complexType name="ReceiptContent"> <xsd:annotation> <xsd:documentation xml:lang="en">The contents of the SignedFile in the receipt</xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="Ticket" type="Ticket"> <xsd:annotation> <xsd:documentation xml:lang="en">The ticket number that will reference the registered mail item internally within RegisteredMail</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="Token" type="Token"> <xsd:annotation> <xsd:documentation xml:lang="en">The token sent by the user in the send request, which serves as a user provided identifier</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="Recipient" type="Entity"> <xsd:annotation> <xsd:documentation xml:lang="en">The recipient of the document</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="Sender" type="Sender"> <xsd:annotation> <xsd:documentation xml:lang="en">The sender of the document</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="Document" type="FileDescription"> <xsd:annotation> <xsd:documentation xml:lang="en">FileDescription of the document that was </pre> |

| | |
|--|--|
| | <pre> sent as registered mail</xsd:documentation> </xsd:annotation> </xsd:element> </xsd:sequence> </xsd:complexType> </pre> |
|--|--|

8.15 ComplexType StatusRequest

8.15.1 Functional description

The StatusRequest type contains the information necessary to make a getStatus request. It only has one child item; the Ticket of the transaction for which the requester wishes to know the status.

8.15.2 Example

| | |
|---------|--|
| diagram | <pre> classDiagram class StatusRequest { Ticket } class Ticket StatusRequest "1" -- "1" Ticket </pre> <p>The diagram shows a class StatusRequest containing a class Ticket. The StatusRequest class is described as "Request message containing a ticket number for which the sender wants the proof of dispatch". The Ticket class is described as "The ticket number that references a registered mail item internally within RegisteredMail".</p> |
| source | <pre> <xsd:complexType name="StatusRequest"> <xsd:annotation> <xsd:documentation xml:lang="en">Request message containing a ticket number for which the sender wants the proof of dispatch</xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="Ticket" type="Ticket"> <xsd:annotation> <xsd:documentation xml:lang="en">The ticket number that references a registered mail item internally within RegisteredMail</xsd:documentation> </xsd:annotation> </xsd:element> </xsd:sequence> </xsd:complexType> </pre> |

8.16 ComplexType StatusResponse

8.16.1 Functional description

The StatusResponse type contains the response message the sender receives after making a getStatus request. It only has three child items:

- Ticket: The ticket number of this request which identifies this registered mail item and can serve for future reference (e.g. to make a status request)
- Token: The token that was included in the sendDocument request which serves as the identifier for the user
- Status: The status of the document
- Document: The proof of dispatch when the registered mail item was already dispatched

8.16.2 Example

| | |
|----------------|---|
| <p>diagram</p> | <pre> classDiagram class StatusResponse { Ticket Token Status ProofOfDispatch } class Ticket { <!-- Description: The ticket number that references a registered mail item internally within RegisteredMail --> } class Token { <!-- Description: The token sent by the user in the send request, which serves as a user provided identifier --> } class Status { <!-- Description: The status of the registered mail item --> } class ProofOfDispatch { <!-- Description: The proof of dispatch --> } StatusResponse -- Ticket StatusResponse -- Token StatusResponse -- Status StatusResponse .. ProofOfDispatch </pre> |
| <p>source</p> | <pre> <xsd:complexType name="StatusResponse"> <xsd:annotation> <xsd:documentation xml:lang="en">Response message containing the document status and the rproof of dispatch if available</xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="Ticket" type="Ticket"> <xsd:annotation> <xsd:documentation xml:lang="en">The ticket number that references a registered mail item internally within RegisteredMail</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="Token" type="Token"> <xsd:annotation> <xsd:documentation xml:lang="en">The token sent by the user in the send request, which serves as a user provided identifier</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="Status" type="Status"> <xsd:annotation> <xsd:documentation xml:lang="en">The status of the registered mail item</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="ProofOfDispatch" type="SignedFile" minOccurs="0"> <xsd:annotation> <xsd:documentation xml:lang="en">The proof of dispatch</xsd:documentation> </xsd:annotation> </xsd:element> </xsd:sequence> </xsd:complexType> </pre> |

9 Common error codes

System Error Codes:

These error codes indicate a problem in the request sent or a system error.

| Code | Description | Cause | Explication |
|-----------|---|----------|--|
| SOA-00001 | Service error | | This is the default error sent to the consumer in case no more details are known. |
| SOA-01001 | Service call not authenticated | Consumer | From the security information provided, either the consumer could not be identified or the credentials provided are not correct. |
| SOA-01002 | Service call not authorized | Consumer | The consumer is identified and authenticated, but is not allowed to call the given service. |
| SOA-02001 | Service not available. Please contact service desk. | Provider | An unexpected error has occurred. Retries will not work. Service desk may help with root cause analysis. |
| SOA-02002 | Service temporarily not available. Please try later. | Provider | An unexpected error has occurred. Retries should work. If the problem persists service desk may help. |
| SOA-03001 | Malformed message | Consumer | This is the default error for content related errors in case no more details are known. |
| SOA-03002 | Message must be SOAP | Consumer | Message does not respect the SOAP standard. |
| SOA-03003 | Message must contain SOAP body | Consumer | Message respects the SOAP standard, but body is missing. |
| SOA-03004 | WS-I compliance failure | Consumer | Message does not respect the WS-I standard. |
| SOA-03005 | WSDL compliance failure | Consumer | Message is not compliant with WSDL. |
| SOA-03006 | XSD compliance failure | Consumer | Message is not compliant with XSD. |
| SOA-03007 | Message content validation failure | Consumer | From the message content (conform XSD): extended checks on the element format failed or cross-checks between fields failed. |

Business Error Codes:

These error codes are defined by the application.

| Error code | Description |
|-----------------|---|
| EPOST_T01 | Internal Server Error |
| EPOST_T02 | Unauthorized |
| EPOST_T03 | Cannot capture soap request |
| EPOST_T04 | Unsupported principal type |
| EPOST_B01 | Business Error |
| | |
| GENERIC_XML_T01 | Cannot generate GenericXmlDocument from xml |
| GENERIC_XML_T02 | Cannot build Request from GenericXmlDocument |
| GENERIC_XML_T03 | Cannot generate xml from GenericXmlDocument |
| GENERIC_XML_B01 | GenericXml does not contain all required data |
| | |
| REPO_T01 | Repository error |
| REPO_T02 | Repository validation error |
| REPO_B01 | Ticket provided does not match Registered Mail Item |
| | |
| FILE_T01 | No such digest method algorithm |

| | |
|-----------|--|
| FILE_T02 | Cannot persist file content |
| FILE_T03 | Cannot read file content |
| FILE_B01 | Digest doesn't match document. |
| FILE_B02 | Digest method not accepted (only SHA-1 |
| | |
| SEPIA_T01 | Sepia system error |
| SEPIA_T02 | Could not sign receipt |
| | |
| TNG_T01 | TNG unreachable - system error |
| TNG_T02 | Cannot create ticket |
| | |
| JMS_T01 | Cannot send JMS message |
| JMS_T02 | Problem getting JMS queue |
| JMS_T03 | Problem getting JMS connection factory |
| JMS_T04 | Unable to set JMS message properties |
| JMS_T05 | Unable to get JMS message properties |
| JMS_T06 | Cannot read JMS message |
| JMS_T07 | Cannot build JMS message |
| JMS_B01 | JMS message properties are missing |

10 Annex

Comment: Libraries that could/should be used in combination with the use of this service, should be documented here