**Ebox Citizen**

**Integration specification.**

| **Version** | **Status** | **Date** | **Authors** | **Modifications** |
| --- | --- | --- | --- | --- |
| 0.1 | Draft |  | Adrien Clerbaut |  |
|  |  |  |  |  |

**Table of Contents**

1 Service Rest 3

1.1 Security 3

1.2 URL 3

1.3 Resources: *messages* 3

List of messages 3

Count messages in eBox 8

1.4 Resources: *senders* 9

1.5 WADL 12

2 Link Partner account with e-Box 13

2.1 Request 14

Post url for mysocialsecurity service 15

2.2 Response 15

# Service Rest

## Security

All the REST request must be send over two-way ssl.

* The certificate for 2 way ssl will be provided by SmalS

## URL

The context-root url are :

* **Validation** : <https://rest-acpt.socialsecurity.be/REST/Ebox/Citizen/v1>
* **Production** : <https://rest.socialsecurity.be/REST/Ebox/Citizen/v1>

## Resources: *messages*

## List of messages

|  |  |
| --- | --- |
| **URL** | /messages/{userId}/list |
| **Method** | GET |
| **Description** | Return a list with the user’s messages  *Ex :*  */messages /userA/list*  Return the last *100* messages in userA’s eBox.  */messages/userB/ /list?status=unread&pageSize=25*  Return the last *25* messages unread in userB’s eBox. |

|  |  |
| --- | --- |
| **URL params** | **Required** :   * userId   + Identify the user   + String   **Optional** : (QUERY PARAM)   * status   + status of message   + String enum : {“**all**”, "read", "unread"}   + Default value = "all" * pageNumber   + page asked   + integer   + default value = 1   + min value = 1 * pageSize   + max element/page   + integer   + default value = 100   + min value = 1   + max value = 1000 * sender   + select only message sended by the sender identify by his cbe.   + String   + min length 10   + max length 10 * orderBy   + String enum : {"**receiptDateDesc**”, "receiptDateAsc”,”titleDsc”,”titleAsc”}   + Default value = "receiptDateDesc” |

|  |  |
| --- | --- |
| **Success Response** | **Code** : 200  **Content** :  {  "type": "object",  "required": false,  "properties": {  "**messages**": {  "type": "array",  "required": false,  "items": {  "type": "object",  "required": true,  "properties": {  "**expirationDate**": {  "type": "string",  "required": true  },  "**sender**": {  "description": " sender’s CBE number",  "type": "string",  "required": true  },    "**mimeType**": {  "type": "string",  "required": false  },  "**receiptDate**": {  "format"  : " YYYY-MM-DD HH:MM:SS",  "type": "string",  "required": true  },  "**status**": {  "type": "string",  “enum”: [“read”,”unread”]  "required": true  },  "**title**": {  "type": "string",  "required": true  },  "**url**": {  "type": "string",  "required": true  }  }  }  },  "**pageNumber**": {  "type": "number",  "required": true  },  "**pageSize**": {  “description”:”max element on page”,  "type": "number",  "required": true  },  "**totalMessages**": {  "description": "total message in ebox for sended parameter",  "type": "number",  "required": true  }  }  }  **Exemple** : */messages /userA/list?pageSize=3*  {  **"messages": [**  {  "**title**": "Avis d'imposition 2013",  "**url**": "https://www.ebox.be/download?messageId=kdsjkfsdjb",  "**receiptDate**": "2014-03-31 14:15:14",  "**expirationDate**": "2019-12-31",  "**sender**": "0206737484",  "**mimeType**": "application/pdf",  "**status**": "unread"  },  {  "**title**": "Déclaration chômage provisoire",  "**url**": "https://www.ebox.be/download?messageId=sdfsdsfds",  "**receiptDate**": "2013-10-31 12:25:17",  "**expirationDate**": "2018-12-31",  "**sender**": "0206737484",  "**mimeType**": "application/pdf",  "**status**": "read"  },  {  "**title**": "Déclaration de revenu 2011",  "**url**": "https://www.ebox.be/download?messageId=dshrerhhhreh",  "**receiptDate**": "2012-03-31 03:39:28",  "**expirationDate**": "2015-12-31",  "**sender**": "0206737484",  "**mimeType**": "application/pdf",  "status": "read"  }  ],  "**pageNumber**": 1,  "**pageSize**": 3,  "**totalMessages**": 15  } |
| **Error Response** | **Code :** 400 – Bad request  **Code :** 403, message : “*No authorization found for this citizen*.”  The citizen does not allow the partner to access his data. The Citizen must link (again) his partner account to eBox. |

## Count messages in eBox

|  |  |
| --- | --- |
| **URL** | /messages/{userId}/count/{status} |
| **Method** | GET |
| **Description** | Return the number of message read, unread or both of them. |
| **URL params** | **Required** :   * userId * status enum : {"read", "unread", "all"} |
| **Success Response** | **Code** : 200  **Content** :  integer  **Exemple** :  2 |
| **Error Response** | **Code :** 400 – Bad request  **Code :** 403, message : “*No authorization found for this citizen*.”  The citizen does not allow the partner to access his data. The Citizen must link (again) his partner account to eBox. |

## Resources: *senders*

|  |  |
| --- | --- |
| **URL** | /senders |
| **Method** | GET |
| **Description** | Return the sender’s data. |
| **URL params** | none |
| **Success Response** | **Code** : 200  **Content** :  {  "properties": {  "**senders**": {  "items": {  "properties": {  "**cbe**": {  "required": true,  "type": "string"  },  "**fullName**": {  "properties": {  "**de**": {  "required": true,  "type": "string"  },  "**fr**": {  "required": true,  "type": "string"  },  "**nl**": {  "required": true,  "type": "string"  }  },  "required": true,  "type": "object"  },  "**shortName**": {  "properties": {  "**de**": {  "required": true,  "type": "string"  },  "**fr**": {  "required": true,  "type": "string"  },  "**nl**": {  "required": true,  "type": "string"  }  },  "required": true,  "type": "object",  "**url**": {  "properties": {  "**de**": {  "required": false,  "type": "string"  },  "**fr**": {  "required": false,  "type": "string"  },  "**nl**": {  "required": false,  "type": "string"  }  },  "required": false,  "type": "object"  }  }  },  "required": true,  "type": "object"  },  "required": true,  "type": "array"  }  },  "required": false,  "type": "object"  }  **Exemple** :  {  "senders": [  {  "bce": "0367303366",  "shortName": {  "fr": "SPF Sécurité sociale.",  "nl": "FOD Sociale Zekerheid.",  "de": "FöD Soziale Sicherheit."  },  "fullName": {  "fr": "Service public fédéral Sécurité sociale",  "nl": "Federale Overheidsdienst Sociale Zekerheid",  "de": "Föderaler öffentlicher Dienst Soziale Sicherheit"  },  "url": {  "fr": "http://socialsecurity.fgov.be/selection/img/layout/logo\_fr.jpg",  "nl": "http://socialsecurity.fgov.be/selection/img/layout/logo\_nl.jpg",  "de": "http://socialsecurity.fgov.be/selection/img/layout/logo\_fr.jpg"  }  },  {  "bce": "0880820673",  "shortName": {  "fr": "Sigedis.",  "nl": "Sigedis.",  "de": "Sigedis."  },  "fullName": {  "fr": "Sociale Individuele Gegevens - Données Individuelles Sociales",  "nl": "Sociale Individuele Gegevens - Données Individuelles Sociales",  "de": "Sociale Individuele Gegevens - Données Individuelles Sociales"  },  "url": {  "fr": "http://www.sigedis.be/data/layout/logo2.gif",  "nl": "http://www.sigedis.be/data/layout/logo2.gif",  "de": "http://www.sigedis.be/data/layout/logo2.gif"  }  },  {  "bce": "0206736989",  "shortName": {  "fr": "OSSOM.",  "nl": "DOSZ.",  "de": "AUSS."  },  "fullName": {  "fr": "OFFICE DE SECURITE SOCIALE D’OUTRE MER",  "nl": "DIENST VOOR DE OVERZEESE SOCIALE ZEKERHEID",  "de": "AMT FUR UBERSEEISCHE SOZIALE SICHERHEIT"  },  "url": {  "fr": null,  "nl": null,  "de": null  }  }  ]  } |
| **Error Response** | **Code :** 400 – Bad request |

## WADL

<application xmlns="http://wadl.dev.java.net/2009/02">

<doc xmlns:jersey="http://jersey.java.net/"

jersey:generatedBy="Jersey: 1.17.1 02/28/2013 12:47 PM" />

<grammars />

<resources base="https://rest.socialsecurity.be/REST/eBox/Citizen/v1/">

<resource path="/messages">

<resource path="{userId}/list">

<param xmlns:xs="http://www.w3.org/2001/XMLSchema"

type="xs:string" style="template" name="userId" />

<method name="GET" id="getMessageList">

<request>

<param xmlns:xs="http://www.w3.org/2001/XMLSchema"

type="xs:string" style="query" name="status" />

<param xmlns:xs="http://www.w3.org/2001/XMLSchema"

type="xs:int" style="query" name="pageNumber" />

<param xmlns:xs="http://www.w3.org/2001/XMLSchema"

type="xs:int" style="query" name="pageSize" />

<param xmlns:xs="http://www.w3.org/2001/XMLSchema"

type="xs:string" style="query" name="sender" />

<param xmlns:xs="http://www.w3.org/2001/XMLSchema"

type="xs:string" style="query" name="orderBy" />

</request>

<response>

<representation mediaType="application/json; charset=utf-8" />

</response>

</method>

</resource>

<resource path="{userId}/count/{status}">

<param xmlns:xs="http://www.w3.org/2001/XMLSchema"

type="xs:string" style="template" name="status" />

<param xmlns:xs="http://www.w3.org/2001/XMLSchema"

type="xs:string" style="template" name="userId" />

<method name="GET" id="getMessageListSize">

<response>

<representation mediaType="application/json; charset=utf-8" />

</response>

</method>

</resource>

</resource>

<resource path="/senders">

<method name="GET" id="getSenders">

<response>

<representation mediaType="application/json; charset=utf-8" />

</response>

</method>

</resource>

</resources>

</application>

# Link Partner account with e-Box



## Request

To link a Partner account with an eBox account, the partner must POST[[1]](#footnote-1) a FORM to MySocialSecuritywith:

|  |  |  |  |
| --- | --- | --- | --- |
| **key** | **value** | **explain** | **Required** |
| **userId** | Id of the user to link | User who want to link is account | Yes |
| **displayUserId** | Id understanding by the user. Example : nickname, firstName | Used when the user will show his linked account. The userId can be to technical to be display. | Yes |
| **partnerId** | Id of the partner | Use to identify the partner. | Yes |
| **responseUrl** | url to post the response | url use by MySocialSecurityto POST the response of the link request. | Yes |
| **emailAddress** | Email address | Email address of the user who want link his account.  *This data is optional* | No |
| **issueInstant** | Current time | The timestamp is used to prevent replay attack.  Request validity  10 minutes in the future  5 minutes in the past.  Exemple : 2008-04-22T06:00:00Z | Yes |
| **signature** | Base64(  Sign(  ${userId}+  ${timestamp}+  ${responseUrl}  )  ) | The concatenation of userId, timestamp and the response url must be signed with the private key of the partner and the result must be encoded in base64. | Yes |
| **OID** | OID of signing algorithm. | Exemple :  sha 1 with RSA = 1.2.840.113549.1.1.5 | Yes |
| **keyAlgorithm** | Value in {RSA,DSA} | Algo | Yes |
| **language** | Value in { FR,NL,DE} | The user’s navigation language. | Yes |

### Post url for mysocialsecurity service

* Validation : <https://proxyacc.socialsecurity.be/web4/citizenprofile/linkRequestListener/v1>
* Production : <https://www.mysocialsecurity.be/web4/citizenprofile/linkRequestListener/v1>

## Response

In response MySocialSecurityPOST a FORM with

|  |  |  |  |
| --- | --- | --- | --- |
| **key** | **value** | **explain** | **Required** |
| **userId** | Id of the user to link | User who want to link is account | Yes |
| **partnerId** | Id of the partner | MySocialSecurityidentifier  <https://www.mysocialsecurity.be> | Yes |
| **linkStatus** | Status of link. Value in {  LINK\_AUTHORIZED,  LINK\_NOT\_AUTHORIZED,  LINK\_EXIST\_FOR\_OTHER\_CITIZEN  } | **LINK\_ AUTHORIZED**:  the user has linked his account with success. The link is created.  **LINK\_NOT\_AUTHORIZED**:  the user doesn’t authorized Mysocialsecurity to link his e-Box account with his partner account. No link exist.  **LINK\_EXIST\_FOR\_OTHER\_CITIZEN**  A link exist with this userId but not for the Citizen authenticate. The citizen authenticated is not the citizen who make the link. | Yes |
| **issueInstant** | Current time | The timestamp is used to prevent replay attack.  Exemple : 2008-04-22T06:00:00Z | Yes |
| **signature** | Base64(  Sign(  ${userId}+  ${timestamp}+  ${linkStatus}  )  ) | The concatenation of userId, timestamp and linkStatus is signed with the private key of Mysocialsecurity and the result is encoded in base64. | Yes |
| **OID** | OID of signing algorithm. | Exemple :  sha 1 with RSA = 1.2.840.113549.1.1.5 | Yes |
| **keyAlgorithm** | Value in {RSA,DSA} | RSA | Yes |
| **language** | Value in { FR,NL,DE} | The user’s navigation language. | Yes |

The response is POST by mysocialsecurity to url receive on param *${responseUrl}*

## Unlink

The call to create or delete a link is the same.

The response return by the service indicate if the link *exist*, *doesn’t exit* or *can’t be create because this user has already linked is account with another Citizen Identity*.



1. browser post redirect [↑](#footnote-ref-1)