REST/grant/citizen

Utilization Guide REST API

DRAFT

1. Introduction

1.1. Purpose of the API

REST/grant/citizen has been developed in order to let external partners easily get access to data sets related to citizen.

REST/grant/citizen will be used by external enterprises to create and manage citizen mandate.

The social security proposed several digital solutions, in some cases, a citizen mandate is needed to execute some actions on behalf of a citizen or to get access to dataset.

1.2. Target audience

REST/grant/citizen API can be used by:

- Company without staffing members
- Employers
- Social secretariat accredited by social security
- Service provider
- Recruitment agency

Please note that the organization must do onboarding procedures to be whitelisted and be authorized to use this service and using citizen mandate.

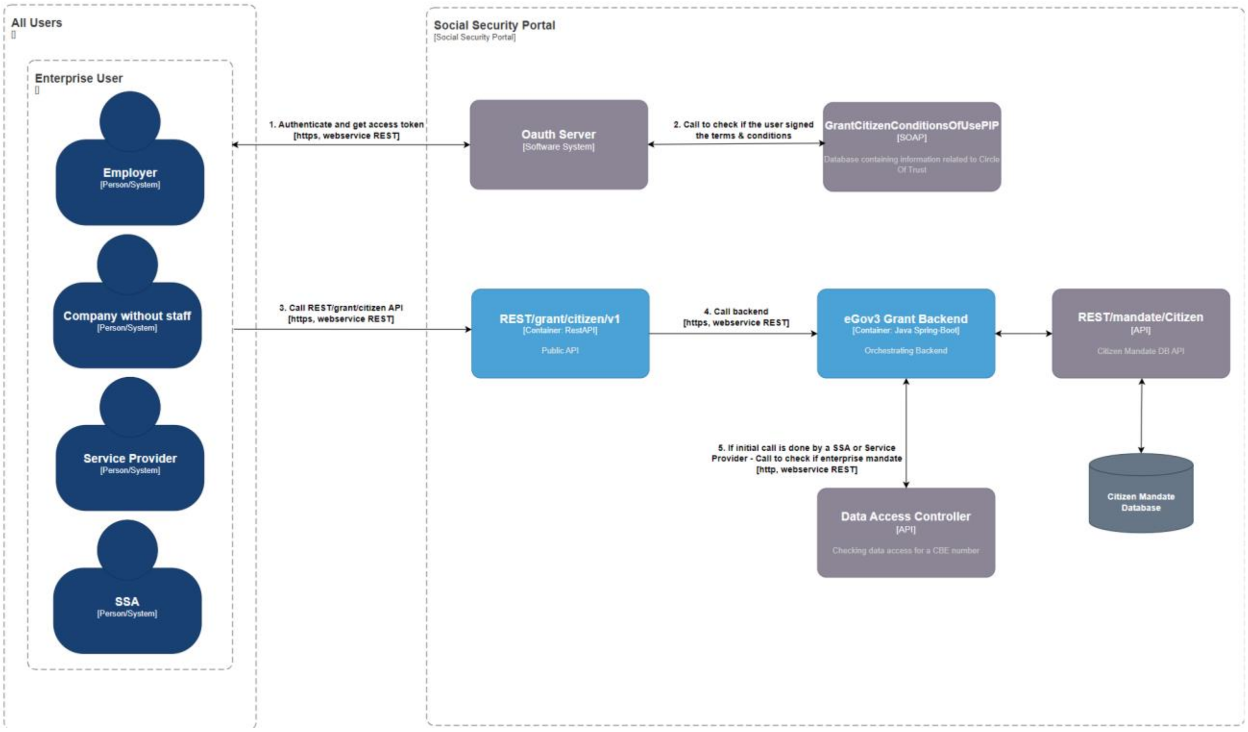
See <Insert link to onboarding document>

1.3. High-level communication

The IT systems of the target group (company, employers, SSA, Service Provider; etc...) can communicate with the REST/grant/citizen API. The communication can only be done via secured channels within the Social Security Portal.

Please note that the onboarding to Social Security Portal needs to be done before using this webservice. See documentation <link to documentation>

Here below you will find a high-level diagram that depicts the flow between the IT system of the target group and the social security system:



1. The user calls the Oauth Server to get an authenticated token from Social Security server with the configured certificate.
2. The Social Security Oauth Server calls the GrantCitizenConditionsOfUsePIP soap service to check if the user approves the terms & conditions related to the usage of citizen mandate.
3. The API call is sent to REST/grant/citizen with the authenticated token.
4. The API call is processed by the eGov3 Grant Backend who will call the REST/mandate/citizen API to get/update/create the mandate information.
5. When the user is a SSA or a Service Provider, using the REST/grant/citizen API on behalf of a client, eGov3 Grant Backend calls Data Access Controller API to check if the user is authorized to do actions on behalf of the client.

Please note that as SSA or Service Provider, you must have an enterprise mandate with your client. See <insert link>

1.4. Base URL

The API can be accessed via the following server:

- Production: http://services.socialsecurity.be/REST/grant/citizen/v1
- Acceptation: http://services-acpt.socialsecurity.be/REST/grant/citizen/v1
- Integration: http://services-int.socialsecurity.be/REST/grant/citizen/v1
- Test: http://services-test.socialsecurity.be/REST/grant/citizen/v1

2. General API operation

2.1. API Architecture

The API is based on a RESTful model, that means that the service works on HTTP standard methods and returns JSON format to facilitate integration.

- Protocol used: HTTPS (to ensure the security of exchanges)
- Data Format: JSON

2.2. Authentication

In order to get access to the REST/grant/citizen API, you must be authenticated by the OAuth server of the social security.

Each call must include a valid access token in the header to be accepted and reach our API.

The authentication flow is of type Oauth via client credentials.

Before being able to call our webservice you must before do a request call to the authentication server in order to get an Oauth token.
Once you received the token from the Oauth Server from Social Security, you will be able to reach or service.

Here you will find detailed documentation on how to implement the Oauth2 client credential flow – click here.

In the section 2.2.1 Scope, you will find the value to provide in the scope field from your request to the Oauth server from Social Security.

Please note that the configuration of a REST Account via Chaman for your organization is a prerequisite to be onboard to the Social Security Network in the context of this webservice.

2.2.1 Scope

In order for you organization to get access to this webservice, you must get the correct scope in your token.
The scope needed is the following one:

```
scope:ksz-bcss:mandates:citizen:management:managemandate
```

For your clientId to obtain this scope from the Oauth server, you must before activate it via Chaman (detailed here) by selecting in the secured permission:

Burgermandaten Beheren
Mandats citoyens - Gérer

Please note that a technical onboarding needs to be done for your organization before to be able to communicate with this webservice.

3. Specific error codes

In this section, you will find the specific error codes:

<u>**400 Bad Request – Invalid CBE number format**</u>

This error code is returned when the CBE number does not conform to the required format.

```
{
  "type": "urn:problem-type:belgif:badRequest",
  "href": "https://www.belgif.be/specification/rest/api-guide/problems/badRequest.html",
  "instance": "urn:trace-id:9e1fd867a08ec4a7c6c1d48c",
  "status": 400,
  "title": "Bad Request",
  "detail": "The input message is incorrect",
  "issues": [
    {
      "in": "body",
      "detail": "[Path '/representedEnterpriseNumber'] ECMA 262 regex \"^[0-1]\\d{9}$\" does not match input string \"09333875470\"",
      "name": "/representedEnterpriseNumber",
      "value": "09333875470"
    }
  ]
}
```

400 Bad Request – Invalid SSIN number format

This error code is returned when the SSIN does not conform to the required 11-digit format.

```
{
  "type": "urn:problem-type:belgif:badRequest",
  "href": "https://www.belgif.be/specification/rest/api-guide/problems/badRequest.html",
  "instance": "urn:trace-id:7c1fd867d08187d3d44b5b8e",
  "status": 400,
  "title": "Bad Request",
  "detail": "The input message is incorrect",
  "issues": [
    {
      "in": "body",
      "detail": "[Path '/mandatorSsin'] ECMA 262 regex \"^\\d{11}$\" does not match input string
\"790723000485\"",
      "name": "/mandatorSsin",
      "value": "790723000485"
    }
  ]
}
```

400 Bad Request – Invalid scope

This error code is returned when the provided scope does not match any of the allowed values. Valid scopes are **precontractualDataFlexiJob** and/or **precontractualDataStudentJob**.

```
{
   "type": "urn:problem-type:belgif:badRequest",
   "href": "https://www.belgif.be/specification/rest/api-guide/problems/badRequest.html",
   "title": "Bad Request",
   "status": 400,
   "detail": "The input message is incorrect",
   "issues": [
      {
         "in": "body",
         "name": "scope",
         "value": [
            "precontractualDataFlexiJobs"
         ]
      }
   ],
   "instance": "urn:trace-id:7321d8673c9b071b5ed15bc6"
}
```

400 Bad Request – missing scope

This error code is returned when the scope is missing from the request body. A mandate request must include at least one scope.

```
{
   "type": "urn:problem-type:belgif:badRequest",
   "href": "https://www.belgif.be/specification/rest/api-guide/problems/badRequest.html",
   "instance": "urn:trace-id:a019d8679b967841520abcde",
   "status": 400,
   "title": "Bad Request",
   "detail": "The input message is incorrect",
   "issues": [
      {
         "in": "body",
         "detail": "Object has missing required properties ([\"scopes\"])"
      }
   ]
}
```

400 Bad Request – Invalid id format

This error code is returned when the resource identifier does not meet the required format (for example, it should be numeric).

```
{
  "type": "urn:problem-type:invalidRequestFormat",
  "title": "Bad Request",
  "status": 400,
  "detail": "The request is not valid, please check the content of the issues array for more information",
  "instance": "urn:traceId:fb21d867f1a0fbec44ae3f87",
  "issues": [
    {
      "in": "query",
      "name": "mandateId",
      "value": "hg",
      "detail": "Value is invalid. Should be numeric."
    }
  ]
}
```

400 Bad Request – Conflicting fields in request body

This error code is returned when the request body contains conflicting fields. For example, **createMandateWithRepresentative** cannot be used together with **representedEnterprise**.

```
{
  "type": "urn:problem-type:citizenMandates:inputFieldsConflict",
  "title": "Input Fields Conflict",
  "status": 400,
  "detail": "These fields are conflicting [createMandateWithRepresentative, representedEnterprise]; only one of them should be provided at a time.",
  "instance": "urn:trace-id:c82cd867b5f6eb200f68fa79"
}
```

403 Forbidden – Enterprise Not Authorized to Access This Resource

This error code is returned when the CBE number in the token is trying to access a resource for which the mandataryEnterpriseNumber does not match the client’ s enterprise number.

```
{
    "type": "urn:problem-type:citizenMandates:userNotPartOfMandate",
    "title": "User not part of mandate",
    "status": 403,
    "detail": "User that tries to get the mandate should either be the mandator or the mandatary of the mandate.",
    "instance": "urn:trace-id:002cd86781f0bf6164049a63"
}
```

403 Forbidden – Enterprise Lacks Necessary Access Rights

This error code is returned when the client uses the **representedEnterpriseNumber** parameter for a CBE number that does not have an active enterprise mandate (cf. Mahis).

```
{
    "type": "urn:problem-type:citizenMandates:representativeEnterpriseForbidden",
    "title": "Representative not authorized to act on behalf of the CBE number",
    "status": 403,
    "detail": "The authenticated user doesn't have the necessary rights to act on behalf of the represented enterprise",
    "instance": "urn:trace-id:a02dd867a1f25600d6806dd6"
}
```

404 Not Found – ID

This error code is returned when the provided identifier does not exist in the database.

```
{
   "type": "urn:problem-type:entityNotFound",
   "title": "Not Found",
   "status": 404,
   "detail": "The following entity [2153151] was not found",
   "instance": "urn:traceId:d02bd86770e53bb31b14e063"
}
```

404 Not Found – RepresentedEnterpriseNumber Not Found in DAC Service

This error code is returned when a Data Access Controller call is made with a representedEnterpriseNumber that does not exist in the Mahis Service.

```
{
   "type": "urn:problem-type:citizenMandates:DacService",
   "title": "Dac Service Problem",
   "status": 404,
   "detail": "Dac Service Problem",
   "instance": "urn:trace-id:1b2dd867a7f7533d304c1338"
}
```

409 Conflict – Mandate Cannot Be Disabled

This error code is returned when the specified mandate is already **stopped** or **inactive**, so it cannot be disabled.

```
{
   "type": "urn:problem-type:citizenMandates:mandateNotDisableable",
   "title": "Mandate Cannot Be Disabled",
   "status": 409,
   "detail": "The requested mandate cannot be disabled because it is not active or has already been
revoked.",
   "instance": "urn:trace-id:8f2cd867daf56563ef425cd9"
}
```

422 Unprocessable Entity – Representative Not Found

This error code is returned when the request body includes **createMandateWithRepresentative** set to ***true*** but the CBE number in the token does not have an active enterprise mandate of type ‘Dmfa’ in Mahis.

```
{
  "type": "urn:problem-type:citizenMandates:representativeNotFound",
  "title": "No representative",
  "status": 422,
  "detail": "Representative not found. A representative is required when createMandateWithRepresentative is set to true."
}
```

503 Service Unavailable – The Service is DOWN

This error code is returned by the get/health endpoint to indicate that the backend service is down.

```
{
  "status": "DOWN"
}
```