RNConsult Permission REST service
Cookbook
Version 1.1

This document is provided to you free, of charge, by the

eHealth platform

Willebroekkaai 38 – 1000 Brussel

38, Quai de Willebroek – 1000 Bruxelles

Table of contents

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

1. Document management

1.1 Document history

Version	Date	Author	Description of changes / remarks
1.0	21/10/2024	eHealth platform	Initial version
1.1	23/03/2026	eHealth platform	Modify tracing section

2. Introduction

2.1 Goal of the service

The goal of this service is to retrieve the list of services, data and their associated conditions, configured by eHealth Security.

Access to this service will be limited to all healthcare organizations (having a valid eHealth certificate) and recognized in the authentic source CoBRHA for RIZIV and EHP organizations.

2.2 Goal of the document

This document is not a development or programming guide for internal applications. Instead, it provides functional and technical information to help an organization integrate the Pseudonymisation REST service into its custom application.

This document will provide all the necessary elements to get started by explaining:

- the main concepts and principles
- technical information about calling the service

This information should enable (the IT department of) an organization to integrate and use the WS call. However, to interact in a smooth, homogeneous and risk controlled way with a maximum of partners, these partners must commit to comply with all the requirements described in this document.

In addition, our partners in the healthcare sector must also comply with the business rules of validation and integration of data within their own applications to minimize errors and incidents. In other words, technical and business requirements must be met to allow the integration and validation of the eHealth platform service in the client application.

2.3 eHealth platform document references

On the portal of the eHealth platform, you can find all the referenced documents.[1]. These versions, or any following ones, can be used for the eHealth platform service.

ID	Title	Version	Date	Author
1	eHealth Services – Web Access	2.0	12/07/2018	eHealth platform
2	Identity & Authorization Management (IAM) Mobile integration	1.9	02/05/2024	eHealth platform

[1] ***www.ehealth.fgov.be/ehealthplatform***

3. Support

3.1 Helpdesk eHealth platform

3.1.1 Certificates

To access the secured eHealth platform environment you have to obtain an eHealth platform certificate, used to identify the initiator of the request. In case you do not have one, please consult the chapter about the eHealth Certificates on the portal of the eHealth platform

- ***https://www.ehealth.fgov.be/ehealthplatform/nl/ehealth-certificaten***
- ***https://www.ehealth.fgov.be/ehealthplatform/fr/certificats-ehealth***

For technical issues regarding eHealth platform certificates

- Acceptance: ***acceptance-certificates@ehealth.fgov.be***
- Production: ***support@ehealth.fgov.be***

3.1.2 For issues in production

eHealth platform contact centre:

- Phone: 02 788 51 55 (on working days from 7 am till 8 pm)
- Mail: ***support@ehealth.fgov.be***
- *Contact Form :*

  https://www.ehealth.fgov.be/ehealthplatform/nl/contact (Dutch)

  https://www.ehealth.fgov.be/ehealthplatform/fr/contact (French)

3.1.3 For issues in acceptance

Integration-support@ehealth.fgov.be

3.1.4 For business issues

- regarding an existing project: the project manager in charge of the application or service
- regarding a new project or other business issues: ***info@ehealth.fgov.be***

3.2 Status

The website ***https://status.ehealth.fgov.be*** is the monitoring and information tool for the ICT functioning of the eHealth services that are partners of the Belgian eHealth system.

4. Requirements

Accessing the RNConsult Permission REST service requires the following condition to be met.

An IAM connect client must be configured across the environment to obtain an access token (JWT) with the appropriate scope required to call the REST service. Once the IAM connect client is configured and an access token (JWT) is obtained, access to the resource will be granted if the role included in the access token is "consult”:

```
"resource_access": {
    "ehealth-rnconsult- permission-api": {
      "roles": [
        "consult"
      ]
    }
  }
```

For more information on the onboarding process, please refer to the following eHealth portal page:

- **FR**: ***https://www.ehealth.fgov.be/ehealthplatform/fr/service-rnconsult***
- **NL**: ***https://www.ehealth.fgov.be/ehealthplatform/nl/service-rrconsult***

4.1 REST and JSON

SOAP and REST are two API styles that approach data transmission from a different point of view. SOAP is a standardized protocol sending messages using other protocols such as HTTP and SMTP.
As opposed to SOAP, REST is not a protocol but an architectural style. The REST architecture defines a set of guidelines to follow if you want to provide a RESTful web service, for example, stateless existence and the use of HTTP status codes. As SOAP is an official protocol, it comes with higher complexity, it requires more bandwidth and resources which can lead to slower page load times. REST was created to address the problems of SOAP.

Therefore, it has a more flexible architecture. It allows different messaging formats, such as HTML, JSON, XML, and plain text, while SOAP only allows XML. REST is also a more lightweight architecture, so **RESTful web services have a better performance**.

The REST architecture allows API providers to deliver data in multiple formats such as plain text, HTML, XML, YAML, and JSON, which is one of its most loved features. Thanks to the increasing popularity of REST, the lightweight and human-readable JSON format has also gained traction, as it is an easy-to-parse and lightweight data-interchange format. In spite of its name, JSON is completely language-agnostic, so it can be used with any programming language, not just JavaScript. JSON files consist of collections of name/value pairs and ordered lists of values that are universal data structures used by most programming languages. Therefore, JSON can be easily integrated with any language.

4.2 Endpoints

The last version of REST interface described with a JSON / Swagger API is available on the eHealth API Portal :

Environment	Endpoint
Acceptance	***https://portal-acpt.api.ehealth.fgov.be***
Production	***https://portal.api.ehealth.fgov.be***

4.3 Technical requirements

4.3.1 Tracing

To use this service, the request SHOULD contain the following two http header values (see RFC *https://datatracker.ietf.org/doc/html/rfc7231#section-5.5.3*):

1. **User-Agent**: information identifying the software product and underlying technical stack/platform. It MUST include the minimal identification information of the software such that the emergency contact (see below) can uniquely identify the component.

    a. Pattern: {minimal software information}/{version} {minimal connector information}/{connector-package-version}

    b. Regular expression for each subset (separated by a space) of the pattern: [[a-zA-Z0-9-\/]*\/[0-9azA-Z-_.]*

    c. *Examples*:
    User-Agent: myProduct/62.310.4 Technical/3.19.0
    User-Agent: Topaz-XXXX/123.23.X freeconnector/XXXXX.XXX

2. **From**: email-address that can be used for emergency contact in case of an operational problem.

    Examples:
    From: ***info@mycompany.be***

Exception for Browser-Based Clients

Browsers control the User-Agent header natively and may block client-side modification for security reasons. When operating in a browser-based environment, supplying the custom User-Agent header is **not required**. That said, it **must** be included whenever the environment permits it.

The From header carries no such restriction and **must** be present in all requests, regardless of environment.

5. API description

5.1 GET /applications/{applicationId}

This resource enables retrieving the detailed configuration of a specific application, identified by its *{applicationId}.*

5.1.1 Request

Example:

```
GET /applications/12345678910
```

Path parameter	Description
applicationId	A string representing the application identifier.

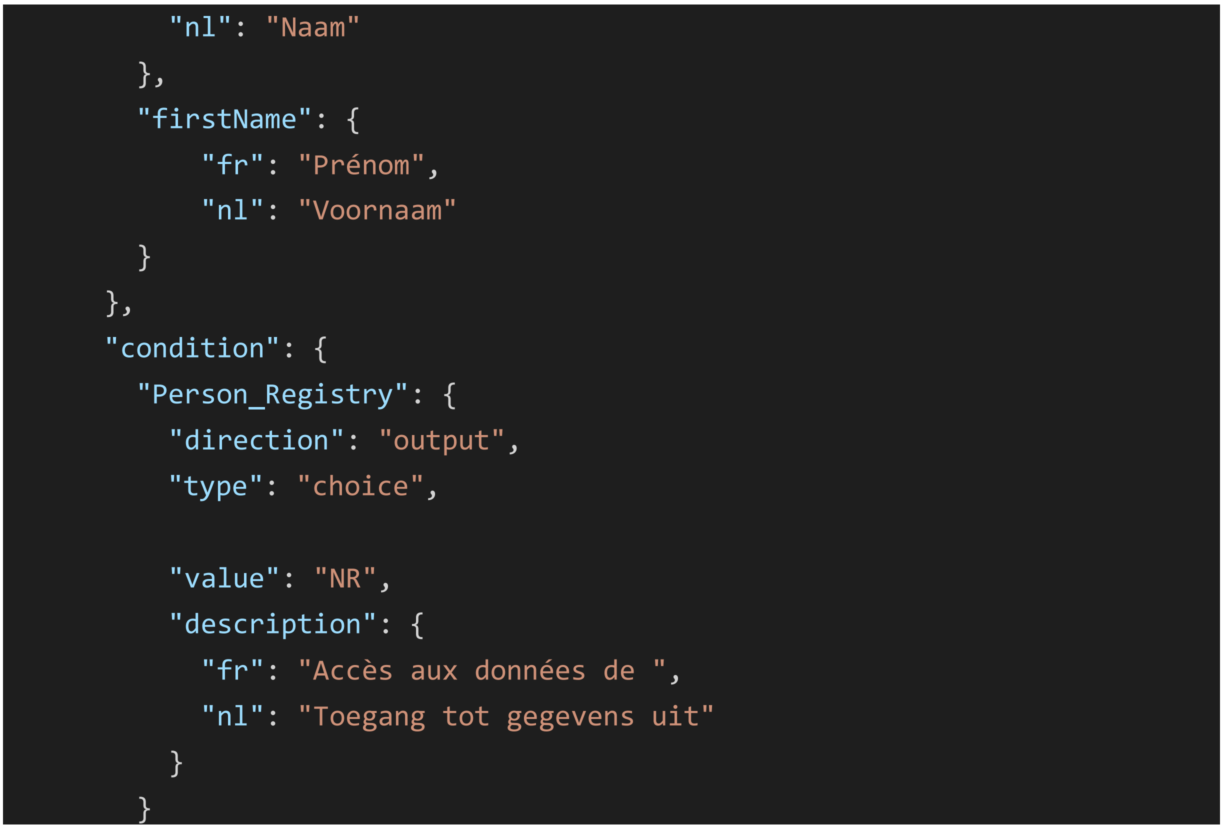
5.1.2 Response

The response returns detailed configuration regarding an application, and the services with data and conditions linked to them.

Element	Description
Application	A object representing the information about the application (please refer to section 5.2).
additionalProperties	A flexible object designed to include an unlimited number of additional properties, each representing a service. The structure of each service will include the objects data and condition, representing data and conditions associated with a service (please refer to section 5.3).

Example:

```
{
    "application": {
      "applicationId": "12345678910",
      "purpose": "Research",
      "status": "Active",
      "period": {
        "startDate": "2023-04-01",
        "endDate": "2024-12-31"
      }
    },
    "personService": {
      "personData": {
        "lastName": {
          "fr": "Nom",
```



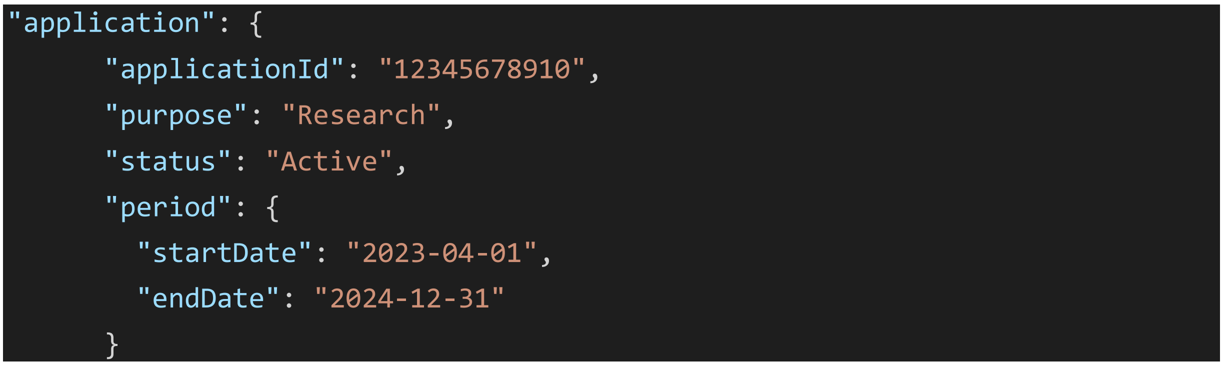
```
          "nl": "Naam"
        },
        "firstName": {
            "fr": "Prénom",
            "nl": "Voornaam"
        }
      },
      "condition": {
        "Person_Registry": {
          "direction": "output",
          "type": "choice",

          "value": "NR",
          "description": {
            "fr": "Accès aux données de ",
            "nl": "Toegang tot gegevens uit"
          }
        }
```

5.2 Application

Property	Description
applicationId	A string representing the application identifier.
purpose	A string representing the purpose of application.
status	A string representing the status of application Identifier. The status can have the following values: “active” or “expired”.
startDate	A string representing the start date of application.
endDate	A string representing the end date of application.

Example:



```
"application": {
      "applicationId": "12345678910",
      "purpose": "Research",
      "status": "Active",
      "period": {
        "startDate": "2023-04-01",
        "endDate": "2024-12-31"
      }
```

5.3 Additional Properties

Element	Description
PersonData	A object representing the data configured for a service (please refer to section 5.3.1).
Condition	A object representing the conditions configured for a service (please refer to section 5.3.2).

Example:

```
"personService": {
      "personData": {
        "lastName": {
          "fr": "Nom",
          "nl": "Naam"
        },
        "firstName": {
            "fr": "Prénom",
            "nl": "Voornaam"
        }
      },
      "condition": {
        "Person_Registry": {
          "direction": "output",
          "type": "choice",

          "value": "NR",
          "description": {
            "fr": "Accès aux données de ",
            "nl": "Toegang tot gegevens uit"
          }
        }
```

5.3.1 Persondata

Element	Description
additionalProperties	A flexible object designed to include an unlimited number of additional properties, each representing a distinct data. The structure for each data will include a description in both FR and NL.

Example:

```
        "firstName": {
            "fr": "Prénom",
            "nl": "Voornaam"
        }
```

5.3.2 Condition

Element	Description
additionalProperties	A flexible object designed to include an unlimited number of additional properties, each representing a condition. The structure of each condition will include a value, an effective value and a description, as detailed in section 5.3.3.

Example:

```
        "Person_Registry": {
          "direction": "output",
          "type": "choice",

          "value": "NR",
          "description": {
            "fr": "Accès aux données de ",
            "nl": "Toegang tot gegevens uit"
          }
```

5.3.3 Condition details

Element	Description
direction	A string indicating whether the condition applies to input or output.
type	A string representing the condition type.
value	A string representing the condition value.
Description	A string representing the condition description in both FR and NL.

Here is the list of conditions that can be returned in a configuration:

condition name	direction	type	Value
person_Registry_RAN	output	choice	RAN (cancelled register)
person_Registry_NR	output	choice	NR (National Register)
person_Registry_RAD	output	choice	RAD (radiated register)
person_Registry_BIS	output	choice	BIS (BIS register)
Content_BirthDate	output	content_age_filtering	The value represents what has been filtered in a date in the format YYYY-MM-DD.

			Example: If the value is ####-MM-DD, the year is masked, and it is impossible to see the year of birth.
Content_Age_BirthDate_+18	output	content_age_filtering	The value represents what has been filtered in a date in the format YYYY-MM-DD. Example: If the value is ####-MM-DD, the year is masked, and it is impossible to see the year of birth.
Content_Age_BirthDate_-18	output	content_age_filtering	The value represents what is filtered in a date in the format YYYY-MM-DD. Example: If the value is ####-MM-DD, the year is masked, and it is impossible to see the year of birth.
Inscription_max_time	output	bigger	The value represents the maximum inscription duration, expressed in month.
Content_InsCode	output	content_single_choice	The value represents the location accuracy for the INS code of the municipality. For example, a value of 1 means that only the first part of the INS code is visible.
nameHistory_modificationDate	output	history	The value represents the number of months of historical data available for the name.
genderHistory_modificationDate	output	history	The value represents the number of months of historical data available for the gender.
nationalityHistory_modificationDate	output	history	The value represents the number of months of historical data available for the nationality.
addressHistory_modificationDate	output	history	The value represents the number of months of historical data available for the address.
civilstateHistory_modificationDate	output	history	The value represents the number of months of historical data available for the civil state.

6. Risks and security

6.1 Security

6.1.1 Business security

In case the development adds a use case based on an existing integration, the eHealth platform must be informed at least one month in advance. A detailed estimate of the expected load is necessary to be able to ensure an effective capacity management.

When technical issues occur on the service, the partner can obtain support from the contact centre (see Chap 3).

If the eHealth platform identifies a bug or vulnerability in its software, the partner must update his application with the latest version of the software, within ten (10) business days.

If the partner finds a bug or vulnerability in the software or web service provided by the eHealth platform, he is obliged to contact and inform us immediately. He is not allowed, under any circumstances, to publish this bug or vulnerability.

6.1.2 The use of username, password and token

The username, password, and token are strictly personal.
Every user is responsible for his username, password, and token, and is required to keep them confidential.
It is prohibited to transfer them or share them with partners and clients. Until account inactivation, each user is responsible for all activities, including those performed by a third party.

7. Implementation aspects

7.1 Procedure

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

7.1.1 Initiation

If you intend to use the eHealth platform service, please contact ***pseudo@ehealth.fgov.be*** who will provide you with the necessary information and mandatory documents.

7.1.2 Development and test procedure

You need to develop a client to connect to our WS. Most of the required integration information is available on the portal of the eHealth platform.

Upon request, and depending on the case, the eHealth platform provides test cases to help you test your client before releasing it to the acceptance environment.

7.1.3 Release procedure

When development tests are successful, you can request access to the acceptance environment of the eHealth platform. From this moment, you begin the integration and acceptance tests. The eHealth platform suggests testing for a minimum of one month.

After successful acceptance tests, the partner sends his test results and performance results with a sample of “eHealth request” and “eHealth answer” by email to his point of contact at the eHealth platform.

Once a release date has been agreed upon, the eHealth platform prepares the connection to the production environment and provides the partner with the necessary information. On the release day, the partner provides the eHealth platform with feedback on the test and performance results.

For further information and instructions, please contact: ***integration-support@ehealth.fgov.be***.

7.1.4 Operational follow-up

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

7.2 Error and failure messages Http codes

These are the error status codes that can be returned by the Permission Rest service:

Code	Message	Description
200	OK	The resource was successfully fetched and the result is included in the message body
400	Bad request	The query parameters or JSON body structure are invalid
401	Not authenticated	A valid JWT was not sent with the request
403	Forbidden access	The access key doesn’t match the user's profile in the JWT

404	Not Found	The resource or endpoint does not exist
500	Internal server error	An unexpected error prevented the request from being fulfilled
503	Service temporarily unavailable	The service is down.