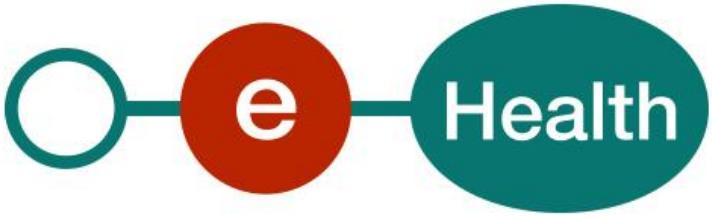


Patient Data Access - Consent WS - REST
Cookbook
Version 1.0

eHealth platform
Willebroekkaai 38
38, Quai de Willebroek
1000 BRUSSELS

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	16/02/2026	eHealth platform	Initial version

2. Introduction

2.1 Goal of the service

The existence of an active '*patient consent*' is one of the fundamental prerequisites for the healthcare providers to access patient's medical data. Therefore, the eHealth platform makes available to the concerned patients and the health care actors involved in the exchange, storage or referencing personal data a service to manage the '*patient consent*' as defined by the deliberation 12/047 of the CSSSS/SCSZG[1].

Technically, we identify the following attributes for an 'patient consent':

- The SSIN of the patient.
- The date of the consent registration.
- The type of the consent.
- The status of the consent.
- The identity of the author of the consent registration.

Multiple types of 'patient consent' can be managed through this service. For each of these types, a default value applies :

Patient consent type	Technical codification	Default status
The consent relating to the electronic sharing of a patient's health data (also called 'informed consent')[2].	dataSharing	Inactive
The consent relating to the referencing of a patient's health data (also called 'opt-out') [3].	dataReferencing	active

The following operations will support the management of these 'patient consents':

GET /refData/consentType	Provides all the resources representing reference data (refData) used in the API for a consent type.
GET /refData/consentStatus	Provides all the resources representing reference data (refData) used in the API for a consent status.
GET /patientConsents	Returns the information about a patient's consent.
PATCH /patientConsents	Updates the status of a patient's consent.
GET /pseudo/patientConsents	Returns the information about a patient's consent for a patient identified by a pseudonymized SSIN.
GET /patientConsents/history	Returns the list of operations that triggered a value change for a consent status either for a specific patient or all patients for a selected period of time. This operations can be used to obtain a "Delta" of the changes.

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

[2] In section 5.2.1, the technical coding relating to this type of consent is 'dataSharing'.

[3] In section 5.2.1, the technical coding relating to this type of consent is 'dataReferencing'.

2.2 Goal of the document

This document describes the management of the 'patient consent' service as provided by the eHealth platform. In this cookbook, we explain the structure and content aspects of the possible requests and the replies of eHealth Consent WS. An example illustrates each of those messages. In addition, a list of possible errors can be found in this document.

This document should support (the IT department of) an organization to develop and use the WS call.

Some technical and legal requirements must be met in order to allow the integration of the eHealth WSs in client applications.

This document is neither a development nor a programming guide for internal applications; eHealth partners always keep total freedom within those fields. Nevertheless, in order to interact in a smooth, homogeneous and risk controlled way with a maximum of partners, eHealth partners must commit to comply with specifications, data format, and release processes described within this document. In addition, our partners in the health sector must also comply with the business rules of validation and integration of data within their applications as to minimize errors and incidents.

2.3 eHealth document references

All the document references can be found on the eHealth platform portal[4]. These versions or any following versions 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	I.AM Connect - Mobile integration - Technical specifications	1.12	15/09/2025	eHealth platform
3	I.AM Connect – HealthCare Client Registration	2.2	30/04/2025	eHealth platform
4	I.AM Connect – M2M Client registration	2.1	03/04/2025	eHealth platform
5	Request test case template	3.0	22/02/2018	eHealth platform
6	IAM Connect Claim mappers	1.0	28/05/2021	eHealth platform

[4] ***https://www.ehealth.fgov.be***

3. Support

3.1 Helpdesk eHealth platform

3.1.1 Certificates

In order 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 8 am till 6 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.

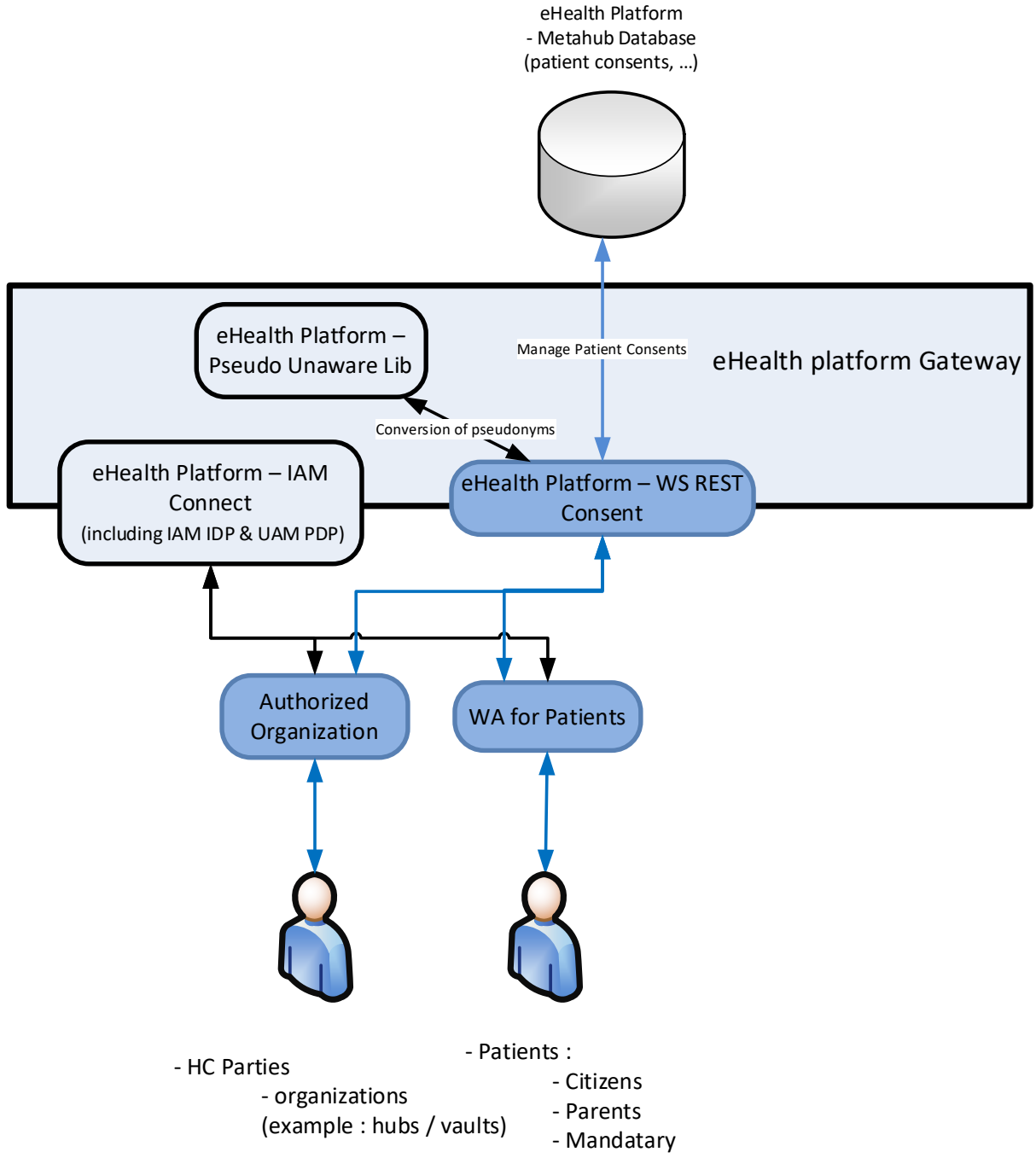
3.3 I.AM Connect

In order to use the Consent REST service you have to obtain an “Access token” which is delivered through I.AM Connect. You can find more information about I.AM Connect and how to register a client in I.AM Connect on the I.AM eHealth portal page:

Dutch version:
https://www.ehealth.fgov.be/ehealthplatform/nl/service-iam-identity-access-management

French version:
https://www.ehealth.fgov.be/ehealthplatform/fr/service-iam-identity-access-management

4. Global overview



This schema gives the users of this cookbook an overview of the management of the patient consents.

5. Step-by-step

5.1 Technical requirements

5.1.1 eHealth platform Authentication

As explained previously, to use the Consent Rest service, you must have an access token delivered through I.AM Connect.

Several roles and profiles are defined for the using of the Consent Rest service.

Possible roles :

- **reader :** This role must be present in the end user's access token to use the patient consent consultation methods.
- **manager :** This role must be present in the end user's access token to use the patient consent consultation and management methods.
- **reader-pseudo :** This role must be present in the end user's access token to use the patient consent consultation methods for a patient identified by a pseudonymized SSIN.
- **reader-audit :** This role must be present in the access token in order to use the patient consent history method (/patientConsents/history) of the service.
- **monitoring :** This role must be present in the access token in order to use the monitoring method (/health) of the service.

Operations allowed by role :

Path	User roles				
	reader	manager	reader-pseudo	reader-audit	monitoring
/patientConsents :					
get:	Y	Y			
patch:		Y			
/pseudo/patientConsents :					
get:			Y		
/patientConsents/history :					
get:				Y	
/health :					
get:					Y
/refData/consentType					
get:	Y	Y	Y	Y	
/refData/consentStatus					
get:	Y	Y	Y	Y	

Presentation of the roles and profiles in the access token:

```
"resource_access":
{
        "ehealth-padac-consent-api":
        {
                "roles":
                [
                   "manager"
                ]
        }
}
```

5.1.2 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***

5.2 The Consent REST Service

The REST interface is described with a JSON/ Swagger API.

The Consent WS has the following endpoints :

- Acceptance environment: ***https://api-acpt.ehealth.fgov.be/patientDataAccess/consent/v1***
- Production environment: ***https://api.ehealth.fgov.be/patientDataAccess/consent/v1***

5.2.1 GET /refData/consentType

The requests of type "GET" on path /refData/consentType allows the user to retrieve the list of reference data (representing a codification) used in the API for a consent type.

5.2.1.1 Request

queryParameter	Description
pageSize (optional)	This parameter allows to select the maximum number of items per page to display. For the last page, its value should be independent of the number of actually returned items. The pageSize value is limited to a minimum of 1 and a maximum of 1000 wich is also the default value. The result is organized alphabetically.
page (mandatory)	This parameter allows to select the specific page of items to display. The default and first page is 1.

Example:

```
GET /patientDataAccess/consent/v1/refData/consentType?pageSize=2&page=1
```

5.2.1.2 Response

Success:

HTTP code 200 (Success) returned in case of an empty or non-empty list of reference data found.
In case of a non-empty list, the body contains an array of reference data objects organized alphabetically.

Examples :

- Empty list of reference data :

```
{
   "items": [],
   "total": 2,
   "pageSize": 2,
   "page": 2
}
```

- Non-empty list of reference data (complete result) :

```
{
   "items":    [
      {"code": "dataReferencing"},
      {"code": "dataSharing"}
   ],
   "total": 2,
   "pageSize": 2,
   "page": 1
}
```

- Non-empty list of reference data (partial result) :

```
{
   "items":    [
      {"code": "dataReferencing"}
   ],
   "total": 2,
   "pageSize": 1,
   "page": 1

"next":"https://api.ehealth.fgov.be/patientDataAccess/consent/v1/refData/consentTyp
e?pageSize=1&page=2"

}
```

Failures:

For failure description please refer to section 8 of this cookbook.

5.2.2 GET /refData/consentStatus

The requests of type “GET” on path /refData/consentStatus allows the user to retrieve the list of reference data (representing a codification) used in the API for a consent status.

5.2.2.1 Request

queryParameter	Description

pageSize (optional)	This parameter allows to select the maximum number of items per page to display. For the last page, its value should be independent of the number of actually returned items. The pageSize value is limited to a minimum of 1 and a maximum of 1000 wich is also the default value. The result is organized alphabetically.
page (mandatory)	This parameter allows to select the specific page of items to display. The default and first page is 1.

Example:

```
GET /patientDataAccess/consent/v1/refData/consentStatus?pageSize=2&page=1
```

5.2.2.2 Response

Success:

HTTP code 200 (Success) returned in case of an empty or non-empty list of reference data found.

In case of a non-empty list, the body contains an array of reference data objects organized alphabetically.

Examples :

- Empty list of reference data :

```
{
   "items": [],
   "total": 2,
   "pageSize": 2,
   "page": 2
}
```

- Non-empty list of reference data (complete result) :

```
{
   "items":    [
      {"code": "active"},
      {"code": "inactive"}
   ],
   "total": 2,
   "pageSize": 2,
   "page": 1
}
```

- Non-empty list of reference data (partial result) :

```
{
   "items":    [
      {"code": "active"}
   ],
   "total": 2,
   "pageSize": 1,
   "page": 1

"next":"https://api.ehealth.fgov.be/patientDataAccess/consent/v1/refData/consentSta
tus?pageSize=1&page=2"

}
```

Failures:

For failure description please refer to section 8 of this cookbook.

5.2.3 GET /patientConsents

The requests of type "GET" on path /patientConsents allows the user to consult a patient's consents.

5.2.3.1 Request

queryParameter	Description
ssin (mandatory)	Social Security Identification Number (SSIN) of the patient concerned.
consentType (optional)	This parameter allows to filter in the consents search based on a list of codifications representing the consent type. By default, all codifications representing the consent type are considered.

Example:

```
GET /patientDataAccess/consent/v1/patientConsents?ssin=12345678910
```

5.2.3.2 Response

Success:

HTTP code 200 (Success) returned in case of an non-empty list of patient consents found.

In all cases, the body contains an array of consents objects organized alphabetically by consent type.

Examples :

- List of patient consents (multiple results) :

```
{
   "items":    [
      {
         "type": "dataReferencing",
         "status": "active"
      },
      {
         "type": "dataSharing",
         "status": "active",
         "since": "2025-01-10"
      }

   ],
   "total": 2
}
```

- List of patient consents (one results) :

```
{
   "items":    [
      {
         "type": "dataReferencing",
         "status": "inactive",
         "since": "2025-01-10"
      }

   ],
   "total": 1
```

```
}
```

Failures:

For failure description please refer to section 8 of this cookbook.

5.2.4 GET /pseudo/patientConsents

The requests of type "GET" on path /pseudo/patientConsents allows the user to consult a patient's consents for a patient identified by a pseudonymized SSIN.

5.2.4.1 Request

queryParameter	Description
ssin (mandatory)[5]	SEC.1 Base64URL (without padding) representation of the point of the elliptic curve representing the pseudonymised data in transit.
transitInfo (mandatory)	Compact JWE representation containing the encrypted transitInfo.
consentType (optional)	This parameter allows to filter in the consents search based on a list of codifications representing the consent type. By default, all codifications representing the consent type are considered.

Example:

```
GET
/patientDataAccess/consent/v1/pseudo/patientConsents?ssin=AgBDSWPpHUCBzzoLxu8yoM5dp
2ZRwm3oAAGWVAtj8k1MRXb9UsaRXz9sx19RsSSIm1fcu0VH52TgwyoxGM1A5qfvYA&transitInfo=eyJhd
WQiOiJodHRwczovL2FwaS1pbnRyYy5laGVhbHRoLmZnb3YuYmUvcHNldWRvL3YxL2RvbWFpbnMvZWhlYWx0
aF92MSIsImVuYyI6IkEyNTZHQ00iLCJleHAiOjE3MzA3Mjg2NjAsImlhdCI6MTczMDcyODA2MCwiYWxnIjo
iZGlyIiwia2lkIjoiZTI0YWVkYjEtZGRjMC00YjI3LWI5OTQtNzgwNDhlNmFlNjUyIn0..uCrxaTU6s936T
hTj.9zvxoXvI-wlQepfG5klv1_msgACSizlqHziP2ID1-d7bB-aBsVlZV4ihh2pcRLNXvc-
Y7H2_2bjsO29c7dGvcTpONv5Ky_ZgMM8y0ElAj_4HPbVNI0s5eg3ftBbJaYeeOQzfCWkIrqp7JvRf-
0yQIhSDYl9ZDeXrK9rmH1jJ98IBg8XcUuCe.Q9-09q_LzNoGqmMEugRA2g
```

5.2.4.2 Response

Success:

HTTP code 200 (Success) returned in case of an non-empty list of patient consents found.

In all cases, the body contains an array of consents objects organized alphabetically by consent type.

Examples :

- List of patient consents (multiple results) :

```
{
   "items":    [
      {
         "type": "dataReferencing",
         "status": "active"
      },
      {
         "type": "dataSharing",
         "status": "active",
         "since": "2025-01-10"
```

[5] Pseudonymised SSIN in transit (for the domain "ehealth_v1").

```
      }

   ],
   "total": 2
}
```

- List of patient consents (one results) :

```
{
   "items":    [
      {
         "type": "dataReferencing",
         "status": "inactive",
         "since": "2025-01-10"
      }

   ],
   "total": 1
}
```

Failures:

For failure description please refer to section 8 of this cookbook.

5.2.5 PATCH /patientConsents

The requests of type “PATCH” on path /patientConsents allows the user to update a patient’s consents.

5.2.5.1 Request

queryParameter	Description
ssin (mandatory)	Social Security Identification Number (SSIN) of the patient concerned.

bodyParameter	Description
items (mandatory)	A list of distinct consents. A consent is composed of 3 elements: - Type[6] : the identifier of the codification which defines a consent type. - Status[7] : the identifier of the codification which defines a consent status. - Since : the date of the consent status update. The element ‘since’ is optional for a PATCH operation and should not be used. Whether it is provided or not, the element is enriched by the date of the declaration.
total (optional)	This parameter allows to specify the total number of items.

[6] Existing codifications can be consulted via the GET /refData/consentType (see section 5.2.1) operation.

[7] Existing codifications can be consulted via the GET /refData/consentStatus (see section 5.2.2) operation.

Examples :

- Update one consent for a patient :

```
PATCH /patientDataAccess/consent/v1/patientConsents?ssin=12345678910

{
   "items":    [
      {
         "type": "dataReferencing",
         "status": "inactive"
      }

   ],
   "total": 1
}
```

- Update multiple consents for a patient :

```
PATCH /patientDataAccess/consent/v1/patientConsents?ssin=12345678910

{
   "items":    [
      {
         "type": "dataReferencing",
         "status": "inactive"
      },
      {
         "type": "dataSharing",
         "status": "inactive"
      }

   ],
   "total": 2
}
```

5.2.5.2 Response

Success:

HTTP code 200 (Success) returned in case of success for the update of the all consents.

In all cases, the body contains an array of consents objects organized alphabetically by consent type.

If consent is already in the desired state, then the PATCH operation will not modify the consent object whose information will be returned in the body resulting from the response.

Examples :

- List of patient consents (for updating one consent) :

```
{
   "items":    [
      {
         "type": "dataReferencing",
         "status": "inactive",
         "since": "2025-01-10"
      }

   ],
   "total": 1
```

```
}
```

- List of patient consents (for updating multiple consents) :

```
{
   "items":    [
      {
         "type": "dataReferencing",
         "status": "inactive",
         "since": "2025-01-10"
      }
      {
         "type": "dataSharing",
         "status": "active",
         "since": "2025-01-10"
      }

   ],
   "total": 2
}
```

Failures:

For failure description please refer to section 8 of this cookbook.

5.2.6 GET /patientConsents/history

The requests of type “GET” on path /patientConsents/history returns the changes related to a value change for a consent status either for a specific patient or all patients for a selected period of time.

5.2.6.1 Request

queryParameter	Description
ssin (optional)	Social Security Identification Number (SSIN) of the patient concerned.
consentType (optional)	This parameter allows to filter in the consents search based on a list of codifications representing the consent type. By default, all codifications representing the consent type are considered.
from (mandatory)	The inclusive start date of the validity period of the search. The format to follow is as follows : - yyyy-MM-dd'T'HH:mm:ss.SSSXXX

until (optional)	The exclusive end date of the validity period of the search. The element *'until'* is optional : - if provided then *'until'* **must be strictly greater than** '*from'* else, the request is rejected. - If not provided then '*until'* is enriched with the current date and time. The format to follow is as follows : - yyyy-MM-dd'T'HH:mm:ss.SSSXXX
pageSize (optional)	This parameter allows to select the maximum number of items per page to display. For the last page, its value should be independent of the number of actually returned items. The pageSize value is limited to a minimum of 1 and a maximum of 1000 wich is also the default value. The result is organized in chronological order.
page (mandatory)	This parameter allows to select the specific page of items to display. The default and first page is 1.

Example:

```
GET /patientDataAccess/consent/v1/patientConsents/history?from=2024-11-
05T16%3A34%3A00%2B01%3A00&until=2024-11-05T16%3A41%3A00%2B01%3A00&pageSize=2&page=1
```

5.2.6.2 Response

Success:

HTTP code 200 (Success) returned in case of an empty or non-empty list of changes found.

In case of a non-empty list, the body contains an array of changes objects organized in chronological order.

Examples :

- Empty list of changes :

```
{
   "items": [],
   "total": 0,
   "pageSize": 2,
   "page": 1,

}
```

- Non-empty list of changes :

```
{
   "items": [
    {
      "ssin": "12345678910",
      "timestamp": "2025-11-14T12:58:21.641Z",
      "author": [         {
            "qualificationCode": "mandatary",
            "ssin": "10987654321"
        }],
```

```
      "consent": {
        "type": "dataSharing",
        "status": "active",
        "since": "2025-01-10"
      }
    },
    {
      "ssin": "10987654321",
      "timestamp": "2025-11-14T12:58:21.641Z",
      "author": [         {
            "qualificationCode": "mandatary",
            "ssin": "12345678910"
         }],
      "consent": {
        "type": "dataReferencing",
        "status": "inactive",
        "since": "2025-01-08"
      }
    }
  ],
  "total": 6,
  "pageSize": 2,
  "page": 1,
  "next":
"https://api.ehealth.fgov.be/patientDataAccess/consent/v1/patientConsents/history?p
ageSize=2&page=2&from=2024-11-05T16%3A34%3A00%2B01%3A00&until=2024-11-
05T16%3A41%3A00%2B01%3A00"}
```


Failures:

For failure description please refer to section 8 of this cookbook.

6. Risks and security

6.1 Risks & safety

6.2 Security

6.2.1 Business security

In case the development adds an additional use case based on an existing integration, the eHealth platform must be informed at least one month in advance with a detailed estimate of the expected load. This will ensure an effective capacity management.

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

If the eHealth platform should find 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 made available 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.

6.2.2 The use of username, password and token

The username, password, and token are strictly personal.
Every user takes care of his username, password and token, and he is forced to confidentiality of it. It is prohibited to transfer them to partners and clients. Until inactivation, every user is responsible for every use, including the use 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 ***info@ehealth.fgov.be***. The project department will provide you with the necessary information and mandatory documents.

7.1.2 Development and test procedure

You have to develop a client in order to connect to our WS. Most of the information needed to integrate is published on the portal of the eHealth platform.

Upon request, the eHealth platform provides you with test cases (see ***Request Test Case template***) in order for you to test your client before releasing it in the acceptance environment.

7.1.3 Release procedure

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

After successful acceptance tests, the partner sends his test 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 on, the eHealth platform prepares the connection to the production environment and provides the partner with the necessary information. During the release day, the partner provides the eHealth platform with feedback on the test and performance tests.

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 Test cases

eHealth recommends performing tests for all of the following cases (depending on your access roles):

- GET /refData/consentType : successful consultation of the reference data for consent types.
- GET /refData/consentStatus : successful consultation of the reference data for consent status.
- GET /patientConsents or /pseudo/patientConsents : successful consultation of the patient consents.
- PATCH /patientConsents : successful update of the patient consents.
- GET /patientConsents/history : successful consultation of the patient consent management history.
- In addition, you should also run negative test cases.

8. Error and failure messages

8.1 HTTP codes

8.1.1 HTTP 2xx

The meaning of 2xx HTTP codes is described in each Operation in section 5 of this document.

8.1.2 HTTP 500

This code means that an internal technical error occurred during the processing of the request.
This does not necessarily mean that the error is not due to a wrong input in the request.
Please double-check that your request is correct before reaching to the helpdesk.

8.1.3 HTTP 400

This code means that the request could not be performed due to a validation error.
The body returned explains what went wrong in your request.
Example:

```
{
   "type": "urn:problem-type:ehealth:consent:refData:invalidCode",
   "href": "https://www.ehealth.fgov.be/ehealthplatform/fr/service-ehealthconsent",
   "title": "Invalid refData",
   "status": 400,
   "detail": "There is no refData matching with the provided parameter.",
   "instance": "5ce98e698e58def7fb41eca5",
   "issues": [   {
      "in": "query",
      "name": "consentType",
      "detail": "The provided parameter dataShared does not match an existing
reference data.",
      "value": "dataShared"
   }],
   "id": "5ce98e698e58def7fb41eca5"}
```

Here are the possible values that you can get in the body for the “400” and “403” errors:

Forbidden operation	The user is not authorized to perform this operation.
Invalid refData	There is no refData matching with the provided parameter.
Invalid Consent Type	The consent type is invalid.
Invalid search period	The provided search period is incorrect.
Invalid identifier	The provided patient identifier is invalid.
Person is deceased	You cannot change the consent status of a deceased person.