UHMEP – API

Referral prescription (RIZIV-INAMI)

Cookbook

MAY 25, 2023

SMALS

Table of Contents

1 Document management

1.1 Document history

Version	Status	Date	Autor(s)	Modifications
0.1	Final	25/05/2023	Gaël Mavangui, Julien Beard, Cyprien Janssens	First version of the cookbook for the release 1.0 scope « MVP » of UHMEP

1.2 Document reviews

Reviewers	Name(s)	Reviewed Version	Comments
SPOC Client	Maarten Cobbaert	0.1	
Project Manager	Maxime Daive	0.1	
Chain Service Manager	Nicolas Rogge	0.1	

2 Reference

2.1 eHealth reference

All referenced documents are available on the portal of the eHealth platform. 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 (I.AM) Mobile integration	1.7	31/03/2023	eHealth platform
3	Pseudonymization REST	1.0	04/04/2023	eHealth platform

2.2 Fhir reference

ID	Fhir documentation type	Link
1	Implementation guide	http://build.fhir.org/ig/hl7-be/referral/guidance.html
2	BeReferralPrescriptionNursing	http://build.fhir.org/ig/hl7-be/referral/StructureDefinition-be-referralprescription-nursing.html
3	BeReferralTask	http://build.fhir.org/ig/hl7-be/referral/StructureDefinition-be-referral-task.html
4	BePerformerTask	http://build.fhir.org/ig/hl7-be/referral/StructureDefinition-be-performer-task.html
5	BePractitionerRole	https://build.fhir.org/ig/hl7-be/core//StructureDefinition-be-practitionerrole.html
6	BePractitioner	https://build.fhir.org/ig/hl7-be/core//StructureDefinition-be-practitioner.html
7	Examples	https://build.fhir.org/ig/hl7-be/referral/artifacts.html#example-example-instances

3 Document information

3.1 Glossary

ID	Term	Definition
1	Reference ID	This is a unique number assigned by UHMEP to each referral prescription issued.
2	UHMEP	This is the name given to the API which allows to create and manage referral prescriptions and propositions. UHMEP stands for “Unaddressed Health Message Exchange Platform”.
3	Therapeutic link	It is a relation that a healthcare professional has to establish with the patient to have access to his medical data.
4	Therapeutic exclusion	This relation can be created by the patient and allows him to prevent an healthcare professional to access to medical data coming from other professionals.
5	Prescriber	The person who can prescribe a referral prescription for a patient.
6	Caregiver	The caregiver is a general term for the person who provides care based on the information written on the referral prescription. Also known as "treatment provider" (prescriber as well as treatment provider are considered to be caregivers).
7	Patient	The citizen which is the subject of the referral prescription.
8	Assignation	The assignation is a recommendation done by the prescriber or is done by the patient to help the caregiver to find the prescription easily.
9	Execution	The execution represents the status of the care that must be provided by a caregiver.

3.2 Formatting explanation

- When a word is in bold and italic : that is a field in the FHIR specification. Ex : ***status***.
- When a word is between quotations mark “” : This is the value that the specific field can have. Ex : “ready”.
- Each time a FHIR resource is referenced, an hyperlink has been set to the resource in the FHIR specification.
- When a role is between parentheses in the begin of sentences, it means that the business rules explained next to it is applicable to this role. Ex : (Prescriber).

4 Goal of the document

The goal of this document is to provide all the information needed to integrate UHMEP.

This document will describe the project but also what is around the project like how an integrator can have some support and what is needed to call UHMEP.

UHMEP uses FHIR to exchange messages with the integrators. This document will explain some fundamental concepts to understand what must be given and what happens by doing the different operations.

The access management will be also explained to allow the integrators to know when an operation can be called.

After these explanations, this document will describe all the operations available respecting the following structure :

- A description of the operation.
- The endpoint to call.
- The roles that a user must have to call the operation.
- The rules implemented in UHMEP
- Some information about the request to send
- Some information about the response

5 Support

5.1 For issues in acceptance

Issues in acceptance can be reported by sending a mail to integration-support@ehealth.fgov.be.

5.2 Certificates

In order to access the secured eHealth platform environment each integrator has to obtain an eHealth platform certificate, used to identify the initiator of the request. 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

6 Global overview

6.1 Context (Goal of the project)

UHMEP ("Unaddressed Health Message Exchange Platform") is an exchange platform that stores referral prescriptions and makes them available to healthcare professionals and patients.

The project wants to achieve 2 objectives:

- More efficient processing of the referral prescription
- Improve the delivery of care and its support activities

Referral prescriptions are the non-drug prescriptions that a patient receives when he goes to his doctor (the prescriber) for a certain problem. A referral prescription is executed by the caregiver. For example, it can be a prescription for wound care, for an X-ray, etc.

A care proposal results from the reverse process, i.e. it is the caregiver who will create a care proposal to propose to prolongate an existing care or to communicate observations about a potentially new health problem.
This feature will come in a next release.

The digitization of referral prescriptions and care proposals will reduce the administrative load thanks to the reduction in the use of paper versions and thanks to the possibility of instantly retrieving a specific prescription. The prescription can also be consulted simultaneously by the different actors, which was not possible with the paper version.

Another advantage of this digitization is the centralization of all this information in one place, which is the UHMEP database.

In addition, UHMEP will use templates for each type of referral prescription as well as for care proposals. These models will be based on the international FHIR standard and adapted at the national level by the eHealth standardization team. FHIR is a standard that describes the format and the exchange of medical data between different computer systems.

The use of models will make it possible to avoid certain errors when writing and to avoid potential confusion when reading the prescription by the different actors.

The UHMEP database will therefore have all the information of referral prescriptions and medical proposals for all Belgian citizens. Referral prescriptions created within a hospital for internal use are not within the scope of UHMEP.

Around this database, an API will be set up in order to allow the various actors to access this information. The API will be used by several applications for healthcare professionals to enable them to carry out their work, but also by a web and mobile application for citizens so that they can consult the status of their prescriptions. This API will be deployed on the eHealth API Gateway.

6.2 Access management :

6.2.1 eHealth token

The token of eHealth allows UHMEP to identify who is connected. Each integrator has to send the user token with the request if they want to use the UHMEP API.

The token has the following information that will be used for the access management :

- Pseudonymized SSIN for the “patient” role or SSIN for the healthcare professional
- The type of professional which give us the discipline
- The role of the user connected (Described in the section 6.2.2)
- The suspension information (In a future release of this document, it will be described how this information will be represented in the token)

There is other information present in the token but those are not used for the access management.

6.2.2 Roles

A connected user can have three roles. This role is used to make a first filter on which operation can be accessed or not.

The three roles are :

- Prescriber
- Caregiver
- Patient

The operations accessible by a patient are the following ones :

- Consultation of a prescription
- Consultation of a list of prescriptions
- Consultation of a task which describes the execution of a caregiver on the prescription
- Assignation of a caregiver to a prescription
- Removal of a caregiver from a prescription
- Cancellation of a prescription

There is no distinction between patients regarding access.

The operations accessible by a prescriber are the following ones :

- Creation of a prescription
- Consultation of a prescription
- Consultation of a list of prescriptions
- Consultation of a task which describes the execution of a caregiver on the prescription
- Assignation of a caregiver to a prescription
- Removal of a caregiver from a prescription
- Cancellation of a prescription

The operations accessible by a caregiver are the following ones :

- Consultation of a prescription
- Consultation of a list of prescriptions
- Consultation of a task which describes the execution of a caregiver on the prescription
- Execution of a treatment (start, finish, cancel, interrupt)
- Rejection of an assignation

For the role Prescriber and Caregiver, after the first filter, the access matrix will be used to determine if the connected user can access to an operation depending on his discipline and the type of prescription he is handling.

6.2.3 Access matrix

In UHMEP, the access matrix checks the authorization for the healthcare professional when he tries to take an action on a prescription. It gives the access management rules and verifies if the user is allowed to make the request.

The access matrix is composed of :

- The methods available in UHMEP.
- The different types of prescription (templates) defined by INAMI
- The discipline of the healthcare professional (retrieved from the token).

Each time the connected user will try to make an action in UHMEP, the application will verify these three data and determine whether or not the user can act on the resource.

The access matrix is defined by the INAMI. The access matrix can be found in the business requirements release.

6.2.4 Suspension

If an healthcare professional is suspended, it means that he is not authorized to work. ***This notion must still be correctly defined by the different stakeholders working on the project UHMEP. The correct definition will come in a future release.***

When a professional is suspended, certain operations are not accessible anymore. For the ones to which he has still access, the operational behavior will be modified. This will be described in the section 8.

6.2.5 Therapeutic link & exclusion

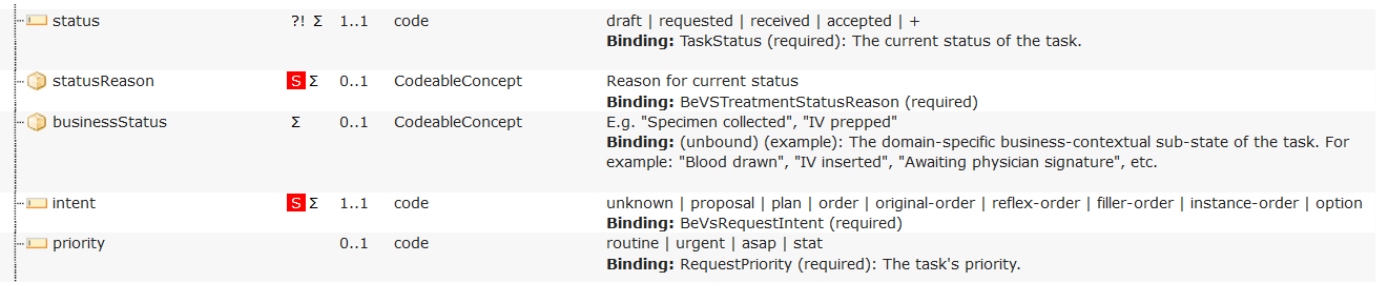
UHMEP verifies the existence verification of the therapeutic link and the therapeutic exclusion between a patient and an healthcare professional. This verification has also an impact on the access right to an operation. The restriction linked to these verifications will be described in the section 8.

6.3 FHIR

FHIR (Fast Healthcare Interoperability Resource) is used by UHMEP as the transport layer. All communications with UHMEP should follow guidelines defined by HL7. The Belgian version of FHIR is managed by eHealth. All links in this document should refer to the Belgian FHIR resources when provided, or to international ones if no changes have been enforced in Belgium.

All resources used in UHMEP are defined and described in the implementation guide artifacts from HL7 Belgium. This document will not explain different fields of these resources, but will refer to the FHIR profile definition used by UHMEP. In general, in each resource, UHMEP supports all fields marked as "Must Support" ***(This will be confirmed in a future release)*** and all mandatory fields in the FHIR documentation. For instance, Figure 1 shows a portion of the implementation guide of the BeReferralTask and it states that :

- ***status*** has a cardinality of "1..1" so it is mandatory and used by UHMEP
- ***statusReason*** and ***intent*** are marked as "Must Support" (the red "S"), so they are used by UHMEP
- ***businessStatus*** is optional and not marked as "Must Support", so it may be discarded by UHMEP ***(This will be confirmed in a future release)***.



status	?! Σ	1..1	code	draft | requested | received | accepted | + **Binding:** TaskStatus (required): The current status of the task.
statusReason	S Σ	0..1	CodeableConcept	Reason for current status **Binding:** BeVSTreatmentStatusReason (required)
businessStatus	Σ	0..1	CodeableConcept	E.g. "Specimen collected", "IV prepped" **Binding:** (unbound) (example): The domain-specific business-contextual sub-state of the task. For example: "Blood drawn", "IV inserted", "Awaiting physician signature", etc.
intent	S Σ	1..1	code	unknown | proposal | plan | order | original-order | reflex-order | filler-order | instance-order | option **Binding:** BeVsRequestIntent (required)
priority		0..1	code	routine | urgent | asap | stat **Binding:** RequestPriority (required): The task's priority.

Figure 1 : Part of the profile definition of the BeReferralTask

The implementation guide should be used by both integrators and UHMEP just before sending the message and before processing it. This implementation guide is a JSON validator on both end of the communication. All information about how to use it is available on the link :

http://build.fhir.org/ig/hl7-be/referral/index.html

6.3.1 Capability Statement

As a FHIR compliant server, UHMEP makes available a statement of the accessible features. It is described in the CapabilityStatement page from the FHIR International website.

This method is accessible to everyone, without any token from eHealth. The endpoint is the following :

`GET /metadata`

UHMEP will send back the resource CapabilityStatement filled with information about the API.

6.3.2 A prescription in FHIR resources

To create a Belgian prescription in UHMEP, the prescriber has to create and send to UHMEP the resource BeReferralPrescription. Depending on its content, UHMEP will make available different other resources linked to this BeReferralPrescription. A BeReferralTask is always made available on creation of a prescription. It has the same ID as the one of the BeReferralPrescription and is linked to the BeReferralPrescription through its ***basedOn*** field and can be consulted with the endpoint :

`GET /Task/{ID}`

On prescription creation, BePractitionerRole and BePractitioner resources are also made available for consultation as described in section 6.3.4.

6.3.3 Execution of caregivers on prescriptions

The Belgian referral prescription is mainly stored in the BeReferralPrescription. Once created by the prescriber, clinical data in this resource are fixed and should not be changed afterward. This prescription will start its "prescription status lifecycle" like described in the business requirements release.

Another resource is used to handle caregiver interactions with the prescription, namely the BePerformerTask. This resource is linked to the BeReferralPrescription through a reference in its ***basedOn*** field and a reference to the BeReferralTask in its ***partOf*** reference. The Figure 2 below shows links between BePerformerTasks, BeReferralTask and BeReferralPrescription when there are two executions on a prescription.

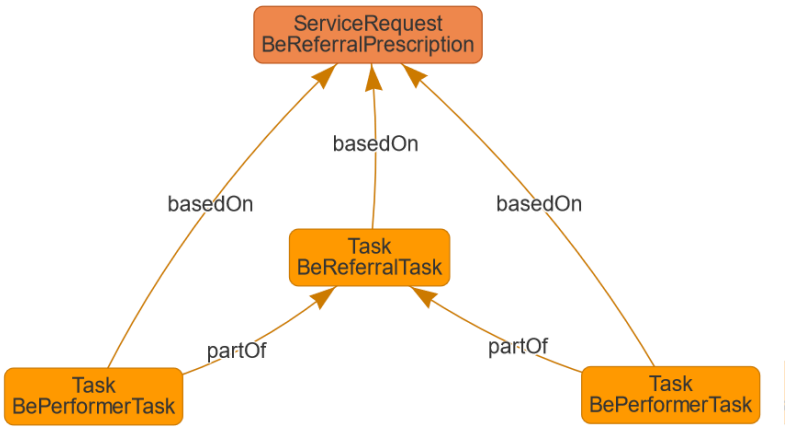


Figure 2 : Links between Task resources to the BeReferralPrescription

All information about operations and consultation that can be done on this resource is available in section 8.

The resource is created in two use cases :

- At the assignation of a caregiver on a prescription by a Prescriber or Patient
- When the caregiver starts his execution and the resource wasn't already created (due to assignation)

After this, all operations are done using the PATCH operation of FHIR. The Figure 3 below shows the status flow of a BePerformerTask. Gray statuses mean that no further operation is allowed on this state. For instance, once the BePerformerTask ***status*** is "rejected", "on-hold" or "completed", the resource cannot be updated anymore.

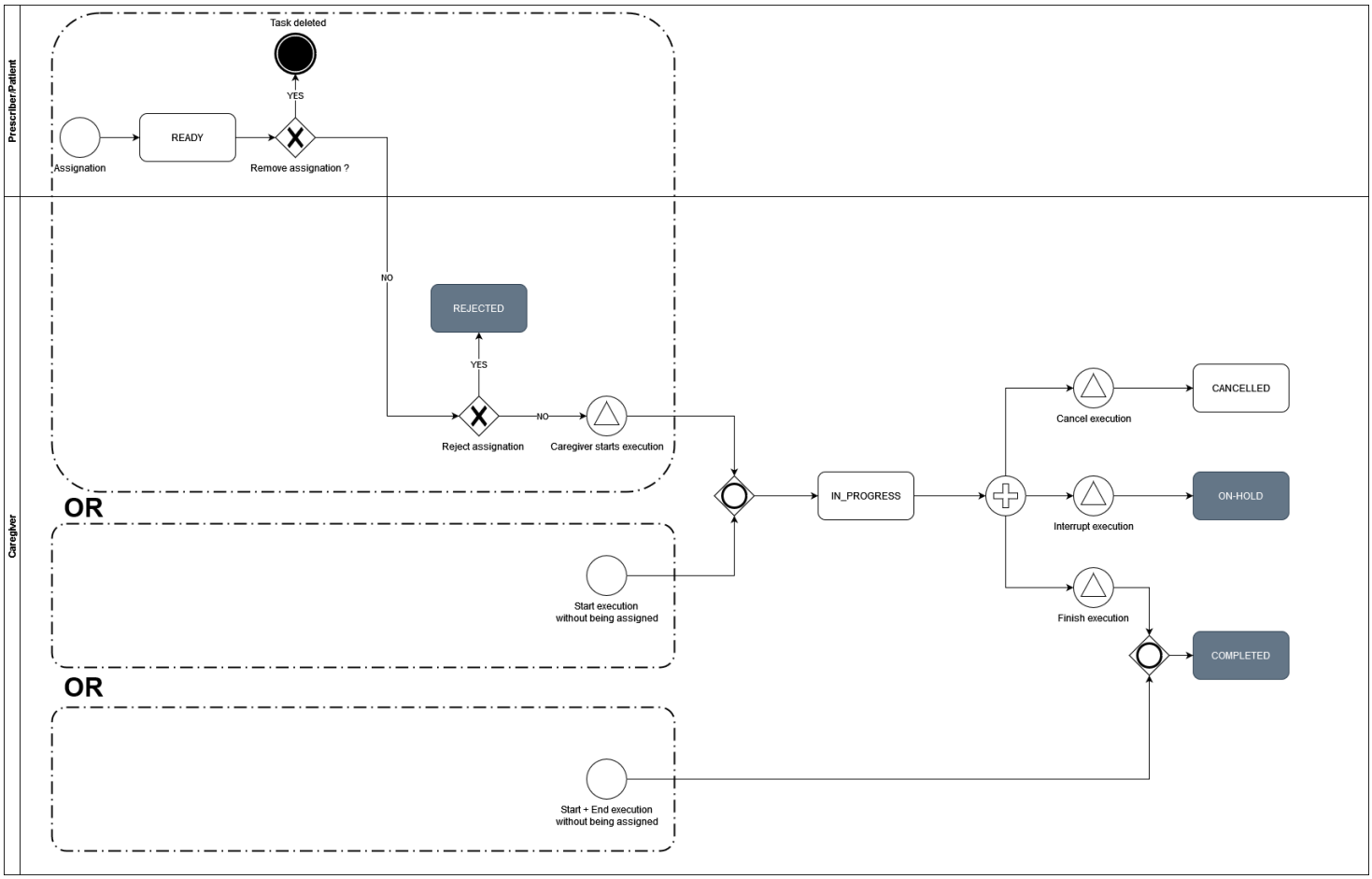


Figure 3 : Status diagram of BePerformerTask

6.3.4 References to healthcare professionals

As mentioned above, UHMEP uses FHIR as the transport layer for all communications. In resources like the BeReferralPrescription and BePerformerTask, references to healthcare professionals are done with references to BePractitionerRole FHIR resource.

IDs of BePractitionerRole in UHMEP are the concatenation between the SSIN and the discipline of the healthcare professional. For example for a “NURSE” with “82042605839” as SSIN, the ID of the BePractitionerRole in UHMEP for this person would be : “82042605839-NURSE”.

Disciplines are defined by eHealth/INAMI and available through CoBRHA. Disciplines should be in UPPERCASE and all spaces replaced by underscores ("_").

For all resources, references can be put directly like "PractitionerRole/82042605839-NURSE". In the BePerformerTask for example, the reference of the ***owner*** field will have the format like below :

```
"owner": {
    "reference": "PractitionerRole/82042605839-NURSE"
}
```

Snippet 1 : Owner reference example in a BePerformerTask

The BePractitionerRole is created by UHMEP and can be consulted using the usual FHIR resource consultation endpoint format :

GET /PractitionerRole/82042605839-NURSE

This resource itself does a reference to a ***practitioner*** that is a reference to a BePractitioner FHIR resource. IDs of BePractitioner are only the SSIN of the healthcare professional.

An example is available below :

```
"practitioner": {
    "reference": "Practitioner/82042605839"
}
```

Snippet 2 : Practitioner reference example in a BePractitionerRole

Similarly, the BePractitioner resource is created by UHMEP and can be consulted using the usual FHIR resource consultation endpoint format :

GET /Practitioner/82042605839

7 Technical requirements

7.1 Tracing

To use this service, the request MUST 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.

    a. Pattern: {company}/{package-name}/{version} {platform-company}/{platform-package-name}/{platform-package-version}

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

    c. Examples:
    User-Agent: MyCompany/myProduct/62.310.4 eHealth/Technical/3.19.0
    User-Agent: Topaz-XXXX/123.23.X Taktik/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

7.2 API documentation

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 (**not available at this moment**)	https://portal.api.ehealth.fgov.be

Endpoint to call UHMEP is described in the API documentation on the eHealth API Portal.

Different operations are available and textually described in section 8.

8 Description of the operations

8.1 Common validations

For all incoming requests, UHMEP applies a suite of common validations if the request is validated by the implementation guide described in section 6.3.

8.1.1 Validation of dates

In the payload, different types of date are used.

For the execution dates, the format has to be YYYY-MM-DD.
For the recorded date (***authoredOn*** field), the date must have the format YYYY-MM-DDThh:mm:ss (for example : `2023-05-12T13:37:42+02:00`)

8.1.2 Validation of the template

UHMEP is using FHIR templates to exchange data between the applications. The templates follows INAMI's templates of referral prescriptions. The template must be known from UHMEP. If it is not the case, the user will receive an error.

Currently, for this version of the MVP, UHMEP supports templates :

- Medication prefill preparation (not refundable)
- Diabetes education via convention center
- Diabetes education with care model (preliminary path) "follow up of patients with diabetes type 2"
- Diabetes education with care path
- Education and self-care for diabetes patients without a care path
- Assisting with personal hygiene
- Chronic peritoneal dialysis

8.1.3 Validation of the prescriber and caregiver information

The existence of the healthcare professionals is checked in the eHealth authentic source CoBRHA. UHMEP will compare the SSIN and the discipline given in the payload or retrieved from the token with CoBRHA. If the healthcare professional is not found, the user will receive an error. If he is found, UHMEP will make other validations linked to the operation called.

For the performance and to avoid useless calls, UHMEP will firstly make the check digits verification to verify that the given SSIN is a valid one respecting the structure definition.

8.1.4 Validation on the patient

The existence of the patient will also be checked by consulting the national register through the eHealh ConsultRN service (thanks to his pseudonymized SSIN). If the patient is not known by the national register, the prescriber will not be able to create a prescription for him. ***This validation will be implemented for the release 1.***

For the performance and to avoid useless calls, UHMEP will firstly make the check digits verification to verify that the given SSIN is a valid one respecting the structure definition.

8.1.5 Authentication

Each user using UHMEP must be authenticated to be allowed to use UHMEP. UHMEP will verify the validity of the access token. UHMEP will also use the other information contained in the token to accept or reject the request. This behavior has been described in the section 6.2 talking about access management.

8.1.6 Visible prescriptions

8.1.6.1 Depending on the template

When a prescriber/caregiver can access an operation because he has the correct role, only the prescriptions that he is allowed to see are returned. To know if he is allowed to see the prescription, a check is done by looking the consultation right in the access matrix like it is described in section 6.2.3.

8.1.6.2 Blacklisted prescriptions

When a prescriber/caregiver can access an operation because he has the correct role, some prescriptions in the ***status*** "entered-in-error" with the ***extension.statusReason*** "blacklisted" will not be returned.

A prescription can be blacklisted at the creation because the prescriber was suspended (see the suspension explanation in section 6.2.4) but it can also be blacklisted after its creation thanks to a retro-action script done by UHMEP.

This script is launched when the information of the prescriber suspension is received only some days after the creation.

Depending on the actual status of the prescription, the blacklisting script will have different behaviors :

- If the prescription was in one of the statuses "draft", "ready", "cancelled", "expired" :
  - The prescription is blacklisted.
  - The prescription will not be returned (in case of a specific consultation, an error will be returned).
- If the prescription was in the status "inProgress":
  - The prescription is blacklisted.
  - The status of all executions in "in-progress" is changed to "interrupted".
  - The prescription will be returned.
- If the prescription was in one of the statuses "inactive", "completed" :
  - The prescription is blacklisted.
  - The prescription will be returned.

To summarize, when a prescription is in the ***status*** "entered-in-error" with the ***extension.statusReason*** "blacklisted", it will be returned if there is one or more executions. If there is nothing or only assignations on it, the prescription will not be returned.

8.2 Operations on prescriptions

8.2.1 Creation of a prescription

This method will allow the end user to create a referral prescription via his user interface in the UHMEP application.

8.2.1.1 Allowed roles

Prescriber.

8.2.1.2 Endpoint

POST /ServiceRequest

8.2.1.3 Implemented rules

There are different rules that this service has to check to allow the end user to be able to create a prescription :

- The prescriber is not suspended.
  - If he is suspended, an error will be returned but the prescription will be created with the ***status*** "entered-in-error" with the ***extension.statusReason*** "blacklisted".
- ***validity.start*** date can be maximum 5 days before the recorded date
  - 5 days is the current default value for all templates but it can be configured by INAMI for each template.
- ***validity.start*** =< ***validity.end*** =< ***latest***
- ***validity.end*** & ***latest*** > ***authoredOn***
- If an extend prescription is given (in the ***basedOn*** field), verification that the two prescriptions are for the same patient.

8.2.1.4 Request

The request must contain the information of the prescription in the body using the BeReferralPrescription resource.

Particularity coming from UHMEP :

Field	Particularity
intent	UHMEP will accept only the value "order".
basedOn	UHMEP will accept only a reference to an existing ServiceRequest.

8.2.1.5 Response

On successful creation of the BeReferralPrescription resource, the server will respond with a HTTP 201. The ID of this newly created resource is accessible in the *Location* header. This ID can be used to further interactions with this resource.

8.2.2 Cancel a prescription

This method allows the end-user to cancel a referral prescription via his user interface in the UHMEP application. It means that the end-user would be able to continue to consult it but the prescription will no longer be executable.

8.2.2.1 Allowed roles

Prescriber and Patient.

8.2.2.2 Endpoint

PATCH /ServiceRequest/{prescriptionID}

8.2.2.3 Implemented rules

The cancellation of a specific prescription is possible only if all the following points are respected :

- The prescription is in one of the combination of fields ***status*** and ***extension.statusReason*** :
  - "draft" – ""
  - "active" – "pending"
  - "active" – "ready"
- (Prescriber) A therapeutic link exists with the patient.
- (Prescriber) No exclusion exists with the patient if he is not the prescriber of the prescription.
- (Prescriber) He is not suspended.
- (Prescriber) He is the prescriber of the prescription.
- (Patient) The prescription is for the patient making the request.

8.2.2.4 Request

The operation is done through a PATCH on the BeReferralPrescription resource. It should only replace the ***status*** of this resource to the one "revoked". An example is available in Snippet 3 :

```
{
    "resourceType": "Parameters",
    "parameter": [
        {
            "name": "operation",
            "part": [
                {
                    "name": "type",
                    "valueCode": "replace"
                },
                {
                    "name": "path",
                    "valueString": "ServiceRequest.status"
                },
                {
                    "name": "value",
                    "valueString": "revoked"
                }
            ]
        }
    ]
}
```

Snippet 3 : PATCH example on a BeReferralPrescription to cancel a prescription

8.2.2.1 Response

On successful cancellation of the prescription, the server sends back an HTTP 200.

8.3 Assignation of a caregiver on a prescription

8.3.1 Add a caregiver

Once a prescription has been created, the system allows users to assign a specific caregiver on it. Giving them an ease of access to this prescription and some access rights.

8.3.1.1 Allowed roles

Prescriber and Patient.

8.3.1.2 Endpoint

POST /Task

8.3.1.3 Implemented rules

The assignation of a caregiver on an existing prescription is possible only if all following rules are respected :

- (Prescriber/Patient) The ***status*** of the BePerformerTask is “ready”.
- (Prescriber/Patient) No ***executionPeriod*** is provided.
- (Prescriber/Patient) The caregiver to assign is not currently suspended.
- (Prescriber/Patient) The caregiver to assign exists in CoBRHA database.
- (Prescriber/Patient) The mentioned prescription is not blacklisted.
- (Patient) The patient is the same as the one of the prescription.
- (Prescriber) A therapeutic link exists with the patient.
- (Prescriber) No exclusion exists between the prescriber and the patient.
- (Prescriber) The prescriber is not suspended.

8.3.1.4 Request

The assignation is done through the creation of a BePerformerTask resource.

For more information about how BePerformerTask are handled in UHMEP, see section 6.3.3.

Particularity coming from UHMEP :

Field	Particularity
basedOn	This is a reference to the prescription, the BeReferralPrescription. (ex: ServiceRequest/{referenceID})
partOf	This is a reference to a BeReferralTask, that currently has the same ID as the prescription. (ex: Task/{referenceID})
intent	UHMEP only allows “order” or “instance-order”

8.3.1.5 Response

On successful creation of the BePerfomerTask resource, the server will respond with a HTTP 201. The ID of this newly created resource is accessible in the *Location* header. This ID can be used for further interactions with this resource.

8.3.2 Remove a caregiver

The inverse operation is also possible to remove an assigned caregiver from a prescription.

8.3.2.1 Allowed roles

Prescriber and Patient

8.3.2.2 Endpoint

DELETE /Task/{executionID}

{executionID} being the ID of the BePerformerTask to be removed from a BeReferralPrescription.

8.3.2.3 Implemented rules

The removal of an assigned caregiver on an existing prescription is possible only if all following rules are respected :

- (Prescriber/Patient) The prescription referenced in the ***basedOn*** field is in one of the combination of fields ***status*** and ***extension.statusReason*** :
  - "draft" – ""
  - "active" – "pending"
  - "active" – "ready"
  - "active" – "in-progress"
  - "active" – "inactive"
- (Prescriber/Patient) The execution of the caregiver (BePerformerTask) is in ***status*** "ready".
- (Prescriber) The connected prescriber is the same as the requester of the prescription.
- (Prescriber) A therapeutic link exists with the patient.
- (Prescriber) No exclusion exists between the prescriber and the patient.
- (Prescriber) The prescriber is not suspended.
- (Patient) The patient is the same as the one of the prescription.

8.3.2.4 Request

In addition of the authorization and header described above, this request does not need a body.

8.3.2.5 Response

On successful deletion of the resource, the server sends back a HTTP 204. After this operation, the resource is definitively deleted and cannot be used anymore.

8.3.3 Reject an assignation

Once assigned to a prescription, the caregiver can reject it to send the information that he will not work on it.

8.3.3.1 Allowed roles

Caregiver.

8.3.3.2 Endpoint

PATCH /Task/{executionID}

{executionID} being the ID of the BePerformerTask to be updated from a BeReferralPrescription.

8.3.3.3 Implemented rules

The rejection of an assignation on an existing prescription is possible only if all following rules are respected :

- The caregiver is not suspended.
- The ***status*** of the BePerformerTask is “ready”.
- The information contained in the ***owner*** field must match with the information of the connected caregiver in the token.
- The ***extension.statusReason*** of the BeReferralPrescription referenced in the ***basedOn*** field is :
  - “ready”
  - “inProgress”
  - “pending”
  - “inactive”

8.3.3.4 Request

The operation is done through a PATCH on the BePerformerTask resource. It should only replace the status of this resource to the one “rejected”. An example is available in Snippet 4 :

```
{
    "resourceType": "Parameters",
    "parameter": [
        {
            "name": "operation",
            "part": [
                {
                    "name": "type",
                    "valueCode": "replace"
                },
                {
                    "name": "path",
                    "valueString": "Task.status"
                },
                {
                    "name": "value",
                    "valueString": "rejected"
                }
            ]
        }
    ]
}
```

Snippet 4 : PATCH example on a BePerformerTask to reject an assignation

8.3.3.5 Response

On successful rejection of the assignation, the server sends back a HTTP 200. After this rejection, no more operation is allowed on this resource.

8.4 Execution of a prescription

This section will describe the different operations that a caregiver is able to do on a prescription.

The different operations available are :

- Start
- Finish
- Interrupt
- Cancel

8.4.1 Start an execution

The start operation allows the caregiver to indicate to UHMEP that he works on the prescription. His execution will be in the ***status*** "in-progress".

There are two ways in UHMEP to start an execution :

- By creating a task
- By updating a task thanks to a PATCH method

For more information about BePerformerTasks handling by UHMEP, see section 6.3.3.

8.4.1.1 Start an execution through a creation of resource

If the caregiver is not assigned on the prescription, this method must be used to create the execution directly in the ***status*** "in-progress".

8.4.1.1.1 Allowed Role

Caregiver.

8.4.1.1.2 Endpoint

POST /Task

8.4.1.1.3 Implemented rules

The creation of an execution task is possible only if all the following points are respected :

- The caregiver is not suspended.
- The prescription referenced in the ***basedOn*** field is in one of the ***extension.statusReason*** :
  - "ready"
  - "inProgress"
  - "inactive"
  - "expired"
- No other execution task in "in-progress" exists for this caregiver (execution with the same reference in the ***partOf*** field).
- The task ***status*** is "ready".
- Only the ***executionPeriod.start*** date is provided for the execution dates.
- ***executionPeriod.start*** date is not in the future.
- There is no exclusion between the patient and the caregiver.

- The information contained in the ***owner*** field matches with the information of the connected caregiver in the token.

8.4.1.1.4 Request

In addition of the authorization and header described above, this request needs a body containing the information of the task to create.

The execution task is represented in FHIR by the Belgian profile of the Task resource called BePerformerTask (see section 6.3.3).

Particularity coming from UHMEP :

Field	Particularity
Intent	UHMEP will only accept “order” or “instance-order”.

8.4.1.1.5 Response

On successful creation of the BePerfomerTask resource, the server will respond with a HTTP 201. The ID of this newly created resource is accessible in the *Location* header. This ID can be used for further interactions with this resource.

8.4.1.2 Start an execution through a PATCH

If the caregiver is already assigned on the prescription, this method must be used to update the existing BePerformerTask (see section 6.3.3).

8.4.1.2.1 Allowed roles

Caregiver.

8.4.1.2.2 Endpoint

PATCH /Task/{executionID}

{executionID} being the ID of the BePerformerTask to be updated from a BeReferralPrescription.

8.4.1.2.3 Implemented rules

The update of an execution task is possible only if all the following points are respected :

- The caregiver is not suspended.
- The prescription referenced in the ***basedOn*** field is in one of the ***extension.statusReason*** :
  - "ready"
  - "inProgress"
  - "inactive"
  - "expired"
- The Task is in the ***status*** "in-progress".
- Only the ***executionPeriod.start*** is provided for the execution dates.
- ***executionPeriod.start*** date is not in the future.
- No exclusion exists between the patient and the caregiver.
- The information contained in the ***owner*** field must match with the information of the connected caregiver in the token.

8.4.1.2.4 Request

The operation is done through a PATCH on the BePerformerTask FHIR resource. It should only add the start execution date (YYYY-MM-DD) on this resource. An example is available below :

```
{
    "resourceType": "Parameters",
    "parameter": [
        {
            "name": "operation",
            "part": [
                {
                    "name": "type",
                    "valueCode": "add"
                },
                {
                    "name": "path",
                    "valueString": "Task.executionPeriod"
                },
                {
                    "name": "name",
                    "valueString": "start"
                },
                {
                    "name": "value",
                    "valueDateTime": "{{StartExecutionDate}}"
                }
            ]
        }
    ]
}
```

Snippet 5 : PATCH example on a BePerformerTask to start an execution

8.4.1.2.5 Response

On successful update of the execution, the server sends back a HTTP 200. The response does not contain a body.

8.4.2 Finish an execution

The finish operation allows the caregiver to indicate to UHMEP that he has finished his work on the prescription.
His execution will be in the ***status*** “completed”.

There are two ways in UHMEP to finish an execution :

- By creating a task
- By updating a task thanks to a PATCH method

8.4.2.1 Finish an execution through a creation of resource

If the caregiver is not assigned on the prescription, this method must be used to create the execution directly in the status “completed”.

8.4.2.1.1 Allowed roles

Caregiver.

8.4.2.1.2 Endpoint

POST /Task

8.4.2.1.3 Implemented rules

The creation of an execution task is possible only if all the following points are respected :

- The caregiver is not suspended.
- The prescription referenced in the ***basedOn*** field is in one of the ***extension.statusReason*** :
  - “ready”
  - “inProgress”
  - “inactive”
  - “expired”
- No other execution task in "in_progress" exists for this caregiver (execution with the same reference in the "partOf" field).
- The BePerformerTask ***status*** is "completed" and the ***executionPeriod.start*** and ***executionPeriod.end*** dates are provided for the execution dates.
- Execution dates are not in the future.
- ***executionPeriod.start*** <= ***executionPeriod.end***.
- No exclusion exists between the patient and the caregiver.
- The information contained in the ***owner*** field matches with the information of the connected caregiver in the token.

8.4.2.1.4 Request

In addition of the authorization and header described above, this request needs a body containing the information of the task to create.

The execution task is represented in FHIR by the Belgian profile of the task resource called BePerformerTask.

Particularity coming from UHMEP :

Field	Particularity
intent	UHMEP will only accept the value “order" or "instance-order".

8.4.2.1.5 Response

On successful creation of the BePerfomerTask resource, the server will respond with a HTTP 201. The ID of this newly created resource is accessible in the *Location* header. This ID can be used for further interactions with this resource.

8.4.2.2 Finish an execution through a PATCH

If the caregiver has already started an execution on the prescription, this method must be used to update the existing execution.

8.4.2.2.1 Allowed roles

Caregiver.

8.4.2.2.2 Endpoint

PATCH /Task/{executionID}

{executionID} being the ID of the BePerformerTask to be updated from a BeReferralPrescription.

8.4.2.2.3 Implemented rules

The update of an execution task is possible only if all the following points are respected :

- The caregiver is not suspended.
- The prescription referenced in the ***basedOn*** field is in one of the ***extension.statusReason*** :
  - "inProgress"
- The BePerformerTask is in the ***status*** "in-progress".
- Only the ***executionPeriod.end*** date is provided in the PATCH request for the execution dates.
- ***executionPeriod.end*** date is not in the future.
- ***executionPeriod.start*** <= ***executionPeriod.end***.
- There is no exclusion between the patient and the caregiver.
- The information contained in the ***owner*** field must match with the information of the connected caregiver in the token.

8.4.2.2.4 Request

The operation is done through a PATCH on the BePerformerTask FHIR resource. It should only add the end execution date (YYYY-MM-DD) on this resource. An example is available below :

```
{
    "resourceType": "Parameters",
    "parameter": [
        {
            "name": "operation",
            "part": [
                {
                    "name": "type",
                    "valueCode": "add"
                },
                {
                    "name": "path",
                    "valueString": "Task.executionPeriod"
                },
                {
                    "name": "name",
                    "valueString": "end"
                },
                {
                    "name": "value",
                    "valueDateTime": "{{EndExecutionDate}}"
                }
            ]
        }
    ]
}
```

Snippet 6 : PATCH example on a BePerformerTask to finish an execution

8.4.2.2.5 Response

On successful update of the execution, the server sends back a HTTP 200. The response does not contain a body.

8.4.3 Interruption of an execution

A caregiver can interrupt his execution on a prescription.

8.4.3.1 Allowed roles

Caregiver

8.4.3.2 Endpoint

PATCH /Task/{executionID}

{executionID} being the ID of the BePerformerTask to be updated from a BeReferralPrescription.

8.4.3.3 Implemented rules

The interruption of an execution on an existing prescription is possible only if all following rules are respected :

- The caregiver is not suspended
- The BePerformerTask is in the ***status*** "in-progress".
- The information contained in the ***owner*** field matches with the information of the connected caregiver in the token.
- The prescription ***extension.statusReason*** is "inProgress".

8.4.3.4 Request

The operation is done through a PATCH on the BePerformerTask FHIR resource. It should only replace the status of this resource to the one "on-hold". An example is available below :

```
{
    "resourceType": "Parameters",
    "parameter": [
        {
            "name": "operation",
            "part": [
                {
                    "name": "type",
                    "valueCode": "replace"
                },
                {
                    "name": "path",
                    "valueString": "Task.status"
                },
                {
                    "name": "value",
                    "valueString": "on-hold"
                }
            ]
        }
    ]
}
```

Snippet 7 : PATCH example on a BePerformerTask to interrupt an execution

8.4.3.5 Response

On successful interruption of the execution, the server sends back a HTTP 200. After this interruption, no more operation is allowed on this resource.

8.4.4 Cancel an execution

A caregiver can cancel an execution on a prescription.

8.4.4.1 Allowed roles

Caregiver.

8.4.4.2 Endpoint

PATCH /Task/{executionID}

{executionID} being the ID of the BePerformerTask to be updated from a BeReferralPrescription.

8.4.4.3 Implemented rules

The cancellation of an execution on an existing prescription is possible only if all following rules are respected :

- The caregiver is not suspended.
- The BePerformerTask is in the ***status*** "in-progress".
- The information contained in the ***owner*** field must match with the information of the connected caregiver in the token.
- The prescription ***extension.statusReason*** is "in-progress".

8.4.4.4 Request

The operation is done through a PATCH on the BePerformerTask resource. It should only replace the status of this resource to the one "cancelled". An example is available below :

```
{
    "resourceType": "Parameters",
    "parameter": [
        {
            "name": "operation",
            "part": [
                {
                    "name": "type",
                    "valueCode": "replace"
                },
                {
                    "name": "path",
                    "valueString": "Task.status"
                },
                {
                    "name": "value",
                    "valueString": "cancelled"
                }
            ]
        }
    ]
}
```

Snippet 8 : PATCH example on a BePerformerTask to cancel an execution

8.4.4.5 Response

On successful cancellation of the execution, the server sends back a HTTP 200. After this operation, the resource can be re-used by the caregiver.

8.5 Consultation of resources

8.5.1 Consultation of a BeReferralPrescription

This method allows the user to get all the information for one specific prescription by giving its reference ID.

8.5.1.1 Allowed roles

Prescriber, Caregiver and Patient.

8.5.1.2 Endpoint

`GET /ServiceRequest/{referenceId}`

8.5.1.3 Implemented rules

The consultation of a specific prescription is possible only if all the following points are respected :

- The referenceId given in the endpoint exists in UHMEP.
- The prescription is in another ***status*** than “entered-in-error” with ***extension.statusReason*** “blacklisted” or in the same state with some executions.
- (Prescriber/Caregiver) A therapeutic link exists with the patient.
- (Prescriber) No exclusion exists with the patient if he is not the prescriber of the prescription.
- (Caregiver) No exclusion exists with the patient if he does not have any BePerformerTask linked to this prescription.
- (Prescriber) He is not suspended if he is not the prescriber of the prescription.
- (Caregiver) He is not suspended if he does not have any BePerformerTask linked to this prescription.
- (Patient) The prescription is for the patient requesting the prescription.

In case of a suspension :

- A prescriber will only be able to see a prescription created by him.
- A caregiver will only be able to see a prescription on which he is assigned or he works.

8.5.1.4 Request

In addition of the authorization and header described above, this request does not need a body.

8.5.1.5 Response

On successful request, the response will contain the requested prescription.

This prescription is represented in FHIR by a Belgian profile of the ServiceRequest resource.

The Belgian profile used depends on the template of the prescription consulted :

Template	Profile
Nursing prescriptions	BeReferralPrescription

8.5.2 Consultation of a list of BeReferralPrescriptions

This method allows the user to get all the prescriptions corresponding to certain filters.

8.5.2.1 Allowed roles

Prescriber, Caregiver and Patient.

8.5.2.2 Endpoint

```
GET  /ServiceRequest?{filterName}={value}
```

Filter available	Value required
requester	Requester SSIN
performer	Performer SSIN
patient	Patient pseudonym (domain UHMEP)

These filters can be combined by putting a “&” symbol between those.

The possible combinations are described below :

Role required	Combination	Description
Prescriber	Requester	Consult prescriptions created by me or another prescriber
Prescriber	Requester + patient	Consult prescriptions created by me or another prescriber for a certain patient
Prescriber	Patient	Consult prescriptions for a certain patient
Caregiver	Performer	Consult prescriptions assigned to me
Caregiver	Performer + patient	Consult prescriptions assigned to me for a certain patient
Caregiver	Patient	Consult prescriptions for a certain patient
Patient	Patient	Consult my prescriptions

8.5.2.3 Implemented rules

The consultation of many prescriptions is possible only if all the following points are respected :

- (Prescriber/Caregiver) There is a therapeutic link with the patient.
- (Patient) The patient given in the URL is the same than the one connected.
- (Caregiver) The caregiver given in the URL is the same than the one connected.

The prescriptions corresponding to the filter(s) will be returned only if all the following points are respected :

- The prescription is in another ***status*** than "entered-in-error" with ***extension.statusReason*** "blacklisted" or in the same state with executions.
- (Prescriber) No exclusion exists with the patient if he is not the prescriber of the prescription.
- (Caregiver) No exclusion exists with the patient if he does not have any BePerformerTask linked to this prescription.

In case of a suspension :

- A prescriber will only be able to see prescriptions created by him.
- A caregiver will only be able to see prescriptions on which he is assigned or he works.

8.5.2.4 Request

In addition of the authorization and header described above, this request does not need a body.

8.5.2.5 Response

On successful request, the response will contain a list of prescriptions corresponding to the filter(s) used.

The list is represented in FHIR by a Bundle resource. The pagination is also managed with this resource.

Each prescription is represented in FHIR by a Belgian profile of the ServiceRequest resource.

The profile used depends on the template of the prescription consulted :

Template	Profile
Nursing prescriptions	BeReferralPrescription

8.5.3 Consultation of BeReferralTask and BePerformerTask

Alongside the direct consultation of FHIR resources using their specific IDs, it is also possible to do a search to receive Tasks linked on a BeReferralPrescription ID. Following the Figure 2, it is possible to consult all BeReferralTask and BePerformerTask linked to a BeReferralPrescription, but also do a filter to only get the BeReferralTask or only all BePerformerTask.

8.5.3.1 Consultation of all Tasks linked to a BeReferralPrescription

8.5.3.1.1 Allowed roles

Prescriber, Caregiver and Patient

8.5.3.1.2 Endpoint

GET /Task?based-on={BeReferralPrescriptionID}

8.5.3.1.3 Implemented rules

The consultation of a specific prescription is possible only if all rules described in section 8.5.1 are followed. Since Tasks linked to a BeReferralPrescription are part of the referral prescription, all same rules should also be applied.

8.5.3.1.4 Request

In addition of the authorization and header described above, this request does not need a body.

8.5.3.1.5 Response

A Bundle of type “searchset” will be returned on successful request. Entries will contain each BeReferralTask and BePerformerTask linked to the BeReferralPrescription

8.5.3.2 Consultation filter of Tasks linked to a BeReferralPrescription

8.5.3.2.1 Allowed roles

Prescriber, Caregiver and Patient

8.5.3.2.2 Endpoint

```
GET /Task?based-on={BeReferralPrescriptionID}&_profile={value}
```

Available “values” are described below :

value	Description
Profile/be-referral-task	Consult the BeReferralTask linked to the BeReferralPrescription
https://www.ehealth.fgov.be/standards/fhir/referral/StructureDefinition/be-referral-task	Consult the BeReferralTask linked to the BeReferralPrescription
Profile/be-performer-task	Consult all BePerformerTask linked to the BeReferralPrescription
https://www.ehealth.fgov.be/standards/fhir/referral/StructureDefinition/be-performer-task	Consult the BePerformerTask linked to the BeReferralPrescription

8.5.3.2.3 Implemented rules

The consultation of a specific prescription is possible only if all rules described in section 8.5.1 are followed. Since Tasks linked to a BeReferralPrescription are part of the referral prescriptions, all same rules should also be applied.

8.5.3.2.4 Request

In addition of the authorization and header described above, this request does not need a body.

8.5.3.2.5 Response

A Bundle of type “searchset” will be returned on successful request. Depending on the filter used, the Bundle will contain all BePerformerTask or the BeReferralTask linked to a BeReferralPrescription.

8.5.4 Consultation of a referral prescription

All clinical information are stocked in the BeReferralPrescription, but other resources are often needed to have a global information. Every user that has the right to consult a prescription (see section 8.5) will also be able to ask for a Bundle that contains every resource that represents the global prescription. A visualization of resources that constitute a referral prescription is in Figure 4.

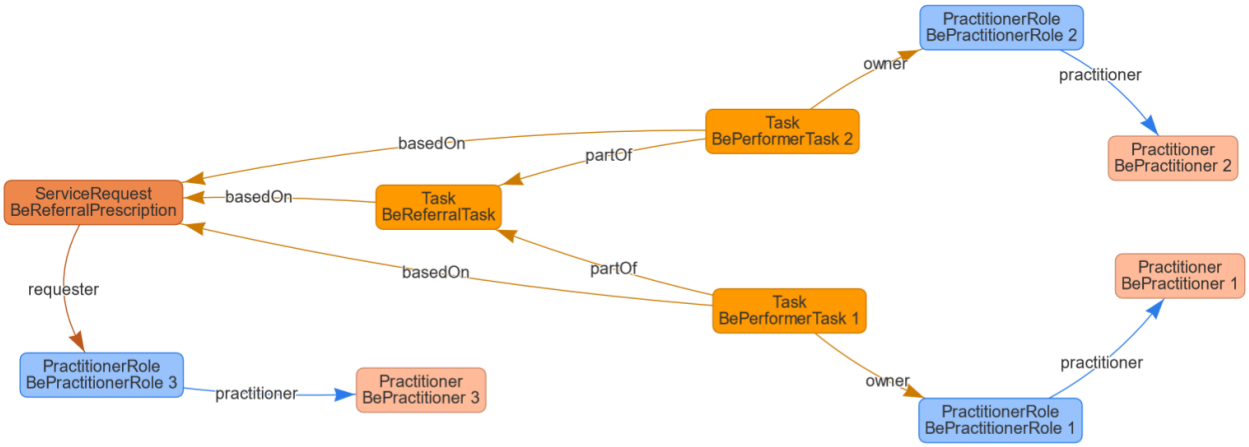


Figure 4 : Example of FHIR resources that compose a referral prescription with two different executions on it

Users will have the possibility to get all these resources, with as many BePerformerTask/BePractitionerRole/BePractitioner as needed, in one call.

The description of this new endpoint will be released in the next version of the cookbook.

9 Error management

In the cycle of prescription, UHMEP API allows different type of requests and multiple different actors can interact with the API. If there is an error in the request, or if the action is forbidden by the system, UHMEP sends back an error. This section documents how errors will be represented by UHMEP. Alongside the correct HTTP code, UHMEP uses the FHIR resource OperationOutcome which is defined by FHIR to display the error.

An error can appear for different reasons. The errors returned by UHMEP are structured in a specific way to allow the end-user to quickly understand where the errors come from.

UHMEP uses issues with fields :

- ***severity*** : defines if this is an “error” or a “warning”. (https://hl7.org/fhir/r4b/valueset-issue-severity.html)
- ***code*** : FHIR error type. UHMEP uses (business-rule, value, security, forbidden, conflict, processing). (https://hl7.org/fhir/r4b/valueset-issue-type.html)
- ***details.coding.system*** : The system where the error is defined. Errors thrown and defined by UHMEP are in the system “urn:uhmep:errors”
- ***details.coding.code*** : The code that defines the error. In the document, it is referenced as the **UHMEP code**
- ***diagnostics*** : Contains the description of the UHMEP code to help understanding. It should always be the same message for one UHMEP code. This message ends with an error id that can used for support purpose since this error is logged and more information can be retrieved by the UHMEP team through the error tracking system of UHMEP.

All the error messages returned by UHMEP are listed in the Excel document “UHMEP_ErrorCodes.xlsx” alongside this cookbook.