Controlled Medication Unified Platform

ePrescription Integration Quick Reference Guide

Company: شركة انسيبندنت هيلث لخدمات تكنولوجيا المعلومات (ذ.م.م) - انهيلث
Author: Andrius Šimėnas, Colin Bleach
Creation Date: 10/12/2017
Version: 0.3
Table of Contents

1 Introduction.......................................................................................................................... 3
1.1 Terms and concepts........................................................................................................... 3

2 Create ePrescription (MedicationRequest) ......................................................................... 4
2.1 Integration scenario........................................................................................................... 4
2.1.1 Create Prescription........................................................................................................ 4
2.1.2 Cancel Prescription ....................................................................................................... 8
2.1.3 Create Patient ............................................................................................................... 8

3 Dispense Medication integration scenario (MedicationDispense) ........................................ 10
3.1 Integration scenario.......................................................................................................... 10
3.1.1 Retrieve ePrescriptions ............................................................................................. 10
3.1.2 Lock the Prescription ................................................................................................. 13
3.1.3 Dispense Medication ................................................................................................. 14
3.1.4 Mark Prescription as dispensed ............................................................................... 15
3.1.5 Cancel Dispensed Prescription ............................................................................... 15

4 General information on System Integration Endpoints ...................................................... 16
4.1 Authentication ................................................................................................................. 17
4.1.1 OpenID Connect ........................................................................................................ 17
4.1.2 Authentication process ............................................................................................... 17
4.1.2.1 Requesting an access token for the External IS...................................................... 17
4.1.2.2 Querying user information from token ................................................................. 18
4.1.2.3 Querying data with access token ........................................................................ 18
4.2 Request and response types ............................................................................................ 19
4.2.1 Request to retrieve Resource (GET)............................................................................. 19
4.2.1.1 Single resource request ....................................................................................... 19
4.2.1.2 Multiple resource request .................................................................................... 21
4.2.2 Request to create Resource (POST) .......................................................................... 22
4.2.3 Request to update Resource (PUT).............................................................................. 24
4.3 Integration endpoint Request errors ................................................................................ 25
1 Introduction

This document describes main integration scenarios and message examples involved in creating Prescriptions for Controlled or Semi-Controlled Medications and in dispensing Medications.

Note: Current version of the document describes the intended and designed System integration interfaces, data structures and integration scenarios. During the design, testing, and other subsequent stages of the Project, this document can be refined in accordance with the actual Project implementation results.

1.1 Terms and concepts

The table below describes the terms used in the document and references to related documents and other sources.

<table>
<thead>
<tr>
<th>No</th>
<th>Term</th>
<th>Description / reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CNM</td>
<td>Controlled (Narcotic) or Semi-Controlled Medicine</td>
</tr>
<tr>
<td>3</td>
<td>System CNM</td>
<td>CNM ePrescription IS</td>
</tr>
<tr>
<td>4</td>
<td>External IS</td>
<td>Healthcare Institution IS or Pharmacy IS, that is integrating with the System (i.e., with CNM ePrescription IS)</td>
</tr>
</tbody>
</table>
2 Create ePrescription (MedicationRequest)

To create an ePrescription, following Integration Scenario has to be executed by the External IS.

2.1 Integration scenario

![Diagram of ePrescription integration scenario]

**Fig. 1 Create ePrescription integration scenario**

2.1.1 Create Prescription

External IS submits POST request to the System’s MedicationRequest endpoint:

```bash
# Submit data required to create a new MedicationRequest Resource:
POST {{SystemURL}}/MedicationRequest
```

Details of how to construct POST request are described in chapter 4.2.2.

MedicationRequest includes multiple references (links) to other Resources – Patient, Practitioner (Healthcare specialist who prescribed the Medicine), Organization (Healthcare institution, in which Practitioner works), Medication.

Note, that to reference other Resources, usage of “real life” identifiers, such as Emirates ID or Practitioner license number, is possible. In this case, reference is described via “?identifier=[value]” tag. Another option is to use System Resource FHIR id, in which case reference is described via “/[value]” syntax.

Example Body content of the POST request:
{  
  "resourceType": "MedicationRequest",  
  "intent": "order",  
  "status": "active",  
  "subject": {  
    "reference": "Patient?identifier=784197012345671",  
    "display": "Mohammed Khan"  
  },  
  "authoredOn": "2018-10-04T14:54:07.41+04:00",  
  "requester": {  
    "agent": {  
      "reference": "Practitioner?identifier=GN00000",  
      "display": "MOHAMMED ABDULAH"  
    },  
    "onBehalfOf": {  
      "reference": "Organization?identifier=PF1147",  
      "display": "TEST MEDICAL CENTER"  
    }  
  },  
  "reasonCode": [  
    {  
      "coding": [  
        {  
          "code": "F33.0",  
          "system": "http://hl7.org/fhir/sid/icd-10"  
        },  
        {  
          "code": "principal",  
          "system": "http://hl7.org/fhir/ex-diagnosistype"  
        }  
      ]  
    },  
    {  
      "coding": [  
        {  
          "code": "F33.0",  
          "system": "http://hl7.org/fhir/sid/icd-10"  
        }  
      ]  
    }  
  ]}
},
{
    "code": "differential",
    "system": "http://hl7.org/fhir/ex-diagnosistype"
}
],
"dispenseRequest": {
    "quantity": {
        "value": 1
    },
    "validityPeriod": {
        "end": "2018-10-06T14:54:07.41+04:00",
        "start": "2018-10-04T14:54:07.41+04:00"
    },
    "expectedSupplyDuration": {
        "unit": "days",
        "value": 5
    },
    "numberOfRepeatsAllowed": 1
},
"dosageInstruction": [
    {
        "patientInstruction": "One tablet every 4 hours",
        "route": {
            "text": "Oral Route"
        },
        "doseQuantity": {
            "unit": "mg",
            "value": 2
        },
        "timing": {
            "repeat": {
                "frequency": 2,
                "period": 1,
                "periodUnit": "d"
            }
        }
    }
]
In successful case, System returns reference to the newly created MedicationRequest Resource in the response header.

In case of an error, System will return OperationOutcome resource with an error message. For example, if Patient with given identifier was not found, following OperationOutcome will be returned:

```json
{
  "resourceType": "OperationOutcome",
  "issue": [
    {
      "severity": "error",
      "details": {
        "coding": [
          {
            "display": "Resource not found by Patient?identifier=7840000000000"
          }
        ]
      }
    }
  ]
}
```

Fig. 2 MedicationRequest example

Fig. 3 OperationOutcome example
2.1.2 Cancel Prescription

External IS submits POST request to the System’s MedicationRequest endpoint:

```
-- Submit data required to Cancel MedicationRequest Resource:

POST {{SystemURL}}/MedicationRequest/$cancel
```

Example Body content of the cancel MedicationRequest

```
{
   "resourceType": "Parameters",
   "parameter": [
      {
         "name": "request",
         "valueReference": {
            "reference": "MedicationRequest/18213"
        }
      },
      {
         "name": "requester",
         "valueReference": {
            "reference": "Practitioner/12"
        }
      }
   ]
}
```

2.1.3 Create Patient

External IS submits GET request to the System’s Patient endpoint:

```
-- Submit data required to create a new Patient Resource:

POST {{SystemURL}}/Patient
```

Example Body content of the POST request:

```
{
   "resourceType": "Patient",
   "name": [
      {
         "given": [
            "Abdul"
        ],
        "family": "Hussain"
      }
   ]
}
```
Fig. 4 Patient example

In the response header, System returns reference to the newly created Patient Resource.
3 Dispense Medication integration scenario (MedicationDispense)

3.1 Integration scenario

3.1.1 Retrieve ePrescriptions

To retrieve list of Patient’s active ePrescriptions, External HIS must submit following GET query with the patients EmiratesId or Passport and Issuing country:

```
-- Find all active Prescriptions issued to Patient with Emirates ID 7840000000000 :
GET {{SystemURL}}/MedicationRequest?subject:Patient.identifier=7840000000000 &status=active
```

```
-- Find all active Prescriptions issued to Patient with Passport and issuing country PD1234AE :
GET {{SystemURL}}/MedicationRequest?subject:Patient.identifier=PD1234AE &status=active
```

System returns all the Prescriptions in the Resource Bundle:

```
{
  "resourceType": "Bundle",
```
"code": "H02.234"

"system": "http://hl7.org/fhir/ex-diagnostics-type",
"code": "principal"
}
]

"coding": [

{"system": "http://hl7.org/fhir/sid/icd-10",
"code": "I69.334"
},

{"system": "http://hl7.org/fhir/ex-diagnostics-type",
"code": "differential"
}
]

"dosageInstruction": [

{"patientInstruction": "One tablet every 4 hours",
"route": {
"text": "test"
}
}

"dispenseRequest": {
"validityPeriod": {
"start": "2018-03-25T14:54:07.41+04:00",
"end": "2018-12-25T14:54:07.46+04:00"
},
"numberOfRepeatsAllowed": 1,
"quantity": {
"value": 1,
"unit": "tablet"
},
"expectedSupplyDuration": {
"value": 5,
"unit": "days"
}

"request": {
"method": "POST"
}
]
3.1.2 Lock the Prescription

To ensure that no Practitioner will simultaneously edit the Prescription while Pharmacist is dispensing medicine, External IS puts a lock on the Prescription by submitting POST request to the System’s Provenance endpoint:

```
-- Submit data required to create a new Provenance Resource:
POST {{SystemURL}}/Provenance
```

Lock (Provenance) will also ensure that any other Pharmacist cannot simultaneously dispense this Prescription.

```
{
  "resourceType": "Provenance",
  "target": [
    {
      "reference": "MedicationRequest/8415"
    }
  ],
  "recorded": "2018-02-15T08:00:00+02:00",
  "period": {
    "start": "2018-02-15T08:00:00+02:00",
    "end": "2018-02-15T09:00:00+02:00"
  },
  "agent": [
    {
      "whoReference": {
        "reference": "Practitioner?identifier=DHA-P-12319831"
      },
      "onBehalfOfReference": {
        "reference": "Organization?identifier=PF0000"
      }
    }
  ]
}
```

Fig. 7 Provenance example
3.1.3 Dispense Medication

External IS submits POST request to the System’s MedicationDispense endpoint:

```plaintext
-- Submit data required to create a new MedicationDispense Resource:
POST {{SystemURL}}/MedicationDispense
```

Similarly, to MedicationRequest, MedicationDispense includes multiple references (links) to other Resources.

```json
{
  "resourceType": "MedicationDispense",
  "whenHandedOver": "2018-01-21T16:30:07.41+04:00",
  "medicationReference": {
    "reference": "Medication?code=1338-1001-001"
  },
  "authorizingPrescription": [
    {
      "reference": "MedicationRequest/8370"
    }
  ],
  "note": [
    {
      "text": "One tablet every day before the sleep."
    }
  ],
  "status": "completed",
  "subject": {
    "display": "Abdul Hussain",
    "reference": "Patient?identifier=7840000000000"
  },
  "quantity": {
    "unit": "tablet",
    "value": 30
  },
  "receiver": [
    {
      "display": "Mohammed Khan",
      "reference": "Patient?identifier=7840000000001"
    }
  ],
  "performer": [
    {
      "actor": {
        "display": "Dr Pharm",
        "reference": "Practitioner?identifier=PS1001"
      },
      "onBehalfOf": {
        "display": "Pharmacy Center UAE",
        "reference": "Organization?identifier=PH2001"
      }
    }
  ]
}
```
Fig. 8 Example of the MedicationDispense request

In case of successful MedicationDispense request, System will return reference to the created MedicationDispense Resource and the System will update linked MedicationRequest marking it as completed.

3.1.4 Mark Prescription as dispensed

External IS submits POST request to the System’s MedicationRequest endpoint:

```bash
-- Submit data required to create a new MedicationRequest Resource:
POST {{SystemURL}}/MedicationRequest
```

Body content for the POST request should be the same that was retrieved in step 3.1.1, with status changed to “completed”:

```json
{
  "resourceType": "MedicationRequest",
  ...
  "status": "completed"
  ...
}
```

3.1.5 Cancel Dispensed Prescription

External IS submits POST request to the System’s MedicationDispense endpoint:

```bash
-- Submit data required to create a new MedicationDispense Resource:
POST {{SystemURL}}/MedicationDispense/$cancel
```

Example Body content of the cancel MedicationDispense
4 General information on System Integration Endpoints

To retrieve data from / submit data to the System, External IS needs to access System Integration Endpoints. Integration Endpoints are presented in the REST architecture. Via Integration Endpoints, data can be sent and received (as HL7 FHIR resources) in XML format (UTF-8 encoding) using HTTP POST, PUT and GET queries. HTTP requests will be sent to a specific Endpoint's URL. /

Endpoint URL addresses and parameters for all requests are case-sensitive, i.e., request to „Encounter“ integration endpoint will not be executed, if requester tries to address „encounter“ or „ENCOUNTER“ in the URL.
Authentication, System Integration Endpoints and the search parameters are described below.

4.1 Authentication

To authenticate and/or authorize the users (i.e., External IS) OAuth and OpenID Connect is used.

4.1.1 OpenID Connect

OpenID Connect 1.0 is a simple identity layer on top of the OAuth 2.0 protocol. It allows clients to verify the identity of the End-User based on the authentication performed by an Authorization Server, as well as to obtain basic profile information about the End-User in an interoperable and REST-like manner.

OpenID Connect is not to be confused with OpenId 1.0 or 2.0, which are not based on OAuth.

The protocol uses several other IETF specifications including JWT (http://jwt.io/).

OpenID Connect server is responsible for client's management, token's granting and validations.

4.1.2 Authentication process

OpenID Connect server should provide client-credential flow for server-to-server integration between applications. Every server application will have own client identifier and secret.

4.1.2.1 Requesting an access token for the External IS

In order for an External IS to get a System access token, an OAuth 2.0 request to the token endpoint of the OpenId Server needs to be made. To compose the token request following details are needed:

- The token endpoint URL of the OpenId server.
- The client ID and secret to compose the HTTP basic authentication header. Please contact System administrator to get External IS credentials.
- The requested scope values. If the scope parameter is omitted, the server will treat this as a request for all registered scope values for the External IS.

```plaintext
-- Request token with Username and password are combined into a string "username:password" and encoded using BASE64
POST https://erxqa.inhealth.ae/cas/oidc/token HTTP/1.1
Authorization: Basic aABCDdds010s
Content-Type: application/x-www-form-urlencoded
grant_type=client_credentials
```
If the client authentication has failed, the OpenId Server will return an HTTP error 40X, and no access token will be issued.

### 4.1.2.2 Querying user information from token

Use returned access token as Bearer token in authorization header for authentication.

```
GET https://erxqa.inhealth.ae/cas/oidc/profile
Authorization: Bearer AT-2-2-C48VB7P7kNSDUt852n_Rk1_fWhq_-u
```

Fig. 11 User information request example

```
{
  "sub": "test",
  "auth_time": 1213766936,
  "app": "ExternalISName123"
}
```

Fig. 12 User information response example

### 4.1.2.3 Querying data with access token

Use returned access token as Bearer token in authorization header for authentication and id token for identity.

```
GET {{SystemURL}}/Organization/123
Authorization: Bearer eyJraWQiOiJyc2ExliwiYWxiSDSYWxlci0NSTVRlc3QiLCJ....
```

Fig. 13 Data access token request example

JWT token will be used as Bearer token.
4.2 Request and response types

4.2.1 Request to retrieve Resource (GET)

In order to receive Resource data from the System, External IS has to submit GET request to corresponding REST integration point.

4.2.1.1 Single resource request

GET request for a particular Resource with its FHIR identifier, for example

GET {{SystemURL}}/Practitioner/123

where 123 is FHIR identifier of the Practitioner that External IS wants to retrieve. Response to such request is single Resource with given FHIR identifier:

```json
{
  "name": [
    {
      "text": "JOHN WILLIAM SMITH",
      "given": [
        "JOHN",
        "WILLIAM"
      ],
      "family": "SMITH"
    }
  ],
  "active": true,
  "gender": "male",
  "telecom": [
    {
      "use": "work",
      "rank": 1,
      "value": "john.w.smith@inhealth.test.ae",
      "system": "email"
    },
    {
      "use": "work",
      "rank": 1,
      "value": "+001122334455",
      "system": "phone"
    }
  ],
  "identifier": [
```
Note, that FHIR identifier is not part of the Response body. FHIR identifier is provided in the response’s header parameter content-location (together with Resource’s actual version):

```
... content-location → {{SystemURL}}/Practitioner/123/_history/1
...
```

In the event of an error, the reply header will list the error status HTML code and the FHIR OperationOutcome resource describing the error. For example, when querying for a non-existing Patient resource, 404 Not Found code and OperationOutcome resource as shown in Error! Reference source not found. will be returned:

```
{
  "resourceType": "OperationOutcome",
  "issue": [
    {
      "severity": "error",
      "details": {
        "coding": [
          {
```

Fig. 15 Single resource request response
4.2.1.2 Multiple resource request

If GET request results in multiple Resources that satisfy search criteria, multiple Resources will be returned in the Response. For example, in order to search for a Patient that has a name John, External IS has to submit GET request:

GET {{SystemURL}}/Patient?name=John

Or in order to get all of the Medications, External IS has to submit GET request:

GET {{SystemURL}}/Medication

Such requests may often return multiple Resources, all of them satisfying given search criteria. Resource Bundle is returned in this case, returning the total count of Resources and paging information. By default, 10 Resources are returned in the single page.

```json
{
  "resourceType": "Bundle",
  "total": 5441,
  "link": [
    {"relation": "self",
     "url": "/Medication?page=1"},
    {"relation": "first",
     "url": "/Medication?page=1"},
    {"relation": "last",
     "url": "/Medication?page=545"},
    {"relation": "next",
     "url": "/Medication?page=2"},
  ],
  "entry": [
    {"id": "2915",
     "resource": {
```

Fig. 16 Response to the request for a non-existing resource
In case of no Resources found, Resource Bundle with zero Resources is returned:

```
{
  "resourceType": "Bundle",
  "total": 0,
  "link": [
    {
      "relation": "self",
      "url": "/Patient?name=Abcd&_page=1"
    },
    {
      "relation": "first",
      "url": "/Patient?name=Abcd&_page=1"
    },
    {
      "relation": "last",
      "url": "/Patient?name=Abcd&_page=1"
    }
  ]
}
```

**Fig. 18 Empty Response to a Multiple resource request**

### 4.2.2 Request to create Resource (POST)

POST requests are used to create Resources. The FHIR resource being transmitted by the POST request to the corresponding integration endpoint is described in the Body part of the request. For example, in order to create a new Patient resource, it is necessary to perform a POST request to {{SystemURL}}/Patient integration endpoint. Example request:
POST {{SystemURL}}/Patient

Headers:

Content-Type = application/json
Authorization = Bearer ...

Body:

{  "name": [  {
      "given": [
        "Mohammed", "Abdul"
      ],
      "family": "Khan"
    },
    "active": true,
    "gender": "male",
    "birthDate": "1970-01-01",
    "identifier": [
      {
        "use": "official",
        "value": "78400000000000000000",
        "system": "ae.inhealth.fhir.EmiratesId"
      }
    ],
    "resourceType": "Patient",
    "telecom": [
      {
        "system": "phone",
        "value": "00123456789",
        "use": "work",
        "rank":1
      },
      {
        "system": "email",
        "value": "test@inhealth.test.ae",
        "use": "work",
        "rank":1
      }
    ]
  }
}

Fig. 19 POST request to create Patient resource example

If the POST request was processed successfully, HTML response code 201 Created is returned, and the reference (FHIR identifier and version) to created resource’s URL is returned in the response header’s parameter Location:

...  
  location → {{SystemURL}}/Patient/8362/_history/1
In case of unsuccessful processing of the POST request, HTTP status 400 Bad Request is returned, and error description is provided in the Response Body.

**4.2.3 Request to update Resource (PUT)**

PUT requests are used in two scenarios:

- To update existing Resource with a new version of the Resource data
- To create a new Resource, when External IS wants to specify particular FHIR identifier that should be assigned to the newly created Resource.

PUT request always requires Resource FHIR identifier to be provided in the request URL:

```plaintext
PUT {{SystemURL}}/Patient/1000
```

In the example above, System will update Patient, whose FHIR identifier is 1000, with the Resource data that is provided in the Request body. However, if Patient with FHIR identifier does not exist in the System yet, System will create a new Patient Resource using the data provided in the request Body.

Difference between POST and PUT in the new Resource creation scenario:

- POST request – new Resource’s FHIR identifier is generated by the System automatically
- PUT request – new Resource’s FHIR identifier is defined by the External IS

If the PUT request was processed successfully, HTTP response code 200 OK is returned, and the reference (FHIR identifier and version) to updated resource’s URL is returned in the response header’s parameter Location:

```plaintext
... content-location → {{SystemURL}}/Patient/8362/_history/2 ...
```

Note that if there was no existing Resource with a FHIR identifier that was provided in the PUT request, new Resource will be created. However, the HTTP response code will still be 200 OK (and not 201 Created, as in POST request case). Content-location parameter in the Response header will point to the version 1:

```plaintext
... content-location → {{SystemURL}}/Patient/9999/_history/1 ...
```
4.3 Integration endpoint Request errors

Possible Error codes that can be returned as a response to the GET/POST/PUT requests to the integration points are described below:

Table 1 HTTP errors that the System returns

<table>
<thead>
<tr>
<th>Request type</th>
<th>Response error code</th>
<th>Response error message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>404</td>
<td>Resource Not Found</td>
<td>This error is returned when GET request is performed for a non-existing resource (i.e., incorrect resource identifier or version number).</td>
</tr>
<tr>
<td>GET</td>
<td>500</td>
<td>Server Error</td>
<td>This error is returned when GET request contains incorrect search parameters. Incorrect parameter is then described in the error message. I.e., field “dummy” does not exist in Patient.</td>
</tr>
<tr>
<td>POST/PUT</td>
<td>400</td>
<td>Bad Request - Failed to process message</td>
<td>This error is returned when provided Resource does not conform to the required FHIR resource structure (i.e., errors in XML formatting etc.).</td>
</tr>
</tbody>
</table>