

CIBMTR Direct FHIR Service API User Guide

v.2101

Introduction

CIBMTR collects clinical research data related to stem-cell transplants including patient characteristics, disease parameters, procedures, treatments, and longitudinal outcomes. Typically, this data is collected using an online form called FormsNet and populated by a data manager associated with a transplant center or hospital. CIBMTR is committed to minimizing the data collection burden on transplant center data managers and is actively working to collect data electronically from the transplant center Electronic Health Records systems. Electronic data is used to prepopulate the questions on the applicable CIBMTR forms.

This document describes how to submit FHIR data electronically using available CIBMTR REST APIs. Data for each patient is submitted using the HL7 FHIR exchange protocol in JSON or XML format. The REST APIs are available for integration into a custom client architecture or for submission using a manual client such as Postman.

The current API supports FHIR STU3 resources. In general, FHIR resources should conform to the US Core Implementation Guide and associated profiles. The Patient FHIR resource is a special exception in that CIBMTR removes any Personally Identifiable Information (PII) from the Patient resource. The Direct FHIR Service API can accept multiple different FHIR resource types, however, only lab data for form 2402 is implemented to fully prepopulate questions on a form. Submitting lab data for form 2402 only requires implementation of the Observation FHIR resource. Additional data types and form questions will be supported in CY '21.

The process for submitting production data to CIBMTR includes three sequential steps:

1. Request CIBMTR Direct FHIR Service API Access Credentials
2. Submit test data using the CIBMTR test API endpoint URLs
3. Submit production data with the CIBMTR production API endpoint URLs

Once electronic data has been submitted via the Direct FHIR Service API, the Data Manager can login to the FormsNet interface to complete the form. Within the form, there will be clarifying contextual questions that provide important information for associating the dates of the labs with the key dates of interest on the form.

Access Credentials

The Your CIBMTR relationship manager or technical lead can initiate a request for API credentials. CIBMTR uses OAuth2.0/OpenID (OIDC) for authentication and access management. This process involves making a request to a third-party authorization server to receive a token. The token is then

passed to the CIBMTR API URL in the request header. The following information will be provided by CIBMTR and is necessary for requesting an authorization token¹:

- CIBMTR Service Account Username
- CIBMTR Service Account Password
- Application Client ID
- Application Client Secret
- Application Scope

Different sets of credentials will be provided for the CIBMTR test and production environments.

To request an authentication token for the test environment, the third-party authorization server URL is:

```
POST https://nmdp.oktapreview.com/oauth2/ausaexcazhLhxKnJs0h7/v1/token
```

or:

```
POST https://nmdp.oktapreview.com/oauth2/aus3ck6q30qmOdpMb1t7/v1/token
```

To request an authentication token for the production environment, the third-party authorization server URL is:

```
POST https://nmdp.okta.com/oauth2/ausaexcazhLhxKnJs0h7/v1/token
```

or:

```
POST https://nmdp.okta.com/oauth2/aus3ck6q30qmOdpMb1t7/v1/token
```

The header of the POST request to the authorization server must have an authorization string. The string is constructed by base64 encoding the Application Client ID, a colon, and the Application Client Secret. The encoded string is then appended to the word "Basic".

```
const auth_string = "Basic " +  
    base64("<Application Client ID>" + ":" + "<Application Client Secret>")
```

An example of the header parameters for the POST request to the authorization server is shown in Figure 1. In the figure, the authorization string is blacked out. Notice the space between the base 64 encoded string and the string prefix, "Basic".

¹ <https://developer.okta.com/docs/guides/implement-password/use-flow/>

POST	https://nmdp.oktapreview.com/oauth2/ausaexcazhLhxKnJs0h7/v1/token	
Params	Authorization ●	Headers (12)
<div> <div>Hide auto-generated headers</div> </div>		
	KEY	VALUE
<input checked="" type="checkbox"/>	Authorization ⓘ	Basic [REDACTED]
<input checked="" type="checkbox"/>	Cookie ⓘ	JSESSIONID=28CF59DB205DF0F8D816724F0FCAA3BF
<input checked="" type="checkbox"/>	Postman-Token ⓘ	<calculated when request is sent>
<input checked="" type="checkbox"/>	Content-Type ⓘ	application/x-www-form-urlencoded
<input checked="" type="checkbox"/>	Content-Length ⓘ	<calculated when request is sent>
<input checked="" type="checkbox"/>	Host ⓘ	<calculated when request is sent>
<input checked="" type="checkbox"/>	User-Agent ⓘ	PostmanRuntime/7.26.8
<input checked="" type="checkbox"/>	Accept ⓘ	*/*
<input checked="" type="checkbox"/>	Accept-Encoding ⓘ	gzip, deflate, br
<input checked="" type="checkbox"/>	Connection ⓘ	keep-alive
<input checked="" type="checkbox"/>	Accept	application/json
<input checked="" type="checkbox"/>	Content-Type	application/x-www-form-urlencoded

Figure 1: Example header information for the POST request to the authorization server

Figure 2 shows the required fields in the body of the POST request to the authorization server API. The value for the “username” key should be the CIBMTR Service Account Username provided by CIBMTR. The value for the “password” key should be the CIBMTR Service Account Password. The “grant_type” key and the “scope” key should have the same values as shown in Figure 2. The response to the POST request will return a JSON object that includes a base64 encoded token. The token can be a long character string (over 1000 chars).

POST ▼ <https://nmdp.oktapreview.com/oauth2/ausaexcazhLhxKnjs0h7/v1/token>

Params Authorization ● Headers (12) ● **Body ●** Pre-request Script Tests

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL

	KEY	VALUE
<input checked="" type="checkbox"/>	grant_type	password
<input checked="" type="checkbox"/>	scope	api_cibmtr_fhir_ehr_client
<input checked="" type="checkbox"/>	username	[REDACTED]
<input checked="" type="checkbox"/>	password	[REDACTED]

Figure 2: Required POST fields to submit for the authorization token.

Once the token has been received, a request to the CIBMTR Direct FHIR Service API can be made. Tokens are valid for 30 minutes in the production environment, but last up to 24 hours in the test environment. **Applications must cache and re-use tokens until they are about to expire because Okta rate limits requests for new tokens.** One workable strategy is to obtain a new token every 25 minutes.

To make a request to the CIBMTR Direct FHIR Backend API, include the token in the header as the authorization key value of the request along with the word “Bearer ” in front of it, as shown in shown in Figure 3.

POST ▼ <https://dev-api.nmdp.org/cibmtrclientbackendtest/v1/Patient>

Params Authorization ● **Headers (10) ●** Body ● Pre-request Script ● Tests Settings

Headers ⚙ Hide auto-generated headers

	KEY	VALUE
<input checked="" type="checkbox"/>	Authorization ①	Bearer eyJraWQIOjNQ1ZEWjFqemRDbEFURTd2eDhweThwYjR2...
<input checked="" type="checkbox"/>	Cookie ①	f5avraaaaaaaaaaaaaa_session_=KFOBCEJCFGMMBCCDKEGH...
<input checked="" type="checkbox"/>	Postman-Token ①	<calculated when request is sent>
<input checked="" type="checkbox"/>	Content-Type ①	application/json
<input checked="" type="checkbox"/>	Content-Length ①	<calculated when request is sent>
<input checked="" type="checkbox"/>	Host ①	<calculated when request is sent>
<input checked="" type="checkbox"/>	User-Agent ①	PostmanRuntime/7.26.8
<input checked="" type="checkbox"/>	Accept ①	*/*
<input checked="" type="checkbox"/>	Accept-Encoding ①	gzip, deflate, br
<input checked="" type="checkbox"/>	Connection ①	keep-alive

Figure 3: Example CIBMTR Direct FHIR API request using a bearer authorization token in the header of the request.

Recommended Data Submission Workflow

Submitting data to CIBMTR via the Direct FHIR Service API involves a three-step process for each patient:

Step 1: Register patient and receive CRID

Register a patient with CIBMTR and receive a CIBMTR Research Identifier (CRID) to use as a patient reference for all subsequent FHIR data submissions. CIBMTR exposes a special service API to handle the submission of personally identifiable information (PII). Data submitted via the CRID Service API has special protections and exposure within CIBMTR to avoid unnecessary handling of PII. For all subsequent FHIR data submissions, the CRID is used to identify the patient and any PII is removed from FHIR resources before being stored on CIBMTR FHIR servers.

The CRID Service API uses a PUT request at the following case-sensitive endpoint URLs:

Test Environment

PUT https://dev-api.nmdp.org/cibmtrehrclientbackendexttest/v1/CRID

Production Environment

PUT https://api.nmdp.org/cibmtrehrclientbackend/v1/CRID

The authorization key and bearer token must be included in the request as mentioned in the previous section. For the body of the PUT request, the following data fields are requested:

Five required attributes

- CCN (5digit)
- First and last name
- Birthdate (YYYY-MM-DD)
- Gender (M/F)

Optional attributes (possibly present)

- SSN (###-###-####)
- Mother's maiden name
- Race (race code)
- Ethnicity (ethnicity code)
- NMDP RID
- EBMT CIC + ID
- CIBMTR Team + IUBMID

Complete list of payload options for CRID registration

```
{
  "ccn": "string",
  "patient": {
    "firstName": "string",
    "lastName": "string",
    "birthDate": "string",
    "gender": "string",
    "ssn": "string",
    "mothersMaidenName": "string",
    "race": ["string"],
    "ethnicity": "string",
    "nmdpRid": 0,
    "ebmtCic": "string",
    "cibmtrIubmid": "string",
    "cibmtrTeam": 0,
    "ebmtId": "string"
  }
}
```

CRID Race Codes

Race Value Code	Description
1002-5	American Indian or Alaska Native
2028-9	Asian
2054-5	Black or African American
2076-8	Native Hawaiian or Other Pacific Islander
2106-3	White
ASKU	Not Reported
UNK	Unknown

CRID Ethnicity Codes

Ethnicity Value Code	Description
2135-2	Hispanic or Latino
2186-5	Non Hispanic or Latino
UNK	Unknown

The CRID Service API is available as a PUT request. Consequently, submitting the same data twice does not re-register the patient, but rather will retrieve the same CRID number registered previously. The

CRID Service API will attempt to perform partial “fuzzy” matches based on data submitted to avoid re-registering the same patient with two different CRID numbers.

The response payload of the CRID Service API is a JSON object that contains the CRID number (lower pane in Figure 4). The CRID number is then used for all other data references to the registered patient.

The screenshot displays a REST client interface with two panes. The top pane shows a PUT request to the URL `https://dev-api.nmdp.org/cibmtrehrclientbackendxttest/v1/CRID`. The request body is a JSON object representing a patient's information. The bottom pane shows the response body, which is a JSON object containing match details and a generated CRID number.

Request:

```
1 {
2   "ccn": "12001",
3   "patient": {
4     "firstName": "Steve",
5     "lastName": "Rogers",
6     "birthDate": "1925-07-04",
7     "gender": "M",
8     "ssn": "098-76-5432",
9     "race": ["2106-3"],
10    "ethnicity": "UNK"
11  }
12 }
```

Response:

```
1 {
2   "perfectMatch": [
3     {
4       "matchedCriteria": [
5         "SSN",
6         "Gender",
7         "DateOfBirth"
8       ],
9       "matchType": "Perfect1",
10      "crid": 3989144
11    }
12  ]
13 }
```

Figure 4: Example CRID registration PUT request with JSON body payload (top pane) and response payload (bottom pane)

Step 2: Submit Patient FHIR Resource

The Patient FHIR resource must be submitted before any other FHIR resources². The Patient FHIR resource ID is part of the response to the Patient POST request. The resource ID is unique to the CIBMTR FHIR server and is used to reference the Patient subject on all subsequently submitted FHIR resources. The resource ID is assigned by the FHIR server and is different from the Patient.identifier section of the FHIR resource.

The Direct FHIR Service API uses a POST request to submit a Patient resource at the following case-sensitive endpoint URLs:

Test Environment

POST https://dev-api.nmdp.org/cibmtrehrclientbackendexttest/v1/Patient
--

Production Environment

POST https://api.nmdp.org/cibmtrehrclientbackend/v1/Patient
--

The authorization key and bearer token must be included in the request as mentioned in the previous section. FHIR JSON submissions should also include a “content-type” key in the header with value: “application/fhir+json”.

The Patient FHIR resource usually contains the demographics data for the patient, however, since the demographics data is already submitted during the CRID registration process, there are only two primary components necessary in the Patient FHIR resource:

1. A security label within the “meta” section of the Patient resource must contain the CIBMTR Center Number (CCN) prepended with “rc_” and associated with the codesystem as shown in Figure 5.
2. A CRID identifier reference within the “identifier” section of the Patient resource as shown in Figure 5.

PII information should be avoided as part of the Patient resource. However, the Direct FHIR Service API will remove PII information, including any that might be contained in “text” or other sections of the resource before storing it on the FHIR server.

The response after submitting a Patient resource request, indicates how the Patient resource is represented on the CIBMTR FHIR server (see Figure 6). The response JSON object has an “id” that is used to reference the Patient resource (circled in red in Figure 6). There is a new security label that

² <http://hl7.org/fhir/STU3/patient.html>

indicates that the resource has been through a redaction process to remove any PII that may have been submitted with the Patient resource.

The Patient resource ID is necessary for submitting other FHIR resources to the Direct FHIR Service API, but if the ID for a Patient resource previously submitted is not known, the following GET request can be submitted to the API to retrieve the Patient resource for a given CRID:

```
GET https://dev-api.nmdp.org/cibmtrehrclientbackendexttest/v1/Patient?
_security=http://cibmtr.org/codesystem/transplant-center%7Crc_<CCN>&
identifier=<CRID>
```

Where the <CCN> should be replaced with the associated CCN and the <CRID> replaced with the CRID number.

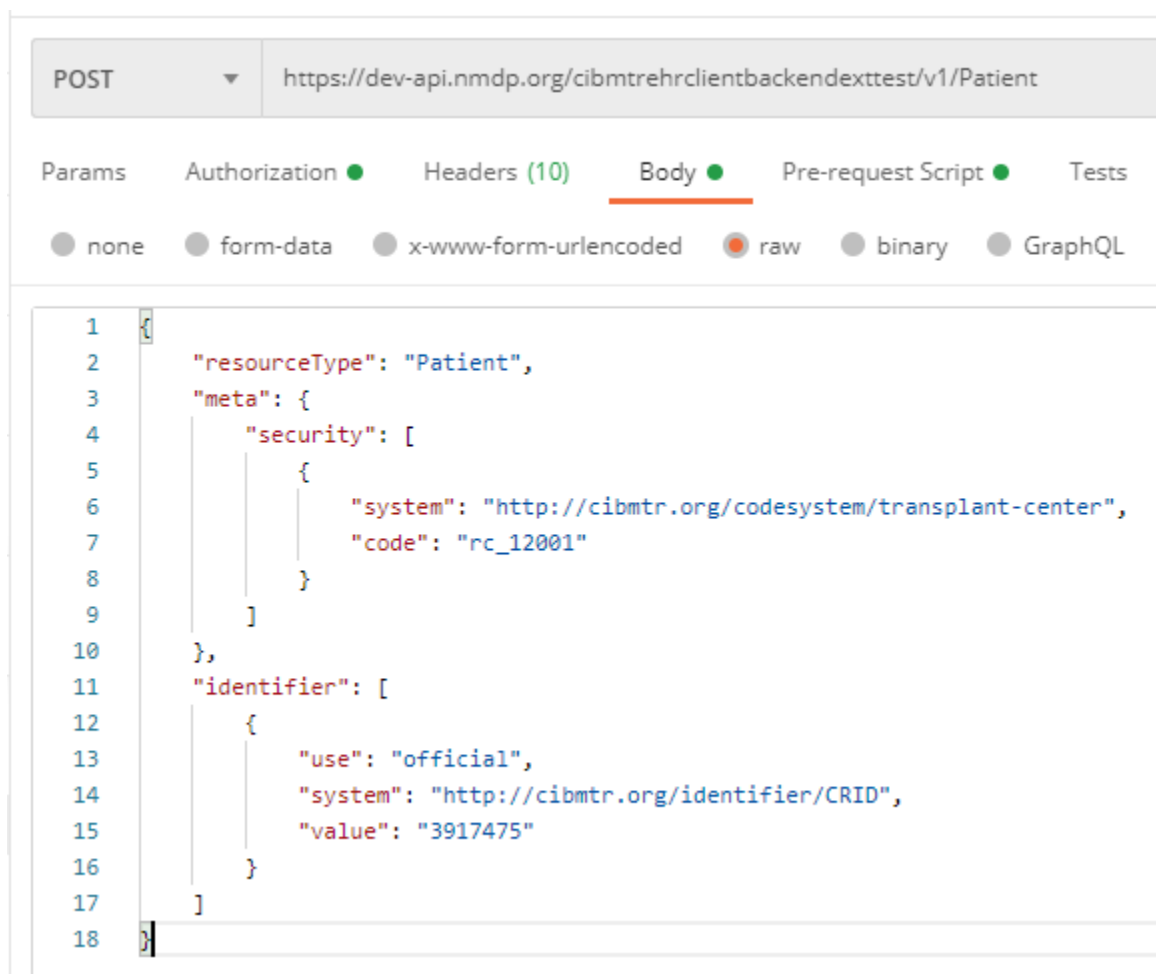
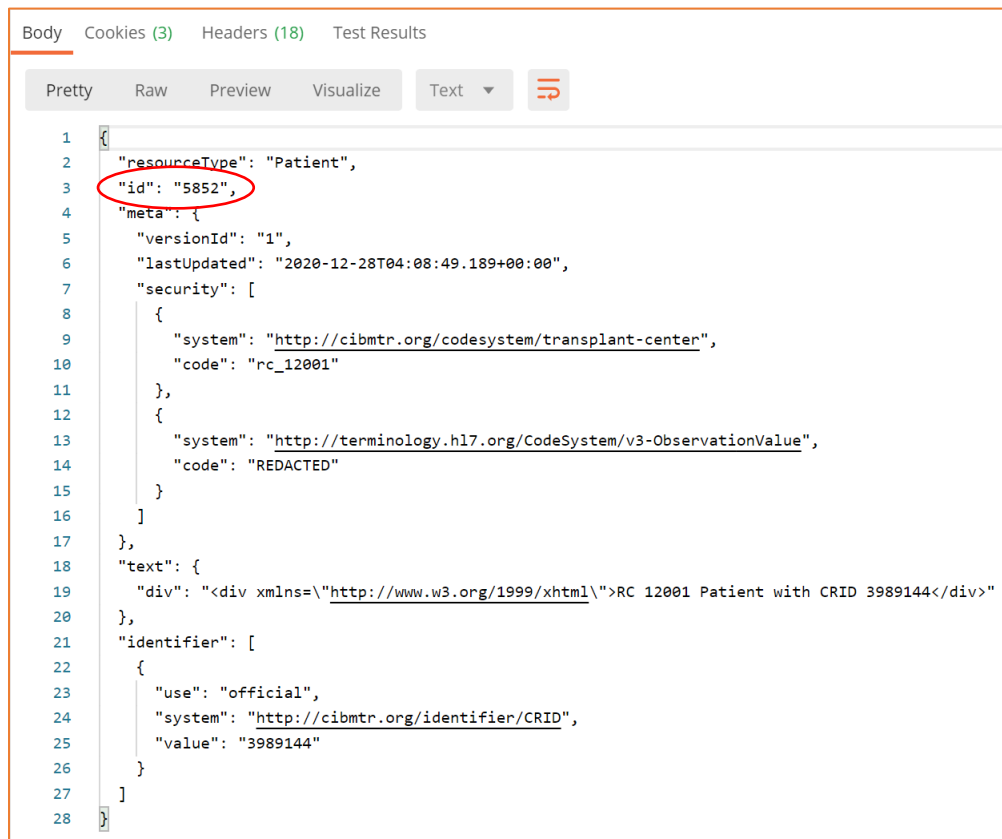


Figure 5: Example POST request to submit a Patient FHIR resource and the required FHIR sections in the body of the request



```
1 {
2   "resourceType": "Patient",
3   "id": "5852",
4   "meta": {
5     "versionId": "1",
6     "lastUpdated": "2020-12-28T04:08:49.189+00:00",
7     "security": [
8       {
9         "system": "http://cibmtr.org/codesystem/transplant-center",
10        "code": "rc_12001"
11      },
12      {
13        "system": "http://terminology.hl7.org/CodeSystem/v3-ObservationValue",
14        "code": "REDACTED"
15      }
16    ]
17  },
18  "text": {
19    "div": "<div xmlns=\"http://www.w3.org/1999/xhtml\">RC 12001 Patient with CRID 3989144</div>"
20  },
21  "identifier": [
22    {
23      "use": "official",
24      "system": "http://cibmtr.org/identifier/CRID",
25      "value": "3989144"
26    }
27  ]
28 }
```

Figure 6: Example FHIR Patient submission response with the Patient resource ID assigned by the FHIR server

Step 3: Submit Observation FHIR Resources

The Direct FHIR Service API uses a POST request to submit an Observation resource at the following case-sensitive endpoint URLs³:

Test Environment

POST https://dev-api.nmdp.org/cibmtrehrclientbackendexttest/v1/Observation
--

Production Environment

POST https://api.nmdp.org/cibmtrehrclientbackend/v1/Observation
--

³ <http://hl7.org/fhir/STU3/observation.html>

The authorization key and bearer token must be included in the request as mentioned in the previous section. FHIR JSON submissions should also include a “content-type” key in the header with value: “application/fhir+json”.

CIBMTR is continually expanding support for data types to pre-populate CIBMTR forms. Currently, laboratory values for form 2402 are supported. Laboratory data required to answer the questions on form 2402 are communicated using the Observation FHIR resource.

Laboratory Data Supported from Form 2402													
Lab Name	UCUM Unit Code	Candidate LOINC Codes											
White Blood Cell (WBC)	10*9/L	26464-8	6690-2	49498-9	804-5								
Neutrophils	%	26511-6	770-8	23761-0									
Blasts in Blood	%	26446-5	709-6	71669-6									
Hemoglobin	g/dL	30313-1	14775-1	718-7	55782-7	20509-6	30351-1	76768-1	30350-3	76769-9	75928-2	59260-0	93846-4
Platelets	10*9/L	26515-7	777-3	49497-1	778-1								
Blasts in Bone Marrow	%	11150-0											
Serum B2-microglobulin	ug/dL	1952-1	76484-5										
Serum Albumin	g/dL	1751-7	61151-7	61152-5	2862-1								
LDH	U/L	2532-0	14804-9	14805-6									
Serum Plasma Percent	%	79426-3											
Serum Plasma Volume	10*9/L	24103-4											

Figure 7: Table of labs currently supported by the CIBMTR for form prepopulation

An example of an Observation FHIR resource is shown in Figure 8. The basic structure of this FHIR resource is the same for all the different types of labs. Important areas to note:

- “meta” Section – This is the metadata section of the resource and includes the same security label as defined in the Patient resource. This security label is required and should include the center specific CCN.
- “category” Section – This section uses the HL7 Observation category code to enable category-based searches. Currently, only data from the “laboratory” category is supported. This section is optional.
- “code” Section – The clinical concept code for the type of laboratory is included in this section. For laboratory data, the primary clinical vocabulary is LOINC⁴. LOINC codes can have different specific applied concepts depending on a variety of lab parameters such as: collection method, measurement method, sub-types, and naming conventions. A list of candidate LOINC codes for each of the supported lab types is included Figure 7. This list is not comprehensive and choosing the correct code can require clinical interpretation, therefore, data managers are encouraged to get clinician review of the selected LOINC code. This section is required.
- “subject” Section – Each Observation resource must reference the patient associated with the lab values. The subject.reference allows the Observation resource to point to the Patient using the Patient resource “id” using the “Patient/<id>” format. This section is required.

⁴ Information on LOINC codes can be found by going to: <https://loinc.org/<code>>

- “effectiveDateTime” – This is a timezone aware datetime format of the date of collection of the lab sample. This section is required.
- “valueQuantity” – The actual value of the measured lab is represented here as a floating-point number. The unit system and code are also specified. The list of preferred units is shown in Figure 7, however the CIBMTR data translation engine will convert the units after submission if necessary. The units system and code should be UCUM. This section is required.
- “referenceRange” – If the high and low range for this lab are known, they can be defined in this section using the same data format as the valueQuantity section. This section is optional.

To search for all Observation resources on the CIBMTR FHIR server for a given CRID, see the below GET request API URL:

GET <a href="https://dev-api.nmdp.org/cibmtrehrclientbackendexttest/v1/Observation?subject:Patient.identifier=<CRID>">https://dev-api.nmdp.org/cibmtrehrclientbackendexttest/v1/Observation? subject:Patient.identifier=<CRID>
--

```
POST https://dev-api.nmdp.org/cibmtrclientbackendexttest/v1/Observation

Params Authorization Headers (10) Body Pre-request Script Tests Set

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL JSON

1 {
2   "resourceType": "Observation",
3   "meta": {
4     "security": [
5       {
6         "system": "http://cibmtr.org/codesystem/transplant-center",
7         "code": "rc_12001"
8       }
9     ]
10  },
11  "status": "final",
12  "category": [
13    {
14      "coding": [
15        {
16          "system": "http://hl7.org/fhir/observation-category",
17          "code": "laboratory",
18          "display": "Laboratory"
19        }
20      ],
21      "text": "The results of observations generated by laboratories."
22    }
23  ],
24  "code": {
25    "coding": [
26      {
27        "system": "https://loinc.org",
28        "code": "6690-2",
29        "display": "Leukocytes [# /volume] in Blood by Automated count"
30      }
31    ]
32  },
33  "subject": {
34    "reference": "Patient/7028"
35  },
36  "effectiveDateTime": "2020-06-15T15:34:10+05:00",
37  "valueQuantity": {
38    "value": 3.67,
39    "unit": "billion per liter",
40    "system": "http://unitsofmeasure.org",
41    "code": "10^9/L"
42  },
43  "referenceRange": [
44    {
45      "low": {
46        "value": 4.5,
47        "unit": "billion per liter",
48        "system": "http://unitsofmeasure.org",
49        "code": "10^9/L"
50      },
51      "high": {
52        "value": 11,
53        "unit": "billion per liter",
54        "system": "http://unitsofmeasure.org",
55        "code": "10^9/L"
56      }
57    }
58  ]
59 }
```

Figure 8: Example POST request for submitting an Observation resource, including the Observation resource format in the body of the request

Submitting CRID/FHIR Data Using the Postman Client

The example API calls in this document are taken from the Postman API client. Postman allows a user to manually configure and test connecting to and interacting with different APIs. Using Postman is a great way to understand an API, see the responses, and submit limited data manually. Once the API is well understood, then a custom client can be implemented programmatically using any number of REST client libraries.

Postman includes the concept of a collection of requests. A collection file can be imported into Postman. CIBMTR has a collection of requests that accomplish all the tasks in this user guide. The collection is available upon request.

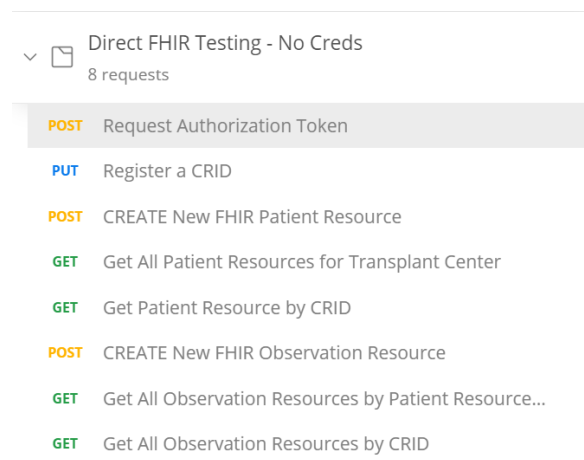


Figure9: Example POSTMAN collection of requests available from CIBMTR

Postman also includes the option to run a pre-request script before making an API request. The CIBMTR collection includes a pre-request script that can get the authentication token automatically each time a request is made. These and other simplifications of the process make Postman an excellent tool for exploring, developing, and using the CIBMTR Direct FHIR Service APIs for submitting patient data.

NOTE: Requesting a new token for manual requests should not cause Okta to rate-limit these requests. However, automated systems must cache and re-use the authentication token to avoid errors. Tokens are valid for 30 minutes in the production environment.

Frequently Asked Questions

Can multiple patients be registered at the same time using the CRID Service API?

No, the API currently supports one CRID registration at a time.

Can demographic data be changed, augmented, or updated using the CRID Service API?

No, contact CIBMTR to have the demographics data changed for a previously registered CRID

What can I do if I forget the CRID for a particular patient?

Send the same PUT request to the CRID Service API with the same patient demographic information and the CRID Service API will return the corresponding CRID number for that patient.

What are the FHIR resources that are supported by the Direct FHIR Service API?

Patient, Observation

What forms are currently supported for pre-population?

Currently, only lab data for form 2402 are supported for disease subtypes: MDS, MPN, and PCD

How will I know if form 2402 was prepopulated with the data I submitted?

If submitted data can prepopulate one of the supported labs in form 2402, then those lab questions will be answered on the form in the standard Formsnet user interface.

What if my lab data is not in the preferred unit of measure?

You may choose to perform the unit/value conversion yourself prior to the data submission or submit the data and the CIBMTR data translation engine will attempt to perform the conversion after submission.

How long does it take for data submitted via Direct FHIR Service API to appear as prepopulated questions on form 2402?

Questions should be populated in less than 10 minutes after submission via the API, however, contextual questions are required to be answered in the FormsNet user interface in order to provide key dates needed to correctly place the lab data on the form.

Can the Observation resources be submitted as a FHIR bundle?

Currently, the Direct FHIR Service API cannot accept resources formatted as a bundle, they must be submitted individually