Health Plan of San Joaquin

Patient Access API Documentation

Getting Started

Welcome to the Developer Portal! The portal provides access to our APIs that are based on the **Health Level 7® (HL7) Fast Healthcare Interoperability Resources (FHIR®) 4.0.1 standard**.

Our APIs give you the ability to build applications for the members so that they can access their claims and encounters information (including cost), as well as a defined sub-set of their clinical information.

You will find information on the portal related to our APIs and how to access them with your application. This includes a Quick Start Guide to help you get your developer account registered and your application authorized, along with instructions on how to get your application connected to our sandbox environment.

Quick Start Guide

Follow these steps to register your account and get your application authorized to begin testing your application.

1. On the Developer Portal, click the **Log in as Developer** button and follow the prompts to create your account.

2. After registering your account, you will receive an email with a link. Use this link to validate your account and log in to the portal.

3. Once you are logged in, click **Register Application**.

4. In the application registration form, provide the required details and register your application.

5. After registering the application, you will receive an email notification on your registered email address once the *Client ID* and *Secret Key* are issued for your application.

6. Once you receive the *Client ID* and *Secret Key*, you can begin using the APIs. You can register any number of applications and delete existing applications. Keep in mind that deleted applications cannot be restored.

Health Plan of San Joaquin

How to Connect

Use your application's *Client ID, Secret Key*, authorization codes, and tokens to securely connect your application to the APIs.

Authorization Overview

Our Access APIs are based on the **FHIR SMART app framework** and rely on the **OAuth 2.0 specification** for securing connections.

Application Registration

To begin, create a developer account and register your application. During the application registration, assign a callback URL (a redirect URI) to your application. This URL will be used during the authorization

In addition, specify the scopes that will define the authorization components of the member that will be using your application. Our API implements the **SMART App Launch: Scopes and Launch Context** to manage access to member data.

Currently, the following scopes are available and enabled by default:

Scope	Grants
`patient/*.read`	This scope permits your application to access the supported resources for a currently logged-in member.
`openid`	This scope permits your application to retrieve information about a currently logged-in member.
`fhirUser`	This scope permits your application to retrieve information about a currently logged-in member.

After registering your application, it will be assigned a *Client ID* and a *Secret Key*. You will use the *Client ID* and the *Secret Key* that you received after registering your application in exchanges with the Identity Server to receive your JSON Web Token (JWT).

Health Plan of San Joaquin

Standard Authorization Code Flow

In the standard authorization code flow, to connect to the Access API, you will need to use the OAuth 2.0 flow for authentication.

In this flow, if a member uses your application to access data through our APIs, your application will send a request to our authorization server to perform authentication. Then the authorization server will redirect member to our Identity Server where member will enter the login credentials.

Upon successfully logging in and providing the required authorization, the Identity Server will redirect the member back to our authorization server. Our authorization server will then call your application using the registered redirect URI, with the authorization code included in the query parameters. The authorization code can then be exchanged for a JWT. The JWT should be included in FHIR requests as an authentication bearer token (within the request header). This token gives your application access to the FHIR server on behalf of the member that is logged in, allowing you to pass data back to the application.

Request Authorization from User

To allow a member to authorize your application, direct them to our `/authorize` endpoint:<**To be entered by customer: Enter the link to the endpoint URL**>. This allows the member to securely log in using their credentials.

The request must include the `response_type` set to `code`, your application's `client_id`, and your application's `redirect_uri`.

The following is an example of a web application's authorization request:

```
GET
https://uat120.fhir.edifecsfedcloud.com/hpsj_auth/authorize?response_type=cod
e&client_id=client123&scope=openid%20read%20search&redirect_uri=https://yourd
omain.com/redirect_uri
```

Authorize Components

```
GET
https://{{domain}}/oauth/oauth2/authorize?response_type={{response_type}}&cli
ent_id={{client_id}}&scope={{scope}}&redirect_uri={{redirect_uri}}
```

URL Protocol: https
Domain: <**To be entered by customer**>
Response Type: `code`
Client ID: The Client ID that you received when you registered your application
Scope: `patient/*.read` `openid` `fhirUser`
Redirect URI: An endpoint in your system that receives callbacks from our API. You entered this URL when you created your application and assigned it as the callback URL.

Health Plan of San Joaquin

Exchange Code for Token

After sending the authorization request, the member will be redirected to a sign-in page, where they will provide their credentials to authenticate themselves. Upon successfully signing in, the member will see a consent page where they provide the consent to authorization server for sharing data with the application. Post successful authorization by the member, the authorization server will provide the full token to make calls to the FHIR server and access member data.

You will send a `POST` request to the token endpoint: POST https://uat120.fhir.edifecsfedcloud.com/hpsj_auth/token

The `POST` request must contain the following in the request body:

```
grant_type: "authorization_code"
code: "code-received-in-redirect-url"
redirect_uri: "https://yourdomain.com/redirect_uri"
client_id: "client123"
client_secret: "iTdhpel10PkeySSdB8nHag7iqBem5aOwf3jZrImkA"
```

The response body will contain the following:

```
{
  "access_token": "eyJraWQiOiJzaTVja3Ny",
  "refresh_token": "cXZwMjAyMDEyMTgwODAx",
  "patient": "123456",
  "scope": "search read openid",
  "id_token": "MzgiLCJhbGciOiJSUzI1NiJ9",
  "token_type": "Bearer",
  "expires_in": 3000
}
```

You can now use this token within the request header in your calls to the FHIR server.

Sandbox Environment Access

How to Use the Sandbox

To use the Sandbox environment, follow these steps:

1. Register as a developer on the **Developer Portal** by clicking **Log in as Developer** and filling in the registration fields.

2. After completing your registration, log in to the developer account and register your application by clicking **Register Application**. **Developer Portal** will send a registration email to the portal admin.

3. Upon receiving your registration email, the portal admin will configure your application for the sandbox environment. You will receive a *Client ID* and *Secret Key* in the **Developer Portal**, which will allow you to begin making authorization calls to the Sandbox environment.

4. Use the Sandbox authentication endpoint URLs, listed in the **Authentication URLs** section below to make the authorization calls. Refer to **How to Connect** section for more information on the authorization calls.

5. After sending the authorization request, your application will be directed to a sign-in page through browser re-directs. When you access the sign-in page, use one of the synthetic users listed in the **Sandbox Test Users** section below. Upon signing in, you will see an authorization page. Check the authorization box to continue.

6. After successfully completing the authentication/authorization calls and receiving your JWT, exchange the JWT for access to the synthetic data contained in the Sandbox.

7. Use the base URLs listed in the **Authentication URLs** section below to make calls to the FHIR server that contains synthetic data. You can also refer the **FHIR APIs** page for swagger definitions of the available endpoints.

Note: If you encounter any issues with your authentication access to the Sandbox or any issues with the Sandbox API, please email a support representative at edi@hpsj.com to resolve your issue.

Sandbox Client Configuration Information

To use the Sandbox environment, perform the following steps:

1. Register using the login link above.

2. After registering, log in to **Developer Portal** and register your application.

3. Provide the following information during application registration:

    o Application Name

    o Scope

    o Standard authorization client

    o Application Callback URL

4. A portal representative will configure your custom Sandbox Client, and you will receive your Sandbox credential information in the portal.

Authentication URLs

Use the following authentication URLs for the Sandbox environment (the authentication URLs in the **Standard Authorization Code Flow** section are for the production environment):

- o https://uat120.fhir.edifecsfedcloud.com/hpsj_auth/authorize
- o https://uat120.fhir.edifecsfedcloud.com/hpsj_auth/token

These will be used when making RESTful API calls to the FHIR server that contains the synthetic data.

Note: For details on how to configure the authorization flow, refer to **How to Connect** section.

Sandbox Test Users

Use these users to simulate member accessing your application. These users allow you to test your workflows using test-user logins supported by synthetic FHIR data.

Synthetic Member	Username	Password
User 1	user1	password1
User 2	user2	password2
User 3	user3	password3
User 4	user4	password4
User 5	user5	password5

Health Plan of San Joaquin

Patient Access API

Data Retrieval

Following successful authentication, the Sandbox Patient Access API Endpoint may be queried at the following base URL. For a complete, interactive directory of Patient Access API Resources, and their definitions, please visit HPSJ's FHIR Developer Portal.

Base URL:
https://uat120.fhir.edifecsfedcloud.com/hpsj_fhir

Resource Directory:
https://fdp.edifecsfedcloud.com/#/portal/health.plan.of.san.joaquin/fhir-apis

Example Resource:
GET/Patient

Parameters

Name	Description
_language String (query)	The language of the resource *_language*
Birthdate string (query)	The patient's date of birth *NOTE*: This US Core SearchParameter definition extends the usage context of capabilitystatement-expectation extension to formally express implementer conformance expectations for these elements: • multipleAnd • multipleOr • comparator • modifier • chain *_birthdate*
Deceased string (query)	This patient has been marked as deceased, or as a death date entered *deceased*

Name	Description
address-state string (query)	A state specified in an address *address-state*
Gender string (query)	Gender of the patient *NOTE*: This US Core SearchParameter definition extends the usage context of capabilitystatement-expectation extension to formally express implementer conformance expectations for these elements: • multipleAnd • multipleOr • comparator • modifier • chain *gender*
Ethnicity string (query)	Returns patients with an ethnicity extension matching the specified code. *ethnicity*
Link string (query)	All patients linked to the given patient *link*
Language string (query)	Language code (irrespective of use value) *language*
address-country string (query)	A country specified in an address *address-country*
death-date string (query)	The date of death has been provided and satisfies this search value
Phonetic string (query)	A portion of either family or given name using some kind of phonetic matching algorithm

Name	Description
Telecom string (query)	The value in any kind of telecom details of the patient
address-city string (query)	A city specified in an address
Email string (query)	A value in an email contact
Given string (query)	A portion of the given name of the patient *NOTE*: This US Core SearchParameter definition extends the usage context of capabilitystatement-expectation extension to formally express implementer conformance expectations for these elements: • multipleAnd • multipleOr • comparator • modifier • chain
Identifier string (query)	A patient identifier *NOTE*: This US Core SearchParameter definition extends the usage context of capabilitystatement-expectation extension to formally express implementer conformance expectations for these elements: • multipleAnd • multipleOr • comparator • modifier • chain
Address string (query)	A server defined search that may match any of the string fields in the Address, including line, city, district, state, country, postalCode, and/or text
Race string (query)	Returns patients with a race extension matching the specified code.

Name	Description
general-practitioner string (query)	Patient's nominated general practitioner, not the organization that manages the record
Active string (query)	Whether the patient record is active
address-postalcode string (query)	A postalCode specified in an address
Phone string (query)	A value in a phone contact
Organization string (query)	The organization that is the custodian of the patient record
address-use string (query)	A use code specified in an address
Name string (query)	A server defined search that may match any of the string fields in the HumanName, including family, give, prefix, suffix, suffix, and/or text *NOTE*: This US Core SearchParameter definition extends the usage context of capabilitystatement-expectation extension to formally express implementer conformance expectations for these elements: • multipleAnd • multipleOr • comparator • modifier • chain
_id string (query)	Logical id of this artifact *NOTE*: This US Core SearchParameter definition extends the usage context of

Name	Description
	capabilitystatement-expectation extension to formally express implementer conformance expectations for these elements: • multipleAnd • multipleOr • comparator • modifier • chain
Family string (query)	A portion of the family name of the patient *NOTE*: This US Core SearchParameter definition extends the usage context of capabilitystatement-expectation extension to formally express implementer conformance expectations for these elements: • multipleAnd • multipleOr • comparator • modifier • chain

Responses

Code	Description
200	OK
404	Resource not found
405	Method not allowed
406	Not acceptable
400	Bad request
500	Internal Server Error
501	Method Not Implemented

Note:
Download Swagger API to get detailed view of example and schema.

```
{
  "id": "string",
  "meta": {
    "Meta": {}
  },
  "implicitRules": "string",
```

```
  "language": "string",
  "text": {
    "Narrative": {}
  },
  "contained": [
    {
      "Resource": {}
    }
  ],
  "extension": [
    {
      "Extension": {}
    }
  ],
  "modifierExtension": [
    {
      "Extension": {}
    }
  ],
  "identifier": [
    {
      "Identifier": {}
    }
  ],
  "active": true,
  "name": [
    {
      "HumanName": {}
    }
  ],
  "telecom": [
    {
      "ContactPoint": {}
    }
  ],
  "gender": "string",
  "birthDate": "string",
  "deceased": true,
  "address": [
    {
      "Address": {}
    }
  ],
  "maritalStatus": {
    "CodeableConcept": {}
  },
  "multipleBirth": true,
  "photo": [
    {
      "Attachment": {}
    }
  ],
  "contact": [
    {
      "ContactComponent": {}
```

```
    }
  ],
  "communication": [
    {
      "PatientCommunicationComponent": {}
    }
  ],
  "generalPractitioner": [
    {
      "Organization": {}
    }
  ],
  "managingOrganization": {
    "Organization": {}
  },
  "link": [
    {
      "PatientLinkComponent": {}
```