Interoperability Product Documentation

NOTE: Some links in this PDF file may access content on the Customer Exchange (CX) website. To view the content, you must log in with your CX account credentials.

Using the Developer Portal

The **Interoperability Developer Portal** is the platform that third-party and payer developers can use to develop and test their FHIR-based healthcare applications under contract with Cognizant clients.

This section provides details for accessing and using the Developer Portal user interface.

Support for OAuth 2.0

The Developer Portal supports oAuth2 protocol. OAuth 2.0 is the industry-standard protocol for authorization. OAuth 2.0 focuses on client developer simplicity while providing specific authorization flows for web applications, desktop applications, mobile phones, and living room devices.

The Client-provided Identity Provider (IDP) authorization service should support OpenID Connect (OIDC). OIDC is a thin layer that sits on top of OAuth 2.0 that adds login and profile information about the person who is logged in.

Start Here

Contents:

Show / Hide Contents

Welcome to the API Developer Portal!

Here you can learn the capabilities of the Developer Portal and our FHIR APIs, and get started on building FHIR-based applications.

Two Types of Developers

Which type of developer am I?

- If you are a **third-party developer**, you are working on a FHIR-based healthcare application under contract from a health plan company.
- If you are a **payer developer**, you are working for a health plan company creating FHIR-based payer-to-payer applications, which are used to transfer health plan data from one payer to another.

FAQs for Developers

What can I do without an account? With no account, you can read Home page content, read about API endpoints, and follow footer links. More importantly, you can explore APIs and try Provider Directory API calls against mock data in the API Sandbox without being logged in.

What's the API Sandbox? The API Sandbox is a canned collection of desensitized, non-PHI test data which API endpoints can access for test purposes.

What can I do when logged in? After signing up for a developer account and logging in, you can view and test Patient Access or Payer-to-Payer APIs against mock data. You can also register your Patient Access application. Payer developers can register their payer-to-payer applications. More importantly, you can download a "swagger" file to get a start on building applications.

Do I have to sign up? No, you don't have to sign up to try the Provider Directory APIs. But you need to sign up to work with Patient Access or Payer-to-Payer APIs, view reports and manage your account.

Launching the Developer Portal

To launch the Developer Portal:

1. Launch the Developer Portal using the URL specific for your health plan. The Home page of the Developer Portal appears.
2. Scroll down the page to read about the APIs and profiles supported by your health plan.
3. From here you can:
    a. View the available API endpoints.
    b. Sign up for a developer account.
    c. Sign in once you have an account.
4. To see a complete list of steps on how to use the Developer Portal, see the Quick Start Guide.

Supported Browsers

Table 1. Supported Browsers

Browser	Supported
Apple Safari	Yes[1]
Google Chrome	Yes[1]
Microsoft Edge	Yes[1]
Microsoft Internet Explorer	No
Mozilla Firefox	Yes[1]

[1] Supported in the two latest production versions.

The Developer Portal is available from any operating system.

API Sandbox Testing Information and Dataset

See API Sandbox Test Data.

API Sandbox Test Data

Sandbox test data is available and the same for all tenants and clients. This is the out-of-box “sandbox” test data.

Data Hub

Type	Entity	Get Statement
Allergy intol Patient	Atp07	/AllergyIntolerance?patient=Ptp07
CarePlan Patient	Cpp07	/CarePlan?patient=Ptp07
CareTeam Patient	Ctp07	/CareTeam?patient=Ptp07&status=active
Claim Patient	Clmp07	/Claim?patient=Ptp07
Condition Patient	Conp07	/Condition?patient=Ptp07&category=[system]|[code]
Coverage Patient	Covp07	/Coverage?patient=Ptp07
Device Patient	Devp07	/Device?patient=Ptp07
DiagnosticReport Patient	Drepp07	/DiagnosticReport?patient=Ptp07
DocumentReference Patient	Docrefp07	/DocumentReference/Docrefp07
Encounter /Id	Encp07	/Encounter?_id=Encp07
End Point	Epp07	/Endpoint?identifier=http://terminology.hl7.org/CodeSystem/v2-0203|TAX
EOB Patient	Eobp07	/ExplanationOfBenefit?patient=Ptp07
Goal Patient	Goalp07	/Goal?patient=Ptp07
Immunization Patient	Immp07	/Immunization?patient=Ptp07
List	Listp07	List?identifier=com.cognizant.tranzform.fhir.identifiers.logical-identifier|Listp07
Location identifier	Locp07	/Location?name=Locname01&_profile=http://hl7.org/fhir/us/core/StructureDefinition/us-core-location
Medication identifier	Medp07	/Medication?code=http://www.nlm.nih.gov/research/umls/rxnorm|103086
MedicationDispense	Mdp07	/MedicationDispense?patient=Ptp07&status=completed

MedicationKnowledge	Mkp07	/MedicationKnowledge?code=http://www.nlm.nih.gov/research/umls/rxnorm|Mkp07
Medication Request Patient	Mrp07	/MedicationRequest?patient=Ptp07&intent=order&status=active|code
Observation Patient	Obsp07	[base]/Observation?patient=Ptp07&code=http://loinc.org|2075-0
Organization identifier	Orgp07 p2ptestdata20	/Organization/Orgp07 /Organization/p2ptestdata20
PatientID	Ptp07 313131	/Patient/Ptp07 /Patient/313131 *(payer to payer)*
PracRole identifier	Pracrolep07 PracRoleLoadNew11508	/PractitionerRole?specialty=http://snomed.info/sct|9632001 *(payer to payer)*
PractitionerID	Pracp03 Pracp07	/Practitioner/Pracp03 /Practitioner/Pracp07 *(payer to payer)*
Procedure Patient	Procp07	/Procedure?patient=Ptp07
RelatedPerson	Rpp07	/RelatedPerson?patient=Ptp07

FHIR Adapter

/Patient/158601
/Patient?identifier=FACETS:tenant4|158601
/Claim/HSD000162900
/Claim?identifier=FACETS:tenant4%7CHSD000162900
/Claim?patient=158601&status=active
/Claim?patient=158601
/Practitioner/IHMPRAC00005
/Practitioner?identifier=http://hl7.org/fhir/sid/us-npi%7C0012016603
/PractitionerRole?practitioner=IHMPRAC00005
/HealthcareService?identifier=FACETS:tenant4%7C500171717001

/ExplanationOfBenefit/04162BM00003
/ExplanationOfBenefit?identifier=FACETS:tenant4%7C04162BM00003
/ExplanationOfBenefit?patient=158601
/ExplanationOfBenefit?patient=158601&status=active
/Location?provider=IHMPRAC00005
/Location?provider=IHMPRAC00005&_profile=http://hl7.org/fhir/us/davinci-pdex-plan-net/StructureDefinition/plannet-Location
/Location?identifier=FACETS:tenant4%7CIHMRAC00005
/Location?identifier=FACETS:tenant4%7CIHMPRAC00005&_profile=http://hl7.org/fhir/us/davinci-pdex-plan-net/StructureDefinition/plannet-Location
/Organization/500101010
/Organization?name=Medical%20Practice%20Associates
/Organization/10040410
/Organization?name=Wyoming%20Department%20of%20Health
/Organization?name=Wyoming%20Department%20of%20Health&_profile=http://hl7.org/fhir/us/carin/StructureDefinition/carin-bb-organization
/Organization?name=FHFSA302
/Organization?name=FHFSA302&_profile=http://hl7.org/fhir/us/carin/StructureDefinition/carin-bb-organization
/OrganizationAffiliation?identifier=383621801
/InsurancePlan?identifier=P2HR
/Endpoint/OrganizationAffiliation
/Endpoint?name=Practitioner

Viewing API Endpoints

You can view API endpoints (or calls) whether you are signed in or not. You can access Provider Directory API endpoints against mock data without being signed in. However, you have to be signed in to try Patient Access and Payer-to-Payer endpoints. The APIs page lets a developer explore portal functionality, input parameters and output responses.

> **Sign in to try an API call**
>
> You have to be signed in to try a Patient Access or Payer-to-Payer API call and receive valid data in return.

To view API endpoints:

1. Launch the Developer Portal as described in Start Here.
2. On the Home page, click **Explore APIs**. You can also click the **APIs** Top navigation menu item.
3. The APIs page appears. Click the name of an API from the **Name** list.
4. The details page for the selected API appears. Here you can see the list of endpoints and all the in-page documentation for the selected endpoint.

   > NOTE: You must be signed in to use the **Try It** button for Patient Access or Payer-to-Payer endpoints.

5. Try grouping the APIs by tag by selecting the **Group By Tag** button. The list updates to show endpoints listed by tag category.

From here you can try calling an endpoint using the **Try It** feature.

Signing up for Developers

Signing up for a Developer account is covered in User Account Management**.** Click the corresponding link on the Help page.

Trying an API Call

The Developer Portal provides a feature where you can try out API endpoints without having to use Postman or SoapUI to make the request. This section describes the "Try It" feature.

> By default, you can try Provider Directory and Payer-to-Payer API calls against mock data without being logged in, but you must be logged in to try Patient Access API calls. You get access to these APIs based on your user role. Tenant administrators can configure a difference access scheme.

To use the "Try It" feature to call an API endpoint:

1. Navigate to the API Details page for an API.
2. Click an endpoint in the left column of the screen, view the input parameters, then click **Try It**.
3. The **Try It** panel pops out from the right. Fill in required input and header parameters. Select the necessary parameters from the dropdown. Sample data is provided in API Sandbox Test Data.
4. In the **Authorization** dropdown, select the **Authorization Code** option. This populates your token.
5. Scroll to the bottom of the **Try It** panel and click **Send**.
6. The HTTP response appears. Look for the return code; the "200 OK" means the request was successful.

> The examples discussed here may differ from the sandbox environment for your system.

Downloading a Swagger File

This topic details accessing and using the Developer Portal in an authenticated (gated) flow.

Downloading a "Swagger" File to Start Development

To get started with development, you can download an OpenAPI (Swagger) file for an API.

1. From the APIs page, click on an API.
2. Choose a download format from the **API Definition** list that fits your development model.
3. Check your Downloads folder for the file, or locate the file on your system. From there you can open the file in an application or in your IDE to start development.

Example Patient Read Response

Example


```
{
   "resourceType": "Patient",
   "meta": {
   "profile": [
      "http://hl7.org/fhir/us/core/StructureDefinition/us-core-patient"
   ]
},
"extension": [{
   "url": "http://hl7.org/fhir/us/core/StructureDefinition/us-core-birthsex",
   "valueCode": "male"
}],
"identifier": [{
   "use": "official",
   "type": {
      "coding": [{
         "system": "http://terminology.tzfInterOp.com/CodeSystem/ValueSet/InterOp/INTMEMID",
         "version": "v001.001",
         "code": "INTMEMID",
         "display": "Member Internal ID",
         "userSelected": true
```

```
        }],
        "text": "Member Internal ID"
    },
    "system": "FACETS:tenant4",
    "value": "158601",
    "assigner": {
        "display": "FACETS:tenant4"
    }
},
{
    "use": "official",
    "type": {
        "coding": [{
            "system": "http://terminology.tzfInterOp.com/CodeSystem/ValueSet/InterOp/MEMID",
            "version": "v001.001",
            "code": "MEMID",
            "display": "MemberId",
            "userSelected": true
        }],
        "text": "MemberId"
    },
    "system": "FACETS:tenant4",
    "value": "CORE-00",
    "assigner": {
      "display": "FACETS:tenant4"

    }
},
{
    "use": "official",
    "type": {
        "coding": [{
            "system": "http://terminology.tzfInterOp.com/CodeSystem/ValueSet/InterOp/SUBID",
            "version": "v001.001",
            "code": "SUBID",
            "display": "HeadOfHouse/SubscriberId",
            "userSelected": true
        }],
        "text": "HeadOfHouse/SubscriberId"
    },
    "system": "FACETS:tenant4",
    "value": "CORE",
    "assigner": {
        "display": "FACETS:tenant4"
    }
}],
"name": [{
    "use": "official",
    "text": "SBSB CORE",
    "family": "CORE",
    "given": [
        "SBSB"
```

```
    ]
}],
"gender": "male",
"birthDate": "1950-01-01T00:00:00.0000000+00:00",
"address": [{
    "use": "home",
    "type": "both",
    "text": "SBSB ADDR1, SBSB ADDR2, SBSB ADDR3, Union, NJ, 07083",
    "line": [
        "SBSB ADDR1",
        "SBSB ADDR2",
        "SBSB ADDR3"
    ],
    "city": "Union",
    "district": "Union",
    "state": "NJ",
    "postalCode": "07083"
},
{
    "use": "work",
    "type": "both",
    "text": "SBSB ADDR1, SBSB ADDR2, SBSB ADDR3, Union, NJ, 07083",
    "line": [
        "SBSB ADDR1",
        "SBSB ADDR2",
        "SBSB ADDR3"
    ],
    "city": "Union",
    "district": "Union",
    "state": "NJ",
    "postalCode": "07083"
    }
    ],
    "maritalStatus": {
        "coding": [{
            "system": "http://terminology.tzfInterOp.com/CodeSystem/ValueSet/InterOp/MaritalStatus",
            "version": "v001.001",
            "code": "Unreported",
            "userSelected": true
    }],
    "text": "UNREPORTED"
    }
}
```

Legal Information:

HL7 is the registered trademark of Health Level Seven International and the use does not constitute endorsement by HL7.
FHIR® and the FLAME mark are the registered trademarks of HL7 and are used with the permission of HL7. The use of these trademarks do not constitute a product endorsement by HL7.

Accessing Reports

The Reports page provides various types of visual presentations on API usage for the signed-in developer.

Report Limitations

- Only API Sandbox requests are tracked.
- Postman requests are not counted.

Table 1. Breakdown of Reports

Report Section	Description
API Calls	Provides a graph of Total, Successful, Failed and Blocked requests for the period of time selected by the buttons.
Data transfer	Shows a graph of the bandwidth usage over a selected period of time.
API response times	Shows a graph of response times comparing minimum, maximum and average times in milliseconds.
APIs	Shows statistics on the number of successful, blocked, failed, other and total calls by API. Also shows average response time in milliseconds and bandwidth used in bytes.
Operations	Shows statistics on the number of successful, blocked, failed, other and total calls by individual operation. Also shows average response time in milliseconds and bandwidth used in bytes.

To access reports:

1. Click the **Reports** Top navigation menu item. The Reports screen appears. It displays a row of buttons that let you filter the graph data. Last Hour, Today, Last 7 days, Last 30 days and Last 90 days are the filters, Last 7 days being the default.
2. Scroll down to see the graphs for API calls, Data transfer, and API response times.
3. Scroll farther down to see the tables showing the call statistics for APIs and Operations. You'll see statistics on successful calls, blocked calls, failed calls, average response time and bandwidth.
4. Pick a report timeframe. Choices are:
   - LAST HOUR
   - TODAY
   - LAST 7 DAYS (default)
   - LAST 30 DAYS
   - LAST 90 DAYS

5. Click on column heading links to sort the table.

Viewing Your User Profile

All the specifics of your Developer Portal account are listed on the Profile page. Here you can see account details, and manage various aspects of your profile.

Getting to Your Profile

To get to your profile:

1. Once signed in, click **Profile** in the Top navigation menu.
2. The **Profile** page appears.
3. Scroll down to see the rest of the page and its **Account Details** section.

Account Details

The Account Details section of the **User Profile** page shows the email address and full name used to create the account.

> For clients who are using the Out of Box Identity Provider (IDP), see the **Account Management** page for details on how to change your account details.

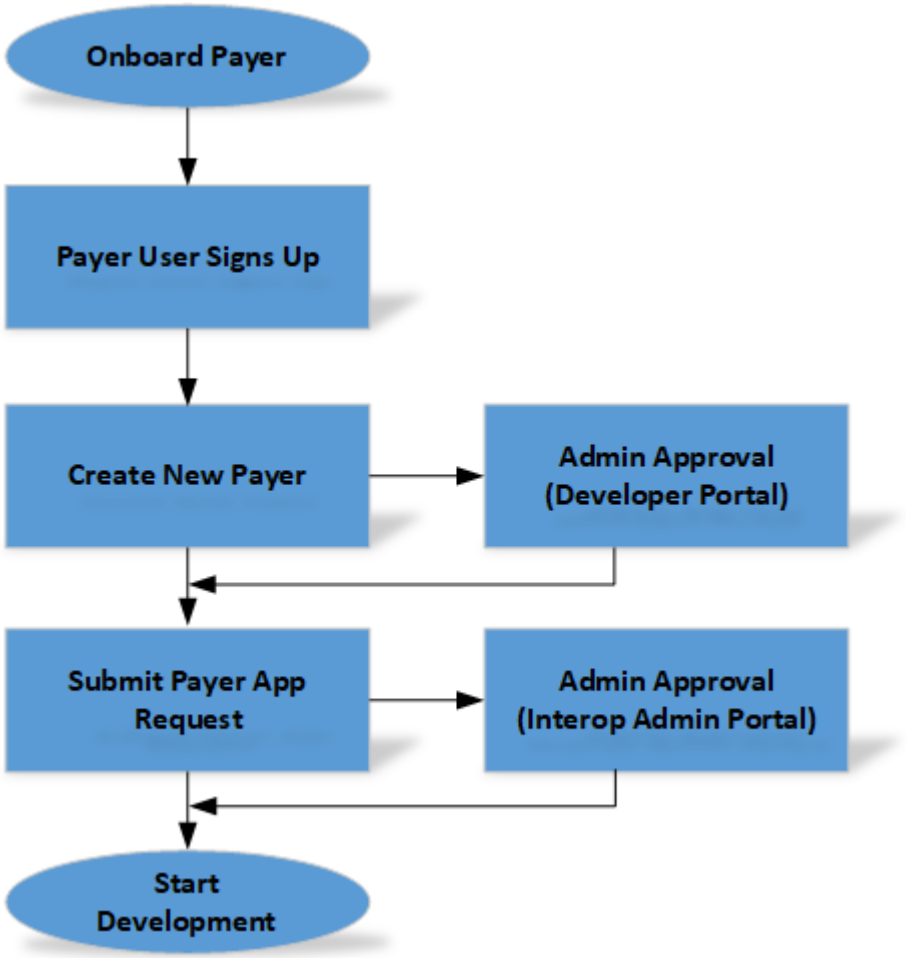
Account Management

The operations you can perform for developer account management are described in User Account Management. Click the corresponding link on the Help page.

Working with Payers

Payer-to-payer applications enable healthcare companies (payers) to exchange member data with other payers for when members change health plans due to job change or change in coverage. Figure 1 shows the process of developing payer-to-payer applications - from onboarding a payer user to submitting the payer application.

Figure 1. Payer-to-Payer Flow



Launching the Payer Registration Dashboard

The Payer Registration Dashboard is where existing payers are viewed and new payers are onboarded. To see the link to the Dashboard, you must have a Developer Portal user account in the payer role.

To launch the Payer Registration Dashboard

1. Launch the Developer Portal.
2. Click **Sign In** with your payer user account. The Home page appears.
3. On the Home page, click the **Payer Registration** button in the Top navigation menu. The Payer Registration page appears. Here you see a **Search and View Registered Payer** dropdown, a **NEW PAYER REGISTRATION** button, and a **Payer Details** list.

From here, you can

- View existing payers in the Payer Details list
- Register a new payer application (for details, see Registering a New Payer Application through the Developer Portal).

Managing Payers

Administrators manage payer registrations via the Payer Registration dashboard in the Developer Portal. Through the dashboard, administrators can approve or deny payer registrations, and register new payers.

This page covers how to view and manage existing registered payers. To see how to register a new payer, see Registering a New Payer through the Developer Portal.

About the Payer Details List

By default, **Payer Details** lists in tabular form all submitted, denied and approved payers. Key data elements are the payer name, status, organization ID and associated users.

The Payer Details list contains the following columns:

Column	Description
Payer Name	The name of the registered payer.
Payer TIN	The payer's tax ID number (TIN).
Address	The payer's address information.
Contacts	The payer's contact information..
Status	The status of the submitted payer registration. Can be: • Submitted • Approved • Denied.
Organization ID	If the app is approved, the Organization resource ID for this payer.
Action	Actions that can be performed for this payer registration. Actions can be: • Assign Self: Take ownership of pending payer registration. • Edit: Edit the address and contact information for an approved payer.
Requested By	The payer user registering the payer.
Assignee	Administrator who is evaluating the payer.
Associated Users	Users associated with this payer when registering.

Viewing a Registered Payer

The Payer Registration dashboard lets you view one registered payer at a time.

To view a specific registered payer

1. Navigate to the **Payer Registration** page as described in Launching the Payer Registration Dashboard.
2. In the **Search and View Registered Payer** dropdown, select a registered payer. The selected payer appears in the Payer Details list below the dropdown.
3. View the information for the payer. The columns are described in About the Payer Details List.
4. If the **STATUS** column contains **Denied**, you can click the **Denied** link to view the reason for the denial in the **Denied Reason** popup.

Editing a Payer

Payer users can edit payers that have submitted their registration or have already been approved.

To edit a registered payer:

1. Sign in to the Developer Portal as a payer user.
2. In the **Top** navigation menu, click **Payer Registration**. The **Payer Registration** page appears.
3. If selecting an **APPROVED** payer from the **Search and View Registered Payer** dropdown, do the following:
    a. In the **Payer Details** list, click **Assign Self** in the **Action** column. The **Assign Self** link turns into an **Edit** link.
    b. Click **Edit**. The Payer Registration page appears with the selected payer's information populating the fields.
4. If looking for a payer in the **SUBMITTED** state, do the following:
    a. In the **Payer Details** list, click **Edit** in the **Action** column. The Payer Registration page appears with the selected payer's information populating the fields.
5. Edit any of the fields and click **SUBMIT**.

The updated information appears in the **Payer Details** list.

Registering a New Payer through the Developer Portal

Contents:

Show / Hide Contents

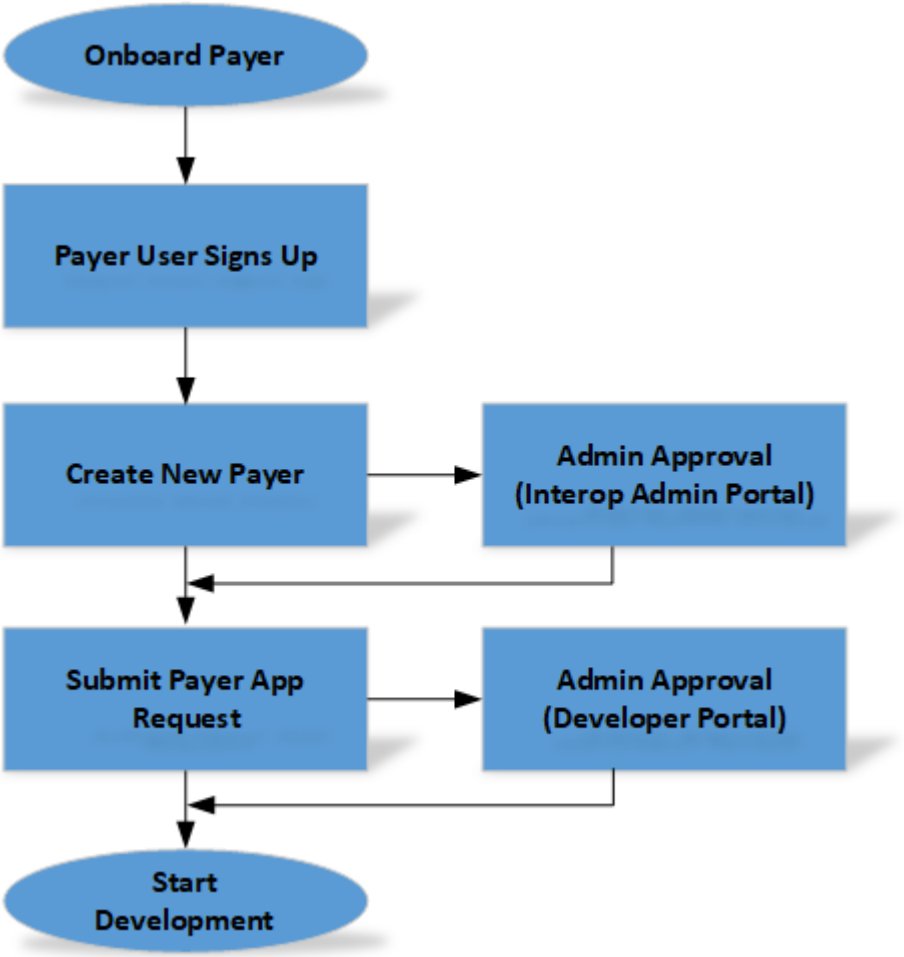
A payer is a healthcare company that offers health insurance coverage to members. In the context of the Developer Portal, payers work to develop payer-to-payer applications that share member data with other health plan payers.

Once registered, a payer can be selected in the payer application flow and is associated with that application.

Payer Registration Flow

The payer registration flow can be configured to meet the needs of health plans that wish to approve or deny payer registration requests manually. See Figure 1.

Figure 1. Payer Registration Flow



If approved, the payer can submit requests for payer-to-payer applications.

Preparing to Register a Payer

Before attempting to register a payer, a payer user must be registered as described in User Account Management. To register a payer, it's beneficial to gather your registration information and settings ahead of time. Here is a mini-checklist to guide you:

Table 1. Checklist for Preparing to Register an Application

Category	Description
Payer Name	The full name of the healthcare payer company.
Payer TIN	The payer's tax ID number.
Address Details	The address of the payer - address, city, state, ZIP code.
Contact Details	The contact information for the payer - telephone number, FAX number, website URL.

You can now start the registration process as described in Registering a Payer below.

Registering a Payer

To register a payer:

1. Prepare for registering a payer as described in Preparing to Register a Payer.
2. Sign in to the Developer Portal as a payer user.
3. In the **Top** navigation menu, click **Payer Registration**. The **Payer Registration** page appears.
4. Click the **NEW PAYER REGISTRATION** button.
5. Fill in the address details section.
6. Fill in the contact details section.
7. Click the **SUBMIT** button.

Health plan administrators then reviews the request and either approve or deny it in the Interoperability Administration portal. The payer user receives an email either way.

If approved, the payer user receives their organization ID, which needs to be supplied in member-match operations for payer-to-payer applications developed either by a payer developer or by a third-party developer who is registering on behalf of a payer.

Once approved, payer users can now register payer-to-payer applications.

Working with Applications

Third-party and payer-to-payer developers start here to register and manage their smartphone and data transfer applications, from submission through approval.

Managing Apps with the App Registration Dashboard

Contents:

Initially submitting **third-party** and **payer-to-payer** applications for approval is just one part of app registration. Once that is done, developers can manage the apps that they've submitted from inside the Developer Portal. The **App Registration Dashboard** lets you view a list of apps you've submitted, their status, and for those that have been approved, it shows the app's primary and secondary Client Secrets.

Launching the App Registration Dashboard

These steps assume you have a Developer Portal user account based on either a third-party developer role or a payer role.

To launch the App Registration Dashboard:

1. Launch the Developer Portal.
2. Click **Sign Up** in the Top navigation menu to create yourself a developer account if you don't already have one.
3. Click **Sign In** to access your account in the Developer Portal. The Home page appears.
4. On the Home page, click the **App Registration** button in the Top navigation menu. The **App Registration** page appears. Here you see a **Filter** section and a complete **List of Registered Applications**.

From here, you can

- Filter the List of Registered Applications for a particular app
- Generate a new client secret
- Register a new application (for details, see Registering a Third-Party Application through the Developer Portal).

Filtering the List of Applications

You may need to narrow the list of applications to make it easier to find the ones you want. The fields you can filter on are:

Field Name	Description
Application Name	Filter by name. Matches full or partial text found in the **App Name** column of the List of Registered Applications.
Application Status	Filter by registration status. Matches full or partial text found in the **Status** column in the List of Registered Applications.

To filter the list of applications:

1. On the App Registration page, locate the Filter panel.
2. Enter a text string into either the **Application Name** or **Application Status** field or both.
3. Click the **Filter** (magnifying glass) button.
4. The **List of Registered Applications** is updated to match the filter criteria.

To reset the filter fields and List of Registered Applications:

1. On the App Registration page, click the **RESET** button. The Application Name and Application Status fields are cleared and the List of Registered Applications now displays all apps.

Using the List of Registered Applications

By default, the List of Registered Applications lists in tabular form all your submitted and approved applications. Key data elements are the App ID, Status and Client Secrets. You can also page through the list of applications. To register a new application, see Registering a Third-Party Application through the Developer Portal.

The List of Registered Applications contains the following columns:

Column	Description
App ID	The application ID used to identify the app.
Environment	The environment in which the app operates. Can be **Sandbox** or **Production**.
App Name	The name of the app.
Description	A blurb about the app.
Requested Date and Time	The date and time the app was submitted.
Status	The status of the submitted app. Can be: • Submitted • Approved • Pending • Revoked
Client Secret1	If the app is approved, the primary client secret can be regenerated. Maybe it has expired or it's time to rotate them.
Client Secret2	If the app is approved, the secondary client secret can be regenerated. Maybe it has expired or it's time to rotate them.
Payer Name / Org Profile	For payer apps only. The name of the payer associated with the application and Organization profile for it.

By filtering the list, you can view a subset of your applications.

Generating a Client Secret

Approved applications are given two client secrets: Client Secret 1 and Client Secret 2. The initial values are generated when the app registration is approved. But the developer can regenerate their own client secrets when needed.

To generate a new Client Secret

1. In the List of Registered Applications, find an approved app.
2. In the Client Secret 1 or Client Secret 2 column, decide which client secret to generate and click its **Generate** button.
3. A popup appears for the selected Client Secret. Copy the secret and paste it into a file for safe keeping.
4. Click Close to dispatch the popup.

Developers can then paste the new Client Secret into their application for authentication purposes.

Registering a Third-Party Application through the Developer Portal

Contents:

Show / Hide Contents

The Interoperability Developer Portal offers signed-in third-party developers a way to register a FHIR-based **Patient Access** or **Provider Directory** healthcare application (called **third-party applications**). These apps must be registered with and fully approved by the health plan before the application can access live data.

Applications can be registered in both **Sandbox** and **Production** environments. Typically you register for Sandbox first to obtain the Application ID and do the development, then you register for Production using the App ID from Sandbox.

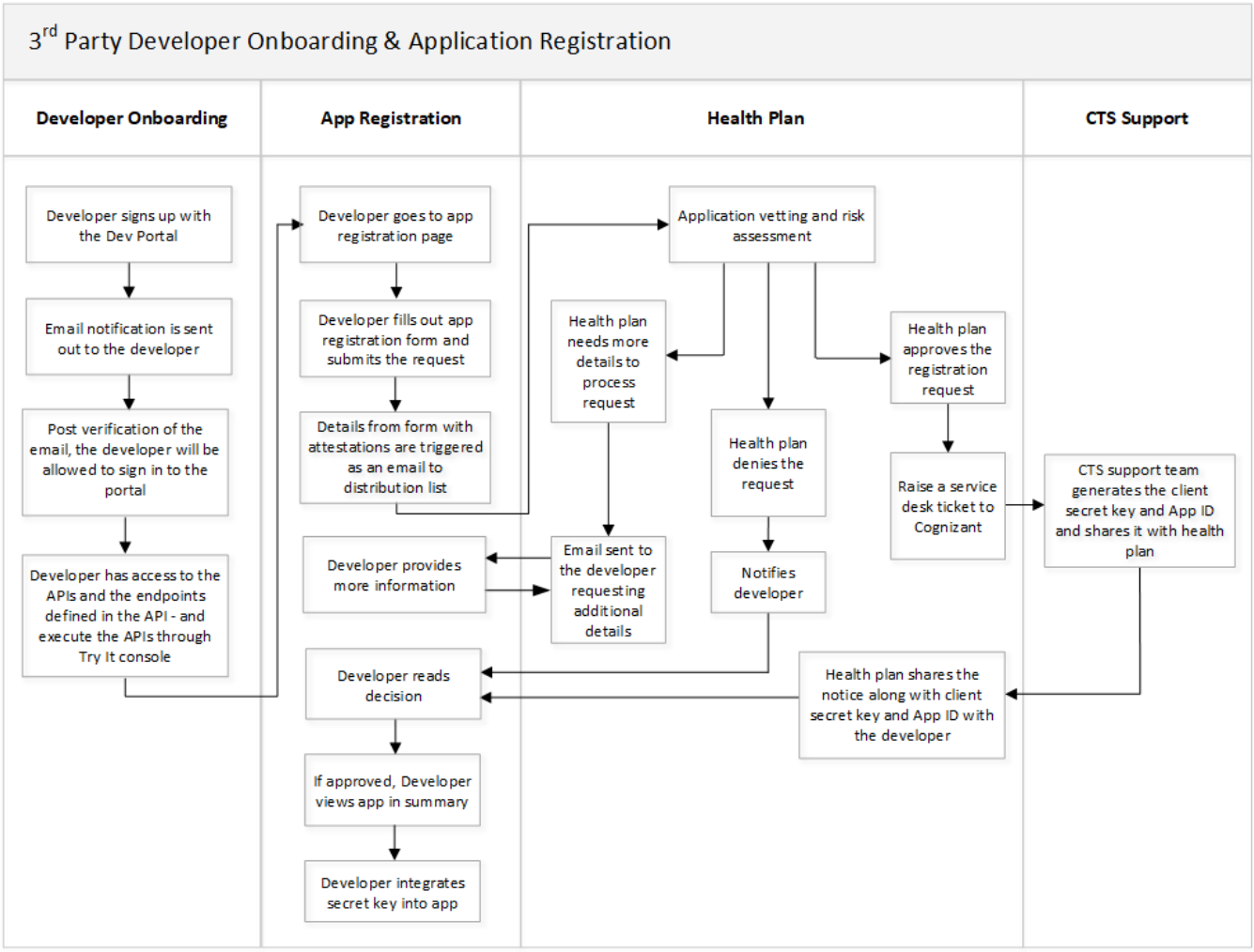
Only **Patient Access** applications need to be registered. **Provider Directory** applications do not.

For details on registering **Payer-to-Payer** applications, see Registering a Payer Application through the Developer Portal.

Configurable Application Registration Flow

The registration flow can be configured to meet the needs of health plans that wish to approve or deny app registration requests manually. See Figure 1.

Figure 1. Application Registration Flow



If approved, the developer accesses the application's client ID and client secret on the Developer Portal. Then the application is ready to start accessing APIs.

Preparing to Register a Third-Party Application

Before attempting to register a third-party application, it's beneficial to gather your registration information and settings ahead of time. Here is a mini-checklist to guide you:

Table 1. Checklist for Preparing to Register an Application

Category	Description
Application ID	The number identifying the application. This ID is auto-generated during Sandbox registration and applied during Production registration.
OAuth redirection	Call back URL and Redirect URI.
OAuth type	Public vs. Confidential. **Confidential** applications can keep the client ID and client secret secure. **Public** applications cannot secure the data.
APIs to be accessed	The only application type that needs to be registered is Patient Access applications.
IP address range	Range of IP addresses to whitelist.
Method of encryption	Encryption support – TLS 1.2 or above. If TLS 1.1 or lower encryption is planned, the application could be denied.
Organizational details	Key website and legal URLs, support contacts, escalation contacts, and developer contacts
Attestation	Attest compliance with the security and privacy policy choices listed on the registration form. These choices ensure the application provides its own privacy policy and terms of use statements. If the developer does not agree to any of the statements, the health plan will notify members to avoid this application and select one that is in compliance.

You can now start the registration process as described in Registering an Application below.

Registering a Third-Party Application

To register an application:

1. Prepare for registering your application as described in Preparing to Register an Application.
2. Sign in to the Developer Portal.
3. Fill in the application and organizational details as described in Preparing to Register an Application.
4. On the Home page, click the **App Registration** button in the Top navigation menu.

5. The **Register New App** page appears. First off, select the **ENVIRONMENT FOR YOUR APP**. The **Sandbox** radio button is selected by default, because you must register the application in Sandbox before attempting a Production registration.

   If you select **Sandbox**, you see fields for the **APP NAME** and **APP DESCRIPTION**. The API Sandbox is a canned collection of desensitized, non-PHI test data which API endpoints can access for test purposes.

   But if you select **Production**, in addition to the **APP NAME** and **APP DESCRIPTION** fields, you see a text field for **TEST APP ID FOR YOUR APP**. This App ID (also known as client ID) is provided once the Sandbox registration is approved.

   The remainder of the form is the same for either environment selection.
6. Fill in the application and organizational details as described in Preparing to Register an Application.
7. Scroll further down and fill out the **ATTESTATION** section of the form. Be sure to check one or more of the Attestation checkboxes.
8. Click the **SUBMIT** button.

The health plan then reviews the request and either approves or denies it. You will receive an email either way.

If approved, you will receive the auto-generated App ID. You can go to the **Profile** page to see your application's information such as client ID and client secret. You can try testing a few APIs either in the **Try It** tool on the Developer Portal or through SoapUI or Postman. You can also download a swagger file to get going with development.

Registering a Payer Application through the Developer Portal

Contents:

Show / Hide Contents

The Interoperability Developer Portal offers signed-in users a way to register FHIR-based payer-to-payer applications using Payer Data Exchange. These apps must be registered with and fully approved by the associated health plan before the application can access live data.

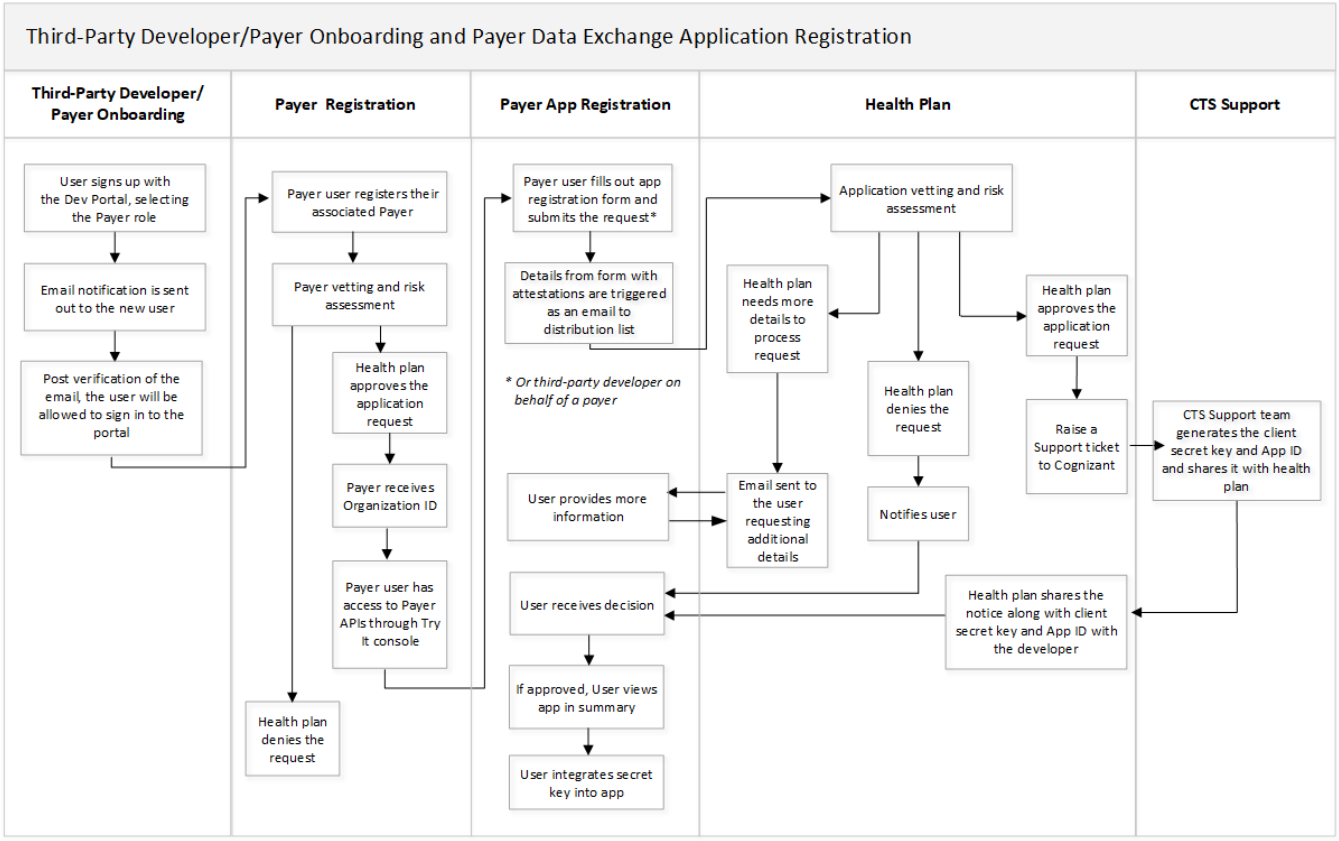
Applications can be registered in both Sandbox and Production environments. Typically you register for Sandbox first to obtain the Application ID and do the development, then you register for Production using the App ID from Sandbox.

Patient Access and Payer-to-Payer applications need to be registered. Provider Directory applications do not.

Configurable Application Registration Flow

The registration flow can be configured to meet the needs of health plans that wish to approve or deny app registration requests manually. See Figure 1.

Figure 1. Application Registration Flow



If the application is approved, the developer accesses the application's client ID and client secret on the Developer Portal. Then the application is ready to start accessing APIs.

Preparing to Register a Payer Application

Before attempting to register an application, it's beneficial to gather your registration information and settings ahead of time. Here is a mini-checklist to guide you:

Table 1. Checklist for Preparing to Register an Application

Category	Description
Application ID	The number identifying the application. This ID is auto-generated during Sandbox registration and applied during Production registration.
OAuth redirection	Call back URL and Redirect URI.
OAuth type	Public vs. Confidential. **Confidential** applications can keep the client ID and client secret secure. **Public** applications cannot secure the data.
APIs to be accessed	Third-party applications work with Patient Access or Provider Directory APIs. Payer-to-payer applications work solely with Payer APIs.
Payer apps only	The associated payer must be registered first, as described in Registering a New Payer through the Developer Portal.
IP address range	Range of IP addresses to whitelist.
Method of encryption	Encryption support – TLS 1.2 or above. If TLS 1.1 or lower encryption is planned, the application could be denied.
Organizational details	Key website and legal URLs, support contacts, escalation contacts, and developer contacts
Attestation	Attest compliance with the security and privacy policy choices listed on the registration form. These choices ensure the application provides its own privacy policy and terms of use statements. If the developer does not agree to any of the statements, the health plan will notify members to avoid this application and select one that is in compliance.

You can now start the registration process as described in Registering a Payer Application below.

Registering a Payer Application

To register a payer application:

1. Prepare for registering your application as described in Preparing to Register a Payer Application.
2. Sign in to the Developer Portal as a payer user.
3. Fill in the application and organizational details as described in Preparing to Register a Payer Application.
4. On the Home page, click the **App Registration** button in the Top navigation menu. The **App Registration** page appears.
5. Click the **REGISTER NEW APPLICATION** button to start the registration. The **Register New Application** page appears.

6. First off, select the **ENVIRONMENT FOR YOUR APP**. The **Sandbox** radio button is selected by default, because you must register the application in Sandbox before attempting a Production registration.

   If you select **Sandbox**, you see fields for the **APP NAME** and **APP DESCRIPTION**. The API Sandbox is a canned collection of desensitized, non-PHI test data which API endpoints can access for test purposes.

   But if you select **Production**, in addition to the **APP NAME** and **APP DESCRIPTION** fields, you see a text field for **TEST APP ID FOR YOUR APP**. This App ID (also known as client ID) is provided once the Sandbox registration is approved.

   The remainder of the form is the same for either environment selection.

7. Fill in the application and organizational details as described in Preparing to Register an Application.
8. Scroll further down and fill out the **ATTESTATION** section of the form. Be sure to check one or more of the Attestation checkboxes.
9. Click the **SUBMIT** button.

The health plan then reviews the request and either approves or denies it. You will receive an email either way.

If approved, you will receive the auto-generated App ID. You can go to the **Profile** page to see your application's information such as client ID and client secret. You can try testing a few APIs either in the **Try It** tool on the Developer Portal or through SoapUI or Postman. You can also download a swagger file to get going with development.

Manual Queries for Application Reports

Contents:

Show / Hide Contents

Tenants who wish to have any of these queries executed in their environment should contact their Cognizant support representative or submit a support ticket.

Query 1: Get Specified App IDs

Instructions: Edit the app IDs list with the specific applications to be retrieved.

```
var appIds = ["10facbfe-036a-4090-9ffa-fcabe78063de", "5d34ebc2-1f76-45d1-a2bb-503288c9ecdd", "34250959-95e9-482d-9941-28bd2bda8cf1"]

db.getCollection("AppRegistration").find({appId: {$in: appIds}}, {appId:1, appName:1, requestedBy:1, apiProductCategory:1, requestedOn:1, environment:1, approvedOn:1, deniedOn:1, status:1, appDevloperEmailAddress:1})
```

Query 1 v2: Get Applications Submitted between Date Range

Instructions: Update the startDate and endDate times to required data ranges to retrieve list of appRegistrations submitted during that period.

```
var startDate = new ISODate("2023-02-01T00:00:00Z");
var endDate = new ISODate("2023-03-01T00:00:00Z");

db.getCollection("AppRegistration").find({requestedOn: { $gte: startDate, $lt: endDate}}, {appId:1, appName:1, requestedBy:1, apiProductCategory:1, requestedOn:1, environment:1, approvedOn:1, deniedOn:1, status:1, appDevloperEmailAddress:1})
```

Query 2: Identify the Last Time an API was Triggered by an App ID/or a List of App IDs

Instructions: Update values in app ID section to retrieve the data for those targets.

```
SELECT [eventTime], [httpMethod], [requestUri], [appId], [xProductCategory]
FROM (
    SELECT *, ROW_NUMBER() OVER (PARTITION BY appId ORDER BY eventTime DESC) AS rn
    FROM [CONSENT_QA].[dbo].[tz_security_audit]
    WHERE appId IN ('@!EF58.35E0.DA10.5058!0001!AB36.2B74!0008!22CE.E3F3.2FBF.3E2B', 'app2')
) AS ranked_data
WHERE rn = 1;
```

Query 3: Return the App That Last Triggered an API

Instructions:

Assumptions:

1. **TenantId** replaced with actual **tenantId** value.
2. Input list of **appIds**.
3. **xProductCategory** can be ignored for Provider Directory API.
4. Replace **appId** list with actual **appIds**, and **tenantId** with actual **tenantId**.

Input: List of **appIds** for the **tenantId**
Output: **appId**, Payload details, userName, product category

```
SELECT tenantId, appId, requestPayload, requestUri, userName, xProductCategory
FROM [CONSENT].[dbo].[tz_security_audit]
WHERE eventTime in
    (SELECT MAX(eventTime) AS last_triggered
        FROM [CONSENT].[dbo].[tz_security_audit]
        WHERE tenantId={tenantId}
        AND appId in ({AppName3}, {AppName2}, {AppName1})
        group by appId
    )
```

Query 4: Get App Names by App IDs

Instructions:

Assumptions:

1. Input list of **appIds**.
2. Database selected based on the **tenantId**.
3. Environment variable determination is based on the **tenantId** selected (mhc -> production, mhcsandbox -> sandbox).

Input: List of **appIds** and environment (**production** or **sandbox**).
Output: **appId**, **appName**

```
// Replace 'AppId List' with array of app IDs and 'environment' based on tenantId
var appIds = ["appId1","appId2"];
var environment = "sandbox";

db.AppRegistration.find({
    appId: { $in: appIds},
    environment: {$regex: environment, $options: 'i'}
}, {
    appName: 1,
    appId: 2
});
```

Query 5: Return Top Three App Usages by Month

Instructions:

Assumptions:

1. **TenantId** replaced with actual **tenantId** value.
2. **userName** column has the value of 'PROVIDER DIRECTORY' and **xProductCategory** is null or empty for Provider Directory APIs.
3. **xProductCategory** has the value that contains 'Patient Access' for Patient Access APIs.
4. Query runs only for Patient Access and Provider Directory API Usage.

Input: **tenantId**, month interval
Output: **appId**, **xProductCategory** for the **appId**, **TotalAppUsageCount** Monthwise, **TotalAppUsageCount**

```
Step 1: (Populate Temp Table with total count by appId and category)
-------------------------------------------------------------------
DROP TABLE IF EXISTS [CONSENT].[dbo].[TEMP_TABLE];

SELECT * INTO [CONSENT].[dbo].[TEMP_TABLE] FROM
(SELECT A.appId, SUM(A.TotalApps) as TotalAppCount FROM
(SELECT appId, Prd_ctgy, count(appId) AS TotalApps FROM
(SELECT appId, CASE
WHEN xProductCategory LIKE '%Patient Access%' then 'Patient Access'
WHEN ((xProductCategory is null or xProductCategory = '') and userName = 'PROVIDER DIRECTORY') THEN
'PROVIDER DIRECTORY'
END AS Prd_ctgy,
DATENAME(MONTH, eventTime) AS Closing_Month
FROM [CONSENT].[dbo].[tz_security_audit]
WHERE eventTime >= '2024-01-01' AND eventTime <= '2024-12-31'
AND appId IS NOT NULL
AND tenantId = 'mhc'
) src1
GROUP BY appId, Prd_ctgy)A
GROUP BY appId) B

Step 2: Populate Temp Table 1 with app Rank:
-----------------------------------------
DROP TABLE IF EXISTS [CONSENT].[dbo].[TEMP_TABLE_1];

SELECT * INTO [CONSENT].[dbo].[TEMP_TABLE_1] FROM
(SELECT appId, TotalAppCount, ROW_NUMBER() OVER(ORDER BY TotalAppCount DESC) AS AppRank from [CONSENT].
[dbo].[TEMP_TABLE]) A
```

```
STEP 3: (Joining the pivot table with temp table to get the Top 3 App usage count)
--------------------------------------------------------------------------------
SELECT final.appId, Prd_ctgy,ISNULL(January,0) AS January, ISNULL(February,0) as February,
ISNULL(March,0) as March, ISNULL(April,0) as April, ISNULL(May,0) as May,
ISNULL(June,0) as June, ISNULL(July,0) as July, ISNULL(August,0) as August,
ISNULL(September,0) as September, ISNULL(October,0) as October, ISNULL(November,0) as November, ISNULL
(December,0) as December,
(ISNULL(January,0) + isnull(February,0) + ISNULL(March,0) + isnull(April,0)
+ ISNULL(May,0) + ISNULL(June,0) + ISNULL(July,0) + ISNULL(August,0)
+ ISNULL(September,0) + ISNULL(October,0) + ISNULL(November,0) + ISNULL(December,0)) AS Total
FROM
(SELECT * FROM
(SELECT appId, Prd_ctgy, Closing_Month, count(appId) AS TotalApps from
(SELECT appId, CASE
WHEN xProductCategory LIKE '%Patient Access%' then 'Patient Access'
WHEN ((xProductCategory is null or xProductCategory = '') and userName = 'PROVIDER DIRECTORY') THEN
'PROVIDER DIRECTORY'
END AS Prd_ctgy,
DATENAME(MONTH, eventTime) AS Closing_Month
FROM [CONSENT].[dbo].[tz_security_audit]
WHERE eventTime >= '2024-01-01' AND eventTime <= '2024-12-31'
AND appId IS NOT NULL
AND tenantId = 'mhc') src1
GROUP BY appId, Prd_ctgy, Closing_Month) SRCTBL
PIVOT
(
sum(TotalApps)
FOR Closing_Month IN ([January], [February], [March], [April], [May], [June], [July], [August],
[September], [October], [November], [December])
) PivotTable
) final
INNER JOIN
[CONSENT].[dbo].[TEMP_TABLE_1] T1
ON final.appId = T1.appId
ORDER BY T1.AppRank
```

Query 6: Return Top Three App Usages by Year to Date

Instructions:

Assumptions:

1. **TenantId** replaced with actual **tenantId** value.
2. **userName** column has the value of 'PROVIDER DIRECTORY' and **xProductCategory** is null or empty for Provider Directory APIs.
3. **xProductCategory** has the value that contains 'Patient Access' for Patient Access APIs.
4. Query runs only for Patient Access and Provider Directory API Usage.
5. May need to tweak it a little bit based on the tenant data.

Input: **tenantId**, **year**
Output: **appId**, **xProductCategory** for the **appId**, **TotalAppUsageCount** for the given year.

```
Step 1: (Populate Temp Table with total count by appId and category)
------------------------------------------------------------------
DROP TABLE IF EXISTS [CONSENT].[dbo].[TEMP_TABLE];

SELECT * INTO [CONSENT].[dbo].[TEMP_TABLE] FROM
(SELECT A.appId, SUM(A.TotalApps) as TotalAppCount FROM
(SELECT appId, Prd_ctgy, count(appId) AS TotalApps FROM
(SELECT appId, CASE
```

```
WHEN xProductCategory LIKE '%Patient Access%' then 'Patient Access'
WHEN ((xProductCategory is null or xProductCategory = '') and userName = 'PROVIDER DIRECTORY') THEN
'PROVIDER DIRECTORY'
END AS Prd_ctgy,
DATENAME(MONTH, eventTime) AS Closing_Month
FROM [CONSENT].[dbo].[tz_security_audit]
where YEAR(eventTime) > 2023 and YEAR(eventTime) <= 2024
AND appId IS NOT NULL
AND tenantId = 'mhc') src1
GROUP BY appId, Prd_ctgy)A
GROUP BY appId) B

Step 2: Populate Temp Table 1 with app Rank:
-----------------------------------------
DROP TABLE IF EXISTS [CONSENT].[dbo].[TEMP_TABLE_1];

SELECT * INTO [CONSENT].[dbo].[TEMP_TABLE_1] FROM
(SELECT appId, TotalAppCount, ROW_NUMBER() OVER(ORDER BY TotalAppCount DESC) AS AppRank from [CONSENT].
[dbo].[TEMP_TABLE]) A

STEP 3: (Joining the pivot table with temp table to get the Top 3 App usage count)
-------------------------------------------------------------------------------
SELECT final.appId, Prd_ctgy, TotalApps
FROM
(SELECT * FROM
(SELECT appId, Prd_ctgy, count(appId) AS TotalApps from
(SELECT appId, CASE
WHEN xProductCategory LIKE '%Patient Access%' then 'Patient Access'
WHEN ((xProductCategory is null or xProductCategory = '') and userName = 'PROVIDER DIRECTORY') THEN
'PROVIDER DIRECTORY'
END AS Prd_ctgy
FROM [CONSENT].[dbo].[tz_security_audit]
where YEAR(eventTime) > 2023 and YEAR(eventTime) <= 2024
AND appId IS NOT NULL
AND tenantId = 'mhc') src1
GROUP BY appId, Prd_ctgy) SRCTBL
) final
INNER JOIN
[CONSENT].[dbo].[TEMP_TABLE_1] T1
ON final.appId = T1.appId
ORDER BY T1.AppRank
```

Developer Portal FAQs for Health Plan Administrators

You can also access additional FAQs from Microsoft here: https://docs.microsoft.com/en-us/azure/api-management/api-management-howto-developer-portal#cors.

Contents:

Show / Hide Contents

1 - What kind of applications can be developed in the Developer Portal?

The Developer Portal supports two main types of applications: third-party healthcare applications and payer-to-payer applications. Third-party applications are typically a member-based smartphone application where members can prove eligibility, check claims and make doctor appointments. Payer-to-payer applications typically are written to transfer member data between two different payers in cases where members change health plans.

The corresponding **roles** for these two application types are: third-party developers and payer developers.

2 - How do third-party and payer developers navigate to the Developer Portal?

The Developer Portal is a public facing internet page. During implementation, this URL is provided to you in case you want to enable a link on your own public-facing website to direct developers to the page. Alternatively, you can share the link in any other way if developers reach out to you directly. Developers can also search for the page using standard internet search engines.

3 - What can a developer do on the portal prior to registering?

After a developer navigates to the Developer Portal URL, they are introduced to the interoperability APIs and the categories of data they can access. In addition, they have the ability to view API documentation for all of the APIs that are offered through the API Gateway. This access also includes request parameters as well as sample data and responses that are viewable in a developer friendly format. After a developer reviews the available APIs, they can then initiate registration to begin their application development and testing.

4 - How does a developer register on the portal?

Registration can occur in two ways:

- You can automate the process automatically after a developer approaches you directly with a request to develop an application against your interoperability APIs. See FAQ 5 for more detail.
- Developers navigate to the “Sign Up” link. There, developers can enter basic user information. After the request is submitted, the Developer Portal performs the following functions:
    - The Developer Portal creates the account.
    - The Developer Portal notifies the developer of the request and your approval process can then begin. See FAQ 5 below for more detail.

      There are APIs available to get information about third-party developers. This information includes a verified email address.

5 - What can a developer do after they are registered?

After registration, a developer can begin calling APIs against mock test data, and they can register their application and begin to build it.

6 - How do I approve a developer to allow them to begin developing an application against your interoperability APIs?

When either a developer approaches you directly with a request to develop, or we notify you during the developer registration process of a request from a developer, your approval process begins. You will approve the developer request following your pre-determined business process, which may include standard attestation statements such as how PHI data will be protected or what secondary usages of data may be. After your business process approves the developer request, you can automate that approval in the Developer Portal by calling an API that then completes the registration process for the developer.

7 - How do I manage developers?

Previous questions have addressed directing developers to the Developer Portal as well as describing the way the developers will be approved to build applications against your APIs. There are APIs available that allow for blocking and deleting user accounts. APIs available that allow a client to remove an account from groups.

Quick Start Guide

Try This and That

1. Start by launching the Developer Portal in a browser. The URL will be specific to your health plan.
2. On the Home page, scroll down to see more content.
3. Back at the top, click on the **Help** menu and explore the documentation links on the Help page.
4. Sign up for a Developer Portal account either as a third-party developer or a payer developer role. Navigate to the **Sign Up** page, fill in the form and submit it, then follow the instructions in the email being sent to you.
5. Sign into the Developer Portal with your credentials.
6. Click on an API name such as "patient-access-sandbox".
7. On the ensuing API page, scroll through the list of API calls and select the "Patient Read" API endpoint you have access to based on your role, and select an API endpoint. For example, **Patient Read**. Explore the description, input parameters and response.
8. Click the **Try It** button on the far right and enter any parameters as needed. For instance, select a patient ID from the **Patient ID** dropdown. Enter any parameters as needed.
9. In the **Authorization** dropdown, select the **Authorization Code** option. This takes you to the Identity Provider (IDP), where the token is generated and populated on the **Try It** page.
10. Scroll farther down the page and click the **SEND** button.
11. If the call succeeds, you'll see "200 OK" returned along with the data from the request.
12. While on the API page, you can download an OpenAPI ("swagger") file to get a start on your application development:
    a. Choose a download format from the **API Definition** dropdown.
    b. The swagger file is downloaded to the **Downloads** folder.

    From there you can pull the downloaded file into Postman or your IDE to start development.