Oracle® Cloud

Using the FHIR Adapter with Oracle Integration 3

F99964-06
June 2025

Oracle Cloud Using the FHIR Adapter with Oracle Integration 3,

F99964-06

Primary Author: Oracle Corporation

Contents

5 Implement Common Patterns Using the FHIR Adapter

Preface

This guide describes how to configure this adapter as a connection in an integration in Oracle Integration.

> **Note:**
>
> The use of this adapter may differ depending on the features you have, or whether your instance was provisioned using Standard or Enterprise edition. These differences are noted throughout this guide.

Topics:

Audience

This guide is intended for developers who want to use this adapter in integrations in Oracle Integration.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `https://www.oracle.com/corporate/accessibility/`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `https://support.oracle.com/portal/` or visit `Oracle Accessibility Learning and Support` if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and

the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related Resources

See these Oracle resources:

- Oracle Cloud at `http://cloud.oracle.com`
- *Using Integrations in Oracle Integration 3*
- *Using the Oracle Mapper with Oracle Integration 3*
- Oracle Integration documentation on the Oracle Help Center.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Understand the FHIR Adapter

Review the following topics to learn about the FHIR Adapter and how to use it as a connection in integrations in Oracle Integration. A typical workflow of adapter and integration tasks is also provided.

Topics:

FHIR Adapter Capabilities

The FHIR Adapter enables you to create an integration between a Fast Healthcare Interoperability Resources (FHIR) application server and Oracle Integration. The FHIR Adapter is similar to the REST Adapter, but with a FHIR-specific configuration experience. You can configure the FHIR Adapter as an invoke connection in an integration in Oracle Integration.

An important aspect of integrating healthcare use cases with Oracle Integration is the ability to allow Oracle Integration to call a FHIR application server. FHIR is a standard used to access and exchange healthcare data. FHIR provides clinicians and patients with access to information in a patient's electronic health records.

The FHIR Adapter provides the following capabilities:

- Supports FHIR version 4.0.1.
- Supports outbound invocations of FHIR endpoints processed by a FHIR application server.
- Supports create, delete, history, history-type, search, patch, read, update, and vread interactions on a FHIR resource.
- Supports the patch interaction to update *specific* fields of a FHIR resource. The patch interaction provides an alternative to the update interaction, which updates *all* fields. A JSON patch document with a content type of `application/json-patch+json` is supported. See the JavaScript Object Notation (JSON) Patch.
  An implementation pattern that describes how to design integrations using the patch interaction is provided. See Update Specific Fields of a FHIR Resource with the Patch Interaction.
- Supports using artificial intelligence (AI) to take your natural language input and return a FHIR-equivalent search query against FHIR-compliant servers such as HAPI FHIR and others. See Create an AI-Generated FHIR Search Query.
- Supports FHIR standard definition resources with the following maturity levels; normative, 5, 4, and 3.
- Supports FHIR profile packages. See Oracle Integration for Healthcare Support for HL7 and FHIR in *Using Oracle Integration for Healthcare in Oracle Integration 3*.

- Supports custom FHIR resources. See Create Custom FHIR Resources in *Using Oracle Integration for Healthcare in Oracle Integration 3*.
- Supports the packaged delivery of all FHIR resources to and from the FHIR base specification.
- Supports custom and standard headers.
- Supports customized query parameters.
- Supports the same security policies for outbound invocations as the REST Adapter. See Configure Connection Security.
- Works exclusively with Oracle Integration for Healthcare to create integrations that interact with healthcare organizations that use FHIR. See Introduction to Oracle Integration for Healthcare in *Using Oracle Integration for Healthcare in Oracle Integration 3*.

The FHIR Adapter is one of many predefined adapters included with Oracle Integration. See the Adapters page in the Oracle Help Center.

FHIR Adapter Restrictions

Note the following FHIR Adapter restrictions.

- Trigger (inbound) connections are not supported. To achieve this functionality, use the REST Adapter.
- The create, read, update, delete, patch, vread, history, and history-type FHIR interactions are supported. Other interactions such as custom operations are not supported.
- Bundle resources with heterogeneous resource objects in the entry array are not supported.
- Primitive extensions and modifier extensions in FHIR profiles are not supported.
- The patch interaction includes the following restrictions:
  - Patch is only supported for standard schema definitions.
  - The copy and move operations on top-level elements are only permitted if the top-level element is an array.
  - All patch operations configured in the Operations table on the Configure Interaction page must have a target node in the mapper. For example, although the remove operation on a non-array element doesn't require any mapping, you must still right-click and select **Create target node**.

> **Note:**
>
> There are overall service limits for Oracle Integration. A service limit is the quota or allowance set on a resource. See Service Limits.

What Application Version Is Supported?

For information about which application version is supported by this adapter, see the Connectivity Certification Matrix.

Workflow to Create and Add a FHIR Adapter Connection to an Integration

You follow a very simple workflow to create a connection with a FHIR Adapter and include the connection in an integration in Oracle Integration.

This table lists the workflow steps for both adapter tasks and overall integration tasks, and provides links to instructions for each step.

Step	Description	More Information
1	Decide where to work	• Work in a project (see why working with projects is preferred in *Using Integrations in Oracle Integration 3*). • Work outside a project.
2	Create the adapter connections for the applications you want to integrate. The connections can be reused in multiple integrations and are typically created by the administrator.	Create a FHIR Adapter Connection
3	Create the integration. When you do this, you add trigger (source) and invoke (target) connections to the integration.	Understand Integration Creation and Best Practices in *Using Integrations in Oracle Integration 3* and Add the FHIR Adapter Connection to an Integration
4	Map data between the trigger connection data structure and the invoke connection data structure.	Map Data in *Using Integrations in Oracle Integration 3*
5	(Optional) Create lookups that map the different values used by those applications to identify the same type of object (such as gender codes or country codes).	Manage Lookups in *Using Integrations in Oracle Integration 3*
6	Activate the integration.	Activate an Integration in *Using Integrations in Oracle Integration 3*
7	Monitor the integration on the dashboard.	Monitor Integrations During Runtime in *Using Integrations in Oracle Integration 3*
8	Track payload fields in messages during runtime.	Assign Business Identifiers for Tracking Fields in Messages and Track Integration Instances in *Using Integrations in Oracle Integration 3*
9	Manage errors at the integration level, connection level, or specific integration instance level.	Manage Errors in *Using Integrations in Oracle Integration 3*

2

FHIR Concepts

The FHIR Adapter supports many capabilities of FHIR. You configure these capabilities for the FHIR Adapter in the Adapter Endpoint Configuration Wizard. This section provides a conceptual overview of key supported capabilities.

Topics:

Supported FHIR Capabilities

This section provides a conceptual overview of key FHIR capabilities supported by the FHIR Adapter.

FHIR

HL7 Fast Healthcare Interoperability Resources (FHIR) is a standard for accessing and exchanging healthcare data between different computer systems regardless of how it is stored in those systems. FHIR has the potential to improve patient care by providing clinicians and patients with timely access to information in a patient's electronic health records. FHIR enables healthcare information, including clinical and administrative data, to be available securely to those who have a need to access it for the benefit of a patient receiving care.

See HL7 FHIR.

FHIR Standard Definition Support

The FHIR standard schema definition is a published set of XSDs that the FHIR standard organization uses to model healthcare resources such as patient, observation, and others. It provides an extension framework that allows different consumers to extend the model to support different attributes while still complying with the schema.

The standard schema definition is automatically selected for you when configuring the FHIR Adapter in the Adapter Endpoint Configuration Wizard. See Perform Actions on a FHIR Resource (Create, Read, Update, and more ...).

FHIR Profile Support

The HL7 FHIR standard defines a set of base resources (for example, patient and observation). The standard base resources have generic definitions. A FHIR profile enables you to create a customized resource definition by specifying a set of constraints and/or extensions on the base resource. Base FHIR resources such as a patient resource can express their conformance to a specific profile. This enables a FHIR server to validate a specific resource against the associated profile definition.

See Import FHIR Profile Packages in *Using Oracle Integration for Healthcare in Oracle Integration 3* and Profiling FHIR.

FHIR Resources

FHIR divides healthcare data into categories such as patients, laboratory results, insurance claims, and many others. Each category is represented by a FHIR resource, which defines the details that comprise an exchangeable patient record. Resources satisfy most common healthcare patient use cases.

See Resource Index.

Oracle Integration supports a number of FHIR resources. You select a supported resource when configuring the FHIR Adapter in the Adapter Endpoint Configuration Wizard. See Perform Actions on a FHIR Resource (Create, Read, Update, and more ...).

FHIR Interactions

FHIR interactions are the set of actions to take on FHIR resources. Interactions can be grouped according to whether they act upon an instance, a type, or the whole system. FHIR supports interactions at the individual instance level (such as getting an individual resource, updating an individual resource, and deleting an individual resource) and interactions at the resource level (such as search and create).

The following interactions are supported:

- Create: Creates a new resource in a FHIR server-assigned location.
  See create.
- Delete: Removes an existing resource.
  See delete.
- History: Retrieve the change history for a particular resource type with an ID specified.
  See history
- History-type: Retrieve the history for a particular resource type.
  See history
- Patch: Updates specific fields of a resource.
  See patch.
- Read: Accesses the current contents of a resource.
  See read.
- Update: Creates a new current version for an existing resource or creates an initial version if no resource already exists for the given ID.
  See update.
- vread: Performs a version-specific read of a resource.
  See vread.

You select the interaction to perform when configuring the FHIR Adapter in the Adapter Endpoint Configuration Wizard. See Invoke Basic Info Page and Perform Actions on a FHIR Resource (Create, Read, Update, and more ...).

Standard and Custom Request and Response Headers

FHIR provides a number of standard headers that control the way the FHIR server returns data. FHIR also specifies a syntax for defining custom headers.

See HTTP Headers and Custom Headers.

You can select standard headers and create custom request and response headers when configuring the FHIR Adapter in the Adapter Endpoint Configuration Wizard. See Perform Actions on a FHIR Resource (Create, Read, Update, and more ...).

Search Parameters

A filter parameter can be used with the search operation. Filter requests have access to the same set of search parameters that are available to the search operation in that resource context on the FHIR server.

See Filter Parameter and Search.

You can specify search parameters when configuring the FHIR Adapter in the Adapter Endpoint Configuration Wizard. See Perform Actions on a FHIR Resource (Create, Read, Update, and more ...).

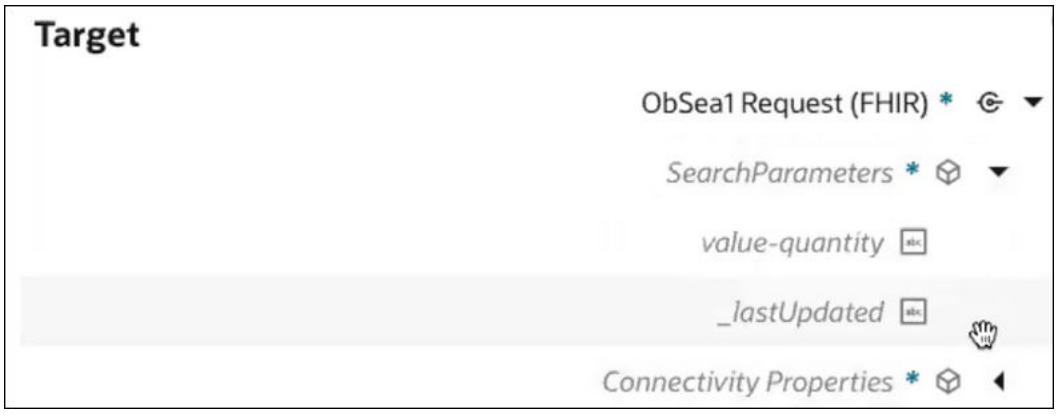
Bundle Support

Search results are returned in a bundle, which is another resource within the FHIR standard definition. A bundle is a predefined container resource that can contain an array of entries of the same resources (homogenous set of resource instances) or different resources (heterogeneous mixture of resource instances).

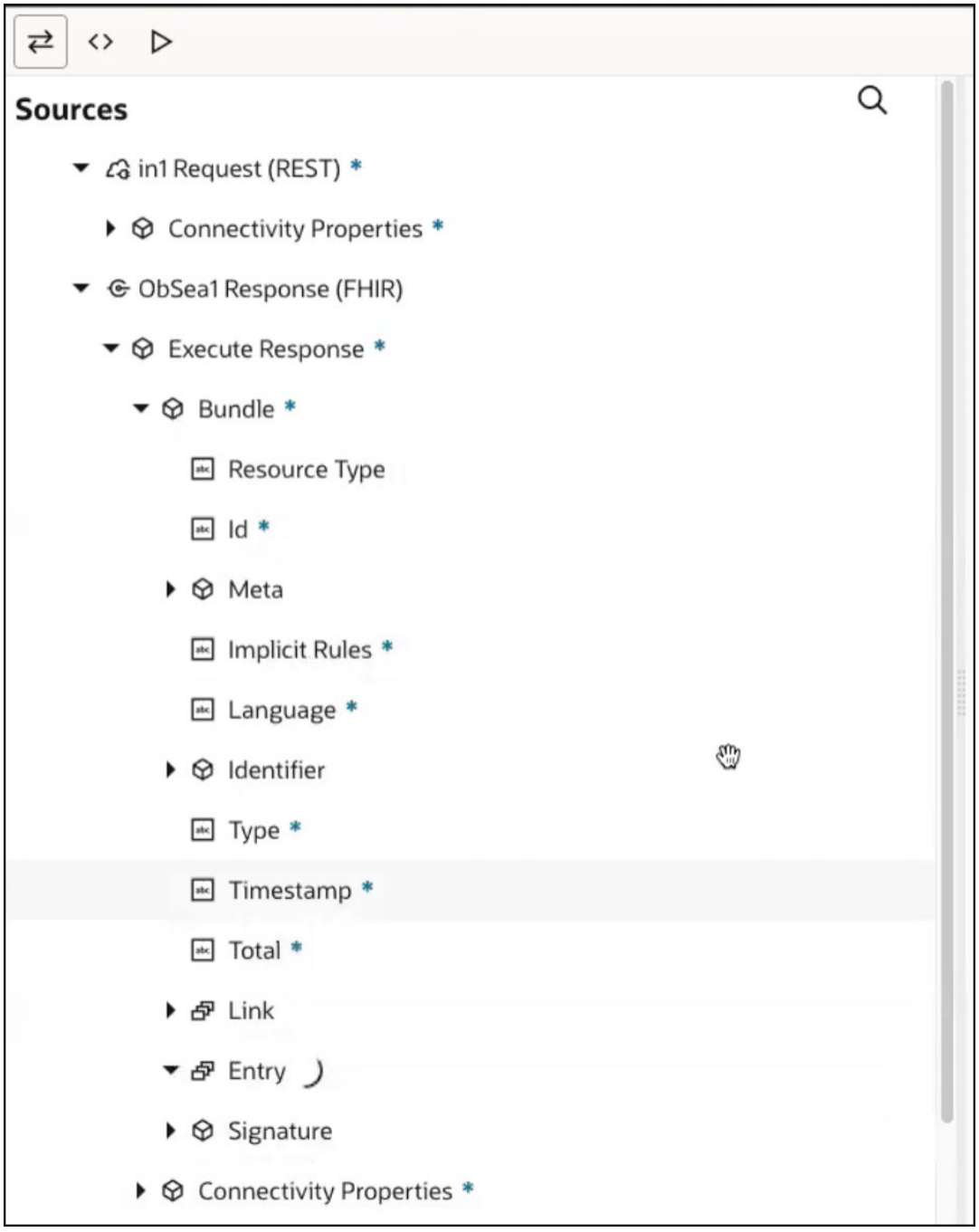
The bundle creation for the case of search or other type of interactions that may use the bundle can be internal and does not need any user selection or configuration. The only selection that drives the overall schema creation is the resource selection.

The following example shows the response schema generated by the FHIR Adapter for a search result with a homogenous set of resource instances in the mapper.

- For the request, the search parameters in the mapper look as follows:



- For the response, the bundle in the mapper looks as follows:


Sources
in1 Request (REST) *
Connectivity Properties *
ObSea1 Response (FHIR)
Execute Response *
Bundle *
Resource Type
Id *
Meta
Implicit Rules *
Language *
Identifier
Type *
Timestamp *
Total *
Link
Entry
Signature
Connectivity Properties *

3

Create a FHIR Adapter Connection

A connection is based on an adapter. You define connections to the specific cloud applications that you want to integrate.

Topics:

Prerequisites for Creating a Connection

You must satisfy the following prerequisites to create a connection with the FHIR Adapter. The FHIR Adapter is similar to the REST Adapter and supports many of the same security policies.

AI-Generated Search Query

To enter search queries in natural language format and generate a FHIR-equivalent search query against a FHIR-compliant server, you must first subscribe to and configure an AI engine. FHIR search queries work with the Cohere and OpenAI keys. See Subscribe to and Configure the AI Engine to Use in *Using Integrations in Oracle Integration 3*.

OAuth Security Policies Use

If you are using one of the OAuth security policies, you must already have registered your client application to complete the necessary fields on the Connections page. The Basic Authentication and No Security Policy security policies are exempted.

Before a client application can request access to resources on a resource server, the client application must first register with the authorization server associated with the resource server.

The registration is typically a one-time task. Once registered, the registration remains valid, unless the client application registration is revoked.

At registration time, the client application is assigned a client ID and a client secret (password) by the authorization server. The client ID and secret are unique to the client application on that authorization server. If a client application registers with multiple authorization servers (for example, Facebook, Twitter, and Google), each authorization server issues its own unique client ID to the client application.

```
@ref: http://tutorials.jenkov.com/oauth2/authorization.html
```

For OAuth configuration, read the provider documentation carefully and provide the relevant values.

SSL Endpoints Use

For SSL endpoints, obtain and upload a server certificate. See Upload a Certificate to Connect with External Services.

JWT Assertions Outbound Use

Perform the following prerequisites to use JWT assertions.

- Manually create a signing key for upload on the Certificates page. See Upload a Certificate to Connect with External Services.
  The service provider typically provides instructions on how to generate the signing keys and the format. For an example, see Required Keys and OCIDs.
- Create the JWT header and JWT payload JSON files. You upload both files on the Connections page when configuring the FHIR Adapter to support JWT assertions. See Prerequisites for Creating a Connection in *Using the REST Adapter with Oracle Integration 3*.

Create a Connection

Before you can build an integration, you must create the connections to the applications with which you want to share data.

> **Note:**
>
> You can also create a connection in the integration canvas. See Define Inbound Triggers, Outbound Invokes, and Actions.

To create a connection in Oracle Integration:

1. Decide where to start:
   - Work in a project (see why working with projects is preferred).
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.
     d. In the **Connections** section, click **Add** if no connections currently exist or **+** if connections already exist. The Create connection panel opens.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Connections**.
     b. Click **Create**. The Create connection panel opens.
2. Select the adapter to use for this connection. To find the adapter, scroll through the list, or enter a partial or full name in the **Search** field.
3. Enter the information that describes this connection.

Element	Description
Name	Enter a meaningful name to help others find your connection when they begin to create their own integrations.
Identifier	Automatically displays the name in capital letters that you entered in the **Name** field. If you modify the identifier name, don't include blank spaces (for example, `SALES OPPORTUNITY`).
Role	Select the role (direction) in which to use this connection. **Note**: *Only* the roles supported by the adapter you selected are displayed for selection. Some adapters support all role combinations (trigger, invoke, or trigger and invoke). Other adapters support fewer role combinations. When you select a role, only the connection properties and security policies appropriate to that role are displayed on the Connections page. If you select an adapter that supports both invoke and trigger, but select only one of those roles, you'll get an error when you try to drag the adapter into the section you didn't select. For example, assume you configure a connection for the Oracle Service Cloud (RightNow) Adapter as only an **invoke**. Dragging the adapter to a **trigger** section in the integration produces an error.
Keywords	Enter optional keywords (tags). You can search on the connection keywords on the Connections page.
Description	Enter an optional description of the connection.
Share with other projects	**Note**: This field only appears if you are creating a connection in a project. Select to make this connection publicly available in other projects. Connection sharing eliminates the need to create and maintain separate connections in different projects. When you configure an adapter connection in a different project, the **Use a shared connection** field is displayed at the top of the Connections page. If the connection you are configuring matches the same type and role as the publicly available connection, you can select that connection to reference (inherit) its resources. See Add and Share a Connection Across a Project.

4. Click **Create**.

   Your connection is created. You're now ready to configure the connection properties, security policies, and (for some connections) access type.

5. Follow the steps to configure a connection.
   The connection property and connection security values are specific to each adapter. Your connection may also require configuration with an access type such as a private endpoint or an agent group.

6. Test the connection.

Configure Connection Properties

Enter connection information so your application can process requests.

1. Go to the **Properties** section.
2. In the **Connection URL** field, enter the FHIR server base URL to use.

   `https://FHIR_server/fhir/r4/`

3. If you want to specify optional details for TLS, two-way SSL, or an identity keystore alias name, click **Optional properties**.

Element	Description
TLS Version	If no value is selected, the default value used for outbound connections is Transport Layer Security (TLS) version 1.3. It's up to your discretion and the end application's requirements to select either TLS version 1.2 or 1.1 as the default. • **TLSv1.1** • **TLSv1.2** TLSv1 is no longer supported. If you previously configured a connection in a version prior to Oracle Integration 3 to use TLSv1.1, either update the connection by not selecting a value for this field or select TLSv1.2. The TLS protocol provides privacy and data integrity between two communicating computer applications. For trigger-only connections, you cannot select a TLS version. Oracle Integration accepts what it receives as long as it's TLSv1.1 or TLSv1.2.
Enable two way SSL for outbound connections (Optional)	If you are configuring the FHIR Adapter for use with a two-way SSL-enabled server, select **Yes**. .
Identity keystore alias name (Optional)	Enter the key alias name from the keystore file that you specified when importing the identity certificate. The alias name to provide must match the name provided for the private key entry in the JKS file.

Configure Connection Security

Configure security for your FHIR Adapter connection.

1. Go to the **Security** section.
2. Select the security policy and specify the required details.

Selected Security Policy	Fields
Basic Authentication	• **Username** — The name of a user who has access to the destination web service. • **Password** — Enter the password. • **Confirm Password** — Reenter the password.

Selected Security Policy	Fields
OAuth Client Credentials	• **Access Token URI** — The URL from which to obtain the access token. • **Client Id** — The client identifier issued to the client during the registration process. • **Client Secret** — The client secret. • **Confirm Client Secret** — Reenter the client secret. • **Scope** — The scope of the access request. Scopes enable you to specify which type of access you need. Scopes limit access for the OAuth token. They do not grant any additional permission beyond that which the user already possesses. • **Auth Request Media Type** — The format of the data you want to receive. This is an optional parameter that can be kept blank. For example, if you are invoking Twitter APIs, you do not need to select any type. • **Client Authentication** — You can optionally configure OAuth flows with client authentication. This is similar to the Postman user interface feature for configuring client authentication. – **Send client credentials as basic auth header**: Pass the client ID and client secret in the header as basic authentication. – **Send client credentials in body**: Pass the client ID and client secret in the body as form fields.
OAuth Authorization Code Credentials	• **Client Id** — The client identifier issued to the client during the registration process. • **Client Secret** — The client secret. • **Confirm Client Secret** — Reenter the client secret. • **Authorization Code URI** — The URI from which to request the authorization code. • **Access Token URI** — URI to use for the access token. • **Scope** — The scope of the access request. Scopes enable you to specify which type of access you need. Scopes limit access for the OAuth token. They do not grant any additional permission beyond that which the user already possesses. • **Client Authentication** — You can optionally configure OAuth flows with client authentication. This is similar to the Postman user interface feature for configuring client authentication. – **Send client credentials as basic auth header**: Pass the client ID and client secret in the header as basic authentication. – **Send client credentials in body**: Pass the client ID and client secret in the body as form fields.

Selected Security Policy	Fields

OAuth Custom Three Legged Flow	• **Authorization Request** — The client application URL to which you are redirected when you provide consent. The authorization server sends a callback to Oracle Integration to obtain an access token for storage. When you create your client application, you must register a redirect URI where the client application is listening.
	• **Access Token Request** — The access token request to use to fetch the access token. Specify the request using CURL syntax. For example:
	`-X POST method -H headers -d string_data access_token_uri?query_parameters`
	• **Refresh Token Request** — The refresh token request to use to fetch the access token. This request refreshes the access token if it expires. Specify the request using CURL syntax. For example
	`-X POST method -H headers -d string_data refresh_token_uri?query_parameters`
	• **Sauth_code** — Use regex to identify the authorization code.
	`code`
	• **Saccess_token** — Use a regular expression (regex) to retrieve the access token.
	`access.[tT]oken`
	• **Srefresh_token** — Use regex to retrieve the refresh token.
	`refresh.[tT]oken`
	• **Sexpiry** — Use regex to identify when the access token expires.
	`expires_in`
	• **Stoken_type** — Use regex to identify the access token type.
	`token.?[tT]ype`
	• **access_token_usage** — Specify how to pass the token as multiple headers or multiple query parameters to access a protected resource. You cannot pass a mix of headers and query parameters.
	For headers:
	`-H Authorization: ${token_type} ${access_token} -H validity: 30000 -H signature: ok`

Selected Security Policy	Fields

	You can optionally specify quotes for headers:
	
	`-H 'Authorization: ${token_type} ${access_token}' -H 'validity: 30000' -H 'signature: ok'`
	
	For query parameters:
	
	`?token=${access_token}&validity=3000&signature=ok`

Selected Security Policy	Fields

OAuth Custom Two Legged Flow	• **Access Token Request** — The access token request to use to fetch the access token. Specify the request using CURL syntax. For example:
	
	`-X POST method -H headers -d string_data access_token_uri?query_parameters`
	
	• **Refresh Token Request** — The refresh token request to use to fetch the access token. This request refreshes the access token if it expires. Specify the request using CURL syntax. For example
	
	`-X POST method -H headers -d string_data refresh_token_uri?query_parameters`
	
	• **Saccess_token** — Use regex to identify the access token.
	
	`access.[tT]oken`
	
	• **Srefresh_token** — Use regex to identify the refresh token.
	
	`refresh.[tT]oken`
	
	• **Sexpiry** — Use regex to identify when the access token expires.
	
	`expires_in`
	
	• **Stoken_type** — Use regex to identify the access token type.
	
	`token.?[tT]ype`
	
	• **access_token_usage** — Specify how to pass the token as multiple headers or multiple query parameters to access a protected resource. You cannot pass a mix of headers and query parameters.
	For headers:
	
	`-H Authorization: ${token_type} ${access_token} -H validity: 30000 -H signature: ok`
	
	You can optionally specify quotes for headers:
	
	`-H 'Authorization: ${token_type} ${access_token}' -H 'validity: 30000' -H 'signature: ok'`
	
	For query parameters:
	
	`?token=${access_token}&validity=3000&signature=ok`

Selected Security Policy	Fields
OAuth Client Credentials using JWT Client Assertion **Note**: This policy is typically used to invoke application-driven APIs.	• **Access token URI** — Enter the URL to which to send a request to obtain the access token. For example: `https://accounts.google.com/o/oauth2/token` • **JWT headers in JSON format** — Upload the JWT header file in JSON format. • **JWT payload in JSON format** — Upload the JWT payload file in JSON format. • **JWT private key alias** — Enter the JWT private key alias. This is the same alias you specified when uploading the signing key certificate on the Certificates page. • **Scope** — (Optional) Enter the scopes. • **Access token request** — (Optional) Enter the request to obtain the access token. The format you specify can vary by service provider. See Variations of JWT Usage by Service Providers in *Using the REST Adapter with Oracle Integration 3*.
OAuth using JWT User Assertion **Note**: This policy is typically used on behalf of a user.	• **Access token URI** — Enter the URL to which to send a request to obtain the access token. For example: `https://accounts.google.com/o/oauth2/token` • **JWT headers in JSON format** — Upload the JWT header file in JSON format. • **JWT payload in JSON format** — Upload the JWT payload file in JSON format. • **JWT private key alias** — Enter the JWT private key alias. This is the same alias you specified when uploading the signing key certificate on the Certificates page. • **Scope** — (Optional) Enter the scopes. • **Access token request** — (Optional) Enter the request to obtain the access token. The format you specify can vary by service provider.
No Security Policy	If you select this security policy, no additional fields are displayed. For example, you need to access a HAPI FHIR server that requires no security policy.

Configure the Endpoint Access Type

Configure access to your endpoint. Depending on the capabilities of the adapter you are configuring, options may appear to configure access to the public internet, to a private endpoint, or to an on-premises service hosted behind a fire wall.

Select the Endpoint Access Type

1. Go to the **Access type** section.
2. Select the option for accessing your endpoint.

Option	This Option Appears If Your Adapter Supports ...
Public gateway	Connections to endpoints using the public internet.

Option	This Option Appears If Your Adapter Supports ...
Connectivity agent	Connections to on-premises endpoints through the connectivity agent. a. Click **Associate agent group**. The Associate agent group panel appears. b. Select the agent group, and click **Use**. To configure an agent group, you must download and install the on-premises connectivity agent. See Download and Run the Connectivity Agent Installer and About Creating Hybrid Integrations Using Oracle Integration in *Using Integrations in Oracle Integration 3*.

Test the Connection

Test your connection to ensure that it's configured successfully.

1. In the page title bar, click **Test**. What happens next depends on whether your adapter connection uses a Web Services Description Language (WSDL) file. Only some adapter connections use WSDLs.

If Your Connection...	Then...
Doesn't use a WSDL	The test starts automatically and validates the inputs you provided for the connection.
Uses a WSDL	A dialog prompts you to select the type of connection testing to perform: • **Validate and Test**: Performs a full validation of the WSDL, including processing of the imported schemas and WSDLs. Complete validation can take several minutes depending on the number of imported schemas and WSDLs. No requests are sent to the operations exposed in the WSDL. • **Test**: Connects to the WSDL URL and performs a syntax check on the WSDL. No requests are sent to the operations exposed in the WSDL.

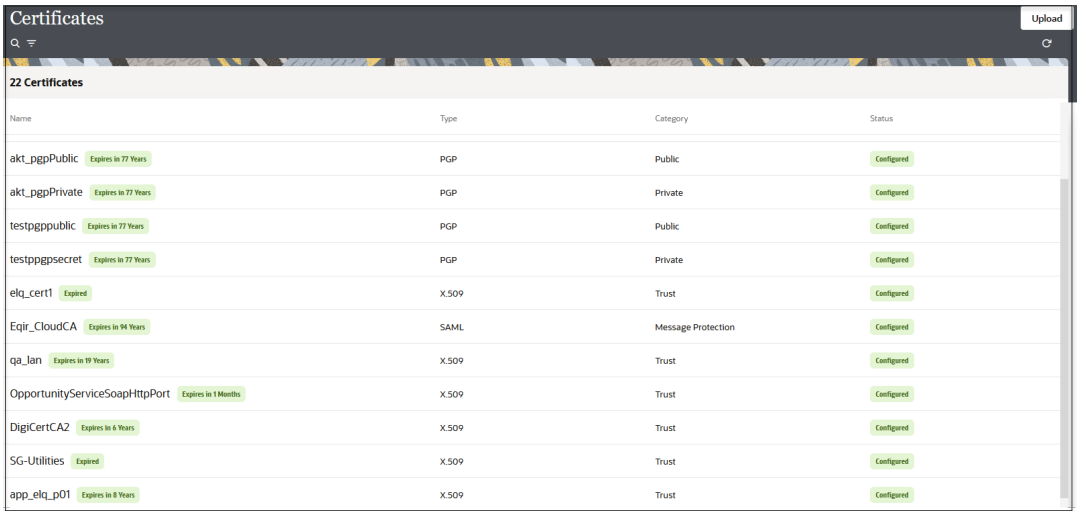
2. Wait for a message about the results of the connection test.
   - If the test was successful, then the connection is configured properly.
   - If the test failed, then edit the configuration details you entered. Check for typos and verify URLs and credentials. Continue to test until the connection is successful.
3. When complete, click **Save**.

Upload a Certificate to Connect with External Services

Certificates allow Oracle Integration to connect with external services. If the external service/endpoint needs a specific certificate, request the certificate and then import it into Oracle Integration.

If you make an SSL connection in which the root certificate does not exist in Oracle Integration, an exception error is thrown. In that case, you must upload the appropriate certificate. A certificate enables Oracle Integration to connect with external services. If the external endpoint requires a specific certificate, request the certificate and then upload it into Oracle Integration.

1. Sign in to Oracle Integration.
2. In the navigation pane, click **Settings**, then **Certificates**.
   All certificates currently uploaded to the trust store are displayed on the Certificates page.
3. Click **Filter** to filter by name, certificate expiration date, status, type, category, and installation method (user-installed or system-installed). Certificates installed by the system cannot be deleted.



4. Click **Upload** at the top of the page.
   The Upload certificate panel is displayed.
5. Enter an alias name and optional description.
6. In the **Type** field, select the certificate type. Each certificate type enables Oracle Integration to connect with external services.
   - Digital Signature
   - X.509 (SSL transport)
   - SAML (Authentication & Authorization)
   - PGP (Encryption & Decryption)
   - Signing key

Digital Signature

The digital signature security type is typically used with adapters created with the Rapid Adapter Builder. See Learn About the Rapid Adapter Builder in Oracle Integration in *Using the Rapid Adapter Builder with Oracle Integration 3*.

1. Click **Browse** to select the digital certificate. The certificate must be an X509Certificate. This certificate provides inbound RSA signature validation. See RSA Signature Validation in *Using the Rapid Adapter Builder with Oracle Integration 3*.
2. Click **Upload**.

X.509 (SSL transport)

1. Select a certificate category.
   a. **Trust**: Use this option to upload a trust certificate.

i. Click **Browse**, then select the trust file (for example, `.cer` or `.crt`) to upload.

b. **Identity**: Use this option to upload a certificate for two-way SSL communication.

i. Click **Browse**, then select the keystore file (`.jks`) to upload.

ii. Enter the comma-separated list of passwords corresponding to key aliases.

> **Note:**
>
> When an identity certificate file (`.jks`) contains more than one private key, all the private keys must have the same password. If the private keys are protected with different passwords, the private keys cannot be extracted from the keystore.

iii. Enter the password of the keystore being imported.

c. Click **Upload**.

SAML (Authentication & Authorization)

1. Note that **Message Protection** is automatically selected as the only available certificate category and cannot be deselected. Use this option to upload a keystore certificate with SAML token support. Create, read, update, and delete (CRUD) operations are supported with this type of certificate.
2. Click **Browse**, then select the certificate file (`.cer` or `.crt`) to upload.
3. Click **Upload**.

PGP (Encryption & Decryption)

1. Select a certificate category. Pretty Good Privacy (PGP) provides cryptographic privacy and authentication for communication. PGP is used for signing, encrypting, and decrypting files. You can select the private key to use for encryption or decryption when configuring the stage file action.

a. **Private**: Uses a private key of the target location to decrypt the file.

i. Click **Browse**, then select the PGP file to upload.

ii. Enter the PGP private key password.

b. **Public**: Uses a public key of the target location to encrypt the file.

i. Click **Browse**, then select the PGP file to upload.

ii. In the **ASCII-Armor Encryption Format** field, select **Yes** or **No**.

- **Yes** shows the format of the encrypted message in ASCII armor. ASCII armor is a binary-to-textual encoding converter. ASCII armor formats encrypted messaging in ASCII. This enables messages to be sent in a standard messaging format. This selection impacts the visibility of message content.
- **No** causes the message to be sent in binary format.

iii. From the **Cipher Algorithm** list, select the algorithm to use. Symmetric-key algorithms for cryptography use the same cryptographic keys for both encryption of plain text and decryption of cipher text. The following supported cipher algorithms are FIPS-compliant:

- AES128
- AES192

- AES256
- TDES

c. Click **Upload**.

Signing key

A signing key is a secret key used to establish trust between applications. Signing keys are used to sign ID tokens, access tokens, SAML assertions, and more. Using a private signing key, the token is digitally signed and the server verifies the authenticity of the token by using a public signing key. You must upload a signing key to use the OAuth Client Credentials using JWT Client Assertion and OAuth using JWT User Assertion security policies in REST Adapter invoke connections. Only PKCS1- and PKCS8-formatted files are supported.

1. Select **Public** or **Private**.
2. Click **Browse** to upload a key file.
   If you selected **Private**, and the private key is encrypted, a field for entering the private signing key password is displayed after key upload is complete.
3. Enter the private signing key password. If the private signing key is not encrypted, you are not required to enter a password.
4. Click **Upload**.

4

Add the FHIR Adapter Connection to an Integration

When you drag the FHIR Adapter into the invoke area of an integration, the Adapter Endpoint Configuration Wizard is invoked. This wizard guides you through configuration of the FHIR Adapter endpoint properties.

The following sections describe the wizard pages that guide you through configuration of the FHIR Adapter as an invoke in an integration.

Topics:

Invoke Basic Info Page

Specify a name, description, and action for the invoke connection.

Element	Description
What do you want to call your endpoint?	Provide a meaningful name so that others can understand the responsibilities of this connection. You can include English alphabetic characters, numbers, underscores, and hyphens in the name. You can't include the following characters: • No blank spaces (for example, `My Inbound Connection`) • No special characters (for example, `#;83&` or `rig(t)now4`) except underscores and hyphens • No multimeter characters
What does this endpoint do?	Enter an optional description of the connection's responsibilities.

Element	Description
What would you want to do?	Select an option to perform: • **Perform actions on a FHIR Resource (Create, Read, Update, and more ...)**: Enables you to perform one of the following interactions: – create: Creates a new resource. (for example, `POST /Patient` with the patient details in the request body). – delete: Removes an existing resource (for example, `DELETE /Patient/123`). – patch: Updates specific fields of a resource (for example, `PATCH /Patient/123`). – read: Accesses the current contents of a resource by its unique ID (for example, `GET /Patient/123` fetches the patient with ID `123`). – update: Creates a new current version for an existing resource or creates an initial version if no resource already exists for the given ID (for example, `PUT /Patient/123` for a full update). – vread: Reads a resource (for example, patient), the ID of the resource, and the version of the resource. You map this information in the mapper. • **Search FHIR Resource**: Enables you to perform a GET or POST operation. • **Search FHIR Resource using AI**: **Note**: This option is not available for selection if your Oracle Integration instance is in a region that cannot access the Oracle Cloud Infrastructure Generative AI Service. For information about which regions can access the Oracle Cloud Infrastructure Generative AI Service, see AI Innovation and Oracle Integration in *Using Integrations in Oracle Integration 3*. Enables you to enter a search query in natural language format to automatically generate a FHIR-equivalent search query against a FHIR server. • **Fetch History**: Enables you to perform one of the following history-related interactions: – history: Fetches the history of an *individual* in a particular resource. For example, all the history related to a specific patient ID in the Patient resource. You map this information in the mapper. – history-type: Fetches the entire history of a particular resource. For example, the entire history for *all* patient in the Patient resource.

Configure Interaction Page

See the section based on your selection on the Basic Info page.

Topics:

Perform Actions on a FHIR Resource (Create, Read, Update, and more ...)

This page appears if you selected **Perform actions on a FHIR Resource (Create, Read, Update, and more ...)** on the Basic Info page.

In the **FHIR Schema Source** field, select the schema source to use.

Element	Description
FHIR Schema Source	Select a source: • **Standard Schema Definition** • **FHIR Profile** • **Custom FHIR resource**

Based on your FHIR schema source selection, see the following section for a description of fields:

- If You Select Standard Schema Definition
- If You Select FHIR Profile
- Custom FHIR resource

If You Select Standard Schema Definition

Element	Description
FHIR Resource	Select one of the supported resources. See Resource Index.
FHIR Interaction	Select the operation to perform. • **update** • **patch** Available only with **Standard Schema Definition**. • **read** • **create** • **delete** • **vread**
Configure Response This field is displayed if you selected **create**, **update**, **patch**, or **delete** from the **FHIR Interaction** list.	Select the response to receive from the FHIR server. • **representation**: Returns the default response. • **minimal**: Returns the simplest possible server response, which typically includes an HTTP status code and potentially minimal resource or error information, without unnecessary details. • **Operation outcome**: Returns a set of error, warning, and information messages that provide detailed information about the outcome of an attempted system operation.

Element	Description
Operations This field is displayed if you selected **patch** from the **FHIR Interaction** list.	Select the patch operations to perform on the selected FHIR resource. The resource elements available for selection are based on the FHIR resource selected. You can define multiple operations and resource elements in your request. • Click **Add**, then double-click inside the row to select the operation and resource element. For example, to add a name to a **Patient** FHIR resource, select an **Operation** of **add** and a **Resource Element** of **name**. The patch operations that you add are populated in the mapper for you to map. Most of the resource elements available for selection are known as top-level elements. While these elements are sufficient for most cases, you may have a need to patch more fine-grained, child-level resource elements. The **FHIR Path Param** resource element is available for these scenarios. Use cases that describe how to select and map both top-level elements and child-level elements are provided. See Update Specific Fields of a FHIR Resource with the Patch Interaction.
Configure Request Headers	Select to add request HTTP headers. • **Standard** • **Custom**
Configure Response Headers	Select to add response HTTP headers. • **Standard** • **Custom**

If You Select FHIR Profile

Element	Description
FHIR Profile Package	Select a profile package. The selections that appear were imported on the FHIR profiles page. See Import a FHIR Profile Package in *Using Oracle Integration for Healthcare in Oracle Integration 3*.
FHIR Resource	Select the FHIR resource. The resources that are displayed are based on the profile package you selected.
Profile	Select the FHIR profile. The profiles that are displayed are based on the resource you selected.
FHIR Interaction	Select the operation to perform. • **create** • **delete** • **read** • **update** • **vread**
Configure Response This field is displayed if you selected **create**, **update**, or **delete** from the **FHIR Interaction** list.	Select the response to receive from the FHIR server. • **representation**: Returns the default response. • **minimal**: Returns the simplest possible server response, which typically includes an HTTP status code and potentially minimal resource or error information, without unnecessary details. • **Operation outcome**: Returns a set of error, warning, and information messages that provide detailed information about the outcome of an attempted system operation.

Element	Description
Configure Request Headers	Select to add request HTTP headers. • **Standard** • **Custom**
Configure Response Headers	Select to add response HTTP headers. • **Standard** • **Custom**

Custom FHIR resource

Element	Description
FHIR Custom Resource	Select the FHIR custom resource. The selections that appear were created on the Custom FHIR resources page. See Create Custom FHIR Resources in *Using Oracle Integration for Healthcare in Oracle Integration 3*.
FHIR Interaction	Select the operation to perform. • **create** • **delete** • **read** • **update** • **vread**
Configure Response This field is displayed if you selected **update**, **create**, or **delete** from the **FHIR Interaction** list.	Select the response to receive from the FHIR server. • **representation**: Returns the default response. • **minimal**: Returns the simplest possible server response, which typically includes an HTTP status code and potentially minimal resource or error information, without unnecessary details. • **Operation outcome**: Returns a set of error, warning, and information messages that provide detailed information about the outcome of an attempted system operation.
Configure Request Headers	Select to add request HTTP headers. • **Standard** • **Custom**
Configure Response Headers	Select to add response HTTP headers. • **Standard** • **Custom**

Search FHIR Resource

This page appears if you selected **Search FHIR Resource** on the Basic Info page.

Element	Description
Request FHIR Resource	Select one of the supported resources. See Resource Index.
Response FHIR Source	Select a source: • **Standard Schema Definition** • **FHIR Profile** • **Custom FHIR resource**

- If You Select Standard Schema Definition

- If You Select FHIR Profile
- If You Select Custom FHIR resource

If You Select Standard Schema Definition

Element	Description
HTTP Verb	Select the operation to perform. • **POST** • **GET**
Configure Request Headers	Select to add request HTTP headers. • **Standard** • **Custom**
Configure Response Headers	Select to add response HTTP headers. • **Standard** • **Custom**
Search Parameters	Specify search parameters for the FHIR endpoint. Click the **Add** icon to display a row for entering the parameter name. Click the **Remove** icon to delete a selected row. See Filter Parameter.

If You Select FHIR Profile

Element	Description
FHIR Profile Package	Select a profile package. The selections that appear were imported on the FHIR profiles page. See Import a FHIR Profile Package in *Using Oracle Integration for Healthcare in Oracle Integration 3*.
Profile	Select the FHIR profile. The profiles that are displayed are based on the resource you selected.
Configure Request Headers	Select to add request HTTP headers. • **Standard** • **Custom**
Configure Response Headers	Select to add response HTTP headers. • **Standard** • **Custom**
Search Parameters	Specify search parameters for the FHIR endpoint. Click the **Add** icon to display a row for entering the parameter name. Click the **Remove** icon to delete a selected row. See Filter Parameter.

For conceptual information about the fields described on this page, see Supported FHIR Capabilities.

If You Select Custom FHIR resource

Element	Description
Response Bundle	Select the custom FHIR resource heterogeneous bundle. The selections that appear were created on the Custom FHIR resources page. See Create Custom FHIR Resources in *Using Oracle Integration for Healthcare in Oracle Integration 3*.
HTTP Verb	Select the operation to perform. • **POST** • **GET**
Configure Request Headers	Select to add request HTTP headers. • **Standard** • **Custom**
Configure Response Headers	Select to add response HTTP headers. • **Standard** • **Custom**
Search Parameters	Specify search parameters for the FHIR endpoint. Click the **Add** icon to display a row for entering the parameter name. Click the **Remove** icon to delete a selected row. See Filter Parameter.

Configure AI search

This page appears if you selected **Search FHIR Resource using AI** on the Basic Info page. Enter a search query in natural language format to automatically generate a FHIR-equivalent search query against FHIR-compliant servers such as HAPI FHIR and more. Query parameters and model values are also automatically generated and available for additional editing.

> **Note:**
>
> This page is not visible if your Oracle Integration instance is in a region that cannot access the Oracle Cloud Infrastructure Generative AI Service. For information about which regions can access the Oracle Cloud Infrastructure Generative AI Service, see AI Innovation and Oracle Integration in *Using Integrations in Oracle Integration 3*.
> If you subscribed to an AI engine and created AI-generated FHIR search queries on this page in release 25.04, and your Oracle Integration instance cannot access the Oracle Cloud Infrastructure Generative AI Service, your previous search queries are visible and backward compatible with the **Search FHIR Resource** option.

Natural language search queries are only supported for standard schema definitions. Natural language queries are not supported for FHIR profiles or custom FHIR resources.

Element	Description

Description Box	Enter a query in natural language format. For example:
	
	`Find all patients named Mark in the city of San Francisco`
	
	Note the following guidelines:
	• Write the query in simple English language.
	• Write the statement clearly and explicitly.
	• Do not include quotes ("") or any other special characters.
	• The query must be related to FHIR search topics. If not, you receive the following message:
	
	`This query is not related to FHIR Search. I can only assist with FHIR Search topics.`

Submit Query	Click to automatically generate the query. For the natural language text entered above, the following query is created in the **Generated FHIR URL** field. The query shows the server you configured in the **Connection URL** field on the Connections page, appended with query parameters derived from your text input. For this example, parameters were created for `name` and `city`. `https://hapi.fhir.org/baseR4/Patient?name=Mark&address-city=SanFrancisco` See Configure Connection Properties.

Test Query	Click to validate your query.
	• If successful, the following message is displayed. Click the message to view details about the FHIR server search response.
	
	`Valid FHIR URL. Click here to see the results.`
	
	• If unsuccessful, the following message is displayed. Click the message to view error details.
	
	`Invalid FHIR query. Please resubmit the query or modify query parameters manually.`
	`Click here to see the error.`
	
	To improve performance, the test query limits the number of records returned.
	Additional validation also occurs when you click **Continue** to move to the next page in the Adapter Endpoint Configuration Wizard. Your query must be valid before you can successful move to the next page.

FHIR Resource	Displays the resource automatically generated for your query. The resource (for this example, `Patient`) is displayed for the generated query `https://hapi.fhir.org/baseR4/Patient?name=Mark&address-city=SanFrancisco`.

Element	Description

Query Parameters	The query parameters and model values automatically generated for your query are displayed. If a parameter or model value requires updating, you can make changes.
	• Click **Add** to display a row for entering a new parameter name and model value. Press **Enter** when done.
	• Click **Remove** to delete a selected parameter and model value.
	• Double-click a selected parameter to modify the name or model value. Press **Enter** when done.
	After completing any changes, click **Update Query** to regenerate the query shown in the **Generated FHIR URL** field. Click **Test Query** to revalidate the query.
	Note: Guardrails are in place to prevent the use of invalid query parameters. For example, if you add a new query parameter (for this example, `PatientName`) without testing the query and click **Continue** to move to the next page in the Adapter Endpoint Configuration Wizard, you receive an error:
	`Configured search parameters {PatientName} are invalid to resource type Patient.`
	`Please update them as per the specification and retry.`

Fetch History

This page appears if you selected **Fetch History** on the Basic Info page.

Element	Description
Request FHIR Resource	Select one of the supported resources. See Resource Index.
FHIR Schema Source	Select a source: • **Standard Schema Definition** • **FHIR Profile** • **Custom FHIR resource**

Based on your FHIR schema source selection, see the following section for a description of fields:

- If You Select Standard Schema Definition
- If You Select FHIR Profile
- If You Select Custom FHIR resource

If You Select Standard Schema Definition

Element	Description
FHIR Interaction	Select the interaction to perform. • **history**: Fetches the history of an individual in a particular resource. For example, all the history related to a specific patient ID in the Patient resource. You map this information in the mapper. • **history-type**: Fetches the entire history of a particular resource. For example, *all* patient histories in the Patient resource. For this reason, no mapping is required because the entire resource is fetched.
Configure Request Headers	Select to add request HTTP headers. • **Standard** • **Custom**
Configure Response Headers	Select to add response HTTP headers. • **Standard** • **Custom**

If You Select FHIR Profile

Element	Description
FHIR Profile Package	Select a profile package. The selections that appear were imported on the FHIR profiles page. See Import a FHIR Profile Package in *Using Oracle Integration for Healthcare in Oracle Integration 3*.
Profile	Select the FHIR profile. The profiles that are displayed are based on the resource you selected.
FHIR Interaction	Select the interaction to perform. • **history**: Fetches the history of an individual in a particular resource. For example, all the history related to a specific patient ID in the Patient resource. You map this information in the mapper. • **history-type**: Fetches the entire history of a particular resource. For example, *all* patient histories in the Patient resource. For this reason, no mapping is required because the entire resource is fetched.
Configure Request Headers	Select to add request HTTP headers. • **Standard** • **Custom**
Configure Response Headers	Select to add response HTTP headers. • **Standard** • **Custom**

If You Select Custom FHIR resource

Element	Description
FHIR Custom Resource	Select the custom FHIR resource. The selections that appear were created on the Custom FHIR resources page. See Create Custom FHIR Resources in *Using Oracle Integration for Healthcare in Oracle Integration 3*.

Element	Description
Configure Request Headers	Select to add request HTTP headers. • **Standard** • **Custom**
Configure Response Headers	Select to add response HTTP headers. • **Standard** • **Custom**

For conceptual information about the fields described on this page, see Supported FHIR Capabilities.

Configure Request Headers Page

This page appears if you selected to add standard or custom HTTP request headers.

Element	Description
Configure Request Headers	• For standard HTTP headers, click **Add**, then double-click the row to display a drop-down list with the following selections: – **If-Match** – **If-Modified-Since** – **If-None-Match** – **If-None-Exist** • For custom HTTP headers, click **Add**, then double-click to add a custom header name and description. See HTTP Headers and Custom Headers.

Configure Response Headers Page

This page appears if you selected to add standard or custom HTTP response headers.

Element	Description
Configure Response Headers	• For standard HTTP headers, click **Add**, then double-click the row to display a drop-down list with the following selections: – **ETag** – **Last Modified** – **Location** – **Content Location** • For custom HTTP headers, click **Add**, then double-click to add a custom header name and description. See HTTP Headers and Custom Headers.

Summary Page

You can review the specified adapter configuration values on the Summary page.

Element	Description
Summary	Displays a summary of the configuration values you defined on previous pages of the wizard. The information that is displayed can vary by adapter. For some adapters, the selected business objects and operation name are displayed. For adapters for which a generated XSD file is provided, click the XSD link to view a read-only version of the file. To return to a previous page to update any values, click the appropriate tab in the left panel or click **Go back**. To cancel your configuration details, click **Cancel**.

5

Implement Common Patterns Using the FHIR Adapter

You can use the FHIR Adapter to implement the following common patterns.

Topics:

Create an AI-Generated FHIR Search Query

You can enter search queries in natural language format and generate a FHIR-equivalent search query against FHIR-compliant servers such as HAPI FHIR and others. You can then edit the generated query parameters and regenerate the query, as needed. FHIR-generated search queries give you a head start on building queries that meet your business requirements.

This section provides an overview of creating and editing several search queries (one simple and one more complex).

Ensure that you have first satisfied the prerequisites. See AI-Generated Search Query.

1. Configure a FHIR Adapter invoke connection.
2. Drag the FHIR Adapter invoke connection into the integration canvas.
3. On the Basic Info page, enter a name, enter an optional description, and select **Search FHIR Resource using OIC for Healthcare Copilot**.
4. On the Configure OIC for Healthcare Copilot page, perform the following:
   a. From the **Select AI Engine** list, select your licensed AI engine.
   b. Enter the search query in English that you want to execute against the FHIR server, then click **Submit Query**. For this example:

      ```
      Search all patients named Marcus from the city of Houston
      ```

      The generated FHIR query is displayed:

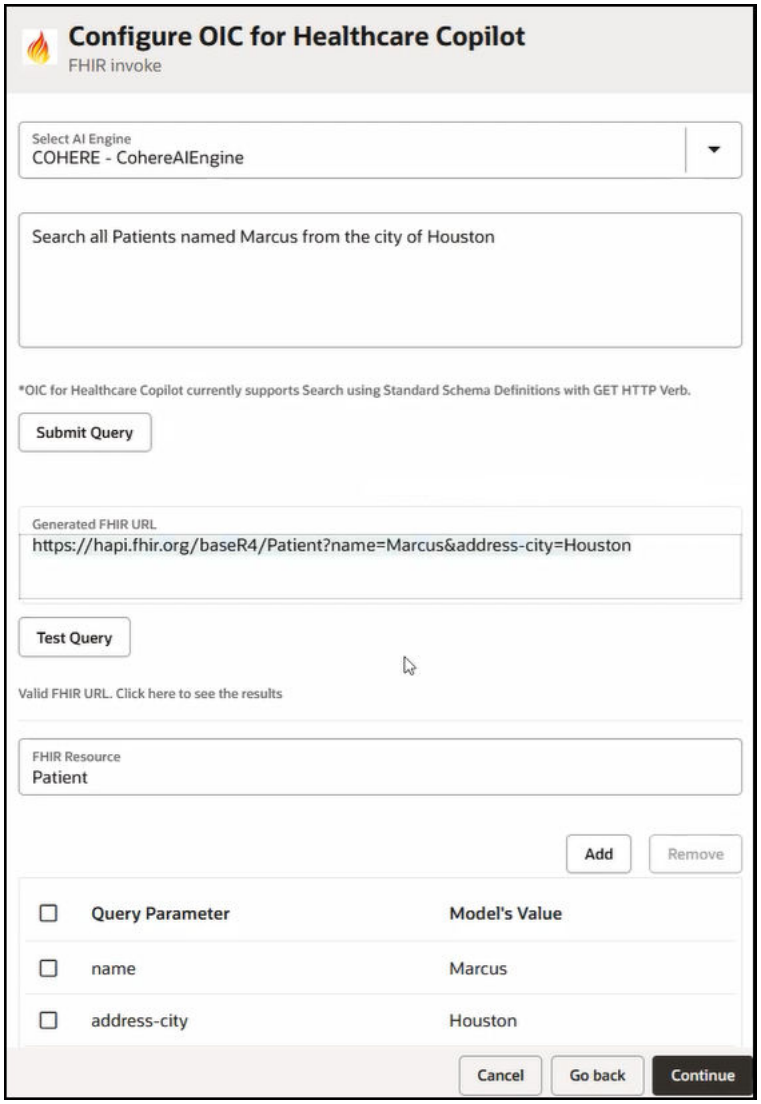
      ```
      https://hapi.fhir.org/baseR4/Patient?name=Marcus&address-city=Houston
      ```

5. Click **Test Query** to validate the query.

   If successful, the following message is displayed.

   ```
   Valid FHIR URL. Click here to see the results.
   ```

   The Configure OIC for Healthcare Copilot page looks as follows:



6. Click the `Valid FHIR URL. Click here to see the results` message to scroll through details about the FHIR search query response.

7. View the automatically generated query parameters and model values.
8. If necessary, add, update, or delete query parameters, then click **Update Query** to regenerate the query in the **FHIR Generated URL** field.

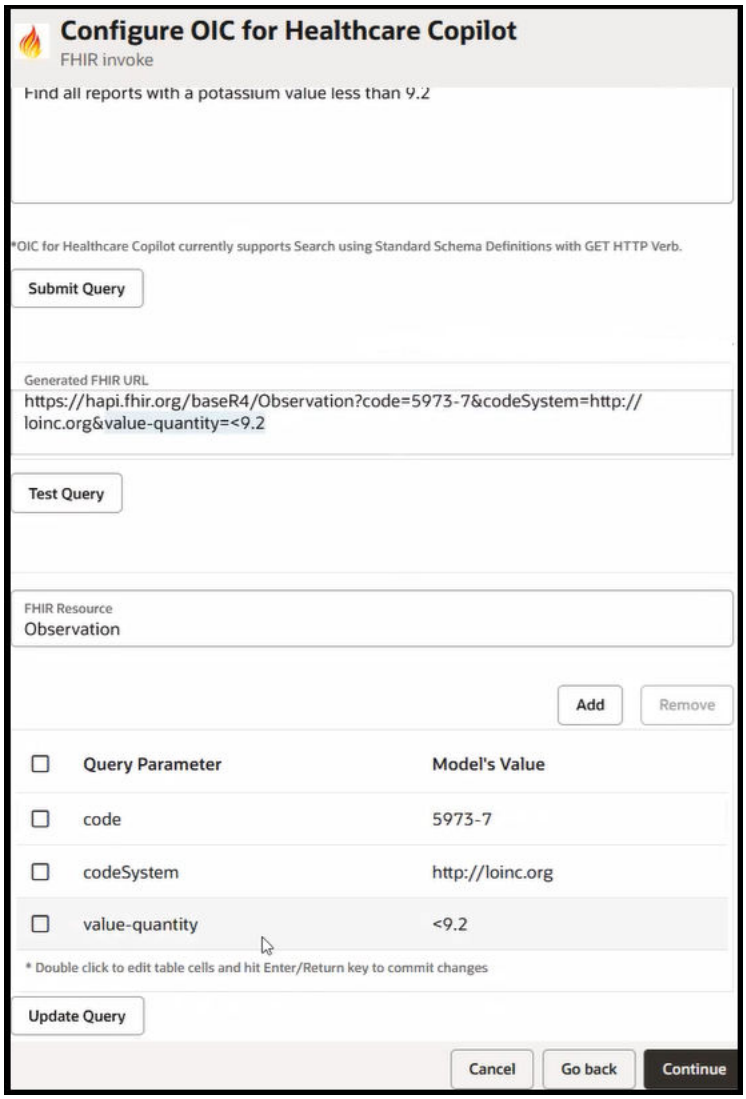
   Let's now generate a second, more complex query.
9. Enter a query, then click **Submit Query**. For this example:

   ```
   Find all reports with a potassium value less than 9.2
   ```

   The generated FHIR query is displayed with a resource of `Observation`.

   ```
   https://hapi.fhir.org/baseR4/Observation?code=5973-7&codeSystem=http://
   loinc.org&value-quantity=<9.2
   ```

   The query parameters are also generated. For this example, the search query was able to automatically generate several complex query parameters and values.



10. Review the query parameters closely in case updates are required. Generated search queries are not always 100% correct. For example:

    Change:

    ```
    value-quantity       <9.2
    ```

    to:

    ```
    value-quantity:exact 9.2
    ```

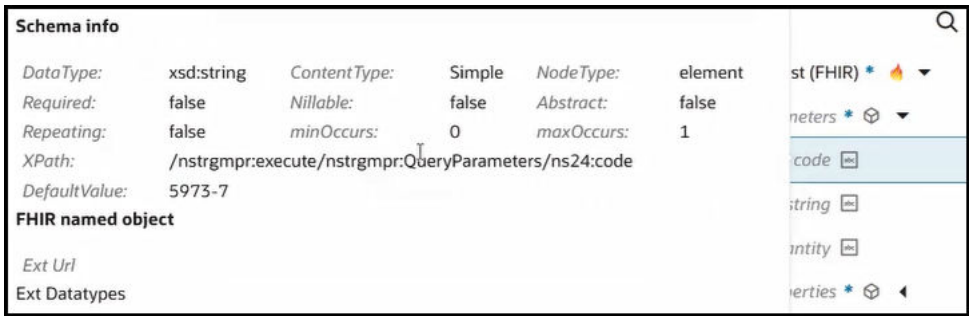
11. Click **Update Query**.

    The generated query is also updated. This functionality provides you with the flexibility to delete, modify, or add new query parameters, update the query, and also test the query as needed.

    ```
    https://hapi.fhir.org/baseR4/Observation?code=5973-7&codeSystem=http://
    loinc.org&value-quantity:exact=9.2
    ```

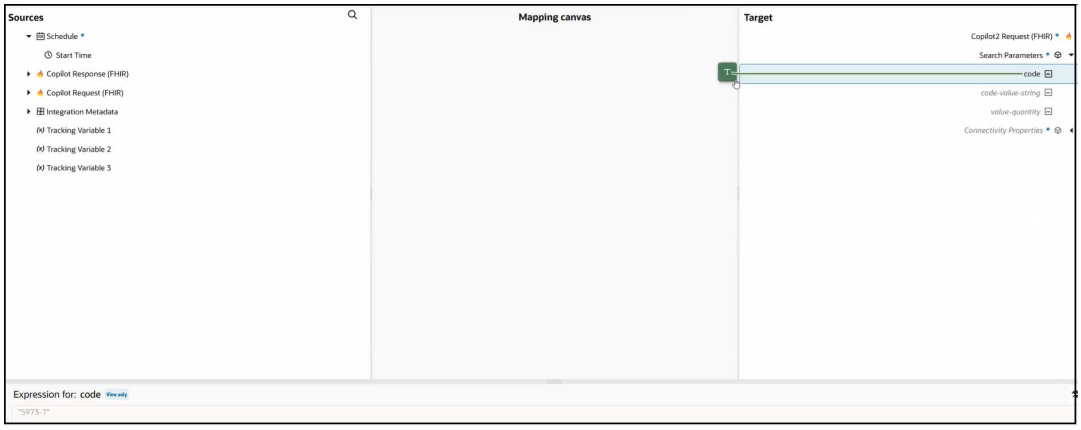
12. When complete, click **Continue**.
13. Click **Finish**.
14. Open the mapper. The query parameters and their values are visible in the target elements.
15. Right-click an element (for this example, `code`) and select **Node info**.

    Note that the default model value of `5973-7` generated by the FHIR search query is displayed.



16. Right-click the same element again (`code`) and select **Add default value**.

    Note that the default model value of `5973-7` generated by the FHIR search query is also automatically mapped and displayed in the Expression Builder.



You can also map appropriate source elements to the target query parameter elements, as needed.

Update Specific Fields of a FHIR Resource with the Patch Interaction

You can update specific fields of a FHIR resource with the patch interaction. This section provides use cases that describe how to patch both top-level elements and more granular, child-level elements in a FHIR resource.

A FHIR resource (for example, **Patient**) consists of an entire schema tree that includes top-level elements and child-level elements.

You can patch top-level elements. For example, a top-level **name** element can be comprised of child-level elements such as **first name**, **last name**, and others. You can only patch **name** in its entirety because you get the entire schema tree.

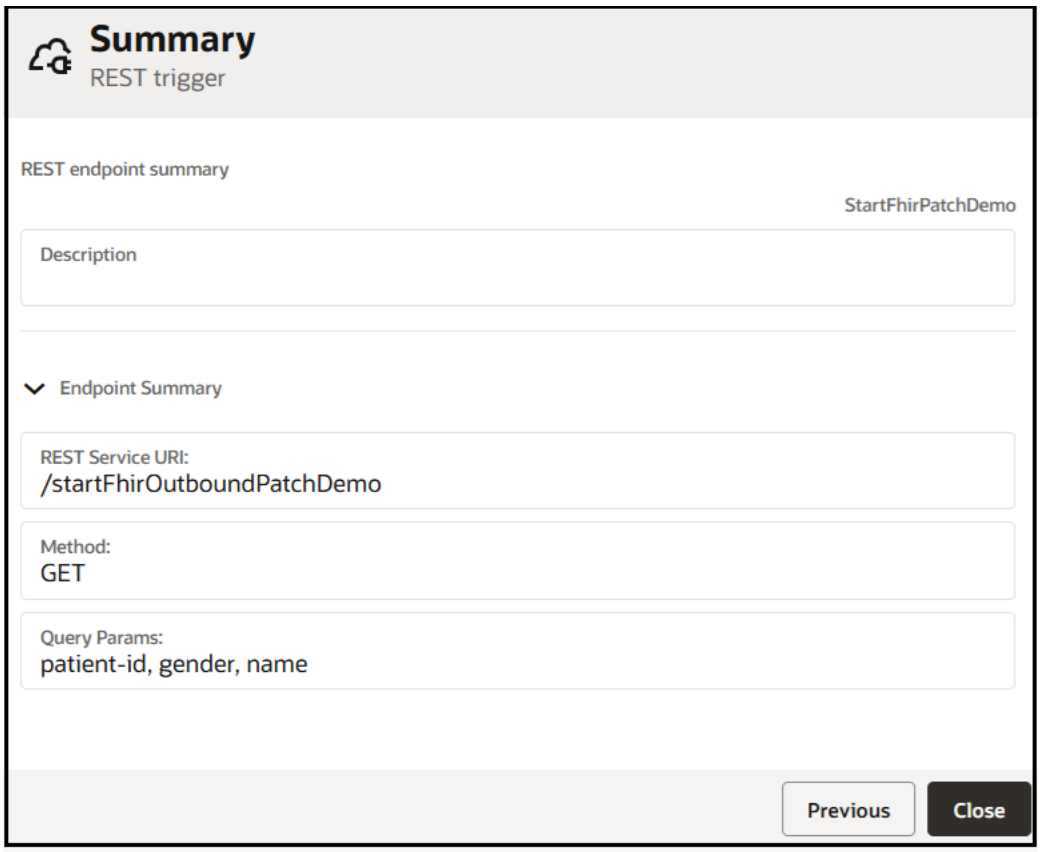
If you need to patch any child-level element in the entire schema tree, you must use the **FHIR Patch Param** resource element.

- Patch Top-Level Elements of the FHIR Resource
- Patch Child-Level Elements of the FHIR Resource

Patch Top-Level Elements of the FHIR Resource

This use case provides a design-time to runtime overview of how to configure patch interaction operations to test the specific gender in a resource, add a name to a resource, and move an address in a resource. You then run the integration to view the results in the activity stream. This use case specifically describes how to patch top-level elements of a FHIR resource (for example, adding a top-level name to a resource, instead of adding a child-level first name, last name, and family name to a resource).

1. Configure a REST Adapter trigger connection and FHIR Adapter invoke connection.
2. Drag the REST Adapter into the integration canvas and specify the configuration details. When complete, the following details are defined:
   - REST service URL of **/startFhirOutboundPatchDemo**.
   - HTTP method of **GET**.
   - Three defined query parameters: **patient-id**, **gender**, and **name**.

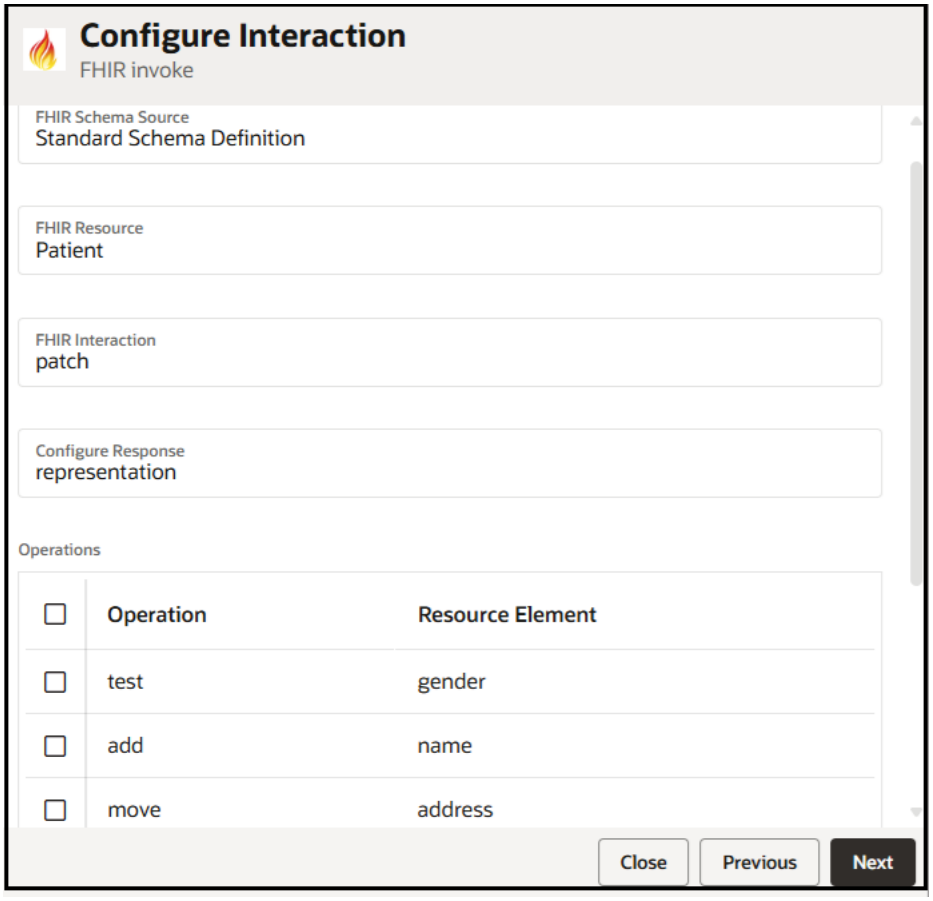


3. Drag the FHIR Adapter into the integration canvas.
4. On the Basic Info page, enter a name (for this example, `PatchInvoke`) and optional description, and select **Perform Actions on a FHIR Resource (Create, Read, Update, and more ...)**.
5. On the Configure Interaction page, specify the following:

Element	Description
FHIR Schema Source	**Standard Schema Definition** **Note**: Only **Standard Schema Definition** is supported for the patch interaction.
FHIR Resource	**Patient**
FHIR Interaction	**patch**
Configure Response	**representation**

Element	Description
Operations	Select the following three patch operations and resource elements to perform on the **Patient** FHIR resource. • Operation one: – **Operation**: **test** – **Resource Element**: **gender** • Operation two: – **Operation**: **add** – **Resource Element**: **name** • Operation three: – **Operation**: **move** – **Resource Element**: **address**

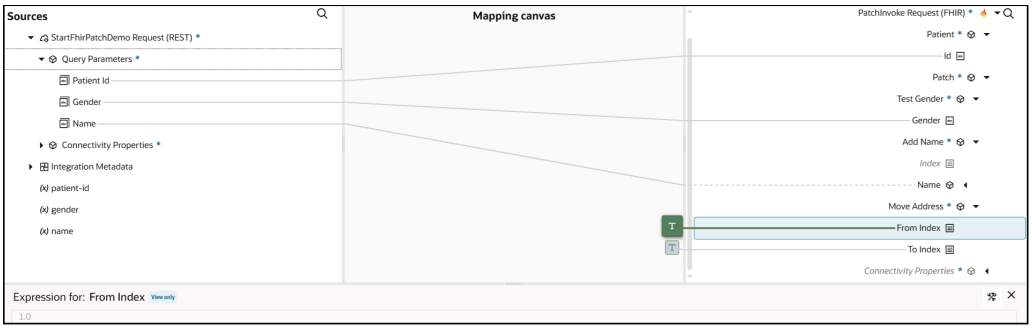
The Configure Interaction page looks as follows:



6. Open and configure the mapper to pass the ID of the **Patient** resource to patch.
7. Configure the operations you defined on the Configure Interaction page:
   - Map the source **Name** to the target **Name**.
     This mapping is to add the name of a patient. Note that **Add Name** appears under an **Index** element because it is an array in the **Patient** resource. You add a new name to a specific index in that array, which prevents the name from being added to the end of the array. If you don't specify the index, the name is added to the end of the array.
   - Map the source **Gender** to the target **Gender**.

This mapping is simply to test that the specific patient gender matches with what is specified at runtime. **Test Gender** does not appear under an **Index** because it is not an array in the **Patient** resource.

- Define the address to move as literal strings in the Expression Builder.
  Note that **Move Address** also appears under an **Index** because it is an array in the **Patient** resource. You must specify the indexes from which and to which to move the address. When you perform a move operation, you are always moving (reordering) elements of arrays.

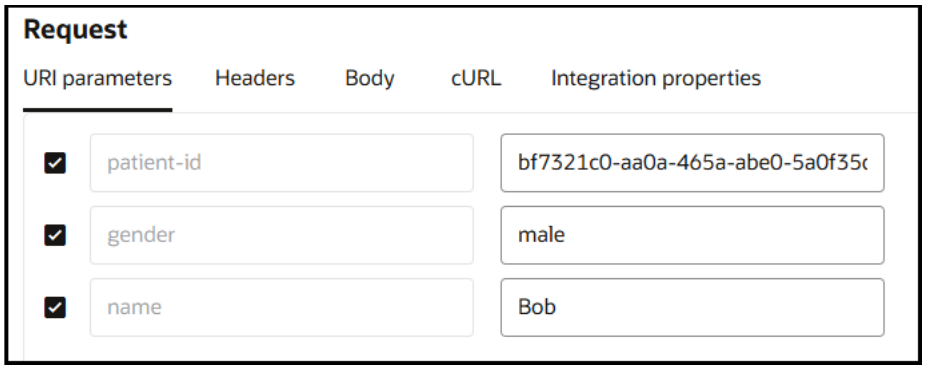


A view of key parts of the existing JSON FHIR `Patient` resource prior to patching shows one current name (`John`), the gender (`male`), and two addresses.

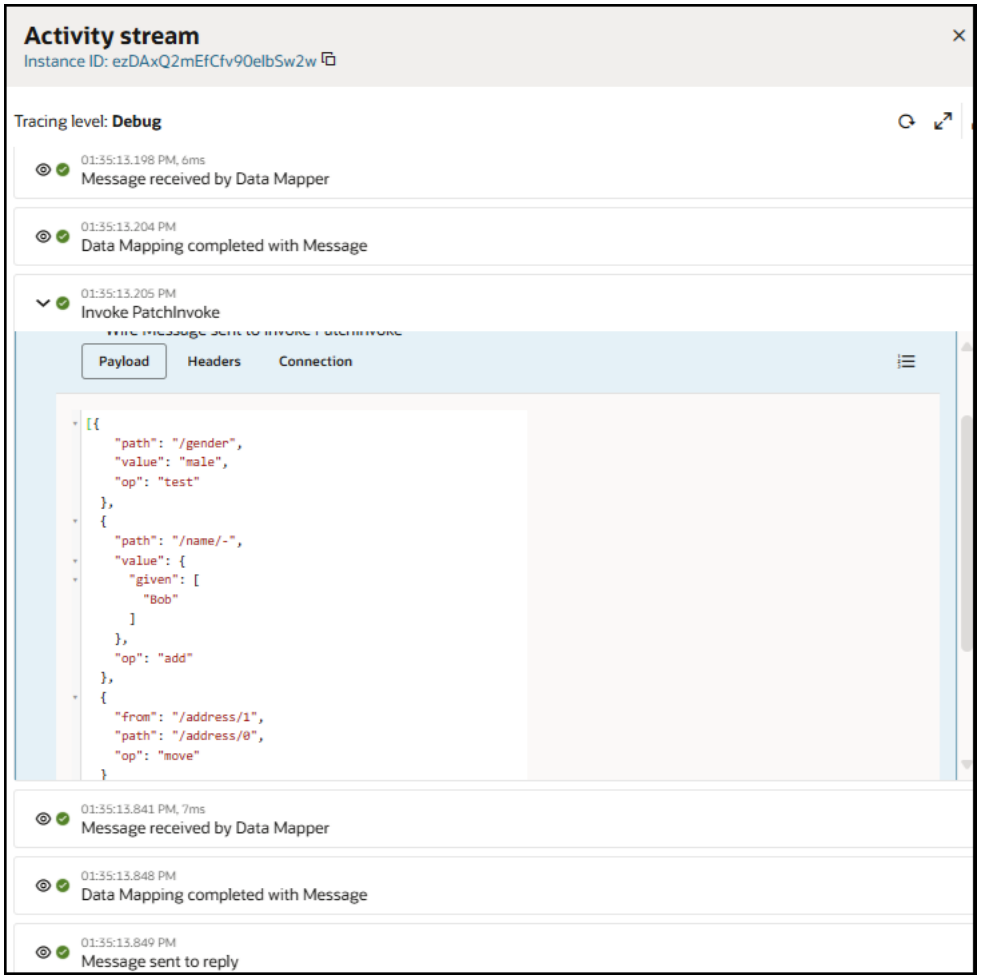
```
{
    "resourceType": "Patient",
    "name": [
        {
            "use": "official",
            "family": "Doe",
            "given": [
                "John"
            ]
        }
    ],
    "telecon": [
        {
            "system": "phone",
            "value": "999 888 7777"
        }
    ],
    "birthdate": "1980-01-01",
    "id": "bf7321c0-aa0a-465a-abe0-5a0f35da709a",
    "gender":  "male",
    "meta": {
        "versionId": "54",
        "lastUpdated": "2025-02-12T08:27:08.776973414+00:00"
    },
    "address": [
          {
            "city": "Chicago"
          },
          {
            "city": "Austin"
          }
```

```
            }
        }
.....
.....
```

8. Save and activate the integration.
9. Select ⋮, and then **Run**.
10. Specify the following parameter values to patch the **Patient** FHIR resource.
    - Add the ID of the **Patient** resource.
    - Test that the gender specified matches with `male` in the FHIR resource. If it does, patching continues. If `female` is instead specified, patch operations stop and the instance run fails with a `value did not match` error.
    - Add a new name of `Bob` to the FHIR resource.



    The operation to move the address of `Austin` above `Chicago` in the FHIR resource was specified as literal strings in the mapper. Therefore, nothing needs to be specified here.
11. When processing completes, expand **invoke PatchInvoke** in the activity stream. The JSON patch payload is visible and consists of the three operations to test the gender, add the name, and move the address.



A review of the JSON-based `Patient` resource shows that the name of `Bob` was added, the gender of `male` was tested successfully, and `Austin` was moved (reordered) to the first array in the index above `Chicago`.

```
{
    "resourceType": "Patient",
    "name": [
        {
            "use": "official",
            "family": "Doe",
            "given": [
                "John"
            ]
        }
            "given": [
                "Bob"
            ]
        }
    ],
    "telecon": [
        {
            "system": "phone",
            "value": "999 888 7777"
        }
```

```
    ],
    "birthdate": "1980-01-01",
    "id": "bf7321c0-aa0a-465a-abe0-5a0f35da709a",
    "gender":  "male",
    "meta": {
        "versionId": "54",
        "lastUpdated": "2025-02-12T08:27:08.776973414+00:00"
    },
    "address": [
          {
            "city": "Austin"
          },
          {
            "city": "Chicago"
          }
       }
    }
.....
.....
```


Patch Child-Level Elements of the FHIR Resource

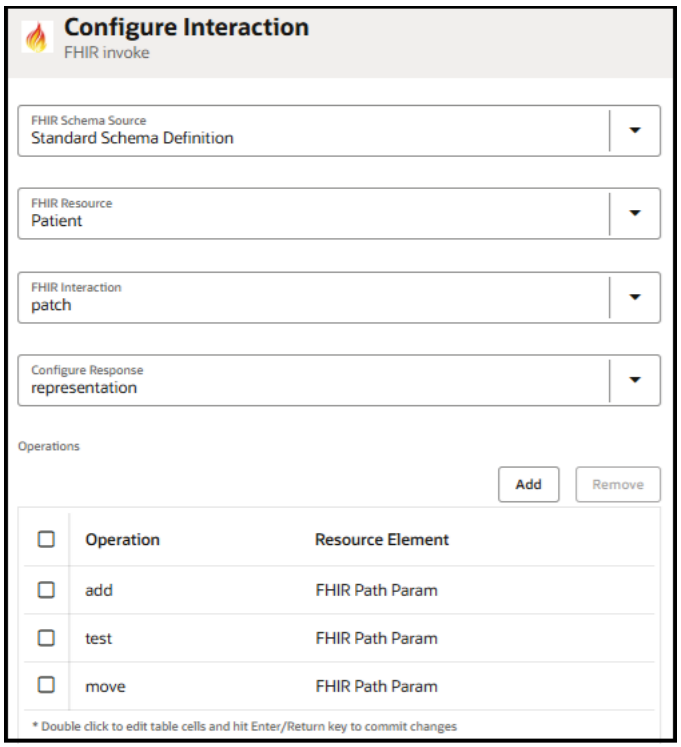
This use case provides a design-time overview of how to configure patch interaction operations for child-level elements in the selected FHIR resource.

You may have scenarios in which you need to perform more granular, child-level element patching in the FHIR resource. For example, you need to patch the first and last names (child-level elements) of a patient, instead of just the name (top-level element). To patch child-level elements, you select the **FHIR Path Param** resource element on the Configure Interaction page during FHIR Adapter configuration. The **FHIR Path Param** resource element works with paths, instead of indexes. The **FHIR Path Param** resource element enables you to literally specify the path to any FHIR resource, child-level element in the mapper. The steps to patch child-level elements in the mapper are slightly more detailed than with patching top-level elements. The following steps provide an overview of child-level patching.

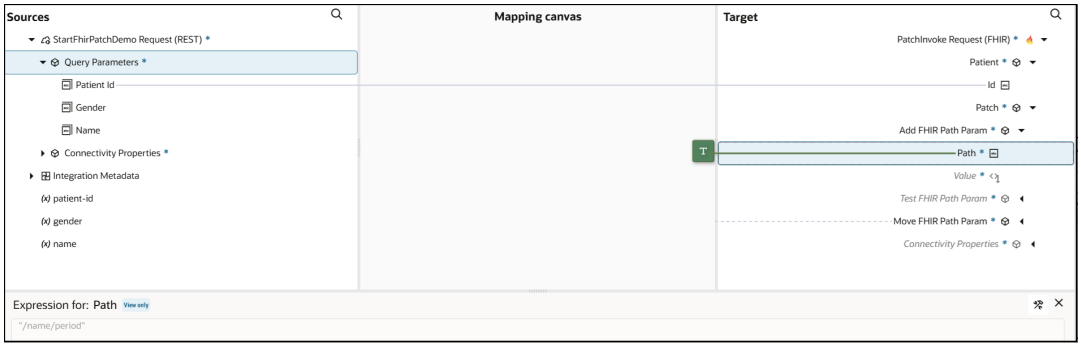
1. Drag the FHIR Adapter invoke connection into the integration canvas.
2. On the Basic Info page, enter a name and optional description, and select **Perform Actions on a FHIR Resource (Create, Read, Update, and more ...)**.
3. On the Configure Interaction page, specify the following:

Element	Description
FHIR Schema Source	**Standard Schema Definition**
FHIR Resource	**Patient**
FHIR Interaction	**patch**
Configure Response	**representation**

Element	Description
Operations	Select the following patch operations and resource elements to perform on the **Patient** FHIR resource. The **FHIR Path Param** resource element lets you literally specify the path to a child-level element to patch in the FHIR resource. This action provides a more fine-grained approach to patching and does not restrict you to patching only top-level elements. • Operation one: – **Operation**: **add** – **Resource Element**: **FHIR Path Param** • Operation two: – **Operation**: **test** – **Resource Element**: **FHIR Path Param** • Operation three: – **Operation**: **move** – **Resource Element**: **FHIR Path Param**

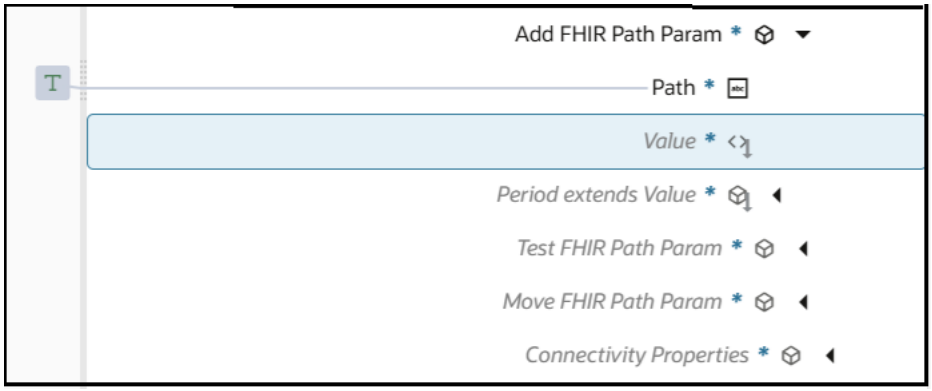


4. Open and configure the mapper to pass the ID of the **Patient** resource to patch.
5. Expand **Add FHIR Path Param**.
   Note that **Add FHIR Path Param** (for patching child-level elements) appears under the **Path** element. This differs from the first use case described in Patch Top-Level Elements of the FHIR Resource in which **Add Name** (for patching top-level elements) appears under the **Index** element. The **Path** element requires you to perform a different set of mapper tasks.
6. Select **Path**, then right-click and select **Create target node** to explicitly specify the path in the Expression Builder.

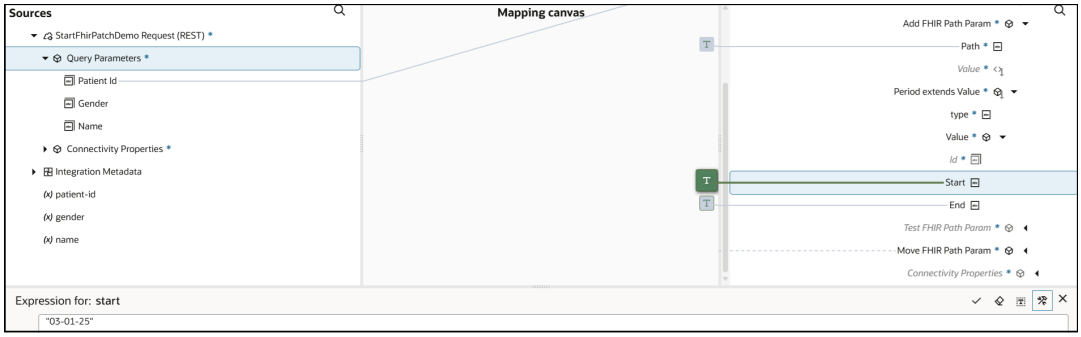


You must now select the corresponding data type of that element.

7. Right-click **Value** and select **Extended Data Types**, then **Period**. This adds a **Period extends Value** element.

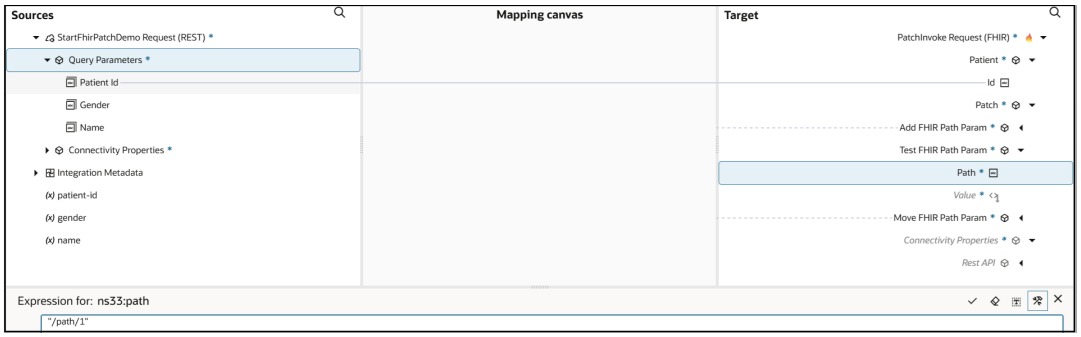


8. Expand **Period extends Value** to set literal string values for child elements such as **Start** and **End**.



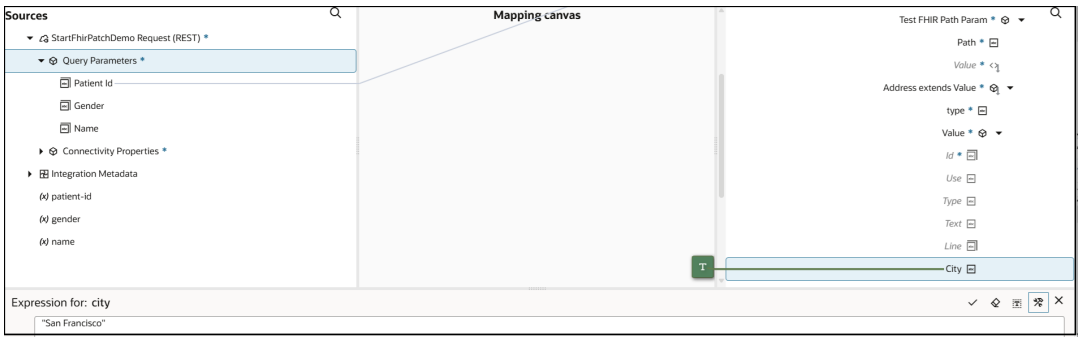
9. Expand **Move FHIR Path Param**.
   The **From Path** and **To Path** nodes are displayed. Unlike **Add FHIR Path Param**, **Move FHIR Path Param** does not appear under a **Path**. **Move FHIR Path Param** also does not require the selection of a data type.
10. Right-click **From Path** and **To Path** and select **Create target node** to set paths as literal values for each.

11. Expand **Test FHIR Path Param**.
12. Select **Path**, then right-click and select **Create target node** to explicitly specify the path in the Expression Builder.



You must now select the corresponding data type of that element.

13. Right-click **Value** below **Path** and select **Extended Data Types**, then **Address**. This adds the **Address extends Value** element.
14. Expand **Address extends Value** to set literal string values for child elements such as **City**.
15. Right-click **City** and select **Create target node** to set a literal value in the Expression Builder.



16. Set literal values for additional elements, as needed.