Oracle® Cloud

Using the GraphQL Adapter with Oracle Integration 3

F86784-11
January 2026

Oracle Cloud Using the GraphQL Adapter with Oracle Integration 3,

F86784-11

Primary Author: Oracle Corporation

Contents

About This Content

1 Understand the GraphQL Adapter

GraphQL Adapter Capabilities 1
GraphQL Adapter Restrictions 2
What Application Version Is Supported? 2
Workflow to Create and Add a GraphQL Adapter Connection to an Integration 2

2 Create a GraphQL Adapter Connection

Prerequisites for Creating a Connection 1
Create a Connection 1
- Configure Connection Properties 3
- Configure Connection Security 3
- Configure the Endpoint Access Type 5
- Test the Connection 6

Upload a Certificate to Connect with External Services 6

3 Add the GraphQL Adapter Connection to an Integration

Basic Info Page 1
Operation Page 2
- Try Query Page 3

Summary Page 4

4 Implement Common Patterns Using the GraphQL Adapter

Insert Zuora Order Records into a Hasura Database Using the GraphQL Adapter 1

About This Content

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

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 http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

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 GraphQL Adapter

Review the following topics to learn about the GraphQL 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:

- GraphQL Adapter Capabilities
- GraphQL Adapter Restrictions
- What Application Version Is Supported?
- Workflow to Create and Add a GraphQL Adapter Connection to an Integration

GraphQL Adapter Capabilities

The GraphQL Adapter enables you to integrate a cloud application that supports GraphQL with Oracle Integration. In addition, the GraphQL Adapter enables you to integrate on-premises applications, which support GraphQL APIs, with Oracle Integration using the connectivity agent. You can configure the GraphQL Adapter as an invoke connection in an integration in Oracle Integration.

The GraphQL Adapter provides the following capabilities:

- Consumes any external GraphQL API from introspection-enabled or disabled applications.
- Allows for fetching an app schema, validating a GraphQL query, and generating a response mapper for introspection-enabled applications.
- Supports query, mutation, multiquery, and multimutation.
- Supports fragments.
- Provides an option for testing a query while configuring.
- Enables you to add custom request and response headers.
- Supports GraphQL APIs protected using API Key-Based Authentication, a header-based security policy, OAuth Authorization Code Credentials, OAuth Authorization Code (Recommended), OAuth Client Credentials, No Security Policy, and Basic Authentication.
- Enables you to implement secure egress (dedicated NAT Gateway) for invoking GraphQL APIs using a private endpoint.
- Supports the connectivity agent.

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

Watch a video to learn more:

Video

GraphQL Adapter Restrictions

Note the following GraphQL Adapter restrictions.

- The input variables are not currently validated.
- In case the GraphQL API comes from an introspection-disabled application server, the GraphQL Adapter does not generate response JSON and validates the GraphQL query. However, if the API comes from an introspection-enabled application server, the GraphQL Adapter validates the GraphQL query and shows errors sequentially, one at a time. After resolving an error, it moves on to show the next error in sequence.
- Extensions in the response are not supported.
- Directives are not supported.

> **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 GraphQL Adapter Connection to an Integration

You follow a very simple workflow to create a connection with an 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 GraphQL 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 GraphQL 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*

Step	Description	More Information
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

Create a GraphQL 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
- Create a Connection
- Upload a Certificate to Connect with External Services

Prerequisites for Creating a Connection

You must satisfy the following prerequisites to create a connection with the GraphQL Adapter:

- Know the GraphQL endpoint URL.
- Understand the security policies and their specific requirements. See Configure Connection Security.
- Understand OAuth security policies.

  If you are using one of the OAuth security policies, you must have already registered your client application to complete the necessary fields on the Connections page.

  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 the time of registration, the client application is assigned a client ID and 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
  ```

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.

Element	Description
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 **GraphQL Endpoint** field, enter a GraphQL endpoint URL. See Prerequisites for Creating a Connection.
3. In the **Header** field under **Optional properties**, enter the header key and header value in the following format, as available in the web service.

   ```
   HeaderKey:HeaderVal
   ```

   ,

Configure Connection Security

Configure security for your GraphQL Adapter connection.

1. Go to the **Security** section.
2. Select the security policy. See Prerequisites for Creating a Connection.
3. If you select **API Key-Based Authentication**:
   a. In the **API Key** field, enter the generated API key used to identify the client.
   b. Optionally, in the **API Key Usage** field, enter the API key again.
4. If you select **Custom Security Policy**:

a. In the **Header Key** field, enter the header key.

b. In the **Header Value** field, enter the header value.

5. If you select **OAuth Authorization Code Credentials**:

   a. In the **Client ID** field, enter the client identifier issued to the client during the registration process.

   b. In the **Client secret** field, enter the client secret that you obtained.

   c. In the **Authorization code URI** field, enter the URI to use for the access token.

   d. In the **Auth Token URI** field, enter the URI to use for the access token.

   e. In the **Scope** field, enter the scope of the access request.

      Scopes enable you to specify which type of access you need. Scopes are used to limit access for the OAuth token. They do not grant any additional permission beyond that which the user already possesses.

   f. 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.

> **Note**
>
> This policy is only visible for connections created prior to the 26.01 release. For new connections, you are advised to use the OAuth Authorization Code (Recommended) security policy explained below.

6. If you select **OAuth Authorization Code (Recommended)**:

   a. Enter the client ID, client secret, authorization code URI, authorization token URI, and scope in the same way as described for the OAuth Authorization Code Credentials security policy.

   b. Optionally add an additional layer of security to the authorization flow by enabling PKCE (Proof Key for Code Exchange) if the target application supports it. When enabled, Oracle Integration generates and includes the code challenge and code verifier in the authorization and token requests, respectively. The code challenge method used is always S256. PKCE adds an additional layer of security to the authorization flow for servers that support it.

      For information about PKCE, see What is PKCE?.

7. If you select **OAuth Client Credentials**:

   a. In the **Client ID** field, enter the client identifier issued to the client during the registration process.

   b. In the **Client secret** field, enter the client secret that you obtained.

   c. In the **Access Token URI** field, enter the URI to use for the access token.

8. If you select **Basic Authentication**:

   a. a. In the **Username** field, enter the name of a user with access to the destination web service.

   b. In the **Password** field, enter the password.

9. If you select **No Security Policy**, no fields are displayed.

   Select **No Security Policy** to create a connection without configuring any 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
- Ensure Private Endpoint Configuration is Successful

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.
Private endpoint	Connections to endpoints using a private virtual cloud network (VCN). **Note**: To connect to private endpoints, you must complete prerequisite tasks in the Oracle Cloud Console. Failure to do so results in errors when testing the connection. See Connect to Private Resources in *Provisioning and Administering Oracle Integration 3* and Troubleshoot Private Endpoints in *Using Integrations in Oracle Integration 3*.
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*.

Ensure Private Endpoint Configuration is Successful

- To connect to private endpoints, you must complete prerequisite tasks in the Oracle Cloud Console. Failure to do so results in errors when testing the connection. See Connect to Private Resources in *Provisioning and Administering Oracle Integration 3*.
- When configuring an adapter on the Connections page to connect to endpoints using a private network, specify the fully-qualified domain name (FQDN) and *not* the IP address. If you enter an IP address, validation fails when you click **Test**.

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.

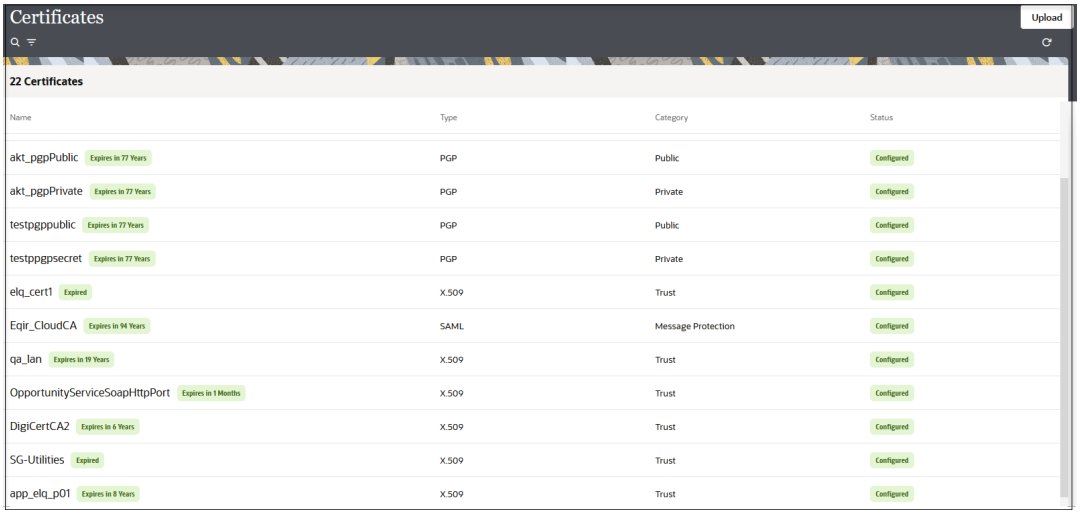
2. Wait for a message about the results of the connection test.
   - If the test was successful, then the connection is configured properly.
   - If the test failed, then edit the configuration details you entered. Check for typos and verify URLs and credentials. Continue to test until the connection 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 certificate expiration date, status, and type. 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**.

3

Add the GraphQL Adapter Connection to an Integration

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

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

Topics:

- Basic Info Page
- Operation Page
- Summary Page

Basic Info Page

You can enter a name and description on the Basic Info page of each adapter in your integration.

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 `righ(t)now4`) except underscores and hyphens • No multibyte characters
What does this endpoint do?	Enter an optional description of the connection's responsibilities. For example: `This connection receives an inbound request to synchronize account information with the cloud application.`
Warning Message (Is displayed only if introspection is disabled.)	Introspection is currently disabled on your GraphQL application server. This prevents the GraphQL Adapter from retrieving the schema that helps to validate the query and construct the response schema. To use this functionality, enable your GraphQL server settings to allow for introspection, then refresh the metadata for this connection. If introspection remains disabled, you may proceed without it by adding a sample JSON payload manually.

Operation Page

Enter the details on the Operation page.

Element	Description

Enter your GraphQL query	Enter the custom GraphQL query.
	Inline fragment example:
	
	`query ($customerID:ID!){`
	`  node(id: $customerID) {`
	`  ... on Customer {`
	`  id`
	`  firstName`
	`  lastName`
	`  email`
	`  }`
	`  }`
	`}`
	
	Named fragment example:
	
	`query`
	`($customerId1:ID!, $customerId2:ID!, $first:Int!){`
	`  OperationOne: customer(id:$customerId1){`
	`  ...CustomerFragment`
	`  }`
	`  OperationTwo: customer(id:$customerId2){`
	`  ...CustomerFragment`
	`  }`
	`}`
	`fragment CustomerFragment on Customer {`
	`  id`
	`  firstName`
	`  lastName`
	`  email`
	`  addresses{`
	`  city`
	`  }`
	`}`

GraphQL Variables	Define the variables for the GraphQL query. **Note** Inline variables are not supported.

Element	Description
GraphQL Response JSON (Appears only if the GraphQL API comes from an introspection-disabled application server.)	Enter the custom GraphQL JSON response. For example: `{ Data: "" }`
Try Query	Execute the query and fetch sample data for reference.
Add custom request headers	Select this check box to add custom request headers on the Request Headers page.
Add custom response headers	Select this check box to add custom response headers on the Response Headers page.

> **Note**
>
> Click **Continue** or outside the GraphQL query box to validate the GraphQL query.

Try Query Page

If you selected **Try Query** on the Operations page, the Try Query page appears with the following options.

Element	Description
GraphQL Variables	Enter the values for the input variables that you defined in the Operations page.
Test	Click **Try Query** to validate the query. The query is tested with the input variables that were entered. Testing the query enables you to check the response while configuring the adapter. You can change or edit your query based on the response.

> **Note**
>
> Testing a mutation query can result in Create/Update/Delete of the actual record in the target application.

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**.

4

Implement Common Patterns Using the GraphQL Adapter

You can use the GraphQL Adapter to implement the following common pattern.

Topics:

- Insert Zuora Order Records into a Hasura Database Using the GraphQL Adapter

> **Note**
>
> Oracle Integration offers a number of prebuilt integrations, known as *recipes*, that provide you with a head start in building your integrations. You can start with a recipe, and then customize it to fit your needs and requirements. Depending upon the solution provided, a variety of adapters are configured in the prebuilt integrations.
> See the Recipes and Accelerators page on the Oracle Help Center.

Insert Zuora Order Records into a Hasura Database Using the GraphQL Adapter

The GraphQL Adapter enables you to seamlessly transfer Zuora order records into the Hasura Database immediately upon their creation in Zuora. Similarly, you can insert other records from an application into the Hasura Database using the GraphQL Adapter.

This use case provides an overview of importing the Zuora order records into Oracle Integration and subsequently inserting those records into the Hasura Database using the GraphQL Adapter.

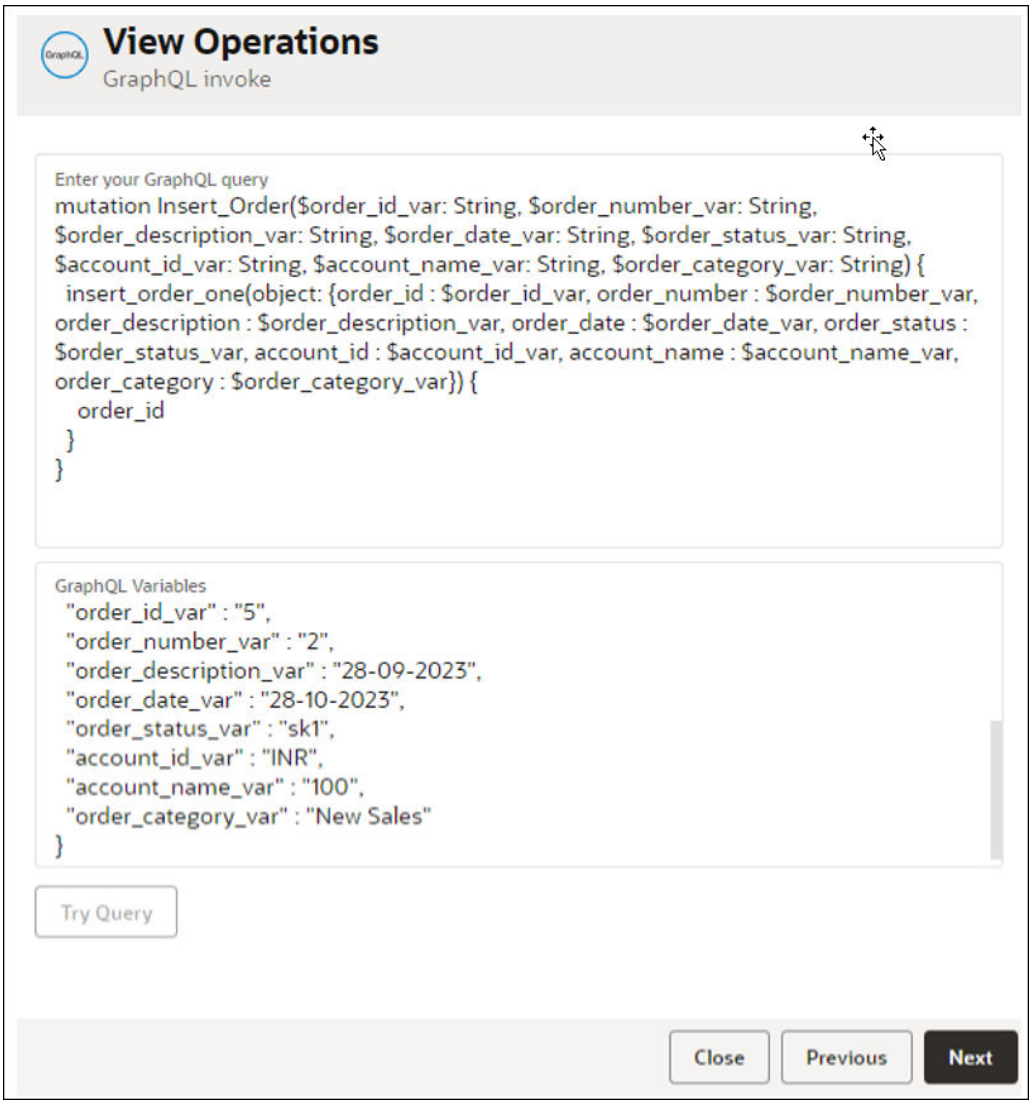
The following adapters and their operations are used in this use case:

- **Order processed event** (Zuora Adapter): Enables you to receive a notification for the configured event (that is, Order Processed Event) in Oracle Integration.
- **Mutation operation** (GraphQL Adapter): Inserts order records into the Hasura Database.

This implementation pattern provides an overview of the steps.

1. Create Zuora Adapter and GraphQL Adapter connections.
2. Create an application integration.
3. Drag a Zuora Adapter into the integration as a trigger connection.
4. Configure the Zuora endpoint as follows:
    a. On the Basic Info page, provide an endpoint name.
    b. Select **Orders** as the business object, and then select **Order Processed Event** as the trigger event name.
    c. Review your selections on the Summary page.

5. Drag a GraphQL Adapter into the integration canvas as an invoke connection.
6. Configure the GraphQL Hasura endpoint as follows:
   a. On the Basic Info page, provide an endpoint name.
   b. On the Operations page, enter the GraphQL mutation query and GraphQL variables of the Hasura Database application.



   c. Review your selections on the Summary page.
7. In the mapper, perform the required mappings to insert the records into the Hasura Database through the GraphQL Adapter.
8. When complete, activate the integration.

   As a result, the integration is invoked when a new order record is created in Zuora and the same record details are imported/inserted into the Hasura Database through the GraphQL Adapter.