Oracle® Cloud

Using the Shopify GraphQL Adapter with Oracle Integration 3

G13455-08
January 2026

Oracle Cloud Using the Shopify GraphQL Adapter with Oracle Integration 3,

G13455-08

Primary Author: Oracle Corporation

Contents

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 Shopify GraphQL Adapter

Review the following topics to learn about the Shopify 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:

Shopify GraphQL Adapter Capabilities

The Shopify GraphQL Adapter enables you to create an integration in Oracle Integration that connects to the Shopify application using GraphQL APIs. You can configure the Shopify GraphQL Adapter as an invoke or trigger connection in an integration in Oracle Integration.

The Shopify GraphQL Adapter provides the following capabilities:

- For trigger endpoints:
  - Receives notifications whenever business events occur in the following modules:
    - Customers (Company Create/Company Delete/Company Update/Customer Creation/Customer Delete/Customer Disable/Customer Enable/Customer Update)
    - FulfillmentEvents (FulfillmentEvent Create/FulfillmentEvent Delete)
    - Fulfillments (Fulfillment Create/Fulfillment Update)
    - Inventory (Inventory Item Create/Inventory Item Update/Inventory Item Delete/ Location Create/Location Update/Location Delete, Inventory Level Connect/ Inventory Level Disconnect/Inventory Level Update)
    - Orders (Order Creation/Order Cancellation/Order Edit/Order Deletion/Order Fulfillment/Order Partial Fulfillment/Order Payment/Order Update/Refund Create/ Transactions Create)
    - Order Returns (Return Approve/Return Cancel/Return Close/Return Decline/ Return Reopen/Return Request/Return Update)
    - Products (Product Creation/Product Update/Product Deletion)

> **Note**
>
> Business events of the Order Returns module and Company events of the Customers module are only supported by API versions 2024-07 and 2024-10. To learn about all the API versions supported by the adapter, see the Certification Matrix.

- For invoke endpoints:

- All the modules supported in the outbound direction are metadata-driven.
- Allows the following operations:
  - fetch (Query) data from Shopify
  - Mutate (Create/Update/Delete) data in Shopify
- Generates a query based on modules and operations selected and automatically validates them.

> **Note**
>
> The depth of this default-generated query is up to six levels.

- Allows you to customize (edit) the query by adding or deleting response objects to the selection. It also allows you to provide or modify input variables, as well as test and update the query as required.

> **Note**
>
> You can update the query and add objects up to a query depth of 17.

- Permits you to restore the original query by using the Reset Configuration option.
- Allows you to try the query (only in case of a data-retrieval operation) and see the sample response.

- Supports multidomain functionality in both the invoke and trigger roles.
- Supports the Shopify Access Token Policy and Shopify Security Policy for establishing a connection with the application.
- Supports implementing secure egress (dedicated NAT gateway) to establish a connection by using a private endpoint. See Connect to Private Resources in *Provisioning and Administering Oracle Integration 3* and Configure the Endpoint Access Type.

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

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 Shopify 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 Shopify 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 Shopify 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*
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 Shopify 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

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

- Create a Custom Application
- Obtain the Admin API Access Token and API Secret Key
- Obtain the Shopify API Credentials and Shared Secret

Create a Custom Application

To create a custom application in Shopify, you use the Shopify administration console. Follow the instructions mentioned under these topics:

- Create the app
- Select API scopes
- Install the app and get the API access tokens

See Create and install a custom app.

Obtain the Admin API Access Token and API Secret Key

You must obtain the values for your Admin API access token and API secret key.

See Get the API credentials for a custom app.

Obtain the Shopify API Credentials and Shared Secret

When you save your private application, Shopify automatically generates an API key, a password, and a shared secret.

To get the values for your API credentials and shared secret:

1. Open the private application that you created and saved.
2. Scroll to the **Admin API** section.

3. Copy the values for your API key, password, and shared secret. You'll need to enter those values on the Connections page when you configure security for your Shopify GraphQL Adapter connection in Oracle Integration. See Configure Connection Security.

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`).

Element	Description
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 **Hostname** field, enter the Shopify host name. The Shopify host name appears in the Shopify home URL. For example:

   `https://hostname.myshopify.com`

3. In the **Shopify GraphQL API Version** field, enter the GraphQL API version. For example:

   `2024-04`

4. In the **Optional properties** field, enter all the connection IDs that you want to associate with this integration.

> **Note**
>
> - Only specify optional properties if you have configured your connection with the **Trigger** and **Trigger and Invoke** roles for inbound operations.
> - Ensure that the provided connection IDs are configured with API versions that support the required events or the Shopify GraphQL Adapter is certified for the specified API version.
>
> These actions help you obtain multidomain functionality in the trigger role.

Configure Connection Security

Configure security for your Shopify GraphQL Adapter connection.

1. Go to the **Security** section.
2. From the **Security Policy** list, select the security policy:
   - **Shopify Access Token Policy**
   - **Shopify Security Policy**
3. If you select **Shopify Access Token Policy**:
   a. In the **Admin API Access token** field, enter the admin API access token that you obtained after performing the prerequisite steps. See Prerequisites for Creating a Connection.
   b. (Optional) In the **API Secret Key** field, enter the API secret key that you obtained after performing the prerequisite steps. See Prerequisites for Creating a Connection.
4. If you select **Shopify Security Policy**:
   a. In the **Username** field, enter the API key that you obtained after performing the prerequisite steps. See Prerequisites for Creating a Connection.
   b. In the **Password** field, enter the password that you obtained after performing the prerequisite steps. See Prerequisites for Creating a Connection.
   c. (Optional) In the **Shared secret** field, enter the shared secret that you obtained after performing the prerequisite steps. See Prerequisites for Creating a Connection.

> **Note**
>
> - The **Shopify Security Policy** can be used for invoke and trigger connections.
> - The **Shared secret** is not marked with an asterisk (*). This incorrectly implies that this field is not mandatory. When you configure this security policy for a trigger connection or trigger and invoke connection, then this field is mandatory. Thus, it is optional only in case of an invoke connection.

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*.

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.

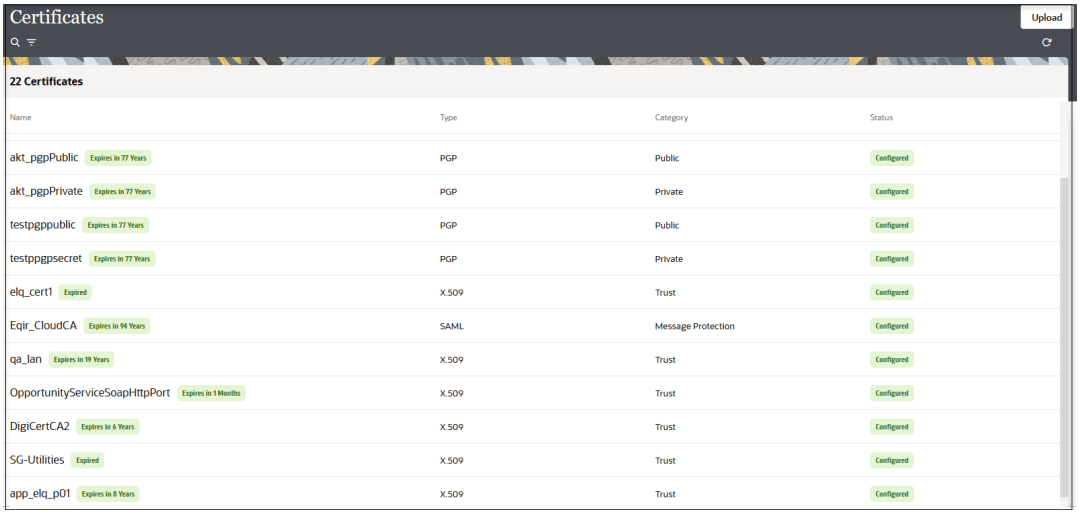
2. Wait for a message about the results of the connection test.
   - If the test was successful, then the connection is configured properly.
   - If the test failed, then edit the configuration details you entered. Check for typos and verify URLs and credentials. Continue to test until the connection 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 Shopify GraphQL Adapter Connection to an Integration

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

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

Topics:

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.`

Trigger Event Page

Select the modules and associated events to receive notifications.

Element	Description
Select Module	Select the module: • **Customers**: Configures customers and customer-related business events such as Customer creation/update/enable/disable/delete. Also configured is Company as a submodule. Company-related business events such as create/update /delete are also supported. • **FulfillmentEvents**: Configures fulfillment events and fulfillment event-related business events such as FulfillmentEvents create/ delete. • **Fulfillments**: Configures fulfillments and fulfillment-related business events such as Fulfillment create/update. • **Inventory**: Configures inventory and inventory-related business events such as Inventory Item create/update and Inventory Level create/update. • **Orders**: Configures orders and order-related business events such as Order creation/edit/update/deletion/cancellation, Order payment, Order fulfillment/partial fulfillment, Transactions create, and Refund create. • **Order Returns**: Configures order returns and order return-related business events such as Return Approve/Cancel/Close/Decline/ Reopen/Update/Request. • **Products**: Configures products and product-related business events such as Product creation/update/deletion.
Select Event	Select an event name, such as Customer creation. The events available for selection are based on the module selected.
Filter by Module Name	Type the initial letters of the module name to filter the display of names in the list.
Filter by Event Name	Type the initial letters of the event name to filter the display of names in the list.

Invoke Dynamic Connections Page

Enter the following details to enable dynamic connections in your integration.

Element	Description
Enable Dynamic Connection	Click the check box if you want to enable dynamic connections in your integration.
Dynamic Connection ID	Click  to switch to designer view. An **Input sources** tab appears. Expand **execute***, then **QueryParameters***, and then drag **Dynamic-Connection-Id** to the text box. When you activate and run the integration, you can pass the connection ID of different stores as the query parameter. For example, if you want to create a customer in store A and Store B, you can use a single integration. Pass the connection IDs of both stores one by one to create customers in both the stores.

Invoke Actions Page

Select the action to perform.

Element	Description
Select the action you want to perform	Select the action you want to perform. • **Query**: Retrieves information from the Shopify application corresponding to the selected object and operation. • **Create, Update and Delete**: Creates, updates, or deletes a record in the Shopify application.

Invoke Operations Page

Select the module, the operations you want to perform in the Shopify application, and the response parameters .

Element	Description
Select Module	Select the module on which you want to perform operations (for example, Customers, Company, Fulfillment, Order, and Payment).
Select Business Object	Select an operation for a module such as Customer Create, Customer Merge, Customer Delete, Customer Merge Job Status, and so on. The operations available for selection are based on the module and action (Query/Create,Update or Delete) selected.
Business Object Description	Describes the business object or the operation selected above. For example, a description for the Customer Payment Method business object might be: `Returns a CustomerPaymentMethod resource by its Id.`

Element	Description
Available Response Objects	Displays a list of objects in a shuttle box. You can select these objects to filter the response of query, create, update, and delete operations. You can select a maximum of five response objects, which allows you to filter results of up to three six-nested levels. The shuttle box shows only objects. Scalar and ENUM type fields are included in the query by default. The objects retrieved from the Shopify application are determined by the eligibility criteria listed below: • Eligibility criteria for objects to appear in the shuttle box based on arguments (first, last, before, after, query, or reverse): **Note**: To understand these criteria, perform the following configurations as an example: select **Query** as the invoke action, **Customer** as the module, and **Customers** as the business object. See Shopify API Documentation for Customers Query. An object appears in the shuttle box only if it fulfills one of the conditions listed below: – Contains only the first argument (for example, addresses). – Contains all or some of the arguments (for example, addressesV2). An object does not appear in the shuttle box if it: – Contains arguments other than first, last, after, before, query or reverse (for example, events). – None of the arguments are present (for example, canDelete). – If the object is deprecated (for example, email). • Eligibility criteria for objects to appear in the shuttle box based on the operation's response schema: To understand these criteria, perform the following configurations as an example: select **Query** as the invoke action, **Customer** as the module, and **Customer Merge Preview** as the business object. See Shopify API Documentation for Customer Merge Preview Query. – If the operation’s response schema has exactly three parent fields (edges, nodes, and page info), then nested objects within nodes and page info are displayed in the shuttle box (for example, addressesV2). – If the operation’s response schema has parent fields other than edges, nodes, and page info, then those parent fields appear as objects in the shuttle box (for example, alternateFields).
Edit Query	Allows you to edit the default query generated based on the action, module, business object, and response objects selected so far. **Note**: You can add a new response object or remove the selected one while editing the query. You can also supply the input variable, test, and update the query, if required.
Try Query (Appears only if you select Query action)	Allows you to execute the query and fetch a sample data for reference.
Reset Configurations	Allows you to discard a customized query and return to the default configuration state.

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 Shopify GraphQL Adapter

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

Topics:

Retrieve Product Data from Shopify for Storage in Amazon S3

You can fetch product data from Shopify using the Shopify GraphQL Adapter, write the data to a stage file action, and upload that file to Amazon S3 using the Put Object operation.

This section provides an overview of the steps for implementing this pattern.

1. Create Amazon Simple Storage Service (S3) Adapter and Shopify GraphQL Adapter connections.
2. Create a schedule integration in the integration canvas.
3. Drag the Shopify GraphQL Adapter into the integration canvas.
4. Configure the Shopify GraphQL Adapter as follows:
    a. On the Basic Info page, enter an endpoint name.
    b. On the Operations page, enter the following:
        i. Select the **Query** operation.
        ii. Select **Products** as the module.
        iii. Select **Products** as the object.
        iv. Select the required object fields for the query.
    c. On the Summary page, review and confirm the configuration.
5. Configure the Shopify GraphQL request mapper by specifying the count/number of products to fetch.
6. Drag a stage file action into the integration canvas and configure it as follows:
    a. On the Basic Info page, select **Write File** as the operation.
    b. Drag and drop the file reference to write.
    c. On the Configuration page, enter the following:
        i. Select **CSV** as the file structure.
        ii. Upload a sample CSV file.
        iii. Enter the file record and file record set names.
    d. On the Summary page, review and confirm the configuration.
7. Use the mapper to map fields from the Shopify GraphQL response to the stage file structure.

8. Drag the Amazon Simple Storage Service (S3) Adapter into the canvas and configure it as follows:
   a. On the Basic Info page, provide a name for the connection.
   b. From the **Category** list, select **Object**.
   c. From the **Action**, select **Put Object** to upload the file.
   d. On the Configuration page, specify the bucket name and object name.
   e. On the Summary Page, review and confirm the configuration.
9. In the mapper, pass the stage file output as input to the Amazon Simple Storage Service (S3) Adapter for upload.
10. Click **Validate** to ensure the integration is correctly configured.

11. Add a tracking element in the **Business Identifier** field.
12. Activate the integration.

    The resulting file in Amazon S3 contains all product data retrieved from Shopify.

5

Troubleshoot the Shopify GraphQL Adapter

Review the following topic to learn about troubleshooting issues with the Shopify GraphQL Adapter.

Topics:

Known Issues When Configuring Multidomain Functionality

Note the following known issues when configuring multidomain functionality.

Optional Properties Persist on the Connections Page

Assume you configure a trigger connection with the Shopify GraphQL Adapter to exhibit multidomain functionality and then export the integration to a global project or a new environment. If you then attempt to edit the connection, you may find that optional properties still persist.

The following image shows this behavior. This is a known issue currently being addressed.

Optional properties Section Appears on the Connections Page When Configuring the Connection with the Invoke Role

Assume you configure the Shopify GraphQL Adapter with the invoke role. You may encounter the **Optional properties** section on the Connections page. This is unexpected and is not required with the invoke role. This is a known issue currently being addressed.

Activation Failed

Assume you configure multidomain functionality for the trigger role and enter multiple connection IDs in the **Optional properties** section. Any connections that have the access types set to **Private endpoint** and **Public gateway** fail during integration activation.

This is a known issue currently being addressed.

Refresh an Endpoint Using an Incorrect Connection ID

After calling the refresh endpoint with the **Secondary Store Connection ID**, the request is still sent to the **Primary Store Connection ID** from the initial integration.

Error While Invoking the API

You may encounter the following error when you perform any outbound operation with the Shopify GraphQL Adapter.

Solution: Shopify does not support several fields and throws an internal server error whenever they are sent as a part of API request. The error-causing fields are:

- **onboardingVisual**
- **usingShopifyBalance**
- **merchantApprovalSignals object**

A simple solution is to edit the query and remove the above fields from the request body. After updating, save and activate the integration for successful results.