

Shopify Adapter Guide

For SAP Integration Suite and SAP Cloud Integration

Version 1.3.1 - March 2026

Table of Contents

1. Introduction	4
1.1. Coding Samples	4
1.2. Internet Hyperlinks	4
2. Shopify Integration	4
2.1. Introduction	4
2.2. Shopify Adapter	5
2.3. Architectural Overview	6
3. Supported Operations	7
3.1. REST Admin API	8
3.2. GraphQL Admin API	19
4. Authentication & Authorization	20
4.1 Creating Secure Parameter in Security Material	20
4.2 Usage of the Secure Parameter	23
4.3 Creating OAuth2 Client Credentials in Security Material	23
4.4 Usage of the OAuth2 Client Credentials	27
5. Shopify Configuration	27
5.1 Shopify legacy app creation	27
5.2 Shopify custom app creation	33
6. Receiver Adapter Configuration	40
6.1. Connection	40
6.2. Processing Configuration for REST Admin API	42
6.3. Processing for GraphQL	62
7. Payload and Dynamic Configuration	62
7.1. Message Payloads	62

Shopify Adapter Guide

7.2. Dynamic configuration	64
8. Troubleshooting	69
8.1. Unsuccessful Request	69
8.2. Base URL Exception	70
8.3. Missing Message Body	70
9. Sample Scenarios Explained	70
9.1. Simple scenario using dynamic configuration and write variables for retrieving list of variables	71
10. References	85
10.1. Locate the Shopify Base URL Address	85
11. API Deprecation Notice	87

1. Introduction

This is the official guide for the Shopify Adapter for SAP Cloud Integration. This guide covers all relevant information for integration developers to start working with the Shopify Adapter. Read this guide carefully before using the Shopify Adapter.

1.1. Coding Samples

Any software coding and/or code lines/strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules of certain coding. The correctness and completeness of the Code given herein are not assured.

1.2. Internet Hyperlinks

The documentation may contain hyperlinks to the Internet. These hyperlinks are intended to serve as hints on where to find related information. The availability and the correctness of these links to serve a particular purpose are not warranted.

2. Shopify Integration

2.1. Introduction

Shopify is a leading global commerce company that provides essential internet infrastructure for commerce, offering trusted tools to start, scale, market, and run a retail business of any size.

Shopify makes commerce better for everyone with a platform and services that are engineered for speed, customization, reliability, and security, while delivering a better shopping experience for consumers online, in store and everywhere in between. Shopify powers millions of businesses in more than 175 countries and is trusted by brands such as Mattel, Gymshark, Heinz, FTD, Netflix, Kylie Cosmetics, SKIMS, Supreme, and many more.

For more information about Shopify, visit www.shopify.com

2.2. Shopify Adapter

The Shopify Adapter for SAP Integration Suite accelerates the implementation time and reduces the complexity of connecting to Shopify.

2.2.1. Features

From a high-level perspective, the following key features are provided by the Shopify Adapter

- Support for multiple Shopify REST Admin API operations, including:
 - Balance
 - Balance Transactions
 - Collect
 - Collection
 - Custom Collection
 - Customer
 - Discount Code
 - Dispute
 - Dispute Evidence
 - Dispute File Upload
 - Event
 - Inventory Item
 - Inventory Level
 - Location
 - Metafield
 - Order
 - Order Risk
 - Payouts
 - Price Rule
 - Product

- Product Variants
- Refund
- Smart Collection
- Tender Transaction
- Transaction
- Support for GraphQL Admin API queries and mutations
- Shopify releases a new API version every 3 months at the beginning of the quarter. Each stable version is supported for a minimum of 12 months. Version names are date-based to be meaningful and semantically unambiguous (for example, 2026-01). The Shopify Adapter supports the following Shopify API versions:
 - 2025-04
 - 2025-07
 - 2025-10
 - 2026-01
 - For more information about Shopify API versioning, refer to <https://shopify.dev/docs/api/usage/versioning>
- Secure authentication for Shopify's Admin APIs
- Polling mechanism for Shopify Admin API Events
- Option to automatically format the Shopify Admin API response in XML and JSON formats.
- Option to add log attachments on failure

2.3. Architectural Overview

From a technical perspective, the Shopify Adapter can be used as a Receiver Adapter. As a Receiver Adapter, SAP Integration Suite acts as the initiator of the calls. In case the calls towards Shopify need to be scheduled, a Scheduler can be utilized within Integration Flows.

Figure 2.1. showcases how the Shopify Adapter can be used in a simple Request-Reply scenario. Various operations can be used in the Adapter, explained in [Section 3](#).

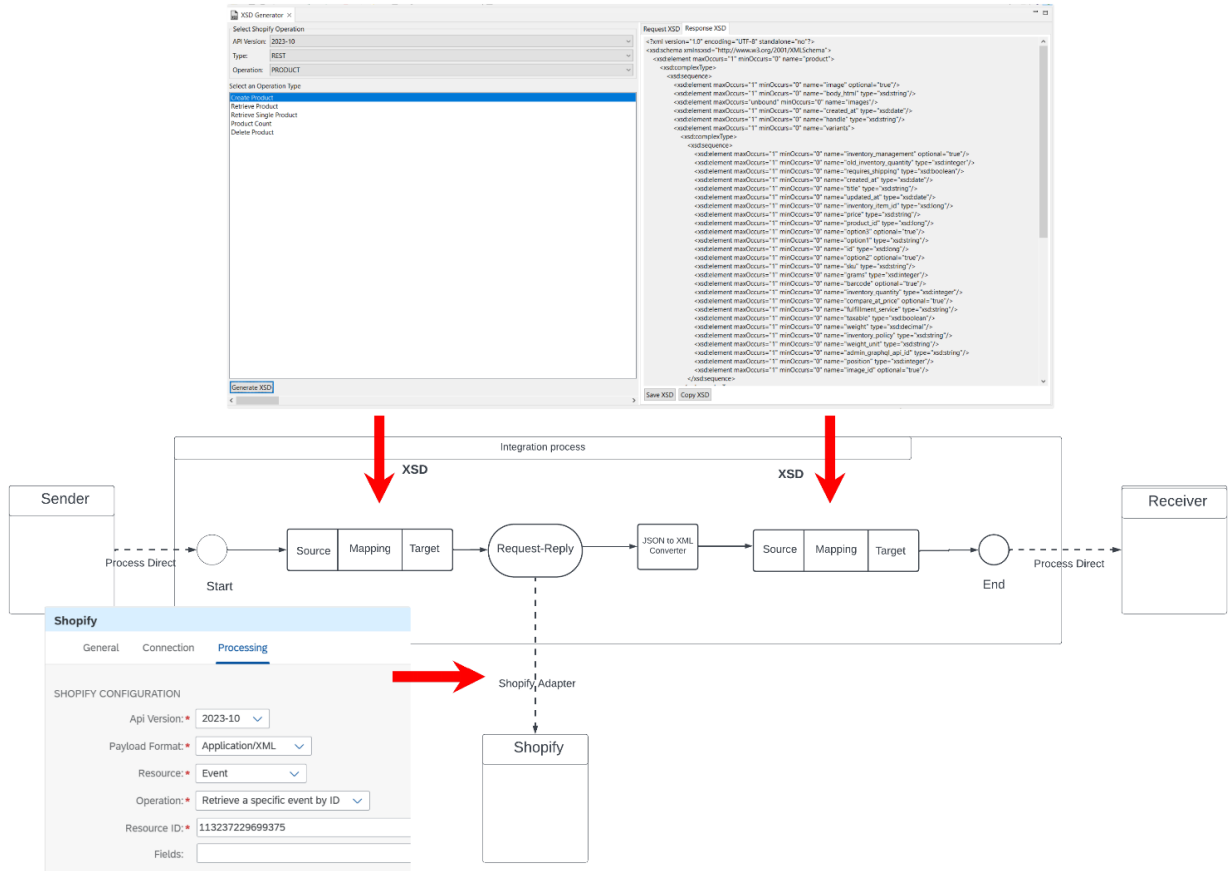


Figure 2.1 Usage of the Shopify Adapter for SAP Integration Suite

Request-Reply steps can use the Shopify Adapter, as shown in Figure 2.1. The Shopify Adapter is responsible for connecting and interacting with Shopify’s Admin APIs and invoking different operations.

3. Supported Operations

The Shopify Adapter supports the following Shopify Admin APIs:

- Shopify REST Admin API
- Shopify GraphQL Admin API

Brief descriptions for both are presented in the following sections.

3.1. REST Admin API

The adapter supports Shopify's REST Admin API and allows integration with Shopify using JSON format. The integration is done securely by utilizing the Shopify's API access tokens for authenticating REST calls. The Admin API lets you build apps and integrations that extend and enhance the Shopify admin.

The following resources are currently supported in Adapter's REST Admin API:

Shopify Adapter Guide

Resource	Description
Orders	An order is a customer's request to purchase one or more products from a shop. You can create, retrieve, update, and delete orders using the Order resource
Product	The Product resource lets you update and create products in a merchant's store
Product Variants	A variant can be added to a Product resource to represent one version of a product with several option
Inventory items	An inventory item represents a physical good
Locations	A location represents a geographical location where your stores, pop-up stores, headquarters, and warehouses exist. You can use the Location resource to track sales, manage inventory, and configure the tax rates to apply at checkout.
Inventory Levels	An inventory level represents the quantities of an inventory item for a location
Customer	The Customer resource stores information about a shop's customers, such as their contact details, their order history, and whether they've agreed to receive email marketing.
Metafield	Metafields are a flexible way to attach additional information to a Shopify resource
Event	Events are generated by some Shopify resources when certain actions are completed, such as the creation of an article, the fulfillment of an order, or the addition of a product.
Transaction	Transactions are created for every order that results in an exchange of money.
Refund	The Refund resource has two major components: <ul style="list-style-type: none"> • Transaction records of money returned to the customer • The line items included in the refund, along with restocking instructions
Order Risk	The Order Risk resource allows you to create, retrieve, update, and delete order risks. Order risks represent the results of fraud checks that have been completed for an order

Resource	Description
Collection	A collection is a grouping of products that merchants can create to make their stores easier to browse
Smart Collection	A smart collection is a grouping of products defined by rules that are set by the merchant
Custom Collection	A collection which contains products that are manually added to a collection by a merchant
Collect	The Collect resource connects a product to a custom collection
Discount Code	The PriceRule DiscountCode resource can be used to create discount codes that enable specific discounts to be redeemed. Merchants can distribute discount codes to their customers using a variety of means, and customers can apply these codes at checkout. Each discount code belongs to a price rule, which holds the logic for the discount.
Price Rule	The PriceRule resource can be used to create discounts using conditions
Tender Transaction	Each tender transaction represents money passing between the merchant and a customer. A tender transaction with a positive amount represents a transaction where the customer paid money to the merchant
Balance	A balance shows the current available funds in a merchant's Shopify Balance account, reflecting the total money they can use. The most important values returned in the response include the available balance, pending balance, currency, and last updated timestamp, providing a comprehensive snapshot of the merchant's financial status. It updates in real-time, helping merchants easily track their financial standing and plan their business expenses.
Payouts	Payout represents the transfer of funds from a merchant's Shopify Balance account to their external bank account. The most important values returned in the response include the payout amount, currency, status, and the date of the payout, which helps merchants track the movement of funds. Payouts are automatically processed and provide a clear record of when and how much money has been transferred.
Balance Transactions	The Balance Transactions API endpoint retrieves a list of financial activities related to a merchant's Shopify Payments account, specifically those linked to a particular payout.

Resource	Description
Dispute	The Payments Dispute API endpoint retrieves information about disputes related to transactions processed through Shopify Payments. It provides detailed data on each dispute, including its status, reason, and the amount under dispute.
Dispute Evidence	Dispute evidence is submitted by the merchant to support the legitimacy of a dispute
Dispute File Upload	Upload and manage dispute evidence files for a dispute

For each resource, there are multiple possible operations, which are listed below:

Resource	Operation	Description
Orders	Cancel an order	Cancels an order by ID. Orders that are paid and have fulfillments can't be canceled.
	Close an order	Closes an order by ID. A closed order is one that has no more work to be done. All items have been fulfilled or refunded.
	Create an order	Creates an order. A payload must be added in a Content Modifier with JSON content specifying the new order.
	Delete an order	Deletes an order by ID. Orders that interact with an online gateway can't be deleted.
	Re-open a closed order	Re-opens a closed order by ID
	Retrieve a list of orders	Retrieves a list of orders that meet the specified criteria. By default, only orders with status open are retrieved.
	Retrieve a specific order	Retrieve an order by specifying the ID. All fields of an order are returned unless specific fields are named. Returns open orders by default.
	Retrieve an order count	Retrieve the total number of orders that meet the specified criteria.
	Update an order	This operation allows for updating properties of an order including `buyer_accepts_marketing`, `email`, `phone`, `note`, `tags`, `metafields` and `shipping_address_attributes`. It is not for editing the items of an order. Update an order by ID. A payload must be added in a Content Modifier with JSON content specifying the order update

Resource	Operation	Description
Products	Create a product	Create a new product. A payload must be added in a Content Modifier with JSON content specifying the new product
	Delete a product	Deletes a product by ID
	Retrieve a list of products	Retrieve a list of products.
	Retrieve a product count	Retrieve a count of products.
	Retrieve a specific product	Retrieve a single product by ID
	Update a product	Update a product by ID. A payload must be added in a Content Modifier with JSON content specifying the product update
Product Variants	Create a new product variant	Creates a new product variant
	Retrieve a list of product variants	Retrieves a list of product variants.
	Product variant count	Retrieves a count of product variants
	Retrieve a single product variant	Retrieves a single product variant by ID
	Modify an existing product variant	Updates an existing product variant
	Remove an existing product variant	Remove an existing Product Variant
Inventory items	Retrieve a list of inventory items	Retrieves a detailed list for inventory items by IDs
	Retrieve a specific inventory item	Retrieves a single inventory item by ID

Shopify Adapter Guide

Resource	Operation	Description
	Update an inventory item	Updates an existing inventory item
Inventory levels	Adjusts an inventory level	Adjusts the inventory level of an inventory item at a single location
	Connects an inventory item to a location	Connects an inventory item to a location by creating an inventory level at that location.
	Sets an inventory level	Sets the inventory level for an inventory item at a location
	Retrieve a list of inventory levels	Retrieves a list of inventory levels.
	Delete an inventory level	Deletes an inventory level from a location
Location	List of locations	Retrieve a list of locations.
	Retrieve a location	Retrieve a single location by its ID.
	List of inventory levels for a location	Retrieve a list of inventory levels for a location
	Count of locations	Retrieve a count of locations.
Customer	Create a customer	Creates a customer.
	Create an account activation URL for a customer	Generate an account activation URL for a customer whose account is not yet enabled.
	Customer Count	Retrieves a list of customers
	Delete a customer	Deletes a customer
	Retrieve a list of customers	Retrieves a list of customers

Shopify Adapter Guide

Resource	Operation	Description
	Retrieve a specific customer	Retrieves a single customer.
	Retrieve all orders that belong to a customer	Retrieves all orders that belong to a customer.
	Search customers	Retrieves all orders that belong to a customer.
	Send an account invite to a customer	Sends an account invite to a customer.
	Update a customer	Updates a customer
Metafield	Create a metafield	You can create any number of metafields for a resource.
	List of metafields by resource	Retrieves a list of metafields attached to a particular resource.
	List of metafields that belong to a resource by using query parameters	Retrieves a list of metafields that belong to a resource by using query parameters.
	Retrieve a specific metafield by ID	Retrieve a metafield by specifying the ID
	Count of a resource's metafields	Retrieve a metafield by specifying the ID
	Update a metafield	Retrieve a metafield by specifying the ID
	Delete a metafield by its ID	Deletes a metafield by its ID.
Event	List of events	Retrieves a list of events
	Retrieve a specific event by ID	Retrieves a single event by its ID
	Count of events	Retrieves a count of events

Shopify Adapter Guide

Resource	Operation	Description
Transaction	Create a transaction	Creates a transaction for an order
	List of transactions	Retrieves a list of transactions
	Count of order's transactions	Retrieves a count of an order's transactions
	Retrieve a specific transaction by ID	Retrieves a specific transaction.
Order risk	Create an order risk	Creates an order risk for an order
	Retrieve a list of order risks	Retrieves a list of all order risks for an order
	Retrieve a specific order risk	Retrieves a single order risk by its ID
	Update an order risk	Updates an order risk of an order
	Delete an order risk	Deletes an order risk for an order
Refund	Create a new order refund	Creates a refund.
	Calculate a refund for an order	Calculates refund transactions based on line items and shipping
	Retrieve a list of order refunds	Retrieves a list of refunds for an order
	Retrieve a single order refund	Retrieves a specific refund
Collection	Retrieve a specific collection	Retrieves a single collection
	Retrieve a list of products belonging to a collection	Retrieve a list of products belonging to a collection
Custom Collection	Create a custom collection	Creates a custom collection

Shopify Adapter Guide

Resource	Operation	Description
	Retrieve a list of custom collections	Retrieves a list of custom collections
	Retrieve a specific custom collection	Retrieves a single custom collection
	Count custom collections	Retrieves a count of custom collections
	Update a custom collection	Updates an existing custom collection
	Delete a custom collection	Deletes a custom collection
Smart Collection	Create a smart collection	Creates a new smart collection using the specified rules
	Retrieve a list of smart collections	Retrieves a list of smart collections
	Retrieve a specific smart collection	Retrieves a single smart collection
	Count smart collections	Retrieves a count of smart collections
	Update a smart collection	Updates an existing smart collection
	Update the ordering type	Updates the ordering type of products in a smart collection
	Delete a smart collection	Removes a smart collection
Collect	Add product to a custom collection	Adds a product to a custom collection
	Retrieve a list of collects	Retrieves a list of collects
	Retrieve a specific collect	Retrieves a specific collect by its ID.
	Count collects	Retrieves a count of collects.

Resource	Operation	Description
	Remove a product from a collection	Removes a product from a collection.
Discount Code	Create a discount code creation job	Creates a discount code creation job.
	Create a discount code	Creates a discount code.
	Count of discount codes	Retrieves a count of discount codes.
	Retrieve a discount code by code name	Retrieve a discount code by given code name
	Retrieve a discount code creation job	Retrieves a discount code creation job.
	Retrieve a list of discount codes for a discount code creation job	Retrieves a list of discount codes for a discount code creation job.
	Retrieve a list of discount codes	Retrieves a list of discount codes.
	Retrieve a discount code by id	Retrieves a single discount code.
	Update a discount code	Updates an existing discount code.
	Delete a discount code	Removes a discount code.
Price Rule	Create a price rule	Creates a price rule.
	Retrieve a list of price rules	Retrieves a list of price rules.
	Retrieve a specific price rule	Retrieves a single price rule.
	Count of Price Rules	Retrieves a count of all price rules.
	Update a price rule	Updates an existing a price rule.

Resource	Operation	Description
	Delete a price rule	Deletes a price rule.
Tender Transaction	Retrieve a list of tender transactions	Retrieves a list of tender transactions.
Balance	Retrieves the account's current balance.	Retrieves the account's current balance.
Payouts	Retrieves a list of all payouts	Retrieves a list of all payouts ordered by payout date, with the most recent being first.
	Retrieves a single payout	Retrieves a single payout by id.
Balance Transactions	Retrieves a list of all balance transactions	Retrieves a list of all balance transactions ordered by processing time, with the most recent being first.
Dispute	Retrieve a list of all disputes	Retrieve all disputes ordered by initiated_at date and time, with the most recent being first.
	Retrieves a single dispute	Retrieves a single dispute by ID.
Dispute Evidence	Retrieves evidence associated with dispute	Retrieves evidence associated with dispute ID.
	Updates dispute evidence	Updates dispute evidence associated with the dispute ID.
Dispute File Upload	Upload a dispute evidence file	Upload file of a category to a dispute.
	Delete a dispute evidence file	Delete a dispute evidence file given an ID

3.2. GraphQL Admin API

The adapter supports Shopify's GraphQL Admin API and allows integration with Shopify by leveraging GraphQL queries and mutations.

The integration is done securely by utilizing the Shopify API access tokens for Authorization or by authenticating with OAuth2 Client Credentials. By defining queries and mutations, all Shopify GraphQL Admin API resources can be accessed through the Shopify Adapter, as long as the necessary permissions are granted in Shopify.

4. Authentication & Authorization

There are two available methods you can use to authorize your integration with Shopify, each designed to give you secure and dependable access to your store. These options let you choose the approach that best fits your setup while keeping the connection simple and reliable.

1. **OAuth2 Client Credentials with a Custom Shopify App** – This is the recommended approach for connecting your Shopify store after the legacy custom app approach has been phased out. When you set up a Custom App in your Shopify store, you can securely provide the required authorization for the integration. To create a Shopify Custom app follow the steps described in [Section 5.2](#).
2. **Access Token from a Legacy Custom App** – This approach is now outdated and only applies to Legacy Custom Apps that already exist. When the Legacy Custom app is installed in the Shopify Store, it provides the necessary authorization for the integration to work. For more details, see [Section 5.1](#).

The Shopify Adapter leverages SAP Integration Suite's security mechanisms by storing sensitive configuration as Security Materials. These artifacts are then referenced within the Shopify Adapter configuration using aliases.

4.1 Creating Secure Parameter in Security Material

The creation of credentials to support the authentication mechanism is explained in the steps below:

4.1.1 Navigate to Monitor View and then to “Security Material”.

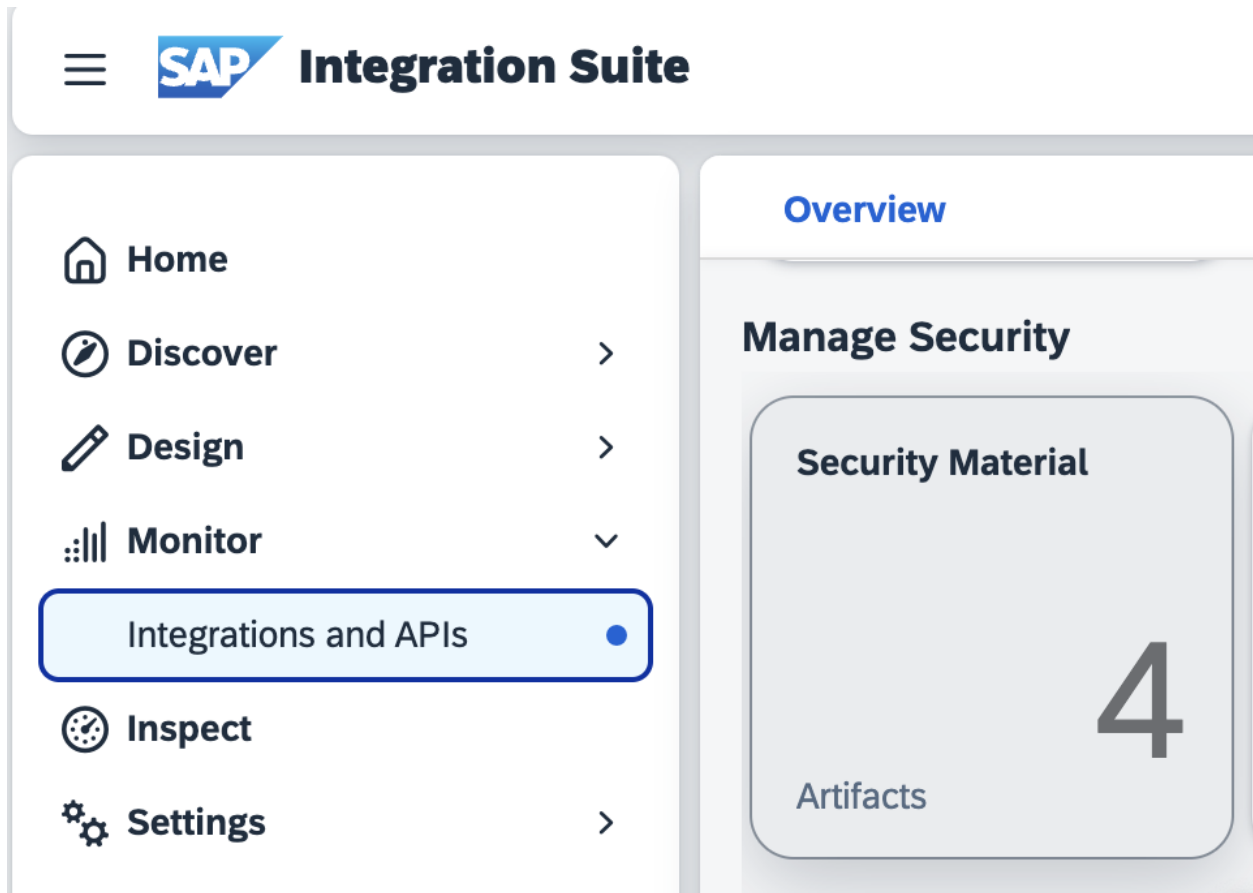


Figure 4.1 Navigating to SAP Integration Suite Monitoring

4.1.2. Create a Secure Parameter for the Shopify Admin API access token

1. Go to Create > Secure Parameter.

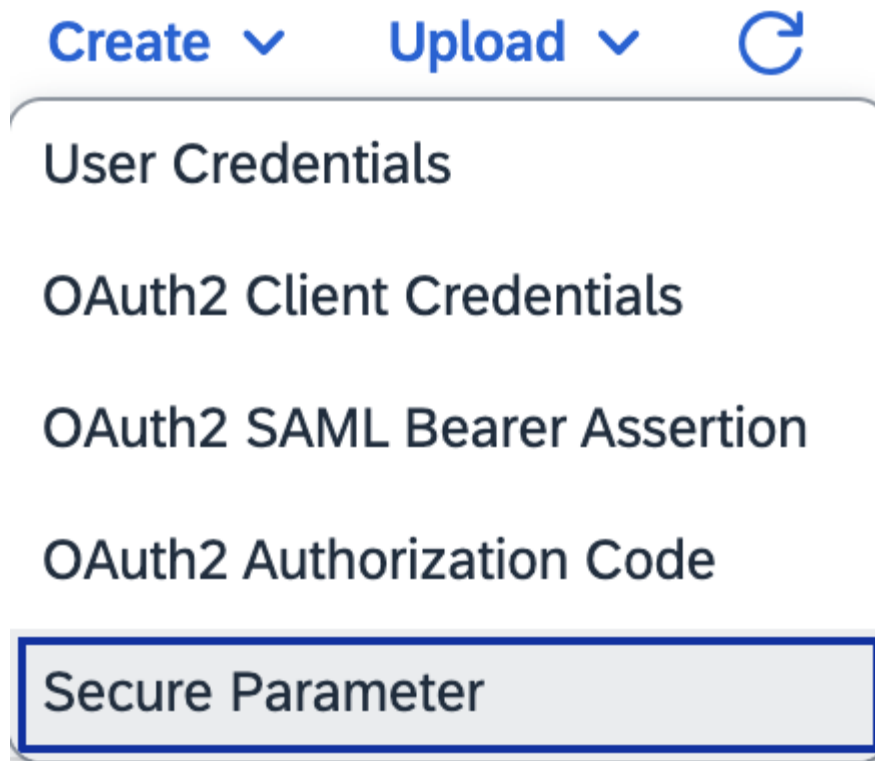


Figure 4.2 Creation of a new Secure Parameter

- 2. Enter the Secure Parameter details

Create Secure Parameter

Name: *

Description:

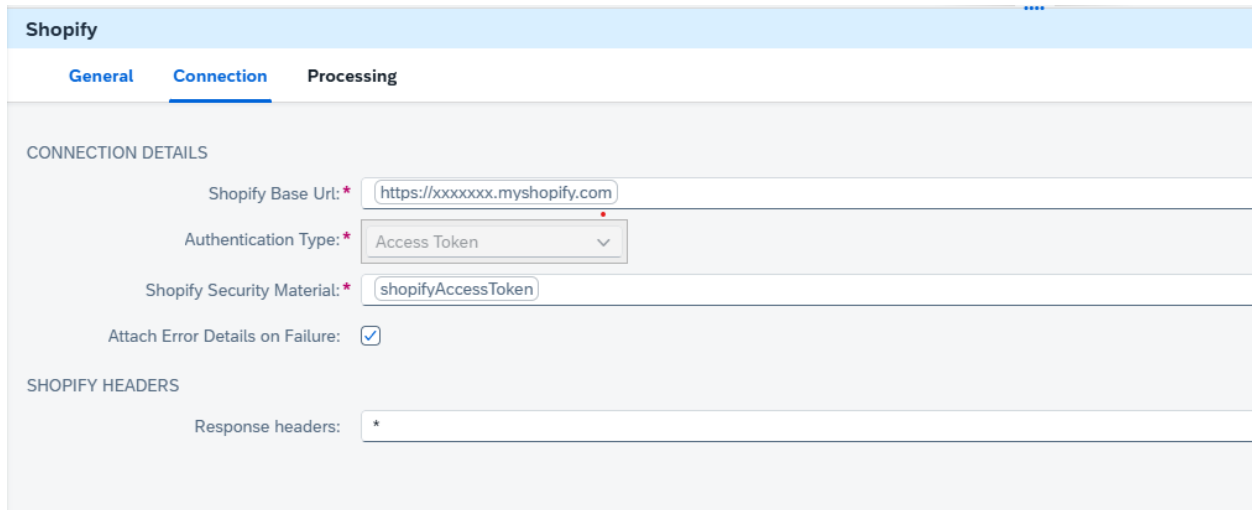
Secure Parameter: *

Repeat Secure Parameter: *

Figure 4.3 Creation of a new Secure Parameter

4.2 Usage of the Secure Parameter

The Security artifact should be used in the Connection tab of the Adapter as shown in Figure 4.4. The option "Access Token" must be selected. Detailed explanation of configuring the connection properties can be found in [Section 6.1](#).



The screenshot displays the configuration interface for the Shopify adapter, specifically the 'Connection' tab. The interface is titled 'Shopify' and has three tabs: 'General', 'Connection', and 'Processing'. The 'Connection' tab is active. Under the heading 'CONNECTION DETAILS', there are four fields: 'Shopify Base Url' with the value 'https://xxxxxx.myshopify.com', 'Authentication Type' set to 'Access Token', 'Shopify Security Material' with the value 'shopifyAccessToken', and 'Attach Error Details on Failure' which is checked. Below this, under 'SHOPIFY HEADERS', there is a 'Response headers' field with an asterisk (*) as its value.

Figure 4.4 Using the Secure Parameter in Adapter Configuration

4.3 Creating OAuth2 Client Credentials in Security Material

The creation of credentials to support the authentication mechanism is explained in the steps below:

4.3.1 Navigate to Monitor View and then to “OAuth2 Client Credentials”.

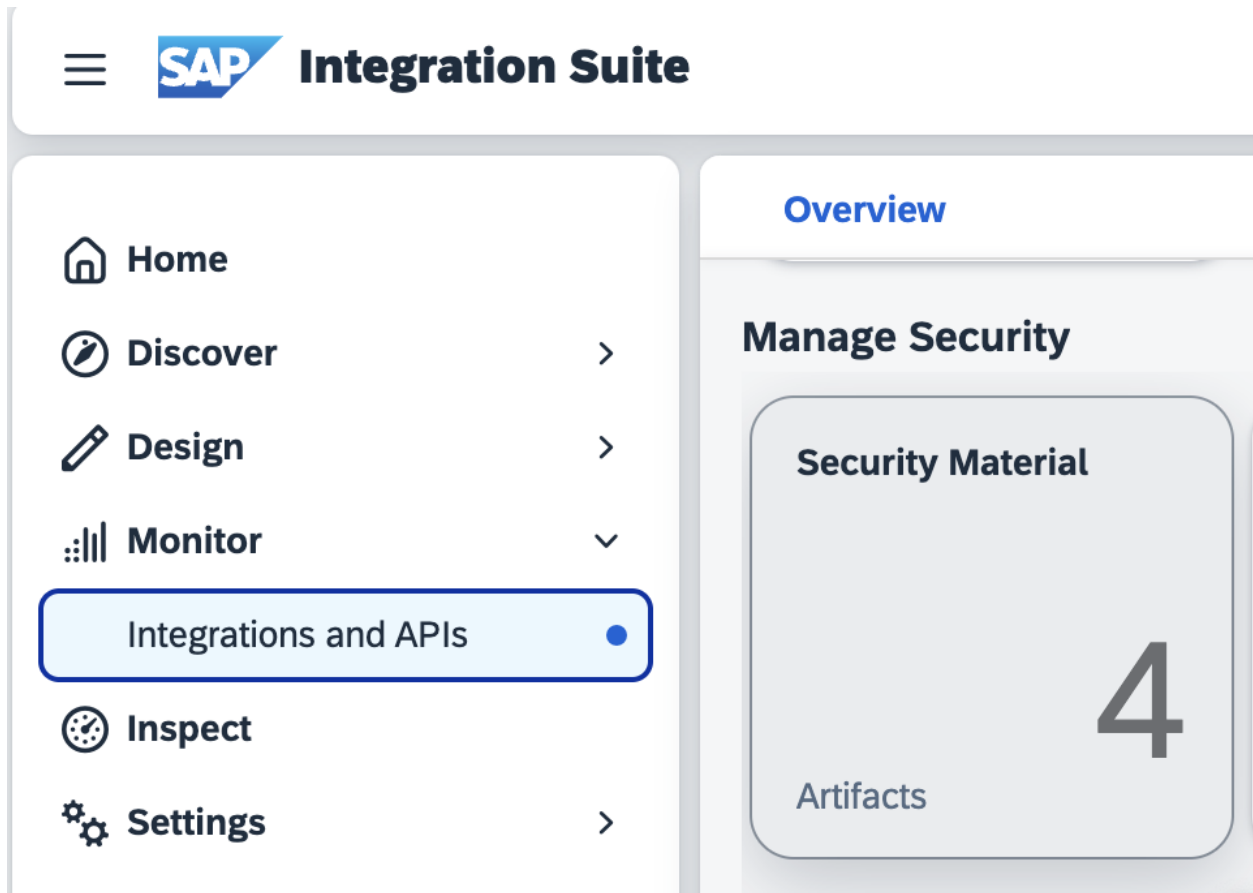


Figure 4.5 Navigating to SAP Integration Suite Monitoring

4.3.2 Create OAuth2 Client Credentials for the generation of Shopify access token.

1. Go to Create > OAuth2 Client Credentials.

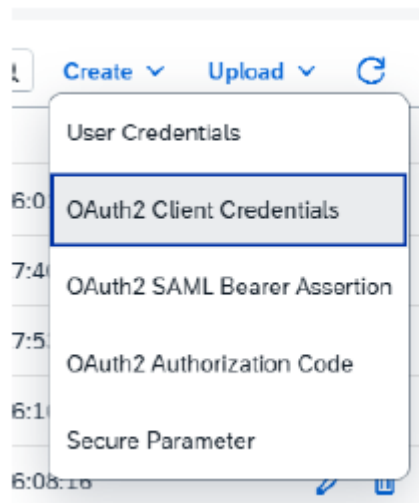


Figure 4.6 Creation of a new OAuth2 Client Credentials

2. Enter the OAuth2 Client Credentials details

The screenshot shows a configuration form for OAuth2 Client Credentials. The fields are as follows:

- Name:** * shopifyOAuthClientCredentials
- Description:** (empty)
- Token Service URL:** * https://xxxxx.myshopify.com/admin/oauth/access_token
- Grant Type:** * Send as Part of Body
- Client ID:** * 849f6c408ec7b0135d4465d0d0cf9469
- Client Secret:** * (empty)
- Client Authentication:** * Send as Request Header
- Scope:** (empty)
- Content Type:** application/x-www-form-urlencoded
- Resource:** (empty)
- Audience:** (empty)

Below the main form is a section for **Custom Parameters** with **Add** and **Delete** buttons. It contains a table with the following structure:

<input type="checkbox"/>	Key	Value	Send as Part of
No data			

At the bottom right of the form are **Deploy** and **Cancel** buttons.

Figure 4.7 Creation of a new OAuth2 Client Credentials

The OAuth2 Client Credentials Configuration contains parameters that define how the token request to Shopify's Admin OAuth endpoint should be executed:

Parameter	Description	Mandatory / Optional
Name	The technical name of the OAuth2 credential configuration For Example: <ul style="list-style-type: none"> shopifyOAuthClientCredentials 	M
Description	Provides an optional explanation for the OAuth2 credential configuration. This field is for documentation purposes only and helps identify the configuration within your integration.	O
Token Service URL	Full URL path to Shopify OAuth endpoint for generation of the access token - The path is: /admin/oauth/access_token Full URL: https://{your-store}.myshopify.com/admin/oauth/access_token	M
Grant type	Defines how the OAuth grant will be sent. For Example: <ul style="list-style-type: none"> Send as Part of Body 	M
Client ID	Take the client ID from your custom app - defined in the Shopify custom app section - Section 5.2	M
Client Secret	Take the client secret from your custom app - defined in the Shopify custom app section Section 5.2	M
Client Authentication	Defines how the client credentials will be sent. For Example: <ul style="list-style-type: none"> Send as Request Header 	M
Scope	Defines the level of access requested during authorization. The scopes are set in the Shopify Custom app described in Section 5.2 . This parameter is not required and can be left empty.	O
Content Type	Defines the format of the request sent to the token endpoint.	M
Resource	Identifies the target resource the credential is intended to access. This parameter is not required for Shopify and can remain empty.	O

Parameter	Description	Mandatory / Optional
Audience	Specifies the intended recipient (audience) of the issued authorization. Shopify does not use this parameter, so it is usually left blank.	O

4.4 Usage of the OAuth2 Client Credentials

The Security artifact should be used in the Connection tab of the Adapter as shown in Figure 4.8.

The option “OAuth2 Client Credentials” must be selected. Detailed explanation of configuring the connection properties can be found in [Section 6.1](#).

The screenshot displays the configuration interface for the Shopify adapter, specifically the 'Connection' tab. It is divided into two main sections: 'CONNECTION DETAILS' and 'SHOPIFY HEADERS'. Under 'CONNECTION DETAILS', the 'Shopify Base Url' is entered as 'https://...shopify.com', the 'Authentication Type' is set to 'OAuth2 Client Credentials' (indicated by a dropdown arrow), and the 'Shopify Security Material' is 'shopifyOAuthAccessToken'. The 'Attach Error Details on Failure' checkbox is checked. Under 'SHOPIFY HEADERS', the 'Response headers' field contains an asterisk (*).

Figure 4.8 Using the OAuth2 Client Credentials in Adapter Configuration

5. Shopify Configuration

5.1 Shopify legacy app creation

The following steps can be followed in order to create a new Shopify App and generate an access token.

1. Login in the Shopify Admin Portal
2. Navigate to “Apps and sales channels”

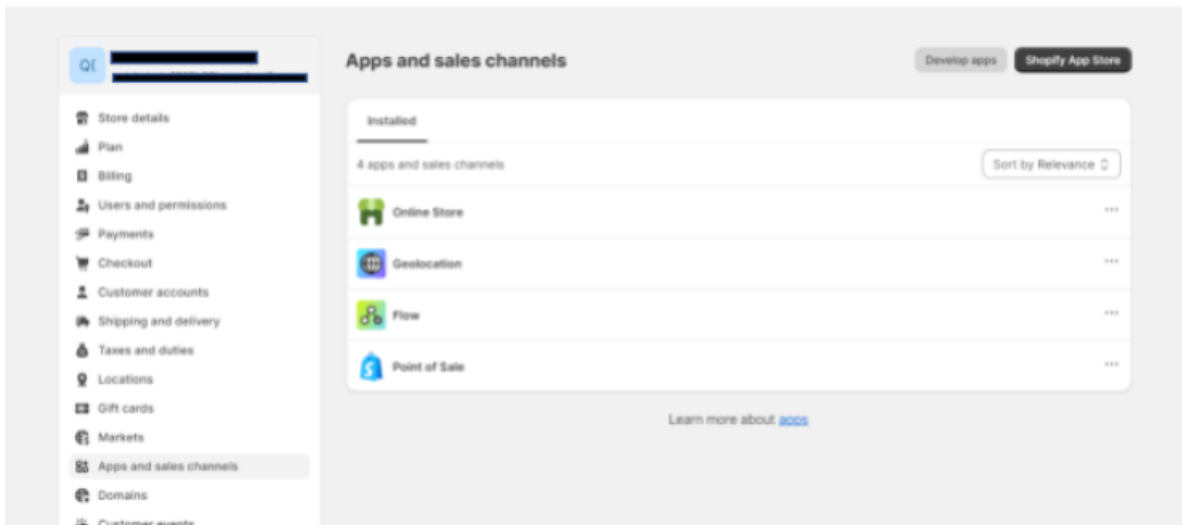


Figure 5.1 Shopify Admin Portal Apps and sales channels

3. Click “Develop apps”

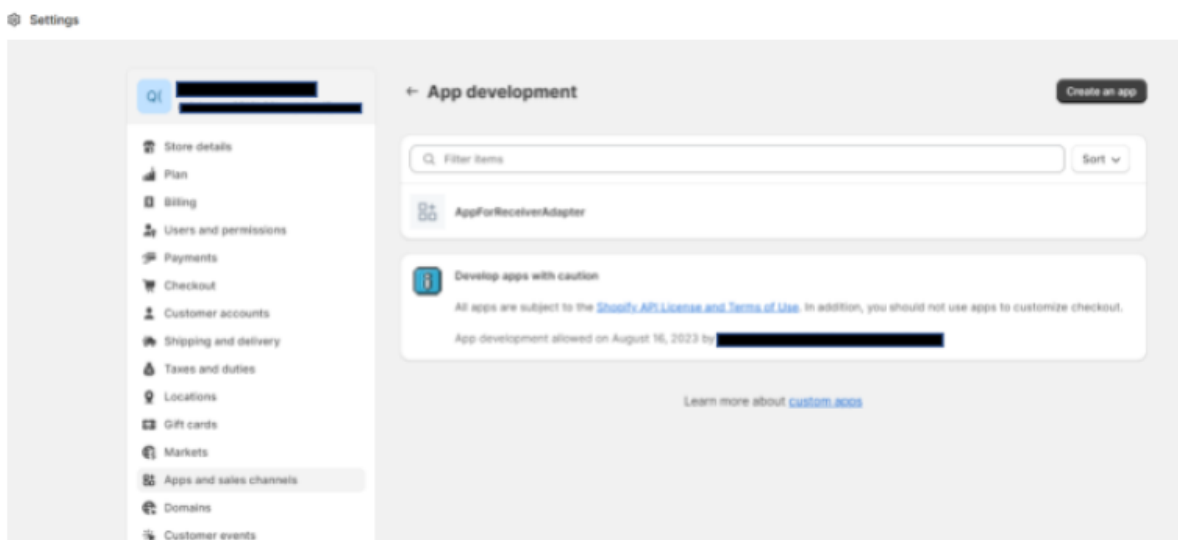


Figure 5.2 Creation of a New App

4. Click on **“Create an app”**

Shopify API License and Terms of Use.' At the bottom right are 'Cancel' and 'Create app' buttons."/>

Create an app [X]

App name

App developer

[Key icon] [Blurred name] [Dropdown arrow]

Can only be assigned to staff and collaborators with permission to develop apps. We'll contact them about development updates, like changes to Shopify APIs.

Apps are subject to the [Shopify API License and Terms of Use](#).

[Cancel] [Create app]

Figure 5.3 Creation of a New App

5. Add an App name and then click the Create app button
6. Click on Configure Admin API scopes

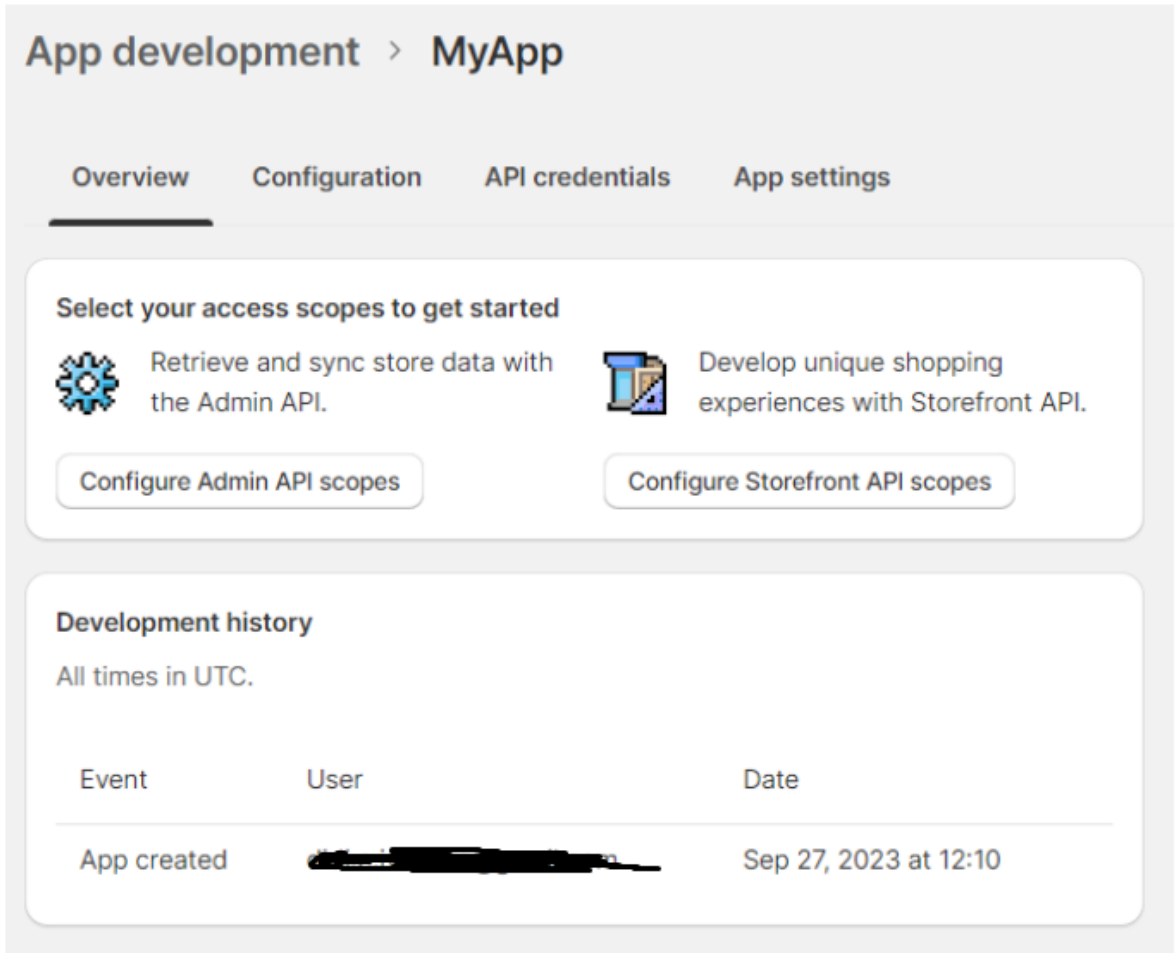


Figure 5.4 Creation of a New App

7. Select the necessary permissions

All Selected **6**

Q product

Browsing behavior

ⓘ Includes personally identifiable information

View or manage online-store browsing behavior including page views, cart updates, product views and searches read_customer_events

Product feeds

View or manage product feeds write_product_feeds
|
 read_product_feeds

Product listings

View or manage product or collection listings write_product_listings
|
 read_product_listings

Products

View or manage products, variants, and collections write_products
|
 read_products

Publications

View or manage groups of products that have been published to an app write_publications
|
 read_publications

Figure 5.5 Granting Permissions for the new Shopify App

8. Save the changes.
9. Select “API credentials” tab and then click “Install app”

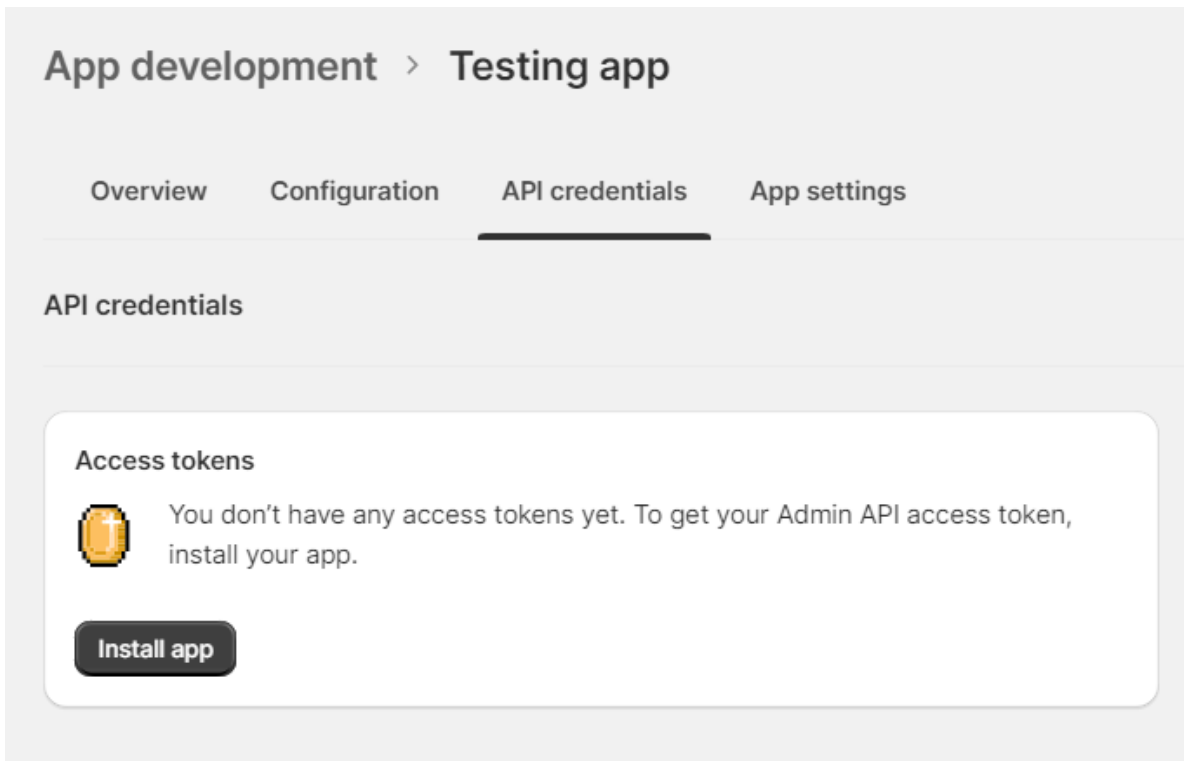


Figure 5.6 Installing the new app

10. In “API Credentials” tab you can reveal the token once which must be safely stored and afterwards configured in SAP Integration Suite

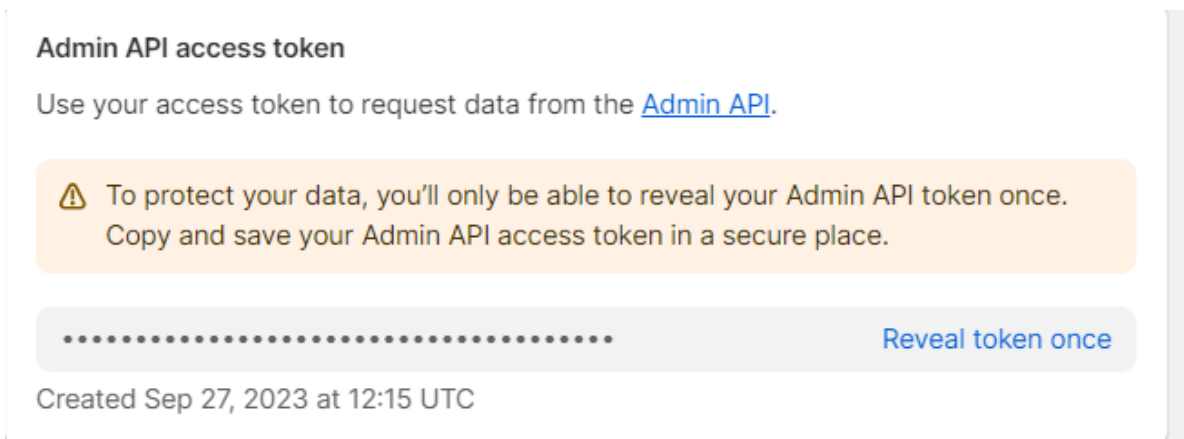


Figure 5.7 Revealing the access token for the new app

11. The configuration and access scope of the selected app can be observed

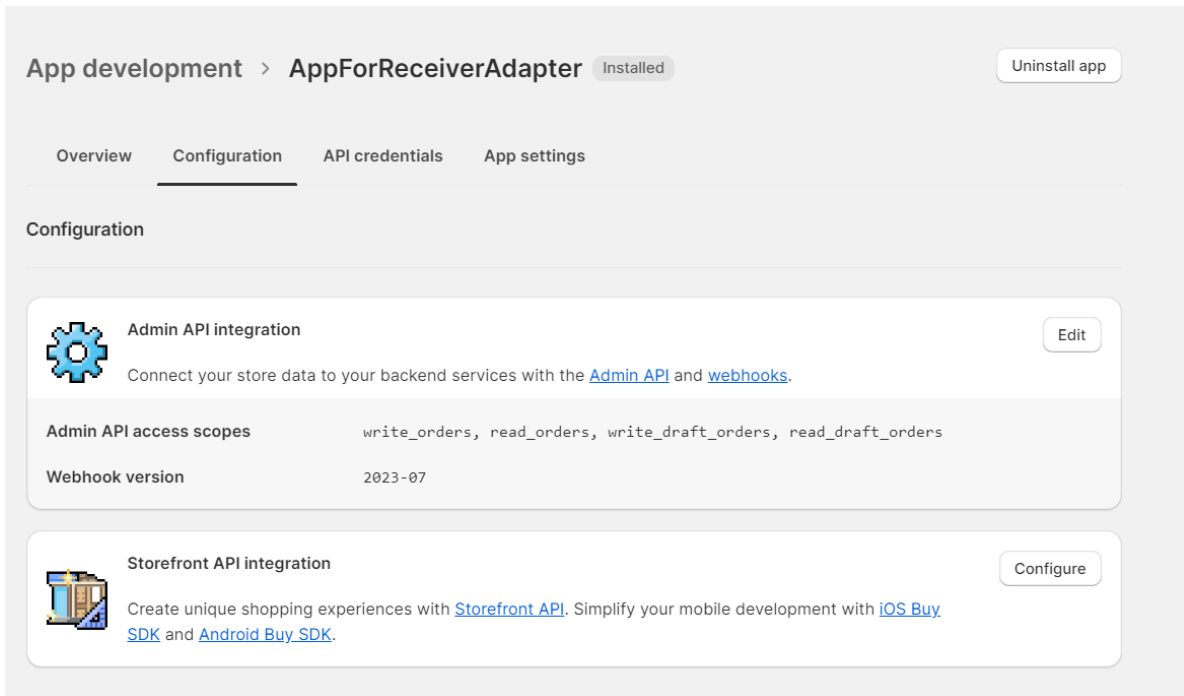


Figure 5.8 Configuration of the new app

5.2 Shopify custom app creation

The following steps can be followed in order to create a new Shopify Custom App and generate OAuth2 Client Credentials.

1. Login in the Shopify Admin Portal
2. Navigate to “Apps and sales channels”

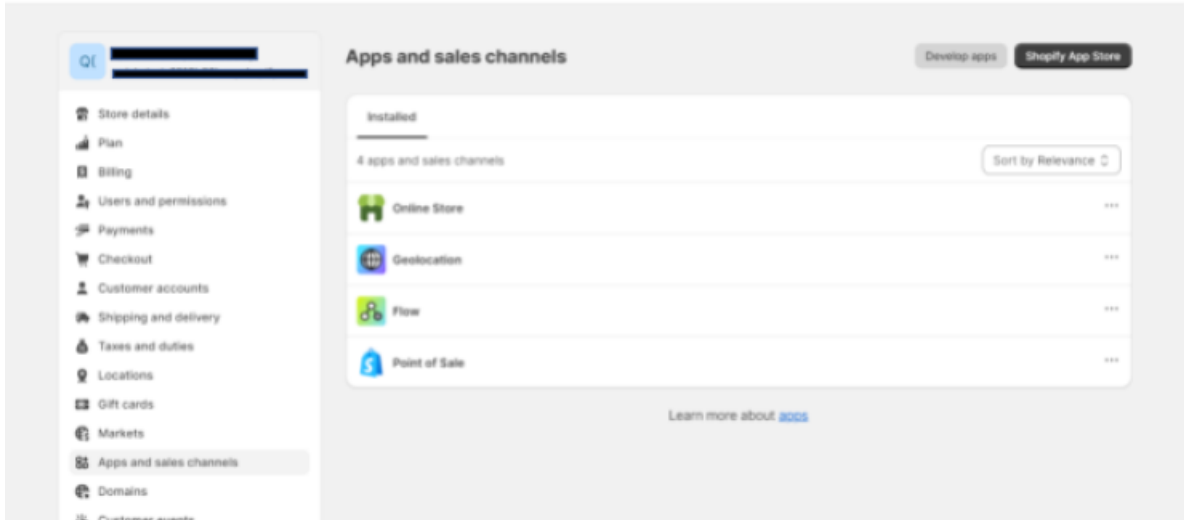


Figure 5.9 Shopify Admin Portal Apps and sales channels

3. Click “Develop apps”

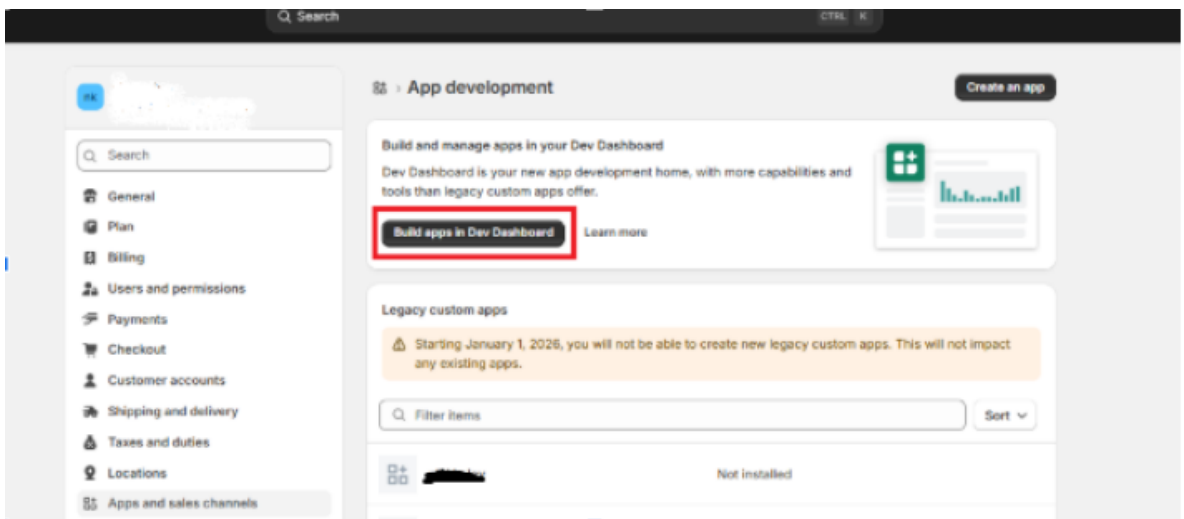


Figure 5.10 Creation of a New App

4. Click on “Build apps in Dev Dashboard”

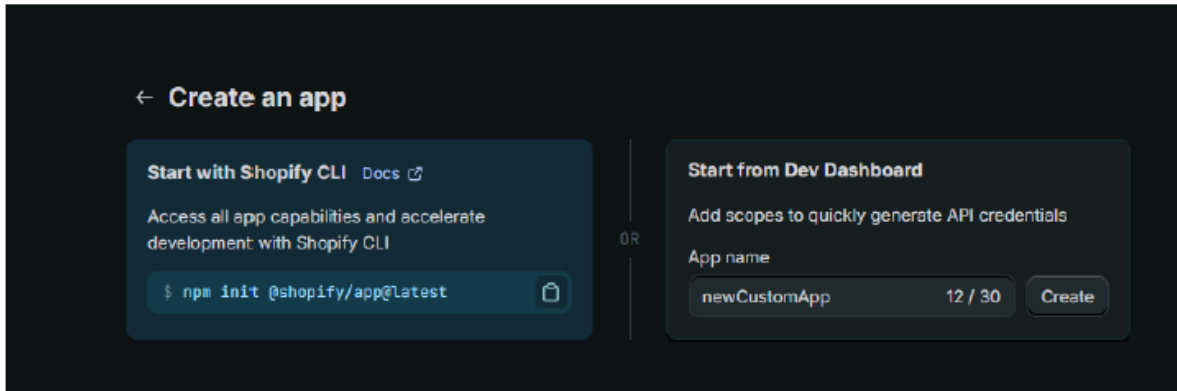


Figure 5.11 Select the name and create

5. Enter your store link, choose the needed scopes

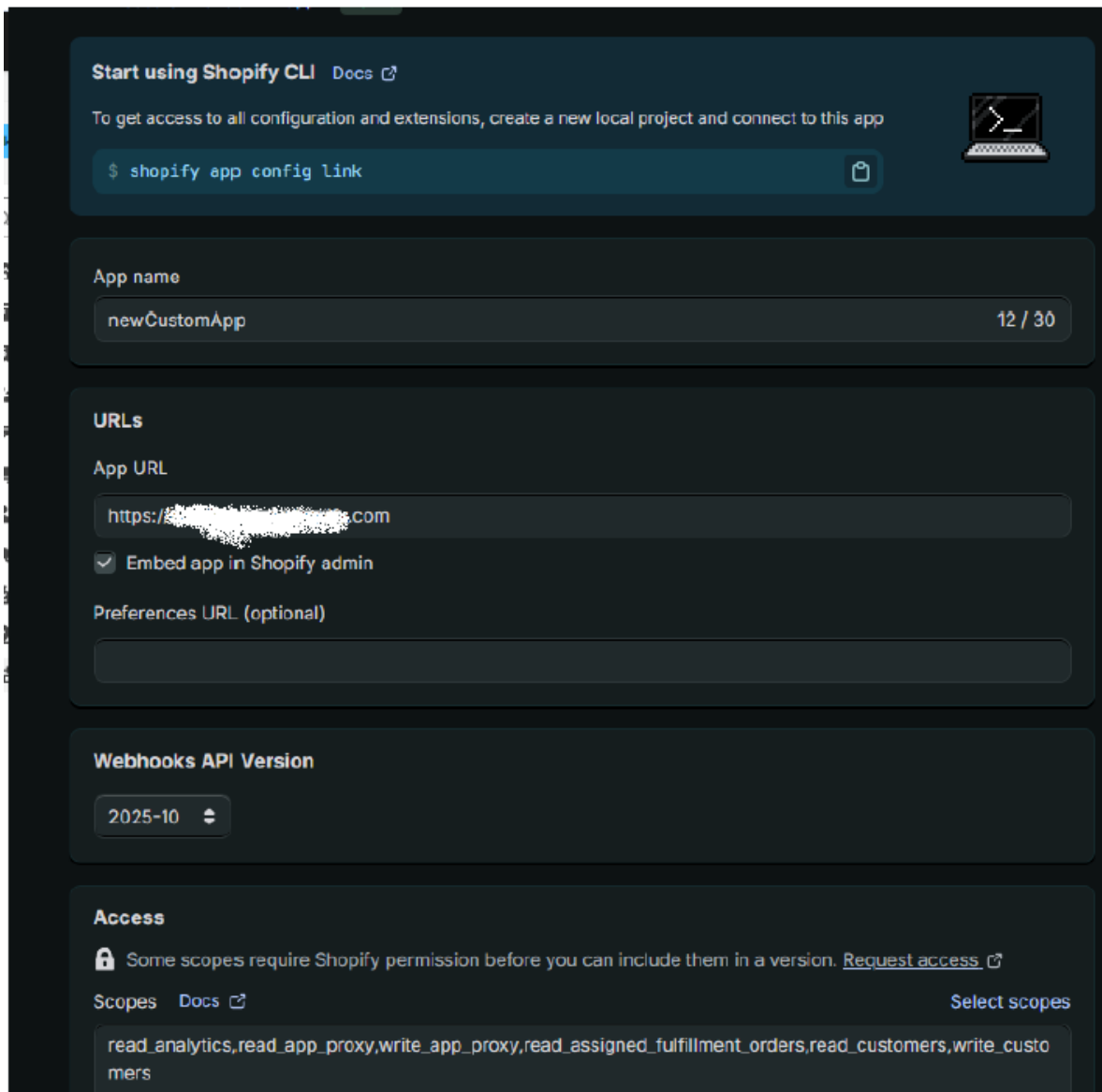


Figure 5.12 Adding your store URL

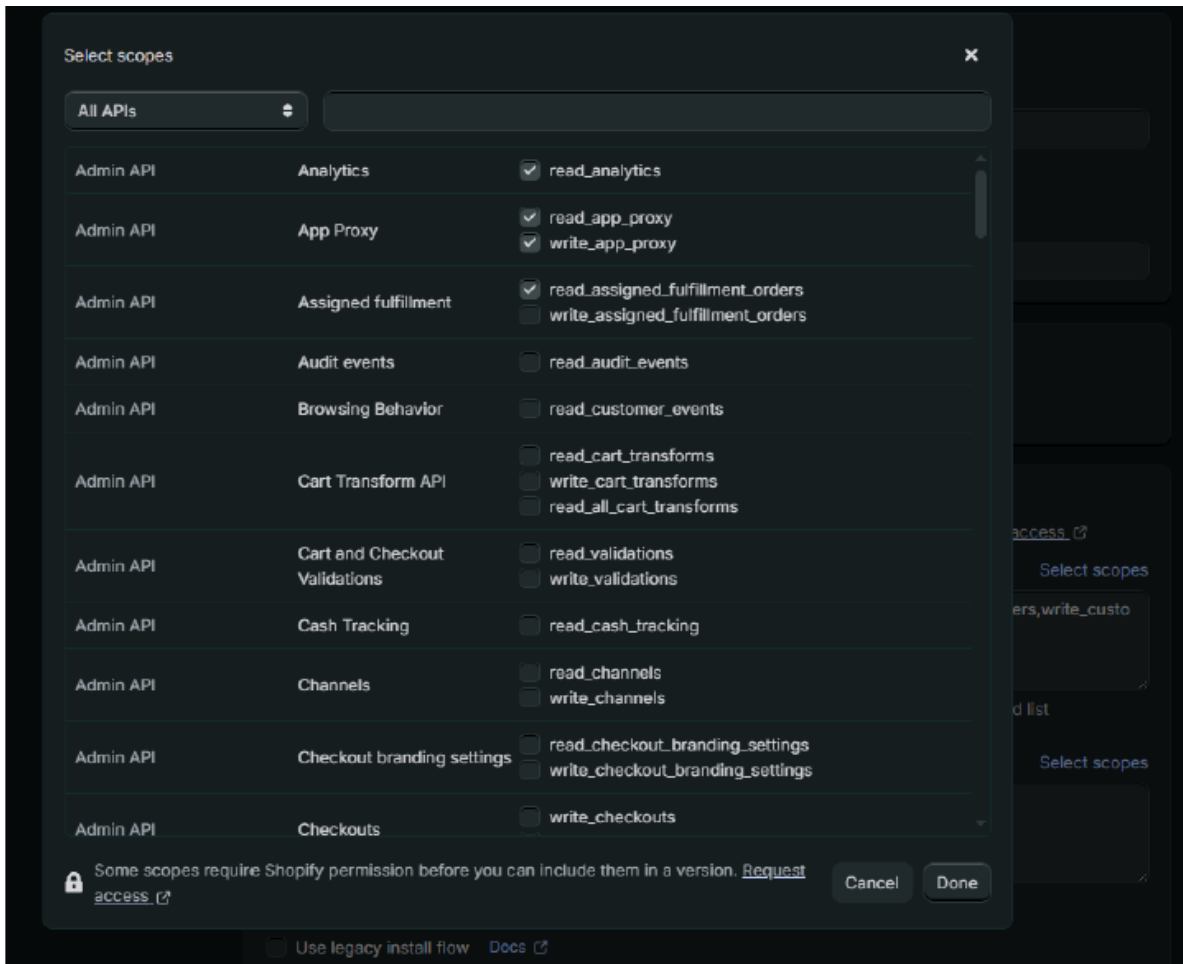


Figure 5.12 Adding the needed scopes

6. Leave “Legacy install flow” unchecked, POS and App Proxy - unchecked

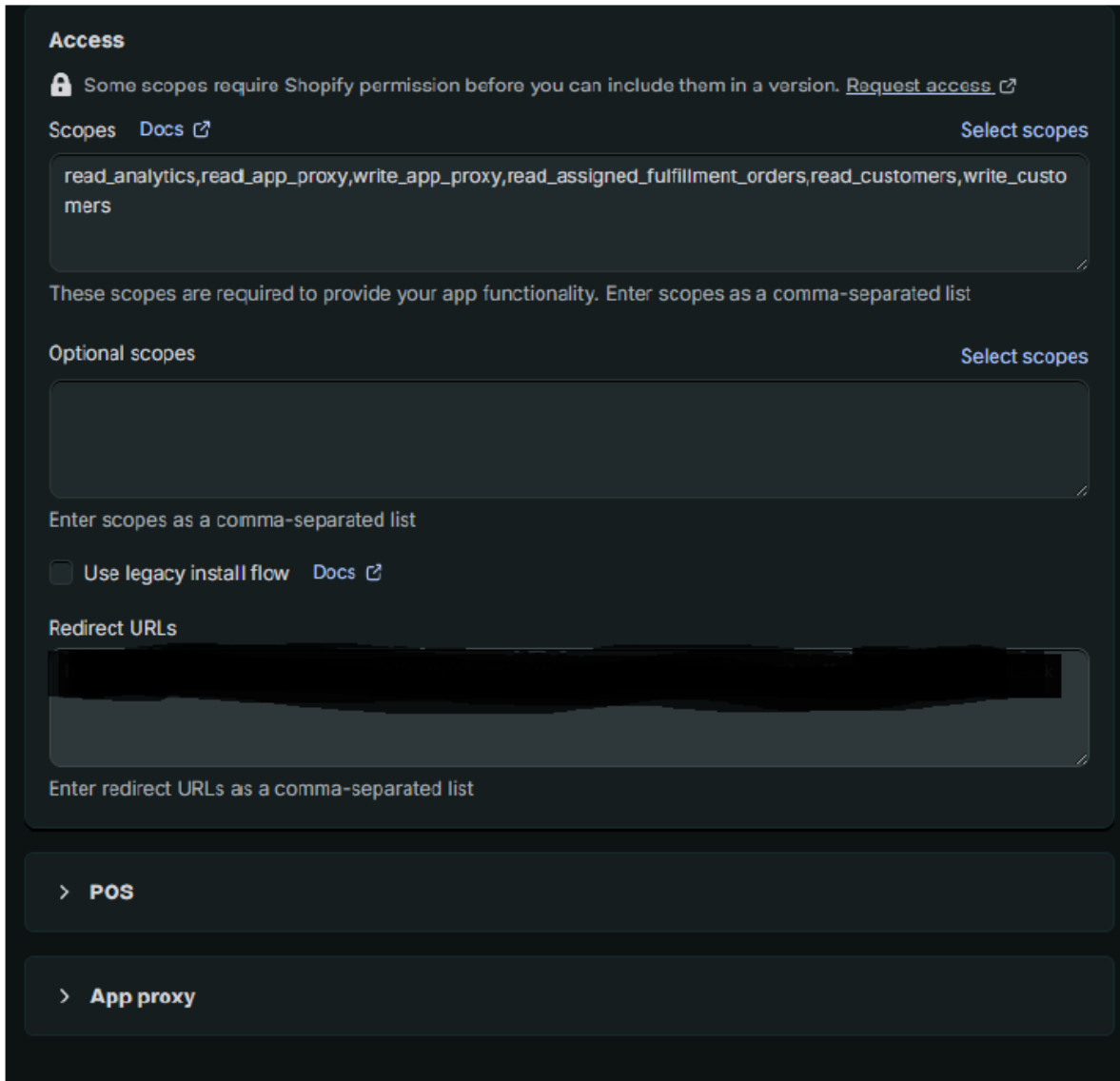


Figure 5.13 Leave the mentioned options unchecked

7. Fill in the “Redirect URL’s” with these paths shown in Figure 5.14

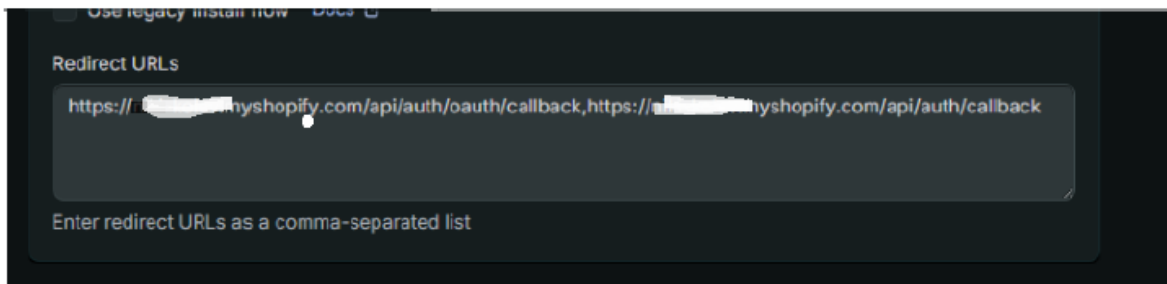


Figure 5.14 Fill in the needed redirect urls for OAuth 2 Client Credentials

8. Click on “Settings” and select the “Manage” button next to “Turn your app into a sales channel”

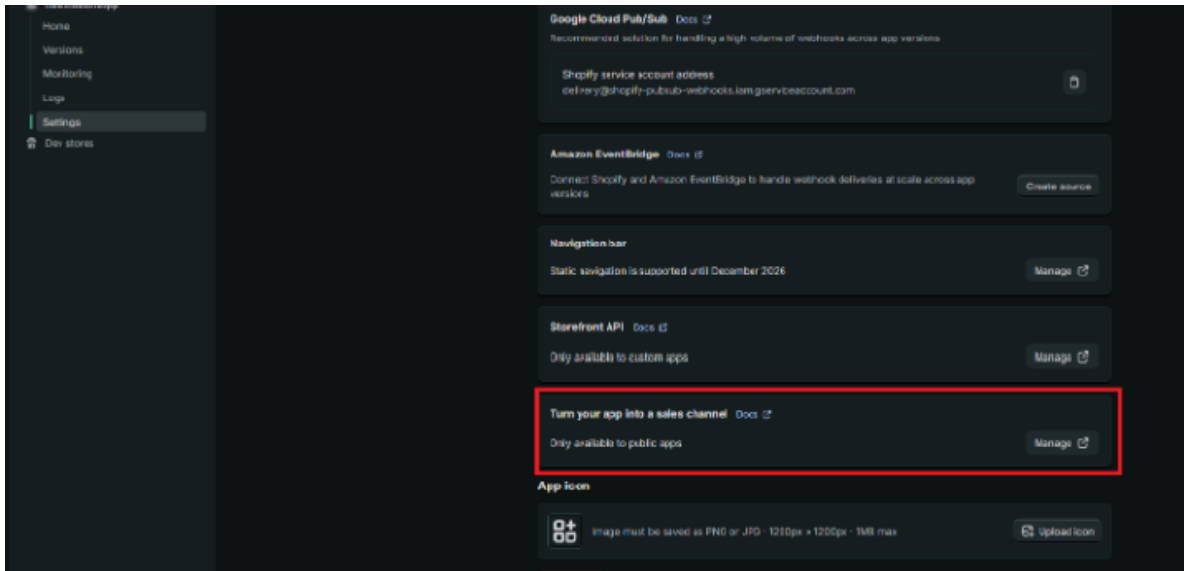


Figure 5.14 Clicking on "Manage button"

9. Select "Distribution" and make your app custom

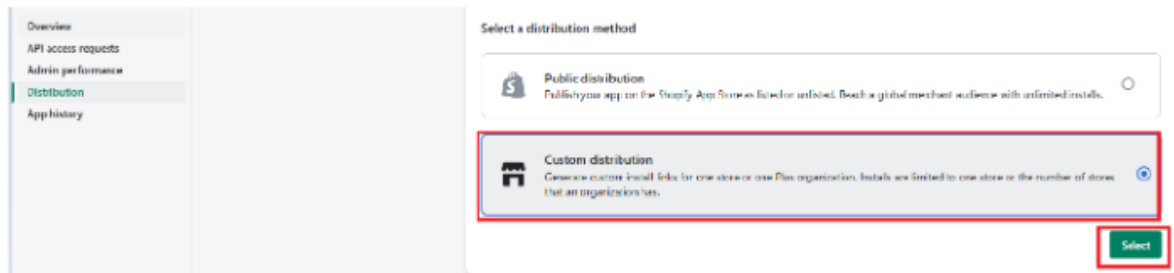


Figure 5.15 Create custom distribution

10. "Allow multi-store install for one Plus organization" must be unchecked -> Generate link

Distribution

Custom distribution

Store domain

shop1.myshopify.com

For example: shop1.myshopify.com or admin.shopify.com/store/shop1

Allow multi-store install for one Plus organization

This app can be installed on stores that belong to the same Plus organization as the store provided

Generate link

Figure 5.16 Generate link for installation

11. Copy the generated link and paste it into your browser. Install the app on your store

Install app

ⓘ This app is exclusive to your store

newCustomApp was created exclusively for your stores and can't be installed by anyone else. Contact the app developer if you need more information.

This app needs access to:

View personal data
Customers, store owner

View and edit store data
Customers, orders, store analytics, Online Store

Cancel **Install**

Figure 5.17 Installation of your custom app into your store

Note: You have successfully installed your app

12. Find your client credentials in Settings tab in the Dev Dashboard

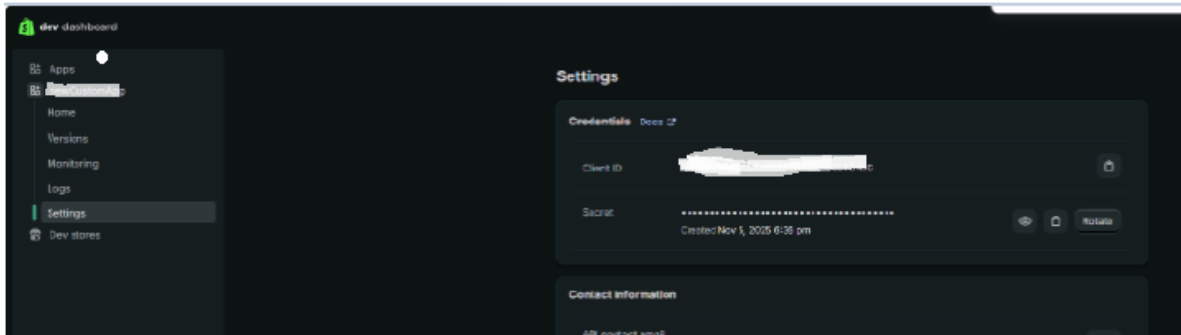


Figure 5.18 Client credentials used in Section 4.4

6. Receiver Adapter Configuration

This section describes the different parameters that can be configured for the Receiver Adapter. This Adapter has Connection and Processing Tabs. Each one of these will be explained in the next sections.

6.1 Connection

The Connection Tab contains parameters that define how to connect and authorize against Shopify's Admin APIs

- Shopify Base Url – the Base URL of the Shopify Store.
- Authentication Type - Dropdown option to select which way you want to authenticate. Option 1 - Access Token, described in [Section 4.2](#). Option 2 - OAuth2 Client Credentials, described in [Section 4.4](#).
- Shopify Security Material - Provides an optional explanation or label for the OAuth2 credential configuration. This field is for documentation purposes only and helps identify the configuration within your integration.
- Attach Error Details on Failure - By default, the option is enabled. This option enables the creation of attachments for request headers, response headers, and response body when the message processing fails. Having these attachments during message processing failures can be unnecessary as it leads to persistence of attachments that doesn't help. Especially, if multiple message processing failures occur, you have attachments piled up for each failure. If you don't require the attachments for failure scenarios, disable the option. Though you disable the creation of attachments, the content of the same are added to the message processing logs.

- Response Headers - Provides a pipe - separated (|) value list of HTTP response headers. The received header values are then converted to message/exchange headers.

If the value * is entered, all the HTTP response header values are converted to message/exchange headers.

The screenshot displays the configuration interface for the Shopify adapter, titled "Shopify". It features three tabs: "General", "Connection" (which is selected), and "Processing".

Under the "CONNECTION DETAILS" section, the following fields are visible:

- Shopify Base Url:** A text input field containing "https://...myshopify.com".
- Authentication Type:** A dropdown menu set to "OAuth2 Client Credentials".
- Shopify Security Material:** A text input field containing "shopifyOAuthClientCredentials".
- Attach Error Details on Failure:** A checkbox that is checked.

Under the "SHOPIFY HEADERS" section, the following field is visible:

- Response headers:** A text input field containing an asterisk (*).

Figure 6.1 Connection Configuration for the Shopify Receiver Adapter

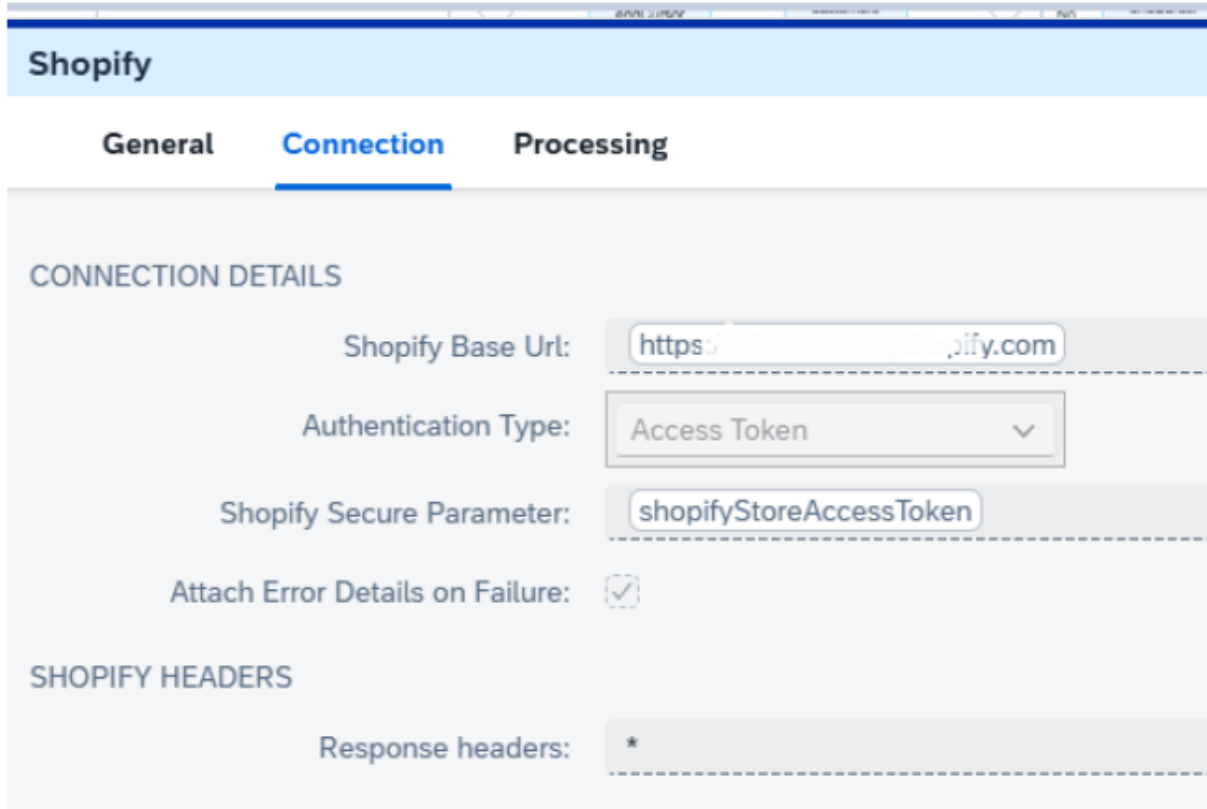


Figure 6.1.1 Connection Configuration for the Shopify Receiver Adapter with Access Token authorization

6.2. Processing Configuration for REST Admin API

The Processing tab contains all the operation configurations for the selected Shopify API with an example configuration for Shopify REST Admin API shown below

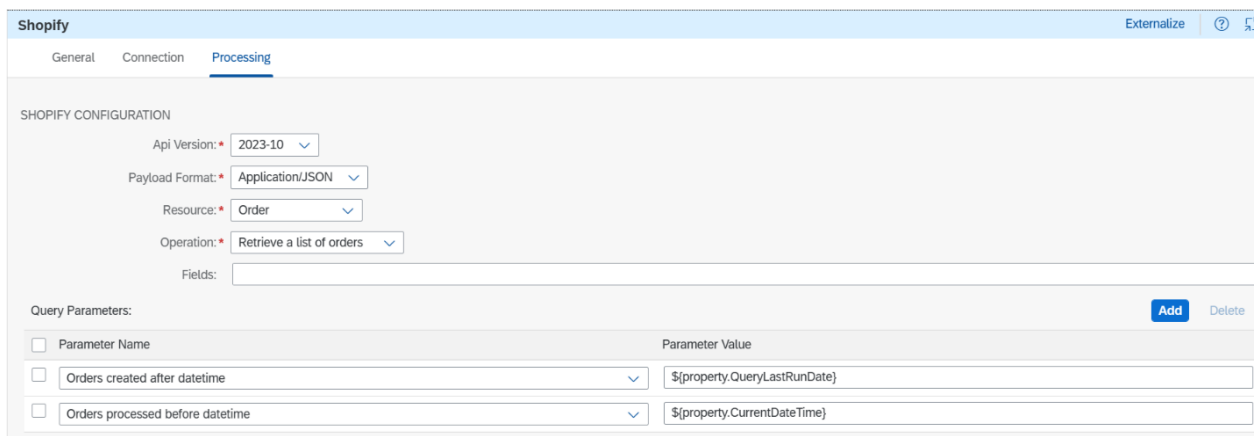


Figure 6.2 Processing Configuration for the Shopify Receiver Adapter

Mandatory fields

- **API Version** - select the version of Shopify Admin API to be used from the API Version dropdown menu.
- **Resource** - specify the type of resource. Resources refer to the fundamental data structures or objects that represent different aspects of an online store.
- **Operation** – specify an operation for the currently selected resource. Operations are actions which can be performed over a certain resource. They can modify or read the resource.

Mandatory & Optional Fields

The table below highlights the Resources and each Operation’s Mandatory & Optional Parameters.

Cells that hold query parameter values in the “Optional Parameter” column are marked with a blue background.

Re-open a closed order Resource ID

Shopify Adapter Guide

Resource	Operation	Mandatory Parameters	Optional Parameters
Collect	Add product to a custom collection		
	Count collects		
	Remove a product from a collection	Resource ID	
	Retrieve a list of collects		Fields
			Maximum results on a page
Retrieve only collects after the specified ID			
Retrieve a specific collect	Resource ID	Fields	
Collection	Retrieve a list of products belonging to a collection	Resource ID	The number of products to retrieve
	Retrieve a specific collection	Resource ID	Fields
Custom Collection	Create a custom collection		
	Count custom collections		Collections published after datetime
			Collections published before datetime
Collections updated after datetime			

Shopify Adapter Guide

Resource	Operation	Mandatory Parameters	Optional Parameters	
			Collections update before datetime	
			Product ID	
			Published status	
			Title	
	Delete a custom collection	Resource ID		
	Retrieve a list of custom collections			Fields
				Collection IDs
				Collections published after datetime
				Collections published before datetime
				Collections updated after datetime
				Collections updated before datetime
				Handle
				Maximum results on a page
				Product ID
Published status				
Show collections after the specified ID				
Title				

Shopify Adapter Guide

Resource	Operation	Mandatory Parameters	Optional Parameters	
	Retrieve a specific custom collection	Resource ID	Fields	
	Update a custom collection	Resource ID		
Customer	Create a customer			
	Create an account activation URL for a customer	Resource ID		
	Send an account invite to a customer	Resource ID		
	Retrieve a list of customers			Customer IDs
				Customers created after datetime
				Customers created before datetime
				Customers updated after datetime
				Customers updated before datetime
				Fields
Maximum results on a page				
Show customers after the specified ID				
Delete a customer	Resource ID			

Shopify Adapter Guide

Resource	Operation	Mandatory Parameters	Optional Parameters	
	Retrieve a specific customer	Resource ID	Fields	
	Update a customer	Resource ID		
	Retrieve all orders that belong to a customer	Resource ID	Order Status	
	Customer count			Customers created after datetime
				Customers created before datetime
				Customers updated after datetime
				Customers updated before datetime
	Search for customers			Fields
				Maximum results on a page
				Order
				Query
Discount Code	Create a discount code creation job	Price Rule ID		
	Create a discount code	Price Rule ID		
	Count of discount codes			Show discount codes used greater than or equal to this value
				Show discount codes used less than or equal to this value

Shopify Adapter Guide

Resource	Operation	Mandatory Parameters	Optional Parameters
			Show discount codes with times used
	Retrieve a discount code by code name	Discount Code Name	
	Retrieve a discount code by ID	Price Rule ID	
		Discount Code ID	
	Retrieve a discount code creation job	Price Rule ID	
		Discount code creation job ID	
	Retrieve a list of discount codes	Price Rule ID	
	Retrieve a list of discount codes for a discount code creation job	Price Rule ID	
		Discount code creation job ID	
	Update a discount code	Price Rule ID	
		Discount code ID	
	Delete a discount code	Price Rule ID	
		Discount code ID	
Event	List of events		Fields
			Action
			Events created after datetime
			Events created before datetime

Shopify Adapter Guide

Resource	Operation	Mandatory Parameters	Optional Parameters
			Filter by resource
			Maximum results on a page
			Show events after the specified ID
	Count of events		Events created after datetime
			Events created before datetime
	Retrieve a specific event by ID	Resource ID	Fields
Inventory Item	Retrieve a list of inventory items	IDs	The maximum number of results to show
	Retrieve a specific inventory item	Resource ID	
	Update an inventory item	Resource ID	
Inventory Level	Adjust an inventory level	Available Adjustment	
		Inventory Item ID	
		Location ID	
	Connect an inventory item to a location	Inventory Item ID	Relocate if necessary
		Location ID	
	Delete an inventory location	Inventory Item ID	
Location ID			

Shopify Adapter Guide

Resource	Operation	Mandatory Parameters	Optional Parameters
	Retrieve a list of inventory levels		Location IDs
			Inventory item IDs
			Inventory levels updated after datetime
			The maximum number of results to show
	Set an inventory level	Inventory Item ID	Disconnect if necessary
		Location ID	
Available Adjustment			
Location	Count of locations		
	List of inventory levels for a location	Resource ID	
	List of locations		
	Retrieve a location	Resource ID	
Metafield	Count of a resource's metafields	Metafield's owner resource	
		Resource ID	
	Create a metafield	Metafield's owner resource	
		Resource ID	

Resource	Operation	Mandatory Parameters	Optional Parameters	
	Delete a metafield by its ID	Metafield's owner resource		
		Resource ID		
		Metafield ID		
	List of metafields by resource	Metafield's owner resource		Fields
				Maximum results on a page
				Metafield key
				Metafield namespace
				Metafields created after datetime
		Resource ID		Metafields created before datetime
				Metafields updated after datetime
				Metafields updated before datetime
				Show metafields after the specified ID
				Type of data
	List of metafields that belong to a resource by using query parameters	Metafield's owner resource		
		Resource ID		
Retrieve a specific metafield by ID	Metafield's owner resource		Fields	
	Resource ID			

Shopify Adapter Guide

Resource	Operation	Mandatory Parameters	Optional Parameters	
		Metafield ID		
	Update a metafield	Metafield's owner resource		
		Resource ID		
		Metafield ID		
Order	Cancel an order	Resource ID		
	Close an order	Resource ID		
	Create an order			
	Delete an order	Resource ID		
	Order count			Financial status
				Fulfilment status
				Order Status
				Orders created after datetime
				Orders created before datetime
				Orders updated after datetime
				Orders updated before datetime
Retrieve a list of orders			Fields	
			Financial status	
			Fulfilment status	
			Maximum results on a page	

Shopify Adapter Guide

Resource	Operation	Mandatory Parameters	Optional Parameters
			Order IDs
			Order Status
			Orders created after datetime
			Orders created before datetime
			Orders processed after datetime
			Orders processed before datetime
			Orders updated after datetime
			Orders updated before datetime
			Show orders after the specified ID
			Show orders attributed to a certain app
	Retrieve a specific order	Resource ID	Fields
	Update an order	Resource ID	
Order risk	Create an order risk	Order ID	
	Delete an order risk	Resource ID	
		Order ID	
	Retrieve a list of order risk	Order ID	
Retrieve a specific order risk		Resource ID	
		Order ID	

Shopify Adapter Guide

Resource	Operation	Mandatory Parameters	Optional Parameters	
	Update an order risk	Resource ID		
		Order ID		
Price Rule	Create a price rule			
	Delete a price rule	Resource ID		
	Price Rule count			
	Retrieve a list of price rules			Restrict results to after the specified ID
				Show price rule created before date
				Show price rules created after date
				Show price rules ending after date
				Show price rules ending before date
				Show price rules last updated after date
				Show price rules last updated before date
		Show price rule starting after date		
		Show price rule starting before date		
		Show price rules used * times		

Shopify Adapter Guide

Resource	Operation	Mandatory Parameters	Optional Parameters	
			The maximum number of results to retrieve	
	Retrieve a specific price rule	Resource ID		
	Update a price rule	Resource ID		
Product	Create a product			
	Delete a product	Resource ID		
	Product count			Collection ID
				Product type
				Products created after datetime
				Products created before datetime
				Products updated after datetime
				Products updated before datetime
				Published status
	Vendor			
	Retrieve a list of products			Fields
				Collection ID
				Handle
Maximum results on a page				
Presentment Currencies				
			Product IDs	

Shopify Adapter Guide

Resource	Operation	Mandatory Parameters	Optional Parameters
			Product type
			Products created after datetime
			Products created before datetime
			Products published after datetime
			Products published before datetime
			Products updated after datetime
			Products updated before datetime
			Published status
			Show products after the specified ID
			Status
			Title
			Vendor
	Retrieve a specific product	Resource ID	Fields
	Update a product	Resource ID	
Products Variants	Create a new product variant	Product ID	
	Modify an existing product variant	Resource ID	
	Product Variant count	Product ID	

Shopify Adapter Guide

Resource	Operation	Mandatory Parameters	Optional Parameters
	Retrieve a single product variant	Resource ID	
	Remove an existing product variant	Resource ID	
		Product ID	
	Retrieve a list of product variants	Product ID	Currency codes
			Maximum results per page
			Restrict result to after the specified ID
Refund	Calculate a refund for an order	Order ID	
	Create a new order refund	Order ID	
	Retrieve a list of order refunds	Order ID	Fields to show
			Maximum results on a page
			Show amounts in the shop currency
	Retrieve a single order refund	Resource ID	Fields to show
Order ID		Show amounts in the shop currency	
Smart Collection	Count smart collection		Collections published after datetime
			Collections published before datetime

Shopify Adapter Guide

Resource	Operation	Mandatory Parameters	Optional Parameters
			Collections updated after datetime
			Collections updated before datetime
			Product ID
			Published status
			Title
	Create a smart collection		
	Delete a smart collection	Resource ID	
	Retrieve a list of smart collections		Collection IDs
			Collections published after datetime
			Collections published before datetime
			Collections updated after datetime
			Collections updated before datetime
			Handle
			Maximum results on a page
			Product ID
			Published status
			Show collections after the specified ID

Shopify Adapter Guide

Resource	Operation	Mandatory Parameters	Optional Parameters
			Title
	Retrieve a specific smart collection	Resource ID	Fields
	Update a smart collection	Resource ID	
	Update the ordering type	Resource ID	Sort type Product IDs
Tender transaction	Retrieve a list of tender transactions		Maximum results on a page
			Show tender transaction after the specified ID
			Show tender transactions ordered by processed_at
			Tender transactions processed after datetime
			Tender transactions processed at the specified date
			Tender transaction processed before datetime
Transaction	Count of an order's transaction	Order ID	
	Create a transaction	Order ID	Origin of the transaction
	List of transactions	Order ID	Fields to show

Shopify Adapter Guide

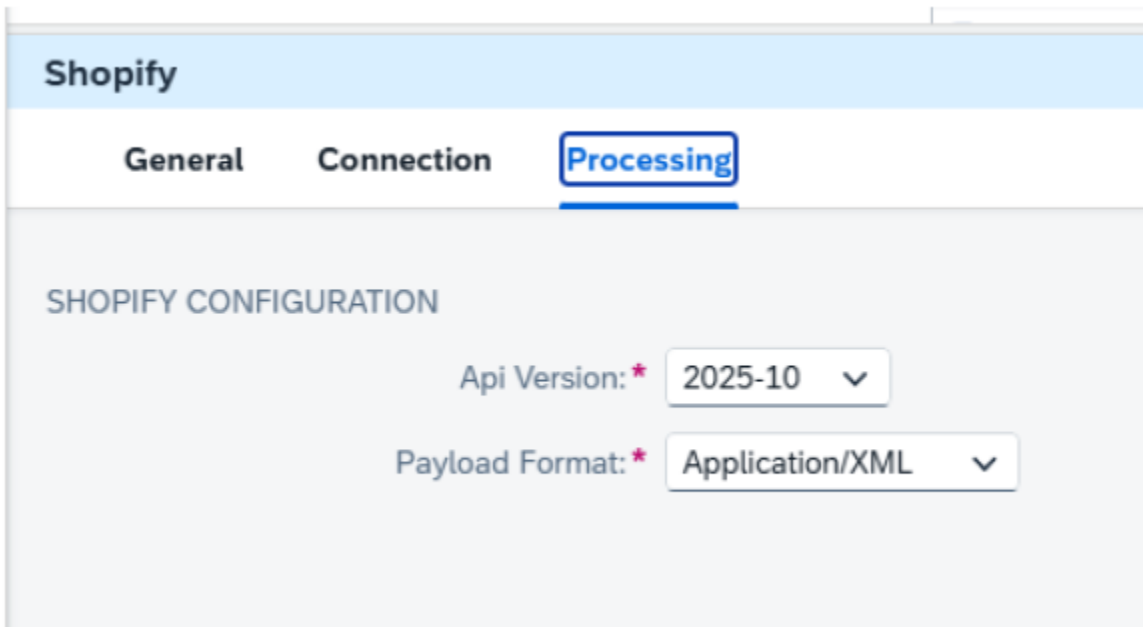
Resource	Operation	Mandatory Parameters	Optional Parameters
			Retrieve only transactions after the specified ID
			Show amounts in the shop currency
Balance	Retrieve the account's current balance		
Payouts	Return a list of all payouts		Filter the response to payouts made on the specified date.
			Filter the response to payouts made inclusively before the specified date.
			Filter the response to payouts made inclusively after the specified date.
			Filter the response to payouts made before the specified ID.
			Filter the response to payouts made after the specified ID.
			Filter the response to payouts made with the specified status.
	Return a single payout	Resource ID	
Balance Transactions	Retrieves a list of all balance transactions		Filter response to transactions exclusively before the specified ID
			Filter response to transactions paid out in the specified payout.

Shopify Adapter Guide

Resource	Operation	Mandatory Parameters	Optional Parameters
			Filter response to transactions with the specified payout status
			Filter response to transactions exclusively after the specified ID.
			Filter response to transactions placed in test mode.
Dispute	Retrieve a list of all disputes		Return only disputes with the specified initiated_at date
			Return only disputes before the specified ID.
			Return only disputes after the specified ID.
			Return only disputes with the specified status.
	Retrieves a single dispute by ID	Resource ID	
Dispute evidence	Updates dispute evidence associated with the dispute ID.	Dispute ID	
	Returns evidence associated with the dispute	Dispute ID	
Dispute File Upload	Upload a file of a category to a dispute.	Dispute ID	
	Delete a dispute evidence file given an ID.	Resource ID	
		Dispute ID	

6.3. Processing for GraphQL

The Processing Configuration contains information regarding Api Version and Payload Format.



The screenshot shows a configuration window titled "Shopify". It has three tabs: "General", "Connection", and "Processing", with "Processing" being the active tab. Below the tabs, the section is titled "SHOPIFY CONFIGURATION". There are two dropdown menus: "Api Version: *" with the value "2025-10" and "Payload Format: *" with the value "Application/XML".

7. Payload and Dynamic Configuration

7.1 Message Payloads

A content modifier is used to attach a payload to be sent to Shopify Admin API. Payloads must be in JSON format.

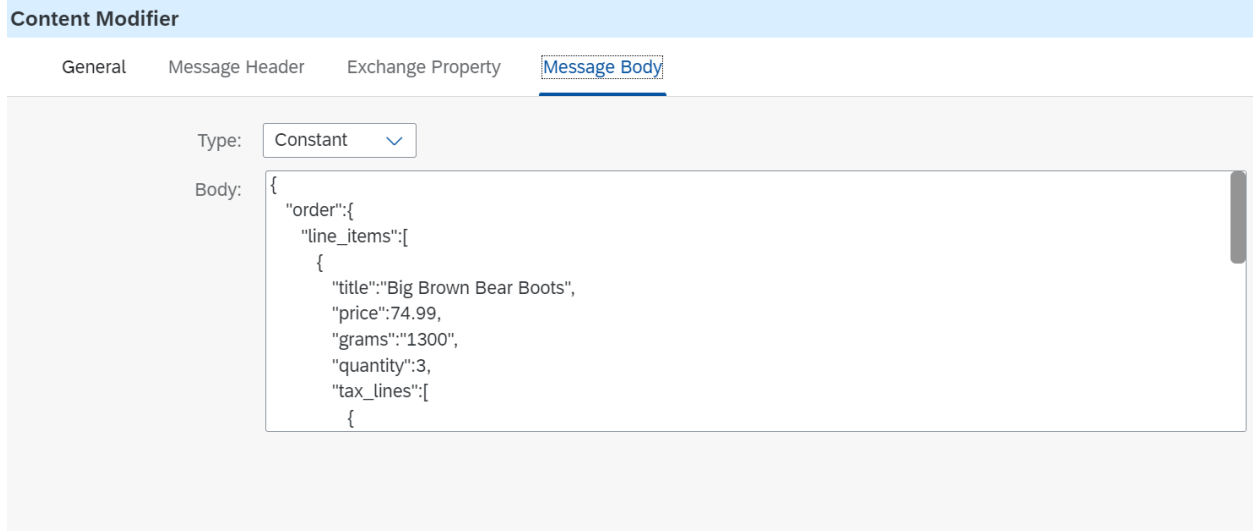


Figure 7.1.1

Shopify GraphQL API payloads are in GraphQL format and have a specific structure.

GraphQL queries can be either a mutation or a query. They can be simplified by extracting data into separate input variables. GraphQL variables let you reuse the same requests with different arguments.

The structure of the request with input variables must be the following

```

{
  "query": "...",
  "variables": {...}
}

```

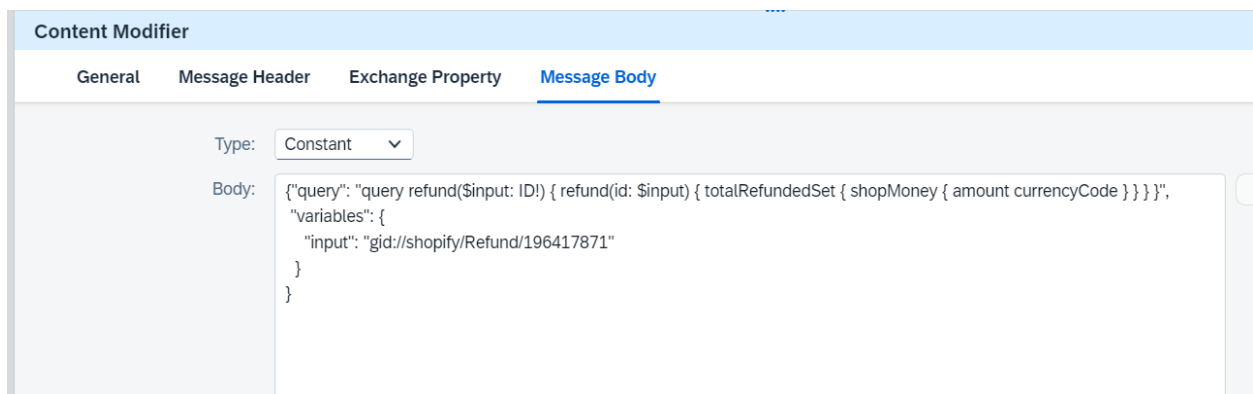


Fig 7.1.2

The variable section contains the input variables, which are defined as a JSON object. If variables are present in the query but are not defined correctly or are missing, an error would occur in the integration flow with a message indicating the problem.

The structure of the request *without* input variables must be the following

```
{
  "query" : "..."}
}
```

To build a valid query or mutation, the user can utilize the provided GraphiQL in-browser tool for writing, validating, and testing GraphQL queries. When working with Shopify's GraphQL Admin API, GraphiQL can be useful for learning about the available queries and mutations.

This following is a read-only demo in which queries can be executed without the need to connect to a development store.

1. Navigate to <https://shopify.dev/graphiql/admin-graphiql>
2. Click **Explorer** and use checkboxes and fields to build GraphQL queries.
3. Click **Docs** to expand the **Documentation Explorer**. You can search schemas to view descriptions and supported fields.

To understand more about GraphQL for the Shopify Admin API, go to [GraphiQL for the Admin API](#)

7.2 Dynamic configuration

This section illustrates the use of dynamic values in the Adapter configuration. It depicts a simple Shopify “Retrieve a list of orders” operation configuration using a dynamic property reference.

Use a Content Modifier before the Request-Reply step in the integration flow. The Content Modifier should create an Exchange property. The name of the Exchange property is the name required to be referred to in the Shopify Adapter. Any name is allowed here. It is possible to use a header instead of a property to achieve the same results. In the example below we create 2 properties. The first one is of type local variable and its value is the value held in the lastQueriedDate source value. The second property is of type expression. Expressions have the following structure: \${...}

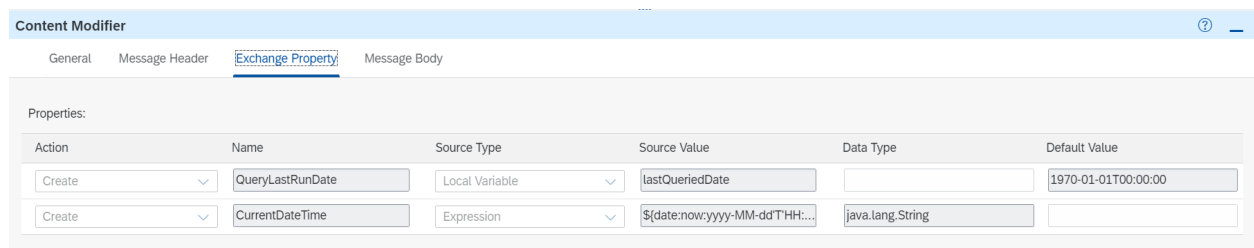


Figure 7.2

The Shopify Receiver Adapter (which is linked to the Request-Reply step) needs to refer to this property by using `${property.name}`. To refer to a header, a syntax similar to `${header.name}` should be used. As seen in Figure 7.2, the `QueryLastRunDate` property is referred to with the syntax `${property.QueryLastRunDate}`. The Shopify Adapter automatically replaces this reference with the datetime of the last run of the query. Also, `CurrentDateTime` is referred to by using `${property.CurrentDateTime}`. Fig. 7.3

Shopify

General Connection **Processing**

SHOPIFY CONFIGURATION

Api Version: * 2023-10

Payload Format: * Application/JSON

Resource: * Order

Operation: * Retrieve a list of orders

Fields:

Query Parameters:

<input type="checkbox"/>	Parameter Name	Parameter Value
<input type="checkbox"/>	Orders created after datetime	<code>\${property.QueryLastRunDate}</code>
<input type="checkbox"/>	Orders processed before datetime	<code>\${property.CurrentDateTime}</code>

Figure 7.3

8. Troubleshooting

In case of issues or errors within Integration Flows, change the Log Level of the Integration Flows to **“Trace”**. This can be done by following these steps:

1. Navigate to the Monitor Tab of the SAP Cloud Integration tenant.
2. Proceed to the Manage Integration Content section.
3. Change the Log Level to Trace

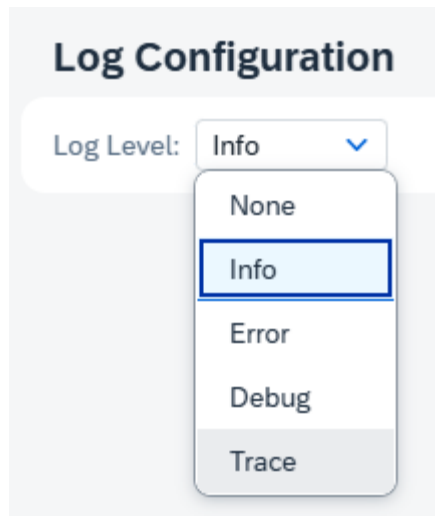


Figure 8.1 Changing the Log Level to “Trace” for an Integration Flow

The Trace Log level enables the collection of more traces that can be used to effectively understand problems.

The following sections contain issues that might be encountered when using the Shopify Adapter with possible solutions.

8.1. Unsuccessful Request

This error generally occurs when the HTTP Status Code returned from Shopify is other than “2xx success”

Example 1:

Error Message:

com.sap.shopify.exception.ShopifyUnsuccessfulRequestException: The response to the http request sent to Shopify url: <https://quickstart-xxxxxx.myshopify.com> has a response code 401 due to {"errors":{"API Invalid API key or access token (unrecognized login or wrong password)"}}

Possible solution:

Verify the validity of the provided access token.

Example 2:

Error Message:

com.sap.shopify.exception.ShopifyUnsuccessfulRequestException: The response to the http request sent to Shopify url: <https://quickstart-xxxxxx.myshopify.com> has a response code 403 due to {"errors":{"API This action requires merchant approval for read_orders scope."}}

Possible solution: Ensure the necessary access scopes are configured for the Shopify App & access token.

8.2. Base URL Exception

Error Message:

com.sap.shopify.exception.ShopifyBaseUrlException: Invalid shopify url: <https://quickstart-xxxxxx.com>

Possible solution: Verify the validity of the configured Shopify Base URL for each Integration Flow.

8.3. Missing Message Body

Error Message:

com.sap.shopify.exception.MessageBodyMissingException: Message body for resource CUSTOMER with operation CREATE_CUSTOMER

Possible solution: Ensure that for operations that modify Shopify Entities the necessary request body is present in a content modifier before the Request-Reply Integration Flow Step.

9. Sample Scenarios Explained

In this chapter, a sample business scenario will be explained consisting of the usage of the Shopify Adapter.

In this scenario, a Shopify Receiver is periodically pulls information for new orders from the Shopify Store. This is achieved by using a Timer component which is set to trigger on a schedule determined by the user.

On each execution of the Integration Flow, orders that have been created in a certain time range are pulled from Shopify. This range is marked at its two ends by two variables representing datetimes. The first one is the datetime when the last poll occurred. The second one is the current datetime.

9.1. Simple scenario using dynamic configuration and write variables for retrieving list of variables

9.1.1. Assign Timer

1. Click on the integration flow artifact name and choose Edit.

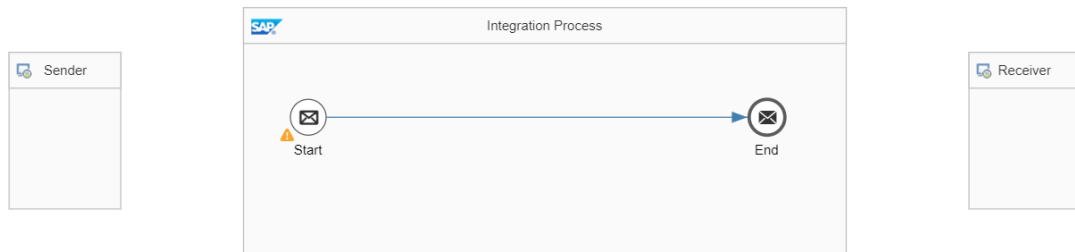


Figure 9.1 Creating a new Integration Flow

2. Select Sender and click Delete
3. Select Start Message and click Delete.

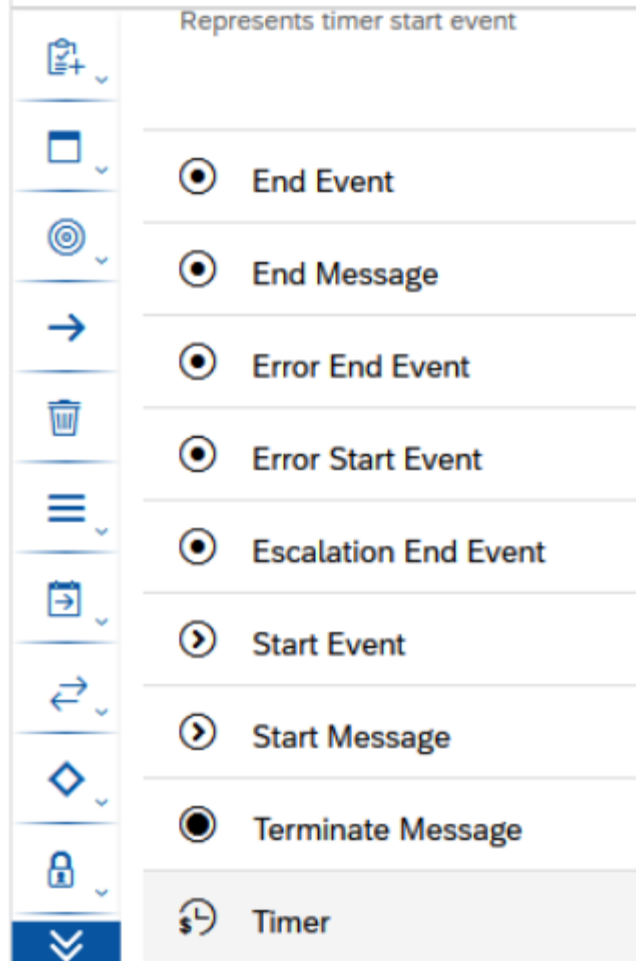


Figure 9.2 Adding a Timer to the Integration Flow

4. Click on Events and select the Timer. Connect the Timer to End Message

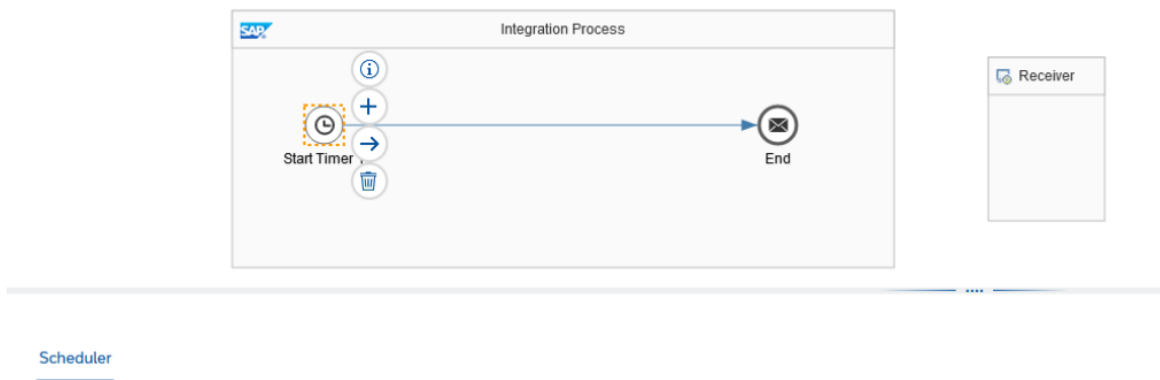


Figure 9.3

9.1.2. Add a Content Modifier and connect it to the Start Message.

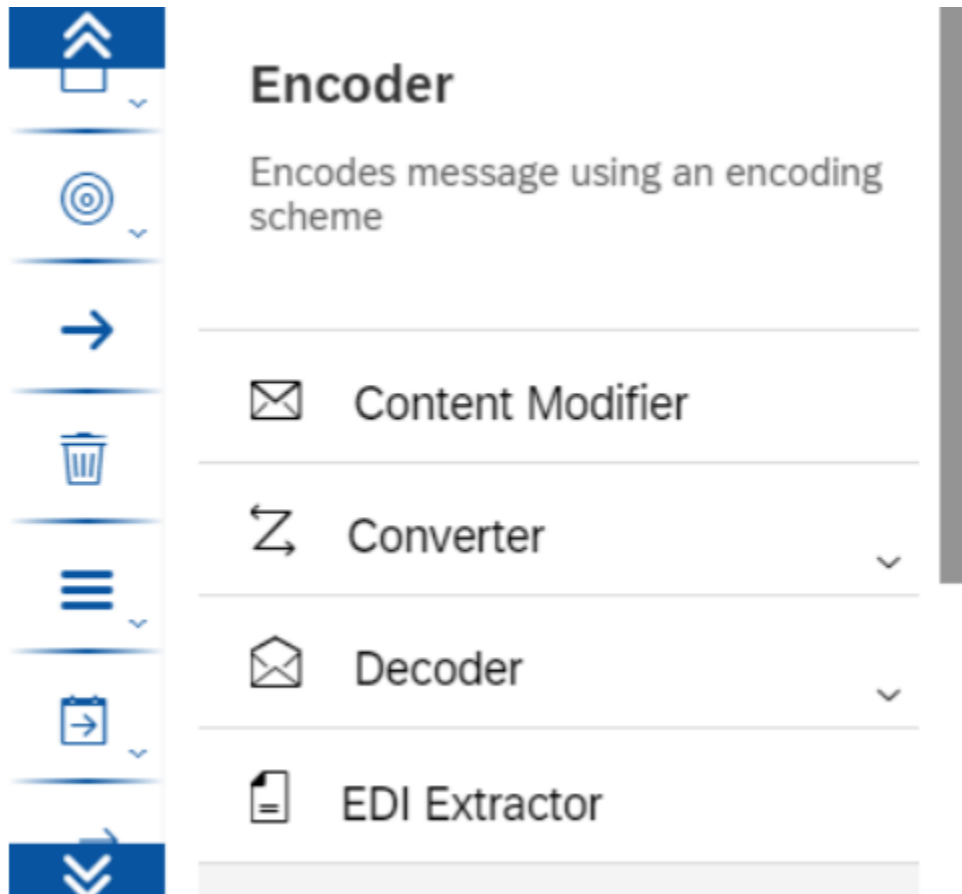


Figure 9.4

9.1.3. Add Exchange Properties

In this example the first property represents the datetime of the last time when the Integration Flow had been ran. The second property is an expression representing the current datetime.

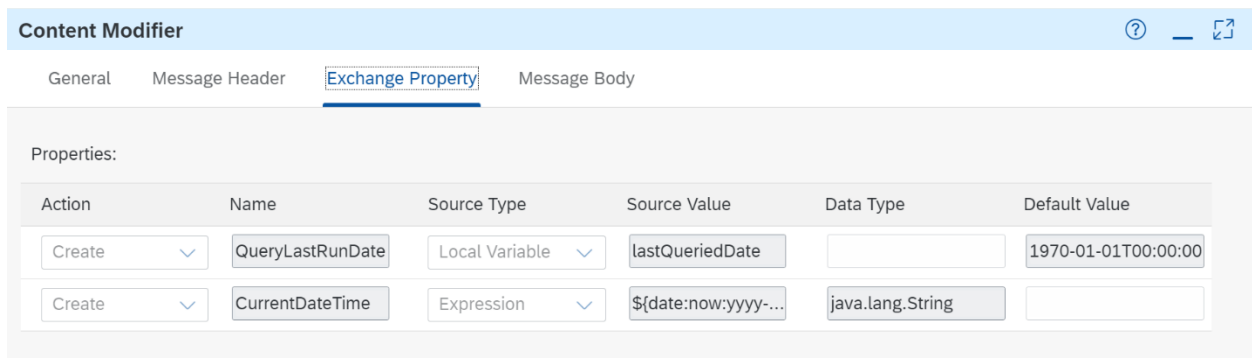


Figure 9.5

9.1.4. Configure Receiver Shopify Adapter

1. Add a Request Reply step to the integration flow after the Content Modifier

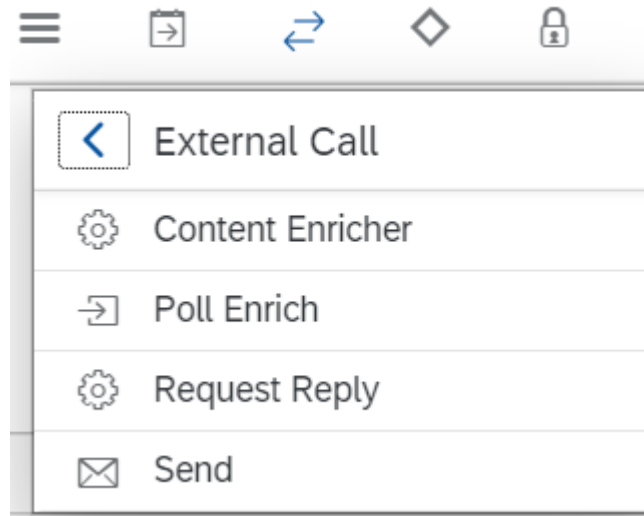


Figure 9.6

2. Select Request Reply and choose the arrow to drag the pointer to Receiver. Then, select the Shopify Adapter.
3. Choose REST

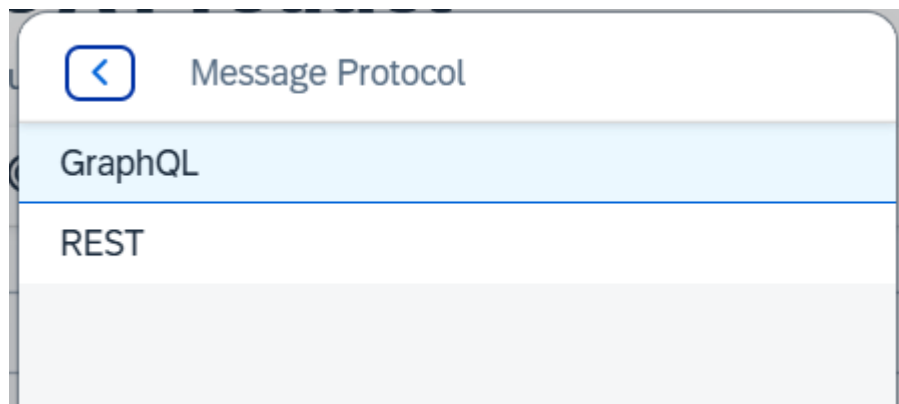


Figure 9.7 Selecting the Message Protocol in the Shopify Adapter

4. In the **“Connection”** tab
 - 4.1 Add in the Base URL of your Shopify Store
 - 4.2 Supply the needed security material by referring to the Shopify Security Material. “shopifyOAuthClientCredentials” has been created as a OAuth2 Client Credentials within SAP Integration Suite.
 - 4.3 Attach Error Details on Failure - By default, the option is enabled. This option enables the creation of attachments for request headers, response headers, and response body when the

message processing fails. Having these attachments during message processing failures can be unnecessary as it leads to persistence of attachments that doesn't help. Especially, if multiple message processing failures occur, you have attachments piled up for each failure. If you don't require the attachments for failure scenarios, disable the option. Though you disable the creation of attachments, the content of the same are added to the message processing logs.

4.4 At the Response Headers field - Provide a pipe - separated (|) value list of HTTP response headers. The received header values are then converted to message/exchange headers. If the value * is entered, all the HTTP response header values are converted to message/exchange headers.

The screenshot shows the configuration interface for the Shopify adapter, specifically the 'Connection' tab. The interface is divided into three sections: 'General', 'Connection', and 'Processing'. The 'Connection' tab is active and contains the following fields:

- Shopify Base Url:** A text input field containing the value `https://xxxxxxx.myshopify.com`.
- Authentication Type:** A dropdown menu with the selected option being `OAuth2 Client Credentials`.
- Shopify Security Material:** A text input field containing the value `shopifyOAuthClientCredentials`.
- Attach Error Details on Failure:** A checkbox that is checked.
- SHOPIFY HEADERS:** A section containing a **Response headers:** text input field with the value `*`.

Figure 9.8 Configuring the Shopify Adapter Connection

5. In the “**Processing**” tab

5.1 For operation choose “Retrieve a list of orders”

5.2 Add two query parameters:

Orders created after datetime with value `${property.QueryLastRunDate}`

Orders created before datetime with value `${property.CurrentDateTime}`

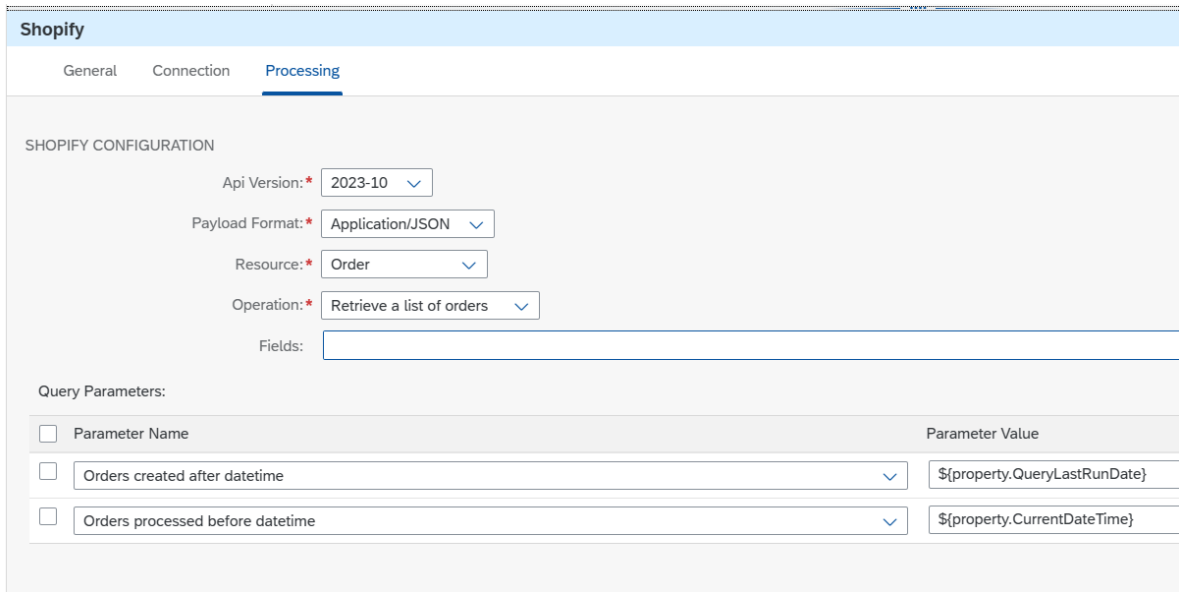


Figure 9.9 Processing Configuration for the Shopify Adapter

9.1.5. Configure Write Variables Component

1. Add “Write Variables” after the Request-Reply step

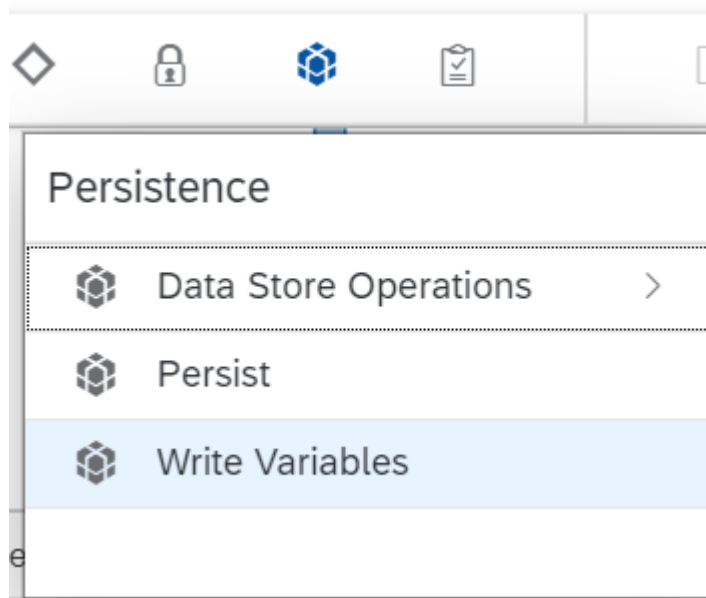


Figure 9.10 Introducing a “Write Variables” Integration Flow step

2. In Processing tab set “lastQueriedDate” to be assigned the value of “CurrentDateTime” and to be of type “Property”

Write Variables

General Processing

Variables:

Name	Type	Data Type	Value	Global Scope
lastQueriedDate	Property		CurrentDateTime	<input type="checkbox"/>

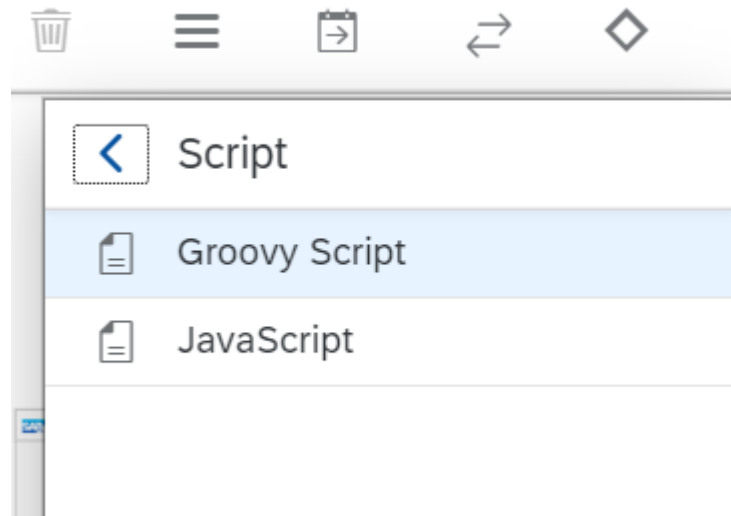
Figure 9.11 Writing a new variable

Note: Variables are persisted outside the Integration Flow. Every time the Integration Flow is triggered, the value of the variable will be overwritten.

Variables can be observed and managed through “Monitor -> Integrations -> Manage Stores -> Variables”.

9.1.6. Log the results from the request to Shopify

1. Add a script step after the “Write Variables” Component:



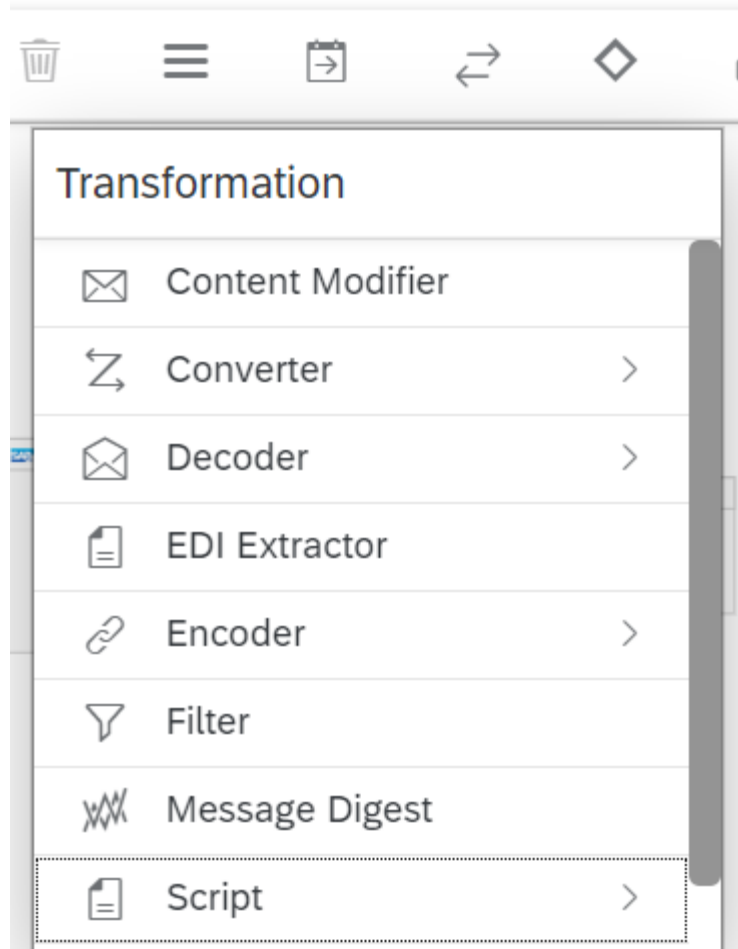


Figure 9.12 Adding a "Groovy Script" Integration Flow

2. Add a Groovy Script

- 2.1 Click on the Groovy Script Component.
- 2.2 Click on Create icon.

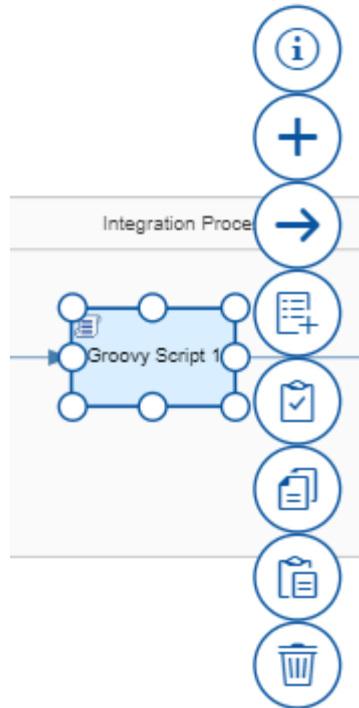


Figure 9.13 Creating a new Groovy Script

Add the following Groovy script:

```

import com.sap.gateway.ip.core.customdev.util.Message;
import java.util.HashMap;
import com.sap.it.api.ITApiFactory;
import com.sap.it.api.mapping.ValueMappingApi
import com.sap.it.api.securestore.SecureStoreService
import com.sap.it.api.securestore.UserCredential
import com.sap.it.api.securestore.exception.SecureStoreException

def Message processData(Message message)
{
    def body = message.getBody(java.lang.String) as String;
    def messageLog = messageLogFactory.getMessageLog(message);

    if (messageLog != null) {
        messageLog.addAttachmentAsString("Log current Payload:", body, "text/plain");
    }
    return message;
}

```

9.1.7. Result of the flow

1. Save the integration flow as a version and click on Deploy.
2. Check the results of the logs
 - a. Navigate to “Monitor -> Integrations -> Monitor Message Processing -> All artifacts”
 - b. Select the artifact that you have deployed and then click on “Log current Payload”

Status Properties Logs **Attachments** Artifact Details

Attachments

Name	Modified At
Log current Payload: text/plain	Sep 28, 2023, 12:07:31

Artifact Details

Figure 9.14 Observing the log from the Integration Flow Execution

- c. A list of Shopify Orders must be present in the payload.

[Overview](#) / [Monitor Message Processing](#) / Message Processing Log Attachments

Name: Test Shopify Fetch Orders with Timer Status: Completed Processing
 Last Updated at: Sep 28, 2023, 11:49:51 Log Level: Info

Log **[Log current Payload:](#)**

```

{"orders": [{"id": 5462702653743, "admin_graphql_api_id": "gid://shopify/Order/5462702653743"}]}
    
```

Figure 9.15 Shopify Orders present in the Log Attachment

10. References

10.1. Locate the Shopify Base URL Address

To find the address that can be used in the Shopify Base URL field of the Connection Tab, login into your Shopify admin. Then click on Settings.

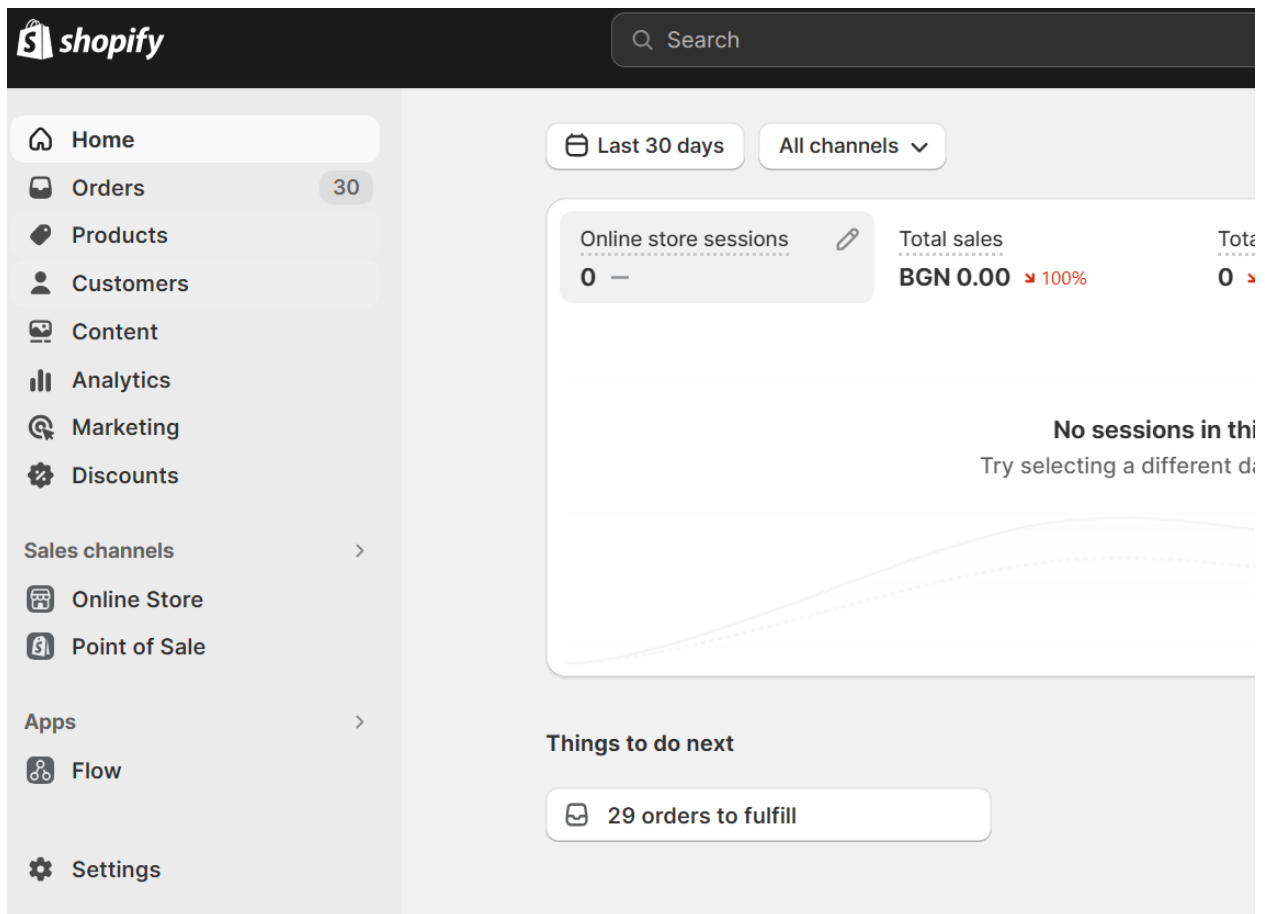


Figure 10.1 Shopify Admin Portal

The Store's Address can be found in the upper left corner and is outlined with red in the figure below

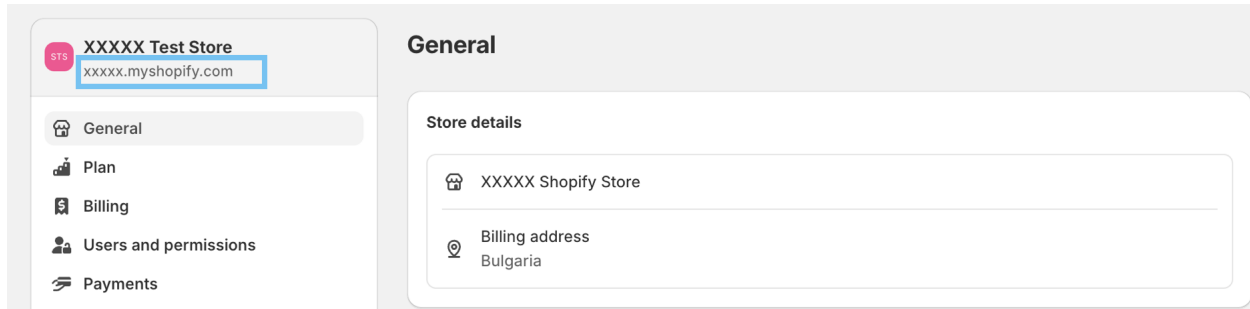


Figure 10.2 Shopify Store Address within the Shopify Admin Portal

11. API Deprecation Notice

Shopify periodically deprecates older versions of their API to streamline operations and introduce more robust and feature-rich versions. Deprecation does not mean immediate removal but indicates that the version will no longer receive updates and will ultimately be discontinued. For detailed information, you can visit [Shopify's API versioning and deprecation practices](#).

Shopify releases a new API version every 3 months at the beginning of the quarter. Version names are date-based to be meaningful and semantically unambiguous (for example, 2026-01, 2025-10, 2025-07, 2025-04).

Each stable version is supported for a minimum of 12 months. This means that there are at least nine months of overlap between two consecutive stable versions.

It is strongly recommended that you migrate to a newer version every quarter by updating the usage of the Shopify Adapter in your custom Integration Flows. However, if you are still using a stable version that is no longer supported, then Shopify falls forward and responds to your request with the same behavior as the oldest supported stable version. For example, when Shopify removes 2025-01, API requests to version 2025-01 will be served version 2025-04, because that will be the oldest supported stable version.

Part of the Shopify APIs can be deprecated if they become unnecessary, unsafe, or outdated. Parts of the APIs, that are marked as deprecated, will be removed in newer versions of the APIs. The deprecation is then retroactively applied to previous stable versions of the API.

Because each version is supported for a minimum of a year, there are always at least 9 months of overlap between versions for you to migrate to a newer version of the APIs in your Integration Flows.

Due to Shopify's deprecation practice, older API versions, which are no longer stable and officially supported by Shopify, can be removed from the Adapter.