BigCommerce Adapter Guide

For SAP Integration Suite

Version 1.1.0 – March 2026

Table of Contents

1. Introduction 4

1.1. Coding Samples 4

1.2. Internet Hyperlinks 4

2. BigCommerce Integration 4

2.1. Introduction 4

2.2. BigCommerce Adapter 5

2.3. Architectural Overview 5

3. Supported Operations 6

3.1. REST API 6

3.2. GraphQL Admin API 14

4. Authentication & Authorization 15

4.1. Creating Secure Parameter in Security Material 16

4.2 Usage of the Secure Parameters 18

5. BigCommerce Configuration 19

5.1 Create access token 19

5.2 Create a Bearer Token 22

6. Receiver Adapter Configuration 23

6.1. Connection 23

6.2. Processing Configuration for REST Admin API 24

6.3. Processing for GraphQL 30

7. Payload and Dynamic Configuration 31

7.1. Message Payloads 31

7.2. Dynamic configuration 32

8. Troubleshooting 33

8.1. Unsuccessful Request 34

8.2. Base URL Exception 35

8.3. Missing Message Body 35

9. Sample Scenarios Explained 35

9.1. Simple scenario using dynamic configuration and write variables for retrieving list of variables 36

10. References 46

10.1. Locate the BigCommerce Store Hash 46

1. Introduction

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

2.1. Introduction

BigCommerce is a scalable, open-source ecommerce platform that offers solutions for B2B, B2C, and omnichannel businesses. It provides powerful features, integrations, services, and resources to help you shape your future on your terms.

BigCommerce operates as a cloud-based, software-as-a-service (SaaS) platform. This means that merchants don’t need to install or maintain any software on their own servers. Everything is hosted and managed by BigCommerce.

For more information about BigCommerce, visit www.bigcommerce.com

2.2. BigCommerce Adapter

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

2.2.1. Features

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

- Support for multiple BigCommerce REST API operations, including:
    - Customer
    - Inventory
    - Order
    - Product
    - Promotion
- Support for GraphQL API queries and mutations
- Secure authentication for BigCommerce's APIs
- Option to automatically format the BigCommerce API response in XML and JSON formats.
- Option to add log attachments on failure

2.3. Architectural Overview

From a technical perspective, the BigCommerce 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 the BigCommerce API need to be scheduled, a Scheduler can be utilized within Integration Flows.

Figure 2.1. *showcases how the BigCommerce Adapter can be used in a simple Request-Reply scenario. Various operations can be used in the Adapter, explained in Section 3.*

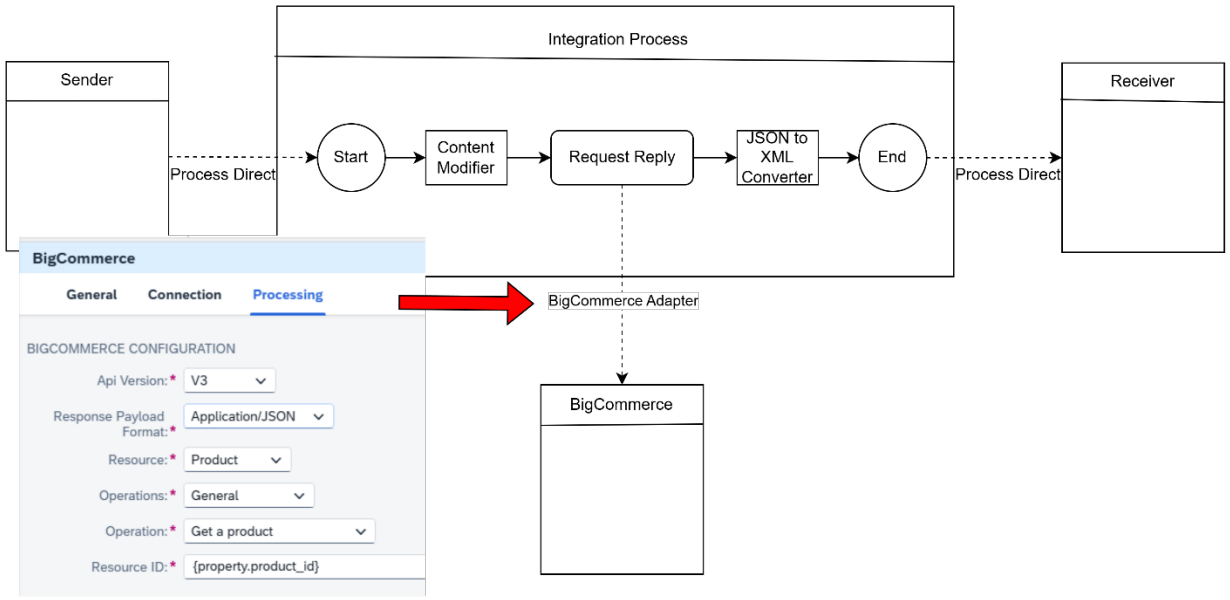

Figure 2.1 Usage of the BigCommerce Adapter for SAP Integration Suite

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

3. Supported Operations

The BigCommerce Adapter supports the following APIs:

- BigCommerce Catalog REST API
- BigCommerce Management REST API
- BigCommerce GraphQL Admin API

Brief descriptions for all are presented in the following sections.

3.1. REST API

The adapter supports BigCommerce`s REST Catalog and Management APIs and allows integration with BigCommerce using JSON format. The integration is done securely by utilizing the BigCommerce`s API access tokens for authenticating REST calls.

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

Resource	Description
Orders	An order is a customer's request to purchase one or more products from a shop. You can create, retrieve, update, delete and get a count of orders using the Order resource
Order Shipping Addresses	An order shipping address is the address associated with an order.
Order Shipments	An order shipment is a partial or full fulfillment of all ordered products.
Order Taxes	An order tax contains the tax information for the order.
Order Products	The Order Products resource lets you retrieve individual line-items within an order or list all products associated with a given order.
Order Metafields	Metafields are a flexible way to attach additional information to a BigCommerce Order
Order Transactions	Order Transaction is the payment information associated with an order
Payment Actions	The Payment Actions resource lets you manage refunds like retrieving single or multiple refunds, generation of refund quotes, and creation of refunds.
Product	The Product resource lets you update and create products in a merchant's store
Product Custom Fields	Custom fields allow you to specify additional information that will appear on the products page.
Product Metafields	Metafields are a flexible way to attach additional information to a BigCommerce Product
Product Channel Assignments	Channel assignments define where a product is visible and available for purchase—across different sales channels
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.
Customer Attributes	Customer attributes define the name of a name-value pair, and the type of information stored (for example, "name": "Shoe size", "type": "number").

Resource	Description
Customer Metafields	Metafields are a flexible way to attach additional information to a BigCommerce Customer
Inventory Adjustments	Inventory Adjustments are used to modify inventory stock.
Inventory Items	Inventory Items are all product variants whose inventories are tracked by variant
Promotions	Promotions are a way to give discounts based on whether the shopper has met specific criteria such as ordering a certain amount, purchasing certain brands, or being a repeat customer.

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

Resource	Operation	Description
Orders	Archive an order	Archives an order by ID
	Create an order	Creates an order. A payload must be added in a Content Modifier with JSON content specifying the new order.
	Delete all orders	Archives all orders in the store.
	Get a count of orders	Gets an array of orders in the store organized by order status.
	Get all orders	Get a list of orders using the filter query.
	Get an order	Retrieve an order by specifying the ID.
	Update an order	Updates an order by ID. A payload must be added in a Content Modifier with JSON content specifying the update to the order.
Order Shipping Addresses	Get a Shipping Address	Gets a shipping address associated with an order by specifying the ID.
	Get order shipping addresses	Get all shipping addresses on an order.
	Update a shipping address	Update a shipping address associated with an order.
Order Shipments	Get order shipments	Gets a list of all shipments on an order
	Create order shipment	Creates an order shipment. A payload must be added in a Content Modifier with JSON content specifying the order shipment to create
	Delete order shipments	Deletes all shipments associated with an order.

Resource	Operation	Description
Order Taxes	Get order taxes	Get all taxes associated with an order.
Order Products	Get Order Product	Retrieves a specific product within an order by specifying both the order ID and the product ID.
	List Order Product	Retrieves all products associated with a specified order by providing the order ID.
Order Metafields	Create metafields	Creates an order metafield. A payload must be added in a Content Modifier with JSON content specifying the order metafield to create
	Delete a metafield	Deletes a metafield by specifying the ID.
	Get a metafield	Gets a metafield by specifying the ID.
	Get order metafields	Get all metafields associated with an order.
	Update a metafield	Updates a metafield object. A payload must be added in a Content Modifier with JSON content specifying the order metafield update
Order Transactions	Get order transactions	Get all transactions associated with an order.
Payment Actions	Get Refund	Retrieves details of a single refund by specifying its refund ID.
	Get Refunds for Order	Retrieves all refunds associated with a specific order by providing the order ID.

Resource	Operation	Description
	Get All Refunds	Retrieves a list of all refunds across the store.
	Create a Refund	Creates a refund for a specified order. A payload must be added in a Content Modifier with JSON content specifying the refund amount, items, and any tax/ shipping.
	Create a Refund Quote	Generates an estimated refund quote for an order, by supplying a JSON payload with the items to quote.
Products	Create a product	Creates a product. A payload must be added in a Content Modifier with JSON content specifying the product to create
	Delete a product	Deletes a product by specifying the ID.
	Delete products (Batch)	Deletes all products matching the query filter sent.
	Get a product	Get a single product by specifying the ID.
	Get all products	Returns a list of all products matching the query filter.
	Update a product	Updates a single product by specifying the ID. A payload must be added in a Content Modifier with JSON content specifying the product update
	Update products (Batch)	Update products in batches. Batches are limited to 10 products.

Resource	Operation	Description
Product Custom Fields	Create a product custom field	Creates a custom field for a particular product. A payload must be added in a Content Modifier with JSON content specifying the custom field to create.
	Delete a product custom field	Delets a product custom field by specifying the ID.
	Get a product custom field	Returns a single product custom field by specifying the ID.
	Get a product's custom fields	Returns a list of a product's custom fields.
	Update a product custom field	Updates a specific product custom field. A payload must be added in a Content Modifier with JSON content specifying the custom field to update.
Product Metafields	Create a product metafield	Creates a product metafield. A payload must be added in a Content Modifier with JSON content specifying the product metafield to create.
	Delete a product metafield	Deletes a product metafield by specifying the ID.
	Get a product metafield	Returns a single product metafield by specifying the ID.
	Get product metafields	Returns a list of product metafields.
	Update a product metafields	Update a specific product metafield. A payload must be added in a Content Modifier with JSON content specifying the product metafield update
Product Channel Assignments	Create Channel Assignment	Assigns one or more products to specific channels

Resource	Operation	Description
	Get Channel Assignments	Retrieves current channel assignments for products
	Delete Channel Assignment	Removes a product from specific channels
Customer	Get Customers	Retrieves a list of customers in the store, optionally filtered by query parameters.
	Get Customer Addresses	Retrieves all saved addresses for a given customer by specifying the customer ID.
	Update Customers	Updates one or more existing customers. A payload must be added in a Content Modifier with JSON content specifying the changes.
Customer Attributes	Get Customer Attributes	Retrieves all custom attribute definitions for customers in the store.
	Create Customer Attributes	Creates a new custom attribute definition for customers. A payload must be added in a Content Modifier with the attribute JSON.
	Update Customer Attributes	Updates an existing customer attribute definition by specifying the attribute ID and providing a JSON payload with the updates.
	Delete Customer Attributes	Deletes a customer attribute definition by specifying its attribute ID.
Customer Metafields	Get Customer Metafields	Retrieves all metafields associated with a specific customer by providing the customer ID.

Resource	Operation	Description
	Get a Customer Metafield	Retrieves a single metafield for a customer by specifying both the customer ID and the metafield ID.
	Create Customer Metafields	Creates a new metafield for a customer. A payload must be added in a Content Modifier with JSON content specifying the metafield.
	Update Customer Metafields	Updates an existing customer metafield by specifying IDs and providing a JSON payload with the updated metafield properties.
	Delete Customer Metafields	Deletes a customer metafield by specifying both the customer ID and the metafield ID.
Inventory Adjustments	Update absolute adjustments	Override the existing inventory levels for an inventory item at a location.
	Update relative adjustments	Add or subtract inventory for an item at a location.
Inventory Items	Retrieve a list of inventory items all locations	Return a list of inventory levels and settings for all items in all locations matching a query filter.
	Retrieve a list of inventory items of a location	Return a list of inventory levels and settings for all items in a location.
	Update inventory settings	Update inventory settings for items at a location.
Promotions Bulk	Delete multiple promotions	Deletes multiple promotions matching the query filter. Batches are limited to 50 items.

Resource	Operation	Description
	Get all promotions	Returns a list of all promotions matching a query filter.
Promotions Single	Create a promotion	Creates a single promotion. A payload must be added in a Content Modifier with JSON content specifying the promotion to create
	Delete a promotion	Deletes a promotion by specifying the ID.
	Get a promotion	Get a single promotion by specifying the ID.
	Update a promotion	Update a specific promotion. A payload must be added in a Content Modifier with JSON content specifying the promotion update.

3.2. GraphQL Admin API

The adapter supports BigCommerce`s GraphQL Admin API and allows integration with BigCommerce by leveraging GraphQL Admin queries. The integration is done securely by utilizing the BigCommerce access and bearer token for Authorization. By defining queries, all resources included in GraphQL Admin API can be accessed through the BigCommerce Adapter.

4. Authentication & Authorization

All REST and GraphQL Admin API endpoints utilize the X-Auth-Token header for authentication with BigCommerce servers. This header employs access tokens to validate requests. Within the BigCommerce Adapter, the access token is used as an alias for the value of the X-Auth-Token header in the request that requires authentication. Additionally, when sending requests to the GraphQL Admin API, a bearer token must be provided for authorization.

To create both tokens refer to the official BigCommerce Documentation: Authentication | BigCommerce Dev Center API Accounts Tokens | BigCommerce Dev Center

The X-Auth-Token is generated with specific access scope to keep the platform secure. When creating the token, we need to provide permission for the scopes we need.

The bearer token can be generated by using the storefront/api-token REST API endpoint. There we provide the channel id and when this bearer token will expire (timestamp format in seconds, which should be greater than the current time).

The BigCommerce Adapter utilizes the security mechanism of SAP Integration Suite by storing sensitive configuration as Security Materials. These security artifacts are then referenced within the BigCommerce 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”.

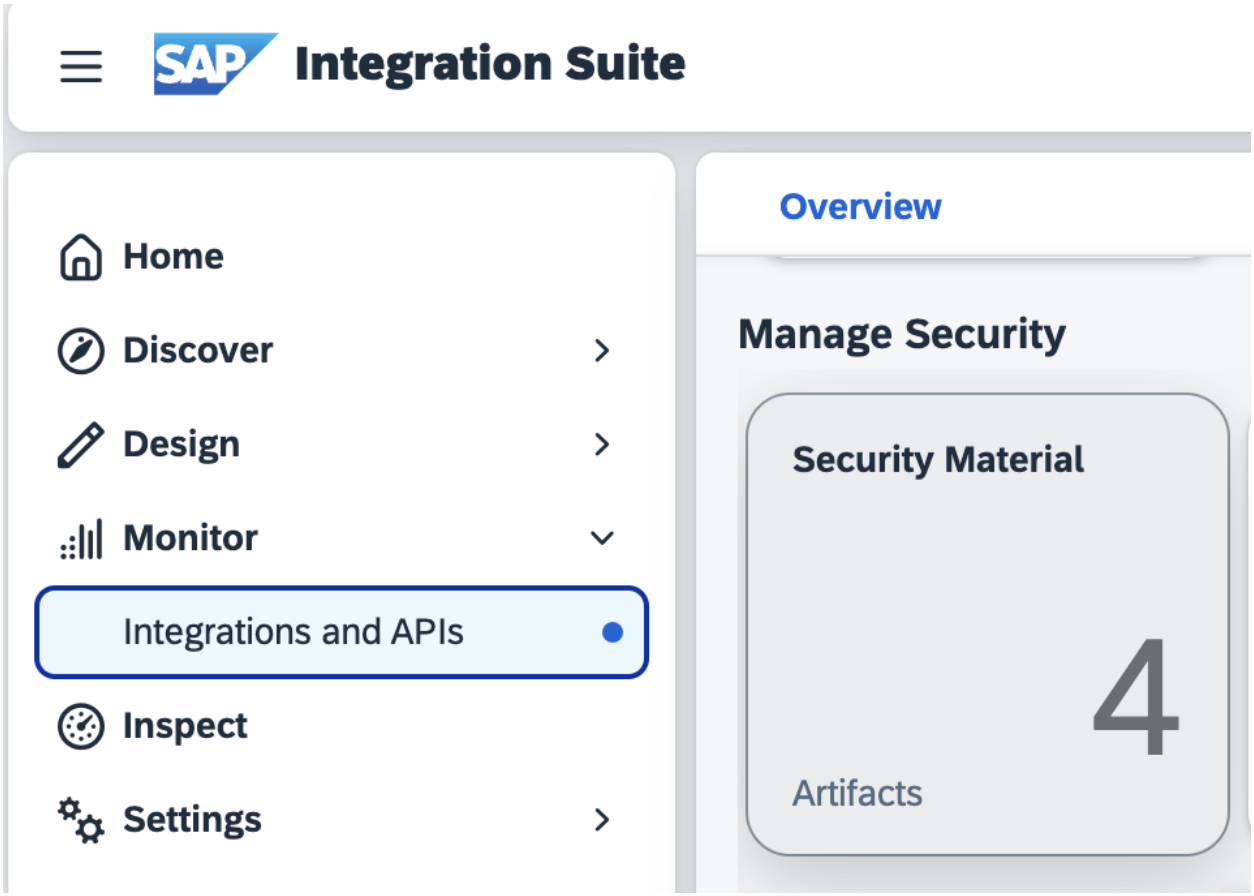

Figure 4.1 Navigating to SAP Integration Suite Monitoring

4.1.2. Create Secure Parameters for the BigCommerce APIs

- Go to Create > Secure Parameter.

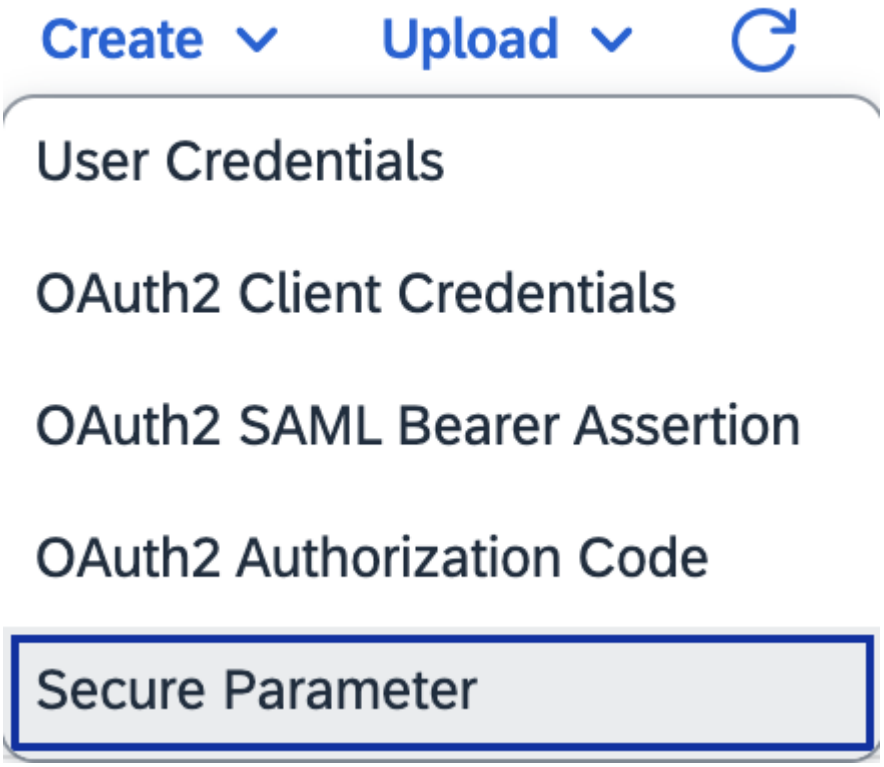

Figure 4.2 Creation of a new Secure Parameter

- Enter the Secure Parameter details

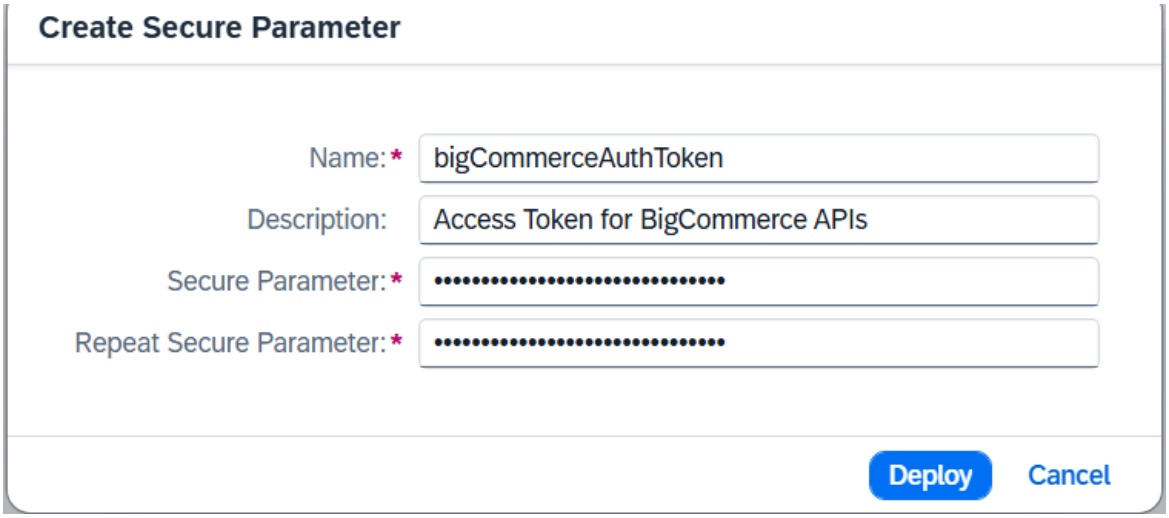

Figure 4.3.1 Creation of a new Secure Parameter Access Token

- Repeat steps 1 and 2 for the bearer token.

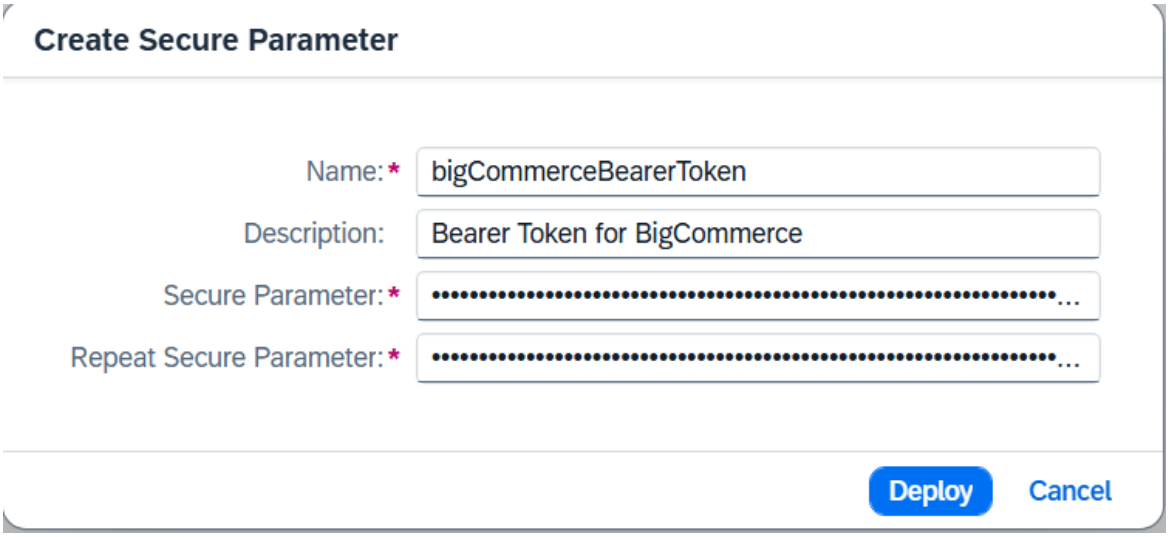

Figure 4.3.2 Creation of a new Secure Parameter Bearer Token

4.2 Usage of the Secure Parameters

The Security artifacts should be used in the Connection tab of the Adapter as shown in Figure 4.4. For the REST APIs we need only the auth token, for the GraphQL APIs we need a bearer token, as well. Detailed explanation of configuring the connection properties can be found in Section 7.1.

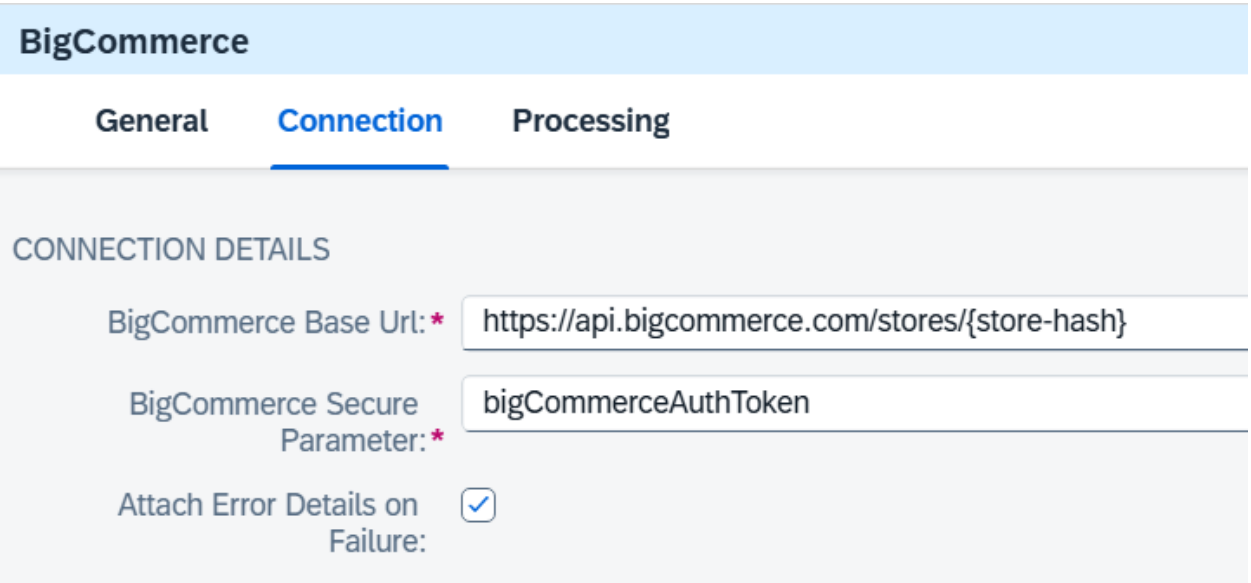

Figure 4.4 Using the Secure Parameter in Rest Configuration

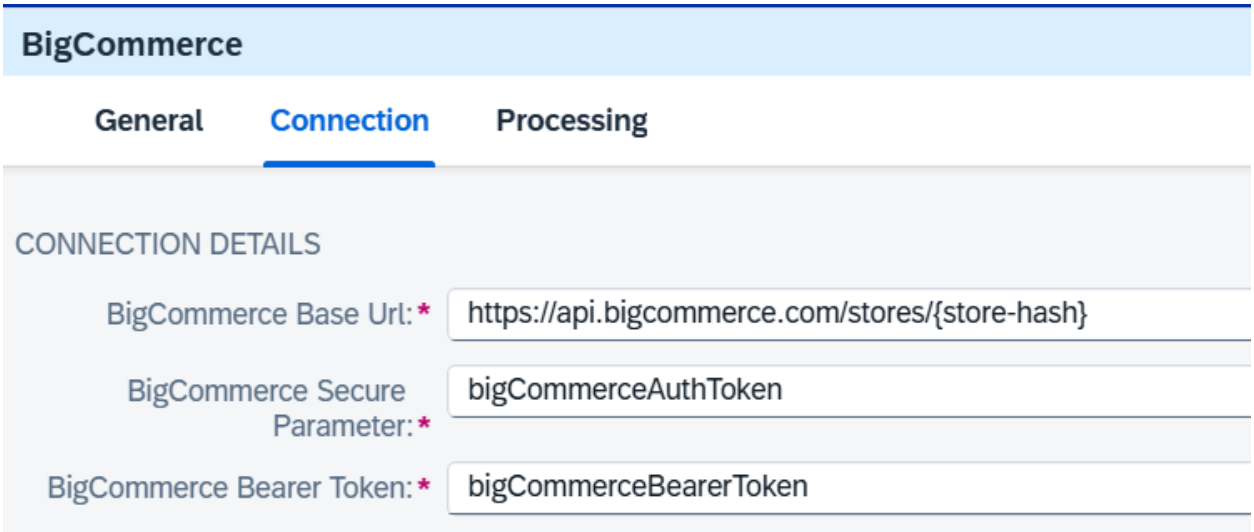

Figure 4.5 Using the Secure Parameters in GraphQL Configuration

5. BigCommerce Configuration

In BigCommerce, the authentication process for accessing API endpoints involves two types of tokens: the '**X-Auth-Token**' and **bearer token**. The **X-Auth-Token** is primarily used for authenticating requests to the BigCommerce REST and GraphQL Admin APIs. It acts as a key that grants access to the API services, verifying that the request comes from a trusted source. This token is sent in the HTTP headers, specifically in the X-Auth-Token header, with each API request. The token is generated and provided by BigCommerce, often associated with a specific API account or application. The bearer token is used for authorization, specifically when accessing the GraphQL Admin API. It further secures the request by ensuring that the entity making the request has the necessary permissions. The bearer token is included in the Authorization header of the HTTP request, formatted as Bearer <token_value>.

The following steps can be taken in order to generate both access and bearer tokens.

5.1 Create access token

- Login in the BigCommerce Control Panel
- Navigate to Settings -> API -> Store-level API accounts

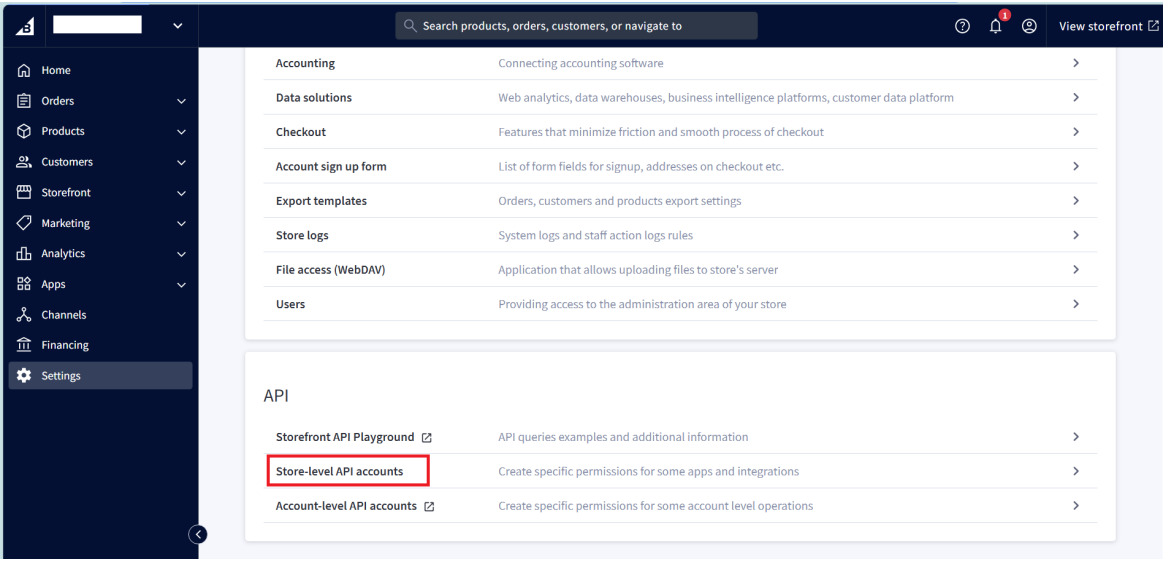

Figure 5.1 BigCommerce Settings View

- Then Click Create API account

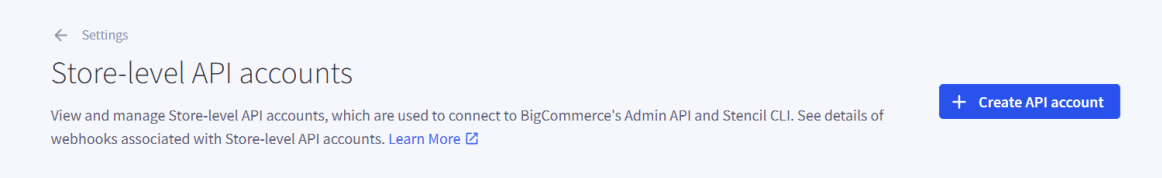

Figure 5.2 Creation of API account

- Provide Name and OAuth scopes, select the necessary permissions for Read/Modify of the needed Entities – more detailed configuration is available in the guide for integration content (in API path after */store* you can see the hash for your store, that hash will later be provided in the Adapter)

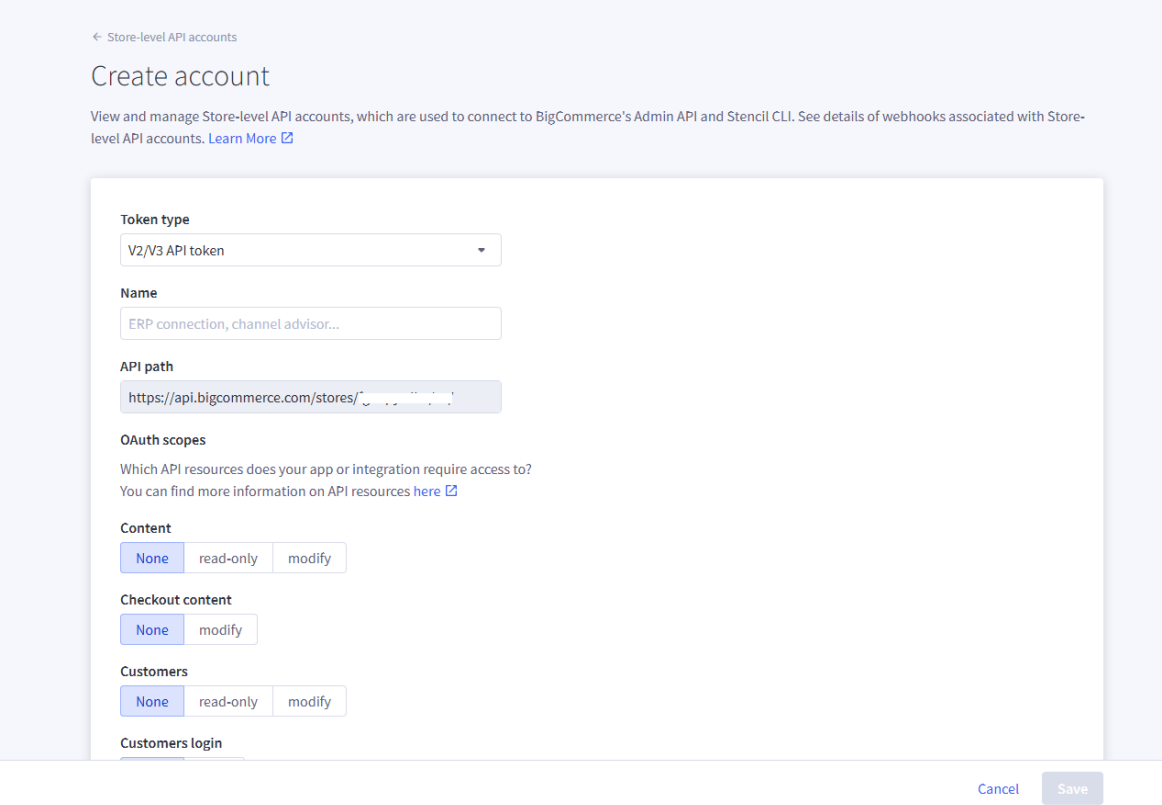

Figure 5.3 Creation of a New API account

- Click Save
- A pop up with credential information appears

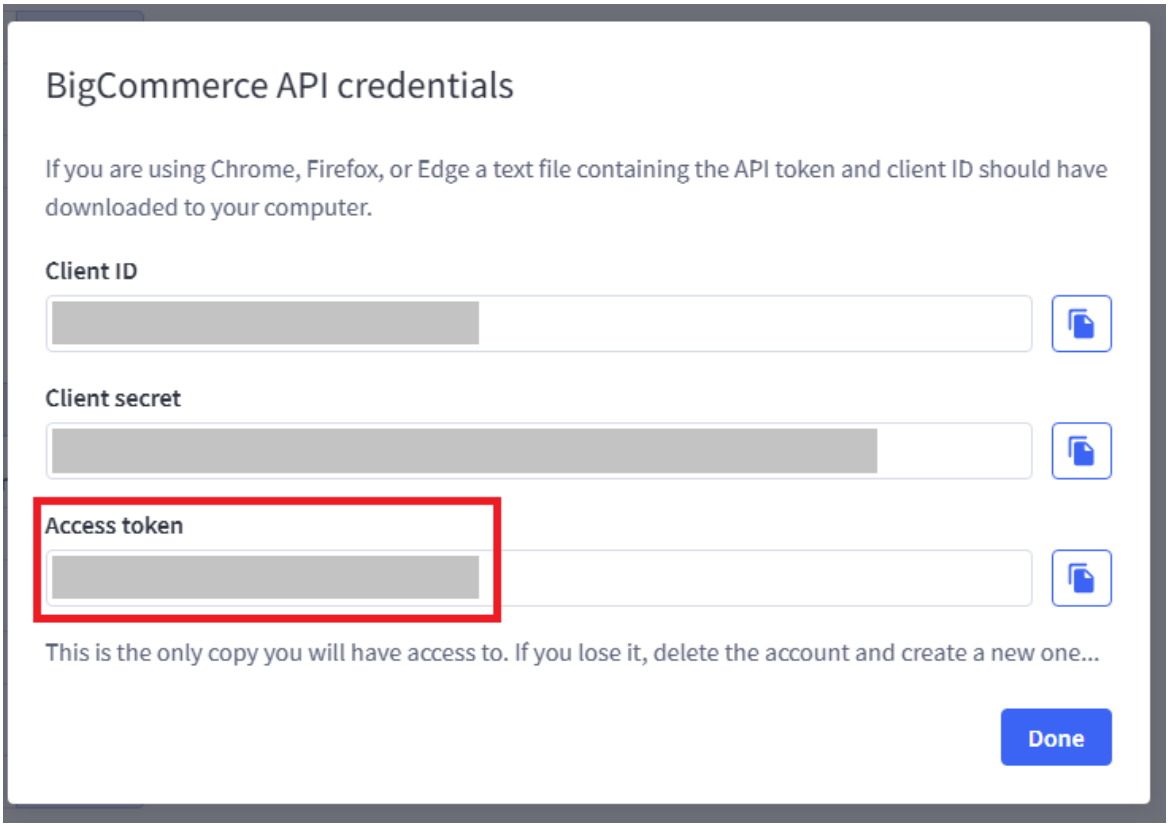

Figure 5.4 BigCommerce API Credentials

- Copy the Access Token and save it for later, make sure to save it as this is the only copy you will have access to, otherwise a new one needs to be created!

5.2 Create a Bearer Token

After an access token is created, we can proceed with the bearer token.

You can use the provided curl request command to create the bearer token.

- Open a terminal application
- Paste the following curl request

```
curl -X POST https://api.bigcommerce.com/stores/{{STORE_HASH}}/v3/storefront/api-token -H 'x-auth-token: {{ACCESS_TOKEN}}' -H 'accept: application/json' -H 'content-type: application/json' -d '{ "channel_ids": [ {{REQUIRED_CHANNEL_IDS}} ], "expires_at": {{EXPIRATION_TIMESTAMP_IN_SECONDS}} }'
```

- Replace the **{{STORE_HASH}}** with the hash of your store.
    - The hash can be retrieved, as per [Section 5.1,](#create-access-token) when creating the access token.

- Populate the **{{ACCESS_TOKEN}** with the saved access token
- Replace the **{{EXPIRATION_TIMESTAMP_IN_SECONDS}}** with a timestamp in seconds. The timestamp cannot be greater than 2147483647 (18 Jan 2038).
- Replace the **{{REQUIRED_CHANNEL_IDS}}** with the required storefront channels in a comma-separated list (e.g. 1, 2, 3). The default channel is **1**.
    - To find more about channels, visit the [BigCommerce Documentation…](https://developer.bigcommerce.com/docs/storefront/multi-storefront#channels)
- Send the request
- Copy the token from the response body

For more information on the GraphQL token, check the BigCommerce documentation

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 BigCommerce APIs

- BigCommerce Base Url – the Base URL of the BigCommerce Store including the store hash. It can be dynamically configured using the **${property.name}** or **${header.name}** notation.
- BigCommerce Secure Parameter - Specifies the name of the Secure Parameter artifact that contains the access token needed to connect to BigCommerce. It can be dynamically configured using the **${property.name}** or **${header.name}** notation.
- 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.

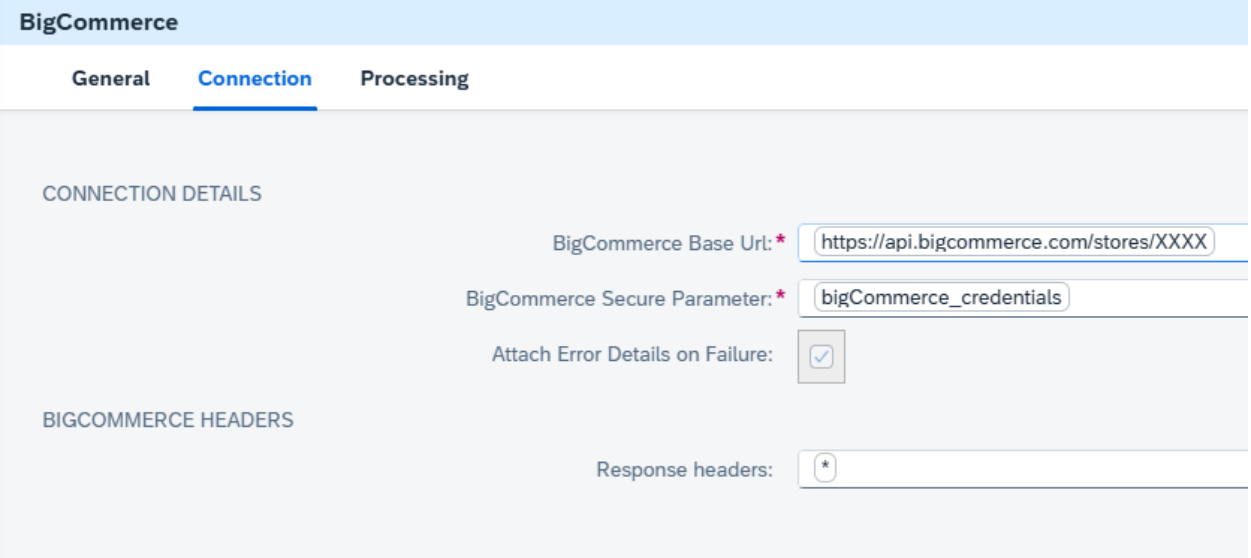

Figure 6.1 Connection Configuration for the BigCommerce Receiver Adapter

6.2. Processing Configuration for REST Admin API

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

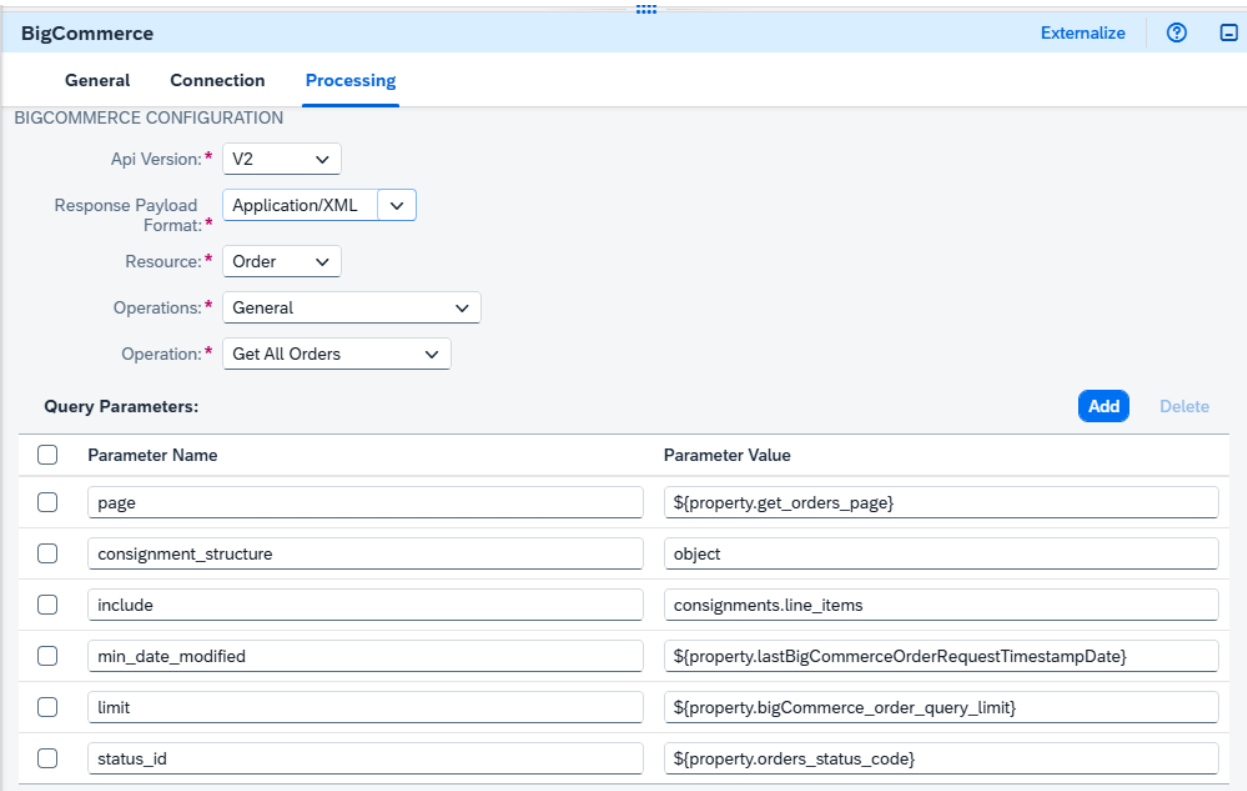

Figure 6.2 Processing Configuration for the BigCommerce REST Receiver Adapter

Mandatory fields

- **API Version -** select the BigCommerce API version to be used from the API Version dropdown menu.
- **Resource Payload Format –** select the payload format – JSON or XML for the response
- **Resource** - specify the type of resource. Resources refer to the fundamental data structures or objects that represent different aspects of an online store.
- **Operations –** specify the group of operations for the selected resource.
- **Operation –** specify an operation for the currently selected group of operations. Operation is an action which can be performed over a certain resource. It can modify or read the resource.

Mandatory Operation Parameter Fields

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

Resource	Operation	Mandatory Parameters
Orders	Archive an order	Resource ID
	Create an order	
	Delete all orders	
	Get a count of orders	
	Get all orders	
	Get an order	Resource ID
	Update an order	Resource ID
Order Shipping Addresses	Get a shipping address	Resource ID
		Order ID
	Get order shipping addresses	Order ID
	Update a shipping address	Resource ID
		Order ID
Order Shipments	Get order shipments	Order ID
	Create order shipment	Order ID
	Delete order shipments	Order ID
Order Taxes	Get order taxes	Order ID
Order Products	Get Order Product	Order ID Product ID
	List Order Product	Order ID

Resource	Operation	Mandatory Parameters
Order Metafields	Create metafields	Order ID
	Delete a metafield	Resource ID
		Order ID
	Get a metafield	Resource ID
		Order ID
	Get order metafields	Order ID
	Update a metafield	Resource ID
		Order ID
Order Transactions	Get order transactions	Order ID
Payment Actions	Get Refund	Refund ID
	Get Refunds for Order	Order ID
	Get All Refunds	
	Create a Refund	Order ID
	Create a Refund Quote	Order ID
Products	Create a product	
	Delete a product	Resource ID
	Delete products (Batch)	
	Get a product	Resource ID
	Get all products	

Resource	Operation	Mandatory Parameters
	Update a product	Resource ID
	Update products (Batch)	Resource ID
Product custom fields	Create a product custom field	Product ID
	Delete a product custom field	Product ID
		Product Custom Field ID
	Get a product custom field	Product ID
		Product Custom Field ID
	Get a product's custom fields	Product ID
	Update a product custom field	Product ID
		Product Custom Field ID
Product Metafields	Create a product metafield	Product ID
	Delete a product metafield	Product ID
		Product Metafield ID
	Get a product metafield	Product ID
		Product Metafield ID
	Get product metafields	Product ID
	Update a product metafield	Product ID
		Product Metafield ID

Resource	Operation	Mandatory Parameters
Product Channel Assignments	Delete Products Channel Assignments	At least one of the following query parameters must be present • product_id:in • channel_id:in For more information, refer to the BigCommerce documentation for "**Channel assignments**"
Customer	Get Customers	
	Get Customer Addresses	
	Update Customers	
Customer Attributes	Get Customer Attributes	
	Create Customer Attributes	
	Update Customer Attributes	
	Delete Customer Attributes	Resources IDs
Customer Metafields	Get Customer Metafields	Customer ID
	Get a Customer Metafield	Customer ID Resource ID
	Create Customer Metafields	Customer ID
	Update Customer Metafields	Customer ID Resource ID
	Delete Customer Metafields	Customer ID Resource ID

Resource	Operation	Mandatory Parameters
Inventory Adjustments	Update absolute adjustments	
	Update relative adjustments	
Inventory Items	Retrieve a list of inventory items all locations	
	Retrieve a list of inventory items of a location	Resource ID
	Update inventory settings	Resource ID
Promotions Bulk	Delete multiple promotions	Resource IDs
	Get all promotions	
Promotions Single	Create a promotion	
	Delete a promotion	Resource ID
	Get a promotion	Resource ID
	Update a promotion	Resource ID

6.3. Processing for GraphQL

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

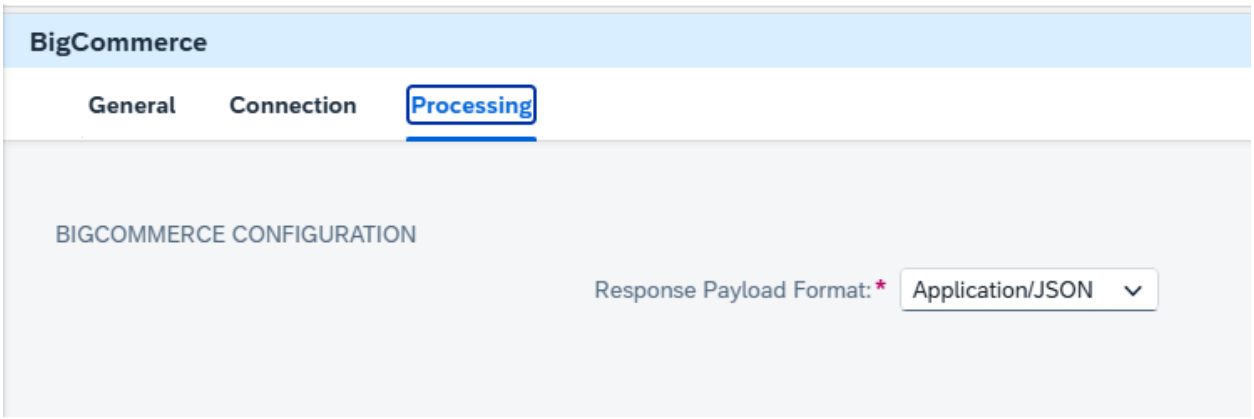

Fig 6.3 Processing Configuration for the BigCommerce GraphQL Receiver Adapter

7. Payload and Dynamic Configuration

7.1. Message Payloads

A content modifier is used to attach a payload to be sent to BigCommerce API's. Payloads must be in JSON format.

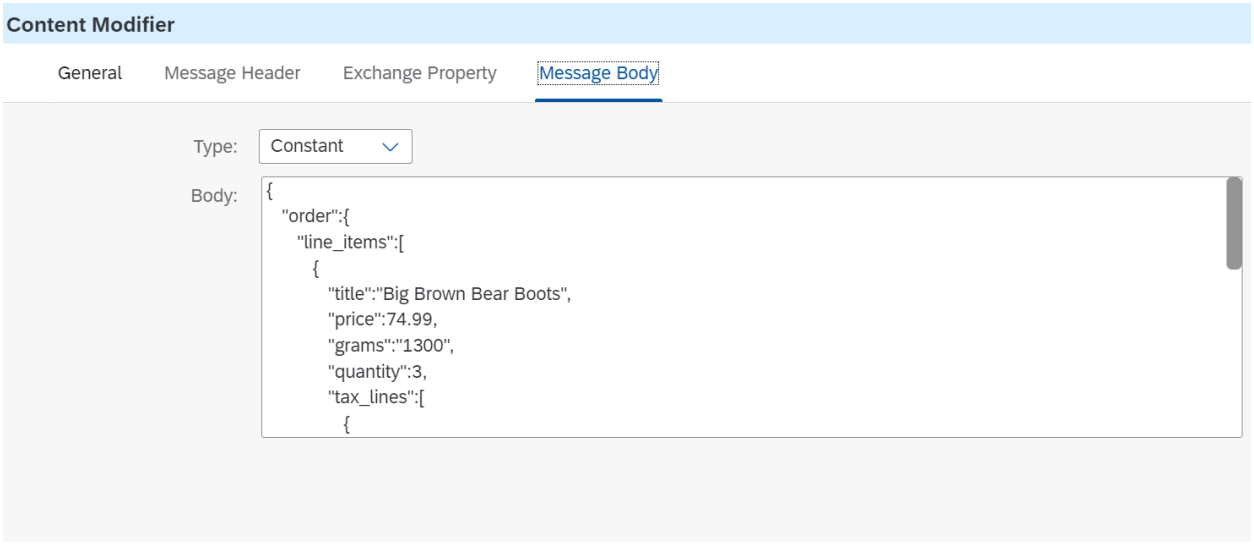

Figure 7.1.1 Example of a REST API payload in JSON format

BigCommerce 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": "..."
}
```

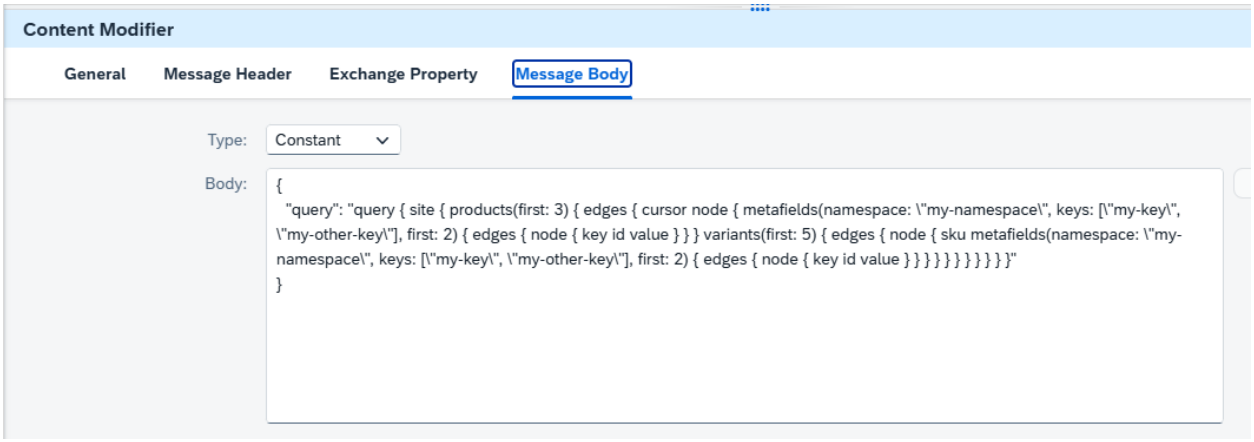

Fig 7.1.2 Example of a GraphQL API payload

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" : "…"

}
```

7.2. Dynamic configuration

This section illustrates the use of dynamic values in the Adapter configuration. It depicts a simple BigCommerce “Get all 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 BigCommerce 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 lastQueredDate source value. The second property is of type expression. Expressions have the following structure: ${...}

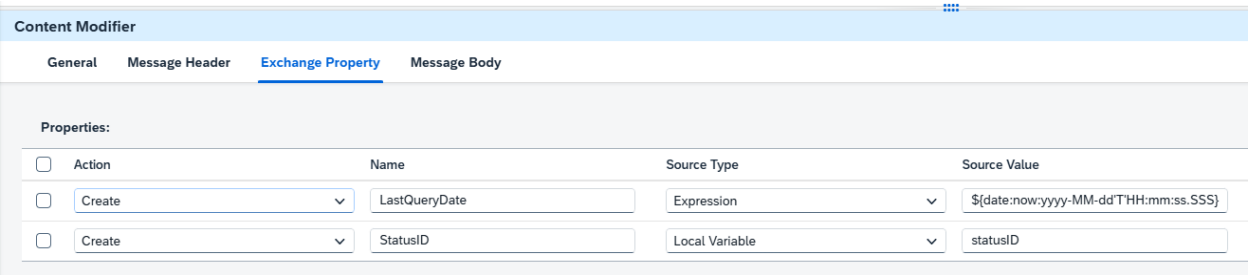

Figure 7.2 Example of a Content Modifier creating exchange properties

The BigCommerce 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 lastQueryDate property is referred to with the syntax ${property. lastQueryDate}. The BigCommerce Adapter automatically replaces this reference with the datetime of the last run of the query. Also, StatusID is referred to by using ${property.StatusID}.

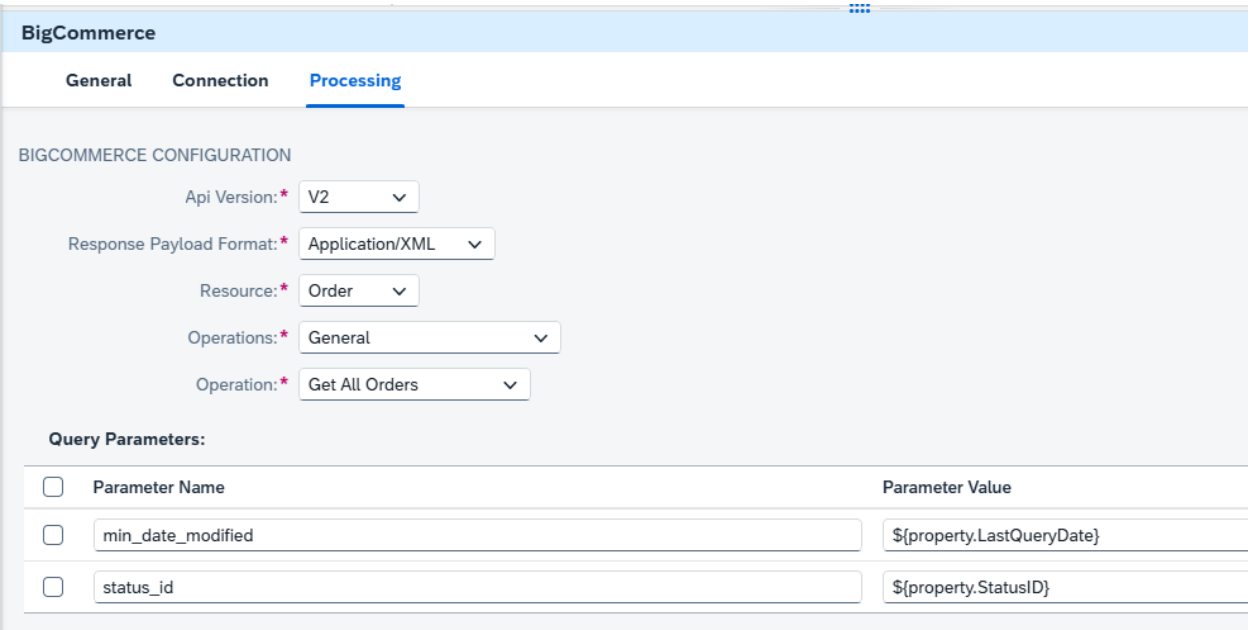

Figure 7.3 Example of query parameters using the predefined exchange properties

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:

- Navigate to the Monitor Tab of the SAP Cloud Integration tenant.
- Proceed to the Manage Integration Content section.
- Change the Log Level from Info to Trace

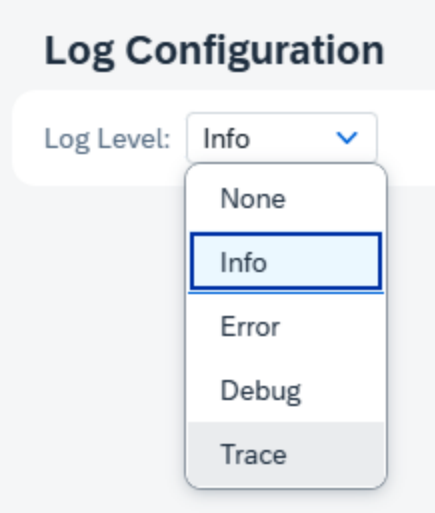

Figure 8.1 Changing the Log Level to "Trace" for an Integration Flow

The Trace Log level enables the collection of more traces that can be used to effectively understand problems. Please note that the Trace level is temporal and will be returned to Info in 10 minutes, then you need to enable it again.

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

8.1. Unsuccessful Request

This error generally occurs when the HTTP Status Code returned from BigCommerce is other than "2xx success"

Example 1:

Error Message:

com.sap.big.commerce.exceptions.base.BigCommerceUnsuccessfulRequestException: The response to the http request sent to Big Commerce url: https://api.bigcommerce.com/stores/XXXX/v2/orders has a response code 401 due to {"status":401,"title":"Unauthorized","type":"https://developer.bigcommerce.com/api-docs/getting-started/api-status-codes","errors":{}}

Possible solution:

Verify the validity of the provided access token.

Example 2:

Error Message:

com.sap.big.commerce.exceptions.base.BigCommerceUnsuccessfulRequestException: The response to the http request sent to Big Commerce url: https://api.bigcommerce.com/stores/XXXX/v2/orders has a response code 403 due to {"status":403,"title":"You don't have a required scope to access the endpoint","type":"https://developer.bigcommerce.com/api-docs/getting-started/api-status-codes","errors":{}}

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

8.2. Base URL Exception

Error Message:

com.sap.big.commerce.exceptions.base.BigCommerceBaseUrlException: Invalid BigCommerce Url provided: https://api.bigcommerce.com/stores/

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

8.3. Missing Message Body

Error Message:

com.sap.big.commerce.exceptions.base.MessageBodyMissingException: Message body is missing

Possible solution:
Ensure that for operations that modify BigCommerce 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 BigCommerce Adapter.

In this scenario, a BigCommerce Receiver periodically pulls information for new orders from the BigCommerce Store. This is achieved by using a Timer component which is set to trigger based 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 BigCommerce. 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

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

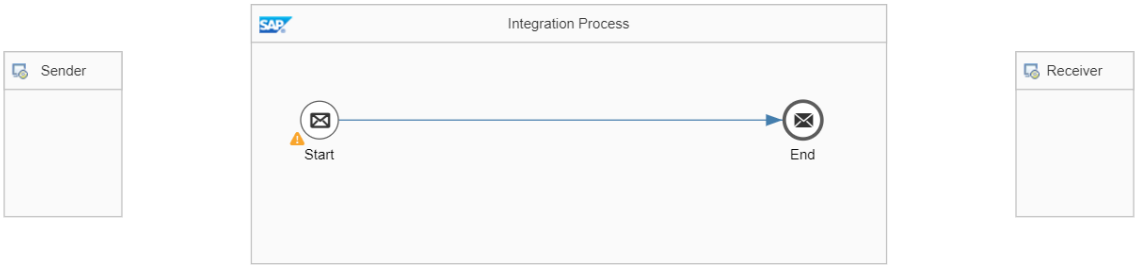

Figure 12.1 Creating a new Integration Flow

- Select Sender and click Delete
- Select Start Message and click Delete.

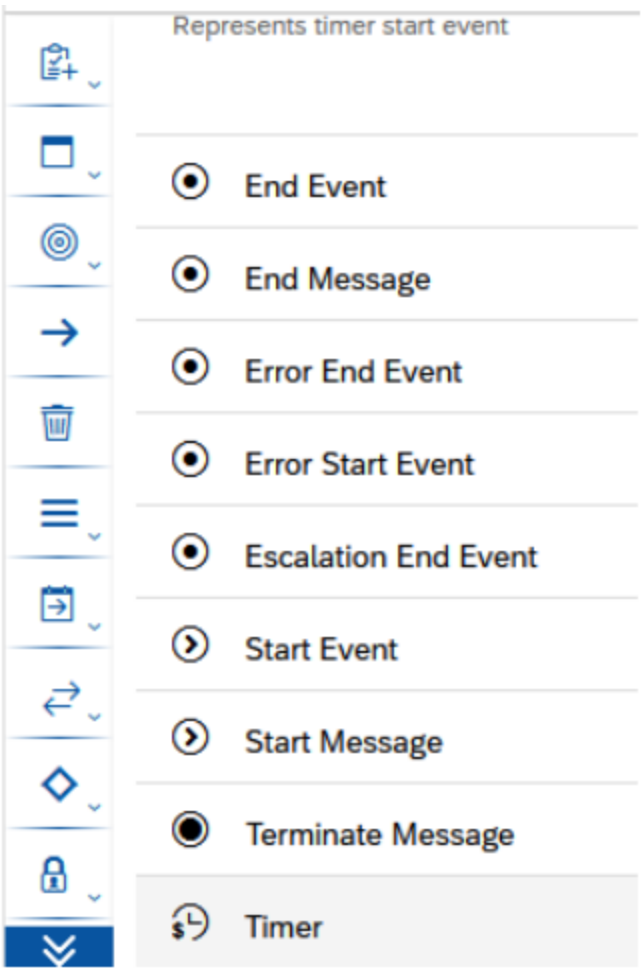

Figure 12.2 Adding a Timer to the Integration Flow

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

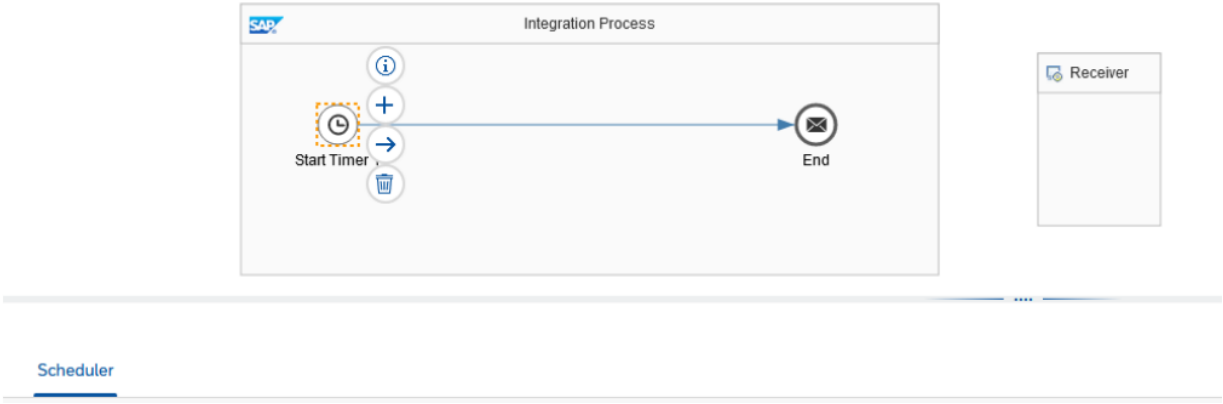

Figure 12.3 Example after the addition of a timer

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

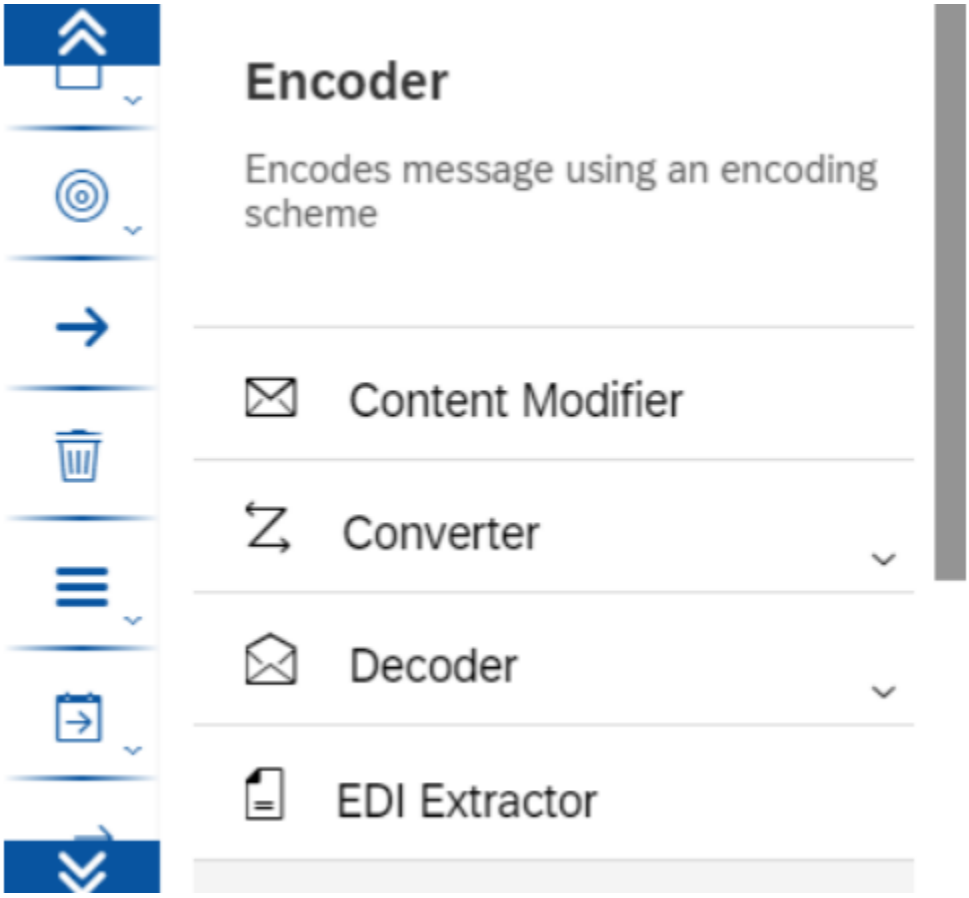

Figure 12.4 Adding a Content Modifier

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 run. The second property is an expression representing the current datetime.

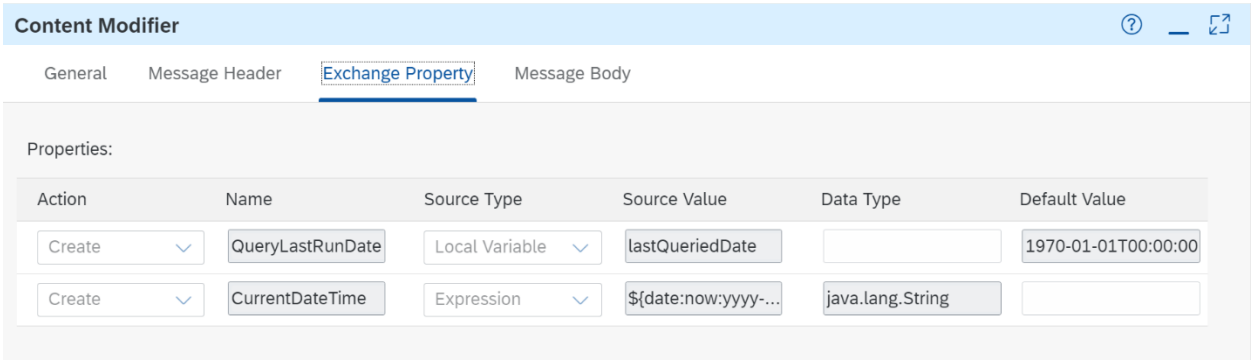

Figure 12.5 Content Modifier Exchage Property setup

9.1.4. Configure Receiver BigCommerce Adapter

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

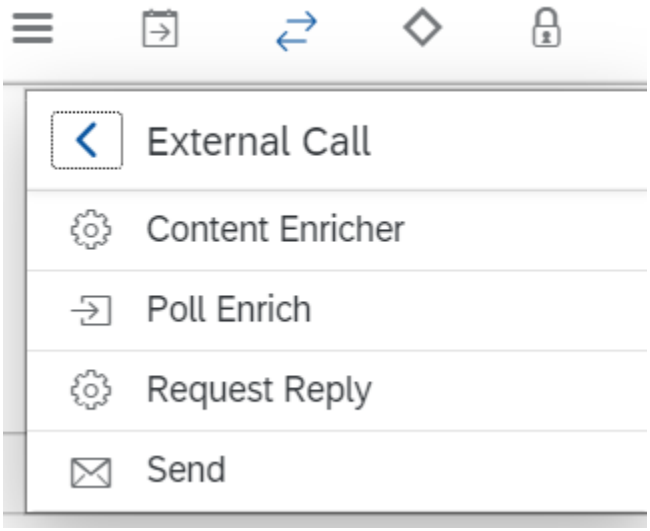

Figure 12.6 Adding a Request Reply step

- Select Request Reply and choose the arrow to drag the pointer to Receiver. Then, select the BigCommerce Adapter.
- Choose REST

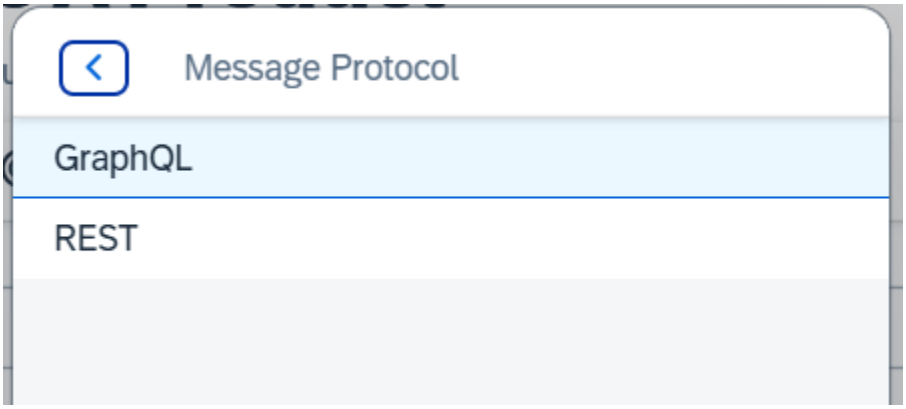

Figure 12.7 Selecting the Message Protocol in the BigCommerce Adapter

- In the "**Connection"** tab
- Add in the Base URL of your BigCommerce Store (change the XXXX part of the default value with the actual hash of your store)
- Supply the access token by referring to the Secure Parameter. In this scenario, the "bigCommerceStoreAccessToken" has been created as a Secure Parameter within SAP Integration Suite Security Materials, and it contains the actual Store Access Token.

- 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 is added to the message processing logs.
    - 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.

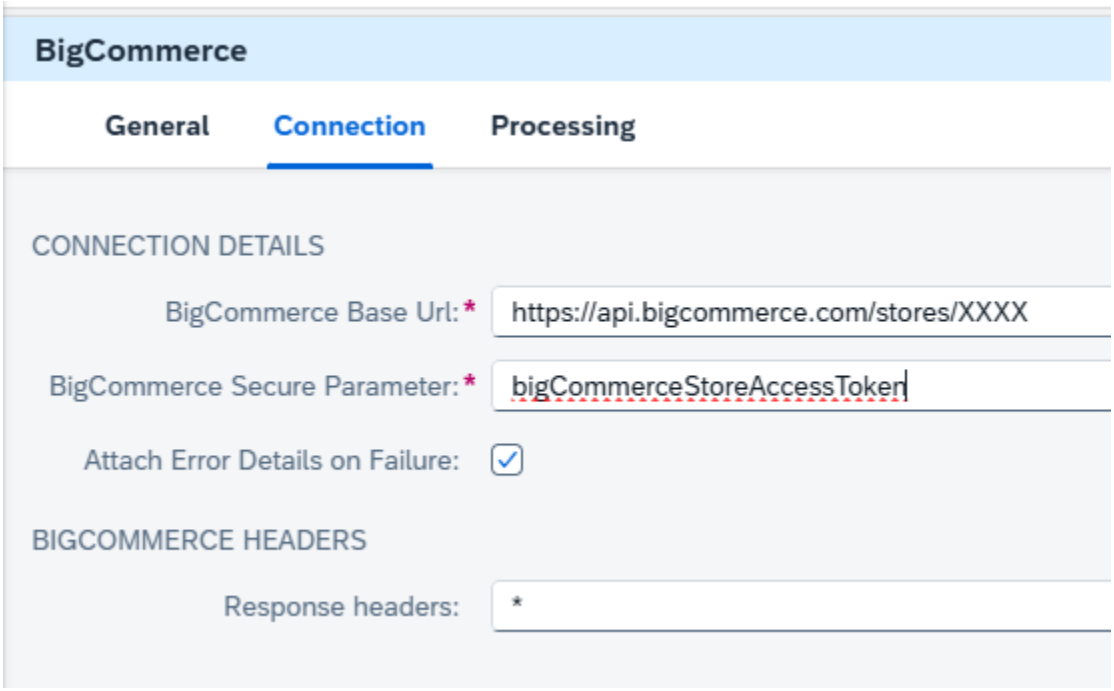

Figure 12.8 Configuring the BigCommerce Adapter Connection

- In the “**Processing”** tab
- Choose V2 for Api Version
- Choose Order for Resource
- For Operations choose General
- For Operation choose Get All Orders
- Add two query parameters:

min_date_created with value ${property.QueryLastRunDate}

max_date_created with value ${property.CurrentDateTime}

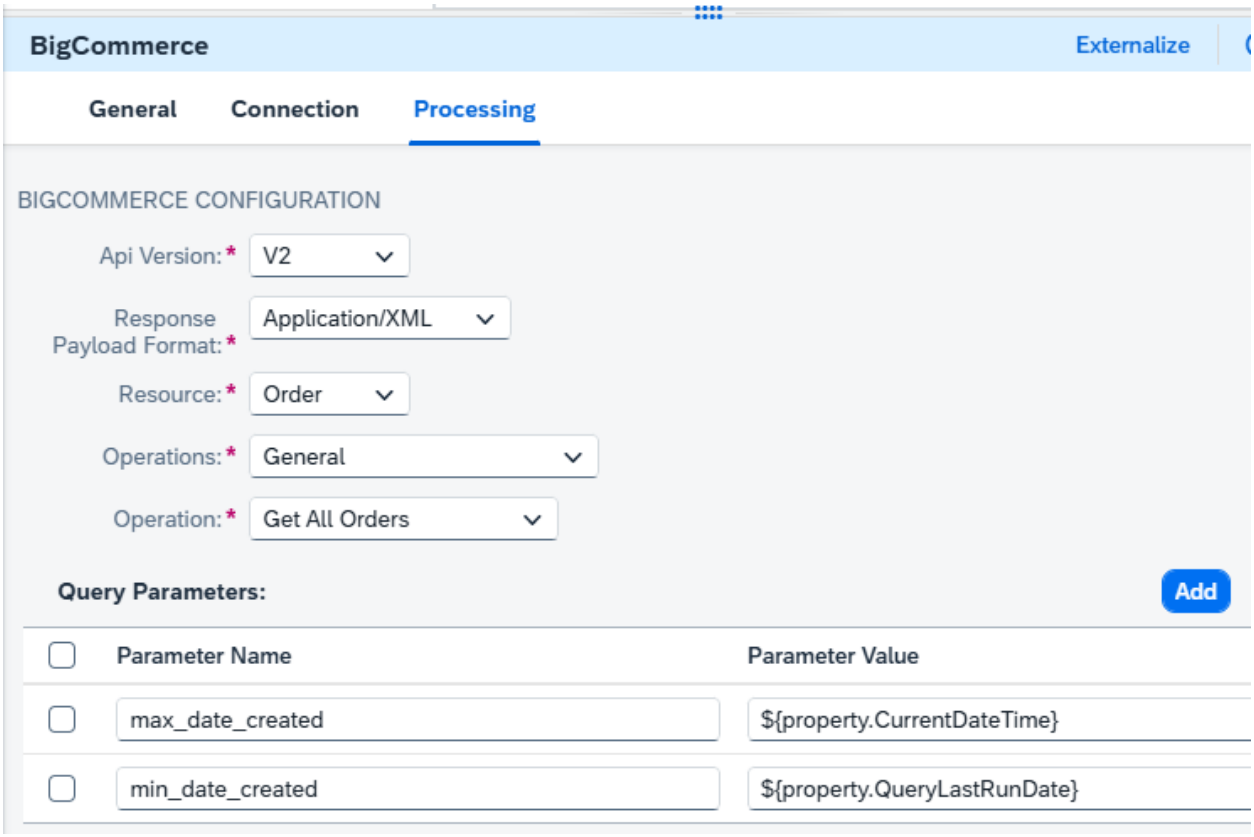

Figure 12.9 Processing Configuration for the BigCommerce Adapter

9.1.5. Configure Write Variables Component

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

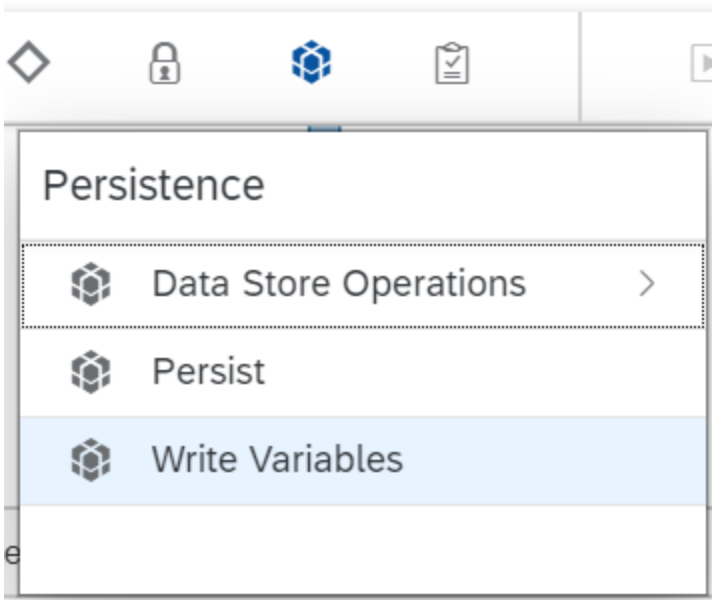

Figure 12.10 Introducing a "Write Variables" Integration Flow step

- In Processing tab set "lastQueriedDate" to be assigned the value of "CurrentDateTime" and to be of type "Property"

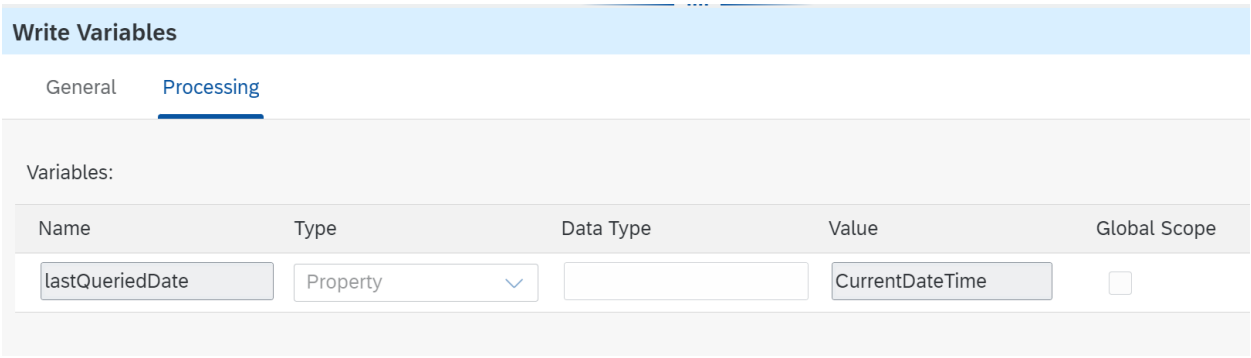

Figure 12.11 Writing a new variable

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 BigCommerce

- Add a script step after the "Write Variables" Component:

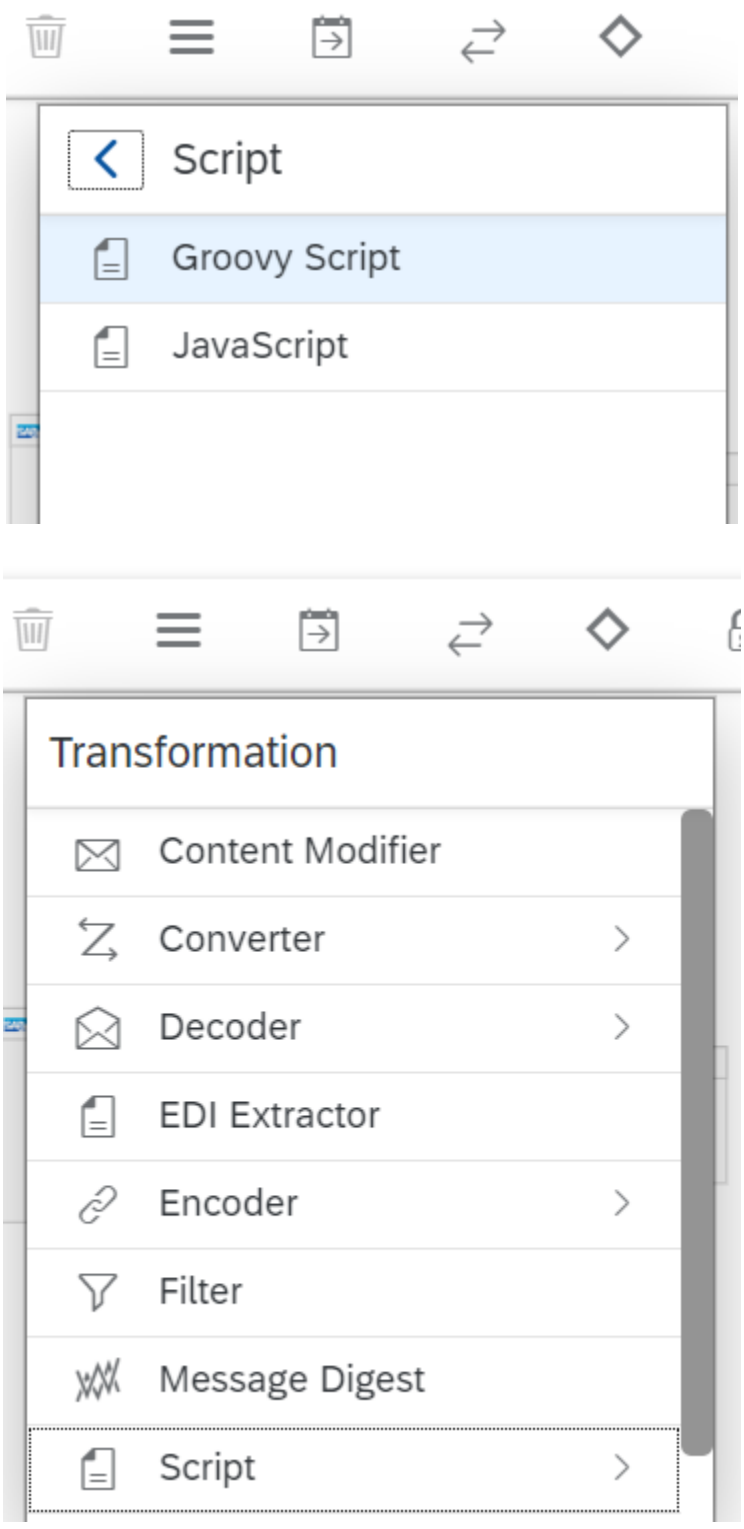

Figure 12.12 Adding a "Groovy Script" Integration Flow

- Add a Groovy Script

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

.

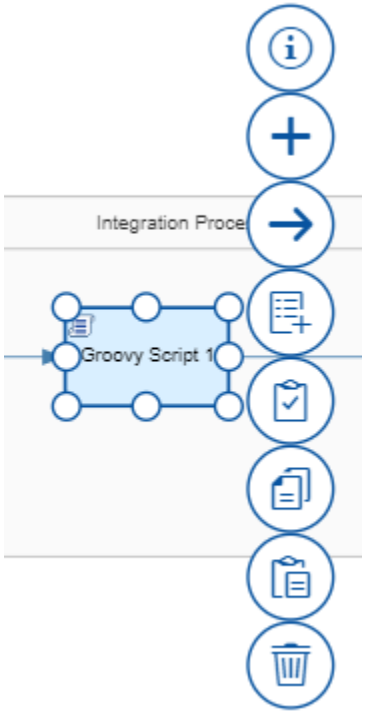

Figure 12.13 Creating a new Groovy Script

Add the following Groovy script:

```
import com.sap.gateway.ip.core.customdev.util.Message;

def Message processData(Message message) {
    def body = message.getBody(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

- Save the integration flow as a version and click on Deploy.
- Check the results of the logs

- Navigate to “Monitor -> Integrations and API’s -> Monitor Message Processing -> All artifacts”
- Select the artifact that you have deployed and then click on “*Log current Payload*”

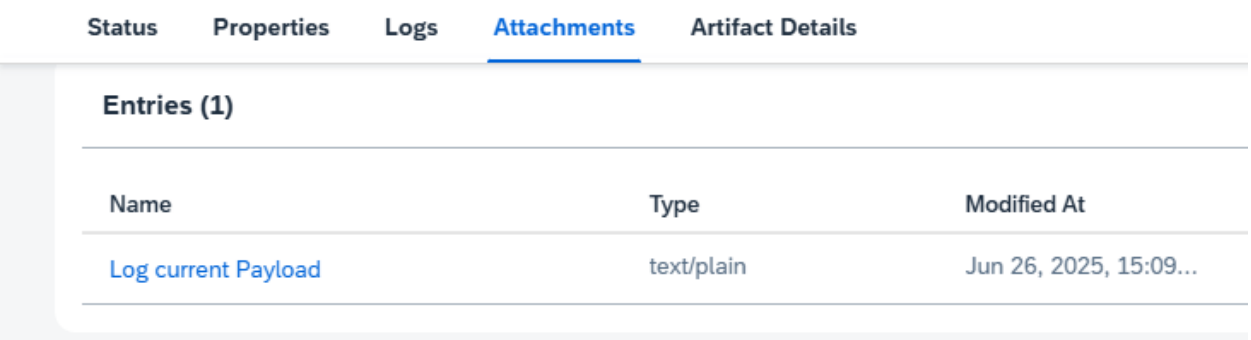

Figure 12.14 Observing the log from the Integration Flow Execution

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

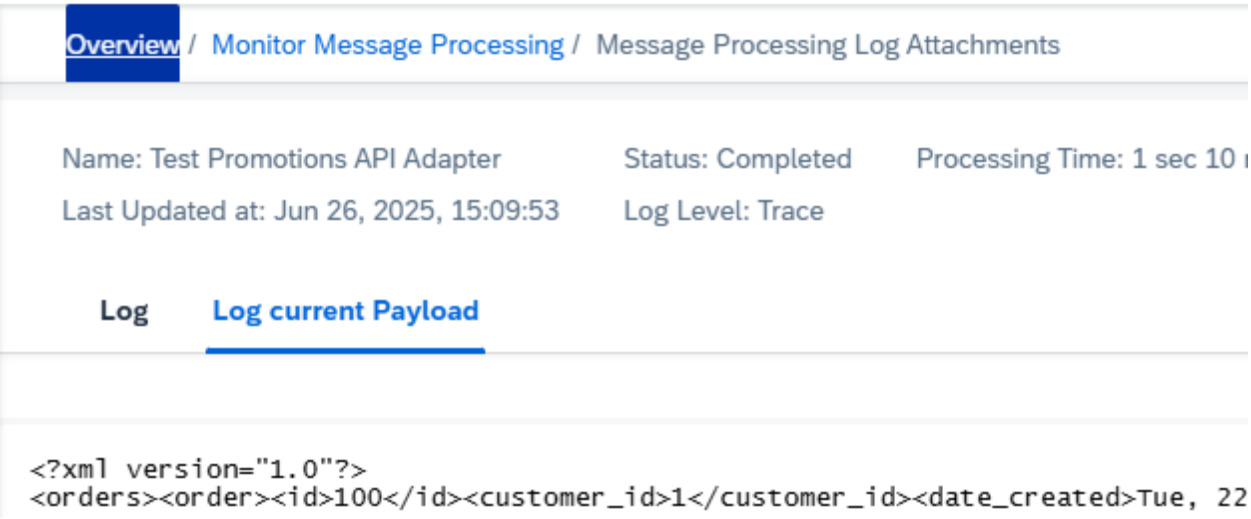

Figure 12.15 BigCommerce Orders present in the Log Attachment

10. References

10.1. Locate the BigCommerce Store Hash

To find the store hash that can be used as part of the BigCommerce Base URL field of the Connection Tab, login to your BigCommerce Management Dashboard. Then check the URL address.

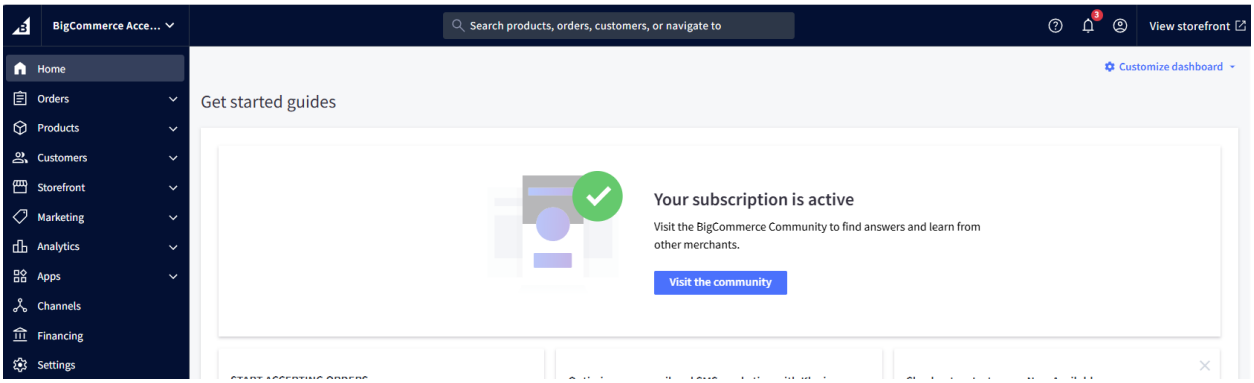

Figure 13.1 BigCommerce Management Dashboard

The Store's Hash can be found right after the **store-** part of the URL and right before the **.mybigcommerce.com** (the hash is a unique identifier for your store comprised of a short sequence of lower-case letters and numbers).

Figure 13.2 BigCommerce Store Management Dashboard URL Address