# Shopify Export Integration

You can use the Shopify Export Integration connector to export Metafield data to Shopify.

This topic contains:


- Prerequisites
- Limitations
- Use the TD Console to Create Your Connection
    - Obtain Shopify Credentials
        - Store Name Configuration
    - Create a New Connection
    - Configure Export Results in Your Data Connection
        - Configure the Connection by Specifying the Parameters
            - Use an Existing Connection
            - Create a New Shopify Connection.
    - Optional: Use of Scheduled Jobs for Export
    - Optional: Configure Export Results in Workflow


## Prerequisites

- Basic knowledge of Treasure Data
- Basic knowledge of Shopify

## Limitations

- Result output schema must match require columns(name & data type) with resource

## Use the TD Console to Create Your Connection

### Obtain Shopify Credentials

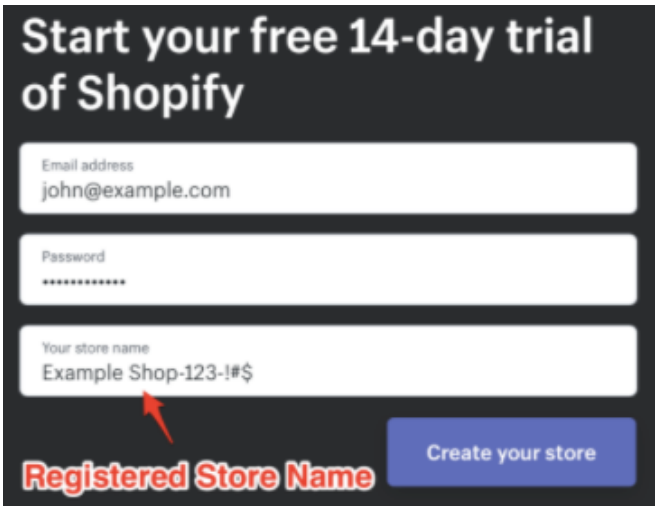
To obtain Shopify Credentials, complete the following steps:

1. Sign up at https://www.shopify.com/ to create an online store for the user.
2. Enter the details about the user.
3. Enter additional details about the business.
4. Create a private app.
5. The user can see and use the API credentials for the store and can to connect with different external applications. This API key is required to create a connection to Treasure Data.

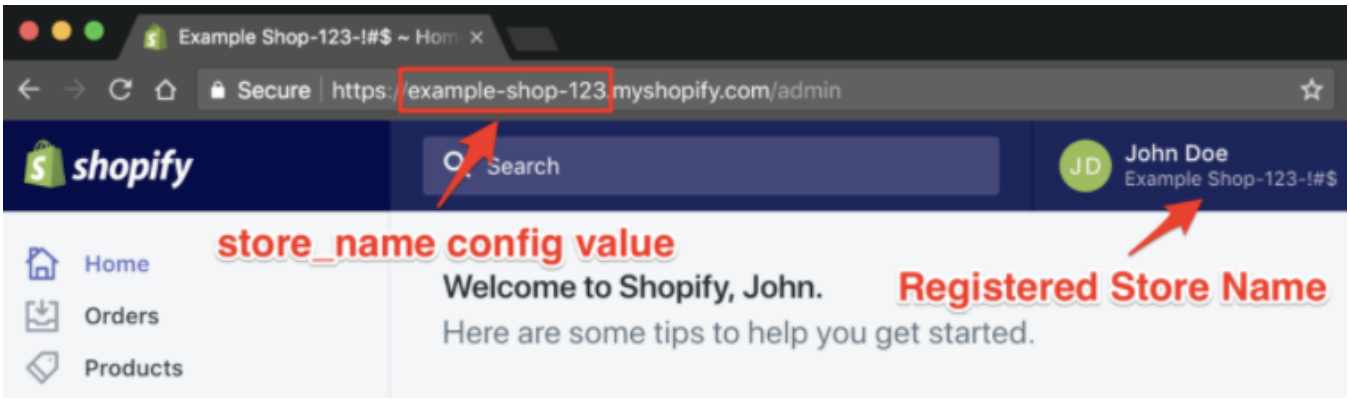
#### Store Name Configuration

Shopify translates your free-form Store Name into a URL-friendly value. It truncates special characters and replaces spaces with hyphens as shown in this example:

Example Shop-123-!#$ becomes example-shop-123



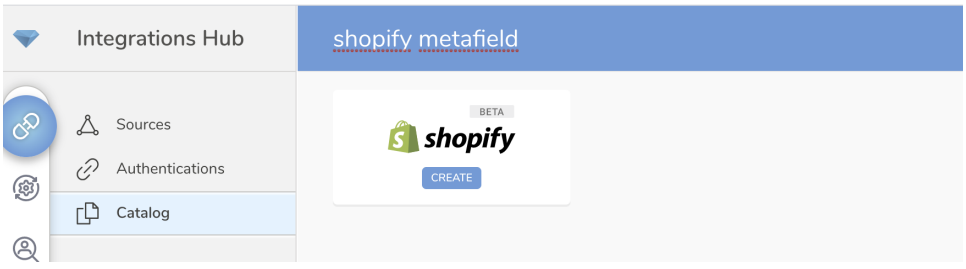

You need to use the translated value (in Admin URL, after signing in):



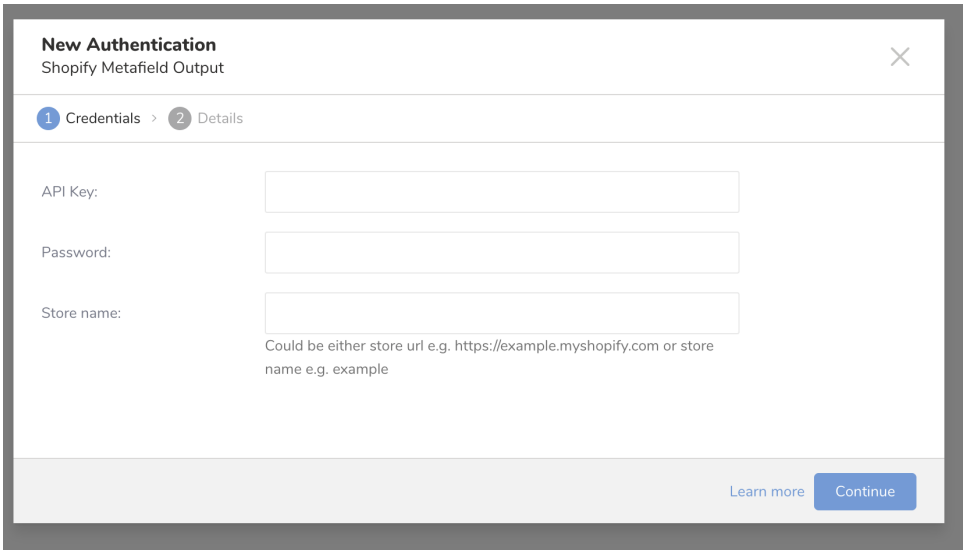

## Create a New Connection

In Treasure Data, you must create and configure the data connection, to be used during export, prior to running your query. As part of the data connection, you provide authentication to access the integration.

1. Open the **TD Console**.
2. Navigate to **Integrations Hub** > **Catalog**
3. Search for **Shopify Metafield**. Select **Create**.




4. The following dialog opens. Complete the required fields and then select **Continue**.




5. Enter a name for your connection.
6. Select **Done.**

## Configure Export Results in Your Data Connection

In this step you create or reuse a query. In the query, you configure the data connection.

You sometimes need to define the column mapping in the query.

### Configure the Connection by Specifying the Parameters

1. Open the **TD Console**.
2. Navigate to **Data Workbench** > **Queries**.
3. Select the query that you plan to use to export data.

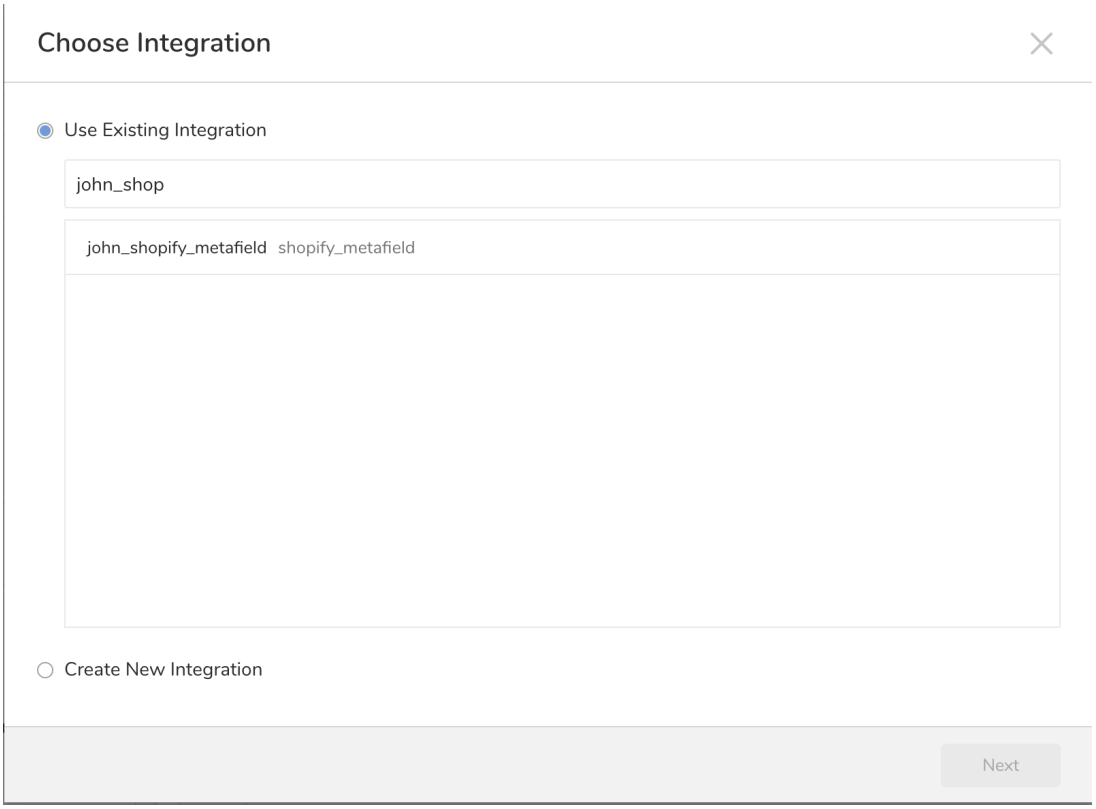
Each type of resource requires specific column and column data type.

| Action | Resource | Column require | Query | Note |
|---|---|---|---|---|
| Create | Shop | [key: String, namespace: String, value: any] | SELECT key, namespace, value FROM table | |
| Create | Product | [product_id: Integer, key: String, namespace: String, value: any] | SELECT product_id, key, namespace, value FROM table ORDER BY product_id | Please use ORDER BY to increase performance |
| Create | Product Variant | [variant_id: Integer, key: String, namespace: String, value: any] | SELECT variant_id, key, namespace, value FROM table ORDER BY variant_id | Please use ORDER BY to increase performance |
| Create | Product Image | [product_id: Integer, image_id: Integer, key: String, namespace: String, value: any] | SELECT product_id, image_id, key, namespace, value FROM table ORDER BY product_id, image_id | Please use ORDER BY to increase performance |
| Create | Custom Collection | [custom_collection_id: Integer, key: String, namespace: String, value: any] | SELECT custom_collection_id, key, namespace, value FROM table ORDER BY custom_collection_id | Please use ORDER BY to increase performance |
| Create | Smart Collection | [smart_collection_id: Integer, key: String, namespace: String, value: any] | SELECT smart_collection_id, key, namespace, value FROM table ORDER BY smart_collection_id | Please use ORDER BY to increase performance |
| Create | Customer | [customer_id: Integer, key: String, namespace: String, value: any] | SELECT customer_id, key, namespace, value FROM table ORDER BY customer_id | Please use ORDER BY to increase performance |
| Create | Order | [order_id: Integer, key: String, namespace: String, value: any] | SELECT order_id, key, namespace, value FROM table ORDER BY order_id | Please use ORDER BY to increase performance |
| Create | Draft Order | [draft_order_id: Integer, key: String, namespace: String, value: any] | SELECT draft_order_id, key, namespace, value FROM table ORDER BY draft_order_id | Please use ORDER BY to increase performance |
| Create | Blog | [blog_id: Integer, key: String, namespace: String, value: any] | SELECT blog_id, key, namespace, value FROM table ORDER BY blog_id | Please use ORDER BY to increase performance |
| Create | Article | [blog_id: Integer, article_id: Integer, key: String, namespace: String, value: any] | SELECT blog_id, article_id, key, namespace, value FROM table ORDER BY blog_id, article_id | Please use ORDER BY to increase performance |
| Create | Page | [page_id: Integer, key: String, namespace: String, value: any] | SELECT page_id, key, namespace, value FROM table ORDER BY page_id | Please use ORDER BY to increase performance |
| Update | | [metafield_id: Integer, value: any] | SELECT metafield_id, value FROM table | |

4. Select **Export Results** located at top of your query editor. The Choose Integration dialog opens.
You have two options when selecting a connection to use to export the results, using an existing connection or creating a new one.

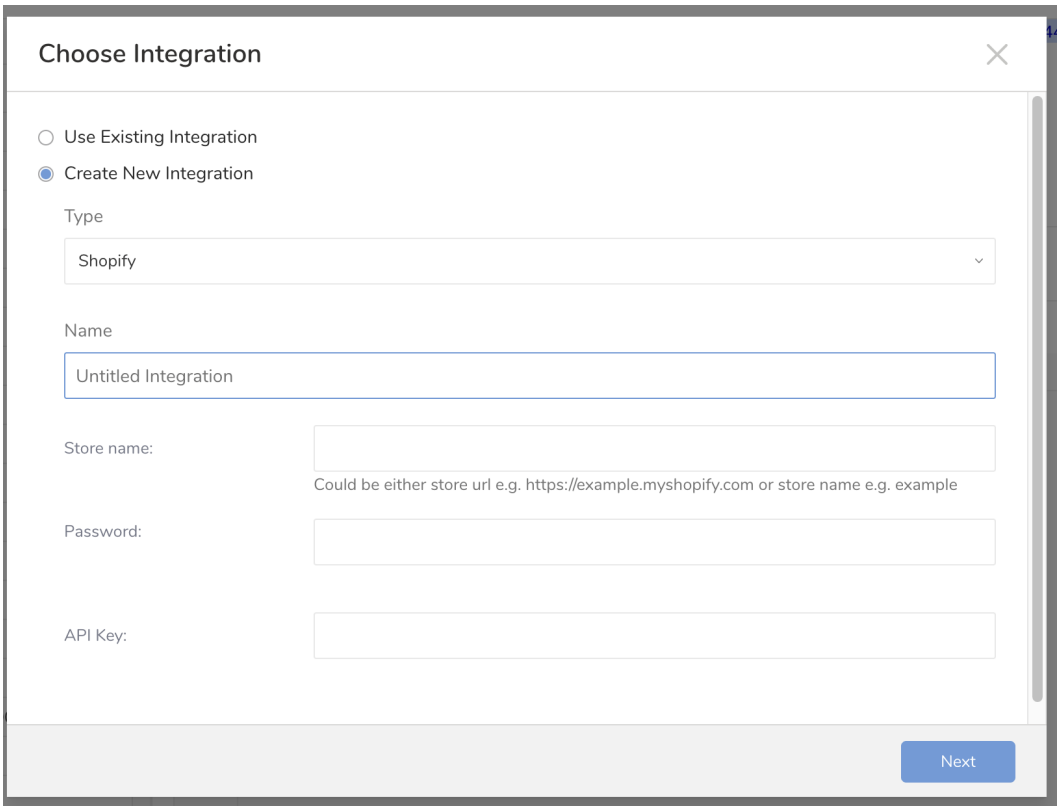
### Use an Existing Connection

1. Type the connection name in the search box to filter.

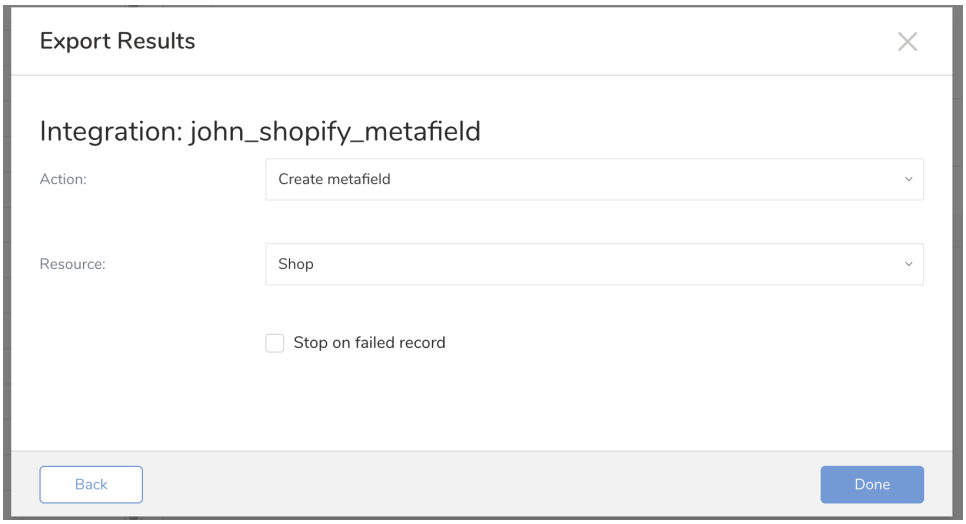



2. Select your connection. Select **Next**.

### Create a New Shopify Connection.




1. Select **Create New Integration.**
2. Choose the connection **Type.**
3. Type a **Name** for your connection.
4. Enter the Store Name, Password, and API Key**.**
5. Select Next. The following dialog opens.




6. Select an **Action**. If the action is to **Create Metafield**, select the resource to be created with the metafield.
7. Select **Stop on failed record** if you want the job to stop when an error occurs.
8. Select **Done.**

## Optional: Use of Scheduled Jobs for Export

You can use Scheduled Jobs with Result Export, to periodically write the output result to a target destination that you specify.

## Optional: Configure Export Results in Workflow

*Create metafield*

```
timezone: UTC

_export:
  td:
    database: sample_datasets

+td-result-into-target:
  td>: queries/sample.sql
  result_connection: your_connections_name
  result_settings:
    apikey: {apikey}
    password: {password}
    store_name: {store_name}
    action: create
    resource: shop
    stop_on_failed_record: false
```

Update metafield

```
timezone: UTC

_export:
  td:
    database: sample_datasets

+td-result-into-target:
  td>: queries/sample.sql
  result_connection: your_connections_name
  result_settings:
    apikey: {apikey}
    password: {password}
    store_name: {store_name}
    action: update
    stop_on_failed_record: false
```

Click here for more information on using data connectors in workflow to export data.