**Documentation for Shopify Data Extraction Script**

Overview

The Shopify Data Extraction script is designed to retrieve order data from the Shopify API for a specific shop. It utilizes the requests library to interact with the Shopify API and the pandas library for data manipulation. The script extracts relevant information about orders, including order details and line items, and stores the data in a structured CSV format.

Prerequisites

Before running the script, ensure that the required libraries (pandas, requests, pytz) are installed. You can install them using the following command:

pip install pandas requests pytz

Configuration

The script requires the following configuration parameters to access the Shopify store:

* token: Shopify API access token.
* shop: Shopify store name.
* api\_version: Shopify API version.
* shop\_url: Shopify API endpoint for the specific shop.

Ensure that these parameters are correctly set before running the script.

* **Workflow**
* Calculate Time Window: The script calculates the start of the previous day (yesterday) from 12 am in the UTC timezone and sets the end time to the current UTC time. This time window is used to extract orders created during this period.
* Get Data from Shopify API:
* The get\_data function sends a GET request to the Shopify API's Orders endpoint with the specified time window.
* It retrieves paginated data, handling multiple pages if necessary.
* Data Normalization:
* The script uses pd.json\_normalize to normalize the nested JSON data returned from the Shopify API into a flat DataFrame.
* Data Processing:
* Relevant columns (name, id, created\_at, line\_items) are selected from the DataFrame.
* The line\_items column is exploded to handle multiple line items per order.
* Flatten Nested JSON:
* The flatten\_json function is used to flatten the nested JSON structure within the line\_items column.
* Expand Line Items:
* The expand\_line\_items function is applied to expand the line\_items column, creating new columns for each line item attribute.
* Concatenate DataFrames:
* The original and expanded line items DataFrames are concatenated to create the final structured DataFrame.
* Display Selected Columns:
* The script displays selected columns (name, id, created\_at, line\_item.name, line\_item.id) from the final DataFrame.
* Save to CSV:
* The final DataFrame is saved to a CSV file named "shopify\_data.csv" in the local directory.
* Running the Script

To execute the script, run the Python file shopify.py using the correct Python interpreter path. Make sure the required libraries are installed.

**python shopify.py**

The script will output information about the selected columns and save the data to a CSV file.

Notes

• Ensure that the Shopify store credentials and API version are correctly configured.

• Review the generated CSV file for the extracted data.

This documentation provides an overview of the script's functionality and usage. Refer to the inline comments in the code for more detailed explanations of each section.