int_klaviyo

Version <19.3.0>

Table of Contents

Table of Contents 2
Summary 3
Component Overview 4
Functional Overview 4
Prerequisites 4
Compatibility 4
Privacy, Payment 4
Cartridge Implementation Guide 5
Setup 5
Klaviyo Configuration 5
Business Manager Setup 7
Custom Code 8
External Interfaces 10
Firewall Requirements 10
Testing 10
Operations 13
Data Storage 13
Availability 13
Support 13
OCAPI Implementation Guide 14
Endpoints 14
Setup 14
Klaviyo App-side integration 15
User Guide 16
Roles, Responsibilities 16
Business Manager 16
Storefront Functionality 16
Known Issues 16
Release History 17

Summary

This documentation describes how to implement Klaviyo's cartridge and OCAPI integration into your Salesforce Commerce Cloud application. The Klaviyo integration features a self-contained cartridge that handles passing events in real-time from your website to Klaviyo, as well as integrations to OCAPI endpoints for product catalog and order data.

The cartridge includes preferences to be configured in the Business Manager and once enabled, contains all elements necessary to perform a successful best practices implementation of Klaviyo.

The OCAPI integration utilizes both "order_search" and "product_search" endpoints to sync catalog and order data.

Events tracked through cartridge:

- Active on Site
- Added to Cart
- Checkout Started
- Order Confirmation
- Site Search
- Viewed Category
- Viewed Product

Events tracked through OCAPI:

- Placed Order
- Ordered Product

Component Overview

Functional Overview

The Klaviyo cartridge provides integration for the following functionality:

- Integration of Klaviyo JavaScript library (referenced here: https://www.klaviyo.com/docs/getting-started)
- Integration of Klaviyo Server Side library (referenced here: https://www.klaviyo.com/docs/http-api)
- Klaviyo JavaScript snippet integration and setup – Klaviyo tags can be fired in any page with the help of a modular ISML template and controller integration.
- Klaviyo Track Server Side API – Cartridge includes a controller to call the Klaviyo Track Server Side API. This is to fire order related events such as Order Confirmation to Klaviyo.
- Klaviyo Subscribe API – The cartridge also includes a controller to call the Klaviyo Lists API to check for the presence of and subscribe a user to a Klaviyo List. This can be used for functionality such as Newsletters and other forms.

Prerequisites

Before performing the steps in this document, ensure that you and your organization have satisfied the following criteria:

- You have reached out to Klaviyo and have set up a Klaviyo Account (where you can find your Klaviyo Organization ID and API Key, more details below).
- You have downloaded and unzipped the Klaviyo cartridge ZIP file from Salesforce Commerce Cloud Marketplace or your Klaviyo Consultant. The cartridge ZIP file contains the Klaviyo cartridge package residing in the folder int_klaviyo, the preferences configuration XML file and custom attribute definition file.

Compatibility

The cartridge is designed for Salesforce Commerce Cloud API version 18.8+, but if you are on older versions please contact us for confirmation, as it should be supported.

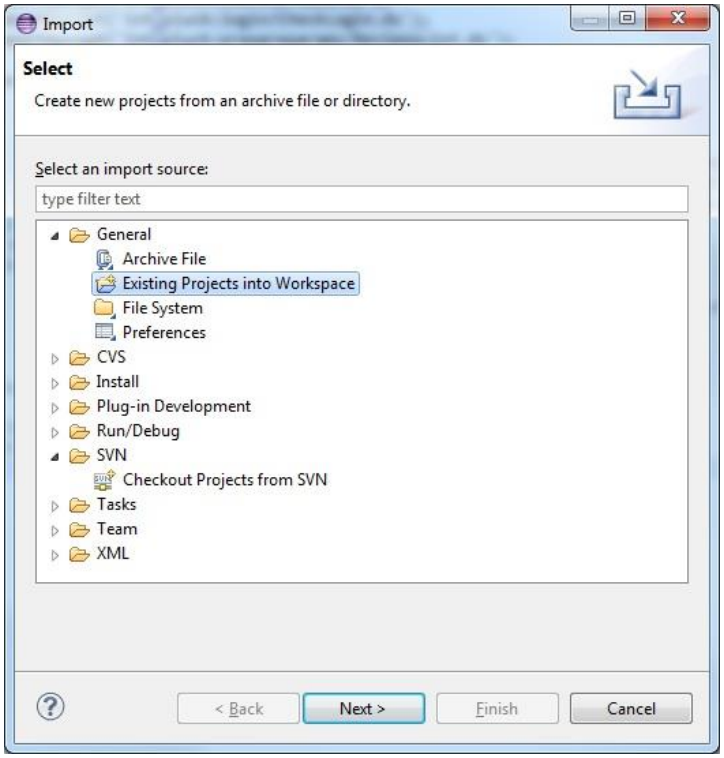
Privacy, Payment

Please see the Klaviyo Privacy Policy here - https://www.klaviyo.com/privacy
Contact your account manager for pricing.

Cartridge Implementation Guide

Setup

- Open Salesforce Commerce Cloud Studio.
- Import downloaded cartridge. Select on import screen **int_klaviyo**

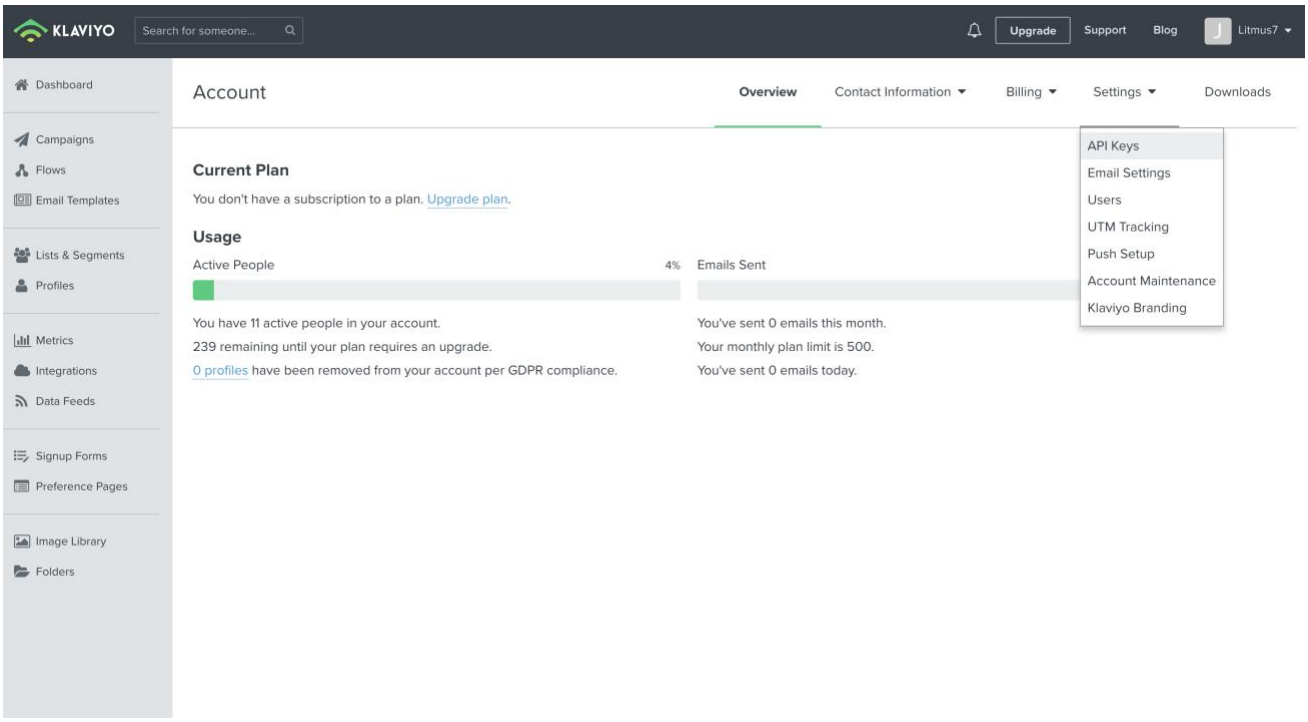


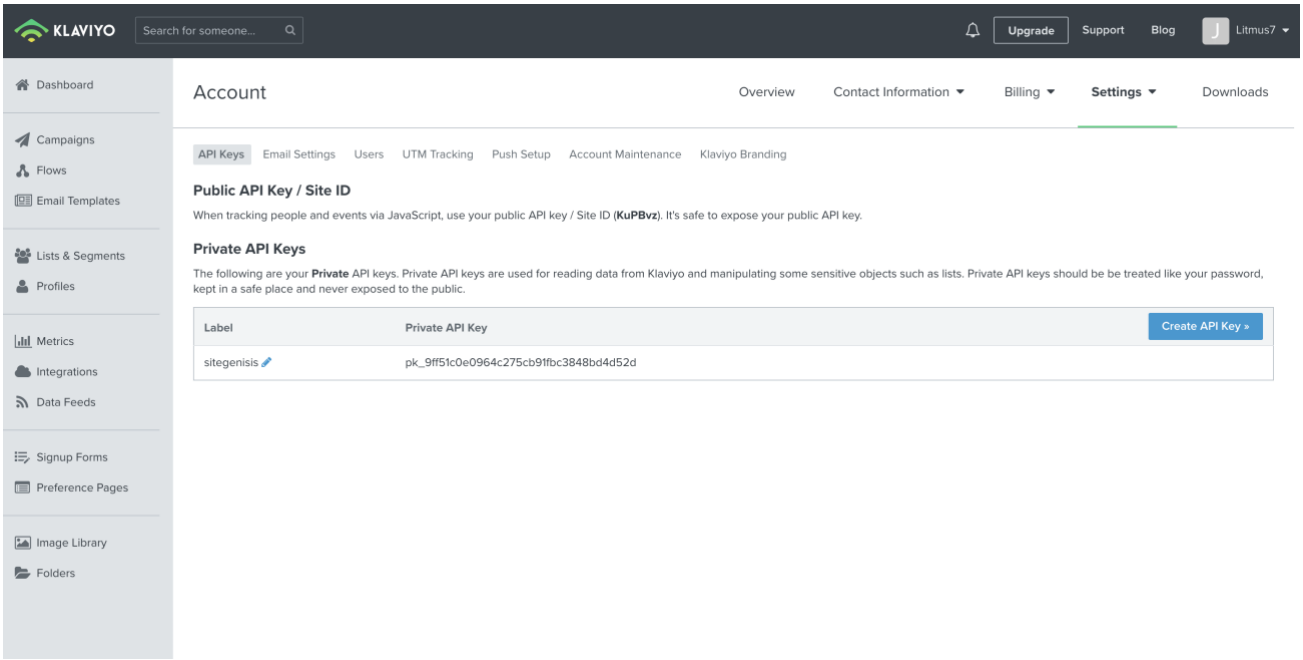
- Link cartridge to the sandbox: Select the Salesforce Commerce Cloud connection and select Properties. Select Project References and check in **int_klaviyo**

Klaviyo Configuration

Once you sign up for your Klaviyo account, you can get your Account ID and API key for setup from your Klaviyo account by following the below steps:

Navigate to the account section by selecting the drop down on the top right. Once in the account page, select the API Keys option under settings to reveal the Account ID and API key.


KLAVIYO
Search for someone...
Upgrade
Support
Blog
Litmus7
Dashboard
Campaigns
Flows
Email Templates
Lists & Segments
Profiles
Metrics
Integrations
Data Feeds
Signup Forms
Preference Pages
Image Library
Folders
Account
Overview
Contact Information
Billing
Settings
Downloads
API Keys
Email Settings
Users
UTM Tracking
Push Setup
Account Maintenance
Klaviyo Branding
Current Plan
You don't have a subscription to a plan. Upgrade plan.
Usage
Active People
4%
Emails Sent
You have 11 active people in your account.
239 remaining until your plan requires an upgrade.
0 profiles have been removed from your account per GDPR compliance.
You've sent 0 emails this month.
Your monthly plan limit is 500.
You've sent 0 emails today.

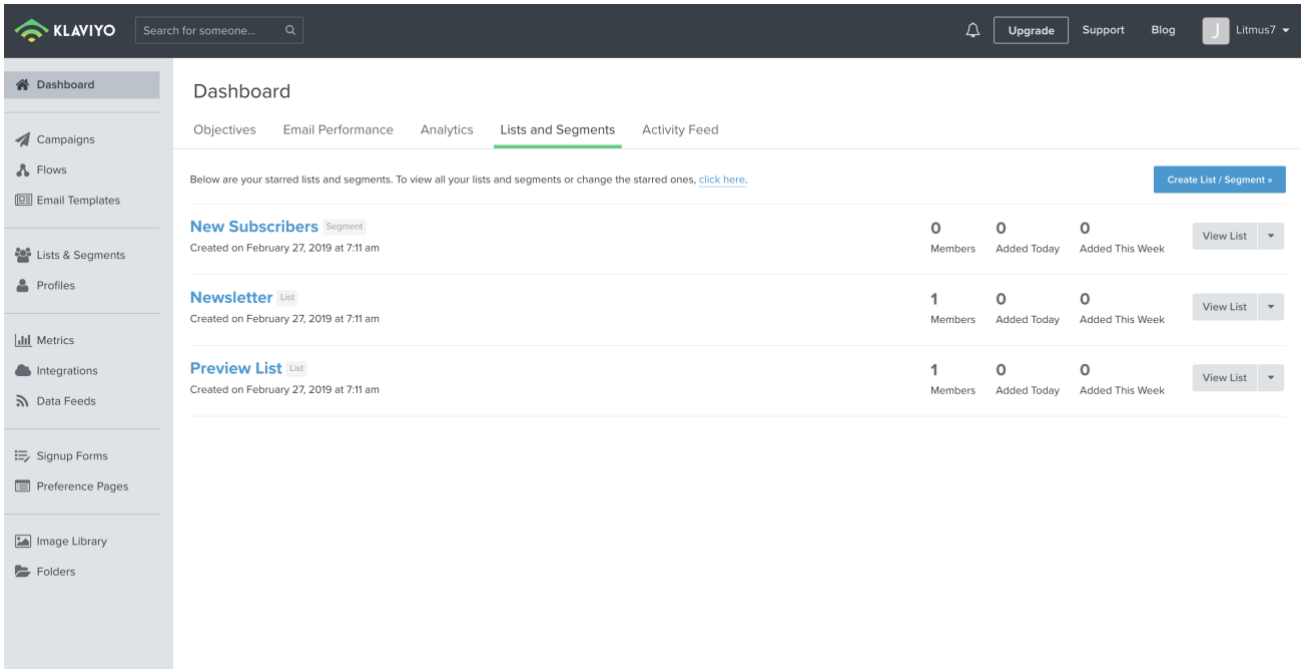

KLAVIYO
Search for someone...
Upgrade
Support
Blog
Litmus7
Dashboard
Campaigns
Flows
Email Templates
Lists & Segments
Profiles
Metrics
Integrations
Data Feeds
Signup Forms
Preference Pages
Image Library
Folders
Account
Overview
Contact Information
Billing
Settings
Downloads
API Keys
Email Settings
Users
UTM Tracking
Push Setup
Account Maintenance
Klaviyo Branding
Public API Key / Site ID
When tracking people and events via JavaScript, use your public API key / Site ID (KuPBvz). It's safe to expose your public API key.
Private API Keys
The following are your Private API keys. Private API keys are used for reading data from Klaviyo and manipulating some sensitive objects such as lists. Private API keys should be be treated like your password, kept in a safe place and never exposed to the public.
Label
Private API Key
Create API Key »
sitegenisis
pk_9ff51c0e0964c275cb91fbc3848bd4d52d

Business Manager Setup

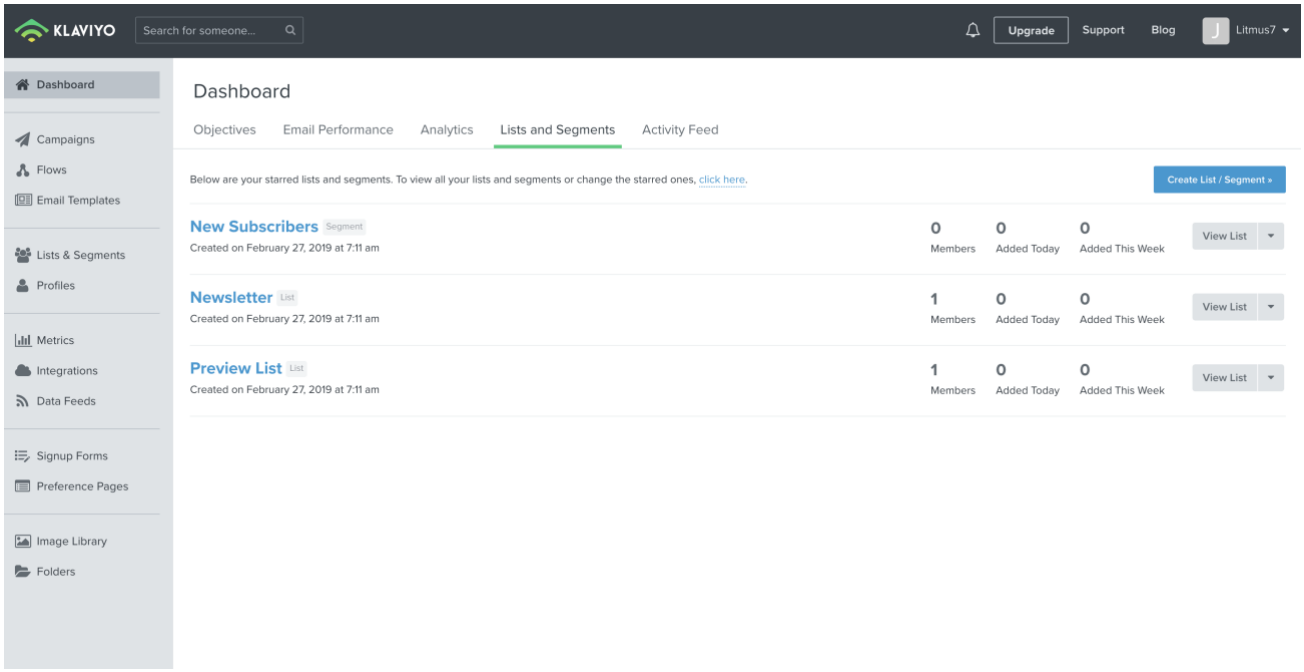
- Once the cartridge is imported, it needs to be added to the list of cartridges in your site. Go to Business Manager: Administration -> Sites -> Manage Sites. Select your site, then select **Settings** tab. In cartridge path at the end, add **int_klaviyo.**
- Import the services file that come with the cartridge (int_klaviyo\metadata\klaviyo-services.xml). This creates the service definitions for the Klaviyo track and lists server side API. Go to Administration->Operations->Import and Export->Services. Upload and import the klaviyo-services.xml file. This will create three services:
    - KlaviyoSubscriptionService
    - KlaviyoTrackService
    - KlaviyoCheckSubscriptionService

For the KlaviyoSubscriptionService and KlaviyoCheckSubscriptionService, replace the ID in the service URL with the Klaviyo List ID that you will be using to subscribe users. For e.g.: the import file will have the service URL as https://a.klaviyo.com/api/v2/list/**Mm2HRr**/members. Replace the section highlighted in blue with your List ID from klaviyo

To get the List ID in Klaviyo, navigate to Lists and Segments in the Dashboard section.



Select the List you want and the URL will contain the List ID.



- Import the meta-data file (int_klaviyo\metadata\custom_atribute_klaviyo.xml). This creates site level custom attributes for storing the Klaviyo account ID and API Key. To import, navigate to Merchant Tools > Administration >Site Development->Import and Export. Under the meta data section, upload and import the custom_atribute_klaviyo.xml file. This creates the following attributes in Site Preferences:
    - Klaviyo Account - This is mandatory and used to authenticate to the Javascript API.
    - Klaviyo API Key - The private API key for the server side API.
    - Klaviyo Enabled – To Enable the Klaviyo Service

Custom Code

- Import Add the following code to the end of your minicart.isml file:

```
<isif condition="${pdict.CurrentHttpParameterMap.cartAction == 'add' ||
pdict.CurrentHttpParameterMap.cartAction == 'update'}">
    <isinclude url="${URLUtils.url('Klaviyo-RenderKlaviyoAddToCart')}"/>
</isif>
```

- Add the following code to your footer_UI.isml file:

```
<script src="${URLUtils.staticURL('/js/klaviyoFooterUtils.js')}"></script>
```

include this script after the includes for *jquery-1.11.1.min.js* & *jquery-ui.min.js*

- Add the following code to the end of your footer_UI.isml file:

```
<isinclude
url="${URLUtils.url('Klaviyo-RenderKlaviyo',
        'title', request.pageMetaData.title,
        'pagecontexttype', ('pageContext' in this && !empty(pageContext)) ? ''+pageContext.type : null,
        'pagecontexttitle', ('pageContext' in this && !empty(pageContext)) ? ''+pageContext.title : null,
        'searchterm', request.httpParameterMap.q.stringValue,
        'searchresultscount', (!empty(pdict.ProductSearchResult) ? ''+pdict.ProductSearchResult.count : null),
        'productid', (!empty(pdict.Product) ? pdict.Product.ID : null),
        'pagecgid',  request.httpParameterMap.cgid.stringValue,
        'orderno', (!empty(pdict.Order) ? pdict.Order.orderNo : null)
)}" />
```

- Change the storefront controllers cartridge name to your cartridge name wherever refernced in the Klaviyo.js and EmailUtils.js files

  For e.g.: in line 5 of the Klaviyo.js file, replace the app_controllers with your controller cartridge name.

External Interfaces

All calls are made from tracking and identification on the pages are done through the Javascript Track and Identify calls which can be referenced in https://www.klaviyo.com/docs/getting-started.

For the server side calls pertaining to List Subscription and Track call for order confirmation, please refer https://www.klaviyo.com/docs/http-api and https://www.klaviyo.com/docs/api/v2/lists .

Firewall Requirements

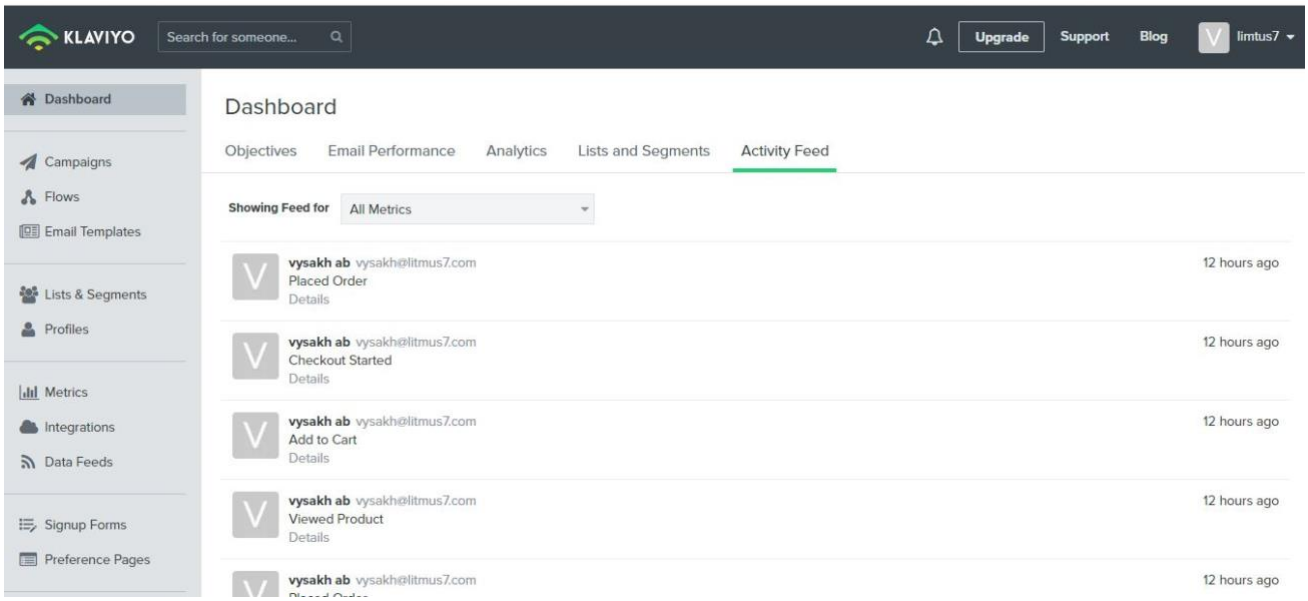
N/A

Testing

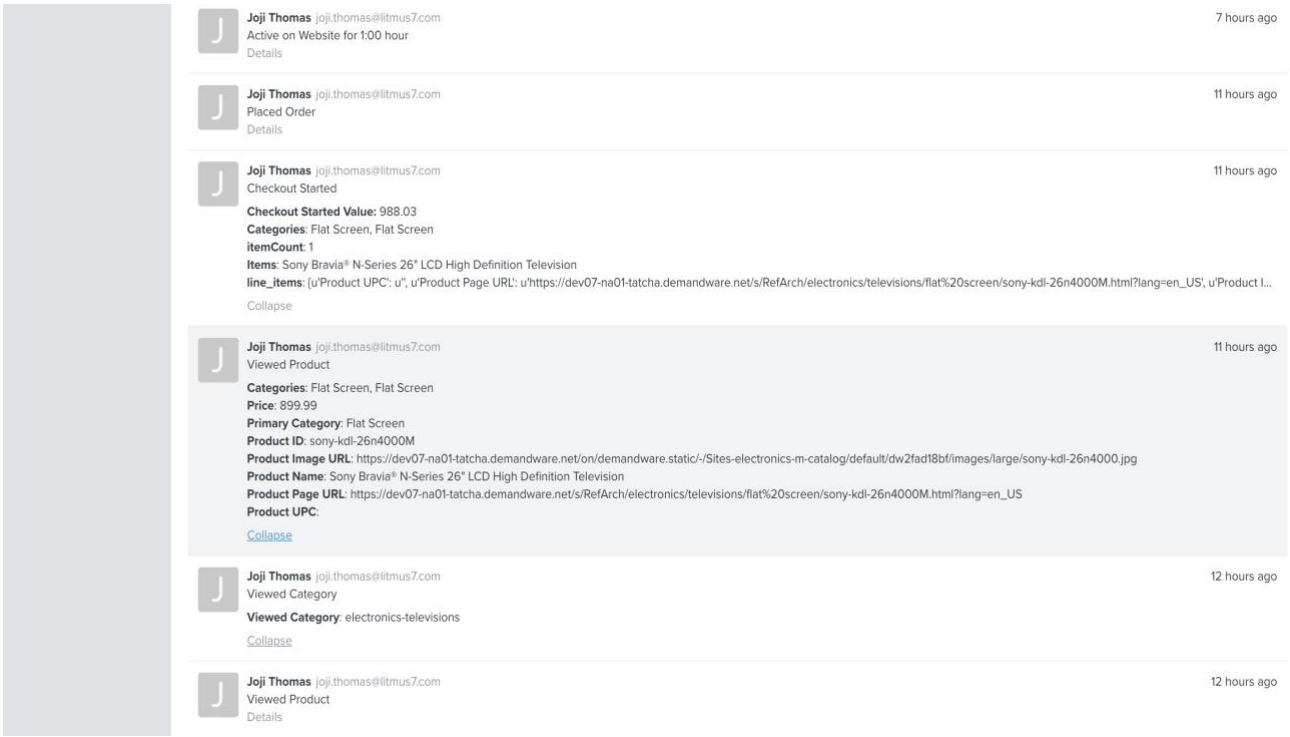
To see the service in action, login to the storefront and rest of the users activities like add to cart , active on site, search, order placed etc. will be tracked.

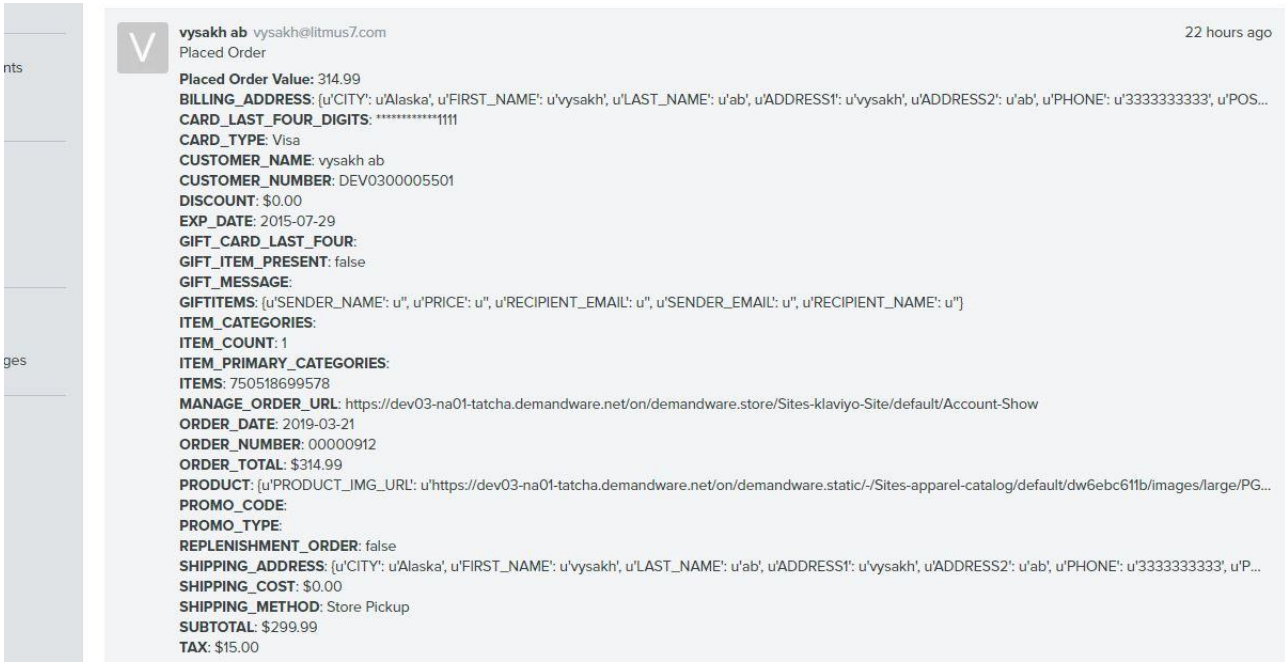
Login to Klaviyo and go to https://www.klaviyo.com/dashboard

Click on the Activity Feed to see all the activities and events tracked like Viewed Product, Add to Cart, Checkout Started , Subscribed to list etc..



Click on Details to see the event data




V
vysakh ab vysakh@litmus7.com
22 hours ago
Placed Order
Placed Order Value: 314.99
BILLING_ADDRESS: {u'CITY': u'Alaska', u'FIRST_NAME': u'vysakh', u'LAST_NAME': u'ab', u'ADDRESS1': u'vysakh', u'ADDRESS2': u'ab', u'PHONE': u'3333333333', u'POS...
CARD_LAST_FOUR_DIGITS: ************1111
CARD_TYPE: Visa
CUSTOMER_NAME: vysakh ab
CUSTOMER_NUMBER: DEV0300005501
DISCOUNT: $0.00
EXP_DATE: 2015-07-29
GIFT_CARD_LAST_FOUR:
GIFT_ITEM_PRESENT: false
GIFT_MESSAGE:
GIFTITEMS: {u'SENDER_NAME': u'', u'PRICE': u'', u'RECIPIENT_EMAIL': u'', u'SENDER_EMAIL': u'', u'RECIPIENT_NAME': u''}
ITEM_CATEGORIES:
ITEM_COUNT: 1
ITEM_PRIMARY_CATEGORIES:
ITEMS: 750518699578
MANAGE_ORDER_URL: https://dev03-na01-tatcha.demandware.net/on/demandware.store/Sites-klaviyo-Site/default/Account-Show
ORDER_DATE: 2019-03-21
ORDER_NUMBER: 00000912
ORDER_TOTAL: $314.99
PRODUCT: {u'PRODUCT_IMG_URL': u'https://dev03-na01-tatcha.demandware.net/on/demandware.static/-/Sites-apparel-catalog/default/dw6ebc611b/images/large/PG...
PROMO_CODE:
PROMO_TYPE:
REPLENISHMENT_ORDER: false
SHIPPING_ADDRESS: {u'CITY': u'Alaska', u'FIRST_NAME': u'vysakh', u'LAST_NAME': u'ab', u'ADDRESS1': u'vysakh', u'ADDRESS2': u'ab', u'PHONE': u'3333333333', u'P...
SHIPPING_COST: $0.00
SHIPPING_METHOD: Store Pickup
SUBTOTAL: $299.99
TAX: $15.00

Operations

Data Storage

N/A

Availability

N/A

Support

For technical support or any issues related to Klaviyo, please contact your Klaviyo Contact. For specific questions related to the cartridge, please reach out via email to support@klaviyo.com

OCAPI Implementation Guide

Endpoints

Klaviyo requires access to both “order_search” and “product_search” endpoints.

- Product_search: Connects your catalog to Klaviyo to enable functionality/features including product recommendation blocks in emails
- Order_search: Historically syncs order data to Klaviyo, and syncs order events every 30 minutes. For real-time order receipts, use the “Order Confirmation” event from the cartridge. The Ordered Product and Placed Order events will sync in additional data for segmentation and flow filtering, so is ideal for enhanced personalization not available from the Order Confirmation event.

Setup

- Navigate to https://account.demandware.com/dw/account/APIAdmin for the customer and add an API client for Klaviyo. The API Client ID and password will be required to generate the bearer token for the OCAPI.
- Once the API client is added, navigate to Administration -> Open Commerce API Settings (under Site Development) in the SFCC Business Manager.
- Add the following JSON under the type 'Data' and context 'Global (Organization Wide)'. If the client already has an OCAPI setting, add the section highlighted in blue to the existing 'clients' JSON array, substituting ${API_VERSION} for your OCAPI version:

```
{
 "_v":"${API_VERSION",
 "clients":
 [
  {
                                        "client_id": "xxxx-xxxx-xxxx-xxxx-xxxx",
                                        "resources": [
                                                {
                                                        "resource_id": "/product_search",
                                                        "methods": [
                                                                "post"
                                                        ],
                                                        "read_attributes": "(**)",
                                                        "write_attributes": "(**)"
                                                }, {
                                                        "resource_id": "/order_search",
                                                        "methods": [
                                                                "post"
                                                        ],
                                                        "read_attributes": "(**)",
                                                        "write_attributes": "(**)"
                                                }

                                        ]
                                }
 ]
}
```

- Replace the client ID with the API Client ID. This will expose the API to only the specified client ID. Replace the _v value with the OCAPI version. This is currently 19.5 but we can use 18.8 as well. Please note that the OCAPI version is reflected in the URL you call as well.

- The bearer token to call the OCAPI can be generated using the following:

  username: <API Client ID>

  password: <API Client Password>

- Please use the below request to get the OAuth token. It uses a base64 of the above username/password combination to ping the account management server to get an Oauth code for use.

- curl -X POST -H "Content-Type: application/x-www-form-urlencoded" -H "Authorization: Basic <xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>" -H "Cache-Control: no-cache" -d 'grant_type=client_credentials' 'https://account.demandware.com/dw/oauth2/access_token'

- The resulting code needs to be passed as a bearer token within the OCAPI request to https://xxxxxx.demandware.net/s/-/dw/data/${API_VERSION}/product_search?site_id=<Site Name>
  - ${API_VERSION} – is the version of the OCAPI you have configured
- NOTE* please let your Klaviyo Account Manager know your OCAPI version

Klaviyo App-side integration

- Navigate directly to: https://www.klaviyo.com/integration/demandware

- Use the following data:
  - Store Name: your website domain (example.com)

    (this will be the domain to create the URL from, so for the dev site it would be something likedev03-na01-example.demandware.net)
  - Site Id: your demandware store name (usually company name)

  - Auth token: create an Auth Token for this integration that will be used to request a bearer token

User Guide

Roles, Responsibilities

N/A

Business Manager

Please refer the Business Manager Section above for setup related information. There is no other Business Manager setup required other than the initial configuration stated in that section.

Storefront Functionality

The storefront integration will invoke the following events and tracking for the following functionality:

- Added to Cart – Event fired when user adds/modifies his/her cart
- Checkout Started – Event fired when user initiates the checkout process
- Order Confirmation – Event fired when the user places his/her order, useful for real-time receipts
- Site Search – Event fired when a user initiates a search.
- Viewed Category – Event fired when a user views a category page.
- Viewed Product – Event fired when a user views a product page. The information regarding the viewed product is passed along to Klaviyo here.

In addition to this, the add to list functionality is triggered when a customer subscribes to Newsletter via the footer.

User details also get passed to Klaviyo via the Track API when a user logs in or registers via the storefront.

Known Issues

N/A

Release History

Version	Date	Changes
01/01/00	22nd March 2019	Initial release
01/02/00	22th September 2019	Updated for OCAPI endpoints