



Unific Magento 2 to HubSpot Integration Technical Specifications

Summary

Unific's Magento 2 to HubSpot integration solution is comprised of these components:

- An extension called the “Unific Connector for Magento 2” available in the Magento Marketplace at <https://marketplace.magento.com/revenueconduitunific-connector.html>
- Unific - a Software as a Service Application with an API endpoint
- HubSpot's Ecommerce Bridge API

The Unific Connector extension creates webhooks inside of Magento 2. These webhooks are sent to Unific's servers where Unific will receive, validate and process them. Unific then sends the data contained in the webhook to HubSpot via the Ecommerce Bridge API.

How does the “Unific Connector for Magento 2” Extension process the Magento webhooks?

The Unific Connector uses Magento 2's Plugin (Interceptor) pattern (<https://devdocs.magento.com/guides/v2.3/extension-dev-guide/plugins.html>) to fire webhooks when certain actions happen within Magento. This is a best practice.

Asynchronous Webhooks

Webhooks are not immediately sent to Unific's server, rather they are queued up and sent in a highly controlled and asynchronous fashion. This allows Magento 2 administrators much needed control on how to moderate the speed by which data is sent.

The advantage of this method is it helps mitigate problems if website traffic spikes and CPU, memory or database loads start to climb quickly. A common example would be during a big sale or promotion where abandoned carts are created en masse.



Most websites are capable of handling these spikes, however our philosophy is that the Unific Connector should give as much control to Magento 2 administrators as possible since they have the most 'by the second' knowledge of the stores they are managing.

Asynchronous webhooks also allow for retry capabilities in the event of any network issues in the system. The Unific Connector will retry webhooks 3 times.

Using asynchronous webhooks also enables administrators to pause the connector from syncing if needed and the webhooks will still be available to be synced when the connector is resumed. The "Enable Connector" setting allows the administrator to "Pause" the sync at any time without having to uninstall the whole extension.

How Unific Provides More Control

As mentioned above, the Unific Connector doesn't just send the webhooks immediately. Instead it queues them up to a database table inside of Magento's local database.

The Unific Connector uses cron jobs that run once per minute that send the Live and Historical web hooks that were already queued up to be sent.

This allows the Unific Connector to spread the load on the Magento 2 host server's CPU and memory server over time instead of overloading it all at once - especially if there's a spike in activity on the Magento 2 site (ie. abandoned carts).

The following settings can be configured as desired:

- Enable Connector:
 - Yes/No setting that enables and disables the sending of webhooks respectively.
 - 'Yes' turns on or resumes the sync
 - 'No' pauses or turns off the sync
 - Default value is set to 'Yes'
- Message Batch Size:
 - Defines the maximum number of messages that are loaded into the cron job process at any one time.
 - Default value is set to 10
 - Allowable range is from 1 to 25
- Single Cron Run Message Limit:
 - Defines maximum number of messages that can be sent during one cron job run.
 - Defaulted value is set to 100
 - Allowable range is from 50 to 200

- Historical Message Batch Size:



- Defines maximum number of message that can be sent during one cron job run for historical sync
- Default value is set to 10
- Allowable range is from 1 to 25

- Single Cron Run Historical Message Limit
 - Defines maximum number of message that can be sent during one cron job run for historical sync
 - Default value is set to 100
 - Allowable ranges is from 50 to 200

Events that Unific Detects and Sends Corresponding Webhooks

- order/create
- order/update
- customer/create
- customer/update
- product/create
- product/update
- category/create
- category/update
- checkout/create
- checkout/update

How does Unific Backend Connect to Magento 2?

Connection during Install

Unific follows the OAUTH 2.0 protocol (<https://tools.ietf.org/html/rfc6749>). Magento 2 is the OAUTH Provider and Unific is the OAUTH Consumer.

After installing the Unific Connector Extension on the Magento 2 store, the user will need to log in to their Unific account fill in:

1. Magento 2 store URL
2. The consumer key
3. The consumer secret
4. The access token

These can be found inside of the Magento 2 “Integration” settings for Unific inside of Magento 2 Admin.



During the connection process, Unific will send Magento 2 an “Integration Identifier” value and generate an HMAC field inside of Magento 2. These are explained below:

1. Integration Identifier
 - a. The “Integration Identifier” is generated on install and is sent with every webhook to Unific.
 - b. Unific uses this value to validate whether the request is coming from a Magento 2 store that has been connected to Unific.
 - c. For example, if Unific cannot find a matching integration identifier within it’s systems, Unific will not process the webhook. If it finds the identifier, it will know to go forward with processing.
2. HMAC
 - a. Unific will only process Magento 2 webhooks that include a HMAC header (<https://www.wolfe.id.au/2012/10/20/what-is-hmac-authentication-and-why-is-it-useful/>).
 - b. The HMAC header is used to prove the authenticity and integrity of the webhook.
 - c. For example, it helps Unific prove that the webhook data was not altered after it was sent by Magento. Unific will deny the webhook if it does not pass the HMAC check.

Connection After Install

After the initial installation, Unific connects to Magento 2 via the Magento 2 REST API. There are several features, including Unific’s Dynamic Coupon Generator that uses this method of connecting to Magento 2.

To authenticate these requests, Unific uses the Consumer Key, Consumer Secret and Access Token that it saved from the connection process.

The Consumer Key, Consumer Secret and Access Token can be revoked at any time from the Magento 2 Admin.

How does the Unific Backend work?

Unific backend is an “Event Streaming Architecture” ([What is Event Streaming? | VMware Tanzu](#)). Unific was built with the goal of being “Fast, Reliable, Easy and Deep” and the Event Streaming architecture allows Unific to accomplish those goals.

Webhooks are a core requirement to achieve this and thus we have invested in building the Unific Connector Extension for Magento 2 to reliably send webhooks in a timely fashion. Webhooks also allow for easy recovery if there are any failures.



All cart webhooks require an HMAC signature which allows us to check their authenticity and integrity of webhooks.

Unific is monitored 24/7 with automated alerts being sent when issues are detected. It is scalable as all internal components can be scaled independently as usage changes.

Enable Connector Setting

Store View: Default Config ? Save Config

UNIFIC ^ **General Settings** ⌵

Connector Configuration Enable Connector [store view] Yes ▼

Webhook Processing Settings

Message Batch Size <small>[global]</small>	10 ▼	Maximum number of messages that are loaded into the cron job process at any one time. It is recommended to keep this low for servers with low CPU and RAM to prevent cron jobs from using more memory than available and crashing the cron job. Lower values mean a slower sync.
Single Cron Run Message Limit <small>[global]</small>	100 ▼	Maximum number of message that can be sent during one cron job run. It is recommended to keep this low for smaller servers to prevent single cron jobs from running too long. Lower values mean a slower sync.
Historical Message Batch Size <small>[global]</small>	10 ▼	Maximum number of message that can be sent during one cron job run. It is recommended to keep this low for smaller servers to prevent single cron jobs from running too long. Lower values mean a slower sync.
Single Cron Run Historical Message Limit <small>[global]</small>	100 ▼	Maximum number of message that can be sent during one cron job run. It is recommended to keep this low for smaller servers to prevent single cron jobs from running too long. Lower values mean a slower sync.