



Magento 2.x Integration Module  
Merchant and Developer Documentation - Version 1.1.0

**Prepared by:** Matthew Muscat  
**Last Updated:** 18th of July 2016 @ 1:30pm

# Table of Contents

## [Module Configuration](#)

### [Shippit](#)

#### [General Settings](#)

#### [Checkout Options](#)

#### [Order Sync](#)

#### [Shipping Sync](#)

### [Shipping Method](#)

#### [Shippit](#)

## [Custom Integrations](#)

### [Event Listener Method](#)

#### [Event Name](#)

#### [Description](#)

#### [Parameters](#)

#### [Example Code](#)

### [Model API Method](#)

#### [Model Class](#)

#### [Function](#)

#### [Description](#)

#### [Parameters](#)

#### [Example Code](#)

## *References*

- “The Platform” - refers to the Shippit Platform, located at Shippit.com
- “Magento” - refers to the Magento eCommerce Platform, version 2.x

# Module Configuration

## Shippit

The Shippit Module Configuration sections enables you to set the general settings of the overall module, as well as your preferences on how you would like the module to operate in the relevant feature areas.

The Shippit Module Configuration is split into 3 sections, each representing a certain area of the store or functionality covered by the Shippit module.

The screenshot shows the 'Configuration' page for the Shippit module. The left sidebar contains navigation menus for 'SALES', 'CUSTOMERS', 'MARKETING', 'CONTENT', 'REPORTS', 'STORES', 'SYSTEM', and 'FIND PARTNERS & EXTENSIONS'. The 'SALES' menu is expanded, showing 'Sales', 'Sales Emails', 'PDF Print-outs', 'Tax', 'Checkout', 'Shipping Settings', 'Multishipping Settings', 'Shipping Methods', 'Shippit', 'Google API', and 'Payment Methods'. The 'Shippit' option is selected and highlighted.

The main content area is titled 'Configuration' and includes a search icon, a notification bell, and a user profile 'matthew'. Below this is a 'Store View' dropdown set to 'Default Config' and a 'Save Config' button.

The configuration is divided into three sections:

- General Settings:** Includes 'Enabled' (Yes), 'API Key' (## API KEY HERE ##), 'Environment' (Sandbox), and 'Debug Mode' (Yes).
- Order Sync:** Includes 'Enabled' (Yes), 'Mode' (Scheduled), 'Send All Orders to Shippit' (Yes - All Orders), 'Send Product Location to Shippit' (No), and 'Product Location Attribute Code' (No). A note states: 'The Product Attribute to be used for location information on Shippit Pick Slips'.
- Shipping Method Mapping:** A table mapping Magento shipping methods to Shippit service classes. The table has columns for 'Shipping Method', 'Shippit Service Class', and 'Action'. It lists '[freeshipping] Free', '[tablerate] Table Rate', and '[flatrate] Fixed', each with an 'Expr' dropdown and a trash icon. An 'Add' button is at the bottom.

Below the 'Shipping Method Mapping' section is a note: 'A mapping of the Magento Shipping Methods to Shippit Services classes'. The final section is 'Shipping Sync', which includes 'Enabled' (Yes).

## General Settings

The “General Settings” section covers the global Shippit Module configuration options - that is, any options set here take effect on all other sections as well.

- **Enabled**

Allows you to enable or disable Shippit functionality in your store. Note, disabling this area will also disable all Shippit module functionality, including Checkout Options, Order Sync, Shipping Sync and Live Quoting.

- **API Key**

The API Key to be used for all Shippit functionality - this API Key is used for Live Quoting, Order Sync and Shipping Sync. When saving the Shippit Configuration.

Note: The module will check to ensure that the API Key entered is valid. A message is shown in the notification area indicating a success or failure.

- **Environment**

The environment the the module should communicate with - valid options include “Live” and “Sandbox”. The Sandbox environment is available for testing the module and it’s functionality with your store.

- **Debug Mode**

A flag enabling Shippit and your development team to view the module operations. All logging activity is captured and provided in a “shippit.log” file in the “/var/logs” directory.

Shippit will receive a copy of the logging activity via the Bugsnag integration built in to the module

- If set to “Yes”, the module will capture all module activity, include informational and debug messages
- If set to “No”, the module will only capture error activity

## Order Sync

The “Order Sync” section covers the options on how Shippit will display certain items during Checkout.

Order Sync functionality enables the module to communicate with the Shippit platform to automatically receive details on the order, shipping method selection and it’s items.

- **Enabled**

Allows you to enable or disable Order Sync functionality.

- **Mode**

Allows you to specify how Orders should be synced with Shippit. Available options include...

- **Realtime**

Orders are synced with Shippit as soon as they are received

- **Scheduled**

Orders are synced with Shippit in the background using Magento's cron scheduler - if an order has a priority option selected, the order is synced immediately

- **Custom**

For Custom Solutions - Orders are not automatically added to the Shippit Order queue when received - used when custom logic or business rules are required to determine when an order can be synced with Shippit

Some Important Notes...

- If a Realtime sync event fails, the system will process this on the next scheduled run
- For any orders that use a "Priority" shipping service, they will be synced immediately, regardless of the "Realtime" or "Scheduled" setting
  - If the sync mode is "Custom", the order will not be synced immediately
  - If the order is not yet "processing", the order will not be synced immediately
- Shippit automatically schedules an order syncs for all pending orders every 5 minutes - only orders that are with a status of "pending" and relate to an order in the state of "processing" is processed by the schedule.
- All order syncs are attempted upto a maximum of 5 times, upon which they are transitioned to a status of "failed"
- The "Shippit Order Sync" page can be used to re-attempt syncs with items - re-attempting a scheduled sync will reset the attempt count to 0.

- **Send All Orders to Shippit**

Allows you to Send All Orders to Shippit, regardless of the Shipping Method selected during checkout. Available options include...

- **Yes - All Orders**

All orders, regardless of their destination are synced with Shippit.

- **Yes - All Australian Orders**

All orders that have a shipping address destination for "Australia" are synced with Shippit.

- **No**

Only orders that utilise a Shippit Shipping Method, or use a mapped Shipping Method are synced with Shippit.

- **Send Product Locations To Shippit**

Allows you to send a product location to Shippit using a Magento Attribute Value on the product in the order. The Product Location is displayed on Shippit's Pack Slips.

- **Product Location Attribute Code**

The Product Attribute to be used for the Product Location.

- **Shipping Method Mapping**

Enables you to map other Shipping Methods with a Shippit Shipping Method. Useful when you want to provide customers with options outside of live quoting, for example - Flat Rate Shipping or Free Shipping. You can select the Magento Shipping Method and map this with a Shippit Service Class.

## Shipping Sync

The “Shipping Sync” section covers the options in regards to the Shipping Updates received by your Store from Shippit.

The module will update orders as Shipped automatically as you transition the order in Shippit’s Systems. A order is marked as Shipped soon after a booking has been made in the Shippit Platform and the booking has reached the status of “Ready for Pickup”.

- **Enabled**

Allows you to enable or disable Shipping Sync functionality.

# Shipping Method

The Shippit Module adds a new “Shippit” shipping method in the Magento “Shipping Methods” configuration. This shipping method communicates directly with Shippit to provide live quoting based on the contents of the cart during a customer’s visit to your store.

The screenshot shows the Magento Configuration interface for the Shipping Methods section. The left sidebar contains a navigation menu with categories: GENERAL, CATALOG, CUSTOMERS, SALES (expanded), SERVICES, and ADVANCED. Under SALES, the following options are listed: Sales, Sales Emails, PDF Print-outs, Tax, Checkout, Shipping Settings, Multishipping Settings, Shipping Methods (highlighted), Google API, and Payment Methods. The main content area is titled "Configuration" and shows the "Shippit" shipping method configuration. The "Store View" is set to "Default Config". A "Save Config" button is visible in the top right. The configuration fields are as follows:

- Enabled** [website]: Yes
- Title** [store view]: Shippit
- Allowed Methods** [website]: Standard, Express, Priority
- Maximum Timeslots** [website]: -- No Max Timeslots --
- Filter by Enabled Products** [website]: No
- Filter by Product Attribute** [website]: No
- Ship to Applicable Countries** [website]: All Allowed Countries
- Ship to Specific Countries** [website]: Afghanistan, Åland Islands, Albania, Algeria, American Samoa, Andorra, Angola, Anguilla, Antarctica, Antigua and Barbuda
- Displayed Error Message** [store view]: This shipping method is not available. To use this shipping method, please contact us.
- Sort Order** [website]: 10

## Shippit

- **Enabled**

Allows you to enable or disable Shippit live quoting in your store. Note, disabling this area will prevent the option of utilising a “priority” delivery method, as this requires live quoting to determine the service availability.

- **Title**

Allows you to customise the display of the Shipping Options available via Shippit’s Live Quoting.

- **Allowed Methods**

Allows you to specify the methods that should be displayed to customers when retrieving live quotes.

- **Maximum Timeslots**

Allows you to specify the maximum amount of time slots displayed to customers for Priority Delivery Services. Priority Delivery Services will display an individual Shipping Option for every date / timeframe available.

- **Filter by Enabled Products**

Allows you to selectively provide live quoting when the cart contains only certain items.

- **Enabled Products**

The products that are available for live quoting

- **Filter by Product Attributes**

Allows you to selectively provide live quoting for selected items based on a product attribute matching a specified value

- **Product Attribute Code**

The product attribute code to use for the filtering

- **Product Attribute Value**

The product attribute value that the product should match. Note: The \* character enables you to specify wildcard matching.

- **Ship to Applicable Countries**

Allows you to selectively provide quotes for specific countries. *Note: Currently, Shippit is only able to provide services to Australia*



- **Ship to Specific Countries**

The countries that live quoting should be active for. *Note: Currently, Shippit is only able to provide services to Australia*

- **Show Method if Not Applicable**

Allows you to specify if the Shipping option should be shown if there are no methods available.

- **Displayed Error Message**

Indicates the message to be displayed to customers when this shipping method is unavailable. Note: This option is only shown if the above value is set to “Yes”.

- **Sort Order**

Allows you to specify the display sort order in relation to other Shipping Methods in the Magento Installation.

## Custom Integrations

The Shippit Module custom integration feature enables merchants to implement custom logic in the Magento eCommerce Platform as to when an order should be synced with the Shippit Platform.

This functionality enables merchants and developers the flexibility to determine at what point the order should be synced with the Shippit Platform and determine what items and quantities should be sent to the Shippit Platform.

The custom integrations functionality has been built to provide you with a simple, efficient method of integrating with Shippit, without requiring the knowledge or expertise in handling API requests to Shippit’s Systems.

We only require that you provide us with some basic details on what order you would like to sync, along with some custom options on specific items or shipping method, upon which the module will handle building the API request and adding it to the module’s request queue.

Shippit provide two options for custom solutions, both options are similar in the arguments structure and will utilise the core module features, such as validation, queuing and handling the API request.

1. **Event Listener Method**

An event called `shippit\_add\_order` is listener to by the Shippit Module - this enable you easily call this event and pass over basic details.

2. **Model API Method**

Similar to the module event listener, a direct Model API method is available. This takes the same request format to the event listener, but enables you to retrieve the response of your request.

## Event Listener Method

An event called `shippit\_add\_order` is listener to by the Shippit Module - this enable you easily call this event and pass over basic details.

Details of the event, it's available parameters and example code are provided below.

### Event Name

`shippit_add_order`

### Description

Enables merchants to trigger the event and pass order details to send an order to the Shippit Platform.

### Parameters

Parameter Key	Type	Required	Description
<code>order</code>	<code>string object</code>	Yes	The Magento Order Entity ID or Order Object
<code>items</code>	<code>array</code>	No	An array of the items to be sent to the Shippit Platform, if this value is empty, all items pending shipment are sent to Shippit
<code>item[x]</code>	<code>array</code>		An array of the item details
<code>item[x][sku]</code>	<code>string</code>	Yes	The product sku <b>Notes:</b> — <i>either id or sku is required</i>
<code>item[x][qty]</code>	<code>integer</code>	No	The quantity to be sent to Shippit <b>Notes:</b> <ul style="list-style-type: none"><li>- <i>If no value is provided, this defaults to the amount pending delivery according to Magento</i></li><li>- <i>The value sent to Shippit will be the maximum amount that is pending delivery according to Magento, that is, if you provide a value of "10" for an item that has "5"</i></li></ul>

			<i>pending delivery, "5" will be sent to Shippit</i>
	item[x][location]	string	No For Custom Solutions - the warehouse location
shipping_method	string	No	<p>The Shipping Method to be utilised for the request - the passed value can be either a Shippit Service Class, or the order shipping method.</p> <p>If a value is not provided, or an invalid value is provided, this defaults to the "Standard" Shippit Service Class.</p> <p>Valid options include</p> <ul style="list-style-type: none"> <li>- <b>`standard`</b></li> <li>- <b>`express`</b></li> <li>- <b>`priority`</b></li> <li>- <b>Order shipping method - uses mapping configuration</b></li> </ul> <p>Notes:</p> <ul style="list-style-type: none"> <li>- <b>`priority`</b> services are only available when an order has a live quoting shipping method attached to the order - this is due to requirements on checking priority service availability dates + timeslots</li> </ul>
api_key	string	No	The API Key to be utilised for the request - if the value is not provided, this defaults to the configured api_key value in the Magento Admin
sync_mode	string	No	<p>The Sync mode to utilise, valid options include</p> <ul style="list-style-type: none"> <li>- <b>`realtime`</b> Indicates the request should be made immediately, that is, the request will run immediately</li> <li>- <b>`scheduled`</b> Indicates the request should be added to the queue, that is, the request will run in the next sync run (every 5 minutes)</li> </ul> <p>Notes:</p> <ul style="list-style-type: none"> <li>- The default value of "scheduled" is used if no value is provided</li> <li>- <b>`Realtime`</b> sync mode is only recommended for</li> </ul>

			immediate actions, bulk actions should utilise the `scheduled` mode
display_notifications	boolean	No	A flag indicating if notifications should be shown in the Magento interface - useful if you are running manual actions and require visual confirmation in the Magento Admin - if no value is provided, defaults to false.

## Example Code

```
// Sends an individual order to Shippit
$this->_eventManager->dispatch(
    'shippit_add_order',
    array(
        'order' => 4538, // Required
        'items' => array( // Optional - defaults to all items pending shipment if not passed
            array(
                'sku' => 'sku-1', // Optional - either id or sku is required
                'qty' => '1', // Optional - defaults to items remaining to be shipped on order
                'weight' => '1', // Optional - defaults to item weight calculation based on qty
                'location' => 'M1' // Optional - used for custom warehousing solutions
            )
        ),
        'shipping_method' => 'express', // Optional - defaults to standard if no value is passed
        'sync_mode' => 'realtime' // Optional - defaults to module configuration value
    )
);
```

## Model API Method

A model api available under Shippit\_Shippit\_Model\_Shippit enables you to add orders to the request queue by simply calling the “addOrder” method and passing over basic details.

Details of the model api, it’s available parameters and example code are provided below.

## Model Class

\Shippit\Shipping\Model\Shippit

## Model Interface

\Shippit\Shipping\Api\Request\ShippitInterface

## Function

addOrder

## Description

Enables merchants to call the model function and pass order details to send an order to the Shippit Platform.

## Parameters

Parameter Key	Type	Required	Description
order	string object	Yes	The Magento Order Entity ID or Order Object
items	array	No	An array of the items to be sent to the Shippit Platform, if this value is empty, all items pending shipment are sent to Shippit
item[x]	array		An array of the item details
item[x][sku]	string	Yes	The product sku <i>Notes:</i> — <i>either id or sku is required</i>
item[x][qty]	integer	No	The quantity to be sent to Shippit <i>Notes:</i> <ul style="list-style-type: none"><li>- If no value is provided, this defaults to the amount pending delivery according to Magento</li><li>- The value sent to Shippit will be the maximum amount that is pending delivery according to Magento, that is, if you provide a value of “10” for an item that has “5” pending delivery, “5” will be sent to Shippit</li></ul>

	item[x][location]	string	No	For Custom Solutions - the warehouse location
shippingMethod		string	No	<p>The Shipping Method to be utilised for the request - the passed value can be either a Shippit Service Class, or the order shipping method.</p> <p>If a value is not provided, or an invalid value is provided, this defaults to the “Standard” Shippit Service Class.</p> <p>Valid options include</p> <ul style="list-style-type: none"> <li>- <b>`standard`</b></li> <li>- <b>`express`</b></li> <li>- <b>`priority`</b></li> <li>- <b>Order shipping method - uses mapping configuration</b></li> </ul> <p>Notes:</p> <ul style="list-style-type: none"> <li>- The default value of “scheduled” is used if no value is provided</li> <li>- <b>`priority`</b> services are only available when an order has a live quoting shipping method attached to the order - this is due to requirements on checking priority service availability dates + timeslots</li> </ul>
apiKey		string	No	The API Key to be utilised for the request - if the value is not provided, this defaults to the configured api_key value in the Magento Admin
syncMode		string	No	<p>The Sync mode to utilise, valid options include</p> <ul style="list-style-type: none"> <li>- <b>`realtime`</b> Indicates the request should be made immediately, that is, the request will run immediately</li> <li>- <b>`scheduled`</b> Indicates the request should be added to the queue, that is, the request will run in the next sync run (every 5 minutes)</li> </ul> <p>Notes:</p> <ul style="list-style-type: none"> <li>- <b>`Realtime`</b> sync mode is only recommended for immediate actions, bulk actions should utilise the <b>`scheduled`</b> mode</li> </ul>

displayNotifications	boolean	No	A flag indicating if notifications should be shown in the Magento interface - useful if you are running manual actions and require visual confirmation in the Magento Admin - if no value is provided, defaults to false.
----------------------	---------	----	---

## Example Code

```
// Sends an individual order to Shippit
$this->_shippit
    ->addOrder(
        4538, // Order - Required
        array( // Items - Optional - defaults to all items pending shipment if not passed
            array(
                'sku' => 'sku-1', // Optional - either id or sku is required
                'qty' => '1', // Optional - defaults to items remaining to be shipped on order
                'weight' => '1', // Optional - defaults to item weight calculation based on qty
                'location' => 'M1' // Optional - used for custom warehousing solutions
            )
        ),
        'express', // ShippingMethod - optional - defaults to standard if none provided
        'api_key', // ApiKey - optional - defaults to configured api key in the Magento Module
        'realtime' // SyncMode - Optional - defaults to module configuration value
        True // DisplayNotifications - Optional - defaults to false
    )
);
```