# 

# SFCC Dispatch Cartridge Installation Guide

23.7.0

## Table Of Contents

[SFCC Dispatch Cartridge Installation Guide 1](#_Toc143090093)

[Table Of Contents 2](#_Toc143090094)

[About Dispatch 4](#_Toc143090095)

[1. Implementation Overview 4](#_Toc143090096)

[2. Component Overview 5](#_Toc143090097)

[2.1 Supported Features 5](#_Toc143090098)

[2.1.1 Sync Products 5](#_Toc143090099)

[2.1.2 Payment Hook 5](#_Toc143090100)

[2.1.3 Download Settings 5](#_Toc143090101)

[2.1.4 Test Connection 5](#_Toc143090102)

[2.2 Limitations and Constraints 5](#_Toc143090103)

[2.3 Compatibility 5](#_Toc143090104)

[2.4 Privacy 6](#_Toc143090105)

[3. What is SCAPI 6](#_Toc143090106)

[3.1 Overview 6](#_Toc143090107)

[3.2 How Is SCAPI used? 6](#_Toc143090108)

[4. Implementation Guide 6](#_Toc143090109)

[4.1 Overview 6](#_Toc143090110)

[4.2 Installation 7](#_Toc143090111)

[4.2.1 Import metadata archive 7](#_Toc143090112)

[4.2.2 Upload Dispatch Cartridges 7](#_Toc143090113)

[4.2.3 Add Dispatch Cartridges to Path 8](#_Toc143090114)

[4.2.4 Dispatch Business Manager module installation 8](#_Toc143090115)

[4.3 Salesforce Commerce API Setup 9](#_Toc143090116)

[4.3.1 Setup SCAPI Admin API Access 9](#_Toc143090117)

[4.3.2 Setup SCAPI Shopper API Access 11](#_Toc143090118)

[4.3.3 Add API Credentials to Dispatch Services 11](#_Toc143090119)

[4.4 Configuration 12](#_Toc143090120)

[4.4.1 Update Dispatch Settings in Business Manager extension 12](#_Toc143090121)

[4.4.2 Configuring Dispatch Cartridge for Adyen 13](#_Toc143090122)

[4.4.3 Configuring Dispatch Cartridge for Stripe 13](#_Toc143090123)

[4.5 Testing 15](#_Toc143090124)

[5 Operations and Maintenance 15](#_Toc143090125)

[5.1 HTTP Service Availability 15](#_Toc143090126)

[5.2 Support 15](#_Toc143090127)

[5.3 Billing 16](#_Toc143090128)

[6 User Guide 16](#_Toc143090129)

[6.1 Roles and Responsibilities 16](#_Toc143090130)

[Refer to Business Manager extension setup instructions. 16](#_Toc143090131)

[6.2 Business Manager 16](#_Toc143090132)

[7 Release History 16](#_Toc143090133)

# About Dispatch

Dispatch makes content and media shoppable. We enable merchants using Salesforce Commerce Cloud B2C to transform traditionally marketing-only channels into sales channels. It’s a multi-platform “buy now” button that lets customers buy your products at the moment of inspiration.

Install Dispatch and select a product catalog from which you would like to sell products. Your existing marketing content in these channels becomes shoppable with an embedded checkout from Dispatch. Reach new customers and offer a seamless shopping experience across the internet, from discovery to checkout, without managing new commerce logistics.

Your implementation will allow your business to start selling in new channels with Dispatch, while having orders and payments processed through Salesforce Commerce Cloud B2C, just like your ecommerce website.

## Implementation Overview

This implementation guide describes how to integrate **bm\_dispatch**, **int\_dispatch,** **int\_dispatch\_sfra** and **int\_dispatch\_sg** cartridges version 23.7.0 into your SFCC site.

Cartridges includes:

* Dispatch Settings Page
* Functionality to instantly sync settings / catalog with Dispatch
* Payment authorization via SCAPI through Stripe and Adyen
* Download Settings with basic authentication
* Methods for testing

Dispatch uses SCAPI Basket, Order, and Payment API to place an order in your SFCC instance. To get product and catalog data Dispatch uses [SCAPI Admin](https://developer.salesforce.com/docs/commerce/commerce-api/references/about-commerce-api/about.html).

The publishable / client key provided by your payment service provider is used by Dispatch to tokenize and encrypt credit card info using the payment service provider’s tokenization libraries, so Dispatch is at no point handling, storing, or transferring raw credit card data.

Separately, Dispatch will issue you a private and public keypair (**Dispatch API Key** and **Dispatch Account ID**) used to seamlessly sync and test your configuration with Dispatch. Contact [support@dispatch.co](mailto:support@dispatch.co) to obtain these keys when you are ready to test your integration.

## Component Overview

### 2.1 Supported Features

#### 2.1.1 Sync Products

Through Dispatch Settings, merchants can select a product catalog to enable Dispatch embedded checkouts for. Your chosen catalog must be assigned to a site. Pressing the sync settings button will trigger Dispatch to sync the latest products in your selected catalog as well as your Dispatch Settings. Dispatch also syncs your catalog daily to ensure the latest information is available. You need to provide your Dispatch API Key / AccountId obtained from [support@dispatch.co](mailto:support@dispatch.co) to successfully sync your settings.

#### 2.1.2 Payment Hook

The Dispatch cartridge comes with a hook, custom code, and instructions to enable payments with Stripe or Adyen through SCAPI. This custom code is required as payment processors do not enable payment capture and authorization via SCAPI by default. Dispatch’s custom code offers a secure workaround method supported by payment processor APIs.

#### 2.1.3 Download Settings

Basic authentication secured endpoint that returns return settings. Secured by Dispatch API key.

curl --location 'https://{{REALM\_ID}}.commercecloud.salesforce.com/on/demandware.store/Sites-RefArch-Site/en\_US/Dispatch-Settings' \

--header 'Content-Type: application/x-www-form-urlencoded' \

--data-urlencode 'dispatchAPIKey={{API\_KEY}}

#### 2.1.4 Test Connection

Validate if connection settings are working as expected. ￼

### 2.2 Limitations and Constraints

At this time Dispatch can only be used by merchants who use Stripe or Adyen for payment processing. If you use another payment processor, please contact us at [support@dispatch.co](mailto:support@dispatch.co) to request support for your payment processor.

### 2.3 Compatibility

Tested against SFRA

* SFRA - v6.1.0
* SCAPI - v1
* Compatibility Mode: 22.7
* Stripe Cartridge - v23.3.0
* Adyen Cartridge - v23.2.0

Tested against SiteGenesis

* + SiteGenesis - v105.2.1
* SCAPI - v1
* Compatibility Mode: 22.7
* Stripe Cartridge - v23.3.0
* Adyen Cartridge - v22.2.2

### 2.4 Privacy

* This integration requires access to the following customer data elements: Email Address, Shipping Address, Order Details, Customer Email and Name.
* The only personal data stored by Dispatch is the customer’s email address as an identifier, along with their IP address.
* Dispatch is GDPR Compliant and will comply with data requests made by your business or the data subject.

## 3. What is SCAPI

### 3.1 Overview

The Salesforce Commerce API (SCAPI) is an API that allows other applications to securely access the platform’s resources externally from the platform itself. An API, or Application Programming interface, is the preferred method for two applications to communicate in real-time. Simply stated, an API allows a server to return or accept information that is requested or sent by another application. The calling application can then interpret the received data and present it to the user.

### 3.2 How Is SCAPI used?

The SCAPI interface grants other applications access to Salesforce Commerce Cloud resources such as baskets, customers, orders, and products. While these resources are usually consumed through the client site storefront that customers see, the interface provides access to those resources for other applications to read, create, or update through service calls.

The general information about SCAPI you can find here: [SCAPI documentation](https://developer.salesforce.com/docs/commerce/commerce-api/overview).

## 4. Implementation Guide

### 4.1 Overview

4 cartridges support this integration:

* 1. **int\_dispatch** – Base cartridges which include helper methods and services framework code
  2. **int\_dispatch\_sfra** – Enables Dispatch to connect to SFRA based SFCC merchant store
  3. **int\_dispatch\_sg** – Enables Dispatch to connect to SiteGenesis based SFCC merchant store
  4. **bm\_dispatch** – Business Manager extension that helps you manage your Dispatch integration

### Installation

#### Import metadata archive

1. Zip the content of metadata folder (site-import-dispatch) into a file named site-import-dispatch.zip

2. Go to **Business Manager > Administration > Site Development > Site Import & Export.**

3.Upload archive using Local option in the Upload Archive section.

4. After uploading, choose site-import-dispatch.zip in the list and click on import button.

A screenshot of a computer

Description automatically generated

**Figure 1. site-import-dispatch.zip archive import**

#### Upload Dispatch Cartridges

1. Go to the [Dispatch SFCC B2C Cartridge Github Repository](https://github.com/iex-xyz/dispatch-sfcc-b2c-cartridge) and clone down into your local SFRA folder.   
2. Rename the cloned repo/folder from step 1 to “link\_dispatch”. (see example folder structure)

A black background with white text

Description automatically generated

3. Upload the “link\_dispatch” cartridge to your instance via cli (npm run uploadCartridge) or use 3rd party tool.

4. Promote the uploaded cartridge using the version control system in **Administration > Site Development > Code Deployment**. See screenshot below.

A screenshot of a computer

Description automatically generated

#### Add Dispatch Cartridges to Path

1. Go to **Business Manager > Administration > Sites > Manage Sites > Your Target Site > Settings**

2. Append “int\_dispatch\_sfra:int\_dispatch” to cartridges list. See example figure 2 below.

3. Click apply.

A screenshot of a computer

Description automatically generated

**Figure 2. storefront cartridge path**

#### Dispatch Business Manager module installation

1. Go to **Business Manager > Administration > Sites > Manage Sites > Business Manager Site.**

2. Append “bm\_dispatch:int\_dispatch” to cartridges list. See figure 3 below.

A screenshot of a computer

Description automatically generated

**Figure 3. Business Manager Cartridge Path**

3. Click apply.

4. Grant access to Business Manager module:

4a. Go to **Business Manager > Administration > Organization > Roles & Permissions** and select the target role(s) that needs to have access to Dispatch Settings.

4b. Select the **Business Manager Modules** tab in the drop-down list on top. You must select your target site as the active context.

4c. Scroll to the Dispatch Settings module. Grant access to it by selecting the checkbox and clicking Update.

Once you do this, every Business Manager user with that role can manage Dispatch Settings by selecting **Business Manager > Merchant Tools > Dispatch Plugin > Dispatch Settings.   
  
Note: This page may throw an error until you have completed all steps in section 4.2 & 4.3.**

### Salesforce Commerce API Setup

#### Setup SCAPI Admin API Access

Dispatch service requires access to SCAPI Admin credentials. Create API client with the following allowed scopes in the [Account Manager](https://account.demandware.com/). See Figure 8 below

A screenshot of a computer

Description automatically generated

**Figure 8. Account Manager**

Settings To Change:

**Display Name:** Chosen name for API Client key

**Password:** Chosen password for client

**Roles:**

Sandbox API User -> Select all desired instances

Salesforce Commerce API -> Select all desired instances

A screenshot of a computer

Description automatically generated

**Default Scopes**

mail

roles

tenantFilter

profile

openId

**Allowed Scopes**

sfcc.catalogs.rw

sfcc.catalogs

sfcc.products.rw

sfcc.products

sfcc.orders

sfcc.shopper-products

**Token Endpoint Auth Method:** client\_secret\_post

For more information on how to create API client for SCAPI Admin API [check documentation](https://developer.salesforce.com/docs/commerce/commerce-api/guide/authorization-for-admin-apis.html)

#### Setup SCAPI Shopper API Access

1. In new browser tab, go to https://<Short Code>.api.commercecloud.salesforce.com/shopper/auth-admin/v1/ui
2. Login and then go to **Clients > Add Client** (See figure 10)
3. Copy the Copy the API Client ID created for SCAPI Admin API from section 4.3.1 and set it as the **Client Id**.   
   **NOTE:** If it is not the same Client ID for both SCAPI Admin and Shopper APIs, the integration will not work.
4. Once created, enter edit mode for the new client
5. **For Production and Development instances you MUST** configure the redirect URL which matches your alias (see [docs](https://developer.salesforce.com/docs/commerce/commerce-api/guide/authorization-for-shopper-apis.html)) which matches this format:
   1. https://{{REALM\_ID}}.dx.commercecloud.salesforce.com/on/demandware.store/Sites-{{SITE\_NAME}}-Site/default/SLASCallback-RetrieveCode
6. Save changes via Submit button.

A screenshot of a computer

Description automatically generated

**Figure 10. SLAS Admin UI**

#### Add API Credentials to Dispatch Services

To access service credentials, go to **Business Manager > Administration > Operations > Services > Credentials** (See Figure 4)

A screenshot of a computer

Description automatically generated

**Figure 4. Services**

**Dispatch Services**:

* **dispatch.http.scapi** – is used for fetching SFCC data via SCAPI
* **dispatch.http.slasauth** – is used for authenticating and fetching access token
* **dispatch.http.service** – is used or calling Dispatch hook
* **dispatch.http.testservice** – is used for calling the Dispatch test settings endpoint

Update the following services as such:

* **DispatchSCAPIHost**

**URL:** https://<Short Code>.api.commercecloud.salesforce.com

**User:** <OrganizationID> (refer to Figure 5) 

* + - * **DispatchOAuth**

**User:** SCAPI Admin Client ID (generated in section 4.3.1)

**Pass:** SCAPI Admin Password (generated in section 4.3.1)

A screenshot of a computer

Description automatically generated

**Figure 5. Short Code**

### Configuring Dispatch for Payments

#### Update Dispatch Settings in Business Manager extension

Go to **Business Manager > Merchant Tools > Dispatch Plugin > Dispatch Settings**

A screenshot of a computer

Description automatically generated

**Figure 6. Dispatch Settings**

Set the following fields (see figure 6 for reference)

**Dispatch AccountID (Public Key):** obtained previously from [support@dispatch.co](mailto:support@dispatch.co) **Dispatch API Key (Private Key):** obtained previously from [support@dispatch.co](mailto:support@dispatch.co) **Organization ID:** see Figure 6 above

**Site ID:** The desired site of install

**Settings URL:** The URL on which you configure your Dispatch Settings (e.g. https://${RealmID}.dx.commercecloud.salesforce.com/on/demandware.store/Sites-RefArch-Site/en\_US/Dispatch-Settings). **Note**: that when configuring for your production / development instances, you *must* use alias values per 2022 Demandware.net Origin Shielding (see [article](https://medium.com/salesforce-commerce-cloud-tales/setup-storefront-hostname-in-salesforce-commerce-cloud-4b4a1732b40b))

**Catalog ID:** Which catalog to allow Dispatch to pull products from. **The chosen catalog must be assigned to a site.**

**Preferred Payment Method**: The payment method to the use for checkout

**Payment Method Publishable/Client Key:** The public key of your payment account. Usually generated/obtained from chosen providers site e.g. Stripe/Adyen

1. Ensure that the preferred payment method is active and fully functional.
2. Click the “Save & Sync Settings” button after completing all required fields. Clicking this button both updates your Settings and kicks off the Catalog Sync job. **All fields must be valid your account can be marked as active, and any changes to those fields resulting in an invalid state will make your account as inactive.**

#### Configuring Dispatch Cartridge for Adyen

Dispatch support for Adyen comes preconfigured just ensure Adyen is fully functional and enabled for SFCC site.

#### Configuring Dispatch Cartridge for Stripe

Dispatch support Stripe but some changes are required. We need to disable authorizeCreditCard hook that comes with Dispatch cartridge and make changes in authorizeCreditCard that comes with Stripe Cartridge.

1. rename **package.json** to **package-backup.json** located under **link\_dispatch > cartridges > int\_dispatch**, This will ensure that authorizeCreditCard hook that comes with Dispatch cartridge is disabled.
2. Next, we are going to make changes in the Stripe cartridge. To do so, open **authorizeCSC.js** located under **int\_stripe\_core > cartridge > scripts > hooks.** Select from line 62 to 84 (shown in Figure 7) and replace with the following code. (See [Full Example](https://github.com/iex-xyz/dispatch-sfcc-b2c-cartridge/blob/main/documentation/examples/stripe-sample-authorizeCSC.js))

*//*

*// Dispatch specific code.*

*//*

var paymentMethod = stripe.paymentMethods.create({

type: 'card',

card: {

token: paymentInstrument.creditCardToken

},

billing\_details: billingDetails

});

var paymentIntent = stripe.paymentIntents.create({

amount: orderAmount,

currency: amount.currencyCode.toLowerCase(),

payment\_method: paymentMethod.id,

description: 'Dispatch transaction',

metadata: {

order\_id: order.orderNo,

site\_id: Site.getCurrent().getID(),

source: "Dispatch - SFCC - Stripe Cartridge"

},

confirm: true

});

**Figure 7. authorizeCSC.js**

You also need to edit each “error” block, so Dispatch’s backend is notified when a payment instantly fails.

if (

paymentIntent.status === 'succeeded'

|| (paymentIntent.status === 'requires\_capture' && !stripeChargeCapture)

) {

Transaction.wrap(function () {

if (paymentIntent.id) {

paymentInstrument

.getPaymentTransaction()

.setTransactionID(paymentIntent.id);

}

if (paymentIntent.status === 'succeeded') {

order.setPaymentStatus(Order.PAYMENT\_STATUS\_PAID);

paymentInstrument

.getPaymentTransaction()

.setType(PaymentTransaction.TYPE\_CAPTURE);

} else {

paymentInstrument

.getPaymentTransaction()

.setType(PaymentTransaction.TYPE\_AUTH);

}

});

} else {

order.custom.Dispatch\_paymentErrorMessage = 'Transaction authorization was not successful';

throw new Error('Transaction authorization was not successful');

}

} catch (e) {

var m = e.message;

if (e.callResult) {

var o = JSON.parse(e.callResult.errorMessage);

m = o.error.message;

}

Logger.error('Error: {0}', e.message);

order.custom.Dispatch\_paymentErrorMessage = 'Transaction authorization was not successful';

return new Status(Status.ERROR, m);

}

Be sure you have Stripe Webhooks properly configured per <https://github.com/stripe/stripe-sfcc-b2c-connector/tree/master/documentation>

### Testing

* Dispatch can provide you a sandbox for testing purposes. Please contact Dispatch at [support@disptach.co](mailto:support@disptach.co) for access
* To test services credentials, and Dispatch API Key. Go to **Business Manager > Merchant Tools > Dispatch Plugin > Dispatch Settings**

A screenshot of a computer

Description automatically generated

**Figure 11. Test Settings/Connection**

“Test Settings” will run the following tests:

1. Dispatch Settings Test. Checks if Dispatch Server can download settings and pull catalog data.
2. Dispatch SLAS Shopper Login Test. Checks if Dispatch can login as a guest shopper.

## Operations and Maintenance

## HTTP Service Availability

You can track availability and downtime by service status in the Commerce Cloud Business Manager. Go to **Administration > Operations > Service Status > dispatch.http.service**

### Support

If you require assistance on integration or with anything else, please contact us at [support@dispatch.co](mailto:support@dispatch.co)

### Billing

Dispatch invoices your business monthly for commission fees earned during the billing period. Invoices are issued 14 days after the billing period. Commission fees are calculated as a % fee of the GMV generated through the Dispatch SDK. Commission fees are only collected for qualified orders. For an order to be qualified, the payment must be successful. Dispatch will exclude any orders that have been refunded or cancelled.

## User Guide

### 6.1 Roles and Responsibilities

### Refer to Business Manager extension setup instructions.

### 6.2 Business Manager

Refer to installation instructions.

## Release History

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Changes** |
| 23.7.0 | 2023.08.14 | Initial release |