COVA™

CANNABIS RETAIL SOFTWARE

Cova Pay Canada

Integrating your ecommerce solution

2024

Cova Pay Canada - Online

Cova Pay offers fast, reliable, and centralized integrated debit and credit card payment processing to Canadian cannabis retailers. Bring the power of Cova Pay Canada to your ecommerce solution for a seamless experience.

Definitions

Company: A business, firm, or corporation that may operate one or many locations as physical brick-and-mortar stores and/or virtual ecommerce presence. Most calls to Cova APIs require a *CompanyId*, which is included in every integration onboarding package we supply.

Location: A physical store or dispensary customers can visit to purchase cannabis and accessories. A company may have many locations. Each location has its own inventory and may define its own prices. Many Cova API endpoints require a location id, commonly represented as *EntityId*. The Data Platform APIs will use *LocationId*

Adyen: A Cova partner and the financial technology platform powering Cova Pay. Adyen describes themselves as a financial technology platform offering various solutions for enterprise businesses to enhance payments. A payment gateway, processor, and acquirer in one platform.

Summary Overview

Taking online payments with Cova Pay generally requires three steps.

1. Make an API call to a Cova endpoint, to begin a payment session.
2. Using the data returned in #1, initialize the web components provided by Adyen.
3. Using the web components provided by Adyen, submit the payment, and take appropriate action upon a success or failure.

Detail

Development Prerequisites

Contact Cova API Support (APISupport@covasoftware.com) to

- add your test site URL to the allowlist
- ensure you are onboarded as an integrator in the appropriate Cova sandbox. A specific location is pre-configured to allow Cova Pay Canada interaction using test credit cards.
- obtain the Cova Pay Adyen Client Key if it has not already been supplied. This is for use in Step 2 below.

Note: Base URI for APIs below: *https://api.covasoft.net/covapaycanada*

Note: During initial development you will make calls to interact with Adyen's TEST environment. Cova facilitates this through a URL route that *you must change* before going LIVE. This safeguard is in place to avoid live credit card payments while developing and testing your solution. The URLs below include the *test* route. Simply remove the */test* segment for routing to the LIVE services.

Step 1. Creating the payment session.

The first step is to create a payment session with Cova. The API spec is as follows:

Request

POST ~/v1/companies/{companyId}/locations/{locationId}/ecomm-payment/test

Headers: Authorization: Bearer {*authtoken*}

Body:

```
{
    "OrderReference": {{order_number_etc}},
    "Amount": 123.49,
    "ReturnUrl": "https://some.url.here"
}
```

- {companyId} is the identifier of the Cova Company
- {locationId} is the identifier of the Cova location.
- OrderReference is a unique string, such as the customer facing "order number" that will be shown as the Reference Number for the transaction in the Cova Pay Canada reports, available on Cova Hub. Please ensure that the Cova Merchant can use this reference to find the order in your system.
- Amount: A positive, decimal number (with 2 decimal digits) representing the amount to charge the card.

- ReturnUrl: A URL that redirects the shopper back to your site, after payment. See Adyen Docs: `https://docs.adyen.com/online-payments/build-your-integration/sessions-flow/?platform=Web&integration=Components&version=6.0.3#handle-the-redirect`
- The auth token is a Cova token for the environment in context. (See `https://api.covasoft.net/Documentation/Api/POST-v1-oauth2-token` for information on obtaining a token)

Response:

```
{
    "sessionId": "unique_covapay_session_id",
    "id": "unique_adyen_session_id",
    "sessionData": "encoded_data"
}
```

- sessionId: A unique Cova Pay session ID generated. Required to initiate any refunds through Cova Pay
- id: A unique Adyen session ID generated and needed to initialize the Adyen component.
- sessionData: A string of encoded data needed to initialize the Adyen web components.

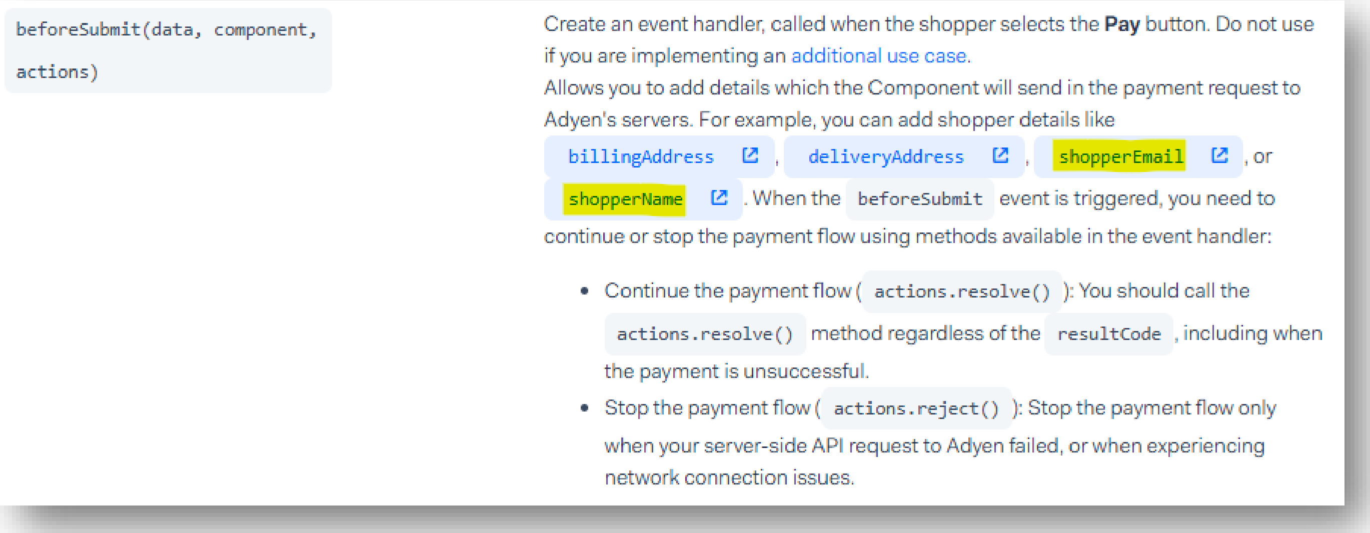
Step 2. Initialize the web components provided by Adyen.

See Adyen documentation here: https://docs.adyen.com/online-payments/build-your-integration/sessions-flow/?platform=Web&integration=Components

The “Drop-in” component or “Components” integration are valid technical options. However, Cova has learned that the “Drop-in" option does not give sufficient customization needed to do things such as hiding options for “Apple Pay”, “Pay-Pal”, etc, if they are not supported. As a result, Cova strongly suggests implementing the “Components”, as per the documentation above.

Follow the documentation above, with one exception: Instead of calling the Adyen /session endpoint, call “Cova Session” endpoint, documented in step 1. It is important to follow this step as critical data is captured in the Cova System needed for reporting and reconciliation, and potential refunds in the POS later.

In addition to the required events when creating the component configuration object (https://docs.adyen.com/online-payments/build-your-integration/sessions-flow/?platform=Web&integration=Components&version=6.4.0#configure), Cova requires you to include the beforeSubmit event and pass through the shopperEmail and shopperName



`beforeSubmit(data, component, actions)`	Create an event handler, called when the shopper selects the **Pay** button. Do not use if you are implementing an additional use case. Allows you to add details which the Component will send in the payment request to Adyen's servers. For example, you can add shopper details like `billingAddress`, `deliveryAddress`, `shopperEmail`, or `shopperName`. When the `beforeSubmit` event is triggered, you need to continue or stop the payment flow using methods available in the event handler: • Continue the payment flow (`actions.resolve()`): You should call the `actions.resolve()` method regardless of the `resultCode`, including when the payment is unsuccessful. • Stop the payment flow (`actions.reject()`): Stop the payment flow only when your server-side API request to Adyen failed, or when experiencing network connection issues.

Step 3. Response

Handle the response from Adyen. Adyen will return either a success or failure, and you need to control the user experience based on the result. See documents from Step 2.

Note: You can extract the last 4 digits from the Adyen web component from the "onFieldValid" event as documented here: `https://docs.adyen.com/payment-methods/cards/web-component/#optional-configuration`

Refunding

An ecommerce refund can be initiated for the full or partial amount of the original sale by calling the following API, supplying the Cova Pay `sessionId` captured in the response from the initial payment.

POST ~`/v1/companies/{CompanyId}/locations/{LocationId}/ecomm-refund/test`

Body:

```
{
    "RefundReference": {{original_order_reference}},
    "SessionId": {{original_covapay_sessionid}},
    "RequestedAmount": 123.49
}
```

- {companyId} is the identifier of the Cova Company
- {locationId} is the identifier of the Cova location.
- RefundReference – supply the original order reference to assist in retailer reconciliation
- SessionId is the original Cova Pay session id, captured from the payment session
- RequestedAmount is the amount to be refunded to the original payment method. **Note**: an amount higher than the original payment will fail asynchronously and not be raised to the refund API caller. Ensure your refund amount is correct.

Going LIVE

Prerequisites

- Ensure all Cova Pay Canada API URLs are for the LIVE environment (*/test* removed)
- Ensure Adyen components are configured for the LIVE environment
- Contact Cova Pay Canada support (CovaPaySupport@covasoftware.com) to:
  - add the site URL to the allowlist. This must be done for each website
  - ensure payment fees are configured for the online locations taking payment. Don't assume an existing Cova Pay customer is configured, as this is separate from a pre-existing configuration that may exist for in-store payments. This must be done for each website.
  - obtain the Cova Pay Adyen Client Key if it has not already been supplied. This key will remain constant for every website. Once it has been provided, you will use it for subsequent websites.