Integrate Fastlane by PayPal

Before beginning your integration, you need to set up your development environment. You can refer to this flow diagram, and watch a video demonstrating how to integrate PayPal Fastlane.

Start your integration by grabbing the sample code from PayPal's GitHub repo, or visiting the PayPal GitHub Codespace. Read the Codespaces guide for more information.

</> Download sample code

Open in Codespaces

1. Integrate front end CLIENT

Set up your front end to integrate Fastlane.

Front-end process

1. Your app shows Fastlane Payment component.
2. Your app calls server endpoints to create the order and capture payment.

Step 1: Generate client token

To instantiate the PayPal JS SDK and the Fastlane components, a client token will need to be generated via a server-side call and then passed into the SDK.

1. Replace `CLIENT ID` with your client ID
2. Replace `CLIENT_SECRET` with your client secret. Get your client ID and secret.
3. Replace domain examples (`example.com` , `example2.com`) with your own domain(s). Provide the root domain name only. No subdomains such as `sub.example.com` . No wildcard characters such as `*.example.com` . No protocols such as `http` or `https` .

Step 2: Initialize PayPal JS SDK and Fastlane

First, Initialize the SDK through a script tag on your HTML page.

The fields required to initialize the PayPal JS SDK are:

- `client-id` : client-id of your application.
- `data-sdk-client-token` : generated from the token initialization.

Fastlane is initialized with a call to `paypal.Fastlane` in the app.js file. You must pass a configuration object during initialization.

Specify locale

You can specify the language in which the Fastlane components are rendered. After you initialize the Fastlane component, you can set the locale.

Fastlane supports the following languages; `en_us` (English, default), `es_us` (Spanish), `fr_us` (French), and `zh_us` (Mandarin).

Step 3: Capture user email address

You’ll need to render your own email field to capture the payer’s email address.

Because the email address will be shared with PayPal, it is crucial to inform the payer. We recommend to display the Fastlane watermark below the email field.

ⓘ **Privacy**
PayPal is a data controller and business under the California Consumer Privacy Act. You'll be sharing consumer's email addresses with PayPal. We recommend you make PayPal known to your consumers by leveraging our SDK to render the "Powered by Fastlane" logo and information tooltip. If you have any questions about this feature or your compliance with data protection laws, consult your legal advisors.

After collecting the email address, PayPal determines whether the email is associated with a Fastlane profile or if it belongs to a PayPal member. Use the following code snippet to look up the customer's email.

Authenticate profile with one-time password

If the user is identified with a Fastlane profile, they need to authenticate before Fastlane retrieves their saved payment and address information. The user is presented with a screen to authenticate themselves by entering a one-time password sent to their registered mobile number.

The `triggerAuthenticationFlow()` method returns an `AuthenticatedCustomerResult` object. Use the `authenticationState` property in the `AuthenticatedCustomerResult` object to check if the payer has authenticated.

On authentication:

- The Fastlane member's card details and default shipping address are returned with the `profileData` object contents.
- The `renderFastlaneMemberExperience` is set to `True`.

If a user fails or declines to authenticate, render the same experience as you would for a guest payer.

Step 4: Render shipping address

This step is not required for:

- Fastlane members without a shipping address.
- Fastlane members with a shipping address in a region you don't support.
- Guest payers

Continue to render your own shipping address collection form and pass it into the server-side transaction request.

For Fastlane members with a shipping address, you'll need to render:

- Shipping address returned from profile data on successful authentication.
- Fastlane watermark.
- Fastlane by PayPal logo.
- Change address button that invokes `showShippingAddressSelector()`.

  This provides a payer with the option to change or add a new address.

ⓘ **Note:**

- Fastlane is only available for US-based customers. However, PayPal doesn't have restrictions on shipping addresses, so the merchant can decide who they want to ship to. This can be done using the `allowedShippingLocations` parameter in the SDK. The merchant can disallow either countries or regions, states, or provinces and PayPal will honor that in our shipping address component and forms.
- If the user adds a new address to Fastlane profile, send the new address in the Orders v2 server-side request.

Step 5: Accept payments

Fastlane offers two different integration patterns to accept payments; quick start and flexible.

Quick Start Flexible

Quick Start

The quick start payment integration loads a pre-built template form to collect payments, requiring less integration effort. The quick start integration is PCI DSS compliant, ensuring that customer payment information is handled securely.

The ready-made payment UI component will automatically render the following:

1. Selected card for the Fastlane member.
2. “Change card” link which allows payers to change the selection or add a new card.
3. Card fields for guest users or for Fastlane members that don’t have a card in their profile.
4. Billing address fields.

After you have received the `paymentTokenId` from the `paymentToken` object , it should be sent to your server to create a transaction with it.

Refer to the Reference Types section for more information on what options can be passed to each of the functions shown in the code sample above.

Render watermark

When displaying the card from payer’s Fastlane profile, you must inform them about it by displaying the Fastlane watermark below the card.

For a better payer experience of Fastlane, preload the watermark asset by adding the following code to the `<head>` section of the page.

The image tag in the watermark container div allows the watermark to render instantly.

The mouse-over tooltip functionality will be added to the image when the Watermark component completes loading. After it loads, the information is displayed when the user hovers over the i icon.

When opting to integrate using the flexible pattern, load card assets for the selected card so that the payer can see their card brand.

Step 6: Configure the styling of the Fastlane component OPTIONAL ✓

When styling the Fastlane components to match the look and feel of your checkout page, here are some guidelines to provide an accessible and transparent experience to your payer:

- Ensure that there is adequate contrast between the `backgroundColor` and `textColor` to ensure that all text, especially the legal text under the opt-in, is clear and legible. If the contrast ratio between the two is not 4.5:1 or greater, PayPal will automatically set the contrast to their default values.
- Ensure there is adequate contrast between the `borderColor`, which drives the consent toggle coloring, and the `backgroundColor`.

To override the default style settings for your page, use a `StyleOptions` object inside the `Fastlane` component. You can customize the styling with different colors and shapes.

To override the default style settings for your page, use a `StyleOptions` object inside the `Fastlane` component.

You can customize the styling with different colors and shapes.

Visit the Reference page for more details about additional customization and style options.

2. Integrate back end SERVER

This section explains how to set up your back end to integrate standard checkout payments.

Back-end process

Your app creates and confirms an order on the back end by calling to the Create Orders V2

API endpoint.

Create order

On your server, you need to create an order by invoking the PPCP Orders API and passing the single use token, along with the item details and the shipping address.

Intent	Intent CAPTURE
Amount	Amount 100

Note: If the payer adds a new address, be sure to send that address in the server-side `orders API` request.

> Providing detailed item descriptions and product images for each item creates a better user experience. It helps email communications and streamlines any potential dispute resolution.
>
> ### Special use-cases
>
> **Store pick-up:** If the buyer is picking up item from a storefront, modify the shipping type parameter in the call and ensure the shipping method is set to `PICKUP_IN_STORE` . This ensures that buyer profiles aren't created with the address of your store as their shipping address.
>
> **Vaulting:** Ensure you have configured your REST app to enable vault.
>
> If you want to save the `paymentToken` returned by the Fastlane SDK at the time of transaction, you can do so only when using the `store_in_vault` attribute in the request to `/v2/orders` on your server. This returns a vault ID, which can be saved and used for future captures. See the Orders v2 documentation for more details.

3. Test integration

You'll need to test your integration for guest payers, Fastlane members, and PayPal members. If you're using the flexible integration, make sure your address and payment form fields render as expected.

Card numbers for testing

Use any of the following card numbers as test cards.

Brand	**Test Number**
Visa	4005 5192 0000 0004
Visa	4012 0000 3333 0026
Visa	4012 0000 7777 7777
Mastercard	5555 5555 5555 4444
American Express	3782 822463 10005

Test guest payers

Before you test guest payers, make sure you:

- Create a new email address: Provide a new email address which is not associated with a sandbox Fastlane account.
- Ensure opt-in toggle is ON: Go through the checkout process using one of the card numbers available for testing. Be sure that the opt-in toggle is in the "on" state.
- Enter any valid phone number for sandbox: Make sure to pass a valid area code and prefix. A Fastlane profile is not created if you pass an invalid number such as `111–111–1111` . No SMS is sent in sandbox mode. Upon completing the transaction, a Fastlane profile is created. Use that profile to test subsequent transactions as a Fastlane member.

Test guest payer flows with the consent toggle both on and off. If the consent toggle is off, the payment completes successfully and no Fastlane profile is created.

Test Fastlane members

Before you test Fastlane payers, make sure you:

- Create a Fastlane Member profile: Go through the previous step and register a new Fastlane account. Be sure to remember the email address used when you created the account so that you can use it for additional testing.
- Use OTP for testing: When the authentication modal appears and you are prompted for a one-time password (OTP) use `111111` to trigger a successful authentication and any other 6-digit number to simulate a failed authentication.
- Test updating payment method and shipping addresses to existing Fastlane Profiles: Make sure that you test the payer's ability to update shipping addresses and cards associated with their profile.

Test the Fastlane member scenario as the member toggles all the options such as adding or changing an address or card, or failing the OTP.

Test PayPal members

PayPal members that choose not to save their profile will be treated as guest users.

PayPal members that choose to save their profile will be treated as returning Fastlane member for future transactions.

PayPal members do not require any additional handling within your integration because our client SDK handles this use case for you in the following ways:

- After performing the `lookupCustomerByEmail` method, we will return a `customerContextId` as if this were a Fastlane member.
- When you call the `triggerAuthenticationFlow` method, our SDK will display a call to action to the customer explaining that they can create a Fastlane profile populated with information from their PayPal account with one click.

- If the consumer clicks yes, we will return `profileData` exactly as we would for a Fastlane member.
- If the consumer closes the dialog, we will return an empty `profileData` object and you would handle this as you would any Fastlane guest member.

Visit the Reference page for more details about troubleshooting, best practices, customization and style options.

4. Go live

If you have fulfilled the requirements for accepting card payments via Fastlane for your business account, review Move your app to production to test and go live.

If this is your first time testing in a live environment, follow these steps:

1. Log into the PayPal Developer Dashboard with your PayPal business account.
2. Complete production onboarding so you can process card payments with your live PayPal business account.
3. Request Advanced Credit and Debit Card Payments for your business account if you are not using cards currently with PayPal. Fastlane will be enabled along with the Advanced Credit and Debit Card payments processing.

Preview Code Expand

Test Cards

Customer

Email

Continue

Frontend
HTML

Backend
Node.js

Credentials ?

App
Default Application

Client ID
AZ1GC3JHk-Q07h_fQh81...

kies Support Legal Contact