

🔔 We've made changes to this page and its layout to improve the developer experience. Let us know what you think of the updated documentation by selecting the feedback tab below.

Checkout

Standard

Advanced

Copy and Paste

Pay Later Offers

Pay with Venmo

Save Payment Methods

Alternative Payment Methods

Apple Pay

Bancontact

BLIK

eps

giropay

Google Pay

Test Card Suite

iDEAL

Multibanco

MyBank

Pay upon Invoice

Przelewy24

Trustly

Reference

Payment Methods

Checkout / Alternative Payment Methods / Google Pay

Integrate Google Pay with JS SDK for direct merchants

[Docs](#) [Current](#) Last updated: December 13th 2023, @ 3:50:11 pm

Overview

How it works

Know before you code

Setup

Getting started

Integrate Google Pay

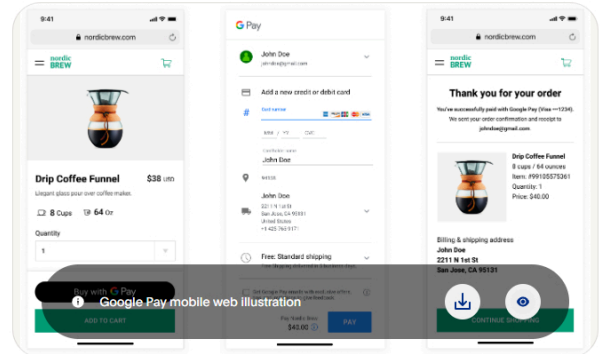
Button setup

Google Pay integration

Google Pay is a mobile payment and digital wallet service provided by Alphabet Inc.

Buyers can use Google Pay on PayPal to make payments on the web using a web browser.

Sellers can use PayPal with Google Pay to sell physical goods, such as clothes and electronics, and intangible professional services, such as concerts or gym memberships.



Supported countries and currencies

Google Pay supports payments in 32 countries and 22 currencies:

- Countries:** Australia, Austria, Belgium, Bulgaria, Canada, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Ireland, Italy, Latvia, Liechtenstein, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, United States, United Kingdom
- Currencies:** [AUD](#), [BRL](#), [CAD](#), [CHF](#), [CZK](#), [DKK](#), [EUR](#), [GBP](#), [HKD](#), [HUF](#), [ILS](#), [JPY](#), [MXN](#), [NOK](#), [NZD](#), [PHP](#), [PLN](#), [SEK](#), [SGD](#), [THB](#), [TWD](#), [USD](#)

Tip: If you want to integrate additional methods of accepting payment beyond Google Pay, visit our [Advanced Checkout guide](#) for additional integration choices.

How it works

1. The Google Pay button shows up on your website when a customer uses a web browser.
2. The buyer selects the Google Pay button on your website.
3. Your website shows the buyer a payment sheet.
4. The buyer can choose a different shipping address and payment method.
5. The buyer authorizes the payment.

Know before you code



Required

Google Pay works on most web browsers, including:

- Google Chrome.
- Mozilla Firefox.
- Apple Safari.
- Microsoft Edge.

[Check Compatibility](#)

Required

Currently supports Google Pay one-time payments with the buyer present.

Review Google's Google Pay API Terms of Service and Acceptable Use Policy for more information.

[Google Terms of Service](#)[Acceptable use policy](#)

Get up and running in GitHub Codespaces

GitHub Codespaces are cloud-based development environments where you can code and test your PayPal integrations. [Learn more](#)

1 Set up your sandbox account to accept Google Pay

Before you can accept Google Pay on your website, verify that your sandbox business account supports Google Pay.

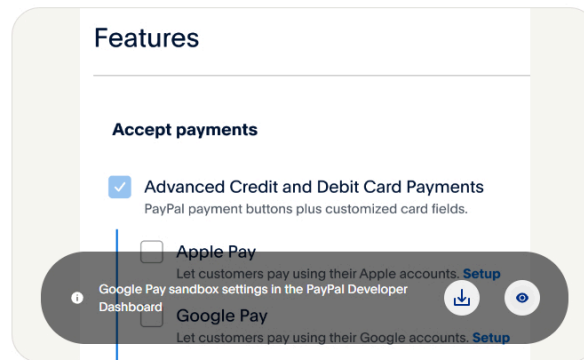
Direct merchants can use the PayPal Developer Dashboard to set up their sandbox accounts to accept Google Pay.

1. Log into the PayPal [Developer Dashboard](#) and go to your sandbox account.
2. Go to **Apps & Credentials**.
3. Make sure you are in the PayPal sandbox environment by selecting **Sandbox** at the top.
4. Select or create an app.
5. Scroll down to **Features** and check if Google Pay is enabled. If Google Pay isn't enabled, select the **Google Pay** checkbox and select the "Save" link to enable Google Pay.

If you created a sandbox business account through sandbox.paypal.com, and the Google Pay status for the account shows as disabled, [complete the sandbox onboarding steps](#) to enable Google Pay.

Tip: When your integration is ready to go live, read the **Go live** section for details about the additional steps needed for Google Pay onboarding.

This screenshot shows the Google Pay sandbox settings in the mobile and digital payments section of the PayPal Developer Dashboard. This only applies to direct merchant integrations:



2 Getting started in your testing environment

Before you develop your Google Pay on the Web integration, you need to complete [Get started](#) to set up your PayPal account, client ID, and sandbox emails for testing.

3 Integrate Google Pay checkout

Follow this integration process to add Google Pay as a checkout option, customize the payment experience, and process payments.

Call the Orders API

To accept Google Pay directly on your website, create API endpoints on your server that communicate with the [PayPal Orders V2 API](#). These endpoints can create an order, authorize payment, and capture payment for an order.

Server-side example (Node.js)

This code demonstrates using the [PayPal Orders V2 API](#) to add routes to an Express server for creating orders and capturing payments.

Find the complete sample code in the [GitHub repo](#).

server.js

paypal-api.js



```
1 import * as PayPal from "../paypal-api.js";
2
3 /* Create Order route Handler */
4 app.post("/api/orders", async (req, res) => {
5   const order = await PayPal.createOrder();
6   res.json(order);
7 });
8
9 /* Capture Order route Handler */
10 app.post("/api/orders/:orderID/capture", async (req, res) => {
11   const {
12     orderID
13   } = req.params;
14   const captureData = await PayPal.capturePayment(orderID);
```

```
15 res.json(captureData);
16 });
```

4 Set up your Google Pay button

You need to integrate with the Google Pay JavaScript SDK and PayPal JavaScript SDK to add Google Pay to your site.

Integrate PayPal JavaScript SDK

Use this script to integrate with the PayPal JavaScript SDK:

```
1 <script src="https://www.paypal.com/sdk/js?client-id=YOUR_CLIENT_ID&currency=USD&buyer-country=US&merchant-id=SUB_MERCHANT_ID&components=googlepay">
```

Include `googlepay` in the `components` list.

Integrate Google JavaScript SDK

Use this script to integrate with the Google Pay JavaScript SDK:

```
1 <script async src="https://pay.google.com/gp/p/js/pay.js" onload="onGooglePayLoaded()"></script>
```

PayPal's Google Pay component interacts with your JavaScript code in 2 areas:

1. Checking merchant eligibility and providing `PaymentDataRequest` parameters for Google Pay: `paypal.GooglePay().config()`.
2. Handling the `onPaymentAuthorized()` callback: `paypal.GooglePay().confirmOrder()`.

Check for device and merchant eligibility before setting up the Google Pay Button.

The PayPal JavaScript SDK API `paypal.GooglePay().config()` response object provides the `allowedPaymentMethods` parameter, which is part of the "Google API's `isReadyToPayRequest` object.

Check whether the Google Pay API supports a device, browser, and payment method:

- Add `allowedPaymentMethods` to the `isReadyToPayRequest`.
- Call the `isReadyToPay()` method to check compatibility and render the Google Pay Button.

```
1 /**
2  * Initialize Google PaymentsClient after Google-hosted JavaScript has Loaded
3  *
4  * Display a Google Pay payment button after confirmation of the viewer's
5  * ability to pay.
6  */
7  function onGooglePayLoaded() {
8    const paymentsClient = getGooglePaymentsClient();
9    paymentsClient.isReadyToPay(isReadyToPayRequest)
10     .then(function(response) {
11       if (response.result) {
12         addGooglePayButton();
13       }
14     })
15     .catch(function(err) {
16       console.error(err);
17     });
18 }
19 /**
20  * Add a Google Pay purchase button
21  */
22  function addGooglePayButton() {
23    const paymentsClient = getGooglePaymentsClient();
24    const button =
25      paymentsClient.createButton({
26        onClick: onGooglePaymentButtonClicked /* To be defined later */,
27        allowedPaymentMethods: [baseCardPaymentMethod]
28      });
29    document.getElementById('container').appendChild(button);
30 }
```

Tip: For more information refer to steps 6 and 7 in [Google's developer documentation](#).