



Google Pay™ Integration Guide for Merchants

How to accept Google Pay through the CetraPay payment gateway

Gateway identifier
cetrapay

Platforms
Web, Android, Hosted Checkout

Version
1.0, October 2026

1 Overview

CetraPay S.p.A. is a supported payment gateway for Google Pay™. With Google Pay, your customers pay with cards saved to their Google Account or with cards stored as device tokens on their Android devices.

You can accept Google Pay through CetraPay in two ways:

- **API integration.** You integrate the Google Pay API into your website or Android app, receive an encrypted payment token from Google, and send it to the CetraPay API. CetraPay decrypts the token and processes the payment.
- **Hosted Checkout.** You redirect the customer to the CetraPay payment page, where the Google Pay button is already available. No direct integration with the Google Pay API is needed.

Terms. By using Google Pay through CetraPay, with either integration path, you must adhere to the [Google Pay APIs Acceptable Use Policy](#) and accept the terms defined in the [Google Pay API Terms of Service](#).

2 Before you start

- An active CetraPay merchant account with Google Pay enabled. Contact your account manager or info@cetrapay.pro.
- Your API credentials (`merchantKey` and `secret`) and your merchant account name.
- Your `gatewayMerchantId` , issued by CetraPay. It identifies you in the CetraPay system and is bound to the Google Pay token.
- For production on the web: a Google merchant ID from the [Google Pay & Wallet Console](#), issued after Google approves your integration.

3 Google Pay API integration

Use the `PAYMENT_GATEWAY` tokenization type. Follow the Google documentation for your platform and complete the integration checklist before requesting production access.

Platform	Developer documentation	Integration checklist	Brand guidelines
Web	Google Pay Web developer documentation	Google Pay Web integration checklist	Google Pay Web Brand Guidelines
Android	Google Pay Android developer documentation	Google Pay Android integration checklist	Google Pay Android brand guidelines

3.1 Gateway parameters

Pass the following `tokenizationSpecification` in your `PaymentDataRequest` :

Parameter	Value	Description
<code>gateway</code>	<code>cetrapay</code>	CetraPay gateway identifier registered with Google. Always this exact value.
<code>gatewayMerchantId</code>	<code><provided by CetraPay></code>	Your merchant identifier in the CetraPay system. A token created with a different value is rejected.

```
const tokenizationSpecification = {
  type: 'PAYMENT_GATEWAY',
  parameters: {
    'gateway': 'cetrapay',
    'gatewayMerchantId': '<provided by CetraPay>'
  }
};
```

3.2 Authentication methods

Google Pay offers two card authentication methods. CetraPay supports both, and we recommend enabling both in `allowedAuthMethods` :

Method	What it is	3D Secure
<code>PAN_ONLY</code>	Cards stored on file with the customer's Google Account. The payment data contains the card number (PAN) with its expiration month and year.	Required. CetraPay applies 3D Secure authentication. See section 4.
<code>CRYPTOGRAM_3DS</code>	Cards stored as device tokens on an Android device. The payment data contains a 3D Secure cryptogram generated on the device.	Already authenticated on the device. No additional step is required from you.

Availability by country and currency follows the configuration of your CetraPay merchant account.

3.3 Card networks

Set `allowedCardNetworks` to the networks enabled on your CetraPay account. CetraPay supports:

- VISA
- MASTERCARD

3.4 Billing address

A billing address is **not required** to process a Google Pay payment through CetraPay. You may leave `billingAddressRequired` unset. If your business needs the address, request it with `billingAddressRequired`:

`true` and [BillingAddressParameters](#), and pass it to CetraPay in the optional `customer_*` fields described in section 5.

3.5 Full configuration example (Web)

```
const baseCardPaymentMethod = {
  type: 'CARD',
  parameters: {
    allowedAuthMethods: ['PAN_ONLY', 'CRYPTOGRAM_3DS'],
    allowedCardNetworks: ['VISA', 'MASTERCARD']
  }
};

const paymentDataRequest = {
  apiVersion: 2,
  apiVersionMinor: 0,
  allowedPaymentMethods: [{
    ...baseCardPaymentMethod,
    tokenizationSpecification: {
      type: 'PAYMENT_GATEWAY',
      parameters: {
        'gateway': 'cetrapay',
        'gatewayMerchantId': '<provided by CetraPay>'
      }
    }
  }],
  merchantInfo: {
    merchantId: '<your Google merchant ID>', // production only
    merchantName: '<your business name>'
  },
  transactionInfo: {
    totalPriceStatus: 'FINAL',
    totalPrice: '10.00',
    currencyCode: 'EUR',
    countryCode: 'SM'
  }
};
```

4 3D Secure for PAN_ONLY

CetraPay supports 3D Secure for card payments, and the same applies to Google Pay `PAN_ONLY` credentials returned in the encrypted payload.

- **How to enable it.** You do not need to enable anything or send extra parameters. 3D Secure is applied by CetraPay on the gateway side for transactions made with `PAN_ONLY` credentials.
- **What you need to implement.** When authentication is required, the API response contains a `form_url`. Redirect the customer to this URL to complete the challenge. After it, the customer returns to your `success_url` or `fail_url`.
- **CRYPTOGRAM_3DS.** Device-token transactions carry a cryptogram and are not sent through an additional 3D Secure challenge, unless the issuer requires it.

We recommend sending the optional `browser_*` parameters and `customer_ip_address` (section 5) so that authentication can complete without friction where possible.

5 Sending the payment token to CetraPay

After the customer confirms the payment, Google returns a `PaymentData` object. Take the encrypted token from `paymentMethodData.tokenizationData.token`, Base64-encode it, and send it from your server to the CetraPay

API. Do not decrypt or modify the token.

5.1 Request

POST https://<CETRAPAY_CHD_API_HOST>/initPaymentHost2Host

Authentication: HTTP Basic, Authorization: Basic `base64(merchantKey:secret)` . Content type: `application/json` .

```
curl --location 'https://<CETRAPAY_CHD_API_HOST>/initPaymentHost2Host' \
--header 'Content-Type: application/json' \
--header 'Authorization: Basic <base64(merchantKey:secret)>' \
--data-raw '{
  "payment_method": "googlePayDPan",
  "account": "<your account name>",
  "amount": 1000,
  "currency": "EUR",
  "order_id": "order_1015",
  "purpose": "Order 1015",
  "customer_email": "customer@example.com",
  "customer_ip_address": "203.0.113.10",
  "success_url": "https://site.com/success",
  "fail_url": "https://site.com/fail",
  "callback_url": "https://site.com/callback",
  "google_pay_dpan": "<Base64 of paymentMethodData.tokenizationData.token>",
  "google_pay_payment_method_network": "VISA"
}'
```

5.2 Request parameters

Parameter	Required	Type	Description
payment_method	Yes	String	Always <code>googlePayDPan</code> for Google Pay payments.
account	Yes	String	Your merchant account name.
amount	Yes	Long	Amount in minor units (cents).
currency	Yes	String	ISO 4217 alpha-3 currency code.
order_id	Yes	String	Your order identifier.
google_pay_dpan	Yes	String	Base64-encoded value of <code>paymentMethodData.tokenizationData.token</code> .
google_pay_payment_method_network	Yes	String	Value of <code>paymentMethodData.info.cardNetwork</code> .
callback_url	No	String	URL that receives a notification when the transaction status changes.
success_url , fail_url	No	String	Where the customer is returned after a 3D Secure challenge.
purpose	No	String	Order description.
customer_first_name , customer_last_name , customer_email , customer_phone , customer_ip_address	No	String	Customer details.

customer_address , customer_city , customer_zip_code , customer_country	No	String	Billing address, if you collect it.
browser_accept_header , browser_user_agent , browser_language , browser_time_zone , browser_color_depth , browser_screen_height , browser_screen_width , browser_java_enabled , browser_js_enabled	No	String / Boolean	Customer browser data used for 3D Secure.

5.3 Response

```
{
  "form_url": "https://<CETRAPAY_PAYMENT_HOST>/64a7eec2d46c5db10d082b3d",
  "order_id": "64a7eec2d46c5db10d082b3d",
  "status": 1,
  "responseType": "REDIRECT",
  "err": null
}
```

Parameter	Type	Description
status	Int	Transaction status.
form_url	String	Returned when the customer must be redirected, for example for a 3D Secure challenge.
order_id	String	Order identifier in the CetraPay system.
err	String	Error reason, if the request failed.

Treat the payment as successful only after the callback or a transaction status check reports the final successful status. A customer returning to `success_url` is not a payment confirmation.

6 Hosted Checkout

With Hosted Checkout, the Google Pay button is shown on the CetraPay payment page and CetraPay builds the Google Pay request for you. You do not call the Google Pay API, do not handle the payment token, and do not need a Google merchant ID.

1. Ask CetraPay to enable Google Pay on your merchant account.
2. Create a payment form with `POST https://<CETRAPAY_API_HOST>/initPayment`, passing `account`, `amount`, `currency` and `order_id`.
3. Redirect the customer to the `form_url` from the response. The customer chooses Google Pay on the payment page.
4. Wait for the callback, or check the transaction status, to confirm the payment.

No changes to your website's Content-Security-Policy and no SDK are required, because the customer leaves your site for the hosted page. The authentication methods, card networks and 3D Secure handling described in sections 3 and 4 apply to Hosted Checkout in the same way.

Terms. When you use the CetraPay hosted checkout page, you must adhere to the [Google Pay APIs Acceptable Use Policy](#) and accept the terms defined in the [Google Pay API Terms of Service](#).

7 Testing and going live

- Develop against the Google Pay `TEST` environment. It returns test payment data that cannot be charged.
- Complete the Google integration checklist for your platform (section 3) and follow the Google Pay brand guidelines for the button and for any mention of Google Pay in your interface.
- Request production access in the [Google Pay & Wallet Console](#), and select **CetraPay** as your gateway.
- After approval, switch to `PRODUCTION`, set your Google merchant ID in `merchantInfo`, and use your production CetraPay credentials and `gatewayMerchantId`.

8 Support

For questions about your integration, your `gatewayMerchantId` or enabling Google Pay on your account, contact your account manager or write to info@cetrapay.pro.

CetraPay S.p.A., Registration Nr. 8667, Via Consiglio Dei Sessanta 99, 47891, Republic of San Marino. MSB License Nr. M23007089. Google Pay is a trademark of Google LLC.