



---

<https://verifone.cloud/docs/online-payments/api-integration/google-paytm>

Updated: 03-Apr-2026

## Google Pay™

### Overview

You can accept payments through Google Pay™ via [Checkout?](#)(Hosted Payments Page - HPP) or via [eComm API](#).

### Hosted checkout integration

Follow the guide [here](#) for integrating Hosted Checkout with Google Pay as a payment option.

### Server-to-Server integration

This guide covers a two-part integration: (1) a client-side Google Pay button that collects an encrypted payment token, and (2) a server-side call to Verifone [eCommerce API](#) to process it.

Follow the integration steps below to process transactions via Verifone eComm API.

### Required fields

Parameters	Description
payment_provider_contract	Find it in <i>Verifone Central -&gt; Administration -&gt; Payment Provider Contracts</i> . Copy the Payment Provider Contract ID.
amount	Amount of the transaction
merchant_reference	Unique UUID you generate and can link the transaction to when the customer returns
currency_code	More on the all currencies supported here: <a href="#">Verifone eCommerce API   Verifone Developer Portal</a>
wallet_type	string Enum: "GOOGLE_PAY"
wallet_payload	object The encrypted payload object provided by the Wallet on the frontend

A Google Pay web integration consists of implementing both client-side and server-side components. You will need to implement the following:

- Configure the Verifone payment gateway ID and your merchant ID to initialize the Google pay client.

<https://developers.google.com/pay/api/web/guides/tutorial>

- Use the Google Pay token to make a wallet transaction API call and complete the session based on the response

<https://verifone.cloud/api-catalog/verifone-ecommerce-api#operation/walletTransaction>

## Code sample

```
// Specify verifone as the gateway and define your wallet contract
const tokenizationSpecification = {
  type: 'PAYMENT_GATEWAY',
  parameters: {
    'gateway': 'verifone',
    'gatewayMerchantId': 'BCR2DN6T2OAK7HIW'
  }
};

// Specify your google pay merchant ID and name when setting up your paymentDataRequest
paymentDataRequest.merchantInfo = {
  // @todo a merchant ID is available for a production environment after approval

  // See {@link https://developers.google.com/pay/api/web/guides/test-and-deploy/instrumentation/initialize-payment-data-request#merchant-id}
  merchantId: 'YOUR_MERCHANT_ID',
  merchantName: 'YOUR_MERCHANT_NAME'
};

// Pass the google pay token to your server to make the wallet transaction API call

function onGooglePaymentButtonClicked() {
  const paymentDataRequest = getGooglePaymentDataRequest();
  paymentDataRequest.transactionInfo = getGoogleTransactionInfo();

  const paymentsClient = getGooglePaymentsClient();
  paymentsClient.loadPaymentData(paymentDataRequest)
    .then(function(paymentData) {
      // handle the response
      paymentToken = paymentData.paymentMethodData.tokenizationData.token;
      fetch("YOUR_SERVER", {
        method: "POST",
        headers: {"Content-Type": "application/json"},
        body: getPayload(JSON.parse(paymentToken)),
      }).then((res) => res.json())
        .then((res) => {
          // handle wallet transaction response
        });
    });
}
```

```
        if (res.status.includes('AUTHORIZED')) {
            alert('Payment Successful');
        }
    })
    .catch((err) => console.log(err));
})
.catch(function(err) {
    // show error in developer console for debugging
    console.error(err);
});
}
```