

Google Pay™

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

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

Required fields

Parameters	Description
<code>payment_provider_contract</code>	In the Payment Provider Contracts section in Verifone Central, set the <i>Payment Type</i> to <i>Google Pay for web</i> , select your contract and copy the Payment Provider Contract ID. Please note this value is different in Sandbox and in Production.
<code>amount</code>	Amount of the transaction
<code>merchant_reference</code>	Unique UUID you generate and can link the transaction to when the customer returns
<code>currency_code</code>	More on the all currencies supported here: Verifone eCommerce API Verifone Developer Portal
<code>wallet_type</code>	string Enum: "GOOGLE_PAY"
<code>wallet_payload</code>	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': 'YOUR_WALLET_CONTRACT'
  }
}
```

```
};

// 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 by Google
  // See {@link https://developers.google.com/pay/api/web/guides/test-and-deploy/integration-checklist|Integration
  // checklist}
  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);
    });
}
```