Google Pay
Simple Order API
Developer Guide
Contents

Introduction .......................................................................................................................... 4
Google Pay Overview ........................................................................................................... 5
Payment Network Tokens ..................................................................................................... 6
  Requirements ....................................................................................................................... 6
  Supported Processors .......................................................................................................... 6
  How Google Pay Works .................................................................................................... 8
  Additional Services ............................................................................................................ 10
  Transaction Endpoints ...................................................................................................... 10
Payment Data ........................................................................................................................ 12
  Configuring Google Pay ..................................................................................................... 12
    Example: Java Code .......................................................................................................... 12
    Example: JavaScript Code ................................................................................................. 12
  Formatting Payment Blobs ............................................................................................... 12
    Sample Code ................................................................................................................... 12
    Example: Unencrypted Google Pay Response ................................................................... 13
    Example: Base64 Encoded Google Pay Response .............................................................. 13
Authorizing Payments .......................................................................................................... 15
Authorizing a Payment .......................................................................................................... 16
  Example: Authorization Request ....................................................................................... 16
  Example: Authorization Request ....................................................................................... 17
Introduction

Cybersource can be used to process and manage Google Pay transactions.
Google Pay Overview

Google Pay is a simple, secure in-app mobile and Web payment solution. You can choose Cybersource to process Google Pay transactions through all e-commerce channels.

You can simplify your payment processing by allowing Cybersource to decrypt the payment data for you during processing.

This method integrates simply and allows you to process transactions without seeing the payment network token and transaction data.

1. Using the Google API, request the customer’s encrypted payment data.
2. Using the Cybersource API, construct and submit the authorization request and include the encrypted payment data from the Google Pay call back.
3. Cybersource decrypts the encrypted payment data to create the payment network token and processes the authorization request.
Payment Network Tokens

Authorizations with payment network tokens enable you to securely request a payment transaction with a payment network token instead of a customer’s primary account number (PAN).

The payment network token is included in the customer’s encrypted payment data, which is returned by the payment processor.

For information about authorizations with payment network tokens, see the Authorizations with Payment Network Tokens Guide.

Requirements

Before using Google Pay, you will need:

- A Cybersource merchant evaluation account if you do not have one already: https://www.cybersource.com/register/
- A merchant account with Visa Platform Connect.
- A Google developer account.
- Google Pay APIs embedded into your application or website.
  For details about integrating Google Pay, see the Google Pay API Documentation.

Supported Processors

Merchant-initiated transactions and multiple partial captures are described in Authorizations with Payment Network Tokens Developer’s Guide. Recurring payments and split shipments are described in Credit Card Services Developer’s Guide.

<table>
<thead>
<tr>
<th>Processor</th>
<th>Card Types</th>
<th>Optional Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>American Express Direct</td>
<td>• American Express</td>
<td>• Recurring Payments</td>
</tr>
<tr>
<td>Barclays</td>
<td>• Mastercard</td>
<td>• Multiple Partial Captures</td>
</tr>
<tr>
<td></td>
<td>• Visa</td>
<td>• Recurring Payments</td>
</tr>
<tr>
<td>Processor</td>
<td>Card Types</td>
<td>Optional Features</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>Chase Paymentech Solutions</td>
<td>• American Express</td>
<td>• Recurring Payments</td>
</tr>
<tr>
<td></td>
<td>• Discover</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Mastercard</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Visa</td>
<td></td>
</tr>
<tr>
<td>Credit Mutuel-CIC</td>
<td>• Cartes Bancaires</td>
<td>• Recurring Payments</td>
</tr>
<tr>
<td></td>
<td>• Mastercard</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Visa</td>
<td></td>
</tr>
<tr>
<td>Elavon Americas</td>
<td>• American Express</td>
<td>• Merchant-initiated transactions</td>
</tr>
<tr>
<td></td>
<td>• Discover</td>
<td>• Multiple partial captures</td>
</tr>
<tr>
<td></td>
<td>• JCB</td>
<td>• Recurring payments</td>
</tr>
<tr>
<td></td>
<td>• Mastercard</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Visa</td>
<td></td>
</tr>
<tr>
<td>FDC Compass</td>
<td>• American Express</td>
<td>• Recurring Payments</td>
</tr>
<tr>
<td></td>
<td>• Mastercard</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Visa</td>
<td></td>
</tr>
<tr>
<td>FDMS Nashville</td>
<td>• American Express</td>
<td>• Multiple partial captures</td>
</tr>
<tr>
<td></td>
<td>• Discover</td>
<td>• Recurring Payments</td>
</tr>
<tr>
<td></td>
<td>• Mastercard</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Visa</td>
<td></td>
</tr>
<tr>
<td>GPN</td>
<td>• American Express</td>
<td>• Split shipments</td>
</tr>
<tr>
<td></td>
<td>• Mastercard</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Visa</td>
<td></td>
</tr>
<tr>
<td>JCN Gateway</td>
<td>• JCB</td>
<td>• Multiple partial captures</td>
</tr>
<tr>
<td>Moneris</td>
<td>• American Express</td>
<td>• Merchant-initiated captures</td>
</tr>
<tr>
<td></td>
<td>• Mastercard</td>
<td>• Recurring Payments</td>
</tr>
<tr>
<td></td>
<td>• Visa</td>
<td></td>
</tr>
<tr>
<td>OmniPay Direct</td>
<td>• Mastercard</td>
<td>• Recurring Payments</td>
</tr>
<tr>
<td>Supported acquirers are:</td>
<td>• Visa</td>
<td></td>
</tr>
<tr>
<td>Processor</td>
<td>Card Types</td>
<td>Optional Features</td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>• Bank of America Merchant Services • Cardnet International • First Data Europe through OmniPay Direct • Global Payments International Acquiring through OmniPay Direct</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIX</td>
<td>• Mastercard • Visa</td>
<td>• Recurring Payments</td>
</tr>
<tr>
<td>SIX only supports card-present processing.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Streamline</td>
<td>• Mastercard • Visa</td>
<td>• Recurring Payments</td>
</tr>
<tr>
<td>SIX only supports card-present processing.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSYS Acquiring Solutions</td>
<td>• American Express • Mastercard • Visa</td>
<td>• Multiple partial captures</td>
</tr>
<tr>
<td>Visa Platform Connect Supported acquirerers are:</td>
<td>• Mastercard • Visa</td>
<td>• Recurring Payments</td>
</tr>
<tr>
<td>• Australia and New Zealand Banking Group Limited (ANZ) • Vantiv • Westpac</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Worldpay VAP</td>
<td>• Mastercard • Visa</td>
<td>• Recurring Payments</td>
</tr>
</tbody>
</table>

How Google Pay Works

The following describes the Google Pay workflow:
1. The customer chooses the Google Pay button. Using the Google API, your system initiates the Google Pay request identifying Cybersource as your payment gateway, passing your Cybersource merchant ID as the gateway merchant ID.

2. The customer confirms the payment. The Google API contacts Google Pay services to retrieve the consumer’s payment parameters.

3. If the customer's selected payment credentials are tokenized or you are tokenizing new payment credentials, the Google Pay service contacts the appropriate payment network to retrieve the appropriate cryptogram.

4. The payment network returns the appropriate token and cryptogram to the Google Pay service.

5. Google creates encrypted payment data using the gateway-specific key that is supplied in the Wallet request and includes it in the Google API response.

6. The Google Pay call back returns the encrypted payment data.

7. Your system prepares the Google Pay response information for submission to the Cybersource service.

8. a. Cybersource sends the authorization request to the acquirer.
   b. The acquirer processes the request from Cybersource and creates the payment network authorization request.
   c. The payment network processes the request from the acquirer and creates the issuer authorization request.
d. The issuer processes the request from the payment network. The issuer looks up the payment information and returns an approved or declined authorization message to the payment network.
e. The payment network returns the authorization response to the acquirer.
f. The acquirer returns the authorization response to Cybersource.

9. Cybersource returns the authorization response to your system.
10. Your system returns the authorization response to the payment application.
11. The payment application displays the confirmation or decline message to the customer.
12. a. The acquirer submits the settlement request to the issuer for funds.
    b. The issuer supplies the funds to the acquirer for the authorized transactions.

Additional Services

The following additional services can be used with Google Pay. For more information on these services, see the Credit Card Services Developer’s Guide.

**Capture**

A follow-on service that uses the request ID returned from the previous authorization. The request ID links the capture to the authorization. This service transfers funds from the customer’s account to your bank and usually takes two to four days to complete.

**Sale**

A sale is a bundled authorization and capture. Request the authorization and capture services at the same time. Cybersource processes the capture immediately.

**Authorized Reversal**

A follow-on service that uses the request ID returned from the previous authorization. An authorization reversal releases the hold that the authorization placed on the customer's credit card funds. Use this service to reverse an unnecessary or undesired authorization.

Transaction Endpoints

The following endpoints are used with Google Pay.

**CAS (test transactions)**

http://ics2testa.ic3.com

**Production (live transactions)**
Payment Data

The following examples show you how to format encrypted payment data.

Configuring Google Pay

You must provide your Cybersource merchant ID to Google in order to ensure proper encryption of the Google Pay payload and authenticity of the request.
For a Google Pay tutorial, see: https://developers.google.com/pay/api/android/guides/tutorial
Set the gateway and gateway merchant ID to the appropriate indicators. The following code examples show how to configure the PaymentMethodTokenizationParameters object using Cybersource as the gateway.

Example: Java Code

```java
.setPaymentMethodTokenizationType(WalletConstants.PAYMENT_METHOD_TOKENIZATION_TYPE_PAYMENT_GATEWAY)
    .addParameter("gateway", "cybersource")
    .addParameter("gatewayMerchantId", "[yourCybersourceMID]")
```

Example: JavaScript Code

```javascript
tokenizationType: 'PAYMENT_GATEWAY',
parameters: {
    gateway: 'cybersource',
    gatewayMerchantId: '[yourCybersourceMID]'
}
```

Formatting Payment Blobs

To prepare the google payload for submission to Cybersource, you must extract the token data element from the Google Pay payload and encode the token data element using Base64.

Sample Code

The following samples can be used to base64 encode payment responses:

JavaScript
let token = paymentData.paymentMethodData.tokenizationDta.token;
console.log(token);
var enc=window.btoa(token);

Android with Java
This sample uses the Android Studio base64 utility.

public static <outputString> encodeToString (byte[] <inputToken>, int DEFUALT)

Apple iPhone with Swift 3
This sample requires the Foundation utility.

extension String {
    func base64Encoded() -> <outputString>
    if let data = self.dat(using:.utf8) {
        return data.base64EncodedString()
    }
    return nil
}

Example: Unencrypted Google Pay Response

{"signature":"MEUCIQDhTkhHqyY8pXB9hpYxaSK5jFgsqG2E1rX77QXssK8tAIgUBvYYAI/bnBS8T/Tfxml2AF981Mv5y0pHyGexM5dMJk\u003d","protocolVersion":"ECv1","signedMessage":"\"encryptedMessage":\"\nodyUGGA7B+b1letYcJbS43AQFQJpWEFCN4UUExQ5LX0/\nXcLwKELxXcB95nMmmp091M2KGpl3FYsL768ccCzAkJBLGFY+
fugcJTcvkrUhCNsYx7hwf12BEsrweqJ3Ml7t5sIfrPaukRJeLDQQ4FxmTLW4Q9yP8vI3C+
tz2c+Z3oZZZi5oB9j8EFA2do1Fa13Cu6gXqKH/
IHRh7UnILUUtY+0G5FQV2pwST2uBSNnkZhb8WYJDHbxBjz0UebVP+
ObmT5cc8AKU5dgHRdfr4GKpeZ4EBzB90_BPxLqYHopriJ6IbFgFVsQQ6/
8HBq7ImIMH5y7G8p8qAFKwncn78ZcLOFh5BjXojxxGoF2pgjAsrhtttHAFbe3WQBupkwJu09/
6/\MyJpCSRpmHFOuF/\dj0SYjQ+xI0971CH2ec7jQRahISLWZ9DzuMvGKWPwu0CKnZxqTXQ=\n",\"ephemeralPublicKey":\"MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEnn4yjy0N6x1X08/8j7/
4jvmLJeyaQxLwP1FhjuTgIM9oCtPijZfI9so2QEOs2ZnVp3D0d13JYIDVe+396KkAQ==\n","tag":\"DRp cc+YQ33RNgsTcxztnJbMJnirbU5DW3dStjhfIwcm=\"\}"

Example: Base64 Encoded Google Pay Response

eyJzaWduYXRlcmUiOiJNJRVDVSDVEFEA4raEhxd1k4cFhCOWwhWXhhU0slakZnc3FwRzJ
FMXJYNzdRHWHzSz0QInVuJ2WV1BSS9ibkJTofFQvVgZ4bm0yQY5ODFjV5MHBIEU
d1eE0I1E1akxLMAzZCISnYbyb3RvY29sVmVyc2lvdjElcJfJaWdU2WNRZ
XNzYwdlijoie1wi2W5jcn1wdGVkTUVzc2Fnc2VwOi1wib2R5VUdHQtdC2JsbGV0WV你們
Y1M0M0FRVU2RSnBXRUZDTtRvXyVFVeFIETFqXC9Y0x3S0VsWGNCOTVtvW5tUE85bE0
yS0dwMTNGWXNMDzY4Y2NDekFQkdMWUyZnVn0pUyY3ZrclVoY05TeVhyN2h3ZjeYQk
Authorizing Payments

The following examples show you how to send payment authorization.
Authorizing a Payment

To request the authorization for Google Pay transaction:

1. Set the `encrypted_payment_data` field to the string value generated from the Full Wallet response.
2. Set the `payment_solution` field to 012.

Example: Authorization Request

```xml
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.121">
  <merchantID>demomerchant</merchantID>
  <merchantReferenceCode>demorefnum</merchantReferenceCode>
  <billTo>
    <firstName>James</firstName>
    <lastName>Smith</lastName>
    <street1>1295 Charleston Road</street1>
    <city>Test City</city>
    <state>CA</state>
    <postalCode>99999</postalCode>
    <country>US</country>
    <email>demo@example.com</email>
  </billTo>
  <purchaseTotals>
    <currency>USD</currency>
    <grandTotalAmount>5.00</grandTotalAmount>
  </purchaseTotals>
  <encryptedPayment>
    <data>ABCDEFabcdefABCDEFabcdef0987654321234567</data>
  </encryptedPayment>
  <card>
    <cardType>001</cardType>
  </card>
  <ccAuthService run="true"/>
  <paymentSolution>012</paymentSolution>
</requestMessage>
```
Example: Authorization Request

```xml
<c:replyMessage>
  <c:requestID>4465840340765000001541</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>

  <c:requestToken>Ahj/7wSR5C/41cd2fdAKakGLadfg5535r/ghx3Z90AoBj3u</c:requestToken>
    <c:currency>USD</c:currency>
  </c:purchaseTotals>
  <c:ccAuthReply>
    <c:reasonCode>100</c:reasonCode>
    <c:amount>5.00</c:amount>
    <c:authorizationCode>888888</c:authorizationCode>
    <c:avsCode>X</c:avsCode>
    <c:avsCodeRaw>I1</c:avsCodeRaw>
    <c:authorizedDateTime>2015-11-03T20:53:54Z</c:authorizedDateTime>
    <c:processorResponse>100</c:processorResponse>
    <c:reconciliationID>11267051CGJSMQDC</c:reconciliationID>
  </c:ccAuthReply>
  <c:token>
    <c:prefix>294672</c:prefix>
    <c:suffix>4397</c:suffix>
    <c:expirationMonth>08</c:expirationMonth>
    <c:expirationYear>2021</c:expirationYear>
  </c:token>
</c:replyMessage>
```