# Contents

Recent Revisions to This Document................................................................. 4
About This Guide.................................................................................................. 6
Introduction.......................................................................................................... 7
  Google Pay Overview.......................................................................................... 7
  Payment Network Tokens................................................................................... 7
    Requirements.................................................................................................... 7
  Supported Processors.......................................................................................... 8
  How Google Pay Works....................................................................................... 9
  Additional Services............................................................................................. 10
  Transaction Endpoints......................................................................................... 11
Formatting Encrypted Payment Data................................................................. 12
  Configuring Google Pay....................................................................................... 12
    Example: Java Code.......................................................................................... 12
    Example: JavaScript Code................................................................................. 12
  Formatting Payment Blobs.................................................................................. 13
    Sample Code.................................................................................................... 13
    Example: Unencrypted Google Pay Response................................................... 14
    Example: Base64-Encoded Google Pay Response............................................. 15
Authorizing Payments......................................................................................... 16
  Authorizing a Payment....................................................................................... 16
    Example: Authorization Request....................................................................... 17
    Example: Authorization Response..................................................................... 17
API Fields............................................................................................................ 19
Recent Revisions to This Document

22.02
This revision contains only editorial changes and no technical updates.

22.01
Added Diners Club and Discover as supported card types for Barclays. See Supported Processors (on page 8).

21.06
Bug update. Document erroneously said the GPN processor allows split-shipments.

21.05
Changed the name of the merchantURL field to merchantDomainName. See the API Field Reference for the Simple Order API.

21.04
Added the tokenSource_networkTokenOption field.

21.03
This revision contains only editorial changes and no technical updates.

21.02
This revision contains only editorial changes and no technical updates.

21.01
Added the merchantUrl field. See the API Field Reference for the Simple Order API.
20.04

Updated information about recurring payments. See Supported Processors (on page 8).

20.03

Added information about optional features. See Supported Processors (on page 8).
About This Guide

This section describes the audience and purpose of this guide, its use of conventions, and links to further information.

Audience and Purpose

This document is written for merchants who want to enable customers to use Google Pay to pay for in-app purchases. This document provides an overview of integrating the Google API and describes how to request the Cybersource API to process an authorization.

This document describes the Google Pay service and the Cybersource API. You must request the Google API to receive the customer’s encrypted payment data before requesting the Cybersource API to process the transaction.

Conventions

The following special statements are used in this document:

**Important:** An *Important* statement contains information essential to successfully completing a task or learning a concept.

**Warning:** A *Warning* contains information or instructions, which, if not heeded, can result in a security risk, irreversible loss of data, or significant cost in time or revenue or both.

Related Documentation

For further technical documentation, visit the Cybersource Technical Documentation Portal:


Customer Support

For support information about any service, visit the Support Center:

http://www.cybersource.com/support
Introduction

You can use the Cybersource platform 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 enables 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 must have:
• A Cybersource merchant evaluation account if you do not have one already: https://www.cybersource.com/register/

• A merchant account with a supported processor. See Supported Processors (on page 8).

• 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 the *Authorizations with Payment Network Tokens* guide. Recurring payments and split shipments are described in the *Credit Card Services* guide.

<table>
<thead>
<tr>
<th>Processor</th>
<th>Card Types</th>
<th>Optional Features</th>
</tr>
</thead>
<tbody>
<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>
</tbody>
</table>

**Related information**

- Authorizations with Payment Network Tokens Using the Simple Order API
- Credit Card Services Using the Simple Order API
How Google Pay Works

The following figure 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.

   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 guide.

Capture

A capture is 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

An authorized reversal is 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.

Related information

Credit Card Services Using the Simple Order API
Transaction Endpoints

The following endpoints are used with Google Pay:

- **Test**: http://ics2testa.ic3.com
- **Production**: http://ics2a.ic3.com
Formatting Encrypted 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
.wallet
    .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**

```javascript
let token = paymentData.paymentMethodData.tokenizationData.token;
console.log(token);
var enc=window.btoa(token);
```

**Android with Java**

This sample uses the Android Studio Base64 utility.

```java
public static <outputString> encodeToString (byte[] <inputToken>, int DEFAULT)
```

**Apple iPhone with Swift 3**

This sample requires the Foundation utility.

```swift
extension String {
    func base64Encod() -> <outputString>
        if let data = self.dat(using:.utf8) {
            return data.base64EncodedString()
        }
        return nil
}
```
Example: Unencrypted Google Pay Response

{"signature":"MEUCIQDhTxhHqwY8pXB9hpYxaSK5jFgsqpG2E1rx77QXsSK8tA1gUBvYYAI/
bnBS8T/Tfxnm2AF981Lv5y0pHyGexM5dMjK\u003d","protocolVersion":"ECv1","signedMessage":"\"encryptedMessage\":\"}
odyUGGA7B+b11etYcJbS43AQUFQJpWEFCN4UuUExQ5LX0/\nXcLwKElXcb95nMnmPO91M2Kgp13FYesL768ccCzAjBGLYF+
fugcJTcvkrUhcNSyXr7hwf12BEsrweqJM6I7Vs51frPAukRJeLDQG4FxmTLW49QyP8vIzC+
tz2c+Z3zozzI5oB9jE8fA2dolFa13Cu6gXqdKH/
IHRh7UuiLUuTy+0G5FQV2pwST2uBSNNkZhb8WYJDHbxBjz0UebVP+
ObmT5cc8AKU5dgHRdfr4GKpeZ4EBzB90_BPxLqYHpopiRJ61bFgFVsQQ6/
8HBqQ7IImIMH5y7G8p8qAFkWnB78ZcL0Ph5BjXojkxGoFp2gjAsrNhhttHAFbe3WQBuPkwJu09/
6\//MyJpCSrpMHFouF/\dj0SYjQ+xI0971CHZec7jQrAhISSLZ9ZkuMvGKPwpu0CKn2XqTXQ=
"","ephemeralPublicKey":
"MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEnn4yjy0N6x1X08/\8j7/}
4jvmLJCYAggXLwP1FhjuTgIM9oCtpPij2fI9so2QEOs2ZnVp3D0d13JY1DVe+396KkAQ==
","tag":"DRp cc+YQ33RNgsTcxtznJbMjnrbtn5D5W3dStjfhFiwc="}
Example: Base64-Encoded Google Pay Response

eyJzaWduYXR1cmU0Ij1NNXlOSS9idXIvMDAxNjMxMjY3ODEwNTkyOTc5NTc1LTMwMDMyMDIwLTI1NTYzNzg5MTQ2OTYwMDIuMC4wMjA3MjQ3Mjg0NjYwLTEuMTQ5NzQ5MzI5MjI1MzI2NzE5MTc0MDQ4OTc5OTc3MjU4NjIzNjU5MzYwNDc5NjIuOTAuanBnIn0.

Google Pay | Formatting Encrypted Payment Data | 15
Authorizing Payments

The following examples show you how to authorize payments.

Authorizing a Payment

To request an authorization for a Google Pay transaction:

1. Set the `encryptedPayment_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>
    <street>1295 Charleston Road</street>
    <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"/>
  <paymentNetworkToken>
    <requestorID>987654321plokijuhytfrdeswa</requestorID> <!-- Required only for Visa Platform Connect. -->
  </paymentNetworkToken>
  <paymentSolution>012</paymentSolution>
</requestMessage>
```

Example: Authorization Response

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

  <c:requestToken>Ahj/7wSR5C/4Icd2fdAKakGLadfg5535r/ghx3290AoBj3u</c:requestToken>
  <c:purchaseTotals>
    <c:currency>USD</c:currency>
  </c:purchaseTotals>
</c:replyMessage>
```
API Fields

See the API Field Reference guide for your API type.