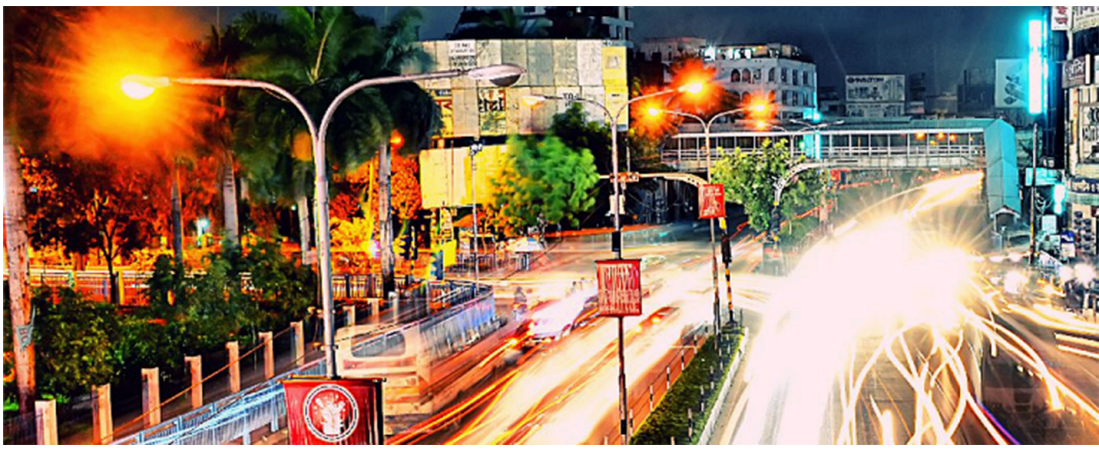
Google Pay

SCMP API

FDC Nashville Global

Developer Guide

cybersource
A Visa Solution



© 2021. Cybersource Corporation. All rights reserved.

Cybersource Corporation (Cybersource) furnishes this document and the software described in this document under the applicable agreement between the reader of this document (You) and Cybersource (Agreement). You may use this document and/or software only in accordance with the terms of the Agreement. Except as expressly set forth in the Agreement, the information contained in this document is subject to change without notice and therefore should not be interpreted in any way as a guarantee or warranty by Cybersource. Cybersource assumes no responsibility or liability for any errors that may appear in this document. The copyrighted software that accompanies this document is licensed to You for use only in strict accordance with the Agreement. You should read the Agreement carefully before using the software. Except as permitted by the Agreement, You may not reproduce any part of this document, store this document in a retrieval system, or transmit this document, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written consent of Cybersource.

Restricted Rights Legends

For Government or defense agencies: Use, duplication, or disclosure by the Government or defense agencies is subject to restrictions as set forth the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and in similar clauses in the FAR and NASA FAR Supplement.

For civilian agencies: Use, reproduction, or disclosure is subject to restrictions set forth in subparagraphs (a) through (d) of the Commercial Computer Software Restricted Rights clause at 52.227-19 and the limitations set forth in Cybersource Corporation's standard commercial agreement for this software. Unpublished rights reserved under the copyright laws of the United States.

Trademarks

Authorize.Net, eCheck.Net, and The Power of Payment are registered trademarks of Cybersource Corporation. Cybersource, Cybersource Payment Manager, Cybersource Risk Manager, Cybersource Decision Manager, and Cybersource Connect are trademarks and/or service marks of Cybersource Corporation. Visa, Visa International, Cybersource, the Visa logo, and the Cybersource logo are the registered trademarks of Visa International in the United States and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners.

Version: 21.05

Contents

Recent Revisions to This Document **4**
About This Guide **5**
Introduction **6**
Google Pay Overview 6
Payment Network Tokens 6
Requirements 6
Supported Processors 7
How Google Pay Works 8
Additional Services 9
Transaction Endpoints 9
Formatting Encrypted Payment Data **10**
Configuring Google Pay 10
Example: Java Code 10
Example: JavaScript Code 10
Formatting Payment Blobs 11
Sample Code 11
Example: Unencrypted Google Pay Response 12
Example: Base64-Encoded Google Pay Response 13
Authorizing Payments **14**
Authorizing a Payment 14
Example: Authorization Request 15
Example: Authorization Response 15
API Fields **16**

Recent Revisions to This Document

21.05

Changed the name of the **merchant_url** field to **merchant_domain_name**. See the API Field Reference for the SCMP API.

21.04

Added the **tms_network_token_option** 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 **merchant_url** field. See the API Field Reference for the SCMP API.

20.04

Updated information about recurring payments. See Supported Processors (on page 7).

20.03

Added information about optional features. See Supported Processors (on page 7).

About This Guide

This section provides you with information about the structure and content within this guide.

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:

https://docs.cybersource.com/en/index.html

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 7).
- 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.

Processor	Card Types	Optional Features
FDC Nashville Global	• American Express • Discover • Mastercard • Visa	• Multiple partial captures • Recurring Payments

Related information

Authorizations with Payment Network Tokens Using the SCMP API

Credit Card Services Using the SCMP 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 SCMP 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

```
.setPaymentMethodTokenizationType(WalletConstants.PAYMENT_METHOD_TOKENIZATION_TYPE_
PAYMENT_GATEWAY)
     .addParameter("gateway", "cybersource")
     .addParameter("gatewayMerchantId", "[yourCybersourceMID]")
```

Example: JavaScript Code

```
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":"MEUCIQDhTxhHqwY8pXB9hpYxaSK5jFgsqpG2E1rX77QXssK8tAIgUBvYYAI/
bnBS8T/Tfxnm2AF981Mv5y0pHyGexM5dMJk\u003d","protocolVersion":"ECv1","
signedMessage":"{\"encryptedMessage\":\"
odyUGGA7B+blletYcJbS43AQUFQJpWEFCN4UuUExQ5LX0\/
XcLwKElXcB95nMnmPO9lM2KGp13FYsL768ccCzAjBGLYF+
fugcJTcvkrUhcNSyXr7hwf12BEsrweqJM6I7Vs5lfrPAukRJeLDQG4FxmTLW49QyP8vIZC+
tz2c+Z3zozzI5oB9jE8fA2dolFa13Cu6gXqdKH\/
IHRh7UniLUuTy+0G5FQV2pwST2uBSNNkZhb8WYJDHbxBjz0UebVP+
ObmT5cc8AKU5dgHRdfr4GKpEZ4EBzB90 BPxLqYHpopriJ6lbFgFVsQQ6\/
8HBqQ7ImIMH5y7G8p8qAFkWnB78ZcL0Fh5BjXojkxGoFp2gjAsrhhttHAFbe3WQBuPkwJu09\/
6\/MyJpCSrpMHFouF\/dj0SYjQ+xI097lCHZec7jQrAhISLWZ9DZkuMvGKPWpu0CKn2XqTXQ=\
",\"ephemeralPublicKey\":\
"MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEnn4yjy0N6xlXO8\/8j7\/
4jvmLJCYAqgXLwP1FhjuTgIM9oCtPijZfI9so2QEOs2ZnVp3D0dl3JYIDVe+396KkAQ==\
",\"tag\":\"DRp cc+YQ33RNgsTcxztnJbMJnirbU5DW3dStjfhFiwc=\"}"}
```

Example: Base64-Encoded Google Pay Response

```
eyJzaWduYXR1cmUiOiJNRVVDSVFEaFR4aEhxd1k4cFhCOWhwWXhhU0s1akZnc3FwRzJ
FMXJYNzdRWHNzSzh0QUlnVUJ2WVlBSS9ibkJTOFQvVGZ4bm0yQUY5ODFNdjV5MHBIeU
dleE01ZE1Ka1x1MDAzZCIsInByb3RvY29sVmVyc2lvbiI6IkVDdjEiLCJzaWduZWRNZ
XNzYWdlIjoie1wiZW5jcnlwdGVkTWVzc2FnZVwiOlwib2R5VUdHQTdCK2JsbGV0WWNK
YlM0M0FRVUZRSnBXRUZDTjRVdVVFeFE1TFgwXC9YY0x3S0VsWGNCOTVuTW5tUE85bE0
yS0dwMTNGWXNMNzY4Y2NDekFqQkdMWUYrZnVnY0pUY3ZrclVoY05TeVhyN2h3ZjEyQk
VzcndlcUpNNkk3VnM1bGZyUEF1a1JKZUxEUUc0RnhtVExXNDlReVA4dklaQyt0ejJjK
1ozem96ekk1b0I5akU4ZkEyZG9sRmExM0N1NmdYcWRLSFwvSUhSaDdVbmlMVXVUeSsw
RzVGUVYycHdTVDJ1QlNOTmtaaGI4V1lKREhieEJqejBVZWJWUCtPYm1UNWNjOEFLVTV
kZ0hSZGZyNEdLcEVaNEVCekI5MEJQeExxWUhwb3ByaUo2bGJGZ0ZWc1FRNlwvOEhCcV
E3SW1JTUg1eTdHOHA4cUFGa1duQjc4WmNMMEZoNUJqWG9qa3hHb0ZwMmdqQXNyaGh0d
EhBRmJlM1dRQnVQa3dKdTA5XC82XC9NeUpwQ1NycE1IRm91RlwvZGowU1lqUSt4STA5
N2xDSFplYzdqUXJBaElTTFdaOURaa3VNdkdLUFdwdTBDS24yWHFUWFE9XCIsXCJlcGh
lbWVyYWxQdWJsaWNLZXlcIjpcIk1Ga3dFd1lIS29aSXpqMENBUVlJS29aSXpqMERBUW
NEUWdBRW5uNHlqeTBONnhsWE84XC84ajdcLzRqdm1MSkNZQXFnWEx3UDFGaGp1VGdJT
TlvQ3RQaWpaZkk5c28yUUVPczJablZwM0QwZGwzSllJRFZlKzM5NktrQVE9PVwiLFwi
dGFnXCI6XCJEUnBjYytZUTMzUk5nc1RjeHp0bkpiTUpuaXJiVTVEVzNkU3RqZmhGaXd
jPVwifSJ9
```

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 **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

```
bill_address1=111 S. Division St.
bill_address2=Suite 123
bill_city=Ann Arbor
bill_country=US
bill_state=MI
bill_zip=48104-2201
encrypted_payment_data=ABCDEFabcdefABCDEFabcdef0987654321234567
card_type=001
currency=usd
customer_email=demo@example.com
customer_firstname=James
customer_ipaddress=66.123.123.2
customer_lastname=Smith
customer_phone=999-999-9999
grand_total_amount=100.00
ics_applications=ics_auth
merchant_id=demomerchant
merchant_ref_number=demorefnum
payment_network_token_requestor_id=987654321plokijuhygtfrdeswa
solution_type=012
```

Example: Authorization Response

```
currency=usd
request_id=4465837560045000001541
auth_rflag=SOK
ics_rmsg=Request was processed successfully. auth_auth_amount=100.00
auth_rcode=1
auth_trans_ref_no=13209254CGJSMQCQ
auth_auth_code=888888
auth_rmsg=Request was processed successfully. ics_rflag=SOK
auth_auth_response=100
auth_avs_raw=I1
auth_auth_time=2015-11-03T204917Z
merchant_ref_number=demorefnum
ics_rcode=1
```

API Fields

See the API Field Reference guide for your API type.