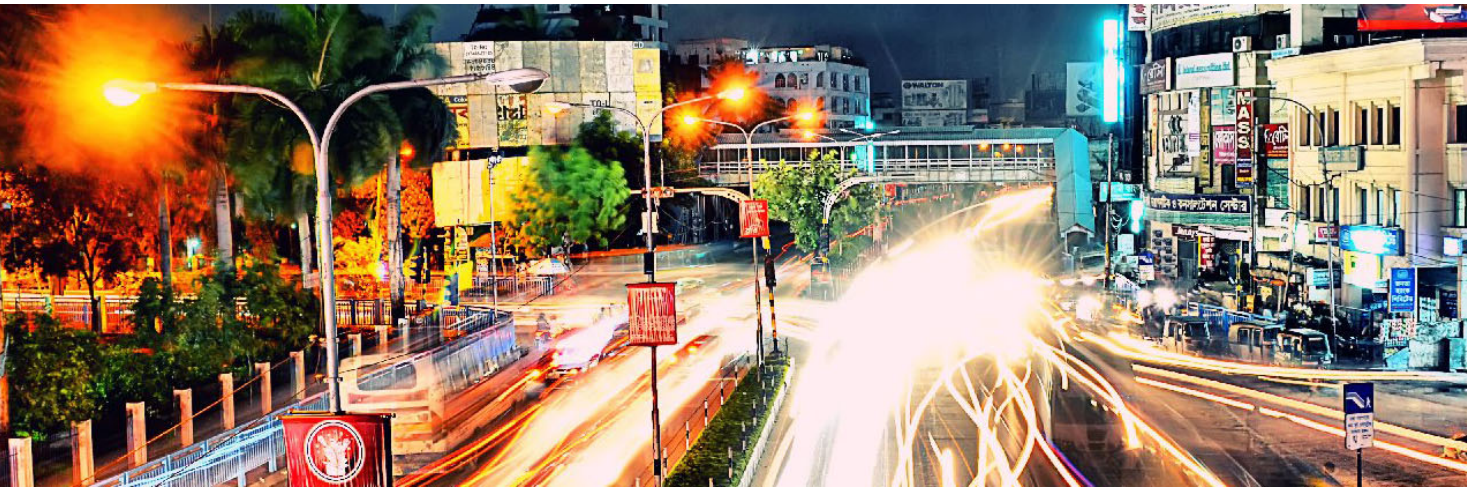# Chase Pay

Using the Simple Order API

CyberSource Contact Information

For general information about our company, products, and services, go to http://www.cybersource.com.

For sales questions about any CyberSource service, email sales@cybersource.com or call 650-432-7350 or 888-330-2300 (toll free in the United States).

For support information about any CyberSource service, visit the Support Center:
http://www.cybersource.com/support

Copyright

Restricted Rights Legends

Trademarks

Revision: May 2020

Contents

Recent Revisions to This Document

Release	Changes
May 2020	Updated information about recurring payments. See "Recurring Payments," page 12.
January 2020	Changed *payment network tokenization* to *authorizations with payment network tokens* throughout this document.
	Updated the requirements. See "Requirements," page 8.
	Updated the **purchaseTotals_grandTotalAmount** request field length. See purchaseTotals_grandTotalAmount, page 15.
May 2019	This revision contains only editorial changes and no technical updates.
July 2018	Updated information about recurring payments. See "Recurring Payments," page 12.
December 2017	This revision contains only editorial changes and no technical updates.
March 2016	Initial release.

About This Guide

Audience and Purpose

This document is written for merchants who want to enable customers to use Chase Pay to pay for in-app transactions. This document provides an overview of integrating Chase Pay and CyberSource services into an order management system and describes how to request the CyberSource API to process authorizations and recurring payments.

Conventions

Important Statements

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

Text and Command Conventions

Convention	Usage
bold	■ Field and service names in text; for example: Include the **ics_applications** field. ■ Items that you are instructed to act upon; for example: Click **Save**.
`screen text`	■ XML elements. ■ Code examples and samples. ■ Text that you enter in an API environment; for example: Set the **ics_applications** field to `ics_auth`.

Related Documents

CyberSource Documents:

- *Getting Started with CyberSource Advanced for the Simple Order API* (PDF | HTML)
- Simple Order API and SOAP Toolkit API Documentation and Downloads page
- *Credit Card Services Using the Simple Order API* (PDF | HTML)
- *Authorizations with Payment Network Tokens Using the Simple Order API* (PDF | HTML)

Chase Pay documents on the Chase Paymentech developer center:

- Chase Pay Service Specification
- Chase Pay Companion Guide

Refer to the Support Center for complete CyberSource technical documentation:

http://www.cybersource.com/support_center/support_documentation

Customer Support

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

http://www.cybersource.com/support

Introduction

Chase Pay

Chase Pay enables registered Chase customers to use credit cards that are stored in their digital wallet as payment methods when making purchases using their mobile device.

Chase Pay requires the customer to enter only a username and password to pay for goods. It eliminates the need to enter account, shipping, and billing information. The customer logs in to their Chase Pay account and chooses the card with which to pay.

To process Chase Pay transactions:

Step 1 Integrate the Chase Pay lightbox (iFrame).

Step 2 Request the Chase Pay API to retrieve the payment network token, the expiration date, the cryptogram, and other payment data associated with the transaction. See .

Step 3 Request the CyberSource authorization service and include the payment network token, the expiration date, the cryptogram, and other payment data associated with the transaction.

This method is best if your business has a fraud management solution or a records management system that requires payment data relating to transactions.

Authorizations with Payment Network Tokens

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

For in-app transactions, authorizations with payment network tokens use some of the CyberSource payer authentication request fields. This approach simplifies your implementation if your order management system already uses payer authentication. See *Authorizations with Payment Network Tokens Using the Simple Order API* (PDF | HTML).

Requirements

You must:

■   Create a CyberSource merchant evaluation account, if you do not have one already:

   https://www.cybersource.com/register/

■   Have a merchant account with Chase Paymentech Solutions. Chase Pay is supported only on Chase Paymentech Solutions.

■   Install the CyberSource Simple Order API client.

Supported Card Type

CyberSource supports the Visa card type for Chase Pay transactions.

How Chase Pay Works

1   You choose the Chase Pay button. A JavaScript tag embedded within the checkout page renders the button.

2   The browser sends a POST request for the purchase selection to the merchant web server.

3   Your (merchant) web server begins a MerchantSession request to the Chase Pay Services.

4   Chase Pay Services returns a Digital Session ID to the merchant web server, and it is used in all subsequent request messages.

5 Your web server returns the Digital Session ID to the browser, which sends a POST request to the Chase Pay site including the Digital Session ID.

6 The Digital Session ID is validated against the open session.

7 A lightbox is returned and displayed to the customer. The customer authenticates using their Chase Pay credentials, which initiates a session with their Chase Pay wallet.

8 The customer confirms their payment options and shipping preferences, and the lightbox closes. The browser sends a POST request to your web server, which includes the resulting Digital Session ID.

9 Your web server initiates a GetCheckoutData request to the Chase Pay Services and includes the Digital Session ID. The GetCheckoutData request retrieves the customer's payment and address information.

10 Chase Pay Services returns the requested data to your web server, including:

- Digital primary account number (DPAN).
- Cryptogram (authentication verification value).
- ECI (transaction type).
- Address information (optional).

11 Your web server formats and displays the payment confirmation page and displays it to the customer. The customer reviews and confirms their final payment details using the credentials and the address information that is displayed.

12 The browser sends a POST request to your web server.

13 Instead of including all of the normally required fields for an authorization request, you should include the following fields:

- card_accountNumber

- ccAuthService_cavv and ccAuthService_xid—set to the cryptogram (authentication verification value).

- paymentNetworkToken_transactionType

14 CyberSource processes the authorization, and the response is sent back to you. It is displayed to the customer, confirming the purchase.

Requesting the CyberSource Service

Authorization Service

Your payment processor can include API reply fields that are not documented in this guide. See *Credit Card Services Using the Simple Order API* for detailed descriptions of any additional API reply fields.

See "Request Fields," page 14, and "Reply Fields," page 16, for detailed descriptions of each API.

To request an authorization using a Visa card:

Step 1 Set the **card_accountNumber** field to the payment network token value.

Step 2 Set the **card_expirationMonth** and **card_expirationYear** fields to the payment network token expiration date fields.

Step 3 Set the **ccAuthService_cavv** field to the 3D Secure cryptogram of the payment network token.

> ⚠ For a 40-byte cryptogram, split the cryptogram into two 20-byte binary values (block A and block B). Send the first 20-byte value (block A) in the cardholder authentication verification value (CAVV) field. Send the second 20-byte value (block B) in the transaction ID (XID) field.

Step 4 Set the **ccAuthService_xid** field to the 3D Secure cryptogram of the payment network token.

Step 5 Set the **paymentNetworkToken_transactionType** field to `1`.

Step 6 Set the **ccAuthService_commerceIndicator** field to `internet`.

Step 7 Set the **paymentSolution** field to `007`.

Example 1   Authorization Request

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.121">
    <merchantID>demomerchant</merchantID>
    <merchantReferenceCode>demorefnum</merchantReferenceCode>
    <purchaseTotals>
        <currency>USD</currency>
        <grandTotalAmount>5.00</grandTotalAmount>
    </purchaseTotals>
    <card>
        <accountNumber>4650100000000839</accountNumber>
        <expirationMonth>12</expirationMonth>
        <expirationYear>2020</expirationYear>
        <cardType>001</cardType>
    </card>
    <ccAuthService run="true">
        <cavv>ABCDEFabcdefABCDEFabcdef0987654321234567</cavv>
        <commerceIndicator>internet</commerceIndicator>
        <xid>ABCDEFabcdefABCDEFabcdef0987654321234567</xid>
    </ccAuthService>
    <paymentNetworkToken>
        <transactionType>1</transactionType>
        <requestorID>1234567890</requestorID>
    </paymentNetworkToken>
    <paymentSolution>007</paymentSolution>
</requestMessage>
```

Example 2   Authorization Response

```
<c:replyMessage>
    <c:merchantReferenceCode>demorefnum</c:merchantReferenceCode>
    <c:requestID>4465840340765000001541</c:requestID>
    <c:decision>ACCEPT</c:decision>
    <c:reasonCode>100</c:reasonCode>
    <c:requestToken>Ahj/7wSR5C/4Icd2fdAKakGLadfg5535r/ghx3Z90AoBj3u</c:requestToken>
    <c:purchaseTotals>
        <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:replyMessage>
```

Recurring Payments

The recurring payments feature is described in *Credit Card Services Using the Simple Order API* (PDF | HTML).

API Fields

Data Type Definitions

For more information about these data types, see the World Wide Web Consortium (W3C) XML Schema Part 2: Datatypes Second Edition.

Table 1   Data Type Definitions

Data Type	Description
Date and time	Format is YYYY-MM-DDThh:mm:ssZ, where: ■ T separates the date and the time ■ Z indicates Coordinated Universal Time (UTC), also known as Greenwich Mean Time (GMT) **Example** 2020-01-11T22:47:57Z equals January 11, 2020, at 22:47:57 (10:47:57 p.m.).
Integer	Whole number {..., -3, -2, -1, 0, 1, 2, 3, ...}
String	Sequence of letters, numbers, spaces, and special characters

Request Fields

Unless otherwise noted, all field names are case sensitive and all fields accept special characters such as @, #, and %.

Table 2   Request Fields

Field	Description	Used By: Required (R) or Optional (O)	Data Type (Length)
card_accountNumber	The payment network token value. Populate this field with the decrypted DPAN value.	ccAuthService (R)	Nonnegative integer (20)
card_cardType	Type of card to authorize. Value: `001` for Visa	ccAuthService (R)	String (3)
card_cvNumber	CVN.	ccAuthService (R)	Nonnegative integer (4)
card_expirationMonth	Two-digit month in which the payment network token expires. Format: MM. Possible values: `01` through `12`.	ccAuthService (R)	String (2)
card_expirationYear	Four-digit year in which the payment network token expires. Format: YYYY.	ccAuthService (R)	Nonnegative integer (4)
ccAuthService_cavv	Cryptogram for payment network token transactions. The value for this field must be 28-character base64 or 40-character hex binary. All cryptograms use one of these formats.	ccAuthService (R)	String (40)
ccAuthService_ commerceIndicator	For a payment network token transaction. Value: `internet` for the Visa card type	ccAuthService (R)	String (20)
ccAuthService_run	Whether to include **ccAuthService** in your request. Possible values: ■ `true`: Include the service in your request. ■ `false` (default): Do not include the service in your request.	ccAuthService (R)	
ccAuthService_xid	Cryptogram for payment network token transactions. The value for this field must be 28-character base64 or 40-character hex binary. All cryptograms use one of these formats.	ccAuthService (R)	String (40)
merchantID	Your CyberSource merchant ID. Use the same merchant ID for evaluation, testing, and production.	ccAuthService (R)	String (30)
merchantReferenceCode	Merchant-generated order reference or tracking number. CyberSource recommends that you send a unique value for each transaction so that you can perform meaningful searches for the transaction. For information about tracking orders, see *Getting Started with CyberSource Advanced for the Simple Order API*.	ccAuthService (R)	String (50)

Table 2 Request Fields (Continued)

Field	Description	Used By: Required (R) or Optional (O)	Data Type (Length)
paymentNetworkToken_ transactionType	Type of transaction that provided the token data. This value does not specify the token service provider; it specifies the entity that provided you with information about the token. Set the value for this field to 1.	ccAuthService (R)	String (1)
paymentNetworkToken_ requestorID	Value that identifies your business and indicates that the cardholder's account number is tokenized. This value is assigned by the token service provider and is unique within the token service provider's database.	ccAuthService (R)	String (1)
paymentSolution	Identifies Chase Pay as the payment solution that is being used for the transaction: Set the value for this field to 007. **Note** This unique ID differentiates digital solution transactions within the CyberSource platform for reporting purposes.	ccAuthService (R)	String (3)
purchaseTotals_currency	Currency used for the order: Value: USD	ccAuthService (R)	String (5)
purchaseTotals_ grandTotalAmount	Grand total for the transaction. This value cannot be negative. You can include a decimal point (.), but you cannot include any other special characters. CyberSource truncates the amount to the correct number of decimal places.	ccAuthService (R)	String (15)

Reply Fields

⚠ Because CyberSource can add reply fields and reason codes at any time:

- You must parse the reply data according to the names of the fields instead of the field order in the reply. For more information about parsing reply fields, see the documentation for your client.

- Your error handler should be able to process new reason codes without problems.

- Your error handler should use the **decision** field to determine the result if it receives a reply flag that it does not recognize.

Your payment processor can include API reply fields that are not documented in this guide. See *Credit Card Services Using the Simple Order API* for detailed descriptions of additional API reply fields.

Table 3   Reply Fields

Field	Description	Returned By	Data Type & Length
ccAuthReply_amount	Amount that was authorized.	ccAuthReply	String (15)
ccAuthReply_authorizationCode	Authorization code. Returned only when the processor returns this value.	ccAuthReply	String (7)
ccAuthReply_authorizedDateTime	Time of authorization.	ccAuthReply	Date and time (20)
ccAuthReply_avsCode	AVS results. See *Credit Card Services Using the Simple Order API* for a detailed list of AVS codes.	ccAuthReply	String (1)
ccAuthReply_avsCodeRaw	AVS result code sent directly from the processor. Returned only when the processor returns this value.	ccAuthReply	String (10)
ccAuthReply_cvCode	CVN result code. See *Credit Card Services Using the Simple Order API* for a detailed list of CVN codes.	ccAuthReply	String (1)
ccAuthReply_cvCodeRaw	CVN result code sent directly from the processor. Returned only when the processor returns this value.	ccAuthReply	String (10)
ccAuthReply_processorResponse	For most processors, this is the error message sent directly from the bank. Returned only when the processor returns this value.	ccAuthReply	String (10)
ccAuthReply_reasonCode	Numeric value corresponding to the result of the credit card authorization request. See *Credit Card Services Using the Simple Order API* for a detailed list of reason codes.	ccAuthReply	Integer (5)

Table 3   Reply Fields (Continued)

Field	Description	Returned By	Data Type & Length
ccAuthReply_ reconciliationID	Reference number for the transaction. This value is not returned for all processors.	ccAuthReply	String (60)
decision	Summarizes the result of the overall request. Possible values: ■ ACCEPT ■ ERROR ■ REJECT ■ REVIEW: Returned only when you use CyberSource Decision Manager.	ccAuthReply	String (6)
invalidField_0...N	Fields in the request that contained invalid data. For information about missing or invalid fields, see *Getting Started with CyberSource Advanced for the Simple Order API*.	ccAuthReply	String (100)
merchantReferenceCode	Order reference or tracking number that you provided in the request. If you included multi-byte characters in this field in the request, the returned value might include corrupted characters.	ccAuthReply	String (50)
missingField_0...N	Required fields that were missing from the request. For information about missing or invalid fields, see *Getting Started with CyberSource Advanced for the Simple Order API*.	ccAuthReply	String (100)
purchaseTotals_currency	Currency used for the order. For the possible values, see the *ISO Standard Currency Codes*.	ccAuthReply	String (5)
reasonCode	Numeric value corresponding to the result of the overall request. See *Credit Card Services Using the Simple Order API* for a detailed list of reason codes.	ccAuthReply	Integer (5)
requestID	Identifier for the request.	ccAuthReply	String (26)
requestToken	Request token data created by CyberSource for each reply. The field is an encoded string that contains no confidential information such as an account or card verification number. The string can contain a maximum of 256 characters.	ccAuthReply	String (256)