Chase Pay POS Transactions

Using the Simple Order API

December 2016

CyberSource®
the power of payment

CyberSource Corporation HQ | P.O. Box 8999 | San Francisco, CA 94128-8999 | Phone: 800-530-9095

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

Copyright

Restricted Rights Legends

Trademarks

Contents

Recent Revisions to This Document

Release	Changes
December 2016	Initial release.

About This Guide

Audience and Purpose

This guide is written for merchants who want to enable customers to use Chase Pay point of sale (POS) with a mobile app. This document provides an overview of integrating Chase Pay POS and CyberSource services into an order management system and describes how to use the CyberSource API to request authorizations.

Implementing the CyberSource services requires software development skills. You must write code that uses the API request and reply fields to integrate the services into your existing order management system.

Conventions

The following special statements are used in this document:

Note A *Note* contains helpful suggestions or references to material not contained in this document.

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

The following text conventions are used in this document:

Table 1 Text Conventions

Convention	Meaning
bold	Field and service names in text; for example: Include the **ccAuthService_run** field.
`monospace`	■ XML elements ■ Code examples ■ Values for API fields; for example: Set the **ccAuthService_run** field to `true`.

Related Documentation

- *Getting Started with CyberSource Advanced for the Simple Order API* (PDF | HTML)
- *Card-Present Processing Using the Simple Order API* (PDF | HTML)

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

CHAPTER 1

Introduction to Chase Pay POS Transactions

Requirements

You must:

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

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

- Obtain a merchant category code from your Chase account manager, and then contact CyberSource Customer Support to add the merchant category code to your CyberSource account.

- Have a merchant account with Chase Paymentech Solutions.

- Install a CyberSource Simple Order API client.

Supported Card Type

CyberSource supports the Visa card type for Chase Pay POS transactions.

How Chase Pay POS Works

1. The customer uses your app and chooses Chase Pay.

2. Your app uses the Chase Pay SDK to authenticate the customer credentials for the Chase Pay wallet with the Chase Pay host, obtains the session ID, and retrieves a list of the credit cards that are in the customer's Chase Pay wallet.

3. Your app displays the list of credit cards, and the customer selects a card.

4. Your app sends the session ID, account index, and order details to your order management system.

5 Your order management system uses the session ID and account index to retrieve the payment details from the Chase Pay host.

6 Your order management system matches the customer's shopping cart to the customer and the payment details.

7 To process the payment, your order management system sends a request through CyberSource to Chase Paymentech Solutions. In addition to the basic fields required for retail chip card processing, you must also include the following fields in your request:

- paymentNetworkToken_requestorID—Set the value for this field to the value assigned by the token service provider.
- paymentNetworkToken_transactionType—Set the value for this field to `1`, which indicates that tokenized track data is included in the request.
- pos_entryMode—Set the value for this field to `QRCode`.
- pos_terminalCapability—Set the value for this field to `4`.
- payment_solution—Set the value for this field to `007`, which indicates that this is a Chase Pay transaction.

The preceding fields are described in Appendix B, "API Fields," on page 20.

8 In the reply message that you receive from CyberSource, the **ccAuthReply_paymentNetworkTransactionID** value is the Visa transaction ID. This value is returned only for approved transactions.

Testing

For the Chase Paymentech Solutions test data, see:

http://www.cybersource.com/developers/getting_started/test_and_manage/simple_order_api/paymentech/

Examples

Name-Value Pair Examples

Chase Pay POS Authorization

Example 1 Request Message: Chase Pay POS Authorization

```
ccAuthService_run=true
merchantID=JanesPlants
merchantReferenceCode=ABC123
purchaseTotals_currency=usd
purchaseTotals_grandTotalAmount=75.00
pos_entryMode=QRCode
pos_cardPresent=Y
pos_terminalCapability=4
pos_trackData=;4111111111111111=25122010123499999991?
pos_terminalID=87654321
card_cardType=001
ccAuthService_commerceIndicator=retail
emvRequest_combinedTags=9F330320400095050000000000009F3704238561349F10070
   6011103A000009F26088717A1A173EAA04D9F36020065820200009C01009F1A02084
   09A030006209F02060000000020005F2A0208409F0306000000000000
paymentNetworkToken_requestorID=12323453456
paymentNetworkToken_transactionType=1
paymentSolution=007
```

Example 2 Reply Message: Chase Pay POS Authorization

```
merchantReferenceCode=ABC123
requestID=0305782650000167905080
decision=ACCEPT
reasonCode=100
purchaseTotals_currency=usd
ccAuthReply_reasonCode=100
ccAuthReply_amount=75.00
ccAuthReply_authorizationCode=831000
ccAuthReply_avsCode=2
ccAuthReply_authorizedDateTime=2016-08-11T22:47:57Z
ccAuthReply_processorResponse=100
ccAuthReply_reconciliationID=1094820975023470
ccAuthReply_paymentNetworkTransactionID=0412MCCNYJPWY
emvReply_combinedTags=9F330320400095050000000000009F3704518823719F100706
   011103A000009F26081E1756ED0E2134E29F36020015820200009C01009F1A020840
   9A030006219F02060000000020005F2A0208409F0306000000000000
```

Chase Pay POS Capture

Example 3 Request Message: Chase Pay POS Capture

```
ccCaptureService_authRequestID=0305782650000167905080
merchantID=JanesPlants
merchantReferenceCode=ABC123
ccCaptureService_run=true
purchaseTotals_currency=usd
purchaseTotals_grandTotalAmount=75.00
```

Example 4 Reply Message: Chase Pay POS Capture

```
requestID=1019827520348290570029
merchantReferenceCode=ABC123
decision=ACCEPT
reasonCode=100
purchaseTotals_currency=usd
ccCaptureReply_reasonCode=100
ccCaptureReply_amount=75.00
ccCaptureReply_reconciliationID=482046C3A7E94F5BD1FE3C66C
```

Chase Pay POS Sale

Example 5 Request Message: Chase Pay POS Sale

```
merchantID=JanesPlants
merchantReferenceCode=ABC123
purchaseTotals_currency=usd
purchaseTotals_grandTotalAmount=75.00
pos_entryMode=QRCode
pos_cardPresent=Y
pos_terminalCapability=4
pos_trackData=;4111111111111111=25122010123499999991?
pos_terminalID=87654321
card_cardType=001
ccAuthService_run=true
ccAuthService_commerceIndicator=retail
ccCaptureService_run=true
mvRequest_combinedTags=9F3303204000950500000000009F3704238561349F100706
   011103A000009F26088717A1A173EAA04D9F3602006582020009C01009F1A020840
   9A030006209F02060000000020005F2A0208409F0306000000000000
paymentNetworkToken_requestorID=12323453456
paymentNetworkToken_transactionType=1
paymentSolution=007
```

Example 6 Reply Message: Chase Pay POS Sale

```
merchantReferenceCode=ABC123
requestID=0305782650000167905080
decision=ACCEPT
reasonCode=100
purchaseTotals_currency=usd
ccAuthReply_reasonCode=100
ccAuthReply_amount=75.00
ccAuthReply_authorizationCode=831000
ccAuthReply_avsCode=2
ccAuthReply_authorizedDateTime=2016-08-11T22:47:57Z
ccAuthReply_processorResponse=100
ccAuthReply_reconciliationID=1094820975023470
ccAuthReply_paymentNetworkTransactionID=0412MCCNYJPWY
ccAuthReply_cardCategory=J1
ccAuthReply_cardGroup=0
ccCaptureReply_reasonCode=100
ccCaptureReply_amount=75.00
ccCaptureReply_reconciliationID=1094820975023470
emvReply_combinedTags=9F3303204000950500000000009F3704518823719F1007060
   11103A000009F26081E1756ED0E2134E29F3602001582020009C01009F1A0208409
   A030006219F02060000000020005F2A0208409F0306000000000000
```

Chase Pay POS Authorization Reversal

Example 7 Request Message: Chase Pay POS Authorization Reversal

```
merchantID=JanesPlants
merchantReferenceCode=ABC123
purchaseTotals_currency=usd
purchaseTotals_grandTotalAmount=75.00
ccAuthReversalService_run=true
ccAuthReversalService_authRequestID=0305782650000167905080
```

Example 8 Reply Message: Chase Pay POS Authorization Reversal

```
merchantReferenceCode=ABC123
requestID=9827520348290570291013
decision=ACCEPT
reasonCode=100
purchaseTotals_currency=usd
ccAuthReversalReply_reasonCode=100
ccAuthReversalReply_amount=75.00
ccAuthReversalReply_processorResponse=100
```

Chase Pay POS Follow-On Credit

Example 9 Request Message: Chase Pay POS Follow-On Credit

```
merchantID=JanesPlants
merchantReferenceCode=ABC123
purchaseTotals_currency=usd
purchaseTotals_grandTotalAmount=75.00
ccCreditService_run=true
ccCreditService_captureRequestID=1019827520348290570293
```

Example 10 Reply Message: Chase Pay POS Follow-On Credit

```
merchantReferenceCode=ABC123
requestID=9057101982752034820293
decision=ACCEPT
reasonCode=100
purchaseTotals_currency=usd
ccCreditReply_reasonCode=100
ccCreditReply_amount=75.00
ccCreditReply_reconciliationID=C3A7E94F5BD1FE3C64820466C
```

XML Examples

Chase Pay POS Authorization

Example 11 Request Message: Chase Pay POS Authorization

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.128">
   <merchantID>JanesPlants</merchantID>
   <merchantReferenceCode>ABC123</merchantReferenceCode>
   <purchaseTotals>
      <currency>usd</currency>
      <grandTotalAmount>75.00</grandTotalAmount>
   </purchaseTotals>
   <pos>
      <entryMode>QRCode</entryMode>
      <cardPresent>Y</cardPresent>
      <terminalCapability>4</terminalCapability>
      <trackData>;4111111111111111=2512201012349999991?</trackData>
      <terminalID>87654321</terminalID>
   </pos>
   <card>
      <cardType>001</cardType>
   <card>
   <ccAuthService run="true">
      <commerceIndicator>retail</commerceIndicator>
   </ccAuthService>
   <emvRequest>
      <combinedTags>9F3303204000950500000000009F3704238561349F100706011103A000009F260
         88717A1A173EAA04D9F3602006582020009C01009F1A0208409A030006209F0206000000002
         0005F2A0208409F0306000000000000</combinedTags>
   </emvRequest>
   <paymentNetworkToken>
      <requestorID>12323453456</requestorID>
      <transactionType>1</transactionType>
   </paymentNetworkToken>
   <paymentSolution>007</paymentSolution>
</requestMessage>
```

Example 12 Reply Message: Chase Pay POS Authorization

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.128">
   <c:merchantReferenceCode>ABC123</c:merchantReferenceCode>
   <c:requestID>0305782650000167905080</c:requestID>
   <c:decision>ACCEPT</c:decision>
   <c:reasonCode>100</c:reasonCode>
   <c:purchaseTotals>
      <c:currency>usd</c:currency>
   </c:purchaseTotals>
   <c:ccAuthReply>
      <c:reasonCode>100</c:reasonCode>
      <c:amount>75.00</c:amount>
      <c:authorizationCode>831000</c:authorizationCode>
      <c:avsCode>2</c:avsCode>
      <c:authorizedDateTime>2016-08-11T22:47:57Z</c:authorizedDateTime>
      <c:processorResponse>100</c:processorResponse>
      <c:reconciliationID>1094820975023470</c:reconciliationID>
      <c:paymentNetworkTransactionID>0412MCCNYJPWY</c:paymentNetworkTransactionID>
   </c:ccAuthReply>
   <c:emvReply>
      <c:combinedTags>9F3303204000950500000000009F3704518823719F100706011103A000009F2
         6081E1756ED0E2134E29F3602001582020009C01009F1A0208409A030006219F02060000000
         020005F2A0208409F0306000000000000</c:combinedTags>
   </c:emvReply>
</c:replyMessage>
```

Chase Pay POS Capture

Example 13 Request Message: Chase Pay POS Capture

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.128">
   <merchantID>JanesPlants</merchantID>
   <merchantReferenceCode>ABC123</merchantReferenceCode>
   <purchaseTotals>
      <currency>usd</currency>
      <grandTotalAmount>75.00</grandTotalAmount>
   </purchaseTotals>
   <ccCaptureService run="true">
      <authRequestID>0305782650000167905080</authRequestID>
   </ccCaptureService>
</requestMessage>
```

Example 14 Reply Message: Chase Pay POS Capture

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.128">
   <c:merchantReferenceCode>ABC123</c:merchantReferenceCode>
   <c:requestID>1019827520348290570293</c:requestID>
   <c:decision>ACCEPT</c:decision>
   <c:reasonCode>100</c:reasonCode>
   <c:purchaseTotals>
      <c:currency>usd</c:currency>
   </c:purchaseTotals>
   <c:ccCaptureReply>
      <c:reasonCode>100</c:reasonCode>
      <c:amount>75.00</c:amount>
      <c:reconciliationID>482046C3A7E94F5BD1FE3C66C</c:reconciliationID>
   </c:ccCaptureReply>
</c:replyMessage>
```

Chase Pay POS Sale

Example 15 Request Message: Chase Pay POS Sale

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.128">
   <merchantID>JanesPlants</merchantID>
   <merchantReferenceCode>ABC123</merchantReferenceCode>
   <purchaseTotals>
      <currency>usd</currency>
      <grandTotalAmount>75.00</grandTotalAmount>
   </purchaseTotals>
   <pos>
      <entryMode>QRCode</entryMode>
      <cardPresent>Y</cardPresent>
      <terminalCapability>4</terminalCapability>
      <trackData>;4111111111111111=25122010123499999991?</trackData>
      <terminalID>87654321</terminalID>
   </pos>
   <card>
      <cardType>001</cardType>
   <card>
   <ccAuthService run="true">
      <commerceIndicator>retail</commerceIndicator>
   </ccAuthService>
   <ccCaptureService run="true"/>
   <emvRequest>
      <combinedTags>9F33032040009505000000000009F3704238561349F100706011103A000009F260
         88717A1A173EAA04D9F36020065820200009C01009F1A0208409A030006209F0206000000002
         0005F2A0208409F0306000000000000</combinedTags>
   </emvRequest>
   <paymentNetworkToken>
      <requestorID>12323453456</requestorID>
      <transactionType>1</transactionType>
   </paymentNetworkToken>
   <paymentSolution>007</paymentSolution>
</requestMessage>
```

Example 16 Reply Message: Chase Pay POS Sale

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.128">
   <c:merchantReferenceCode>ABC123</c:merchantReferenceCode>
   <c:requestID>0305782650000167905080</c:requestID>
   <c:decision>ACCEPT</c:decision>
   <c:reasonCode>100</c:reasonCode>
   <c:purchaseTotals>
      <c:currency>usd</c:currency>
   </c:purchaseTotals>
   <c:ccAuthReply>
      <c:reasonCode>100</c:reasonCode>
      <c:amount>75.00</c:amount>
      <c:authorizationCode>831000</c:authorizationCode>
      <c:avsCode>2</c:avsCode>
      <c:authorizedDateTime>2016-08-11T22:47:57Z</c:authorizedDateTime>
      <c:processorResponse>100</c:processorResponse>
      <c:reconciliationID>1094820975023470</c:reconciliationID>
      <c:paymentNetworkTransactionID>0412MCCNYJPWY</c:paymentNetworkTransactionID>
      <c:cardCategory>J1</c:cardCategory>
      <c:cardGroup>0</c:cardGroup>
   </c:ccAuthReply>
   <c:ccCaptureReply>
      <c:reasonCode>100</c:reasonCode>
      <c:amount>75.00</c:amount>
      <c:reconciliationID>1094820975023470</c:reconciliationID>
   </c:ccCaptureReply>
   <c:emvReply>
      <c:combinedTags>9F3303204000950500000000009F3704518823719F100706011103A000009F2
         6081E1756ED0E2134E29F36020015820200009C01009F1A0208409A030006219F02060000000
         020005F2A0208409F0306000000000000</c:combinedTags>
   </c:emvReply>
</c:replyMessage>
```

Chase Pay POS Authorization Reversal

Example 17 Request Message: Chase Pay POS Authorization Reversal

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.128">
   <merchantID>JanesPlants</merchantID>
   <merchantReferenceCode>ABC123</merchantReferenceCode>
   <purchaseTotals>
      <currency>usd</currency>
      <grandTotalAmount>75.00</grandTotalAmount>
   </purchaseTotals>
   <ccAuthReversalService run="true">
      <authRequestID>0305782650000167905080</authRequestID>
   </ccAuthReversalService>
</requestMessage>
```

Example 18 Reply Message: Chase Pay POS Authorization Reversal

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.128">
   <c:merchantReferenceCode>ABC123</c:merchantReferenceCode>
   <c:requestID>9827520348290570291013</c:requestID>
   <c:decision>ACCEPT</c:decision>
   <c:reasonCode>100</c:reasonCode>
   <c:purchaseTotals>
      <c:currency>usd</c:currency>
   </c:purchaseTotals>
   <c:ccAuthReversalReply>
      <c:reasonCode>100</c:reasonCode>
      <c:amount>75.00</c:amount>
      <c:processorResponse>100</c:processorResponse>
   </c:ccAuthReversalReply>
</c:replyMessage>
```

Chase Pay POS Follow-On Credit

Example 19 Request Message: Chase Pay POS Follow-On Credit

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.128">
   <merchantID>JanesPlants</merchantID>
   <merchantReferenceCode>ABC123</merchantReferenceCode>
   <purchaseTotals>
      <currency>usd</currency>
      <grandTotalAmount>75.00</grandTotalAmount>
   </purchaseTotals>
   <ccCreditService run="true">
      <captureRequestID>1019827520348290570293</authRequestID>
   </ccCreditService>
</requestMessage>
```

Example 20 Reply Message: Chase Pay POS Follow-On Credit

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.128">
   <c:merchantReferenceCode>ABC123</c:merchantReferenceCode>
   <c:requestID>9057101982752034820293</c:requestID>
   <c:decision>ACCEPT</c:decision>
   <c:reasonCode>100</c:reasonCode>
   <c:purchaseTotals>
      <c:currency>usd</c:currency>
   </c:purchaseTotals>
   <c:ccCreditReply>
      <c:reasonCode>100</c:reasonCode>
      <c:amount>75.00</c:amount>
      <c:reconciliationID>C3A7E94F5BD1FE3C64820466C</c:reconciliationID>
   </c:ccCreditReply>
</c:replyMessage>
```

APPENDIX

API Fields B

XML Schema Version

Use XML schema 1.128 or later.

Formatting Restrictions

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

Note

The values of the **item_#_** fields must not contain carets (^) or colons (:) because these characters are reserved for use by CyberSource services.

Values for request-level and item-level fields must not contain new lines or carriage returns. However, they can contain embedded spaces and any other printable characters. CyberSource removes all leading and trailing spaces.

Data Type Definitions

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

Data Type	Description
Integer	Whole number {..., -3, -2, -1, 0, 1, 2, 3, ...}
String	Sequence of letters, numbers, spaces, and special characters

Request Fields

Table 2 Request Fields

Field	Description	Used By: Required (R) or Optional (O)	Data Type & Length
billTo_city	City of the billing address.	ccAuthService (O)	String (50)
billTo_country	Country of the billing address. Use the two-character *ISO Standard Country Codes*.	ccAuthService (O)	String (2)
billTo_email	Customer's email address.	ccAuthService (O)	String (255)
billTo_firstName	Customer's first name. For a credit card transaction, this name must match the name on the card.	ccAuthService (O)	String (60)
billTo_lastName	Customer's last name. For a credit card transaction, this name must match the name on the card.	ccAuthService (O)	String (60)
billTo_phoneNumber	Customer's phone number. CyberSource recommends that you include the country code when the order is from outside the U.S.	ccAuthService (O)	String (15)
billTo_postalCode	Postal code for the billing address. The postal code must consist of 5 to 9 digits. When the billing country is the U.S., the 9-digit postal code must follow this format: [5 digits][dash][4 digits] Example: 12345-6789 When the billing country is Canada, the 6-digit postal code must follow this format: [alpha][numeric][alpha][space] [numeric][alpha][numeric] Example: A1B 2C3	ccAuthService (O)	String (10)
billTo_state	State or province of the billing address. Use the *State, Province, and Territory Codes for the United States and Canada*.	ccAuthService (O)	String (2)
billTo_street1	First line of the billing street address.	ccAuthService (O)	String (60)
billTo_street2	Additional address information. Example: Attention: Accounts Payable	ccAuthService (O)	String (60)
card_cardType	Type of card to authorize. Possible value: ▪ `001`: Visa	ccAuthService (R)	String (3)

Table 2 Request Fields (Continued)

Field	Description	Used By: Required (R) or Optional (O)	Data Type & Length
ccAuthService_commerceIndicator	Type of transaction. For a Chase Pay POS transaction, you must set this field to `retail`.	ccAuthService (R)	String (13)
ccAuthService_run	Set to `true` to request credit card authorization.	ccAuthService (R)	String (255)
emvRequest_combinedTags	The EMV data is in the tag-length-value format and includes chip card tags, terminal tags, and transaction detail tags. Include all tags required by Chase Pay POS as described in the *Chase Pay API Specification*, Version 0.94. For information about the individual tags, see the “Application Specification” section in the *EMV 4.3 Specifications*: http://emvco.com **Important** The following tags contain sensitive information and must not be included in this field: ■ 56: Track 1 equivalent data ■ 57: Track 2 equivalent data ■ 5A: Application PAN ■ 5F20: Cardholder name ■ 5F24: Application expiration date ■ 99: Transaction PIN ■ 9F0B: Cardholder name (extended) ■ 9F1F: Track 1 discretionary data ■ 9F20: Track 2 discretionary data	ccAuthService (R)	String (999)
item_#_productCode	Type of product. This value is used to identify the product category (electronic, handling, physical, service, or shipping). The default value is `default`. For a list of valid values, see the information about product codes in *Credit Card Services Using the Simple Order API*. When this field is not set to `default` or one of the values related to shipping and/or handling, the **item_#_quantity**, **item_#_productName**, and **item_#_productSKU** fields are required. For information about items and grand totals, see *Getting Started with CyberSource Advanced for the Simple Order API*.	ccAuthService (O)	String (30)
item_#_productName	Required when **item_#_productCode** is not `default` or one of the values related to shipping and/or handling.	ccAuthService (See description)	String (30)

Table 2 Request Fields (Continued)

Field	Description	Used By: Required (R) or Optional (O)	Data Type & Length
item_#_productSKU	Product identifier code. Required when **item_#_productCode** is not `default` or one of the values related to shipping and/or handling.	ccAuthService (See description)	String (15)
item_#_quantity	Default is `1`. Required when **item_#_productCode** is not `default` or one of the values related to shipping and/or handling.	ccAuthService (See description)	Integer (10)

item_#_taxAmount	Total tax to apply to the product. This value cannot be negative. The tax amount and the unit price must be in the same currency.
	The tax amount field is additive. The following example uses a two-exponent currency such as USD:
	1 You include the following items in your request:
	`item_0_unitPrice=10.00`
	`item_0_quantity=1`
	`item_0_taxAmount=0.80`
	`item_1_unitPrice=20.00`
	`item_1_quantity=1`
	`item_1_taxAmount=1.60`
	2 The total amount authorized will be 32.40, not 30.00 with 2.40 of tax included.
		ccAuthService (O)	String (15)

item_#_unitPrice	Per-item price of the product. You must include either this field or **purchaseTotals_grandTotalAmount** in your request. This value cannot be negative. For information about items and grand totals, see *Getting Started with CyberSource Advanced for the Simple Order API*. You can include a decimal point (.) in the value for this field, but you cannot include any other special characters. CyberSource truncates the amount to the correct number of decimal places.	ccAuthService (See description)	String (15)
merchantID	Your CyberSource merchant ID. Use the same merchant ID for evaluation, testing, and production.	ccAuthService (R)	String (30)

Table 2 Request Fields (Continued)

Field	Description	Used By: Required (R) or Optional (O)	Data Type & Length
merchantReference Code	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)
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 (O)	String (11)
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. Possible value: ■ `1`: In-app transaction. An application on the customer's mobile device provided the token data for an e-commerce transaction.	ccAuthService (R)	String (1)
paymentSolution	Digital solution identifier. Possible value: ■ `007`: Chase Pay	ccAuthService (R)	String (3)
pos_cardPresent	Indicates whether the card is present at the time of the transaction. Possible value: ■ `N`: Card is not present.	ccAuthService (R)	String (1)
pos_entryMode	Method of entering credit card information into the POS terminal. Possible value: ■ `QRCode`	ccAuthService (R)	String (11)
pos_terminalCapability	POS terminal's capability. Possible value: ■ `4`: Terminal can read chip cards.	ccAuthService (R)	Integer (1)

Table 2 Request Fields (Continued)

Field	Description	Used By: Required (R) or Optional (O)	Data Type & Length
pos_terminalID	Identifier for the terminal at your retail location. You can define this value yourself, but consult the processor for requirements.	ccAuthService (R)	String (8)
pos_trackData	Track 2 data. This value must include: ▪ Semicolon (;) for the start sentinel ▪ Equals sign (=) after the card account number ▪ Question mark (?) for the end sentinel Example: ;4111111111111111=25122010123499999991?	ccAuthService (R)	String (119)
purchaseTotals_currency	Currency used for the order. Use the three-character *ISO Standard Currency Codes*.	ccAuthService (R)	String (5)
purchaseTotals_grandTotalAmount	Grand total for the order. 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. You must include either this field or **item_#_unitPrice** in your request. For information about items and grand totals, *Getting Started with CyberSource Advanced for the Simple Order API*.	ccAuthService (See description)	String (15)
shipTo_city	City of shipping address.	ccAuthService (Required if any shipping address information is included in the request and shipping to the U.S. or Canada; otherwise, optional.)	String (50)
shipTo_country	Country of shipping address. Use the two-character *ISO Standard Country Codes*.	ccAuthService (Required if any shipping address information is included in the request; otherwise, optional.)	String (2)
shipTo_firstName	First name of the person receiving the shipment.	ccAuthService (O)	String (60)
shipTo_lastName	Last name of the person receiving the shipment.	ccAuthService (O)	String (60)

Table 2 Request Fields (Continued)

Field	Description	Used By: Required (R) or Optional (O)	Data Type & Length
shipTo_postalCode	Postal code for the shipping address. The postal code must consist of 5 to 9 digits. When the shipping country is the U.S., the 9-digit postal code must follow this format: [5 digits][dash][4 digits] Example: 12345-6789 When the shipping country is Canada, the 6-digit postal code must follow this format: [alpha][numeric][alpha][space][numeric][alpha][numeric] Example: A1B 2C3	ccAuthService (Required if any shipping address information is included in the request and shipping to the U.S. or Canada; otherwise, optional.)	String (10)
shipTo_state	State or province of shipping address. Use the *State, Province, and Territory Codes for the United States and Canada*.	ccAuthService (Required if any shipping address information is included in the request and shipping to the U.S. or Canada; otherwise, optional.)	String (2)
shipTo_street1	First line of shipping address.	ccAuthService (Required if any shipping address information is included in the request; otherwise, optional.)	String (60)
shipTo_street2	Second line of shipping address.	ccAuthService (O)	String (60)

Reply Fields

Important

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 reason code that it does not recognize.

Note

Your payment processor can include additional API reply fields that are not documented in this guide.

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. Format: YYYY-MM-DDThh:mm:ssZ Example: 2016-08-11T22:47:57Z equals August 11, 2016, at 22:47:57 (10:47:57 p.m.). The T separates the date and the time. The Z indicates UTC.	ccAuthReply	String (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_paymentNetworkTransactionID	Network transaction identifier (TID). You can use this value to identify a specific transaction when you are discussing the transaction with your processor.	ccAuthReply	String (15)

Table 3 Reply Fields (Continued)

Field	Description	Returned By	Data Type & Length
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)
ccAuthReply_ reconciliationID	Reference number for the transaction.	ccAuthReply	String (60)
ccAuthReversalReply_ processorResponse	For most processors, this is the error message sent directly from the bank. Returned only when the processor returns this value.	ccAuthReversalReply	String (10)
decision	Summarizes the result of the overall request. Possible values: ▪ `ACCEPT` ▪ `ERROR` ▪ `REJECT` ▪ `REVIEW`: Returned only when you use CyberSource Decision Manager. For details about these values, see the information about handling replies in *Getting Started with CyberSource Advanced for the Simple Order API*.	ccAuthReply	String (6)
emvReply_ combinedTags	The EMV data is in the tag-length-value format and includes chip card tags, terminal tags, and transaction detail tags. For information about the individual tags, see the “Application Specification” section in the *EMV 4.3 Specifications*: http://emvco.com	ccAuthReply	String (999)
invalidField_0...N	Fields in the request that have invalid data. For information about missing or invalid fields, see *Getting Started with CyberSource Advanced for the Simple Order API*. **Note** These fields are included as an aid to software developers only. Do not use these fields to interact with your customers.	ccAuthReply	String (100)

Table 3 Reply Fields (Continued)

Field	Description	Returned By	Data Type & Length
merchantReference Code	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*. **Note** These fields are included as an aid to software developers only. Do not use these fields to interact with your customers.	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 generated by the client.	ccAuthReply	String (26)